OSPF

この章では、Open Shortest Path First（OSPF）ルーティング プロトコルを使用してデータをルーティングし、認証を実行し、ルーティング情報を再配布するように ASA を設定する方法について説明します。

- OSPF について （1 ページ）
- OSPF のガイドライン （5 ページ）
- OSPFv2 の設定 （9 ページ）
- OSPFv2 ルータ ID の設定 （13 ページ）
- OSPF fast hello パケットの設定 （14 ページ）
- OSPFv2 のカスタマイズ （15 ページ）
- OSPFv3 の設定 （32 ページ）
- グレースフル リスタートの設定 （55 ページ）
- OSPFv2 の例 （61 ページ）
- OSPFv3 の例 （62 ページ）
- OSPF のモニタリング （63 ページ）
- OSPF の履歴 （67 ページ）

OSPF について

OSPF は、パスの選択にディスタンス ベクターではなくリンク ステートを使用する Interior Gateway Routing Protocol です。OSPF は、ルーティング テーブル更新ではなく、リンクステート アドバタイズメントを伝達します。ルーティング テーブル全体ではなく LSA だけが交換されるため、OSPF ネットワークは RIP ネットワークよりも迅速に収束します。

OSPF は、リンクステート アルゴリズムを使用して、すべての既知の接続先までの最短パスを構築し、計算します。OSPF エリア内の各ルータには、同一のリンクステート データベース（ルータが使用可能なインターフェイスおよび到達可能なネイバーの各一覧）が置かれています。

RIP と比べ OSPF には次の利点があります。

- OSPF では、リンクステート データベースの更新が RIP ほど頻繁に送信されません。また、ステート情報がタイムアウトすると、リンクステート データベースは徐々にではなく、すぐに更新されます。
- ルーティングはコスト、つまり特定のインターフェイスを介してパケットを送信するために必要なオーバーヘッドに基づいて決定されます。ASA は、インターフェイスのコストをリンク帯域幅に基づいて計算し、接続先までのホップ数は使用しません。コストを設定して優先パスを指定することができます。

最短パスを優先するアルゴリズムの欠点は、CPU サイクルとメモリが大量に必要になることです。

ASA は、OSPF プロトコルのプロセスを 2 つ同時に異なるインターフェイス セット上で実行できます。同じ IP アドレスを使用する複数のインターフェイス（NAT ではこのようなインターフェイスが共存可能ですが、OSPF ではアドレスは重複できません）がある場合に、2 つのプロセスを実行できます。あるいは、一方のプロセスを内部で実行しながら別のプロセスを外部で実行し、ルートのサブセットをこの 2 つのプロセス間で再配布することもできます。同様に、プライベート アドレスをパブリック アドレスから分離する必要がある場合もあります。

OSPF ルーティング プロセスには、別の OSPF ルーティング プロセスや RIP ルーティング プロセスから、または OSPF 対応インターフェイスに設定されているスタティック ルートおよび接続ルートから、ルートを再配布できます。

ASA では、次の OSPF の機能がサポートされています。

- エリア内ルート、エリア間ルート、および外部ルート（タイプ I とタイプ II）。
- 仮想リンク。
- LSA フラッディング。
- OSPF パケットの認証（パスワード認証と MD5 認証の両方）。
- ASA の代表ルータまたはバックアップ代表ルータとしての設定。ASA は、ABR として設定することもできます。
- スタブ エリアと Not-So-Stubby Area。
- エリア境界ルータのタイプ 3 LSA フィルタリング。

OSPF は、MD5 およびクリア テキスト ネイバー認証をサポートします。OSPF と他のプロトコル（RIP など）の間のルート再配布にあたっては、攻撃者によるルーティング情報の悪用の可能性があるため、できる限りすべてのルーティング プロトコルで認証を行う必要があります。

NAT を使用していて、OSPF がパブリック エリアおよびプライベート エリアで動作している場合、またアドレス フィルタリングが必要な場合は、2 つの OSPF プロセス（1 つはパブリック エリア用、1 つはプライベート エリア用）を実行する必要があります。

複数のエリアにインターフェイスを持つルータは、エリア境界ルータ（ABR）と呼ばれます。ゲートウェイとして動作し、OSPF を使用しているルータと他のルーティング プロトコルを使用しているルータ間でトラフィックを再配布するルータは、自律システム境界ルータ（ASBR）と呼ばれます。

ABR は LSA を使用して、使用可能なルートに関する情報を他の OSPF ルータに送信します。ABR タイプ 3 LSA フィルタリングを使用して、ABR として機能する ASA により、プライベートエリアとパブリック エリアを分けることができます。タイプ 3 LSA（エリア間ルート）は、プライベート ネットワークをアドバタイズしなくても NAT と OSPF を一緒に使用できるように、1 つのエリアから他のエリアにフィルタリングできます。

（注） フィルタリングできるのはタイプ 3 LSA のみです。プライベート ネットワーク内の ASBR として設定されている ASA は、プライベート ネットワークを記述するタイプ 5 LSA を送信しますが、これは AS 全体（パブリック エリアも含む）にフラッディングされます。

NAT が採用されているが、OSPF がパブリック エリアだけで実行されている場合は、パブリック ネットワークへのルートを、デフォルトまたはタイプ 5 AS 外部 LSA としてプライベート ネットワーク内で再配布できます。ただし、ASA により保護されているプライベート ネットワークにはスタティック ルートを設定する必要があります。また、同一の ASA インターフェイス上で、パブリック ネットワークとプライベート ネットワークを混在させることはできません。

ASA では、2 つの OSPF ルーティング プロセス（1 つの RIP ルーティング プロセスと 1 つの EIGRP ルーティング プロセス）を同時に実行できます。

fast hello パケットに対する OSPF のサポート

fast hello パケットに対する OSPF のサポートには、1 秒未満のインターバルで hello パケットの送信を設定する方法が用意されています。このような設定により、Open Shortest Path First（OSPF）ネットワークでのコンバージェンスがより迅速になります。

Fast Hello パケットに対する OSPF サポートの前提条件

OSPF がネットワークですでに設定されているか、Fast Hello パケット機能向けの OSPF のサポートと同時に設定される必要があります。

fast hello パケットに対する OSPF のサポートについて

次に、fast hello パケットに関する OSPF のサポートと、OSPF fast hello パケットの利点について説明します。

OSPF Hello インターバルと dead 間隔

OSPF hello パケットとは、OSPF プロセスがネイバーとの接続を維持するために OSPF ネイバーに送信するパケットです。hello パケットは、設定可能なインターバル（秒単位）で送信されます。デフォルトのインターバルは、イーサネット リンクの場合 10 秒、ブロードキャスト以外のリンクの場合 30 秒です。hello パケットには、dead 間隔中に受信したすべてのネイバーのリストが含まれます。dead 間隔も設定可能なインターバル（秒単位）で送信されます。デフォルトは Hello インターバルの値の 4 倍です。Hello インターバルの値は、ネットワーク内ですべて

同一にする必要があります。dead 間隔の値も、ネットワーク内ですべて同一にする必要があります。

この 2 つのインターバルは、リンクが動作していることを示すことにより、接続を維持するために連携して機能します。ルータが dead 間隔内にネイバーから hello パケットを受信しない場合、ルータはこのネイバーがダウンしていると判定します。

OSPF fast hello パケット

OSPF fast hello パケットとは、1 秒よりも短い間隔で送信される hello パケットのことです。fast hello パケットを理解するには、OSPF hello パケット インターバルと dead 間隔との関係についてあらかじめ理解しておく必要があります。OSPF Hello インターバルと dead 間隔 （3 ページ）を参照してください。

OSPF fast hello パケットは、ospf dead-interval コマンドで設定されます。dead 間隔は 1 秒に設定され、hello-multiplier の値は、その 1 秒間に送信する hello パケット数に設定されるため、1 秒未満の「fast」hello パケットになります。

インターフェイスで fast hello パケットが設定されている場合、このインターフェイスから送出される hello パケットでアドバタイズされる Hello インターバルは 0 に設定されます。このインターフェイス経由で受信した hello パケットの Hello インターバルは無視されます。

dead 間隔は、1 つのセグメント上で一貫している必要があり、1 秒に設定するか（fast hello パケットの場合）、他の任意の値を設定します。dead 間隔内に少なくとも 1 つの hello パケットが送信される限り、hello multiplier がセグメント全体で同じである必要はありません。

OSPF Fast Hello パケットの利点

OSPF Fast Hello パケット機能を利用すると、ネットワークがこの機能を使用しない場合よりも、コンバージェンス時間が短くなります。この機能によって、失われたネイバーを 1 秒以内に検出できるようになります。この機能は、ネイバーの損失がオープン システム相互接続（OSI）物理層またはデータリンク層で検出されないことがあっても、特に LAN セグメントで有効です。

OSPFv2 および OSPFv3 間の実装の差異

OSPFv3 には、OSPFv2 との後方互換性はありません。OSPF を使用して、IPv4 および IPv6 トラフィックの両方をルーティングするには、OSPFv2 および OSPFv3 の両方を同時に実行する必要があります。これらは互いに共存しますが、相互に連携していません。

OSPFv3 では、次の追加機能が提供されます。

- リンクごとのプロトコル処理。
- アドレッシング セマンティックの削除。
- フラッディング スコープの追加。
- リンクごとの複数インスタンスのサポート。
- ネイバー探索およびその他の機能に対する IPv6 リンクローカル アドレスの使用。

- プレフィックスおよびプレフィックス長として表される LSA。
- 2 つの LSA タイプの追加。
- 未知の LSA タイプの処理。
- RFC-4552 で指定されている OSPFv3 ルーティング プロトコル トラフィックの IPsec ESP 標準を使用する認証サポート。

OSPF のガイドライン

コンテキスト モードのガイドライン

OSPFv2 は、シングル コンテキスト モードとマルチ コンテキスト モードをサポートしています。

- デフォルトでは、共有インターフェイス間でのマルチキャスト トラフィックのコンテキスト間交換がサポートされていないため、OSPFv2 インスタンスは共有インターフェイス間で相互に隣接関係を形成できません。ただし、OSPFv2 プロセスの OSPFv2 プロセス設定で静的ネイバー設定を使用すると、共有インターフェイスでの OSPFv2 ネイバーシップを形成できます。
- 個別のインターフェイスでのコンテキスト間 OSPFv2 がサポートされています。

（ポイントツーポイント トポロジの場合）マルチコンテキストモードでは、スタティック OSPFv2 ネイバーを設定して、ポイントツーポイントの非ブロードキャスト ネットワークを介して OSPFv2 ルートをアドバタイズできます。OSPF 隣接関係を正常に形成するには、共有インターフェイスの OSPF を削除して再設定するときに、次の順序でコマンドを実行します。

- `no router <ospf name>` を使用して OSPF 設定を削除します。その後、`no ospf network point-to-point non-broadcast` を使用して、インターフェイスの OSPF ポイントツーポイント非ブロードキャスト設定を削除します。
- 共有インターフェイスで OSPF を再設定する場合は、`router <ospf name>` を使用して OSPF ルータを設定します。次に `ospf network point-to-point non-broadcast` でインターフェイスを設定します。

OSPFv3 は、シングル モードのみをサポートしています。

キー チェーン認証のガイドライン

OSPFv2 は、単一モードと複数モードの両方で、物理モードでも、仮想モードでも、キーチェーンの認証をサポートしています。ただし、複数モードでキーチェーンが設定できるのはコンテキスト モードのみです。

- 循環キーは OSPFv2 プロトコルにのみ適用されます。キー チェーンを使用した OSPF エリア認証はサポートされていません。

- OSPFv2 内に時間範囲がない既存の MD5 認証も、新しい循環キーとともにサポートされています。
- プラットフォームは SHA1 と MD5 の暗号化アルゴリズムをサポートしていますが、認証には MD5 暗号化アルゴリズムのみが使用されます。

ファイアウォール モードのガイドライン

OSPF は、ルーテッド ファイアウォール モードのみをサポートしています。OSPF は、トランスペアレント ファイアウォール モードをサポートしません。

フェイルオーバー ガイドライン

OSPFv2 および OSPFv3 は、ステートフル フェイルオーバー をサポートしています。

IPv6 のガイドライン

- OSPFv2 は IPv6 をサポートしません。
- OSPFv3 は IPv6 をサポートしています。
- OSPFv3 は、IPv6 を使用して認証を行います。
- ASA は、OSPFv3 ルートが最適なルートの場合、IPv6 RIB にこのルートをインストールします。
- OSPFv3 パケットは、**capture** コマンドの IPv6 ACL を使用してフィルタリングで除外できます。

OSPFv3 Hello パケットと GRE

通常、OSPF トラフィックは GRE トンネルを通過しません。IPv6 の OSPFv3 が GRE 内でカプセル化されている場合、マルチキャスト宛先などのセキュリティチェックで IPv6 ヘッダー検証が失敗します。このパケットは、宛先が IPv6 マルチキャストであるため、暗黙的なセキュリティチェックの検証でドロップされます。

GRE トラフィックをバイパスするプレフィルタルールを定義できます。ただし、プレフィルタルールでは、内部パケットはインスペクションエンジンによって問い合わせられません。

クラスタリングのガイドライン

- OSPFv3 暗号化はサポートされていません。クラスタリング環境で OSPFv3 暗号化を設定しようとすると、エラー メッセージが表示されます。
- スパンド インターフェイス モードでは、ダイナミック ルーティングは管理専用インターフェイスではサポートされません。
- 個別インターフェイスモードで、OSPFv2 または OSPFv3 ネイバーとして制御ユニットおよびデータユニットが確立されていることを確認します。

- 個別インターフェイスモードでは、OSPFv2 との隣接関係は、制御ユニットの共有インターフェイスの 2 つのコンテキスト間でのみ確立できます。スタティック ネイバーの設定は、ポイントツーポイン リンクでのみサポートされます。したがって、インターフェイスで許可されるのは 1 つのネイバー ステートメントだけです。
- クラスタで制御ロールの変更が発生した場合、次の挙動が発生します。
  - スパンド インターフェイス モードでは、ルータプロセスは制御ユニットでのみアクティブになり、データユニットでは停止状態になります。コンフィギュレーションが制御ユニットと同期されているため、各クラスタユニットには同じルータ ID があります。その結果、隣接ルータはロール変更時のクラスタのルータ ID の変更を認識しません。
  - 個別インターフェイス モードでは、ルータ プロセスはすべての個別のクラスタ ユニットでアクティブになります。各クラスタ ユニットは設定されたクラスタ プールから独自の個別のルータ ID を選択します。クラスタで制御ロールが変更されても、ルーティングトポロジは変更されません。

マルチプロトコル ラベル スイッチング（MPLS）と OSPF のガイドライン

MPLS 設定ルータから送信されるリンク ステート（LS）アップデート パケットに、Opaque Type-10 リンクステート アドバタイズメント（LSA）が含まれており、この LSA に MPLS ヘッダーが含まれている場合、認証は失敗し、アプライアンスはアップデート パケットを確認せずにサイレントにドロップします。ピア ルータは確認応答を受信していないため、最終的にネイバー関係を終了します。

ネイバー関係の安定を維持するため、ASA の Opaque 機能を無効にします。

```
router ospf process_ID_number
no nsf ietf helper
no capability opaque
```

（注） Firepower 4100/9300 モデルでは、複数の受信キュー間のロードバランシング不足のため、MPLS を使用した際に遅延が大きくなる可能性があります。

双方向フォワーディング検出（BFD）および OSPF に関する注意事項

- OSPFv2 および OSPFv3 インターフェイス（物理インターフェイス、サブインターフェイス、およびポートチャネル）で BFD を有効にできます。
- BFD は、VTI トンネル、DVTI トンネル、ループバック、スイッチポート、VNI、VTEP、および IRB インターフェイスではサポートされません。

ルートの再配布のガイドライン

- OSPFv2 の IPv4 プレフィックスリストを使用したルートマップの再配布はサポートされていません。ただし、IPv6 プレフィックスリストを使用した OSPFv3 でのルートマップの再

配布はサポートされていません。再配布に、OSPF のルートマップでアクセスリストを使用します。

- OSPF が、EIGRP ネットワークの一部であるデバイスで設定されている場合、またはその逆の場合は、ルートにタグを付けるように OSPF ルータが設定されていることを確認します（EIGRP はルートタグをまだサポートしていません）。

  OSPF を EIGRP に再配布し、EIGRP を OSPF に再配布する場合は、いずれかのリンクまたはインターフェイスで障害が発生したときや、ルート発信元がダウンしたときにも、ルーティングループが発生します。あるドメインから同じドメインに再度ルートを再配布することを避けるため、ルータは、再配布する際にドメインに属しているルートにタグ付けすることができます。そして、そのタグに基づいて、リモートルータでそれらのルートをフィルタ処理できます。それらのルートはルーティングテーブルにインストールされないため、再度同じドメインに再配布されることはありません。

その他のガイドライン

- OSPFv2 および OSPFv3 は 1 つのインターフェイス上での複数インスタンスをサポートしています。
- OSPFv3 は、非クラスタ環境での ESP ヘッダーを介した暗号化をサポートしています。
- OSPFv3 は非ペイロード暗号化をサポートします。
- OSPFv2 は RFC 4811、4812 および 3623 でそれぞれ定義されている、Cisco NSF グレースフル リスタートおよびIETF NSF グレースフル リスタート メカニズムをサポートします。
- OSPFv3 は RFC 5187 で定義されているグレースフル リスタート メカニズムをサポートします。
- 配布可能なエリア内（タイプ 1）ルートの数は限られています。これらのルートでは、1 つのタイプ 1 LSA にすべてのプレフィックスが含まれています。システムではパケットサイズが 35 KB に制限されているため、3000 ルートの場合、パケットがこの制限を超過します。2900 本のタイプ 1 ルートが、サポートされる最大数であると考えてください。
- ルートアップデートがリンク上の最小 MTU より大きい場合に、ルートアップデートがドロップされることによる隣接フラップを回避するには、リンクの両側のインターフェイスで同じ MTU を設定する必要があります。
- ASA Virtual は、Azure クラウドルーティングの特性により、EIGRP や OSPF などのダイナミックな内部ルーティングプロトコルを使用できません。有効なルーティング テーブルは、仮想クライアントにスタティック/ダイナミック ルートが設定されているかどうかに関係なく、ネクスト ホップを決定します。

  現在、有効なルーティングテーブルまたはシステム ルーティング テーブルはどちらも表示できません。
- パケットサイズが 8190 を超えた場合、OSPFv3 は LS アップデートをドロップします。その結果、隣接関係は終了します。そのため、「ospfv3 mtu-ignore」コマンドを使用してスイッチを設定し、隣接関係の終了を回避してください。

OSPFv2 の設定

ここでは、ASA で OSPFv2 プロセスを有効化する方法について説明します。

OSPFv2 をイネーブルにした後、ルート マップを定義する必要があります。詳細については、ルート マップの定義を参照してください。その後、デフォルト ルートを生成します。詳細については、スタティック ルートの設定を参照してください。

OSPFv2 プロセスのルート マップを定義した後で、ニーズに合わせてカスタマイズできます。ASA 上で OSPFv2 プロセスをカスタマイズする方法については、OSPFv2 のカスタマイズ （15 ページ）を参照してください。

OSPFv2 をイネーブルにするには、OSPFv2 ルーティング プロセスを作成し、このルーティング プロセスに関連付ける IP アドレスの範囲を指定し、さらにその IP アドレスの範囲にエリア ID を割り当てる必要があります。

最大 2 つの OSPFv2 プロセス インスタンスをイネーブルにできます。各 OSPFv2 プロセスには、独自のエリアとネットワークが関連付けられます。

OSPFv2 をイネーブルにするには、次の手順を実行します。

手順

ステップ 1 OSPF ルーティング プロセスを作成します。

router ospf *process_id*

例：

```
ciscoasa(config)# router ospf 2
```

process_id 引数は、このルーティング プロセス内部で使用される識別子です。任意の正の整数が使用できます。この ID は内部専用のため、他のどのデバイス上の ID とも照合する必要はありません。最大 2 つのプロセスが使用できます。

ASA 上で OSPF プロセスが 1 つしか有効化されていないと、そのプロセスがデフォルトで選択されます。既存のエリアを編集する場合、OSPF プロセス ID を変更できません。

ステップ 2 OSPF を実行する IP アドレスを定義し、そのインターフェイスのエリア ID を定義します。

network *ip_address mask* **area** *area_id*

例：

```
ciscoasa(config)# router ospf 2
ciscoasa(config-rtr)# network 10.0.0.0 255.0.0.0 area 0
```

新しいエリアを追加する場合、そのエリア ID を入力します。このエリア ID には、10 進数か IP アドレスを指定できます。有効な 10 進値の範囲は、0 ～ 4294967295 です。既存のエリアを編集する場合、エリア ID は変更できません。

認証用のキー チェーンの設定

デバイスのデータ セキュリティと保護を向上させるため、循環キーを有効にして IGP ピアを認証することができます。循環キーは、悪意のあるユーザーがルーティング プロトコル認証に使用されているキーを推測できないようにし、ネットワークによる誤ったルートのアドバタイズやトラフィックのリダイレクトを防ぎます。頻繁にキーを変更することで、推測されるリスクを最終的に軽減します。キー チェーンを提供するルーティング プロトコルの認証を設定する場合は、キー チェーン内でキーを設定してライフタイムを重複させます。こうすることによって、キーで保護された通信がアクティブなキーがないことよって損失することを防ぐために役立ちます。キーのライフタイムが切れ、アクティブなキーがなくなると、OSPF は最後に有効だったキーを使用してピアとの隣接関係を維持します。

この項では、OSPF ピア認証用のキーチェーンを作成する方法について説明します。キーチェーンオブジェクトを設定した後、それを使用して、インターフェイスおよび仮想リンクの OSPFv2 認証を定義することができます。隣接関係を正常に確立するには、ピアに対して同じ認証タイプ（MD5 またはキー チェーン）とキー ID を使用します。インターフェイスの認証を定義する方法についてはOSPFv2 インターフェイス パラメータの設定 （19 ページ）を参照してください。。

キー チェーンを設定するには、次のステップを実行します。

手順

ステップ 1 名前を使用してキー チェーンを設定します。

key chain*key-chain-name*

例：

```
ciscoasa(config)# key chain CHAIN1
ciscoasa(config-keychain)#
```

これで、キー チェーンの関連パラメータの定義に進むことができます。

ステップ 2 キー チェーンの識別子を設定します。

key*key-id*

キー ID の値には 0 ～ 255 を使用できます。無効なキーを通知する場合にのみ、値 0 を使用します。

例：

```
ciscoasa(config-keychain)# key 1
ciscoasa(config-keychain-key)#
```

ステップ 3 キー チェーンのキーまたはパスワードを設定します。

key-string [**0** | **8**] *key-string-text*

- 例に示すように、暗号化されていないパスワードが続くことを示すために **0** を使用します。
- 暗号化されたパスワードが続くことを示すには **8** を使用します。
- **key-string** には 80 文字を使用できますが、OSPF 認証の場合は最大 64 文字までです。
- パスワードは 10 文字以上必要です。また、数字の後に空白を含む文字列は使用できません。たとえば、「0 pass」や「1」は無効です。

例：

```
ciscoasa(config-keychain-key)# key-string 0 CHAIN1KEY1STRING
ciscoasa(config-keychain-key)#
```

ステップ 4 キー チェーンの暗号化アルゴリズムを設定します。

cryptographic-algorithm*md5*

暗号化認証アルゴリズムを指定する必要があります。プラットフォームは SHA1 と MD5 をサポートしていますが、キー チェーン管理でサポートしているのは MD5 のみです。

例：

```
ciscoasa(config-keychain-key)# cryptographic-algorithm md5
ciscoasa(config-keychain-key)#
```

ステップ 5 （オプション）キー チェーンのライフタイムを次のように設定します。

accept-lifetime [**local** | *start-time*] [**duration** *duration value* | **infinite** | *end-time*]

send-lifetime [**ocal** | *start-time*] [**duration** *duration value* | **infinite** | *end-time*]

別のデバイスとのキー交換時にキーを受け入れるか、または送信するための時間間隔をデバイスに指定できます。終了時刻は、期間、受け入れ/送信ライフタイムが終了する絶対時間、または無限です。

次に、開始と終了の値についての検証ルールを示します。

- 終了ライフタイムを指定した場合、開始ライフタイムを null にできません。
- 受け入れまたは送信のライフタイムの開始ライフタイムは、終了ライフタイムよりも前である必要があります。

例：

```
ciscoasa(config-keychain-key)# accept-lifetime 11:22:33 1 SEP 2018 infinite
ciscoasa(config-keychain-key)#
```

デバイスのスタートアップキー チェーン設定を表示するには、 **show key chain** コマンドを使用します。**show run key chain** コマンドを実行して、デバイスで現在実行されているキー チェーンの設定を表示します。

```
ciscoasa# show key chain
Key-chain CHAIN2:
    key 1 -- text "KEY1CHAIN2"
        accept lifetime (always valid) - (always valid) [valid now]
        send lifetime (always valid) - (always valid) [valid now]
  * key 2 -- text "(unset)"
        accept lifetime (11:00:12 UTC Sep 1 2018) - (11:12:12 UTC Sep 1 2018)
        send lifetime (always valid) - (always valid) [valid now]
Key-chain CHAIN1:
    key 1 -- text "CHAIN1KEY1STRING"
        accept lifetime (11:22:33 UTC Sep 1 2018) - (-1 seconds)
        send lifetime (always valid) - (always valid) [valid now]
ciscoasa#
```

```
ciscoasa# sh run key chain
key chain CHAIN2
 key 1
  key-string KEY1CHAIN2
  cryptographic-algorithm md5
 key 2
  accept-lifetime 11:00:12 Sep 1 2018 11:12:12 Sep 1 2018
  cryptographic-algorithm md5
key chain CHAIN1
 key 1
  key-string CHAIN1KEY1STRING
  accept-lifetime 11:22:33 Sep 1 2018 duration -1
  cryptographic-algorithm md5
ciscoasa# sh run key chain CHAIN1
key chain CHAIN1
 key 1
  key-string CHAIN1KEY1STRING
  accept-lifetime 11:22:33 Sep 1 2018 duration -1
  cryptographic-algorithm md5
ciscoasa#
```

次のタスク

これで、設定したキーチェーンを適用してインターフェイスのOSPFv2認証を定義できるようになりました。

- OSPFv2 インターフェイス パラメータの設定 （19 ページ）

OSPFv2 ルータ ID の設定

OSPF ルータ ID は、OSPF データベース内の特定のデバイスを識別するために使用されます。OSPF システム内の 2 台のルータが同じルータ ID を持つことはできません。

ルータ ID が OSPF ルーティングプロセスで手動で設定されていない場合、ルータはアクティブインターフェイスの最も高い IP アドレスから決定されたルータ ID を自動的に設定します。ルータ ID を設定すると、ルータに障害が発生するか、または OSPF プロセスがクリアされ、ネイバー関係が再確立されるまで、ネイバーは自動的に更新されません。

OSPF ルータ ID の手動設定

ここでは、ASA の OSPFv2 プロセスで router-id を手動で設定する方法について説明します。

手順

ステップ 1 固定ルータ ID を使用するには、**router-id** コマンドを使用します。

router-id *ip-address*

例：

```
ciscoasa(config-router)# router-id 193.168.3.3
```

ステップ 2 以前の OSPF ルータ ID の動作に戻すには、**no router-id** コマンドを使用します。

no router-id *ip-address*

例：

```
ciscoasa(config-router)# no router-id 193.168.3.3
```

移行中のルータ ID の挙動

ある ASA、たとえば ASA 1 から別の ASA、たとえば ASA 2 に OSPF 設定を移行すると、次のルータ ID 選択動作が見られます。

1. すべてのインターフェイスがシャットダウン モードの場合、ASA 2 は OSPF router-id に IP アドレスを使用しません。すべてのインターフェイスが「admin down」ステートまたはシャットダウン モードの場合に考えられる router-id の設定は次のとおりです。
   - ASA 2 に以前設定された router-id がない場合は、次のメッセージが表示されます。

     ```
     %OSPF: Router process 1 is not running, please configure a router-id
     ```

最初のインターフェイスが起動すると、ASA 2 はこのインターフェイスの IP アドレスをルータ ID として取得します。

- ASA 2 に router-id が以前設定されていて、「no router-id」コマンドが発行されたときにすべてのインターフェイスが「admin down」ステートになっていた場合、ASA 2 は古いルータ ID を使用します。ASA 2 は、「clear ospf process」コマンドが発行されるまで、起動されたインターフェイスの IP アドレスが変更されても、古いルータ ID を使用します。

2. ASA 2 に router-id が以前設定されていて、「no router-id」コマンドが発行されたときに少なくとも 1 つのインターフェイスが「admin down」ステートまたはシャットダウン モードになっていない場合、ASA 2 は新しいルータ ID を使用します。インターフェイスが「down/down」ステートの場合でも、ASA 2 はインターフェイスの IP アドレスから新しいルータ ID を使用します。

OSPF fast hello パケットの設定

ここでは、OSPF fast hello パケットを設定する方法について説明します。

手順

ステップ 1 インターフェイスを設定します。

interface *port-channel number*

例：

```
ciscoasa(config)# interface port-channel 10
```

number 引数は、ポートチャネル インターフェイスの番号を示します。

ステップ 2 少なくとも 1 個の hello パケットの受信が必要なインターバルを設定します。受信されなければ、ネイバーがダウンしていると判断されます。

ospf dead-interval minimal hello-multiplier *no.of times*

例：

```
ciscoasa(config-if)# ospf dead-interval minimal hell0-multiplier 5
ciscoasa
```

no.of times 引数は、毎秒送信される hello パケットの数を示します。有効な値は、3 ～ 20 です。

ここでは、minimal キーワードおよび hello-multiplier キーワードと値を指定することにより、fast hello パケットに対する OSPF のサポートがイネーブルになっています。multiplier キーワードが 5 に設定されているため、hello パケットが毎秒 5 回送信されます。

OSPFv2 のカスタマイズ

ここでは、OSPFv2 プロセスをカスタマイズする方法について説明します。

OSPFv2 へのルートの再配布

ASA は、OSPFv2 ルーティング プロセス間のルート再配布を制御できます。

（注） 指定されたルーティング プロトコルから、ターゲット ルーティング プロセスに再配布できるルートを定義することでルートを再配布する場合は、デフォルト ルートを最初に生成する必要があります。スタティック ルートの設定を参照し、その後にルート マップの定義に従ってルート マップを定義します。

スタティック ルート、接続されているルート、RIP ルート、または OSPFv2 ルートを OSPFv2 プロセスに再配布するには、次の手順を実行します。

手順

ステップ 1 OSPF ルーティング プロセスを作成します。

router ospf *process_id*

例：

```
ciscoasa(config)# router ospf 2
```

process_id 引数は、このルーティング プロセス内部で使用される識別子です。任意の正の整数が使用できます。この ID は内部専用のため、他のどのデバイス上の ID とも照合する必要はありません。最大 2 つのプロセスが使用できます。

ステップ 2 接続済みルートを OSPF ルーティング プロセスに再配布します。

redistribute connected [[**metric** *metric-value*] [**metric-type** {**type-1** | **type-2**}] [**tag** *tag_value*] [**subnets**] [**route-map** *map_name*]

例：

```
ciscoasa(config)# redistribute connected 5 type-1 route-map-practice
```

ステップ 3 スタティック ルートを OSPF ルーティング プロセスに再配布します。

redistribute static [**subnets**] [**route-map** *map_name*]

例：

```
ciscoasa(config)# redistribute static subnets
```

このコマンドは、すべてのスタティックルートを OSPF に渡します。選択的スタティックルートを再配布するには、スタティックルートを含むアクセスリストを作成してから、ルートマップに含めてください。

例：

```
ciscoasa(config)# ip access-list extended R1_Loopback
ciscoasa(config-ext-nacl)#permit ip host 1.1.1.1 any
ciscoasa(config-ext-nacl)#exit

ciscoasa(config)#route-map Permit_to_Distribute
ciscoasa(config-route-map)#match ip address R1_Loopback
ciscoasa(config-route-map)#exit
```

ルートマップを作成した後、次のように redistribute コマンドに含めます。

例：

```
ciscoasa(config)#router ospf 2
ciscoasa(config-router)#redistribute static subnets route-map Permit_to_Distribute
```

ステップ 4 ルートを OSPF ルーティング プロセスから別の OSPF ルーティング プロセスに再配布します。

redistribute ospf pid [**match {internal** | **external** [**1** | **2]** | **nssa-external** [**1** | **2**]}] [**metric** *metric-value*] [**metric-type** {**type-1** | **type-2**}] [**tag** *tag_value*] [**subnets**] [**route-map** *map_name*]

例：

```
ciscoasa(config)# route-map 1-to-2 permit
ciscoasa(config-route-map)# match metric 1
ciscoasa(config-route-map)# set metric 5
ciscoasa(config-route-map)# set metric-type type-1
ciscoasa(config-route-map)# router ospf 2
ciscoasa(config-rtr)# redistribute ospf 1 route-map 1-to-2
```

このコマンドの **match** オプションを使用して、ルート プロパティを照合および設定したり、ルート マップを使用したりできます。**subnets** オプションは、**route-map** コマンドで使用する場合と同じではありません。ルート マップと **redistribute** コマンドの **match** オプションの両方を使用する場合、これらは一致している必要があります。

この例では、ルートをメトリック 1 に照合することによる、OSPF プロセス 1 から OSPF プロセス 2 へのルートの再配布を示しています。ASA は、これらのルートをメトリック 5、メトリック タイプ 1 で外部 LSA として再配布します。

ステップ 5 ルートを RIP ルーティング プロセスから OSPF ルーティング プロセスに再配布します。

redistribute rip [**metric** *metric-value*] [**metric-type** {**type-1** | **type-2**}] [**tag** *tag_value*] [**subnets**] [**route-map** *map_name*]

例：

```
ciscoasa(config)# redistribute rip 5
ciscoasa(config-route-map)# match metric 1
ciscoasa(config-route-map)# set metric 5
ciscoasa(config-route-map)# set metric-type type-1
ciscoasa(config-rtr)# redistribute ospf 1 route-map 1-to-2
```

ステップ 6 ルートを EIGRP ルーティング プロセスから OSPF ルーティング プロセスに再配布します。

redistribute eigrp as-num [**metric** *metric-value*] [**metric-type** {**type-1** | **type-2**}] [**tag** *tag_value*] [**subnets**] [**route-map** *map_name*]

例：

```
ciscoasa(config)# redistribute eigrp 2
ciscoasa(config-route-map)# match metric 1
ciscoasa(config-route-map)# set metric 5
ciscoasa(config-route-map)# set metric-type type-1
ciscoasa(config-rtr)# redistribute ospf 1 route-map 1-to-2
```

OSPFv2 にルートを再配布する場合のルート集約の設定

他のプロトコルからのルートを OSPF に再配布する場合、各ルートは外部 LSA で個別にアドバタイズされます。その一方で、指定したネットワーク アドレスとマスクに含まれる再配布ルートすべてに対して 1 つのルートをアドバタイズするように ASA を設定することができます。この設定によって OSPF リンクステート データベースのサイズが小さくなります。

指定した IP アドレス マスク ペアと一致するルートは抑制できます。ルート マップで再配布を制御するために、タグ値を一致値として使用できます。

ルート サマリー アドレスの追加

ネットワーク アドレスとマスクに含まれる再配布ルートすべてに対して 1 つのサマリー ルートをアドバタイズするようにソフトウェアを設定するには、次の手順を実行します。

手順

ステップ 1 OSPF ルーティング プロセスを作成します。

router ospf *process_id*

例：

```
ciscoasa(config)# router ospf 1
```

process_id 引数は、このルーティング プロセス内部で使用される識別子です。任意の正の整数が使用できます。この ID は内部専用のため、他のどのデバイス上の ID とも照合する必要はありません。最大 2 つのプロセスが使用できます。

ステップ 2 サマリー アドレスを設定します。

summary-address *ip_address mask* [**not-advertise**] [tag *tag*]

例：

```
ciscoasa(config)# router ospf 1
ciscoasa(config-rtr)# summary-address 10.1.0.0 255.255.0.0
```

この例のサマリー アドレスの 10.1.0.0 には、10.1.1.0、10.1.2.0、10.1.3.0 などのアドレスが含まれます。外部のリンクステート アドバタイズメントでは、アドレス 10.1.0.0 だけがアドバタイズされます。

OSPFv2 エリア間のルート集約の設定

ルート集約は、アドバタイズされるアドレスを統合することです。この機能を実行すると、1 つのサマリー ルートがエリア境界ルータを通して他のエリアにアドバタイズされます。OSPF のエリア境界ルータは、ネットワークをある 1 つのエリアから別のエリアへとアドバタイズしていきます。あるエリアにおいて連続する複数のネットワーク番号が割り当てられている場合、指定された範囲に含まれるエリア内の個別のネットワークをすべて含むサマリー ルートをアドバタイズするようにエリア境界ルータを設定することができます。

ルート集約のアドレス範囲を定義するには、次の手順を実行します。

手順

ステップ 1 OSPF ルーティング プロセスを作成して、この OSPF プロセスのルータ コンフィギュレーション モードを開始します。

router ospf *process_id*

例：

```
ciscoasa(config)# router ospf 1
```

process_id 引数は、このルーティング プロセス内部で使用される識別子です。任意の正の整数が使用できます。この ID は内部専用のため、他のどのデバイス上の ID とも照合する必要はありません。最大 2 つのプロセスが使用できます。

ステップ 2 アドレス範囲を設定します。

area *area-id* range *ip-address mask* [**advertise** | **not-advertise**]

例：

```
ciscoasa(config-rtr)# area 17 range 12.1.0.0 255.255.0.0
```

この例では、アドレス範囲は OSPF エリア間で設定されます。

OSPFv2 インターフェイス パラメータの設定

必要に応じて一部のインターフェイス固有の OSPFv2 パラメータを変更できます。これらのパラメータを必ずしも変更する必要はありませんが、**ospf hello-interval**、**ospf dead-interval**、**ospf authentication-key** の各インターフェイス パラメータは、接続されているネットワーク内のすべてのルータで一致している必要があります。これらのパラメータを設定する場合は、ネットワーク上のすべてのルータで、コンフィギュレーションの値が矛盾していないことを確認してください。

OSPFv2 インターフェイス パラメータを設定するには、次の手順を実行します。

手順

ステップ 1 OSPF ルーティング プロセスを作成します。

router ospf*process-id*

例：

```
ciscoasa(config)# router ospf 2
```

process_id 引数は、このルーティング プロセス内部で使用される識別子で、任意の正の整数を使用できます。この ID は内部専用のため、他のどのデバイス上の ID とも照合する必要はありません。最大 2 つのプロセスが使用できます。

ステップ 2 OSPF を実行する IP アドレスを定義し、そのインターフェイスのエリア ID を定義します。

network*ip-address mask***area***area-id*

例：

```
ciscoasa(config)# router ospf 2
ciscoasa(config-rtr)# network 10.0.0.0 255.0.0.0 area 0
```

ステップ 3 インターフェイス コンフィギュレーション モードを開始します。

interface *interface-name*

例：

```
ciscoasa(config)# interface my_interface
```

ステップ 4 インターフェイスの認証タイプを指定します。

ospf authentication [**key-chain** *key-chain-name* | **message-digest** | **null**]

設定されているキー チェーン名を入力します。キー チェーンの設定については、次を参照してください。 認証用のキー チェーンの設定 （10 ページ）

例：

```
ciscoasa(config-interface)# ospf authentication message-digest
```

ステップ 5 OSPF 簡易パスワード認証を使用しているネットワーク セグメント上で近接する OSPF ルータが使用するパスワードを割り当てます。

ospf authentication-key*key*

例：

```
ciscoasa(config-interface)# ospf authentication-key cisco
```

key 引数には、最大 8 バイトの連続する文字列が指定できます。

このコマンドで作成するパスワードはキーとして使用され、このキーは ASA のソフトウェアによるルーティング プロトコル パケットの発信時に OSPF ヘッダーに直接挿入されます。各ネットワークにはインターフェイスごとに個別のパスワードを割り当てることができます。OSPF 情報を交換するには、同じネットワーク上のすべての隣接ルータが同じパスワードを持っている必要があります。

ステップ 6 OSPF インターフェイスでパケットを送信するコストを明示的に指定します。

ospf cost*cost*

例：

```
ciscoasa(config-interface)# ospf cost 20
```

cost は、1 ～ 65535 の整数です。

この例では、cost は 20 に設定されています。

ステップ 7 デバイスが hello パケットを受信していないためネイバー OSPF ルータがダウンしていることを宣言するまでデバイスが待機する秒数を設定します。

ospf dead-interval*seconds*

例：

```
ciscoasa(config-interface)# ospf dead-interval 40
```

この値はネットワーク上のすべてのノードで同じにする必要があります。

ステップ 8 ASA が OSPF インターフェイスから hello パケットを送信する時間間隔を指定します。

ospf hello-interval*seconds*

例：

```
ciscoasa(config-interface)# ospf hello-interval 10
```

この値はネットワーク上のすべてのノードで同じにする必要があります。

ステップ 9 OSPF Message Digest 5（MD5）認証を有効にします。

ospf message-digest-key*key-id***md5***key*

例：

```
ciscoasa(config-interface)# ospf message-digest-key 1 md5 cisco
```

次の引数を設定できます。

key-id：1 ～ 255 の範囲の識別子。

key：最大 16 バイトの英数字パスワード

通常は、インターフェイスあたり 1 つのキーを使用して、パケット送信時に認証情報を生成するとともに着信パケットを認証します。隣接ルータの同一キー識別子は、キー値を同一にする必要があります。

1 インターフェイスで 2 つ以上のキーを保持しないことをお勧めします。新しいキーを追加したらその都度古いキーを削除して、ローカル システムが古いキー情報を持つ悪意のあるシステムと通信を続けることのないようにしてください。古いキーを削除すると、ロールオーバー中のオーバーヘッドを減らすことにもなります。

ステップ 10 ネットワークに対して、OSPF で指定されたルータを判別するときに役立つプライオリティを設定します。

ospf priority *number-value*

例：

```
ciscoasa(config-interface)# ospf priority 20
```

number_value 引数の範囲は 0 ～ 255 です。

マルチコンテキストモードでは、共有インターフェイスに 0 を指定して、デバイスが指定ルータにならないようにします。OSPFv2 インスタンスは、共有インターフェイス間で相互に隣接関係を形成できません。

ステップ 11 OSPF インターフェイスに属する隣接ルータに LSA を再送信する間隔を秒単位で指定します。

ospf retransmit-interval *number-value*

例：

```
ciscoasa(config-interface)# ospf retransmit-interval seconds
```

seconds の値は、接続されているネットワーク上の任意の 2 ルータ間で予想されるラウンドトリップ遅延よりも長い秒数でなければなりません。範囲は 1 ～ 8192 秒です。デフォルト値は 5 秒です。

ステップ 12 OSPF インターフェイスでリンクステートアップデートパケットを送信するために必要な予想時間を秒単位で設定します。

ospf transmit-delay*seconds*

例：

```
ciscoasa(config-interface)# ospf transmit-delay 5
```

seconds の値は、1 ～ 8192 秒です。デフォルト値は 1 秒です。

ステップ 13 1 秒間に送信される hello パケットの数を設定します。

ospf dead-interval minimal hello-interval multiplier整数

例：

```
ciscoasa(config-if)# ospf dead-interval minimal hello-multiplier 6
```

有効な値は 3 ～ 20 の整数です。

ステップ 14 インターフェイスをポイントツーポイントの非ブロードキャストネットワークとして指定します。

ospf network point-to-point non-broadcast

例：

```
ciscoasa(config-interface)# ospf network point-to-point non-broadcast
```

インターフェイスをポイントツーポイントの非ブロードキャストとして指定するには、手動で OSPF ネイバーを定義する必要があります。ダイナミック ネイバー探索はできません。詳細については、「スタティック OSPFv2 ネイバーの定義 （27 ページ）」を参照してください。さらに、そのインターフェイスに定義できる OSPF ネイバーは 1 つだけです。

OSPFv2 エリア パラメータの設定

複数の OSPF エリア パラメータを設定できます。これらのエリア パラメータ（後述のタスクリストに表示）には、認証の設定、スタブ エリアの定義、デフォルト サマリー ルートへの特定のコストの割り当てがあります。認証では、エリアへの不正アクセスに対してパスワードベースで保護します。

スタブ エリアは、外部ルートの情報が送信されないエリアです。その代わりに、ABR で生成されるデフォルトの外部ルートがあり、このルートは自律システムの外部の宛先としてスタブ エリアに送信されます。OSPF スタブ エリアのサポートを活用するには、デフォルトのルーティングをスタブ エリアで使用する必要があります。スタブ エリアに送信される LSA の数をさらに減らすには、ABR で実行する **area stub** コマンドの **no-summary** キーワードを使用して、スタブ エリアにサマリー リンク アドバタイズメント（LSA タイプ 3）が送信されないようにします。

手順

ステップ 1 OSPF ルーティング プロセスを作成します。

router ospf *process_id*

例：

```
ciscoasa(config)# router ospf 2
```

process_id 引数は、このルーティング プロセス内部で使用される識別子です。任意の正の整数が使用できます。この ID は内部専用のため、他のどのデバイス上の ID とも照合する必要はありません。最大 2 つのプロセスが使用できます。

ステップ 2 OSPF エリアの認証を有効にします。

area *area-id* authentication

例：

```
ciscoasa(config-rtr)# area 0 authentication
```

ステップ 3 OSPF エリアの MD5 認証を有効にします。

area *area-id* authentication message-digest

例：

```
ciscoasa(config-rtr)# area 0 authentication message-digest
```

OSPFv2 フィルタ ルールの設定

OSPF アップデートで受信または送信されるルートまたはネットワークをフィルタリングするには、次の手順を実行します。

手順

ステップ 1 OSPF ルーティング プロセスを有効にし、ルータ コンフィギュレーション モードを開始します。

router ospf *process_id*

例：

```
ciscoasa(config)# router ospf 2
```

ステップ 2 着信 OSPF アップデートで受信したルートまたはネットワーク、あるいは発信 OSPF アップデートでアドバタイズされたルートまたはネットワークをフィルタリングします。

distribute-list *acl-number* **in** [**interface** *ifname*]

distribute-list *acl-number* **out** [*protocol process-number* | **connected** | **static**]

引数 *acl-number* には、IP アクセス リストの番号を指定します。アクセス リストは、ルーティング アップデートで受信されるネットワークと抑制されるネットワークを定義します。

着信アップデートにフィルタを適用するには、**in** を指定します。オプションで、インターフェイスを指定して、そのインターフェイスが受信するアップデートにフィルタを制限することができます。

発信アップデートにフィルタを適用するには、**out** を指定します。必要に応じて、配布リストに適用するプロトコル（**bgp**、**eigrp**、**ospf**、または **rip**）をプロセス番号付き（RIP を除く）で指定できます。ピアおよびネットワークが **connected** または **static** ルート経由で学習されたかどうかでフィルタすることもできます。

例：

```
ciscoasa(config-rtr)# distribute-list ExampleAcl in interface inside
```

OSPFv2 NSSA の設定

NSSA の OSPFv2 への実装は、OSPFv2 のスタブ エリアに似ています。NSSA は、タイプ 5 の外部 LSA をコアからエリアにフラッディングすることはありませんが、自律システムの外部ルートをある限られた方法でエリア内にインポートできます。

NSSA は、再配布によって、タイプ 7 の自律システムの外部ルートを NSSA エリア内部にインポートします。これらのタイプ 7 の LSA は、NSSA の ABR によってタイプ 5 の LSA に変換され、ルーティング ドメイン全体にフラッディングされます。変換中は集約とフィルタリングがサポートされます。

OSPFv2を使用する中央サイトから異なるルーティングプロトコルを使用するリモートサイトに接続しなければならない ISP またはネットワーク管理者は、NSSA を使用することによって管理を簡略化できます。

NSSA が実装される前は、企業サイトの境界ルータとリモート ルータ間の接続では、OSPFv2 スタブ エリアとしては実行されませんでした。これは、リモート サイト向けのルートは、スタブ エリアに再配布することができず、2 種類のルーティング プロトコルを維持する必要があったためです。RIP のようなシンプルなプロトコルを実行して再配布を処理する方法が一般的でした。NSSA が実装されたことで、企業ルータとリモート ルータ間のエリアを NSSA として定義することにより、NSSA で OSPFv2 を拡張してリモート接続をカバーできます。

この機能を使用する前に、次のガイドラインを参考にしてください。

- 外部の宛先に到達するために使用可能なタイプ 7 のデフォルト ルートを設定できます。設定すると、NSSA または NSSA エリア境界ルータまでのタイプ 7 のデフォルトがルータによって生成されます。
- 同じエリア内のすべてのルータは、エリアが NSSA であることを認識する必要があります。そうでない場合、ルータは互いに通信できません。

手順

ステップ 1 OSPF ルーティング プロセスを作成します。

router ospf *process_id*

例：

```
ciscoasa(config)# router ospf 2
```

process_id 引数は、このルーティング プロセス内部で使用される識別子です。任意の正の整数が使用できます。この ID は内部専用のため、他のどのデバイス上の ID とも照合する必要はありません。最大 2 つのプロセスが使用できます。

ステップ 2 NSSA エリアを定義します。

area *area-id* nssa [**no-redistribution**] [**default-information-originate**]

例：

```
ciscoasa(config-rtr)# area 0 nssa
```

ステップ 3 サマリー アドレスを設定します。これは、ルーティング テーブルのサイズを小さくするために役立ちます。

summary-address *ip_address mask* [**not-advertise**] [tag *tag*]

例：

```
ciscoasa(config-rtr)# summary-address 10.1.0.0 255.255.0.0
```

OSPF でこのコマンドを使用すると、このアドレスでカバーされる再配布ルートすべての集約として、1 つの外部ルートが OSPF ASBR からアドバタイズされます。

この例のサマリー アドレスの 10.1.0.0 には、10.1.1.0、10.1.2.0、10.1.3.0 などのアドレスが含まれます。外部のリンクステート アドバタイズメントでは、アドレス 10.1.0.0 だけがアドバタイズされます。

（注）
OSPF は summary-address 0.0.0.0 0.0.0.0 をサポートしません。

クラスタリングの IP アドレス プールの設定（OSPFv2 および OSPFv3）

個別インターフェイス クラスタリングを使用する場合は、ルータ ID のクラスタ プールの IPv4 アドレスの範囲を割り当てることができます。

OSPFv2 および OSPFv3 の個別インターフェイス クラスタリングのルータ ID のクラスタ プールの IPv4 アドレスの範囲を割り当てるには、次のコマンドを入力します。

手順

個別インターフェイス クラスタリングのルータ ID のクラスタ プールを指定します。

router-id cluster-pool hostname | A.B.C.D ip_pool

例：

```
hostname(config)# ip local pool rpool 1.1.1.1-1.1.1.4
hostname(config)# router ospf 1
hostname(config-rtr)# router-id cluster-pool rpool
hostname(config-rtr)# network 17.5.0.0 255.255.0.0 area 1
hostname(config-rtr)# log-adj-changes
```

cluster-pool キーワードは、個別インターフェイス クラスタリングが設定されている場合に、IP アドレス プールのコンフィギュレーションをイネーブルにします。**hostname**|**A.B.C.D.** キーワードは、この OSPF プロセスの OSPF ルータ ID を指定します。*ip_pool* 引数には、IP アドレス プールの名前を指定します。

（注）
クラスタリングを使用している場合は、ルータ ID の IP アドレス プールを指定する必要はありません。IP アドレス プールを設定しない場合、ASA は自動的に生成されたルータ ID を使用します。

スタティック OSPFv2 ネイバーの定義

ポイントツーポイントの非ブロードキャストネットワークを介してOSPFv2ルートをアドバタイズするには、スタティック OSPFv2 ネイバーを定義する必要があります。この機能により、OSPFv2 アドバタイズメントを GRE トンネルにカプセル化しなくても、既存の VPN 接続でブロードキャストすることができます。

開始する前に、OSPFv2 ネイバーに対するスタティック ルートを作成する必要があります。スタティック ルートの作成方法の詳細については、スタティック ルートの設定を参照してください。

手順

ステップ 1 OSPFv2 ルーティング プロセスを作成します。

router ospf *process_id*

例：

```
ciscoasa(config)# router ospf 2
```

process_id 引数は、このルーティング プロセス内部で使用される識別子です。任意の正の整数が使用できます。この ID は内部専用のため、他のどのデバイス上の ID とも照合する必要はありません。最大 2 つのプロセスが使用できます。

ステップ 2 OSPFv2 ネイバーフッドを定義します。

neighbor *addr* [**interface** if_name]

例：

```
ciscoasa(config-rtr)# neighbor 255.255.0.0 [interface my_interface]
```

addr 引数には OSPFv2 ネイバーの IP アドレスを指定します。*if_name* 引数は、ネイバーとの通信に使用するインターフェイスです。OSPFv2 ネイバーが直接接続されているインターフェイスのいずれとも同じネットワーク上にない場合、interface を指定する必要があります。

ルート計算タイマーの設定

OSPFv2 によるトポロジ変更受信と最短パス優先（SPF）計算開始との間の遅延時間が設定できます。最初に SPF を計算してから次に計算するまでの保持時間も設定できます。

手順

ステップ 1 OSPFv2 ルーティング プロセスを作成します。

router ospf *process_id*

例：

```
ciscoasa(config)# router ospf 2
```

process_id 引数は、このルーティング プロセス内部で使用される識別子です。任意の正の整数が使用できます。この ID は内部専用のため、他のどのデバイス上の ID とも照合する必要はありません。最大 2 つのプロセスが使用できます。

ステップ 2 ルート計算時間を設定します。

timers throttle spf *spf-start spf-hold spf-maximum*

例：

```
ciscoasa(config-router)# timers throttle spf 500 500 600
```

spf-start 引数は、OSPF によるトポロジ変更受信と SPF 計算開始との間の遅延時間（ミリ秒）です。0 ～ 600000 の整数に設定できます。

spf-hold 引数は、2 回の連続する SPF 計算間の最小時間（ミリ秒）です。0 ～ 600000 の整数に設定できます。

spf-maximum 引数は、2 回の連続する SPF 計算間の最大時間（ミリ秒）です。0 ～ 600000 の整数に設定できます。

ネイバーの起動と停止のロギング

デフォルトでは、OSPFv2 ネイバーがアップ状態またはダウン状態になったときに、syslog メッセージが生成されます。

アップ状態またはダウン状態になった OSPFv2 ネイバーについて、**debug ospf adjacency** コマンドを実行せずに確認する必要がある場合に、**log-adj-changes** コマンドを設定します。**log-adj-changes** コマンドでは、少ない出力によってピアの関係が高いレベルで表示されます。それぞれの状態変化メッセージを確認するには、**log-adj-changes detail** コマンドを設定します。

手順

ステップ 1 OSPFv2 ルーティング プロセスを作成します。

router ospf *process_id*

例：

```
ciscoasa(config)# router ospf 2
```

process_id 引数は、このルーティング プロセス内部で使用される識別子です。任意の正の整数が使用できます。この ID は内部専用のため、他のどのデバイス上の ID とも照合する必要はありません。最大 2 つのプロセスが使用できます。

ステップ 2 アップ状態またはダウン状態になったネイバーに対するロギングを設定します。

log-adj-changes [**detail**]

認証用のキー チェーンの設定

デバイスのデータ セキュリティと保護を向上させるため、循環キーを有効にして IGP ピアを認証することができます。循環キーは、悪意のあるユーザーがルーティング プロトコル認証に使用されているキーを推測できないようにし、ネットワークによる誤ったルートのアドバタイズやトラフィックのリダイレクトを防ぎます。頻繁にキーを変更することで、推測されるリスクを最終的に軽減します。キー チェーンを提供するルーティング プロトコルの認証を設定する場合は、キー チェーン内でキーを設定してライフタイムを重複させます。こうすることによって、キーで保護された通信がアクティブなキーがないことよって損失することを防ぐために役立ちます。キーのライフタイムが切れ、アクティブなキーがなくなると、OSPF は最後に有効だったキーを使用してピアとの隣接関係を維持します。

この項では、OSPF ピア認証用のキーチェーンを作成する方法について説明します。キーチェーンオブジェクトを設定した後、それを使用して、インターフェイスおよび仮想リンクの OSPFv2 認証を定義することができます。隣接関係を正常に確立するには、ピアに対して同じ認証タイプ（MD5 またはキー チェーン）とキー ID を使用します。インターフェイスの認証を定義する方法についてはOSPFv2 インターフェイス パラメータの設定 （19 ページ）を参照してください。。

キー チェーンを設定するには、次のステップを実行します。

手順

ステップ 1 名前を使用してキー チェーンを設定します。

key chain*key-chain-name*

例：

```
ciscoasa(config)# key chain CHAIN1
ciscoasa(config-keychain)#
```

これで、キー チェーンの関連パラメータの定義に進むことができます。

ステップ 2 キー チェーンの識別子を設定します。

key*key-id*

キー ID の値には 0 ～ 255 を使用できます。無効なキーを通知する場合にのみ、値 0 を使用します。

例：

```
ciscoasa(config-keychain)# key 1
ciscoasa(config-keychain-key)#
```

ステップ 3 キー チェーンのキーまたはパスワードを設定します。

key-string [**0** | **8**] *key-string-text*

- 例に示すように、暗号化されていないパスワードが続くことを示すために **0** を使用します。
- 暗号化されたパスワードが続くことを示すには **8** を使用します。
- **key-string** には 80 文字を使用できますが、OSPF 認証の場合は最大 64 文字までです。
- パスワードは 10 文字以上必要です。また、数字の後に空白を含む文字列は使用できません。たとえば、「0 pass」や「1」は無効です。

例：

```
ciscoasa(config-keychain-key)# key-string 0 CHAIN1KEY1STRING
ciscoasa(config-keychain-key)#
```

ステップ 4 キー チェーンの暗号化アルゴリズムを設定します。

cryptographic-algorithm*md5*

暗号化認証アルゴリズムを指定する必要があります。プラットフォームは SHA1 と MD5 をサポートしていますが、キー チェーン管理でサポートしているのは MD5 のみです。

例：

```
ciscoasa(config-keychain-key)# cryptographic-algorithm md5
ciscoasa(config-keychain-key)#
```

ステップ 5 （オプション）キー チェーンのライフタイムを次のように設定します。

accept-lifetime [**local** | *start-time*] [**duration** *duration value* | **infinite** | *end-time*]

send-lifetime [**ocal** | *start-time*] [**duration** *duration value* | **infinite** | *end-time*]

別のデバイスとのキー交換時にキーを受け入れるか、または送信するための時間間隔をデバイスに指定できます。終了時刻は、期間、受け入れ/送信ライフタイムが終了する絶対時間、または無限です。

次に、開始と終了の値についての検証ルールを示します。

- 終了ライフタイムを指定した場合、開始ライフタイムを null にできません。

- 受け入れまたは送信のライフタイムの開始ライフタイムは、終了ライフタイムよりも前である必要があります。

例：

```
ciscoasa(config-keychain-key)# accept-lifetime 11:22:33 1 SEP 2018 infinite
ciscoasa(config-keychain-key)#
```

デバイスのスタートアップキー チェーン設定を表示するには、**show key chain** コマンドを使用します。**show run key chain** コマンドを実行して、デバイスで現在実行されているキー チェーンの設定を表示します。

```
ciscoasa# show key chain
Key-chain CHAIN2:
    key 1 -- text "KEY1CHAIN2"
        accept lifetime (always valid) - (always valid) [valid now]
        send lifetime (always valid) - (always valid) [valid now]
  * key 2 -- text "(unset)"
        accept lifetime (11:00:12 UTC Sep 1 2018) - (11:12:12 UTC Sep 1 2018)
        send lifetime (always valid) - (always valid) [valid now]
Key-chain CHAIN1:
    key 1 -- text "CHAIN1KEY1STRING"
        accept lifetime (11:22:33 UTC Sep 1 2018) - (-1 seconds)
        send lifetime (always valid) - (always valid) [valid now]
ciscoasa#
```

```
ciscoasa# sh run key chain
key chain CHAIN2
 key 1
  key-string KEY1CHAIN2
  cryptographic-algorithm md5
 key 2
  accept-lifetime 11:00:12 Sep 1 2018 11:12:12 Sep 1 2018
  cryptographic-algorithm md5
key chain CHAIN1
 key 1
  key-string CHAIN1KEY1STRING
  accept-lifetime 11:22:33 Sep 1 2018 duration -1
  cryptographic-algorithm md5
ciscoasa# sh run key chain CHAIN1
key chain CHAIN1
 key 1
  key-string CHAIN1KEY1STRING
  accept-lifetime 11:22:33 Sep 1 2018 duration -1
  cryptographic-algorithm md5
ciscoasa#
```

次のタスク

これで、設定したキー チェーンを適用してインターフェイスのOSPFv2認証を定義できるようになりました。

- OSPFv2 インターフェイス パラメータの設定 （19 ページ）

OSPFv3 の設定

ここでは、OSPFv3 ルーティング プロセスの設定に関連するタスクについて説明します。

OSPFv3 の有効化

OSPFv3 をイネーブルにするには、OSPFv3 ルーティング プロセスを作成し、OSPFv3 用のエリアを作成して、OSPFv3 のインターフェイスをイネーブルにする必要があります。その後、ターゲットの OSPFv3 ルーティング プロセスにルートを再配布する必要があります。

手順

ステップ 1 OSPFv3 ルーティング プロセスを作成します。

ipv6 router ospf *process-id*

例：

```
ciscoasa(config)# ipv6 router ospf 10
```

process-id 引数は、このルーティング プロセス内部で使用されるタグです。任意の正の整数が使用できます。このタグは内部専用のため、他のどのデバイス上のタグとも照合する必要はありません。最大 2 つのプロセスが使用できます。

ステップ 2 インターフェイスをイネーブルにします。

interface *interface_name*

例：

```
ciscoasa(config)# interface Gigabitethernet0/0
```

ステップ 3 特定のプロセス ID を持つ OSPFv3 ルーティング プロセスおよび指定したエリア ID を持つ OSPFv3 のエリアを作成します。

ipv6 ospf *process-id* **area** *area_id*

例：

```
ciscoasa(config)# ipv6 ospf 200 area 100
```

OSPFv3 インターフェイス パラメータの設定

必要に応じて特定のインターフェイス固有の OSPFv3 パラメータを変更できます。これらのパラメータを必ずしも変更する必要はありませんが、hello interval と dead interval というインターフェイスパラメータは、接続されているネットワーク内のすべてのルータで一致している必要があります。これらのパラメータを設定する場合は、ネットワーク上のすべてのルータで、コンフィギュレーションの値が矛盾していないことを確認してください。

手順

ステップ 1 OSPFv3 のルーティング プロセスをイネーブルにします。

ipv6 router ospf *process-id*

例：

```
ciscoasa(config-if)# ipv6 router ospf 10
```

process-id 引数は、このルーティング プロセス内部で使用されるタグです。任意の正の整数が使用できます。このタグは内部専用のため、他のどのデバイス上のタグとも照合する必要はありません。最大 2 つのプロセスが使用できます。

ステップ 2 OSPFv3 エリアを作成します。

ipv6 ospf **area** [*area-num*] [**instance**]

例：

```
ciscoasa(config-if)# interface GigabitEthernet3/2.200
 vlan 200
 nameif outside
 security-level 100
 ip address 10.20.200.30 255.255.255.0 standby 10.20.200.31
 ipv6 address 3001::1/64 standby 3001::8
 ipv6 address 6001::1/64 standby 6001::8
 ipv6 enable
 ospf priority 255
 ipv6 ospf cost 100
 ipv6 ospf 100 area 10 instance 200
```

area-num 引数は、認証がイネーブルになるエリアであり、10 進数値または IP アドレスを指定できます。**instance** キーワードは、インターフェイスに割り当てられるエリア インスタンス ID を指定します。インターフェイスは、OSPFv3 エリアを 1 つだけ保有できます。複数のインターフェイスで同じエリアを使用でき、各インターフェイスは異なるエリア インスタンス ID を使用できます。

ステップ 3 インターフェイス上でパケットを送信するコストを指定します。

ipv6 ospf cost *interface-cost*

例：

```
ciscoasa(config-if)# interface GigabitEthernet3/2.200
 vlan 200
 nameif outside
 security-level 100
 ip address 10.20.200.30 255.255.255.0 standby 10.20.200.31
 ipv6 address 3001::1/64 standby 3001::8
 ipv6 address 6001::1/64 standby 6001::8
 ipv6 enable
 ospf priority 255
 ipv6 ospf cost 100
 ipv6 ospf 100 area 10 instance 200
```

interface-cost 引数は、リンクステート メトリックとして表される符号なし整数値を指定します。値の範囲は、1 ～ 65535 です。デフォルトのコストは帯域幅に基づきます。

ステップ 4 OSPFv3 インターフェイスへの発信 LSA をフィルタリングします。

ipv6 ospf database-filter all out

例：

```
ciscoasa(config-if)# interface GigabitEthernet3/2.200
 vlan 200
 nameif outside
 security-level 100
 ip address 10.20.200.30 255.255.255.0 standby 10.20.200.31
 ipv6 address 3001::1/64 standby 3001::8
 ipv6 address 6001::1/64 standby 6001::8
 ipv6 enable
 ospf priority 255
 ipv6 ospf cost 100
 ipv6 ospf 100 area 10 instance 200
ipv6 ospf database-filter all out
```

デフォルトでは、すべての発信 LSA がインターフェイスにフラッディングされます。

ステップ 5 秒単位で設定する期間内に hello パケットが確認されないと、当該ルータがダウンしていることがネイバーによって示されます。

ipv6 ospf dead-interval *seconds*

例：

```
ciscoasa(config-if)# interface GigabitEthernet3/2.200
 vlan 200
 nameif outside
 security-level 100
 ip address 10.20.200.30 255.255.255.0 standby 10.20.200.31
 ipv6 address 3001::1/64 standby 3001::8
 ipv6 address 6001::1/64 standby 6001::8
 ipv6 enable
 ospf priority 255
 ipv6 ospf cost 100
 ipv6 ospf 100 area 10 instance 200
ipv6 ospf dead-interval 60
```

この値はネットワーク上のすべてのノードで同じにする必要があります。値の範囲は、1 ～ 65535 です。デフォルト値は、**ipv6 ospf hello-interval** コマンドで設定された間隔の 4 倍です。

ステップ 6 インターフェイスに暗号化タイプを指定します。

ipv6 ospf encryption {**ipsec spi** *spi* **esp** *encryption-algorithm* [[*key-encryption-type*] *key*] *authentication-algorithm* [[*key-encryption-type*] *key* | **null**}

例：

```
ciscoasa(config-if)# interface GigabitEthernet3/2.200
 vlan 200
 nameif outside
 security-level 100
 ip address 10.20.200.30 255.255.255.0 standby 10.20.200.31
 ipv6 address 3001::1/64 standby 3001::8
 ipv6 address 6001::1/64 standby 6001::8
 ipv6 enable
 ospf priority 255
 ipv6 ospf cost 100
 ipv6 ospf 100 area 10 instance 200
ipv6 ospf encryption ipsec spi 1001 esp null sha1 123456789A123456789B123456789C123456789D
```

ipsec キーワードは、IP セキュリティ プロトコルを指定します。**spi** *spi* キーワード引数のペアは、セキュリティ ポリシー インデックスを指定します。値の範囲は 256 ～ 42949667295 である必要があり、10 進数で入力する必要があります。

esp キーワードは、カプセル化セキュリティ ペイロードを指定します。*encryption-algorithm* 引数は、ESP で使用される暗号化アルゴリズムを指定します。有効な値は次のとおりです。

- aes-cdc：AES-CDC 暗号化をイネーブルにします。
- 3des：トリプル DES 暗号化をイネーブルにします。
- des：DES 暗号化をイネーブルにします。
- null：暗号化なしの ESP を指定します。

key-encryption-type 引数に、次の 2 つのうちいずれかの値を指定します。

- 0：キーは暗号化されません。
- 7：キーは暗号化されます。

key 引数は、メッセージ ダイジェストの計算で使用される番号を指定します。この番号の長さは 32 桁の 16 進数（16 バイト）です。キーのサイズは、使用される暗号化アルゴリズムによって異なります。AES-CDC など、一部のアルゴリズムでは、キーのサイズを選択することができます。*authentication-algorithm* 引数は、使用される次のいずれかの暗号化認証アルゴリズムを指定します。

- md5：Message Digest 5（MD5）をイネーブルにします。
- sha1：SHA-1 をイネーブルにします。

null キーワードはエリアの暗号化より優先されます。

インターフェイスで OSPFv3 暗号化が有効化されており、ネイバーが異なるエリア（たとえば、エリア 0）にあり、ASA がそのエリアとの隣接関係を形成する場合は、ASA のエリアを変

更する必要があります。ASA のエリアを 0 に変更すると、OSPFv3 の隣接関係が確立される前に 2 分の遅延が発生します。

ステップ 7 インターフェイスに LSA のフラッディング削減を指定します。

ipv6 ospf flood-reduction

例：

```
ciscoasa(config-if)# interface GigabitEthernet3/2.200
 vlan 200
 nameif outside
 security-level 100
 ip address 10.20.200.30 255.255.255.0 standby 10.20.200.31
 ipv6 address 3001::1/64 standby 3001::8
 ipv6 address 6001::1/64 standby 6001::8
 ipv6 enable
 ospf priority 255
 ipv6 ospf cost 100
 ipv6 ospf 100 area 10 instance 200
ipv6 ospf flood reduction
```

ステップ 8 インターフェイス上で送信される hello パケット間の間隔（秒数）を指定します。

ipv6 ospf hello-interval *seconds*

例：

```
ciscoasa(config-if)# interface GigabitEthernet3/2.200
 vlan 200
 nameif outside
 security-level 100
 ip address 10.20.200.30 255.255.255.0 standby 10.20.200.31
 ipv6 address 3001::1/64 standby 3001::8
 ipv6 address 6001::1/64 standby 6001::8
 ipv6 enable
 ospf priority 255
 ipv6 ospf cost 100
 ipv6 ospf 100 area 10 instance 200
ipv6 ospf hello-interval 15
```

この値は特定のネットワーク上のすべてのノードで同じにする必要があります。値の範囲は、1 ～ 65535 です。デフォルトの間隔は、イーサネット インターフェイスで 10 秒、非ブロードキャスト インターフェイスで 30 秒です。

ステップ 9 DBD パケットを受信した場合の OSPF MTU 不一致検出をディセーブルにします。

ipv6 ospf mtu-ignore

例：

```
ciscoasa(config-if)# interface GigabitEthernet3/2.200
 vlan 200
 nameif outside
 security-level 100
 ip address 10.20.200.30 255.255.255.0 standby 10.20.200.31
 ipv6 address 3001::1/64 standby 3001::8
 ipv6 address 6001::1/64 standby 6001::8
 ipv6 enable
```

```
 ospf priority 255
 ipv6 ospf cost 100
 ipv6 ospf 100 area 10 instance 200
ipv6 ospf mtu-ignore
```

OSPF MTU 不一致検出は、デフォルトでイネーブルになっています。

ステップ 10 ネットワーク タイプに依存するデフォルト以外のタイプに OSPF ネットワーク タイプを設定します。

ipv6 ospf network {**broadcast** | **point-to-point non-broadcast**}

例：

```
ciscoasa(config-if)# interface GigabitEthernet3/2.200
 vlan 200
 nameif outside
 security-level 100
 ip address 10.20.200.30 255.255.255.0 standby 10.20.200.31
 ipv6 address 3001::1/64 standby 3001::8
 ipv6 address 6001::1/64 standby 6001::8
 ipv6 enable
 ospf priority 255
 ipv6 ospf cost 100
 ipv6 ospf 100 area 10 instance 200
ipv6 ospf network point-to-point non-broadcast
```

point-to-point non-broadcast キーワードは、ネットワーク タイプをポイントツーポイント、非ブロードキャストに設定します。**broadcast** キーワードは、ネットワーク タイプをブロードキャストに設定します。

ステップ 11 ルータ プライオリティを設定します。これは、ネットワークにおける指定ルータの特定に役立ちます。

ipv6 ospf priority *number-value*

例：

```
ciscoasa(config-if)# interface GigabitEthernet3/2.200
 vlan 200
 nameif outside
 security-level 100
 ip address 10.20.200.30 255.255.255.0 standby 10.20.200.31
 ipv6 address 3001::1/64 standby 3001::8
 ipv6 address 6001::1/64 standby 6001::8
 ipv6 enable
 ospf priority 255
 ipv6 ospf cost 100
 ipv6 ospf 100 area 10 instance 200
ipv6 ospf priority 4
```

有効値の範囲は 0 ～ 255 です。

ステップ 12 非ブロードキャスト ネットワークへの OSPFv3 ルータの相互接続を設定します。

ipv6 ospf neighbor *ipv6-address* [**priority** *number*] [**poll-interval** *seconds*] [**cost** *number*] [**database-filter all out**]

例：

```
ciscoasa(config-if)# interface GigabitEthernet3/2.200
 vlan 200
 nameif outside
 security-level 100
 ip address 10.20.200.30 255.255.255.0 standby 10.20.200.31
 ipv6 address 3001::1/64 standby 3001::8
 ipv6 address 6001::1/64 standby 6001::8
 ipv6 enable
 ospf priority 255
 ipv6 ospf cost 100
 ipv6 ospf 100 area 10 instance 200
ipv6 ospf neighbor FE80::A8BB:CCFF:FE00:C01
```

ステップ 13 インターフェイスに属する隣接関係の LSA 再送信間の時間を秒単位で指定します。

ipv6 ospf retransmit-interval *seconds*

例：

```
ciscoasa(config-if)# interface GigabitEthernet3/2.200
 vlan 200
 nameif outside
 security-level 100
 ip address 10.20.200.30 255.255.255.0 standby 10.20.200.31
 ipv6 address 3001::1/64 standby 3001::8
 ipv6 address 6001::1/64 standby 6001::8
 ipv6 enable
 ospf priority 255
 ipv6 ospf cost 100
 ipv6 ospf 100 area 10 instance 200
ipv6 ospf retransmit-interval 8
```

接続ネットワーク上の任意の2台のルータ間で想定される往復遅延より大きな値にする必要があります。有効な値の範囲は、1 ～ 65535 秒です。デフォルトは 5 秒です。

ステップ 14 インターフェイス上でリンクステート更新パケットを送信する時間を秒単位で設定します。

ipv6 ospf transmit-delay *seconds*

例：

```
ciscoasa(config-if)# interface GigabitEthernet3/2.200
 vlan 200
 nameif outside
 security-level 100
 ip address 10.20.200.30 255.255.255.0 standby 10.20.200.31
 ipv6 address 3001::1/64 standby 3001::8
 ipv6 address 6001::1/64 standby 6001::8
 ipv6 enable
 ospf priority 255
 ipv6 ospf cost 100
 ipv6 ospf 100 area 10 instance 200
ipv6 ospf retransmit-delay 3
```

有効な値の範囲は、1 ～ 65535 秒です。デフォルト値は 1 秒です。

OSPFv3 ルータ パラメータの設定

手順

ステップ 1 OSPFv3 のルーティング プロセスをイネーブルにします。

ipv6 router ospf *process-id*

例：

```
ciscoasa(config)# ipv6 router ospf 10
```

process-id 引数は、このルーティング プロセス内部で使用される識別子です。ローカルに割り当てられ、1 ～ 65535 の任意の正の整数を指定できます。この ID は内部管理専用のため、他のどのデバイス上の ID とも照合する必要はありません。最大 2 つのプロセスが使用できます。

ステップ 2 OSPFv3 エリア パラメータを設定します。

area

例：

```
ciscoasa(config-rtr)# area 10
```

サポートされているパラメータには、0 ～ 4294967295 の 10 進数値のエリア ID、**A.B.C.D** の IP アドレス形式のエリア ID などがあります。

ステップ 3 コマンドをデフォルト値に設定します。

デフォルト

例：

```
ciscoasa(config-rtr)# default originate
```

originate パラメータはデフォルト ルートを配布します。

ステップ 4 デフォルト情報の配布を制御します。

default-information

ステップ 5 ルート タイプに基づいて、OSPFv3 ルート アドミニストレーティブ ディスタンスを定義します。

distance

例：

```
ciscoasa(config-rtr)# distance 200
```

サポートされるパラメータには、1～254の値のアドミニストレーティブディスタンス、OSPFv3ディスタンスの **ospf** などがあります。

ステップ 6 ルータがタイプ 6 Multicast OSPF（MOSPF）パケットのリンクステート アドバタイズメント（LSA）を受信した場合に、**lsa** パラメータが指定されている syslog メッセージの送信を抑止します。

ignore

例：

```
ciscoasa(config-rtr)# ignore lsa
```

ステップ 7 OSPFv3 ネイバーが起動または停止したときに、ルータが syslog メッセージを送信するように設定します。

log-adjacency-changes

例：

```
ciscoasa(config-rtr)# log-adjacency-changes detail
```

detail パラメータによって、すべての状態変更がログに記録されます。

ステップ 8 インターフェイスでのルーティング アップデートの送受信を抑止します。

passive-interface [*interface_name*]

例：

```
ciscoasa(config-rtr)# passive-interface inside
```

interface_name 引数は、OSPFv3 プロセスが実行されているインターフェイスの名前を指定します。

ステップ 9 あるルーティング ドメインから別のルーティング ドメインへのルートの再配布を設定します。

redistribute {**connected** | **ospf** | **static**}

それぞれの説明は次のとおりです。

- **connected**：接続ルートを指定します。
- **ospf**：OSPFv3 ルートを指定します。
- **static**：スタティック ルートを指定します。

例：

```
ciscoasa(config-rtr)# redistribute ospf
```

ステップ 10 指定したプロセスの固定ルータ ID を作成します。

router-id {*A.B.C.D* | **cluster-pool** | **static**}

それぞれの説明は次のとおりです。

A.B.C.D：IP アドレス形式の OSPF ルータ ID を指定します。

cluster-pool：個別インターフェイス クラスタリングが設定されている場合に、IP アドレス プールを設定します。クラスタリングで使用される IP アドレス プールの詳細については、クラスタリングの IP アドレス プールの設定（OSPFv2 および OSPFv3）（26 ページ）を参照してください。

例：

```
ciscoasa(config-rtr)# router-id 10.1.1.1
```

ステップ 11 0 ～ 128 の有効な値で IPv6 アドレス サマリーを設定します。

summary-prefix *X:X:X:X::X/*

例：

```
ciscoasa(config-if)# ipv6 router ospf 1
ciscoasa(config-router)# router-id 192.168.3.3
ciscoasa(config-router)# summary-prefix FECO::/24
ciscoasa(config-router)# redistribute static
```

X:X:X:X::X/ パラメータは、IPv6 プレフィックスを指定します。

ステップ 12 ルーティング タイマーを調整します。

timers

ルーティング タイマー パラメータは次のとおりです。

- **lsa**：OSPFv3 LSA タイマーを指定します。
- **nsf**：OSPFv3 NSF 待機タイマーを指定します。
- **pacing**：OSPFv3 ペーシング タイマーを指定します。
- **throttle**：OSPFv3 スロットル タイマーを指定します。

例：

```
ciscoasa(config)# ipv6 router ospf 10
ciscoasa(config-rtr)# timers throttle spf 6000 12000 14000
```

OSPFv3 エリア パラメータの設定

手順

ステップ 1 OSPFv3 のルーティング プロセスをイネーブルにします。

ipv6 router ospf *process-id*

例：

```
ciscoasa(config)# ipv6 router ospf 1
```

process-id 引数は、このルーティング プロセス内部で使用される識別子です。ローカルに割り当てられ、1 ～ 65535 の任意の正の整数を指定できます。

この ID は内部管理専用のため、他のどのデバイス上の ID とも照合する必要はありません。最大 2 つのプロセスが使用できます。

ステップ 2 NSSA エリアまたはスタブ エリアのサマリー デフォルト コストを設定します。

area *area-id* **default-cost** *cost*

例：

```
ciscoasa(config-rtr)# area 1 default-cost nssa
```

ステップ 3 アドレスおよび境界ルータ専用のマスクと一致するルートを集約します。

area *area-id* **range** *ipv6-prefix*/ *prefix-length* [**advertise** | **not advertise**] [**cost** *cost*]

例：

```
ciscoasa(config-rtr)# area 1 range FE01:1::1/64
```

- *area-id* 引数は、ルートが集約されているエリアを識別します。値には、10 進数または IPv6 プレフィックスを指定できます。
- *ipv6-prefix* 引数は、IPv6 プレフィックスを指定します。*prefix-length* 引数は、プレフィックス長を指定します。
- **advertise** キーワードは、アドレス範囲ステータスをアドバタイズに設定し、Type 3 サマリー LSA を生成します。
- **not-advertise** キーワードはアドレス範囲ステータスを DoNotAdvertise に設定します。
- Type 3 サマリー LSA は抑制され、コンポーネント ネットワークは他のネットワークから隠された状態のままです。
- **cost** *cost* キーワード引数のペアは、宛先への最短パスを決定するために OSPF SPF 計算で使用されるサマリー ルートのメトリックまたはコストを指定します。

- 有効値の範囲は 0 ～ 16777215 です。

ステップ 4 NSSA エリアを指定します。

area *area-id* **nssa**

例：

```
ciscoasa(config-rtr)# area 1 nssa
```

ステップ 5 スタブ エリアを指定します。

area *area-id* **stub**

例：

```
ciscoasa(config-rtr)# area 1 stub
```

ステップ 6 仮想リンクとそのパラメータを定義します。

area *area-id* **virtual-link** *router-id* [**hello-interval** *seconds*] [**retransmit-interval** *seconds*] [**transmit-delay** *seconds*] [**dead-interval** *seconds*] [**ttl-security hops** *hop-count*]

例：

```
ciscoasa(config-rtr)# area 1 virtual-link 192.168.255.1 hello-interval 5
```

- *area-id* 引数は、ルートが集約されているエリアを識別します。**virtual link** キーワードは、仮想リンク ネイバーの作成を指定します。
- *router-id* 引数は、仮想リンク ネイバーに関連付けられたルータ ID を指定します。
- ルータ ID を表示するには、**show ospf** コマンドまたは **show ipv6 ospf** コマンドを入力します。デフォルト値はありません。
- **hello-interval** キーワードは、インターフェイス上で送信される hello パケット間の時間を秒単位で指定します。hello 間隔は、hello パケットでアドバタイズされる符号なし整数です。この値は、共通のネットワークに接続されているすべてのルータおよびアクセス サーバーで同じであることが必要です。有効値の範囲は 1 ～ 8192 です。デフォルトは 10 です。
- **retransmit-interval** *seconds* キーワード引数のペアは、インターフェイスに属する隣接関係の LSA 再送信間の時間を秒単位で指定します。再送信間隔は、接続されているネットワーク上の任意の 2 台のルータ間の予想されるラウンドトリップ遅延です。この値は、予想されるラウンドトリップ遅延より大きくなり、1 ～ 8192 の範囲で指定できます。デフォルトは 5 分です。
- **transmit-delay** *seconds* キーワード引数のペアは、インターフェイス上でリンクステート更新パケットを送信するために必要とされる時間を秒単位で設定します。ゼロよりも大きい整数値を指定します。アップデート パケット内の LSA 自体の経過時間は、転送前にこの値の分だけ増分されます。値の範囲は 1 ～ 8192 です。デフォルトは 1 です。

- **dead-interval seconds** キーワード引数のペアは、ルータがダウンしていることをネイバーが示す前に hello パケットを非表示にする時間を秒単位で指定します。Dead 間隔は符号なし整数です。デフォルトは hello 間隔の 4 倍または 40 秒です。この値は、共通のネットワークに接続されているすべてのルータおよびアクセス サーバーで同じであることが必要です。有効値の範囲は 1 ～ 8192 です。
- **ttl-security hops** キーワードは仮想リンクの存続可能時間（TTL）セキュリティを設定します。*hop-count* 引数の値は 1 ～ 254 の範囲で指定できます。

OSPFv3 受動インターフェイスの設定

手順

ステップ 1 OSPFv3 のルーティング プロセスをイネーブルにします。

ipv6 router ospf *process_id*

例：

```
ciscoasa(config-if)# ipv6 router ospf 1
```

process_id 引数は、このルーティング プロセス内部で使用される識別子です。ローカルに割り当てられ、1 ～ 65535 の任意の正の整数を指定できます。この ID は内部管理専用のため、他のどのデバイス上の ID とも照合する必要はありません。最大 2 つのプロセスが使用できます。

ステップ 2 インターフェイスでのルーティング アップデートの送受信を抑止します。

passive-interface [*interface_name*]

例：

```
ciscoasa(config-rtr)# passive-interface inside
```

interface_name 引数は、OSPFv3 プロセスが実行されているインターフェイスの名前を指定します。*no interface_name* 引数を指定すると、OSPFv3 プロセス *process_id* のすべてのインターフェイスがパッシブとなります。

OSPFv3 アドミニストレーティブ ディスタンスの設定

手順

ステップ 1 OSPFv3 のルーティング プロセスをイネーブルにします。

ipv6 router ospf *process_id*

例：

```
ciscoasa(config-if)# ipv6 router ospf 1
```

process_id 引数は、このルーティング プロセス内部で使用される識別子です。ローカルに割り当てられ、1 ～ 65535 の任意の正の整数を指定できます。この ID は内部管理専用のため、他のどのデバイス上の ID とも照合する必要はありません。最大 2 つのプロセスが使用できます。

ステップ 2 OSPFv3 ルートのアドミニストレーティブ ディスタンスを設定します。

distance [**ospf** {**external** | **inter-area** | **intra-area**}] *distance*

例：

```
ciscoasa(config-rtr)# distance ospf external 200
```

ospf キーワードは、OSPFv3 ルートを指定します。**external** キーワードは、OSPFv3 の外部タイプ 5 およびタイプ 7 ルートを指定します。**inter-area** キーワードは、OSPVv3 のエリア間ルートを指定します。**intra-area** キーワードは、OSPFv3 のエリア内ルートを指定します。*distance* 引数は、10 ～ 254 の整数であるアドミニストレーティブ ディスタンスを指定します。

OSPFv3 タイマーの設定

OSPFv3 の LSA 到着タイマー、LSA ペーシング タイマー、およびスロットリング タイマーを設定できます。

手順

ステップ 1 OSPFv3 のルーティング プロセスをイネーブルにします。

ipv6 router ospf *process-id*

例：

```
ciscoasa(config-if)# ipv6 router ospf 1
```

process-id 引数は、このルーティング プロセス内部で使用される識別子です。ローカルに割り当てられ、1 ～ 65535 の任意の正の整数を指定できます。この ID は内部管理専用のため、他のどのデバイス上の ID とも照合する必要はありません。最大 2 つのプロセスが使用できます。

ステップ 2 ASAが OSPF ネイバーから同一の LSA を受け入れる最小間隔を設定します。

timers lsa arrival *milliseconds*

例：

```
ciscoasa(config-rtr)# timers lsa arrival 2000
```

milliseconds 引数は、ネイバーから到着する同じ LSA の受け入れの間で経過する最小遅延をミリ秒単位で指定します。有効な範囲は 0 ～ 6,000,000 ミリ秒です。デフォルトは 1000 ミリ秒です。

ステップ 3 LSA フラッド パケット ペーシングを設定します。

timers pacing flood *milliseconds*

例：

```
ciscoasa(config-rtr)# timers lsa flood 20
```

milliseconds 引数は、フラッディング キュー内の LSA が更新と更新の間にペーシングされる時間（ミリ秒）を指定します。設定できる範囲は 5 ～ 100 ミリ秒です。デフォルト値は、33 ミリ秒です。

ステップ 4 OSPFv3 LSA を収集してグループ化し、リフレッシュ、チェックサム、またはエージングを行う間隔を変更します。

timers pacing lsa-group *seconds*

例：

```
ciscoasa(config-rtr)# timers pacing lsa-group 300
```

seconds 引数は、LSA がグループ化、リフレッシュ、チェックサム計算、またはエージングされる間隔を秒単位で指定します。有効な範囲は 10 ～ 1800 秒です。デフォルト値は 240 秒です。

ステップ 5 LSA 再送信パケット ペーシングを設定します。

timers pacing retransmission *milliseconds*

例：

```
ciscoasa(config-rtr)# timers pacing retransmission 100
```

milliseconds 引数は、再送信キュー内の LSA がペーシングされる時間（ミリ秒）を指定します。設定できる範囲は 5 ～ 200 ミリ秒です。デフォルト値は、66 ミリ秒です。

ステップ 6 OSPFv3 LSA スロットリングを設定します。

timers throttle lsa *milliseconds1 milliseconds2 milliseconds3*

例：

```
ciscoasa(config-rtr)# timers throttle lsa 500 6000 8000
```

- *milliseconds1* 引数は、LSA の最初のオカレンスを生成する遅延をミリ秒単位で指定します。*milliseconds2* 引数は、同じ LSA を送信する最大遅延をミリ秒単位で指定します。*milliseconds3* 引数は、同じ LSA を送信する最小遅延をミリ秒単位で指定します。
- LSA スロットリングでは、最小時間または最大時間が最初のオカレンスの値よりも小さい場合、OSPFv3 が自動的に最初のオカレンス値に修正します。同様に、指定された最遅延が最小遅延よりも小さい場合、OSPFv3 が自動的に最小遅延値に修正します。
- *milliseconds1* の場合、デフォルト値は 0 ミリ秒です。
- *milliseconds2* および *milliseconds3* の場合、デフォルト値は 5000 ミリ秒です。

ステップ 7 OSPFv3 SPF スロットリングを設定します。

timers throttle spf *milliseconds1 milliseconds2 milliseconds3*

例：

```
ciscoasa(config-rtr)# timers throttle spf 5000 12000 16000
```

- *milliseconds1* 引数は、SPF 計算の変更を受信する遅延をミリ秒単位で指定します。*milliseconds2* 引数は、最初と 2 番目の SPF 計算の間の遅延をミリ秒単位で指定します。*milliseconds3* 引数は、SPF 計算の最大待機時間をミリ秒単位で指定します。
- SPF スロットリングでは、*milliseconds2* または *milliseconds3* が *milliseconds1* よりも小さい場合、OSPFv3 が自動的に *milliseconds1* の値に修正します。同様に、*milliseconds3* が *milliseconds2* より小さい場合、OSPFv3 が自動的に *milliseconds2* の値に修正します。
- *milliseconds1* の場合、SPF スロットリングのデフォルト値は 5000 ミリ秒です。
- *milliseconds2* および *milliseconds3* の場合、SPF スロットリングのデフォルト値は 10000 ミリ秒です。

スタティック OSPFv3 ネイバーの定義

ポイントツーポイントの非ブロードキャストネットワークを介して OSPFv3 ルートをアドバタイズするには、スタティック OSPF ネイバーを定義する必要があります。この機能により、OSPFv3 アドバタイズメントを GRE トンネルにカプセル化しなくても、既存の VPN 接続でブロードキャストすることができます。

開始する前に、OSPFv3 ネイバーに対するスタティック ルートを作成する必要があります。スタティック ルートの作成方法の詳細については、スタティック ルートの設定を参照してください。

手順

ステップ 1 OSPFv3 ルーティング プロセスをイネーブルにし、IPv6 ルータ コンフィギュレーション モードを開始します。

ipv6 router ospf *process-id*

例：

```
ciscoasa(config)# ipv6 router ospf 1
```

process-id 引数は、このルーティング プロセス内部で使用される識別子です。ローカルに割り当てられ、1 ～ 65535 の任意の正の整数を指定できます。この ID は内部管理専用のため、他のどのデバイス上の ID とも照合する必要はありません。最大 2 つのプロセスが使用できます。

ステップ 2 非ブロードキャスト ネットワークへの OSPFv3 ルータの相互接続を設定します。

ipv6 ospf neighbor *ipv6-address* [**priority** *number*] [**poll-interval** *seconds*] [**cost** *number*] [**database-filter all out**]

例：

```
ciscoasa(config-if)# interface ethernet0/0 ipv6 ospf neighbor FE80::A8BB:CCFF:FE00:C01
```

OSPFv3 デフォルト パラメータのリセット

OSPFv3 パラメータをデフォルト値に戻すには、次の手順を実行します。

手順

ステップ 1 OSPFv3 のルーティング プロセスをイネーブルにします。

ipv6 router ospf *process-id*

例：

```
ciscoasa(config-if)# ipv6 router ospf 1
```

process_id 引数は、このルーティング プロセス内部で使用される識別子です。ローカルに割り当てられ、1 ～ 65535 の任意の正の整数を指定できます。この ID は内部管理専用のため、他のどのデバイス上の ID とも照合する必要はありません。最大 2 つのプロセスが使用できます。

ステップ 2 オプションのパラメータをデフォルト値に戻します。

default [**area** | **auto-cost** | **default-information** | **default-metric** | **discard-route** | **discard-route** | **distance** | **distribute-list** | **ignore** | **log-adjacency-changes** | **maximum-paths** | **passive-interface** | **redistribute** | **router-id** | **summary-prefix** | **timers**]

例：

```
ciscoasa(config-rtr)# default metric 5
```

- **area** キーワードは、OSPFv3 エリア パラメータを指定します。**auto-cost** キーワードは、帯域幅に従って OSPFv3 インターフェイス コストを指定します。
- **default-information** キーワードはデフォルト情報を配布します。**default-metric** キーワードは、再配布ルートのメトリックを指定します。
- **discard-route** キーワードは、廃棄ルートのインストールをイネーブルまたはディセーブルにします。**distance** キーワードはアドミニストレーティブ ディスタンスを指定します。
- **distribute-list** キーワードは、ルーティング アップデートのネットワークをフィルタリングします。
- **Ignore** キーワードは、特定のイベントを無視します。**log-adjacency-changes** キーワードは、隣接状態の変更をログに記録します。
- **maximum-paths** キーワードは、複数のパスを介して複数のパケットを転送します。
- **passive-interface** キーワードは、インターフェイス上のルーティング アップデートを抑止します。
- **redistribute** キーワードは、別のルーティング プロトコルからの IPv6 プレフィックスを再配布します。
- **router-id** キーワードは、指定されたルーティング プロセスのルータ ID を指定します。
- **summary-prefix** キーワードは、IPv6 サマリー プレフィックスを指定します。
- **timers** キーワードは、OSPFv3 タイマーを指定します。

Syslog メッセージの送信

OSPFv3 ネイバーが起動または停止したときに、ルータが syslog メッセージを送信するように設定します。

手順

ステップ 1 OSPFv3 のルーティング プロセスをイネーブルにします。

ipv6 router ospf *process-id*

例：

```
ciscoasa(config-if)# ipv6 router ospf 1
```

process-id 引数は、このルーティング プロセス内部で使用される識別子です。ローカルに割り当てられ、1 ～ 65535 の任意の正の整数を指定できます。この ID は内部管理専用のため、他のどのデバイス上の ID とも照合する必要はありません。最大 2 つのプロセスが使用できます。

ステップ 2 OSPFv3 ネイバーが起動または停止したときに、ルータが syslog メッセージを送信するように設定します。

log-adjacency-changes [**detail**]

例：

```
ciscoasa(config-rtr)# log-adjacency-changes detail
```

detail キーワードは、OSPFv3 ネイバーが起動または停止したときだけではなく、各状態の syslog メッセージを送信します。

Syslog メッセージの抑止

ルータがサポートされていない LSA タイプ 6 Multicast OSPF（MOSPF）パケットを受信した場合の syslog メッセージの送信を抑止するには、次の手順を実行します。

手順

ステップ 1 OSPFv2 のルーティング プロセスをイネーブルにします。

router ospf *process_id*

例：

```
ciscoasa(config-if)# router ospf 1
```

process_id 引数は、このルーティング プロセス内部で使用される識別子です。ローカルに割り当てられ、1 ～ 65535 の任意の正の整数を指定できます。この ID は内部管理専用のため、他のどのデバイス上の ID とも照合する必要はありません。最大 2 つのプロセスが使用できます。

ステップ 2 ルータが、サポートされていない LSA タイプ 6 MOSPF パケットを受信した場合の syslog メッセージの送信を抑止します。

ignore lsa mospf

例：

```
ciscoasa(config-rtr)# ignore lsa mospf
```

集約ルート コストの計算

手順

RFC 1583 に従ってサマリー ルート コストの計算に使用される方式に復元します。

compatible rfc1583

例：

```
ciscoasa (config-rtr)# compatible rfc1583
```

OSPFv3 ルーティング ドメインへのデフォルトの外部ルートの生成

手順

ステップ 1 OSPFv3 のルーティング プロセスをイネーブルにします。

ipv6 router ospf *process-id*

例：

```
ciscoasa(config-if)# ipv6 router ospf 1
```

process-id 引数は、このルーティング プロセス内部で使用される識別子です。ローカルに割り当てられ、1 ～ 65535 の任意の正の整数を指定できます。この ID は内部管理専用のため、他のどのデバイス上の ID とも照合する必要はありません。最大 2 つのプロセスが使用できます。

ステップ 2 OSPFv3 ルーティング ドメインへのデフォルトの外部ルートを生成します。

default-information originate [**always**] **metric** *metric-value* [**metric-type** *type-value*] [**route-map** *map-name*]

例：

```
ciscoasa(config-rtr)# default-information originate always metric 3 metric-type 2
```

- **always** キーワードは、デフォルト ルートがあるかどうかにかかわらず、デフォルト ルートをアドバタイズします。
- **metric** *metric-value* キーワード引数のペアは、デフォルト ルートの生成に使用するメトリックを指定します。
- **default-metric** コマンドを使用して値を指定しない場合、デフォルト値は 10 です。有効なメトリック値の範囲は、0 ～ 16777214 です。
- **metric-type** *type-value* キーワード引数のペアは、OSPFv3 ルーティング ドメインにアドバタイズされるデフォルト ルートに関連付けられる外部リンク タイプを指定します。有効な値は次のいずれかになります。
  - 1：タイプ 1 外部ルート
  - 2：タイプ 2 外部ルート

  デフォルトはタイプ 2 外部ルートです。
- **route-map** *map-name* キーワード引数のペアは、ルート マップが一致している場合にデフォルト ルートを生成するルーティング プロセスを指定します。

IPv6 サマリー プレフィックスの設定

手順

ステップ 1 OSPFv3 のルーティング プロセスをイネーブルにします。

ipv6 router ospf *process-id*

例：

```
ciscoasa(config-if)# ipv6 router ospf 1
```

process_id 引数は、このルーティング プロセス内部で使用される識別子です。ローカルに割り当てられ、1 ～ 65535 の任意の正の整数を指定できます。この ID は内部管理専用のため、他のどのデバイス上の ID とも照合する必要はありません。最大 2 つのプロセスが使用できます。

ステップ 2 IPv6 サマリー プレフィックスを設定します。

summary-prefix *prefix* [**not-advertise** | **tag** *tag-value*]

例：

```
ciscoasa(config-if)# ipv6 router ospf 1
ciscoasa(config-rtr)# router-id 192.168.3.3
ciscoasa(config-rtr)# summary-prefix FECO::/24
ciscoasa(config-rtr)# redistribute static
```

prefix 引数は、宛先の IPv6 ルート プレフィックスです。**not-advertise** キーワードは、指定したプレフィックスとマスク ペアと一致するルートを抑止します。このキーワードはOSPFv3 だけに適用されます。**tag** *tag-value* キーワード引数のペアは、ルート マップで再配布を制御するために一致値として使用できるタグ値を指定します。このキーワードは OSPFv3 だけに適用されます。

IPv6 ルートの再配布

手順

ステップ 1 OSPFv3 のルーティング プロセスをイネーブルにします。

ipv6 router ospf *process-id*

例：

```
ciscoasa(config-if)# ipv6 router ospf 1
```

process-id 引数は、このルーティング プロセス内部で使用される識別子です。ローカルに割り当てられ、1 ～ 65535 の任意の正の整数を指定できます。この ID は内部管理専用のため、他のどのデバイス上の ID とも照合する必要はありません。最大 2 つのプロセスが使用できます。

ステップ 2 ある OSPFv3 プロセスから別の OSPFv3 プロセスに IPv6 ルートを再配布します。

redistribute *source-protocol* [*process-id*] [**include-connected** {[**level-1** | **level-2**}] [*as-number*] [**metric** [*metric-value* | **transparent**}] **[metric-type** *type-value*] [**match** {**external** [**1**|**2**] | **internal** | **nssa-external** [**1**|**2**]}] [**tag** *tag-value*] [**route-map** *map-tag*]

例：

```
ciscoasa(config-rtr)# redistribute connected 5 type-1
```

- *source-protocol* 引数は、ルートの再配布元となるソース プロトコルを指定します。これは、スタティック、接続済み、または OSPFv3 にすることができます。
- *process-id* 引数は、OSPFv3 ルーティング プロセスがイネーブルになったときに管理目的で割り当てられる番号です。
- **include-connected** キーワードは、ソース プロトコルから学習したルートと、ソース プロトコルが動作しているインターフェイス上の接続先プレフィックスを、ターゲット プロトコルが再配布できるようにします。

- **level-1** キーワードは、Intermediate System-to-Intermediate System（IS-IS）用に、レベル 1 ルートが他の IP ルーティング プロトコルに個別に再配布されることを指定します。
- **level-1-2** キーワードは、IS-IS 用に、レベル 1 とレベル 2 の両方のルートが他の IP ルーティング プロトコルに再配布されることを指定します。
- **level-2** キーワードは、IS-IS 用に、レベル 2 ルートが他の IP ルーティング プロトコルに個別に再配布されることを指定します。
- **metric** *metric-value* キーワード引数のペアでは、ある OSPFv3 プロセスのルートを同じルータ上の別の OSPFv3 プロセスに再配布する場合、メトリック値を指定しないと、メトリックは 1 つのプロセスから他のプロセスへ存続します。他のプロセスを OSPFv3 プロセスに再配布するときに、メトリック値を指定しない場合、デフォルトのメトリックは 20 です。
- **metric transparent** キーワードにより、RIP は RIP メトリックとして再配布ルートのルーティング テーブル メトリックを使用します。
- **metric-type** *type-value* キーワード引数のペアは、OSPFv3 ルーティング ドメインにアドバタイズされるデフォルト ルートに関連付けられる外部リンク タイプを指定します。有効な値は、タイプ 1 外部ルートの場合は 1、タイプ 2 外部ルートの場合は 2 です。**metric-type** キーワードに値が指定されていない場合、ASA は、タイプ 2 外部ルートを受け入れます。IS-IS の場合、リンク タイプは、63 未満の IS-IS メトリックの場合は内部、64 を超えて 128 未満の IS-IS メトリックの場合は外部となります。デフォルトは、内部です。
- **match** キーワードは、他のルーティング ドメインにルートを再配布し、次のいずれかのオプションとともに使用されます。自律システムの外部であり、タイプ 1 またはタイプ 2 の外部ルートとして OSPFv3 にインポートされるルートの場合は **external** [**1**|**2**]、特定の自律システムの内部にあるルートの場合は **internal**、自律システムの外部であり、タイプ 1 またはタイプ 2 の外部ルートとして IPv6 の NSSA で OSPFv3 にインポートされるルートの場合は **nssa-external** [**1**|**2**]。
- **tag** *tag-value* キーワード引数のペアは、ASBR 間で情報を通信するために使用できる、各外部ルートに付加される 32 ビットの 10 進数値を指定します。何も指定しない場合、BGP および EGP からのルートにはリモート自律システムの番号が使用されます。その他のプロトコルについては、ゼロが使用されます。有効値の範囲は、0 ～ 4294967295 です。
- **route-map** キーワードは、送信元ルーティング プロトコルから現在のルーティング プロトコルへのルートのインポートのフィルタリングをチェックするルート マップを指定します。このキーワードを指定しない場合、すべてのルートが再配布されます。このキーワードを指定し、ルート マップ タグが表示されていない場合、ルートはインポートされません。*map-tag* 引数は、設定されたルート マップを識別します。

グレースフル リスタートの設定

ASA では、既知の障害状況が発生することがあります。これにより、スイッチング プラットフォーム全体でパケット転送に影響を与えることがあってはなりません。Non-Stop Forwarding（NSF）機能では、ルーティング プロトコル情報を復元している間に、既知のルートへのデータ転送が続行されます。

ハイアベイラビリティモードでは、アクティブユニットが非アクティブになり、スタンバイユニットが新しいアクティブになると、OSPF プロセスが再起動します。同様に、クラスタモードでは、制御ユニットが非アクティブになり、データユニットが新しい制御ユニットとして選択されると、OSPF プロセスが再起動します。このような OSPF 移行プロセスでは、かなりの遅延が発生します。OSPF プロセスの状態変更時のトラフィック損失を回避するように NSF を設定できます。また NSF 機能は、スケジュール済みヒットレス ソフトウェア アップグレードがあるときに便利です。

グレースフル リスタートは、OSPFv2 と OSPFv3 の両方でサポートされています。NSF Cisco（RFC 4811 および RFC 4812）または NSF IETF（RFC 3623）のいずれかを使用して、OSPFv2 上でグレースフル リスタートを設定できます。graceful-restart（RFC 5187）を使用して、OSPFv3 上でグレースフル リスタートを設定できます。

NSF グレースフル リスタート機能の設定には、機能の設定と NSF 対応または NSF 認識としてのデバイスの設定という 2 つのステップが伴います。NSF 対応デバイスは、ネイバーに対して独自のリスタート アクティビティを示すことができ、NSF 認識デバイスはネイバーのリスタートをサポートすることができます。

デバイスは、いくつかの条件に応じて、NSF 対応または NSF 認識として設定できます。

- デバイスは、現在のデバイスのモードに関係なく、NSF 認識デバイスとして設定できます。
- デバイスを NSF 対応として設定するには、デバイスはフェールオーバーまたはスパンド EtherChannel（L2）クラスタ モードのいずれかである必要があります。
- デバイスを NSF 認識または NSF 対応にするには、必要に応じて opaque リンク ステート アドバタイズメント（LSA）/リンク ローカル シグナリング（LLS）ブロックの機能を使って設定する必要があります。

（注） OSPFv2 用に fast hello が設定されている場合、アクティブ ユニットのリロードが発生し、スタンバイ ユニットがアクティブになっても、グレースフル リスタートは発生しません。これは、ロール変更にかかる時間は、設定されているデッド インターバルよりも大きいためです。

機能の設定

Cisco NSF グレースフル リスタート メカニズムは、リスタート アクティビティを示すために、Hello パケットで RS ビットが設定された LLS ブロックを送信するため、LLS 機能に依存して

います。IETF NSF メカニズムは、リスタート アクティビティを示すために、タイプ 9 の opaque LSA を送信するため、opaque LSA 機能に依存しています。機能を設定するには、次のコマンドを入力します。

手順

ステップ 1 OSPF ルーティング プロセスを作成し、再配布する OSPF プロセスのルータ コンフィギュレーション モードに入ります。

router ospf *process_id*

例：

```
ciscoasa(config)# router ospf 2
```

process_id 引数は、このルーティング プロセス内部で使用される識別子です。任意の正の整数を使用できます。この ID は内部専用のため、他のどのデバイス上の ID とも照合する必要はありません。最大 2 つのプロセスが使用できます。

ステップ 2 LLS データ ブロックまたは opaque LSA の使用をイネーブルにして、NSF をイネーブルにします。

capability {lls|opaque}

lls キーワードは、Cisco NSF グレースフル リスタート メカニズムに対して、LLS 機能をイネーブルにするために使用されます。

opaque キーワードは、IETF NSF グレースフル リスタート メカニズムに対して、opaque LSA 機能をイネーブルにするために使用されます。

OSPFv2 のグレースフル リスタートの設定

OSPFv2、Cisco NSF および IETF NSF には、2 つのグレースフル リスタート メカニズムがあります。OSPF インスタンスに対しては、これらのグレースフル リスタート メカニズムのうち一度に設定できるのは 1 つだけです。NSF 認識デバイスは、Cisco NSF ヘルパーと IETF NSF ヘルパーの両方として設定できますが、NSF 対応デバイスは OSPF インスタンスに対して、Cisco NSF または IETF NSF モードのいずれかとして設定できます。

OSPFv2 の Cisco NSF グレースフル リスタートの設定

NSF 対応または NSF 認識デバイスに対して、OSPFv2 の Cisco NSF グレースフル リスタートを設定します。

手順

ステップ 1 NSF 対応デバイスで Cisco NSF をイネーブルにします。

nsf cisco [enforce global]

例：

```
ciscoasa(config-router)# nsf cisco
```

enforce global キーワードは、非 NSF 認識ネイバー デバイスが検出されると、NSF リスタートをキャンセルします。

ステップ 2 NSF 認識デバイスで、Cisco NSF ヘルパー モードをイネーブルにします。

capability {lls|opaque}

例：

```
ciscoasa(config-router)# capability lls
```

このコマンドは、デフォルトでイネーブルになっています。このコマンドの no 形式を使用すると、ディセーブルになります。

OSPFv2 の IETF NSF グレースフル リスタートの設定

NSF 対応または NSF 認識デバイスに対して、OSPFv2 の IETF NSF グレースフル リスタートを設定します。

手順

ステップ 1 NSF 対応デバイスで IETF NSF を有効にします。

nsf ietf [restart-interval *seconds***]**

例：

```
ciscoasa(config-router)# nsf ietf restart-interval 80
```

グレースフルリスタートの間隔を秒単位で指定できます。有効な値は 1 ～ 1800 秒です。デフォルト値は 120 秒です。

隣接関係（アジャセンシー）が有効になるまでにかかる時間よりも再起動間隔が小さい値に設定されている場合、グレースフル リスタートは終了することがあります。たとえば、30 秒以下の再起動間隔はサポートされていません。

ステップ 2 NSF 認識デバイスで、IETF NSF ヘルパー モードをイネーブルにします。

nsf ietf helper [strict-lsa-checking]

例：

```
ciscoasa(config-router)# nsf ietf helper
```

strict-LSA-checking キーワードは、再起動ルータにフラッディングされる可能性がある LSA への変更があることが検出された場合、またはグレースフル リスタート プロセスが開始されたときに再起動ルータの再送リスト内に変更された LSA があると検出された場合、ヘルパールータはルータの再起動プロセスを終了させることを示します。

このコマンドは、デフォルトでイネーブルになっています。このコマンドの no 形式を使用すると、ディセーブルになります。

OSPFv3 のグレースフル リスタートの設定

OSPFv3 の NSF グレースフル リスタート機能を設定するには、2 つのステップを伴います。NSF 対応としてのデバイスの設定と、NSF 認識としてのデバイスの設定です。

手順

ステップ 1 明示的な IPv6 アドレスが設定されていないインターフェイスにおける IPv6 処理をイネーブルにします。

interface physical_interface ipv6 enable

例：

```
ciscoasa(config)# interface ethernet 0/0
ciscoasa(config-if)# ipv6 enable
```

physical_interface 引数は、OSPFv3 NSF に参加するインターフェイスを識別します。

ステップ 2 NSF 対応デバイスで OSPFv3 のグレースフル リスタートをイネーブルにします。

graceful-restart [restart interval seconds]

例：

```
ciscoasa(config-router)# graceful-restart restart interval 80
```

restart interval seconds は、グレースフル リスタート間隔の長さを秒単位で指定します。有効な値は 1 ～ 1800 秒です。デフォルト値は 120 秒です。

隣接関係（アジャセンシー）が有効になるまでにかかる時間よりもリスタート間隔が小さい値に設定されている場合、グレースフル リスタートは終了することがあります。たとえば 30 秒以下の再起動間隔は、サポートされていません。

ステップ 3 NSF 認識デバイスで OSPFv3 のグレースフル リスタートをイネーブルにします。

graceful-restart helper [strict-lsa-checking]

例：

```
ciscoasa(config-router)# graceful-restart helper strict-lsa-checking
```

strict-LSA-checking キーワードは、再起動ルータにフラッディングされる可能性がある LSA への変更があることが検出された場合、またはグレースフル リスタート プロセスが開始されたときに再起動ルータの再送リスト内に変更された LSA があると検出された場合、ヘルパー ルータはルータの再起動プロセスを終了させることを示します。

グレースフル リスタート ヘルパー モードは、デフォルトでイネーブルになっています。

OSPF のグレースフル リスタート待機タイマーの設定

OSPF ルータでは、すべてのネイバーがパケットに含まれているかが不明な場合は、Hello パケットにアタッチされている EO-TLV に RS ビットを設定することが予期されます。ただし、隣接関係（アジャセンシー）を維持するにはルータの再起動が必要です。ただし、RS ビット値は RouterDeadInterval 秒より長くすることはできません。そのため、Hello パケットの RS ビットを RouterDeadInterval 秒未満に設定するための **timers nsf wait** コマンドが導入されました。NSF 待機タイマーのデフォルト値は 20 秒です。

始める前に

- OSPF の Cisco NSF 待機時間を設定するには、デバイスが NSF 認識または NSF 対応である必要があります。

手順

ステップ 1 OSPF ルータ コンフィギュレーション モードを開始します。

例：

```
ciscoasa(config)# router ospf
```

ステップ 2 タイマーを入力し、NSF を指定します。

例：

```
ciscoasa(config-router)# timers?
router mode commands/options:
  lsa       OSPF LSA timers
```

```
  nsf       OSPF NSF timer
  pacing    OSPF pacing timers
  throttle  OSPF throttle timers
ciscoasa(config-router)# timers nsf ?
```

ステップ 3 グレースフルリスタート待機間隔を入力します。この値は、1 ～ 65535 の範囲で指定できます。

例：

```
ciscoasa(config-router)# timers nsf wait 200
```

グレースフルリスタート待機間隔を使用することで、待機間隔がルータの dead 間隔よりも長くならないようにできます。

OSPFv2 設定の削除

OSPFv2 設定を削除します。

手順

イネーブルにした OSPFv2 設定全体を削除します。

clear configure router ospf *pid*

例：

```
ciscoasa(config)# clear configure router ospf 1000
```

設定をクリアした後、**router ospf** コマンドを使用して OSPF を再設定する必要があります。

OSPFv3 設定の削除

OSPFv3 設定を削除します。

手順

イネーブルにした OSPFv3 設定全体を削除します。

clear configure ipv6 router ospf *process-id*

例：

```
ciscoasa(config)# clear configure ipv6 router ospf 1000
```

設定をクリアした後、**ipv6 router ospf** コマンドを使用して OSPFv3 を再設定する必要があります。

OSPFv2 の例

次の例に、さまざまなオプションのプロセスを使用して OSPFv2 をイネーブルにし、設定する方法を示します。

1. OSPFv2 をイネーブルにするには、次のコマンドを入力します。

```
ciscoasa(config)# router ospf 2
ciscoasa(config-rtr)# network 10.0.0.0 255.0.0.0 area 0
```

2. （オプション）1 つの OSPFv2 プロセスから別の OSPFv2 プロセスにルートを再配布するには、次のコマンドを入力します。

```
ciscoasa(config)# route-map 1-to-2 permit
ciscoasa(config-route-map)# match metric 1
ciscoasa(config-route-map)# set metric 5
ciscoasa(config-route-map)# set metric-type type-1
ciscoasa(config-route-map)# router ospf 2
ciscoasa(config-rtr)# redistribute ospf 1 route-map 1-to-2
```

3. （オプション）OSPFv2 インターフェイス パラメータを設定するには、次のコマンドを入力します。

```
ciscoasa(config)# router ospf 2
ciscoasa(config-rtr)# network 10.0.0.0 255.0.0.0 area 0
ciscoasa(config-rtr)# interface inside
ciscoasa(config-interface)# ospf cost 20
ciscoasa(config-interface)# ospf retransmit-interval 15
ciscoasa(config-interface)# ospf transmit-delay 10
ciscoasa(config-interface)# ospf priority 20
ciscoasa(config-interface)# ospf hello-interval 10
ciscoasa(config-interface)# ospf dead-interval 40
ciscoasa(config-interface)# ospf authentication-key cisco
ciscoasa(config-interface)# ospf message-digest-key 1 md5 cisco
ciscoasa(config-interface)# ospf authentication message-digest
```

4. （オプション）OSPFv2 エリア パラメータを設定するには、次のコマンドを入力します。

```
ciscoasa(config)# router ospf 2
ciscoasa(config-rtr)# area 0 authentication
ciscoasa(config-rtr)# area 0 authentication message-digest
ciscoasa(config-rtr)# area 17 stub
ciscoasa(config-rtr)# area 17 default-cost 20
```

5. （オプション）ルート計算タイマーを設定し、ログにネイバーのアップおよびダウンのメッセージを表示するには、次のコマンドを入力します。

```
ciscoasa(config-rtr)# timers spf 10 120
ciscoasa(config-rtr)# log-adj-changes [detail]
```

6. （オプション）現在の OSPFv2 の設定を表示するには、**show ospf** コマンドを入力します。

次に、 **show ospf** コマンドの出力例を示します。

```
ciscoasa(config)# show ospf

Routing Process “ospf 2” with ID 10.1.89.2 and Domain ID 0.0.0.2
Supports only single TOS(TOS0) routes
Supports opaque LSA
SPF schedule delay 5 secs, Hold time between two SPFs 10 secs
Minimum LSA interval 5 secs. Minimum LSA arrival 1 secs
Number of external LSA 5. Checksum Sum 0x 26da6
Number of opaque AS LSA 0. Checksum Sum 0x     0
Number of DCbitless external and opaque AS LSA 0
Number of DoNotAge external and opaque AS LSA 0
Number of areas in this router is 1. 1 normal 0 stub 0 nssa
External flood list length 0
    Area BACKBONE(0)
        Number of interfaces in this area is 1
        Area has no authentication
        SPF algorithm executed 2 times
        Area ranges are
        Number of LSA 5. Checksum Sum 0x 209a3
        Number of opaque link LSA 0. Checksum Sum 0x     0
        Number of DCbitless LSA 0
        Number of indication LSA 0
        Number of DoNotAge LSA 0
        Flood list length 0
```

7. OSPFv2 設定をクリアするには、次のコマンドを入力します。

```
ciscoasa(config)# clear configure router ospf pid
```

OSPFv3 の例

次に、インターフェイス レベルで OSPFv3 をイネーブルにして設定する例を示します。

```
ciscoasa (config)# interface GigabitEthernet3/1
ciscoasa (config-if)# ipv6 enable
ciscoasa (config-if)# ipv6 ospf 1 area 1
```

次に、**show running-config ipv6** コマンドの出力例を示します。

```
ciscoasa (config)# show running-config ipv6
ipv6 router ospf 1
 log-adjacency-changes
```

次に、**show running-config interface** コマンドの出力例を示します。

```
ciscoasa (config-if)# show running-config interface GigabitEthernet3/1
interface GigabitEthernet3/1
 nameif fda
 security-level 100
 ip address 1.1.11.1 255.255.255.0 standby 1.1.11.2
 ipv6 address 9098::10/64 standby 9098::11
 ipv6 enable
 ipv6 ospf 1 area 1
```

次に、OSPFv3 専用インターフェイスを設定する例を示します。

```
ciscoasa (config)# interface GigabitEthernet3/1
ciscoasa (config-if)# nameif fda
ciscoasa (config-if)# security-level 100
ciscoasa (config-if)# ip address 10.1.11.1 255.255.255.0 standby 10.1.11.2
ciscoasa (config-if)# ipv6 address 9098::10/64 standby 9098::11
ciscoasa (config-if)# ipv6 enable
ciscoasa (config-if)# ipv6 ospf cost 900
ciscoasa (config-if)# ipv6 ospf hello-interval 20
ciscoasa (config-if)# ipv6 ospf network broadcast
ciscoasa (config-if)# ipv6 ospf database-filter all out
ciscoasa (config-if)# ipv6 ospf flood-reduction
ciscoasa (config-if)# ipv6 ospf mtu-ignore
ciscoasa (config-if)# ipv6 ospf 1 area 1 instance 100
ciscoasa (config-if)# ipv6 ospf encryption ipsec spi 890 esp null md5
12345678901234567890123456789012

ciscoasa (config)# ipv6 router ospf 1
ciscoasa (config)# area 1 nssa
ciscoasa (config)# distance ospf intra-area 190 inter-area 100 external 100
ciscoasa (config)# timers lsa arrival 900
ciscoasa (config)# timers pacing flood 100
ciscoasa (config)# timers throttle lsa 900 900 900
ciscoasa (config)# passive-interface fda
ciscoasa (config)# log-adjacency-changes
ciscoasa (config)# redistribute connected metric 100 metric-type 1 tag 700
```

OSPFv3 仮想リンクを設定する方法の例については、次の URL を参照してください:

http://www.cisco.com/en/US/tech/tk365/technologies_configuration_example09186a0080b8fd06.shtml

OSPF のモニタリング

IP ルーティング テーブルの内容、キャッシュの内容、およびデータベースの内容など、特定の統計情報を表示できます。提供される情報は、リソースの使用状況を判定してネットワークの問題を解決するために使用することもできます。また、ノードの到達可能性情報を表示して、デバイス パケットがネットワークを通過するときにとるルーティング パスを見つけることもできます。

さまざまな OSPFv2 ルーティング統計情報をモニターまたは表示するには、次のいずれかのコマンドを入力します。

コマンド	目的
show ospf [*process-id* [*area-id*]]	OSPFv2 ルーティング プロセスに関する一般情報を表示します。
show ospf border-routers	ABR および ASBR までの内部 OSPFv2 ルーティング テーブル エントリを表示します。
show ospf [*process-id* [*area-id*]] database	特定のルータの OSPFv2 データベースに関係する情報のリストを表示します。
show ospf flood-list *if-name*	（OSPFv2 パケット ペーシングの観察のため）インターフェイスへのフラッディングを待機している LSA のリストを表示します。 OSPFv2アップデートパケットは、自動的にペーシングされるため、各パケットの送信間隔が 33 ミリ秒未満になることはありません。ペーシングを行わないと、リンクが低速の状態でアップデート パケットの一部が失われたり、ネイバーがアップデートを十分すばやく受信できなくなったり、あるいは、ルータがバッファ スペースを使い切ってしまったりすることがあります。たとえば、ペーシングを行わないと、次のいずれかのトポロジが存在する場合にパケットがドロップされる可能性があります。 • 高速ルータがポイントツーポイント リンクを介して低速のルータと接続している。 • フラッディング中に、複数のネイバーから 1 つのルータに同時にアップデートが送信される。 ペーシングは、再送信間でも、送信効率を高めて再送信パケットの損失を最小にするために利用されます。インターフェイスからの送信を待機している LSA を表示することもできます。ペーシングの利点は、OSPFv2アップデートおよび再送信パケットの送信の効率をよくすることです。 この機能を設定するタスクはありません。自動的に行われます。
show ospf interface [*if_name*]	OSPFv2-relatedインターフェイスの情報を表示します。

コマンド	目的
show ospf neighbor [*interface-name*] [*neighbor-id*] [**detail**]	OSPFv2 ネイバー情報をインターフェイスごとに表示します。
show ospf request-list *neighbor if_name*	ルータで要求されるすべての LSA のリストを表示します。
show ospf retransmission-list *neighbor if_name*	再送信を待機しているすべての LSA のリストを表示します。
show ospf [*process-id*] summary-address	OSPFv2 プロセスで設定されているサマリー アドレスのすべての再配布情報のリストを表示します。
show ospf [*process-id*] **traffic**	特定の OSPFv2 インスタンスで送信または受信されたパケットのさまざまなタイプのリストを表示します。
show ospf [*process-id*] virtual-links	OSPFv2-related 仮想リンク情報を表示します。
show route cluster	クラスタリングの追加 OSPFv2 ルートの同期情報を表示します。

さまざまな OSPFv3 ルーティング統計情報をモニターまたは表示するには、次のいずれかのコマンドを入力します。

コマンド	目的
show ipv6 ospf [*process-id* [*area-id*]]	OSPFv3 ルーティング プロセスに関する一般的な情報を表示します。
show ipv6 ospf [*process-id*] **border-routers**	ABR および ASBR までの内部 OSPFv3 ルーティング テーブル エントリを表示します。
show ipv6 ospf [*process-id* [*area-id*]] database [**external** | **inter-area prefix** | **inter-area-router** | **network** | **nssa-external** | **router** | **area** | **as** | **ref-lsa** | [*destination-router-id*] [**prefix** *ipv6-prefix*] [*link-state-id*]] [**link** [**interface** *interface-name*] [**adv-router** *router-id*] | **self-originate**] [**internal**] [**database-summary**]	特定のルータの OSPFv3 データベースに関する情報のリストを表示します。
show ipv6 ospf [*process-id* [*area-id*]] **events**	OSPFv3 イベント情報を表示します。

コマンド	目的
show ipv6 ospf [*process-id*] [*area-id*] flood-list *interface-type interface-number*	（OSPFv3 パケット ペーシングの観察のため）インターフェイスへのフラッディングを待機している LSA のリストを表示します。 OSPFv3 アップデート パケットは、自動的にペーシングされるため、各パケットの送信間隔が 33 ミリ秒未満になることはありません。ペーシングを行わないと、リンクが低速の状態でアップデート パケットの一部が失われたり、ネイバーがアップデートを十分すばやく受信できなくなったり、あるいは、ルータがバッファ スペースを使い切ってしまったりすることがあります。たとえば、ペーシングを行わないと、次のいずれかのトポロジが存在する場合にパケットがドロップされる可能性があります。 • 高速ルータがポイントツーポイント リンクを介して低速のルータと接続している。 • フラッディング中に、複数のネイバーから 1 つのルータに同時にアップデートが送信される。 ペーシングは、再送信間でも、送信効率を高めて再送信パケットの損失を最小にするために利用されます。インターフェイスからの送信を待機している LSA を表示することもできます。ペーシングの利点は、OSPFv3 アップデートおよび再送信パケットの送信の効率をよくすることです。 この機能を設定するタスクはありません。自動的に行われます。
show ipv6 ospf [*process-id*] [*area-id*] **interface** [*type number*] [**brief**]	OSPFv3 関連のインターフェイス情報を表示します。
show ipv6 ospf neighbor [*process-id*] [*area-id*] [*interface-type interface-number*] [*neighbor-id*] [**detail**]	OSPFv3 ネイバー情報をインターフェイスごとに表示します。
show ipv6 ospf [*process-id*] [*area-id*] **request-list** [*neighbor*] [*interface*] [*interface-neighbor*]	ルータで要求されるすべての LSA のリストを表示します。
show ipv6 ospf [*process-id*] [*area-id*] retransmission-list [*neighbor*] [*interface*] [*interface-neighbor*]	再送信を待機しているすべての LSA のリストを表示します。

コマンド	目的
show ipv6 ospf statistic [*process-id*] [**detail**]	さまざまな OSPFv3 統計情報を表示します。
show ipv6 ospf [*process-id*] summary-prefix	OSPFv3 プロセスで設定されているサマリー アドレスのすべての再配布情報のリストを表示します。
show ipv6 ospf [*process-id*] **timers** [**lsa-group** | **rate-limit**]	OSPFv3 タイマー情報を表示します。
show ipv6 ospf [*process-id*] **traffic** [*interface_name*]	OSPFv3 トラフィック関連の統計情報を表示します。
show ipv6 ospf virtual-links	OSPFv3-related 仮想リンク情報を表示します。
show ipv6 route cluster [**failover**] [**cluster**] [**interface**] [**ospf**] [**summary**]	クラスタ内の IPv6 ルーティング テーブルのシーケンス番号、IPv6 再コンバージェンス タイマーのステータス、および IPv6 ルーティング エントリのシーケンス番号を表示します。

OSPF の履歴

表 *1 : OSPF* の機能履歴

機能名	プラットフォーム リリース	機能情報
OSPF サポート	7.0(1)	Open Shortest Path First（OSPF）ルーティング プロトコルを使用した、データのルーティング、認証、およびルーティング情報の再配布とモニタについて、サポートが追加されました。 **route ospf** コマンドが導入されました。
マルチ コンテキスト モードのダイナミック ルーティング	9.0(1)	OSPFv2 ルーティングは、マルチ コンテキスト モードでサポートされます。
クラスタ	9.0(1)	OSPFv2 および OSPFv3 の場合、バルク同期、ルートの同期およびスパンド EtherChannel ロードバランシングは、クラスタリング環境でサポートされます。 **show route cluster**、**show ipv6 route cluster**、**debug route cluster**、**router-id cluster-pool** の各コマンドが導入または変更されました。

機能名	プラットフォームリリース	機能情報
IPv6 の OSPFv3 サポート	9.0(1)	OSPFv3 ルーティングが IPv6 に対してサポートされます。 **ipv6 ospf**、**ipv6 ospf area**、**ipv6 ospf cost**、**ipv6 ospf database-filter all out**、**ipv6 ospf dead-interval**、**ipv6 ospf encryption**、**ipv6 ospf hello-interval**、**ipv6 ospf mtu-ignore**、**ipv6 ospf neighbor**、**ipv6 ospf network**、**ipv6 ospf flood-reduction**、**ipv6 ospf priority**、**ipv6 ospf retransmit-interval**、**ipv6 ospf transmit-delay**、**ipv6 router ospf**、**ipv6 router ospf area**、**ipv6 router ospf default**、**ipv6 router ospf default-information**、**ipv6 router ospf distance**、**ipv6 router ospf exit**、**ipv6 router ospf ignore**、**ipv6 router ospf log-adjacency-changes**、**ipv6 router ospf no**、**ipv6 router ospf passive-interface**、**ipv6 router ospf redistribute**、**ipv6 router ospf router-id**、**ipv6 router ospf summary-prefix**、**ipv6 router ospf timers**、**area encryption**、**area range**、**area stub**、**area nssa**、**area virtual-link**、**default**、**default-information originate**、**distance**、**ignore lsa mospf**、**log-adjacency-changes**、**redistribute**、**router-id**、**summary-prefix**、**timers lsa arrival**、**timers pacing flood**、**timers pacing lsa-group**、**timers pacing retransmission**、**timers throttle**、**show ipv6 ospf**、**show ipv6 ospf border-routers**、**show ipv6 ospf database**、**show ipv6 ospf events**、**show ipv6 ospf flood-list**、**show ipv6 ospf graceful-restart**、**show ipv6 ospf interface**、**show ipv6 ospf neighbor**、**show ipv6 ospf request-list**、**show ipv6 ospf retransmission-list**、**show ipv6 ospf statistic**、**show ipv6 ospf summary-prefix**、**show ipv6 ospf timers**、**show ipv6 ospf traffic**、**show ipv6 ospf virtual-links**、**show ospf**、**show running-config ipv6 router**、**clear ipv6 ospf**、**clear configure ipv6 router**、**debug ospfv3**、**ipv6 ospf neighbor** の各コマンドが導入または変更されました。
Fast Hello に対する OSPF サポート	9.2(1)	OSPF は、Fast Hello パケット機能をサポートしているため、OSPF ネットワークでのコンバージェンスが高速なコンフィギュレーションになります。 次のコマンドが変更されました。**ospf dead-interval**
タイマー	9.2(1)	新しい OSPF タイマーを追加し、古いタイマーを廃止しました。 次のコマンドが導入されました。timers lsa arrival、timers pacing、timers throttle 次のコマンドが削除されました。Timers spf、timers lsa-grouping-pacing
アクセス リストを使用したルート フィルタリング	9.2(1)	ACL を使用したルート フィルタリングがサポートされるようになりました。 次のコマンドが導入されました。distribute-list
OSPF モニタリングの強化	9.2(1)	OSPF モニタリングの詳細情報が追加されました。 次のコマンドが変更されました。show ospf events、show ospf rib、show ospf statistics、show ospf border-routers [detail]、show ospf interface brief

機能名	プラットフォームリリース	機能情報
OSPF 再配布 BGP	9.2(1)	OSPF 再配布機能が追加されました。 次のコマンドが追加されました。redistribute bgp
ノンストップ フォワーディング（NSF）に対する OSPF のサポート	9.3(1)	NSF に対する OSPFv2 および OSPFv3 のサポートが追加されました。 次のコマンドが追加されました。capability、nsf cisco、nsf cisco helper、nsf ietf、nsf ietf helper、nsf ietf helper strict-lsa-checking、graceful-restart、graceful-restart helper、graceful-restart helper strict-lsa-checking
ノンストップ フォワーディング（NSF）に対する OSPF のサポート	9.13(1)	NSF 待機タイマーが追加されました。 NSF 再起動間隔のタイマーを設定するための新しいコマンドが追加されました。このコマンドが導入され、待機間隔がルータの dead 間隔よりも長くならないようになりました。 次のコマンドが導入されました。 **timers nsf wait <seconds>**

翻訳について
このドキュメントは、米国シスコ発行ドキュメントの参考和訳です。リンク情報につきましては、日本語版掲載時点で、英語版にアップデートがあり、リンク先のページが移動/変更されている場合がありますことをご了承ください。あくまでも参考和訳となりますので、正式な内容については米国サイトのドキュメントを参照ください。

翻訳について
このドキュメントは、米国シスコ発行ドキュメントの参考和訳です。リンク情報につきましては、日本語版掲載時点で、英語版にアップデートがあり、リンク先のページが移動/変更されている場合がありますことをご了承ください。あくまでも参考和訳となりますので、正式な内容については米国サイトのドキュメントを参照ください。