NORTi TCP/IPv6

ユーザーズガイド

2010 年 4 月版

株式会社ミスポ

2010 年 4 月版で改訂された項目

ページ	更新内容
P32	IPV6_ADDRANY を IP アドレスに指定した場合に設定されるアドレスに誤記があったのを修正

2009 年 3 月版で改訂された項目

ページ	更新内容
P15	「3.6 互換性」に「使用する資源」を追加
P18	「3.9 アドレスの追加と取得」を「3.9 アドレスの追加、削除、取得、状態通知コールバック」へ変更し、内容を更新
P33、34	ICMPv6 コールバックに関する説明を修正

2008 年 9 月版で改訂された項目

ページ	更新内容
P6	ファイル構成に nonmcast2.c を追加
P24	暗復号/ハッシュアルゴリズムの修正

2006 年 11 月版で改訂された項目

ページ	更新内容
P6	nonmcast2.c を追加
P17	MLD（マルチキャストリスナー探索）の記述を変更
P18	「3.9 アドレスの追加と削除」を追加

2006 年 4 月版で改訂された項目

ページ	更新内容
P6	ライブラリファイル名を encrxxxx.lib から cryptxxxx.lib に変更

2005 年 10 月版で改訂された項目

ページ	更新内容
P5	1.4 フォルダ構成のディレクトリパスを変更
P30	IPv6 のサンプルアプリケーションにファイルパスを追加

2005 年 8 月版で改訂された項目

ページ	更新内容
P14	「3.4 ICMPv6 と近隣探索」にマルチキャストリスナー照会／報告／終了を追加
P16	「3.8 MLD（マルチキャストリスナー探索)」を追加
P17	「3.9 制限事項」から MLD の記述を削除

目次

第 1 章 導 入

1.1 はじめに

本書では、NORTi TCP/IPv6 に関連する説明が記載されています。NORTi TCP/IPv6 をご使用になる前に、および、お問い合せになる前に、本書をよくお読みいただけますようお願いします。従来の TCP/IP Version4 や NORTi に関連する事項については、次のドキュメントをご覧ください。

NORTi Ver.4 ユーザーズガイド カーネル編 (no4guid.pdf)
NORTi Ver.4 ユーザーズガイド TCP/IP 編 (n4nguide.pdf)
NORTi Version4 ユーザーズガイド 補足説明書 (n4update.pdf)

1.2 特長

・IPv4/IPv6 デュアルスタック
・既存の IPv4 アプリケーションがそのまま動作可能です
・既存の Ethernet ドライバをそのまま利用可能です
・IPsec 対応でより機密性の高いデータの通信が可能になります
・IPv6 Ready Logo を獲得しています。(Phase 1; Logo ID: 01-000179)

1.3 制限事項

NORTi 以外での動作は保証対象外です。
このドキュメントは許可無く変更されます。

1.4 フォルダ構成

NORTi TCP/IPv6 をインストールすると 以下のフォルダに IPv6 のコードがコピーされます。

NORTi/IPV6/SRC	IPv6 固有のソースコードがインストールされます。
NORTi/IPV6/INC	IPv6 固有のヘッダファイルがインストールされます。
NORTi/IPV6/LIB	IPv6 固有のライブラリファイルがインストールされます。

注意事項

近隣探索にはマルチキャストパケットを使用するため、ドライバはマルチキャストパケットを受信できる設定になっている必要があります。

1.5 ファイル構成

nonet6.c
IPv6 に関連するコンフィグレーションやユーティリティ関数、Neighbor Discovery(近隣探索)モジュールが含まれています。

nonetip6.c
IPv6 初期化関数やベースモジュールが含まれています。

noneip6f.c
IPv6 のフラグメント処理を行う関数が含まれています。

nonicmpv6.c
ICMPv6 に関連するモジュールが含まれています。

nonipsec.c
IPsec に関連するモジュールが含まれています。

nonmcast.c
IPv6 のマルチキャストリスナー探索プロトコルに関連するモジュールが含まれています。

nonmcast2.c
IPv6 のマルチキャストリスナー探索プロトコル Ver.2 に関連するモジュールが含まれています。

nonipsec.h
IPsec で使用される変数定義などが含まれています。

ipv6dxxxx.lib
デバッグオプション付き IPv6/IPv4 デュアルスタックライブラリ。

ipv6nxxxx.lib
デバッグオプションなし IPv6/IPv4 デュアルスタックライブラリ。

ipv6dsxxxx.lib
デバッグオプション、IPsec 付き IPv6/IPv4 デュアルスタックライブラリ。

ipv6nsxxxx.lib
デバッグオプションなし、IPsec 付き IPv6/IPv4 デュアルスタックライブラリ。

cryptxxxx.lib
IPsec で使用する暗号化ライブラリ

第 2 章 IPv6 の概要

この章では IPv6 の概要を記述しています。詳細な内容に関しては各専門書などをご覧ください。NORTi TCP/IPv6 では一部未サポートの機能もあります。

2.1 IPv4 からIPv6 への変更点

IPv6 は、IPv4 を引き継いで設計されたインターネットプロトコルの新しいバージョンです。IPv4 から IPv6 への変更点は、主に下記に分類されます。

- **アドレス拡張と自動設定機能**
  より多くのノードと階層構造（アドレス体系の多様化）の使用を可能にするため、またアドレスの自動設定機能をサポートするためにアドレス長が 128bit に拡張されました。

- **ヘッダフォーマットの簡略化**
  ヘッダ長が 40byte になり、IPv4 のいくつかのフィールドがオプションとして扱われるようになりました。これによりヘッダ処理の負荷が軽減されます。

- **拡張機能とオプションのサポート改善**
  IPv4 のオプションはヘッダに含まれていましたが、IPv6 では拡張ヘッダとして扱われます。拡張ヘッダは IPv6 ヘッダとペイロード（データ本体）の間に挿入されます。これにより拡張性が高まり、将来新しいオプションの導入が容易になります。

- **フローラベル機能**
  この機能は主にリアルタイム通信などに利用されます。送信元が特別な処理やサービス品質を必要とするパケットに、ラベルをつけることができます。

- **認証とプライバシー保護の機能**
  認証機能のサポートとデータの完全性および機密性のための拡張機能が追加されました。

2.2 アドレス形式

IPv4 が 32bit のアドレスを使用するのに対して、IPv6 は 128bit のアドレス空間を使用します。IPv6 には 3 つの種類のアドレスに分類されます。

- **ユニキャスト**
  ひとつのインターフェースを示します。ユニキャストアドレスに送られたパケットは、そのアドレスで認識されるインターフェースに届けられます。

- **マルチキャスト**
  IPv6 インターフェースのグループを示します。指定したマルチキャストアドレスに送信されたパケットは、そのマルチキャストグループに属する全てのメンバーに届けられます。

- **エニーキャスト**
  複数のインターフェースに対して付与されます。エニーキャストは一般的に複数の異なったノードに属しています。エニーキャストアドレスに対して送付されたパケットは、そのうちの 1 つのインターフェースのみに届けられます。通常は最も隣接したインターフェースに届けられます。

2.3 アドレス表記

IPv6 では 128bit(16byte)のアドレスを使用します。アドレスは 16 ビットごとに 8 つに区切りそれぞれを 16 進の 4 桁で表し、コロン(:)で区切ります。

FE80:0000:0000:0000:0240:CAFF:FE28:4ADA

16bit 区切りが連続して 0 の場合かアドレスの先頭や末尾が 0 の場合は、これをコロン(:)に置き換えることができます。

FE80::240:CAFF:FE28:4ADA

ただし、2 つのコロンはアドレス内の 1 箇所でしか使用できません。

FE80:0000:0000:0000:0234:0000:FE78:9A9A

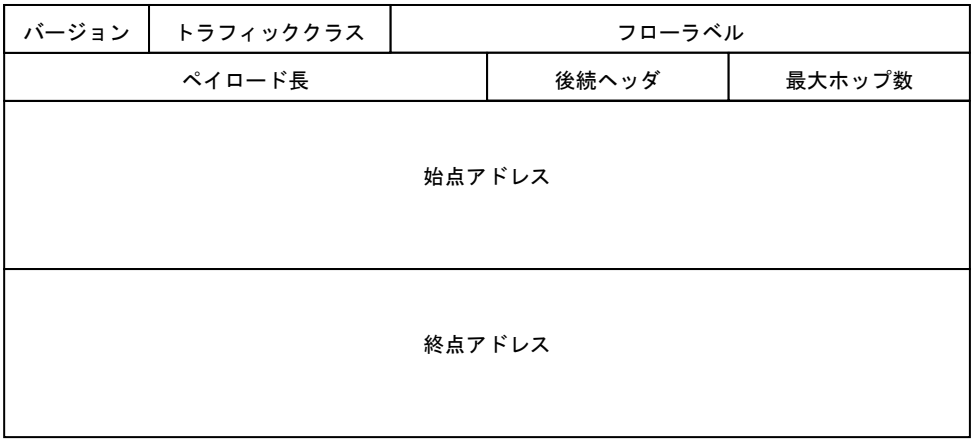
の場合は

FE80::234:0:FE78:9A9A

となります。

2.4 IPパケットヘッダの定義

IPv6 のヘッダ形式



・バージョン(4bit)
プロトコルのバージョン番号

・トラフィッククラス(1byte)
IPv6 パケットの属性を表す

・フローラベル(20bit)
ルータの効率化のために使用される。一連のパケットを同じように処理するようにルータに指示する番号。

・ペイロード長(2byte)
IPv6 ヘッダに続くデータ部の長さ

・後続ヘッダ(1byte)
上位プロトコルのタイプ（IPv4 のプロトコルタイプフィールドと同じ）拡張ヘッダが使用されている場合は、IPv6 ヘッダに続く拡張ヘッダの値が書き込まれる。

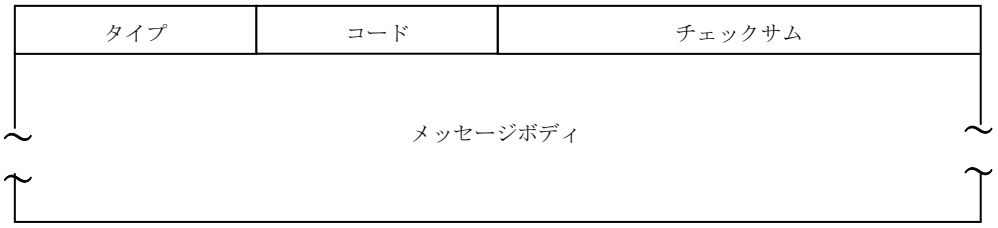
・最大ホップ数(1byte)
IPv4 の TTL に相当

・始点アドレス(16byte)
送信元の IP アドレス

・終点アドレス(16byte)
宛て先の IP アドレス

2.5 ICMP Version6

IPv4 の ICMP に代わり、IPv6 では ICMPv6 が使用されます。ICMPv6 はパケットを処理する際に発生したエラーの通知や診断（ICMPv6 "ping"）を実現するために使用されます。



・タイプ(1byte)
メッセージの種類を表す。以降のフィールドのフォーマットを規定する。

・コード(1byte)
コードフィールドはメッセージの種類によって異なり、それぞれのメッセージに応じた詳細な情報が設定される。

・チェックサム(2byte)
ICMPv6 ヘッダと IPv6 ヘッダの一部のデータ損傷を検出するために使用される。

・メッセージボディ(可変長)
それぞれのメッセージの種類に応じ、中身が異なる。

ICMPv6 のメッセージは 3 類に分類できます。

・ICMPv6 エラーメッセージ

タイプフィールドの MSB が 0 の場合はメッセージがエラーメッセージになります。
したがって、ICMP エラーメッセージの値は 0～127 の範囲になります。

1	宛先未到達
2	パケットサイズ超過
3	時間超過
4	パラメータ問題

・ICMPv6 情報メッセージ

情報メッセージは、タイプフィールドの MSB が 1 になっています。したがって ICMP エラーメッセージの値は 128～255 の範囲になります。

128	エコー要求
129	エコー応答

・近隣探索メッセージ

133	ルータ要請
134	ルータ通知
135	近隣要請
136	近隣通知
137	ICMP リダイレクト

2.6 近隣探索

近隣探索には IPv4 の ARP に相当する機能、ICMP ルータ探索、ICMP リダイレクトの機能が含まれています。また、近隣ノードと通信が可能かどうか調べる機能や、重複したアドレスを検出する機能も追加されました。近隣探索では ICMPv6 のメッセージを使用します。

2.6.1 近隣探索で使用するICMPのメッセージ

近隣探索プロトコルでは次の 5 つの ICMP メッセージを使用します。

- **ルータ要請メッセージ／ルータ通知メッセージ**
- **近隣要請メッセージ／近隣通知メッセージ**
- **ICMP リダイレクトメッセージ**

2.6.2 近隣探索の動作

近隣探索はつぎのような目的で使用されます。

１）同一リンクにあるノードのレイヤ 2（Ethernet の場合は MAC アドレス）アドレスを発見する**アドレス解決**（IPv4 の ARP に相当）

マルチキャストアドレスを使って近隣要請メッセージを送信し、受信ホストは対象アドレスに一致した場合、近隣通知メッセージを返します。

２）近隣ノードが通信可能かどうかの情報を保ち、リンク層アドレスが変更されたことを検知する**到達不能検知**

ユニキャストアドレスを使って近隣要請メッセージを送信し、該当のアドレスが有効であるかチェックします。近隣通知メッセージの応答が無い場合は、該当アドレスが無効になっている可能性があります。

３）ノードが使用しようとしているアドレスが他のノードによって既に使用中であるかアドレスの重複をチェックする**重複アドレス検知**
重複チェックを行うアドレスを設定した近隣要請メッセージを送信し、近隣通知メッセージによる応答があれば、重複しているとみなします。

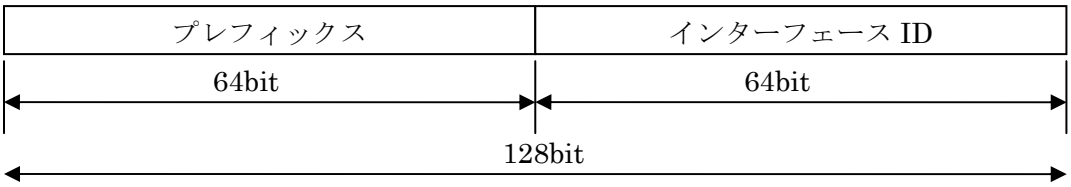
４）パケットを転送する近隣ルータを発見する**デフォルト経路の取得**
ルータは定期的にルータ通知メッセージを送信します。ホストはルータ要請メッセージを発行することにより、ルータ通知メッセージを要求することができます。

2.7 自動設定

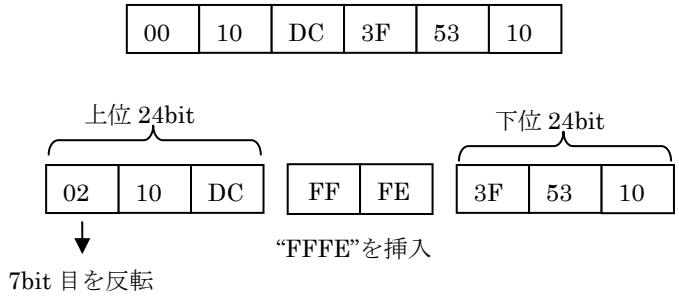
IPv4 ではホストのネットワークに関するコンフィグレーション(IP アドレスの自動取得など)は DHCP を使って行いました。DHCP を使用するには DHCP サーバが必要ですが、IPv6 では DHCP を使用しなくても IP アドレスを自動的に設定することができます。DHCP を用いてアドレスの自動設定を行う方法をステートフルといいます。これに対しホストが自分自身から生成できる情報（インタフェース ID）とルータから通知される情報（プリフィックス）を用いて自分自身のアドレスを生成する方法をステートレスといいます。ステートレス方式を使用すると、自身の IP アドレスを設定する必要がなく DHCP サーバも不要になります。この方法で作成さえたアドレスはリンクローカル・アドレスと呼ばれます。リンクローカル・アドレスはルータを越えないローカルな範囲でのみ使用できます。

IPv6 で使用する 64bit アドレスは 64bit 毎に役割が異なり、上位 64bit をプレフィックス下位の 64bit をインターフェース ID といいます。プレフィックスはネットワークを識別す

るためにしようされ、IPv4 のネットワークアドレスに相当します。インターフェース ID はネットワーク上のホストを識別するために使用され IPv4 のホストアドレスに相当します。



リンクローカル・アドレスでは同一ネットワークのみで使用するため、プレフィックスは FE80::となります。インターフェース ID に関してはホスト毎に固有である必要があり、Ethernet を使用する場合は MAC アドレスを使用してインターフェース ID を決定します。具体的には、48 ビットの MAC アドレスを 24 ビットずつ二つに分け、その間に 16 ビットの“fffe”を挿入し、先頭の 7 ビット目の 0 と 1 を反転させます。例えば、MAC アドレスが 00:10:DC:3F:53:10 とした場合以下のようになります。



アドレスは FF80::2010:DCFF:FE3F:5310 となります。
ホストはこのアドレスを使用する前に、近隣探索を使用してアドレスが重複していないかのチェックをおこない重複していなければ、このアドレスの使用を開始します。

2.8 IPsecを使ったセキュリティ管理

IPv6 では IPsec が標準で実装されます。IPsec では IP 層で**認証**と**暗号化**を行います。IPsec では IP 層でペイロード(データ)を暗/復号化して送受信を行いますので、TCP や UDP は IPsec が使用されていることを意識する必要はありません。

2.8.1 認証メカニズム

IPsec では IP パケットに完全性保証(データの改竄などが行われていないか)を行います。これを行うために認証拡張ヘッダ（**AH** :Authentication extension header)という拡張ヘッダを使用します。認証メカニズムでは一般的に MD5-96 または SHA1-96 というアルゴリズムが使用されます。

2.8.2 暗号化

IPsec ではデータの機密性を高めるために、カプセル化セキュリティペイロードヘッダ（**ESP**:Encapslating security payload header)という拡張ヘッダを使用します。暗号化のアルゴリズムは DES-CBC(Data Encryption Standard in Cypher Block Chaining mode) と 3DES-CBC が標準で指定されています。

2.8.3 セキュリティアソシエーション

IPv6 のセキュリティ機能を使用する際に、通信を行う相互間では鍵、暗号化アルゴリズムやそのアルゴリズムで使用するパラメータなどの情報を交換し、一致させる必要があります。その内容をセキュリティアソシエーション(以降 SA)といいます。SA では 2 つのカプセル化モードをサポートしています。

- **トランスポートモード**
  トランスポートモードは 2 点のエンドシステム間で定義され、全ての IP パケットのペイロード(データ)のみが暗号化されます。

- **トンネルモード**
  トンネルモードでは 2 点のセキュリティゲートウェイ間で定義され、IP パケットは別の IP パケットにカプセル化され、IP ヘッダを含めたパケット全体が暗号化されます。

SA はセキュリティ・アソシエーション・データベース(SAD)とセキュリティ・ポリシー・データベース(SPD)という 2 種類のデータベースで管理されています。NORTi TCP/IPv6 はトランスポートモードのみをサポートしています。

第 3 章 NORTi TCP/IPv6 の概要

この章では NORTi TCP/IPv6 の特長と概要を説明します。

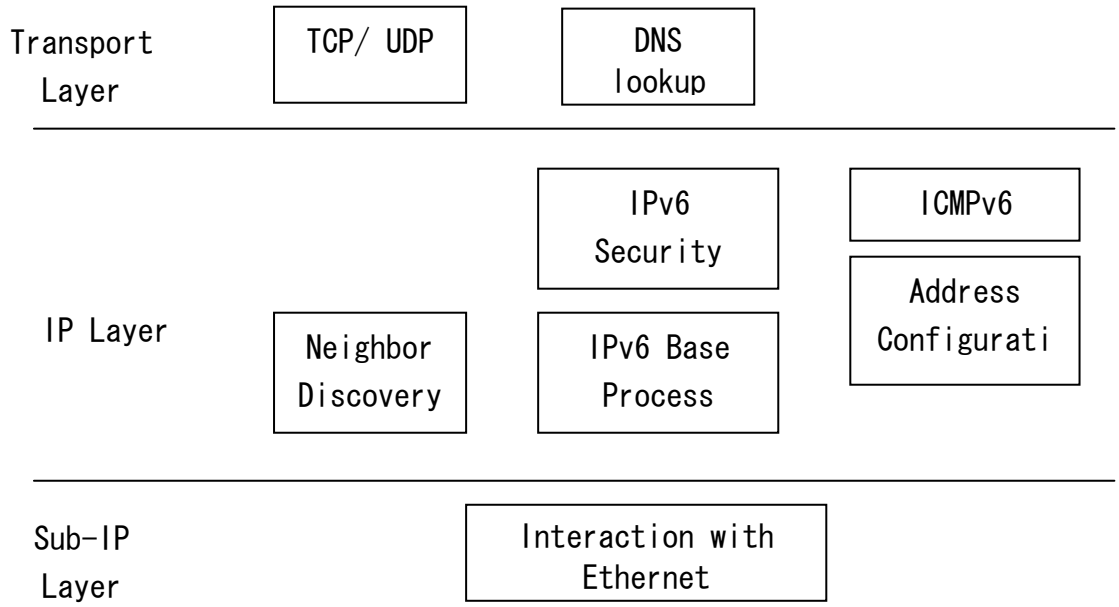
3.1 NORTi TCP/IPv6 のレイヤ構成

NORTi TCP/IPv6 は次のようなレイヤで構成されています。

<table>
<tr><td>アプリケーション層</td><td colspan="3">μITRON TCP/IP API</td></tr>
<tr><td>トランスポート層</td><td colspan="3">TCP/UDP</td></tr>
<tr><td rowspan="3">インターネット層</td><td colspan="2">IPv４</td><td>IPv６</td></tr>
<tr><td>ICMPv４</td><td>ARP</td><td>ICMPv６</td></tr>
<tr><td colspan="3">データリンク・モジュール</td></tr>
<tr><td>ネットワーク I/F 層</td><td colspan="3">デバイス・コントローラー</td></tr>
</table>

3.2 NORTi TCP/IPv6 のモジュール構成

NORTi TCP/IPv6 は次のようなモジュールで構成されています。



3.3 アドレス形式

NORTi TCP/IPv6 では次の 2 種類のアドレス形式をサポートしています。

・ユニキャストアドレス
（リンクローカル・アドレスおよび集約可能型グローバルユニキャスト・アドレス）
・マルチキャストアドレス（リンクローカル・マルチキャスト・アドレスのみ）

3.4 ICMPv6 と近隣探索

NORTi TCP/IPv6 では次の ICMPv6 メッセージをサポートしています。

- **エコー要求／エコー応答**
- **マルチキャストリスナー照会／報告／終了**

NORTi TCP/IPv6 では次の近隣探索用

- **近隣探索メッセージ／近隣通知メッセージ**
- **ルータ要請メッセージ／ルータ通知メッセージ**

3.5 自動設定

NORTi TCP/IPv6 ではステートレス方式のアドレス自動設定をサポートしています。MAC アドレスからリンクローカル・アドレスを作成し近隣探索メッセージを使用してアドレスの重複をチェックします。このチェックは tcp_ini を呼び出したときに実行されます。

3.6 互換性

NORTi TCP/IPv6 は IPv4 および IPv6 の 2 つの IP をサポートしています。
IP レイヤには IPv4、IPv6 の 2 つの IP プロトコルが実装されています。従来の IPv4 をベースに作成されたアプリケーションに加え IPv6 を使ったアプリケーションも作成できます。

・使用する資源
従来の送信、受信の 2 タスクに加え IPv6 の周期実行処理用のタスクが存在します。

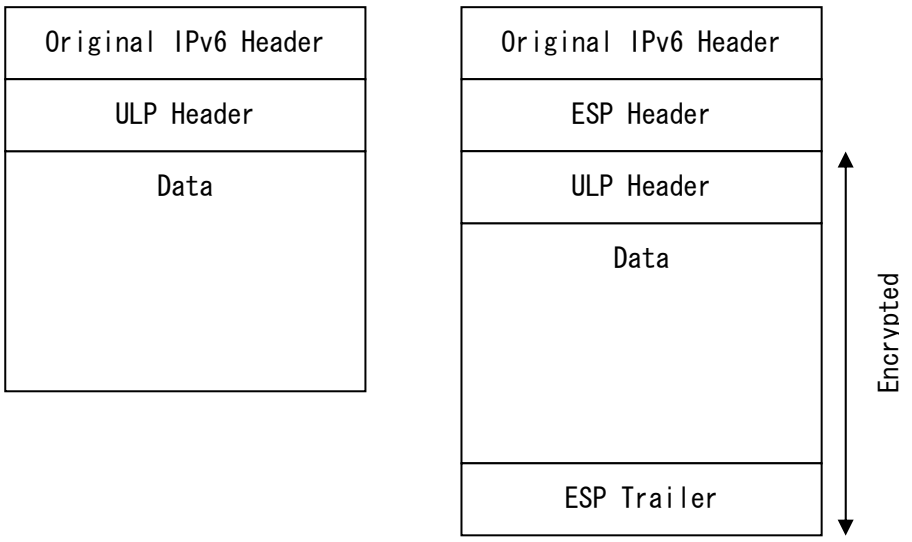
・API
従来の IPv4 用 API はアドレスを指定する部分以外に変更ありません。T_IPV4EP が IPv6 用に拡張され、T_IPEP となりました。

・Ethenet ドライバ
従来の Ethernet ドライバが使用できますが、マルチキャストの受信ができるように設定されている必要があります。もし ping6 などの応答が返らない場合、この設定を確認してください。

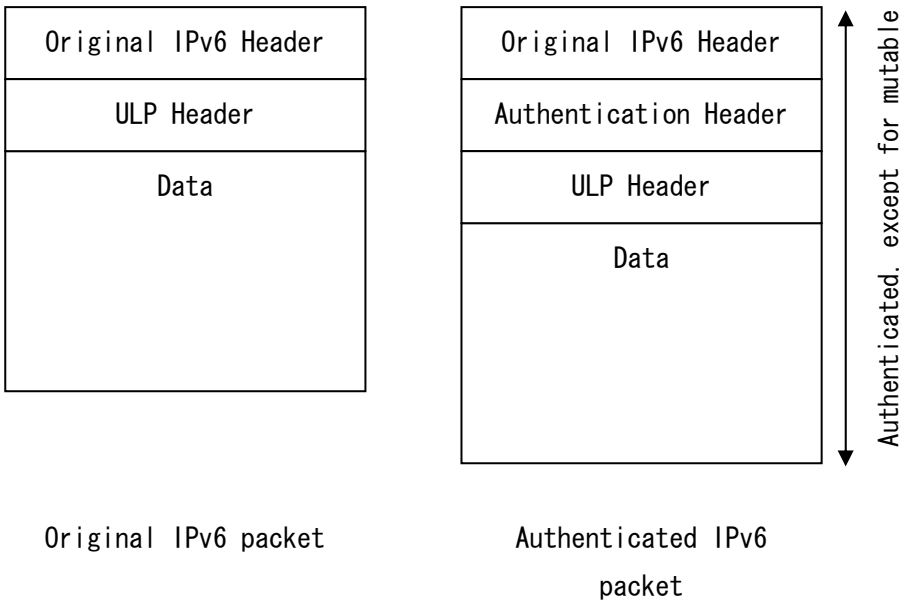
3.7 IPsec

NORTi TCP/IPv6 では IPsec を組み込んだ場合、認証拡張ヘッダ(以降 AH)とカプセル化セキュリティペイロードヘッダ(以降 ESP)の両方を使用します。カプセル化はトランスポートモードのみサポートしています。

・ESP を使ったパケット



・AH を使ったパケット



・暗号化アルゴリズム

AH では SHA1-96 と MD5-96 を、ESP では DES-CBC または 3DES-CBC を使用します。

・SA の管理

SA の管理は手動で行います。セキュリティ・アソシエーション・データベース(SAD)とセキュリティ・ポリシー・データベース(SPD)はファイル ipsec_db.c に定義されています。詳細は第 5 章を参照ください。

3.8 MLD（マルチキャストリスナー探索）

マルチキャストリスナー探索(MLD)はマルチキャストグループの管理で使用します。
NORTi TCP/IPv6 では次の API を使用可能です。

形 式	ER udp_set_opt(ID cepid, INT optname, VP optval, INT optlen) cepid　通信端点の ID optname オプション名 optval　オプションが含まれているバッファのポインタ optlen　オプションの長さ		
戻 値	E_OK　　正常終了 E_ID　　不正 ID E_NOEXS 通信端点が未生成 E_PAR　指定したアドレスがマルチキャストアドレスではない E_NOMEM メモリが不足		
オプション	optname	optval	内容
	IPV6_JOIN_GROUP	Multicast address	マルチキャストグループへの参加
	IPV6_LEAVE_GROUP	Multicast address	マルチキャストグループへの離脱
	MCAST_INCLUDE	Source List	ソースリストを INCLUDE モードに設定する
	MCAST_EXCLUDE	Source List	ソースリストを EXCLUDE モードに設定する
	ALLOW_SOURCES	Source List	ソースリストのフィルタ指定を許可にする
	BLOCK_SOURCES	Source List	ソースリストのフィルタ指定をブロックにする

3.9 アドレスの追加、削除、取得、状態通知コールバック

機 能	ユニキャストアドレスの追加
形 式	ER ipv6_add_addr(T_NIF *nif, UW *addr, UW prefix_len) nif　　ネットワーク I/F 制御ブロックへのポインタ(NULL の場合、 　　デフォルト I/F を使用) addr　　追加する IP アドレスが格納されている領域へのポインタ prefix_len 追加する IP アドレスのプレフィックス長
戻 値	E_OK　正常終了 E_OBJ　ネットワーク I/F が存在しない E_ILUSE アドレスがすでにネットワーク上に存在する、アドレスの格納領域が 　　不足している E_PAR　引数のいずれかが不正
解 説	ネットワーク I/F に対してユニキャストアドレスの追加を行います。1 つの I/F に対し 4 個までユニキャストアドレスを設定することができます。 addr は UW 型で 128 ビット分の領域を確保し追加したいアドレスを格納します。 デフォルト IF に対して FE80:0000:0000:0000:1034:56FF:FE78:9A9A(プレフィクス長は 64 ビット)を追加する場合、下記の例のようになります。 ER ercd; T_NIF *nif = getnif_default(); UW addr = { FE800000, 0, 103456FF, FE789A9A }; ercd = ipv6_add_addr(nif, addr, 64); なお、本 API は重複アドレス検出が完了するまでブロックされます。

機 能	ユニキャストアドレスの追加
形 式	ER ipv6_add_addr_by_ch(int ch, UW *addr, UW prefix_len) nif　　CH 番号 addr　　追加する IP アドレスが格納されている領域へのポインタ prefix_len 追加する IP アドレスのプレフィックス長
戻 値	E_OK　正常終了 E_OBJ　CH 番号に対応するネットワーク I/F が存在しない E_ILUSE アドレスがすでにネットワーク上に存在する、アドレスの格納領域が 　　不足している E_PAR　引数のいずれかが不正
解 説	CH番号で指定下ネットワーク I/Fに対してユニキャストアドレスの追加を行います。CH 番号で指定する点以外は ipv6_add_addr に準じます。

機 能	ユニキャストアドレスの追加
形 式	ER ipv6_add_addr_by_name(const char if_name, UW *addr, UW prefix_len) if_name　ネットワーク I/F 名 addr　追加する IP アドレスが格納されている領域へのポインタ prefix_len 追加する IP アドレスのプレフィックス長
戻 値	E_OK　正常終了 E_OBJ　ネットワーク I/F 名に対応するネットワーク I/F が存在しない E_ILUSE アドレスがすでにネットワーク上に存在する、アドレスの格納領域が不足している E_PAR　引数のいずれかが不正
解 説	ネットワーク I/F 名で指定したネットワーク I/F に対してユニキャストアドレスの追加を行います。ネットワーク I/F 名で指定する点以外は ipv6_add_addr に準じます。

機 能	ユニキャストアドレスの削除
形 式	ER ipv6_del_addr(T_NIF *nif, UW *addr) nif　ネットワーク I/F 制御ブロックへのポインタ(NULL の場合、デフォルト I/F を使用) addr　削除する IP アドレスのポインタ
戻 値	E_OK　正常終了 E_OBJ　ネットワーク I/F が存在しない E_PAR　引数のいずれかが不正 E_NOEXS 指定したアドレスは存在しない
解 説	ネットワーク I/F に対してユニキャストアドレスの削除を行います。 addr は UW 型で 128 ビット分の領域を確保しアドレスを格納します。 デフォルト IF から FE80:0000:0000:0000:1034:56FF:FE78:9A9A を削除する場合、下記の例のようになります。 ER ercd; T_NIF *nif = getnif_default(); UW addr = { FE800000, 0, 103456FF, FE789A9A }; ercd = ipv6_del_addr(nif, addr);

機 能	ユニキャストアドレスの削除
形 式	ER ipv6_del_addr_by_ch(int ch, UW *addr) ch　CH 番号 addr　削除する IP アドレスのポインタ
戻 値	E_OK　正常終了

	E_OBJ　CH 番号に対応するネットワーク I/F が存在しない E_PAR　引数のいずれかが不正 E_NOEXS 指定したアドレスは存在しない
解 説	CH番号で指定したネットワークI/Fに対してユニキャストアドレスの削除を行います。CH 番号で指定する点以外は ipv6_del_addr に準じます。

機 能	ユニキャストアドレスの削除
形 式	ER ipv6_del_addr_by_name(const char *if_name, UW *addr) if_name ネットワーク I/F 名 addr　削除する IP アドレスのポインタ
戻 値	E_OK　正常終了 E_OBJ　ネットワーク I/F 名に対応するネットワーク I/F が存在しない E_PAR　引数のいずれかが不正 E_NOEXS 指定したアドレスは存在しない
解 説	ネットワーク I/F 名で指定したネットワーク I/F に対してユニキャストアドレスの削除を行います。ネットワーク I/F 名で指定する点以外は ipv6_del_addr に準じます。

機 能	アドレス情報の取得
形 式	ER ipv6_get_addr(T_NIF *nif, T_IPV6_ADDR *buf, UW buf_size, 　　UW *found_num, UH addr_type_flag) nif　ネットワーク I/F 制御ブロックへのポインタ(NULL の場合、デフォルト I/F を使用) buf　アドレス情報を格納する領域へのポインタ buf_size　アドレス情報を格納する領域の個数 found_num　指定したネットワーク I/F に割り当てられているアドレス数を格納するための領域へのポインタ addr_type_flag 取得するアドレスのタイプ typedef struct t_ipv6_addr { 　UW addr[4];　IPv6 アドレス 　UH scope;　IPv6 アドレスのスコープ 　UW prefix_len; IPv6 アドレスのプレフィクス長 　B　status;　IPv6 アドレスの状態 } T_IPV6_ADDR;
戻 値	E_OK　正常終了 E_OBJ　ネットワーク I/F が存在しない

	E_PAR　引数のいずれかが不正
解 説	ネットワーク I/F のアドレス情報を取得します。 アドレスのタイプとしては下記の 2 つが指定可能です。 ADDR_UNICAST　指定したネットワーク I/F に設定されているユニキャストアドレスの情報を取得する ADDR_MULTICAST 指定したネットワーク I/F に設定されているマルチキャストアドレスの情報を取得する T_IPV6_ADDR に格納される情報は下記のようになります。 addr　IPv6 アドレス FE80:0000:0000:0000:1034:56FF:FE78:9A9A の場合、 addr[0] FE800000 addr[1] 0 addr[2] 103456FF addr[3] FE789A9A という形で格納されます。 scope　IPv6 アドレスのスコープ 下記の 4 種類のいずれかが格納されます。 ADDR_IF_LOCAL　0x0100 インターフェースローカル ADDR_LINK_LOCAL 0x0200 リンクローカル ADDR_SITE_LOCAL 0x0800 サイトローカル ADDR_GLOBAL　0x2000 グローバル prefix_len IPv6 アドレスのプレフィクス長 status　IPv6 アドレスの状態 下記の 4 種類のいずれかが格納されます。 ADDR_TENTATIVE　(8) 暫定アドレス(重複アドレス検出完了前) ADDR_PREFERRED　(1) 有効アドレス(重複アドレス検出完了後) ADDR_DEPRECATED (2) 廃止予定アドレス(valid lifetime 満了間近) ADDR_PERMANENT　(7) 永続アドレス(ipv6_add_addr などで指定したアドレス) デフォルト I/F からユニキャストアドレスの情報を取得する場合、下記の例のようになります。 ER ercd; T_NIF *nif = getnif_default(); T_IPV6_ADDR buf[4]; UW found_num ercd = ipv6_get_addr(nif, (T_IPV6_ADDR *)buf, 4, &found_num, ADDR_UNICAST);

	1 つのネットワーク I/F に割り当て可能なユニキャストアドレス、マルチキャストアドレスの数は最大 4 であるため buf_size は 4 以下となります。 有効アドレス、永続アドレスが通信に利用可能です。廃止予定アドレスは利用可能ですが、valid lifetime 満了までにルータ広告を受信しなかった場合、削除されます。暫定アドレスは重複アドレス検出が完了していないため通信には利用できません。

機 能	アドレス情報の取得
形 式	ER ipv6_get_addr_by_ch(int ch, T_IPV6_ADDR *buf, UW buf_size, UW *found_num, UH addr_type_flag) ch CH 番号 buf アドレス情報を格納する領域へのポインタ buf_size アドレス情報を格納する領域の個数 found_num 指定したネットワーク I/F に割り当てられているアドレス数を格納するための領域へのポインタ addr_type_flag 取得するアドレスのタイプ typedef struct t_ipv6_addr { UW addr[4]; IPv6 アドレス UH scope; IPv6 アドレスのスコープ UW prefix_len; IPv6 アドレスのプレフィクス長 B status; IPv6 アドレスの状態 } T_IPV6_ADDR;
戻 値	E_OK 正常終了 E_OBJ ネットワーク I/F が存在しない E_PAR 引数のいずれかが不正
解 説	CH 番号で指定したネットワーク I/F のアドレス情報を取得します。CH 番号で指定する点以外は ipv6_get_addr に準じます。

機 能	状態通知コールバック関数の登録/削除
形 式	ER ipv6_def_addr_chg_cbk(T_NIF *nif, IPV6_ADDR_CHG_CALLBACK func) nif ネットワーク I/F 制御ブロックへのポインタ(NULL の場合、デフォルト I/F を使用) func コールバック関数のポインタ。NULL を指定した場合はコールバック関数を削除します。コールバック関数に関しては、後述の“コールバック”を参照して下さい。

戻 値	E_OK　正常終了 E_OBJ　ネットワーク I/F が存在しない E_PAR　引数のいずれかが不正
解 説	指定したネットワーク I/F のアドレス状態が変化した際にコールバックされる関数の登録/削除を行います。 なお、コールバック関数に関する定義は下記のようになっています。 typedef void (*IPV6_ADDR_CHG_CALLBACK)(int ch, const char *if_name);

機 能	状態通知コールバック関数の登録/削除
形 式	ER ipv6_def_addr_chg_cbk_by_ch(int ch, IPV6_ADDR_CHG_CALLBACK func) ch　CH 番号 func　コールバック関数のポインタ。NULL を指定した場合はコールバック関数を削除します。コールバック関数に関しては、後述の“コールバック”を参照して下さい。
戻 値	E_OK　正常終了 E_OBJ　ネットワーク I/F が存在しない E_PAR　引数のいずれかが不正
解 説	CH番号で指定したネットワーク I/Fのアドレス状態が変化した際にコールバックされる関数の登録/削除を行います。CH 番号で指定する点以外は ipv6_def_addr_chg_cbk に準じます。

機 能	状態通知コールバック関数の登録/削除
形 式	ER ipv6_def_addr_chg_cbk_by_name(const char *if_name, IPV6_ADDR_CHG_CALLBACK func) if_name ネットワーク I/F 名 func　コールバック関数のポインタ。NULL を指定した場合はコールバック関数を削除します。コールバック関数に関しては、後述の“コールバック”を参照して下さい。
戻 値	E_OK　正常終了 E_OBJ　ネットワーク I/F が存在しない E_PAR　引数のいずれかが不正
解 説	ネットワーク I/F 名で指定したネットワーク I/F のアドレス状態が変化した際にコールバックされる関数の登録/削除を行います。ネットワーク I/F 名で指定する点以外は ipv6_def_addr_chg_cbk に準じます。

機 能	コールバック
形 式	void (*callback)(int ch, const char *if_name) ch　　アドレス状態が変化したネットワーク I/F の CH 番号 if_name　アドレス状態が変化したネットワーク I/F のネットワーク I/F 名
戻 値	なし
解 説	ネットワーク I/F のアドレス状態が変化した際にコールバックされる関数です。関数名は任意です。 暫定アドレスが有効アドレス/永続アドレスとなった場合、valid lifetime の満了によりアドレスが削除された場合などにコールバックされます。 本コールバック本関数は TCP/IPv6 プロトコルスタックのコンテキストで実行されます。slp_tsk などの待ち状態への遷移が発生する可能性のあるシステムコールは発行しないでください。また、処理時間はできるだけ短くなるようにしてください。 待ち状態への遷移につきましては、NORTi Version 4 ユーザーズガイド カーネル編“1.2 タスクの状態”をご参照ください。 なお、本コールバック関数に関しては nonet.h で下記のように型定義されています。 typedef void (*IPV6_ADDR_CHG_CALLBACK)(int ch, const char *if_name);

3.10 制限事項

NORTi TCP/IPv6 は IPv6 の必要最低限の仕様を実装しています。以下は現バージョンで NORTi TCP/IPv6 に実装されていない機能です。

- IPv6 を使った PPP のサポート
- 経路 MTU 探索
- IPv6 ジャンボグラム
- ステートフル・アドレス自動設定
- DHCPv6
- IPsec で使用する SA（Security Association：セキュリティ・アソシエーション）と鍵管理の自動化
- IPsec のトンネルモード
- IPv4 パケットへのトンネル

第 4 章 コンフィグレーション

4.1 ライブラリ

LIB フォルダに収録されている全てのライブラリは IPv4 と IPv6 のデュアルスタック構造になっています。これらのファイルは DUAL_STK マクロを定義してビルドされています。DUAL_TSK マクロなしでビルドした場合は IPv4 のみのライブラリが作成されます。IPv6 のみのライブラリは作成できません。

4.2 定義

コンフィグレーションヘッダ nonetc.h を#include する前に、次の様な定数を記述することにより、コンフィグレーションが行えます。()はデフォルトの値で指定がない場合に使用されます。

#define NEIGH_CACHE_CNT	(8)	近隣キャッシュの登録可能な数
#define PREFIX_LIST_CNT	(8)	プレフィックスの登録可能な数
#define DFLT_ROUTER_CNT	(2)	ルーターキャッシュの登録可能な数
#define DST_CACHE_CNT	(8)	Next-Hop のアドレスカウンタ
#define DAD_TMO	(1000)	重複アドレス検出のタイムアウト(ミリ秒)
#define IP6F_REASM_TMO	(2)	IPv6 フラグメントパケット再構築のタイムアウト（秒）
#define IP6HASH_QSZ	(2)	フラグメントパケット再構築用ハッシュキューのサイズ
#define MAX_IPSEC_ENTRIES	(15)	IPsec データベースの最大エントリ数

※ 注意
ソフトウェア暗号化ライブラリを使って IPsec を使用する場合、IP 送受信タスクで使用するスタックサイズが不足します。そのため nonetc.h を include する前に必ず IPSEC マクロを定義してください。スタックサイズを内部で自動的に調整します。
例)
#define IPSEC
#include “nonetc.h”

第 5 章 IPsec

IPsec を使用するためには、IPsec 付のライブラリをリンクする必要があります。同時に、アプリケーションはセキュリティ・データベースファイルと暗復号/ハッシュアルゴリズムの関数が収録されたファイルをリンクしてください。

5.1 IPsecデータベース

IPsecは、セキュリティ・アソシエーション(SA)の概念に基づき実装されています。NORTi TCP/IPv6のIPsecでは、SAは手動で管理する必要があります。SAはあるトラフィックのセキュリティを確保する単一方向のコネクションで、セキュリティプロトコル、モード、エンドポイントのアドレス、オプショナルサービスなどを選択します。SAではセキュリティ・ポリシー・データベース(SPD)とセキュリティ・アソシエーション・データベース(SAD)を使用します。IPパケットの通信を行うときセキュリティ・ポリシーが格納されているSPDを参照します。SADにはSAに関連するパラメータが含まれています。

5.1.1 セキュリティ・ポリシー・データベース (SPD)

セキュリティ・ポリシー・データベース(SPD)は ipsec_db.c の T_SPD_ENTRY spd_ref[]で定義されています。ここでは以下の項目を設定しています。

項目	内容
index	データベースレコードのインデックス番号
rmt_addr	リモートホストの IPv6 アドレス
local_addr	ローカルホストの IPv6 アドレス
transport	トランスポートレイヤプロトコル
rmt_port	リモートホストアプリケーションのポート番号
local_port	ローカルホストアプリケーションのポート番号
action	アプリケーションの動作ポリシー
direction	トラフィックの方向
sad_ptr	関連する SA エントリの先頭インデックス番号

index - 1 から始まる SPD レコードのユニークなインデックス番号を指定します。この値はエントリ毎に必ずインクリメントする必要があります。

rmt_addr & local_addr - これらのフィールドには IPsec で通信を行うそれぞれのホストのアドレスが入ります。全てのホストとの通信を行う場合は IPV6_ADDRANY を定義します。

transport - IPsec を使用するトランスポート層のプロトコル(PROT_TCP, PROT_UDP, PROT_ICMPV6)を定義します。 全てのプロトコルを使用する場合は‘*’を指定します。

rmt_port & **local_port** - リモートホスト/ローカルホストのポート番号を指定します。全てのポートと通信を行う場合は ‘*’を指定します。

action - 以下のアプリケーションの動作ポリシーを指定します。
IPSEC_APPLY - IPsec を適用(apply IPsec)
IPSEC_DISCARD - 破棄(discard)
IPSEC_BYPASS - IPsec を適用しない(bypass IPsec)
direction - トラフィックの方向を設定します。値は TRAFFIC_BIDIR 固定です。
sad_ptr - 関連する SA エントリの先頭インデックス番号を指定します。 action に IPSEC_APPLY 以外を指定した場合は使用されません。

※SPD に定義されていないノードとは IPsec で通信を行うことができません。IPsec で通信を行うノードの情報を全て登録する必要があります。もし他のノードとセキュリティを使用しない通信を行う場合は、SPD の最後に以下の定義を追加する必要があります。
rmt-addr と local_addr を IPV6_ADDRANY
transport, rmt_port 及び local_port を '*'
action を IPSEC_BYPASS

5.1.2 セキュリティ・アソシエーション・データベース (SAD)

セキュリティ・アソシエーション・データベース(SAD)は ipsec_db.c の T_SAD_ENTRY sad_ref[]で定義されています。ここでは以下の項目を設定しています。

項目	内容
index	データベースレコードのインデックス番号
seqcntr	AH または ESP ヘッダで使用するシーケンス番号
spi	セキュリティ・パラメータ・インデックス
dst_addr	送信先 IPv6 アドレス
proto	使用するセキュリティプロトコル(AH または ESP)
mode	セキュリティ・モード
auth_alg	認証アルゴリズム
antireply	リプレイ防止機能フラグ
auth_key	認証で使用する秘密鍵
auth_key_len	認証で使用する秘密鍵の長さ
encr_alg	暗号アルゴリズム
esp_ah	ESP で認証データフィールドを含めるか否かのフラグ
encr_key	暗号化に使用する秘密鍵
encr_key_len	暗号化に使用する秘密鍵の長さ
iv	ESP で使用する初期化ベクタ
direction	セキュリティ・アソシエーションの方向

bundle_ptr	次の SA エントリへのインデックス
spd_ptr	関連する SP エントリの先頭インデックス番号

index - 1 から始まるレコードのユニークなインデックス番号を指定します。この値はエントリ毎に必ずインクリメントする必要があります。

seqcntr - AH または ESP ヘッダで使用するシーケンス番号です。32 ビットのランダムな値を指定します。

spi - セキュリティ・パラメータ・インデックスを指定します。通信を行うリモートホストと同じ値を設定する必要があります。

dst_addr - 送信先 IPv6 アドレス。データ送信時はリモートアドレス。データ受信時はローカルアドレス。

proto - セキュリティプロトコルを指定します。AH プロトコルを使用する場合は PROT_AH を ESP プロトコルを使用する場合は PROT_ESP を指定します。

mode - セキュリティ・モードを指定します。この値は MODE_TRANSPORT 固定です。

auth_alg -認証アルゴリズムを指定します。
AUTH_NULL(NULL),AUTH_HMAC_SHA1_96(SHA1)または AUTH_HMAC_MD5_96(MD5)が設定できます。

antireply - リプレイ防止機能を設定します。この値は‘0’固定です。

auth_key and auth_key_len - 認証で使用する秘密鍵と鍵の長さを指定します。20byte の鍵を使用する HMAC-SHA1-96(SHA1)か 16bytes の鍵を使用する HMAC-MD5-96(MD5)が設定できます。

encr_alg - 暗号アルゴリズムを指定します。ESP_NULL(NULL), ESP_DES_CBC(DES)または ESP_3DES_CBC(3DES)が設定できます。

esp_ah - ESP で認証データフィールドを含めるか否かのフラグ。’0’の場合は認証データを含みません。’1’の場合は認証データを含みます。

encr_key and encr_key_len - 暗号で使用する秘密鍵と鍵の長さを指定します。鍵は 8byte の DES-CBC か 24bytes の 3DES-CBC を設定できます。

iv - ESP で使用する初期化ベクタを設定します。ここには 8byte のランダムな値を設定します。

direction - This member should be specified as TRAFFIC_INBOUND for incoming packet and TRAFFIC_OUTBOUND for outgoing packet.

bundle_ptr - 次の SA エントリへのインデックス。この値は‘0’を設定します。

spd_ptr - 関連する SPD エントリの先頭インデックス番号を指定します。

※ SA は単一方向のコネクションであるため、セキュア通信を行うホストは、送信側と受信側の 2 つの SA が SAD に登録されていなければいけません。

※ SPD のエントリで”action”が IPSEC_APPLY 以外が選択されていた場合は、SAD への登録は不要です。

5.2 暗復号/ハッシュアルゴリズム

NORTi TCP/IPv6 の IPsec はハッシュアルゴリズムとして MD5 および SHA1 を、暗復号アルゴリズムは DES-CBC および 3DES-CBC を標準でサポートしています。これらの関数は暗号化ライブラリ cryptxxxx.lib に収録されています。
これらの処理をソフトウェアで行うと通信の速度が数十分の 1 に低下します。IPsec を使用した高速な通信を行うには暗復号やハッシュの処理はハードウェアアクセラレータを使用することをお勧めします。以下の I/F 関数は IPsec 内部から呼び出されます。これらの関数を作成することで、異なった仕様のハードウェアアクセラレータを使用することが出来ます。

機 能	MD5 のハッシュ計算
形 式	ER ipsec_hmac_md5(UB* in_data, UW in_size, UB* secret_key, UW key_len, UB *output, UW req_len); in_data 入力データが格納されているバッファポインタ in_size 入力データの長さ secret_key 秘密キーが格納されているバッファのポインタ key_len 秘密キーの長さ output 計算されたデータを格納するバッファのポインタ req_len 格納するバッファのサイズ
戻 値	格納されたハッシュの長さ
解 説	MD5 のハッシュ計算を行います。

機 能	SHA1 のハッシュ計算
形 式	ER ipsec_hmac_sha1(UB* in_data, UW in_size, UB* secret_key, UW key_len, UB *output, UW req_len); in_data 入力データが格納されているバッファポインタ in_size 入力データの長さ secret_key 秘密キーが格納されているバッファのポインタ key_len 秘密キーの長さ output 計算されたデータを格納するバッファのポインタ req_len 格納するバッファのサイズ
戻 値	格納されたハッシュの長さ
解 説	SHA1 のハッシュ計算を行います。

機 能	DES の初期化	
形 式	ER des_init(T_DES_CTX *ctx, UB *iv, UB *key, UW keylen);	
	ctx	初期化される DES コンテクストのポインタ
	iv	暗号処理用初期化ベクタへのポインタ
	key	64bit の秘密キーが格納されているバッファのポインタ
	keylen	有効なキーの長さ
戻 値	E_OK	正常終了
	E_PAR	NULL ポインタ入力、キーの長さが不正
解 説	DES コンテクストが初期化されます。	

機 能	DES による暗号化	
形 式	ER des_cbc_encr(T_DES_CTX *ctx, UB *pbuf, UB *cbuf, UW len);	
	ctx	DES コンテクストのポインタ
	pbuf	入力データが格納されているバッファポインタ
	cbuf	暗号化されたデータが格納されるバッファのポインタ
	len	入力データの長さ
戻 値	E_OK	正常終了
	E_PAR	NULL ポインタ入力、入力データの長さが 8 の倍数でありません
解 説	DES でデータを暗号化します。	

機 能	DES による復号化	
形 式	ER des_cbc_decr(T_DES_CTX *ctx, UB *cbuf, UB *pbuf, UW len)	
	ctx	DES コンテクストのポインタ
	cbuf	入力データが格納されているバッファポインタ
	pbuf	暗号化されたデータが格納されるバッファのポインタ
	len	入力データの長さ
戻 値	E_OK	正常終了
	E_PAR	NULL ポインタ入力、入力データの長さが 8 の倍数でありません
解 説	DES でデータを複合化します。	

機 能	3DES の初期化	
形 式	ER tdes_init(T_TDES_CTX *ctx, UB *iv, UB *key, UW keylen);	
	ctx	初期化される 3DES コンテクストのポインタ
	iv	暗号処理用初期化ベクタへのポインタ
	key	192bit の秘密キーが格納されているバッファのポインタ
	keylen	有効なキーの長さ
戻 値	E_OK	正常終了
	E_PAR	NULL ポインタ入力、キーの長さが不正
解 説	3DES コンテクストが初期化されます。	

機 能	3DES による暗号化	
形 式	ER tdes_cbc_encr(T_TDES_CTX *ctx, UB *pbuf, UB *cbuf, UW len);	
	ctx	3DES コンテクストのポインタ
	pbuf	入力データが格納されているバッファポインタ
	cbuf	暗号化されたデータが格納されるバッファのポインタ
	len	入力データの長さ
戻 値	E_OK	正常終了
	E_PAR	NULL ポインタ入力、入力データの長さが 8 の倍数でありません
解 説	3DES でデータを暗号化します。	

機 能	3DES による復号化	
形 式	ER tdes_cbc_decr(T_TDES_CTX *ctx, UB *cbuf, UB *pbuf, UW len)	
	ctx	3DES コンテクストのポインタ
	cbuf	入力データが格納されているバッファポインタ
	pbuf	暗号化されたデータが格納されるバッファのポインタ
	len	入力データの長さ
戻 値	E_OK	正常終了
	E_PAR	NULL ポインタ入力、入力データの長さが 8 の倍数でありません
解 説	3DES でデータを複合化します。	

第 6 章 IPv4 スタックからの変更点

6.1 T_IPEP

通信端点や受付口の生成などで使用されるアドレスを指定するために使用する構造体 T_IPV4EP は IPv6 のアドレスを設定できるように拡張され T_IPEP となりました。

```
typedef struct t_ipep
{
  UW            ipaddr;         /* IP4 アドレス*/
  UH            portno;         /* ポート番号 */
  UW            ip6addr[4];     /* IP6 アドレス*/
  BOOL          type;           /* 使用するアドレスタイプ */
} T_IPEP;
```

type には IPv4 のアドレスを指定する際には、**IPV4_ADDR** を IPv6 のアドレスを指定する際には、**IPV6_ADDR** を指定します。
T_IPV4EP 構造体を使用している T_TCP_CREP および T_UDP_CCEP は T_IPEP に変更されました。

6.2 ローカル IP アドレスの指定

自分側の IP アドレスを設定する関数では、従来サポートしていた IPV4_ADDRANY と同じように **IPV6_ADDRANY** が使用できます。IP アドレスに IPV6_ADDRANY を指定した場合、自動生成されたリンクローカルアドレスがプロトコルスタックで設定されます。通常は IPV6_ADDRANY を使用してください。

6.3 バイトオーダー変換

IPv6 で使用する 128-bit の IPv6 アドレスのバイトオーダー変換を行うために次の関数が追加されました。

ntohl6

【機能】ネットワークオーダーからホストオーダーへの変換を行う

【形式】UW *ntohl6(UW *hl, UW *nl);

hl ホストオーダーに変換された IPv6 アドレスへのポインタ

nl ネットワークバイトオーダーの IPv6 アドレスへのポインタ

【戻値】 ホストオーダーに変換された IPv6 アドレスへのポインタ

htonl6

【機能】ホストオーダーからネットワークオーダーへの変換を行う

【形式】UW *htohl6(UW *nl, UW *hl);

nl　　ネットワークバイトオーダーの IPv6 アドレスへのポインタ

hl　　ホストオーダーに変換された IPv6 アドレスへのポインタ

【戻値】 ネットワークオーダーに変換された IPv6 アドレスへのポインタ

6.4 ICMPv6

ICMPv6 用に関数が追加されています。

icmpv6_def_cbk でユーザー側で作成したコールバック関数を登録することによりECHO 以外のパケットの受信を行うことができます。送信にはicmpv6_snd_dat を使用します。

icmpv6_def_cbk

【機能】ICMP 受信コールバック関数の登録

【形式】ER icmpv6_def_cbk(T_ICMP6_CB *b, ICMP6_CALLBACK callback)

b　　ICMPv6制御ブロック

callback　　コールバック関数

【戻値】E_OK　正常終了

負の値　異常終了

【解説】エコー要求、近隣要請、近隣広告、リダイレクト、MLDクエリー、MLDレポート、ルータ広告以外のICMPv6 パケットを受信した時に呼び出される関数を登録できます。

ICMPv6制御ブロックはチェーン構造で管理され、複数のコールバック関数を登録できます。ICMP 制御ブロック領域は、ユーザーが変数としてこの領域を確保して渡すだけで、初期設定の必要はありません。

コールバック

【機能】ICMP の受信コールバック

【形式】VP *callback(T_ICMP6_CB *b, T_IP6 *ip, T_ICMP6_MSG *icmp, INT len);

b　　ICMPv6制御ブロックへのポインタ(通常、使用しない)

ip　　IPv6 パケットへのポインタ

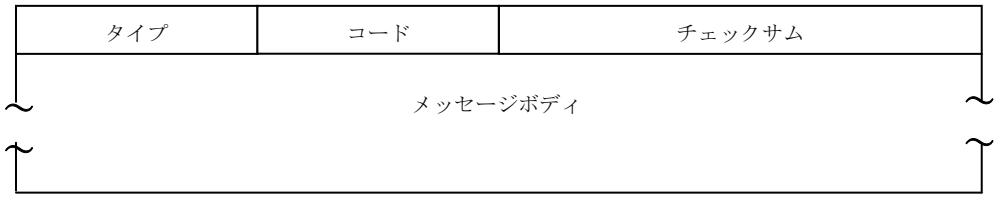
icmp6　ICMPv6 メッセージへのポインタ

len　　パケット長

```
typedef struct t_icmp6_msg {
        T_ICMPV6_HEADER icmpv6;
        UB      data[IP6_HEADER_SZ+8];/* Option Data (Variable Size) */
} T_ICMP6_MSG;
```

【解説】ICMPv6 受信コールバック関数で指定する関数です。関数名は任意です。エコー

要求、近隣要請、近隣広告、リダイレクト、MLDクエリー、MLDレポート、ルータ広告以外のICMPv6パケットを受信するとこの関数が呼び出されます。
ICMPv6 のメッセージフォーマットは次のようになります。先頭 32bit 以降はメッセージによってデータが異なります。



icmpv6_snd_dat

【機能】ICMP パケット送信

【形式】ER icmpv6_snd_dat(UW *src6addr, UW *dst6addr, T_ICMP6_HEADER *icmpv6, VP data, INTlen);

src6addr	自局のIPv6アドレス
dst6addr	相手のIPv6アドレス
icmpv6	送信するICMPv6ヘッダへのポインタ
data	送信するICMPv6データへのポインタ
len	データ長

【戻値】E_OK　正常終了
負の値　異常終了

【解説】任意のICMPv6パケットを送信することができます。上述のコールバック関数の内部からも呼び出すことが可能です。ping6_command に使用例があります。

第 7 章 IPv6 のサンプルアプリケーション

NORTi TCP/IPv6ではIPv6を評価するために次のサンプルアプリケーションを収録しています。

Ping6 (NORTi¥NETSMP¥SRC¥nonping.c)
IPv6版のPingコマンドです。ICMPv6のエコー機能を使用しています。

IPv6版 FTPサーバ／クライアント(NORTi¥NETSMP¥SRC¥nonftpd.c/nonftpc.c)
IPv6版のFTPサーバ／クライアントです。

IPv6版 Telnetサーバ／クライアント(NORTi¥NETSMP¥SRC¥nonteld.c/nontelc.c)
IPv6版のTelnetサーバです。IPv6に対応したTelnetクライアント/サーバへ接続可能です。

IPv6版 ECHOサーバ／クライアント(NORTi¥NETSMP¥SRC¥nonechc.c/nonecho.c)
TCPとUDPを使用したエコープログラム(ポート番号7)です。

IPv6版 DNS リゾルバ(NORTi¥NETSMP¥SRC¥nonedns.c)
ネームからIPアドレスに変換を行うための関数です。

NORTi TCP/IPv6 ユーザーズガイド

株式会社ミスポ	http://www.mispo.co.jp/
一般的なお問い合わせ	sales@mispo.co.jp
技術サポートご依頼	norti@mispo.co.jp