

# IPv6 評価・検証システム 要求仕様書 ( Rev. 1.0 )

岡部宣夫 ( Nobuo\_Okabe@yokogawa.co.jp )

# 目次

---

1Conformance Test .....	4
1.1 テスト仕様 .....	4
1.1.1 参照 .....	4
1.1.2 評価仕様 .....	6
1.1.3 判定結果 .....	6
1.1.4 評価テスト分類 .....	6
1.2 評価ツールの機能 .....	7
1.2.1 パケット定義 .....	7
1.2.2 パケット送出 .....	7
1.2.3 パケット取込み .....	7
1.2.4 ログ機能 .....	8
1.2.5IPv4 対応 .....	8
1.2.6IPsec 対応 .....	8
1.2.7 複数インターフェイス対応 .....	8
1.2.8 フラグメント対応 .....	8
1.2.9 実行スピード .....	8
1.2.10 自動化 .....	9
1.2.11 デバッグ機能 .....	9
1.2.12 取込んだパケットの逆アセンブル .....	9
1.2.13 プログラムの再利用化 .....	9
1.3 評価ツールの構造 .....	11
1.3.1 プラットフォーム .....	11
1.3.2 プログラム構造の概要 .....	11
1.3.3 テスト条件定義ファイル .....	12
1.3.4 パケット定義ファイル .....	13
1.3.5 テスト・シーケンス定義ファイル .....	13
1.3.6 リモート制御定義ファイル .....	13
1.3.7 シーケンサ .....	13
1.3.8 パケット出力器 .....	13
1.3.9 パケット比較器 .....	14
1.3.10 パケット・メモリ .....	14
1.3.11 ログファイル .....	14
1.3.12 リモート制御 .....	15
1.4 テスト環境 .....	17
1.4.1 ホストのテスト環境 .....	17
1.4.2 ルータのテスト環境 .....	17
1.4.3 制約 .....	18
2Robustness Test .....	19
2.1 仕様に関するテスト .....	19
2.2 計算機資源に関するテスト .....	19
2.3 不正なパケットに関するテスト .....	19



# 1 Conformance Test

Conformance Test は、IPv6 の仕様に基づいているかを評価するテストである。

## 1.1 テスト仕様

### 1.1.1 参照

評価仕様は以下のドキュメントに基づく：

表 1. IPv6 関連

機能	RFC、Internet Draft
基本仕様	<ul style="list-style-type: none"><li>• draft-ietf-ipngwg-ipv6-spec-v2-02.txt</li></ul>
Address Architecture	<ul style="list-style-type: none"><li>• draft-ietf-ipngwg-addrconf-v2-02.txt</li></ul>
Address Configuration	<ul style="list-style-type: none"><li>• draft-ietf-ipngwg-addrconf-v2-02.txt</li></ul>
Neighbor Discovery	<ul style="list-style-type: none"><li>• draft-ietf-ipngwg-discovery-v2-03.txt</li></ul>
ICMPv6	<ul style="list-style-type: none"><li>• draft-ietf-ipngwg-icmp-v2-02.txt</li></ul>
ICMP Name Lookup	<ul style="list-style-type: none"><li>• draft-ietf-ipngwg-icmp-namelookups-02.txt</li></ul>
トンネル	<ul style="list-style-type: none"><li>• draft-ietf-ipngwg-ipv6-tunnel-08.txt</li></ul>
Router Alert	<ul style="list-style-type: none"><li>• draft-ietf-ipngwg-ipv6router-alert-04.txt</li></ul>
Jumbo Payload	<ul style="list-style-type: none"><li>• draft-ietf-ipngwg-jumbo-00.txt</li></ul>
Multicast Listener Discovery	<ul style="list-style-type: none"><li>• draft-ietf-ipngwg-mlt-00.txt</li></ul>
Router Renumbering	<ul style="list-style-type: none"><li>• draft-ietf-ipngwg-router-renum-05.txt</li></ul>
Path MTU Discovery	<ul style="list-style-type: none"><li>• rfc1981.txt</li></ul>

表 2. IPsec 関連

機能	RFC、Internet Draft
基本仕様	<ul style="list-style-type: none"><li>• draft-ietf-ipsec-arch-sec-07.txt</li></ul>
AH	<ul style="list-style-type: none"><li>• draft-ietf-ipsec-auth-header-07.txt</li></ul>
ESP	<ul style="list-style-type: none"><li>• draft-ietf-ipsec-esp-v2-06.txt</li></ul>
IKE	<ul style="list-style-type: none"><li>• draft-ietf-ipsec-dhless-enc-mode-00.txt</li><li>• draft-ietf-ipsec-isakmp-10.txt</li><li>• draft-ietf-ipsec-isakmp-oakley-08.txt</li><li>• draft-ietf-ipsec-oakley-02.txt</li><li>• draft-ietf-ipsec-doi-10.txt</li></ul>

更に、以下の機能に関しては、スケジュールの余裕に応じて、実装する可能性がある。

表 3. 保留項目

機能	RFC、Internet Draft
ICMP Name Lookup	<ul style="list-style-type: none"> <li>• draft-ietf-ipngwg-icmp-namelookups-02.txt</li> </ul>
トンネル	<ul style="list-style-type: none"> <li>• draft-ietf-ipngwg-ipv6-tunnel-08.txt</li> </ul>
Router Alert	<ul style="list-style-type: none"> <li>• draft-ietf-ipngwg-ipv6router-alert-04.txt</li> </ul>
Jumbo Payload	<ul style="list-style-type: none"> <li>• draft-ietf-ipngwg-jumbo-00.txt</li> </ul>
Multicast Listener Discovery	<ul style="list-style-type: none"> <li>• draft-ietf-ipngwg-mld-00.txt</li> </ul>
Router Renumbering	<ul style="list-style-type: none"> <li>• draft-ietf-ipngwg-router-renum-05.txt</li> </ul>

### 1.1.2 評価仕様

評価仕様は以下の要件を満たしていること：

- Traceability：リファレンス（ID、RFC）の記述と個々評価の対応付けが明確であること。
- Coverage：評価仕様のリファレンスに対するカバレッジが明確であること。
- 評価条件（ネットワーク環境、手順、印加条件、判定条件）が明確であること。

### 1.1.3 判定結果

評価テストは、判定結果として以下の要件を満たすこと：

- 個々の評価テストは、GO/NG 判定
- 評価ツールは、ID や RFC の定義の強さ（must/required/shall/must-not、should/recommend/should-not、may/may-not）を考慮しない。
- つまり、must の要求項目に関するテストでも should の要求項目に関するテストでも、フェイルしたら結果は同じ NG として表現する。
- その代わりに、テストを定義の強さに応じて分類する。

### 1.1.4 評価テスト分類

評価テストは、以下の分類がされていること：

- 仕様としての強さ：  
RFC、Internet Draft に定義されている要求項目の強さ<sup>1</sup> に応じた分類
  - 必須：must/required/shall/must-not
  - 推奨：should/recommend/should-not
  - 選択自由：may/may-not
- ノードの種類：  
評価対象となる IPv6 ノードの種類
  - ホスト
  - ルータ
  - 共通
- 機能的な分類：  
RFC、Internet Draft の機能的な分類に応じた分類

---

1. 強さ自体は rfc2119 に定義されている。

## 1.2 評価ツールの機能

評価ツールは、以下の要求を満たす必要がある。

### 1.2.1 パケット定義

- パケット定義は再利用できること。
  - 構文的な拡張が容易なこと。
  - ネットワーク・インターフェイス依存していないこと。
  - トンネル・パケットの記述ができること。
  - IPsec パケットな記述ができること。
  - パケットを構成する全てのフィールド定義を明示的に書かなくとも、適当なパケット定義ができること。
  - 不正なパケット、壊れたパケットの定義ができること。  
つまり、一貫性に欠けたパケットも定義できること。
  - 設定記述としては、省略 / 明示 / デフォルトが可能なこと。
  - 特に、デフォルト記述では、そのフィールドに合った値をとること。
  - 取り込むパケットを定義するためには、以下の記述が必要である：
    - マスク定義 ( ex. 0x1000#0xF000 )
    - 数値の範囲指定 ( ex. 100:200 )
  - 取り込みパケットの定義では、拡張ヘッダと省略可能なパラメータやオプションの記述に対応できること：
    - 明示的に定義したデータと正確に比較する ( must )
    - そのデータの前後に別の省略可能なヘッダ、パラメータ、オプションがあっても一致とみなす。 ( want )
- 少なくとも、構文的にはこれらの機能が拡張できるようなものにする。 ( 実装は後でもよし )

### 1.2.2 パケット送出

- 任意のパケットを出力できること。

### 1.2.3 パケット取込み

- 任意のパケットを取り込めること。
- 取り込むパケットのパターンを定義出来ること。
- 複数のパケット条件で待てること。
- 取り込み条件で待っている時に、タイムアウトできること。
- テスト中のパケットを取りこぼさないこと。  
取り込みパケットのバッファリングする
- パケットを取りこんだ時刻がわかること。

#### 1.2.4 ログ機能

- テスト中の動作をログで記録できること。
- 特に、出した / 読んだパケットの特徴を端的に表現できること。
- テスト結果のログは、実際にネットワーク上を流れたパケットの順序で記録すること。
- テスト結果のログに記録されるタイミングは、リアルタイム性を要求しない。つまり、実際にネットワーク上にパケットが流れ時に、ログにそれが記録される必要はない。
- テスト結果をまとめたりレポートを生成できること。
- レポートに必要な項目は：
  - テスト名
  - テスト結果
  - フェイルに関する情報

#### 1.2.5 IPv4 対応

- トンネルに対応するため、本ツールは IPv4 パケットの送出、取り込みにも対応する。
- IPv4 のアドレス解決は、プラットフォームの機能を利用し、独自の実装はしない。
- 複数の IPv4 アドレスが必要な場合には、プラットフォームの alias 機能を利用する。

#### 1.2.6 IPsec 対応

- 送出するパケットの暗号化ができること。
- 送出するパケットのダイジェストを生成し、送出するパケットの任意の位置に代入できること。
- 取り込んだパケットの復号化ができること。
- 取り込んだパケットのダイジェストを計算できること。
- 任意の暗号アルゴリズム、ハッシュ関数を選択できること。

#### 1.2.7 複数インターフェイス対応

- ルータやマルチ・ホームの試験が出来ること。
- なので、複数のインターフェイス環境下の検証ができること。

#### 1.2.8 フラグメント対応

- 任意のパケットの一部を切り出し、それをつなげることで別のパケットを定義できること。

#### 1.2.9 実行スピード

- 本検証プログラムの時間的な精度は、0.1 秒とする。

- 例えば、パケット定義を実行時に翻訳することが、この精度に影響を与えるのなら、事前に翻訳を完了するなどの対策を考えること。

### 1.2.10 自動化

評価ツールで以下のことができねばならない：

- 検証対象の操作（設定、リブートなど）が可能なこと。
- 検証対象の状態が、期待した状態か否か確認できること。

### 1.2.11 デバッグ機能

- シーケンスに対して、再コンパイルなどの特別な作業なしに、デバッグ機能が利用できること。
- 特定の設定点と測定点でシーケンスの実行をポーズできること。
  - パケットを出力した直後
  - パケット待ちから返った直後
- ポーズ中に以下の操作ができること。
  - 取込んだパケットの内容を見られること。
  - 出したパケットの内容を見られること。
  - パケットの測定条件を確認できること。
  - パケットの測定条件を変更できること。
  - 任意のパケットを出力できること。
  - 任意のパケットを待てること。
  - デバッグ中の表示内容をファイルへ保存できること。
  - デバッグ・コマンドの履歴機能があること。
  - デバッグ・コマンドのインライン編集ができること。
  - デバッグ・コマンドのオンラインヘルプがあること。
  - デバッグ・コマンド入力の補完ができること。
  - シーケンスの実行を、ポーズしたところから再開（Step と Continue）できること。

### 1.2.12 取込んだパケットの逆アセンブル

- 取込んだパケットを、パケット定義の書式で表現できること。
- 結果は、ログ・ファイルなどへの出力でよい。

### 1.2.13 プログラムの再利用化

たとえば、以下の状況でプログラムを可能な限り再利用できねばならない：

- 同じテスト・プログラムを異なる機種に対して実行する。この場合、リモート操作の部分が異なるが、それ以外（テストのシーケンスやパケットの定義）は同じ定義を再利用できること。

- 同じテストのアルゴリズムで検証するが、一部のテスト条件を変える。個の場合、テストのシーケンスは同じだが、それ以外（リモート操作、パケットの定義）の一部だけが異なる。同じ定義は再利用できること。

## 1.3 評価ツールの構造

前述した要求を満たすための評価ツールの構造をまとめる。

### 1.3.1 プラットフォーム

- IPv6 スタックを持っていない unix を使う。
- これは、プラットフォームの副作用を除去するため。

### 1.3.2 プログラム構造の概要

- 構造は、
  - 各種定義ファイル (テスト条件、テスト・シーケンス、パケット、リモート制御)
  - テスト結果などのログ・ファイル
  - シーケンサ
  - パケット出力器
  - パケット比較器
  - パケット・メモリ
  - リモート制御
- 各機能は、独自に単体テストが出来ること。
- BPF を使う。
- 複数インターフェイスに対応するため、複数の BPF を扱えること。
- パケットを取りこぼせないので、BPF は開きっぱなし。
- なので、独立したバッファリング機能が必要。
- シーケンサの動作とパケットのバッファリングは非同期で動くので、何らかの同期手段が必要
  - バッファ制御 (取込み開始、取込み終了、バッファ・クリア)

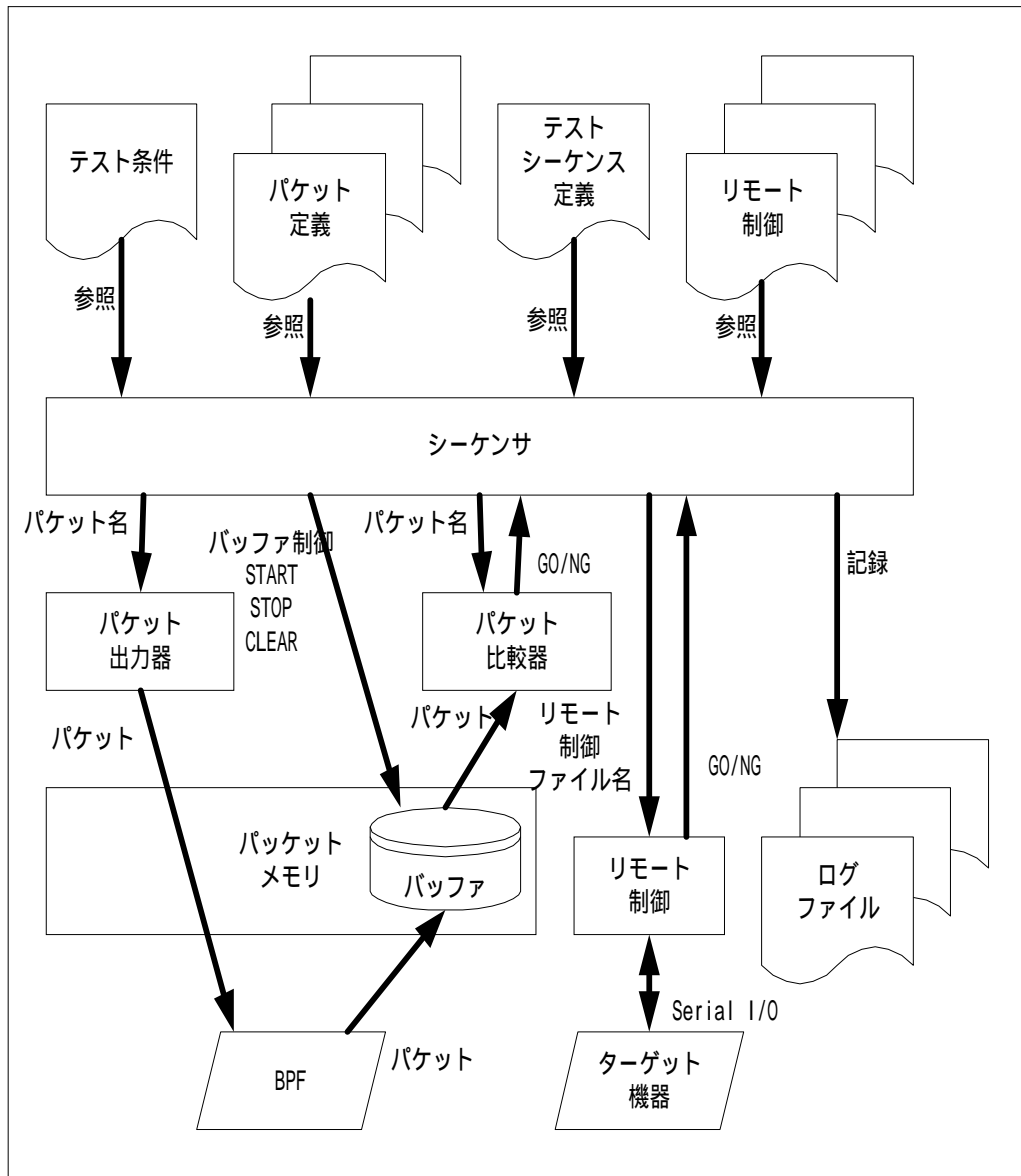


図 1. 評価テスト・ツールの構成

### 1.3.3 テスト条件定義ファイル

テスト条件定義ファイルでは、一連のテストを通して共通な定義をまとめて定義できる。

- ASCII テキスト形式
- ファイル拡張子は、.def。
- テストに用いる物理インターフェイス
- テスト・ツールのリンク層アドレス (IPv4?、IPv6)
- テスト・ツールの IP 層アドレス (IPv4?、IPv6)
- 各種定義ファイルの置き場
- 各種ログファイルの置き場

### 1.3.4 パケット定義ファイル

- ASCII テキスト形式
- ファイル拡張子は、.pkt
- パケット出力器、パケット比較器の双方で参照するパケットを定義できる。
- パケット出力器が、期待値固有の記述をしたパケット定義を参照した場合、実行時エラーとなる。

### 1.3.5 テスト・シーケンス定義ファイル

- 一つのテストの流れを定義する。
- 単純なシーケンス程度で記述する。
- それ以上複雑な記述は、別の部分（パケット定義など）で隠蔽する。
- ASCII テキスト形式、実体は Perl スクリプト
- 拡張子は、.seq
- テスト・ツールとして共通に利用する機能はライブラリとして提供する。

### 1.3.6 リモート制御定義ファイル

- ASCII テキスト形式
- 拡張子は、.rmt
- テスト対象となる機器の操作を定義する。

### 1.3.7 シーケンサ

- テスト・シーケンスを実行する。
- 実体は、Perl と Perl ライブラリから成る。
- テスト結果を、返り値として返す
- デバッグ機能を有する（ページ 9 の「デバッグ機能」）。

### 1.3.8 パケット出力器

- UNIX コマンドとして実装
- パケット定義に従ったパケットを出力する。
- 引数
  - パケット定義ファイル名
  - パケット名
- 返り値
  - GO/NG

### 1.3.9 パケット比較器

- UNIX コマンドとして実装
- パケット定義に従ったパケットを待つ。
- パケット待ちをタイムアウトできる。
- 引数
  - パケット定義ファイル名
  - パケット名
  - タイムアウト
- 返り値
  - GO/NG

### 1.3.10 パケット・メモリ

- UNIX デーモンとして実装
- 一つのパケット・メモリは、一つインターフェイスにしか対応しない。複数のインターフェイスに対応するテストを行うためには、その分だけパケット・メモリを起動する。
- BPF から取込んだパケットに対して、有限個をバッファして、パケット比較器へ渡す。
- バッファがいっぱいになったときの制御は選択可能とする。
  - バッファリング終了
  - 上書き
- バッファリングの制御（スタート、ストップ、クリア、上書きしない、上書きする）は UNIX ドメイン・ソケットでインターフェースする。

### 1.3.11 ログファイル

- テスト結果：
  - ASCII テキスト形式
  - ファイル拡張子は、.log
  - ファイルは、カレントディレクトリ下の log で生成される（デフォルト）
  - テスト結果のログ・レベル切り替えができる：

表 4. テスト結果のログ・レベル

ログ レベル	内容
0	<ul style="list-style-type: none"> <li>• ログを出力しない。</li> </ul>
1	<ul style="list-style-type: none"> <li>• 各イベント毎に、ログを一行に集約して記録する。</li> <li>• 以下のイベントでログを取る： <ul style="list-style-type: none"> <li>- パケットを送信した時</li> <li>- パケットを受けた時</li> <li>- パケットの待ちに入った時</li> <li>- パケットの待ちから抜けた時</li> </ul> </li> <li>• 以下の情報を一行に集約して表現する： <ul style="list-style-type: none"> <li>- 時刻</li> <li>- 上位プロトコルの種別<sup>a</sup></li> <li>- Source Address/port、Destination Address/port<sup>b</sup></li> <li>- 拡張ヘッダ<sup>c</sup></li> </ul> </li> </ul>
2	<ul style="list-style-type: none"> <li>• ログ・レベル0の表示に加えて、送受信したパケットヘッダー部<sup>d</sup>の逆アセンブル表示を行う。</li> <li>• ヘッダー部の逆アセンブル表示の部分は、複数行で表現して可。</li> </ul>
3	<ul style="list-style-type: none"> <li>• ログ・レベル0の表示に加えて、送受信したパケットのヘッダー部<sup>e</sup>を16進形式で表示する。</li> <li>• ヘッダー部の16進形式の表示部分は、複数行で表現して可</li> </ul>
4	<ul style="list-style-type: none"> <li>• ログ・レベル0の表示に加えて、送受信したパケット全体を16進形式で表示する。</li> <li>• パケット全体の16進形式表示の部分は、複数行で表現して可。</li> </ul>

a. ICMP の場合、ICMP TYPE も合わせて表示する。

b. プロトコルが TCP、UDP 以外の場合、Port 番号は意味がないので、表示しない。

c. 拡張ヘッダ毎に英文字を1文字割り当て、その文字列で拡張ヘッダ列を表現する。

d. 上位層のペイロードを除いた部分

e. 上位層のペイロードを除いた部分

• テスト・サマリー：

- HTML 形式

- ファイル名は、log/summary.html ( デフォルト )

### 1.3.12 リモート制御

• テスト対象の機器を、シリアル線経由で制御する。

• 制御内容や期待値は、リモート制御ファイルで定義する。

• 引数

- リモート制御ファイル名

- 返り値
  - GO/NG

## 1.4 テスト環境

### 1.4.1 ホストのテスト環境

- 10/100 イーサネット
- 他のネットワークから切り離し、孤立したネットワーク下でテストする。
- 孤立したネットワーク下で、テスターと検証対象のホストをつなぐ。

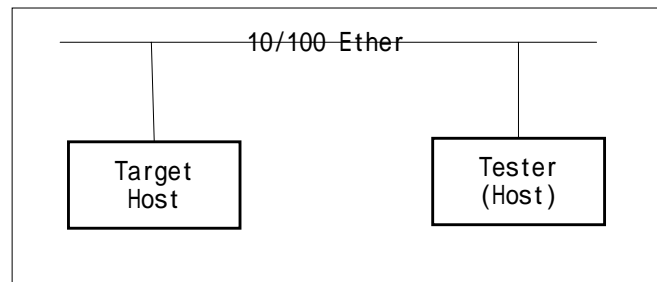


図 2. ホストのテスト環境

### 1.4.2 ルータのテスト環境

- 10/100 イーサネット
- 他のネットワークから切り離し、孤立したネットワーク下でテストする。
- 孤立したネットワーク下で、テスターと検証対象のルータをつなぐ。
- テスターは、マルチホーム構成にし、ルータの複数のインターフェイスを検証する。

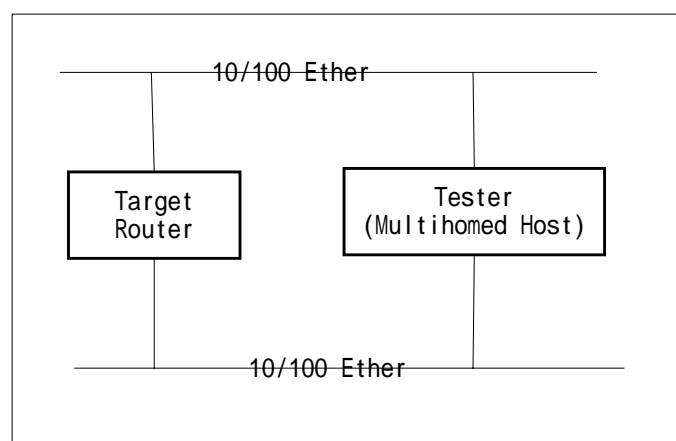


図 3. ルータのテスト環境

### 1.4.3 制約

TBD

## 2 Robustness Test

- Robustness Test は、Conformance Test を同じ手段で実現する。
- 仕様（RFC、ID）で明示されている禁止事項（must not、should not）と本テストが重複する場合には、Conformance Test の範疇で行う。

### 2.1 仕様に関するテスト

- 仕様上曖昧なフォーマットの packets を投げる。
- 最新の仕様で obsolete になった部分をテストする。

### 2.2 計算機資源に関するテスト

- メモリ資源の獲得 / 開放を明示的に繰り返す。
  - Neighbor Cache
  - Destination Cache
  - Default Router List
  - Prefix List
- メモリ試験の獲得 / 開放を暗黙的に繰り返す。
  - Default Router List vs. Router Lifetime
  - Prefix List vs. Prefix Lifetime

### 2.3 不正な packets に関するテスト

- 各種ヘッダーのフィールドの値
- データ長の不一致
- チェックサム的一致
- データ・アライメントの侵害
- 不定なデータ型、データ ID
- おかしなフラグメント順序
- ハッシュ・アルゴリズム、暗号アルゴリズムの不一致
- ハッシュ・データ、暗号データの不一致
- 異常に短い間隔で到着する packets