



# IP Network Enabler

ここでは、StarOS IP Network Enabler (IPNE) 機能について説明します。また、この機能の動作と、IPNE の設定およびモニター方法について説明します。

- [機能説明 \(1 ページ\)](#)
- [機能の仕組み \(3 ページ\)](#)
- [IPNE 機能の設定 \(8 ページ\)](#)
- [IPNE サービスのモニタリング \(10 ページ\)](#)

## 機能説明

この項では、IPNE 機能について説明します。

IPNE (IP Network Enabler) は、P-GW、GGSN、HA、HNBGW など、通信事業者のネットワーク内のさまざまなネットワークノードで実行される MINE クライアントコンポーネントです。セッションおよびネットワークの情報を収集して MINE サーバーに配布します。MINE クラウドサービスは、ワイヤレス通信事業者とパートナーがセッションおよびネットワークの情報を共有および交換してインテリジェントなサービスを実現するための、一元化されたポータルを提供します。

この情報は、MINE サーバーと IPNE サービスの間で XML データの形式で共有されます。IPNE サービスにおけるコアオブジェクトは、XMPP プロトコルエンジンです。設定された MINE サーバーペアごとに 1 つの XMPP プロトコルエンジンインスタンスがあります。エンジンは FSM を使用して XMPP プロトコルを実装します。

共有されるすべての情報は、その時点でのコンテキストから得られます。また、IPNE サービスレベルスケジューラも実装されており、すべてのハンドルのフィードおよび通知アクティビティをレート制御して、コール処理およびデータパスのパフォーマンスに影響を与える過負荷を回避します。

## 他の機能との関係

ここでは、IPNE サービスが他の機能とどのように関連するかについて説明します。

IPNE を設定する前に、次の GW サービスのいずれかを StarOS で設定する必要があります。

- GGSN
- HA
- HNBBGW
- P-GW

設定手順については、『*GGSN Administration Guide*』、『*HA Administration Guide*』、『*HNBBGW Administration Guide*』、および『*P-GW Administration Guide*』を参照してください。

MINE クラウドサービスは、ワイヤレス通信事業者とパートナーがセッションおよびネットワークの情報を共有および交換してインテリジェントなサービスを実現するための一元化されたポータルを提供します。MINE クライアントコンポーネントは、PGW、HA など、通信事業者のネットワーク内のさまざまなネットワークノードで実行され、セッションおよびネットワークの情報を収集して MINE サーバーに配布します。クライアントは IPNE です。

IPNE クライアントは、設定可能なサービスとして StarOS 上で実行されます。Enhanced Charging Service (ECS) コンポーネントは、定義された要件を満たすために IPNE クライアントと連携します。

最高の IPNE パフォーマンスを実現するために、ECS コンポーネントは次の機能を提供する必要があります。

- ECS によって、フロー情報パラメータが IPNE に提供される必要があります。
  - タプル情報
  - URL
  - ユーザエージェント (User Agent)
  - アプリケーションプロトコル
  - フロー作成時間

ECS によって、NBR 情報パラメータが IPNE に提供される必要があります。

- NAT-IP アドレス
- 開始ポート
- 終了ポート

ECS は、サブスクリバのコール ID でインデックス化された MINE サーバーからのクエリに対応する応答内のすべてのアクティブフローに対して、上記のパラメータを提供する必要があります。

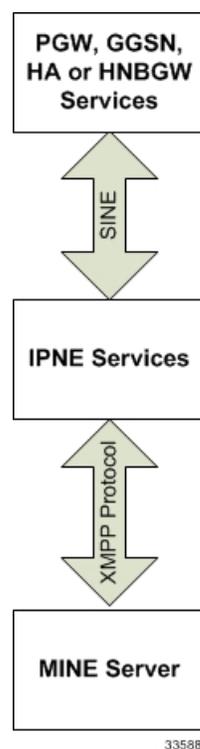
サブスクリバのコール ID に対して IPNE クライアントによってインストールされたサブスクリプションの場合、ECS は、サブスクリプショントリガーが検出されるたびに通知メッセージを IPNE クライアントに送信する必要があります。

## 機能の仕組み

### IPNE

次の図は、IPNE インターフェイスのアーキテクチャについて説明しています。セッションマネージャと IPNE は、SINE インターフェイスを介して対話します。情報は、clp ハンドルの形式でモジュール間で交換されます。セッションごとに1つの IPNE ハンドルが作成されます。情報は、IPNE クライアント側のローカルデータベースに保存されます。

図 1: IPNE アーキテクチャの概要



対話は、次のタイミングで行われます。

- IPNE 側でセッション情報を追加するためのセッションセットアップ
- フィードメッセージを MINE サーバーに渡す
- MINE サーバーから送信されたクエリ要求に応答しているとき
- IPNE クライアントから MINE サーバーへのサブスクリプション通知

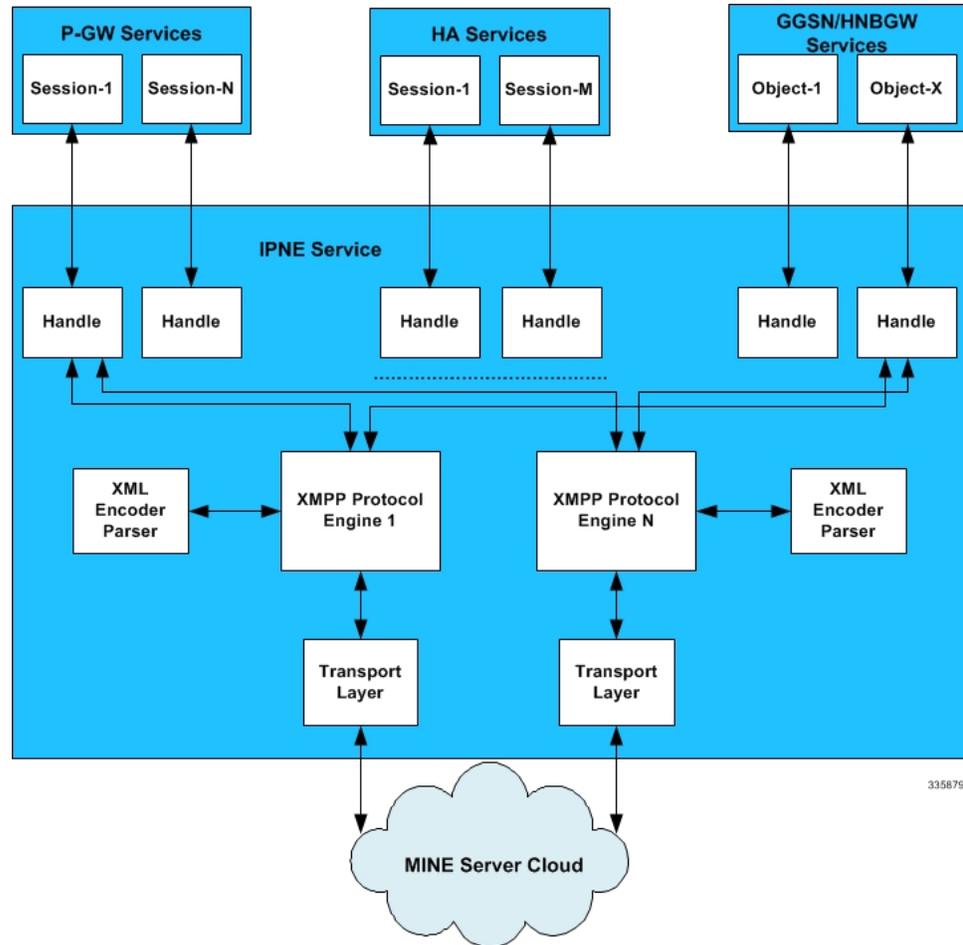
MINE サーバーと IPNE クライアントは、すべての手順において XMPP プロトコルを使用して SINE インターフェイスを介して相互に対話します。IPNE クライアント側に保存されている情報は XML 形式に変換されて MINE サーバーに渡されます。MINE サーバーからメッセージ（クエリ要求）を受信すると、IPNE はそれをデコードし、対応する clp ハンドルをセッションマ

ネージャに送信します。共有される情報は、その時点のセッション/フロー/nbr コンテキストのスナップショットです。

## アーキテクチャ

MINE IPNE クライアントは、次に示すように、P-GW、HA、GGSN、HNBGW サービス上の設定可能なサービスとして実装されます。

図 2: 詳細な IPNE アーキテクチャ



## 制限事項

IPNE 機能には、次の制限があることに注意してください。

- IPNE サービスは、XMPP インターフェイスを介してフロー制御メカニズムを実装します。その結果、設定されたキューのしきい値を超えるこのインターフェイスを介したメッセージングはすべて廃棄されます。

## フロー

この項では、IPNE クエリ、サブスクリプション、フィード、追加、および削除のシナリオのコールフロー図を示します。一部のフロー図では、例として P-GW を使用していますが、GGSN、HA、および HNBGW にも適用されます。

図 3: MINE サーバーからのクエリの IPNE 処理

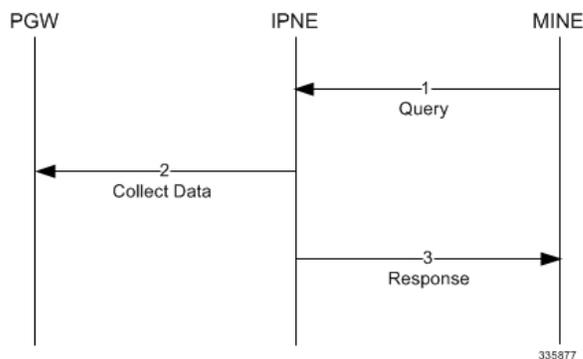


表 1: MINE サーバーからのクエリの IPNE 処理

ステップ	説明
1	MINE サーバーは XMPP ストリームを介して IPNE サービスにクエリを送信します。クエリは XML でエンコードされ、 <code>query-id</code> 、セッションをルックアップするキー（例： <code>sessmgr instance:callid</code> ）、および関連する情報を指定するセグメントのリストが含まれます。
2	クエリを受信すると、IPNE サービスは XML データを解析し、MINE サーバーから提供されたキーを使用してハンドルを特定し、登録済みのコールバック関数を呼び出してセッション情報を収集します。要求された情報は、ビットマスクの形式でもコールバック関数に提供されます。
3	XML エンコーダを使用して、IPNE サービスはセッション情報を XML 形式に変換し、MINE サーバーに送信します。

図 4: MINEサーバーからのサブスクリプションを処理する IPNE サービス

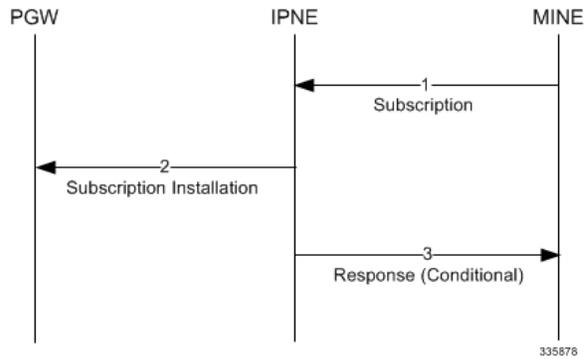


表 2: MINEサーバーからのサブスクリプションを処理する IPNE サービス

ステップ	説明
1	MINE サーバーは XMPP ストリームを介して IPNE サービスにサブスクリプションを送信します。サブスクリプションは XML でエンコードされ、クエリメッセージと同様の形式（フィードトリガーを指定するフラグメントのリストなど）となります。
2	サブスクリプションのインストールは、IPNE によってハンドルごとに維持されます。
3	このステップは条件付きです。サブスクリプションにリストされているトリガーのいずれかに一致する既存のセッションがある場合、成功確認応答メッセージが MINE サーバーに送信されます。

図 5: IPNE サービスが MINEサーバーに要求されていないフィードメッセージを送信する

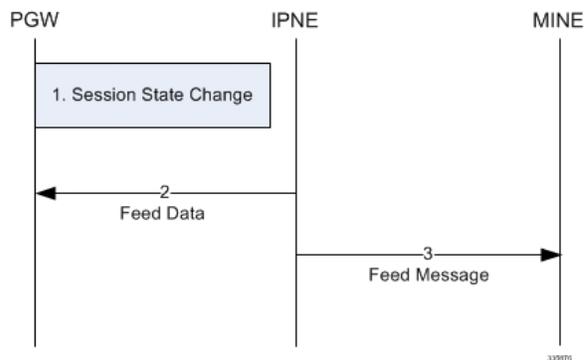


表 3: IPNE サービスが MINE サーバーに要求されていないフィードメッセージを送信する

ステップ	説明
1	セッションで何らかの状態変更（ハンドオフによる RAT の変更など）が検出されます。
2	セッションは、ハンドル上のパブリック API を呼び出して、IPNE サービスに変更を通知します。
3	変更が IPNE ハンドルにインストールされているサブスクリプションのインストールのいずれかと一致する場合、フィードメッセージが作成されて MINE サーバーに送信されます。

図 6: IPNE セッションの追加

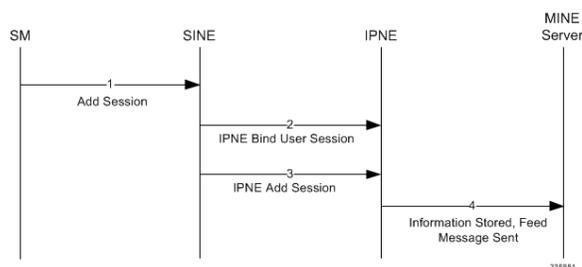


表 4: IPNE セッションの追加

ステップ	説明
1	セッションのセットアップ中に、セッション マネージャ アプリケーションは IPNE が有効かどうかを確認します。
2	IPNE が有効になっている場合、SM はセッション追加情報を SINE インターフェイスに送信します。
3	SINE はセッション情報をバインドし、IPNE アプリケーションに向けて追加イベントを送信します。
4	情報が保存され、その情報がフィードメッセージとして XML データの形式で MINE サーバーに渡されます。

図 7: IPNE セッションの削除

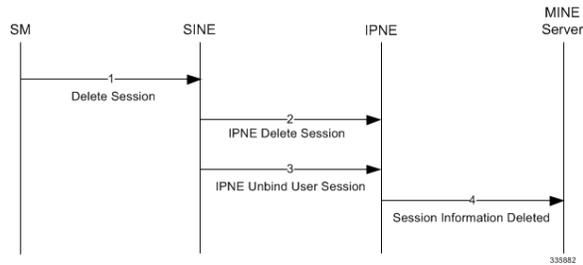


表 5: IPNE セッションの削除

ステップ	説明
1	セッションを解放する前に、セッションマネージャアプリケーションは <code>delete_session</code> を要求します。
2	SINE は、セッション削除イベントを呼び出します。
3	SINE はセッション情報のバインドを解除します。
4	対応するセッション情報が IPNE アプリケーションから削除され、タイプが <code>delete</code> であるフィードメッセージが MINE サーバーに送信されます。

## 標準準拠

StarOS IPNE 機能は、次の標準規格に準拠しています。

- RFC 6120 Extensible Messaging and Presence Protocol (XMPP) : コア、セクション 4.7 (ストリーム属性)

## IPNE 機能の設定

ここでは、IPNE 機能の設定方法と、設定の確認方法について説明します。

### IPNE の設定

ここでは、IPNE 機能の設定方法について説明します。

IPNE の設定には、IPNE サービスの設定と、IPNE サービスと GGSN、HA、HNBGW、または P-GW サービスとの関連付けが含まれます。

次の設定例を使用して、IPNE サービスを作成します。

```

config
  context context_name
  ipne ipne_service_name
  
```

```

ipne-endpoint
bind [ ipv4 ipv4_address | ipv6 ipv6_address ]
peer [ ipv4 | ipv6 ] protocol tcp
end

```

注：

- **bind** と **peer** キーワードは両方とも、IPv4 および IPv6 アドレッシングをサポートします。
- **tcp** はデフォルトのトランスポートプロトコルです。現時点では、SCTP はサポートされていません。
- デフォルトの XMPP プロトコルポートは 5222 です。
- 現時点では、**fqdn**、**priority** および **weight** キーワードはサポートされていません。

HNBGW のみ。通常、通知メッセージはサブスクリプションでのみ送信されます。ただし、HNBGW UE の登録/登録解除については例外が生じています。HNBGW UE の登録/登録解除は、サブスクリプションなしで常に通知されます。このような要求していない通知の送信を制御するには、次のコマンドを入力します。

```

configure
context ipne_service_name
    unsolicited-notify-trigger hnb-ue
end

```

注

- **unsolicited-notify-trigger hnb-ue** が設定されている場合、IPNE サービスは HNBGW からの要求を受信した時点で、UE 登録/登録解除要求の通知を送信します。
- **no unsolicited-notify-trigger hnb-ue** が設定されている場合、IPNE は UE 登録/登録解除通知を送信しません。これがデフォルトの設定です。

IPNE サービスが作成されたら、設定済みの GGSN、HA、P-GW、または HNBGW サービスに関連付ける必要があります。次の例を使用して、IPNE サービスを設定済みのゲートウェイサービスに関連付けます。

```

configure
context gw_context_name
associate ipne-service ipne_service_name
end

```

注：

- **context gw\_context\_name** は、StarOS で設定されている GGSN、HA、P-GW、または HNBGW サービスの名前です。
- IPNE サービスとゲートウェイサービスとの関連付けを削除するには、**no associate ipne-service** コマンドを使用します。

## IPNE 設定の確認

この項では、IPNE 設定を確認する方法について説明します。

Exec モードで次のコマンドを発行して IPNE 設定を確認します。

### **show ipne peers all**

このコマンドの出力は、各 IPNE サービスインスタンスに関する次の情報を提供します。

- IPNE サービス名
- コンテキスト ID
- ピア IP アドレス
- ピアへの TCP 接続の状態

## IPNE サービスのモニタリング

ここでは、StarOS IPNE 機能をモニターする方法について説明します。

### IPNE show コマンド

ここでは、StarOS IPNE 機能をサポートする show コマンドとその出力について説明します。

ここに示されている show コマンドは、StarOS IPNE 機能をサポートするために使用できます。

#### **show ipne peers all**

このコマンドは、各 IPNE サービスのピアのリストと、TCP 接続の状態を表示します。

#### **show ipne statistics all**

このコマンドは、各 IPNE サービスのハンドルの総数と、クエリ、応答、サブスクリプション、およびフィールドのカウンタの合計を表示します。

#### **show active-charging subscribers full all**

このコマンドは、MINE サーバーが現在この ACS セッションの通知を登録しているかどうか (**IPNE enabled** または **disabled**) を表示します。また、この ACS セッションについて MINE サーバーに送信された通知の数も示します。現在のフローと削除されたフローすべてに渡って通知数の履歴が保存されます。MINE サーバーが通知用に登録されていない場合、このフィールドには **n/a** が表示されます。

## 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。