第 X 部

Linux における IPv6/IPsec スタックの研究開発

第 10 部
Linux における IPv6/IPsec スタックの研究開発

第 1 章　USAGI プロジェクトの概要と目的

USAGI プロジェクトは、Linux を中心としてより良い IPv6 環境を提供することを目的に、WIDE プロジェクトを中心に、有志によって構成されたプロジェクトである。KAME プロジェクト、TAHI プロジェクトと連携をとりながら、Linux の IPv6 スタックや、IPv6 に関するライブラリ、アプリケーションを改良し、より良いコードを提供している。また、Linux Symposium（`http://www.linuxsymposium.org`）など、機会を捉えての啓蒙活動にも力を入れている。

USAGI プロジェクトの成果物は 2 週に 1 度の snapshot と、年数回を目処とした stable release として公開している。これと並行して、メインラインカーネルに対して、USAGI 成果物の提供・反映を図っており、多くの改善点と機能がすでに採り入れられている。USAGI プロジェクトは、今後も成果のグローバルな展開を続けていく。

なお、プロジェクトに関する最新の詳しい情報については `http://www.linux-ipv6.org` にて公開している。

第 2 章　2004 年の主な活動

2.1 Mobile IPv6

2.1.1 背景

USAGI プロジェクトでは、ヘルシンキ工科大学（HUT）で開発された Mobile IPv6 プロトコルスタックの実装を元に、IPsec との協調処理などの USAGI プロジェクト独自の拡張機能を Linux 2.4 系カーネル上で実装してきた。

2003 年 3 月より開発対象を 2.5 系カーネルに移し、同年 10 月より HUT と本格的な共同作業を開始した。さらに本年度は、メインラインカーネルが 2.6 系へ更新されたことに追従する形で 2.6 系カーネルを開発対象としている。

一方で、Mobile IPv6 プロトコル仕様は 2004 年 6 月に RFC 3775 及び RFC 3776 として発行された。

USAGI プロジェクトは、RFC 3775 と RFC 3776 を実現するスタックが Linux の機能として取り入れられることを目標としている。

2.1.2 2004 年度のステータス

主な活動

- 2004/10 対向ノード（Correspondent Node; CN）機能は TAHI conformance test（ct-mipv6-cn-2.0b3）の全項目をクリア
- 2004/11 機能制限版のバージョンを MIPL-2.0-rc1 としてリリース（HUT）

実現された機能

MIPL-2.0-rc1 では、以下が実現されている。

- RFC 3775 のうち、Mobile Prefix discovery 以外の全ての機能
- RFC 3776 のうち、モバイルノード（Mobile Node; MN）とホームエージェント（Home Agent; HA）間の位置登録更新要求（Binding Update）と応答（Binding Acknowledgement）メッセージの IPsec による保護機能（手動鍵設定のみ）

2.1.3 Mobile IPv6 プロトコルスタックの設計

Mobile IPv6 の機能がメインラインカーネルに取りこまれるためには、カーネルメンテナから出来るだけ軽微なカーネル修正で済むことが好ましいとアドバイスされている。

そこで USAGI プロジェクトでは、カーネル内の既存の枠組みを極力流用し、カーネル内で持つ必要のないデータ構造や処理はカーネルと分離してユーザランドで実装するデザインを採用した。

カーネルの設計

カーネルの機能ブロック図を図 2.1 に示す。

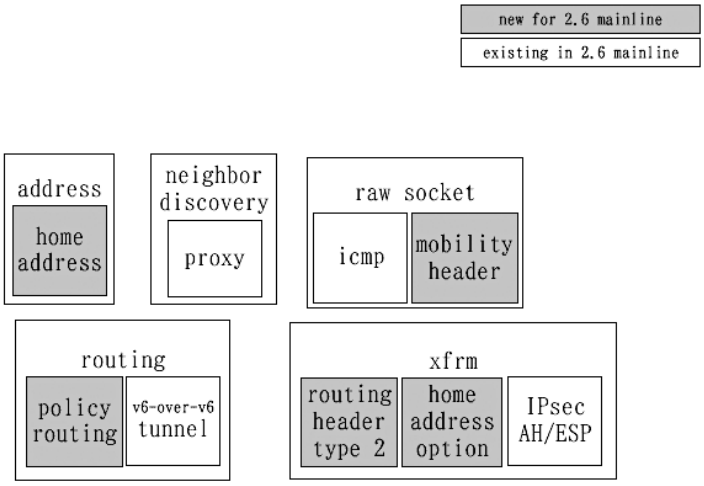

図 2.1.　カーネルの機能

- アドレス管理部の拡張

　MN 用ホームアドレスの識別情報と関連する処理を追加した。

- 経路制御部の拡張

　経路表を多段構成にし、ユーザランドから指定されるポリシによって経路選択できる機能を追加した。この機能は Policy Routing と呼ばれ、これによってたとえば送信アドレスを条件として経路選択が可能になる。

- 近隣探索プロキシ機能（Proxy Neighbor Discovery）の修正

　HA では、MN のホームアドレス宛パケットを捕捉するために Proxy Neighbor Discovery 機能を使う。この機能自体はすでにメインラインカーネルにあるが、多少の修正を入れている。

- RAW ソケットの拡張

　Mobile IPv6 のシグナリングをサポートするために、モビリティヘッダの受信処理を RAW ソケットのパケット処理受信に追加した。ただし、ここではヘッダの最低限の解析処理しか行わず、メッセージはすぐさまユーザランドへ渡される。

- XFRM の拡張

　XFRM とは、“transform” と読み、2.5 および 2.6 系カーネルで採用されている、いわばパケット変換機構である。経路制御部や、ネットフィルタなどのフィルタ部とは別定義されており、IPsec の内部構造として使われている。

　XFRM を、位置情報である Binding Cache 及び Binding Update List のサブセットを管理するように拡張し、終点オプションヘッダ用ホームアドレスオプションと経路ヘッダ（type 2）処理を追加して経路最適化機能をサポートした。また、位置情報がないアドレスを含むパケットを検知して、ユーザランドへの通知を行うインタフェースを追加した。

　例えば、CN が受信したパケットの送信側のホームアドレスが Binding Cache（のサブセット）に存在しない場合、カーネルはユーザランドに位置登録エラーメッセージの送信を促すための通知を行う。また、MN が送信しようとしたパケットの宛先アドレスが Binding Update List（のサブセット）に存在しない場合、認証機能である Return Routability テスト処理の開始を促すための通知を行う。

特に、Mobile IPv6 特有の新機能である Binding Cache と Binding Update List 関連処理が既存の XFRM を拡張して実現できるので、カーネルへの修正が軽微で済んでいる。

ユーザランドの設計

Mobile IPv6 で使用される位置情報管理やシグナリングメッセージの送受信は、1 つのデーモンプログラムで実装されている。Mobile IPv6 で定義されている MN/HA/CN の各ノードの機能は、起動時にオプションとして指定することで切り替えて実行される。

Mobile IPv6 で必要とされる複数の内部処理は、

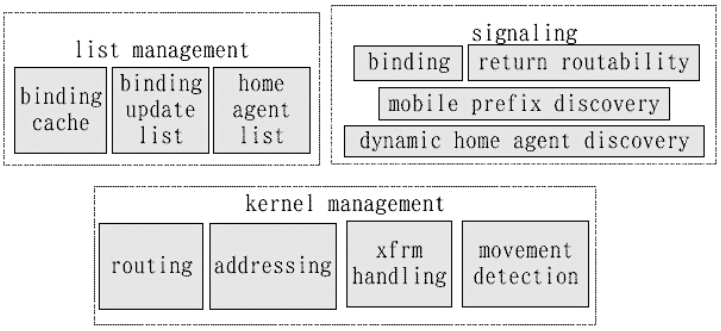

図 2.2. デーモンの機能

軽量化および各データ・パケットの並列処理実現のためスレッドで実装している。

Mobile IPv6 が RFC となる以前のインターネットドラフトでは、すべてのシグナリングメッセージが終点オプションヘッダのオプションとして定義されていたことがあったが、RFC 3775 では IPv6 拡張ヘッダの 1 つであるモビリティヘッダとして定義されたため、カーネルよりもユーザランドでの処理が容易となった。

デーモンプログラムは、イベント管理、タイマ管理の多くを行う。Binding Cache や Binding Update List などの位置情報管理、端末の移動検知はデーモンで処理される（図 2.2）。

システム内では、デーモン内部で管理される Binding Cache と Binding Update List、カーネル内部で保持されるパケット処理のための XFRM 情報の 2 つのデータが存在し、デーモンがそれらの管理および同期を行っている。

Mobile IPv6 では、RFC 3775 により IPsec の使用が必須となっているが、MIPL-2.0 の実装では、カーネルの設計で述べた通り、IPsec で使用されている XFRM を流用している。IPsec と Mobile IPv6 のパケット処理にすべて XFRM が使用されるため、Mobile IPv6 で必要とされる位置情報登録のための通信と、HA と MN 間の双方向トンネルでの通信で必要とされる IPsec の Security Policy Database（SPD）についてもデーモンの設定情報として保持される。

- MN の処理

  MN で行われる処理については、移動検出、HA への位置情報登録、CN への位置情報登録という 3 つの処理が基本となる。

  – 移動検出

  移動検出は、IP 層でのプレフィックスの変化とデフォルトルータの変化により行っている。MN の内部で現在のデフォルトルータを Router Advertisement メッセージから把握しておき、新しい Router Advertisement メッセージを受信したら現在の情報と比較する。変化があった場合は MN は移動したとみなし、移動処理を行う。

  – HA への位置情報登録（Home Registration）

  MN がネットワークを移動したことを検出した後、HA に対して Binding Update メッセージを送信し、Home Registration を行う。

  – CN への位置情報登録（Return Routability/Correspondent Registration）

  CN への通信が始まったことを検出した場合、MN は Return Routability テストを経て CN への Binding Update を送信し、経路最適化を行う。CN との通信は、最初は HA 経由の双方向トンネルを使用する。MN は、このトンネルからパケットを送信するイベントを Return Routability テストを始める契機として利用している。この契機は、XFRM から要求メッセージがデーモンに送信されることを利用している。デーモンはカーネルからのメッセージ受信待ち状態になっており、このメッセージを受信すると Return Routability テストを開始する。Return Routability テストが完了すると、Binding Update 送信用の XFRM ポリシをカーネルに設定して Binding Update を CN へ送信する。Binding Update を送信した後、経路最適化のための XFRM ポリシも登録する。

- HA の処理

  HA は主として位置情報管理、MN 宛パケットの捕捉、双方向トンネルによる MN への転送という処理を行う。

  – 位置情報管理

    MN から Binding Update を受信すると、HA は MN の Binding Cache Entry を作成する。作成後に応答メッセージとして Binding Acknowledgement を MN へ送信する。

  – MN 宛パケットの捕捉

    MN からの Binding Update を正常に処理できた場合、MN のホームアドレス宛パケットが HA へ送信されるように、カーネルの Proxy Neighbor Discovery 機能を有効にする。

  – 双方向トンネルによる MN への転送

    MN からの Binding Update を正常に処理できた場合は、さらに MN への双方向トンネルを設定し、CN からのパケットを MN へ転送する。IPsec を使用しない場合は仮想デバイスとしてのトンネルを使用し、IPsec によりトンネルを保護する場合は XFRM により元々のパケットをカプセル化して転送する。

- CN の処理

  CN は、MN から Binding Update を受信すると経路最適化のための XFRM ポリシをカーネルに設定する。設定された後は経路最適化による通信が可能となる。

Mobile IPv6 と IPsec の連携

USAGI プロジェクトでは、MIPL-2.0 上で IPsec を用いた各種モビリティシグナリングおよびユーザトラフィックの保護を実現すべく実装活動を進めている。これらの具体的な機能は RFC 3776 に記述されている。仕様では手動鍵設定のサポートを必須（MUST）、そして自動鍵交換のサポートを推奨（SHOULD）としている。Mobile IPv6 を用いた実験などでは、手動鍵設定が有効であるが、Mobile IPv6 の本格的な運用や高度なセキュリティ（とくにリプレイ攻撃に対する防御）を実現するためには、自動鍵交換のサポートが必要である。2.6 系カーネルでは、IPsec のコア機能がカーネル内部で実装されており、付随するユーティリティ（ipsec-tools）が BSD から移植されている。したがって、我々はこれらを利用することで上記の目的を実現することができるが、設計上いくつかの修正および拡張が必要であることが判明した。

とくに、Mobile IPv6 における IPsec トンネルの利用は Mobile IPv6 と IPsec の間の密な連携が不可欠であり、これをどのように実現するかが興味深い点といえる。

- IPsec 連携機能の要件

  Mobile IPv6 では、MN と HA の間で双方向トンネルを張り、MN のホームアドレスを利用した通信はすべてこのトンネルを経由する設計となっている。仕様では、この双方向トンネルを IPsec トンネルで代用することを提案しているが、MN が移動する度に IPsec トンネルのエンドポイント（トンネルの外側のヘッダの宛先もしくは送信元アドレス）が変化するため、これを IPsec に知らせてやる必要がある。

  具体的には、特定のトンネルモード Security Association（SA）エントリに含まれるトンネルの入口・出口のアドレスの情報を、必要に応じて与えられた新たなアドレスで更新してやらなければならない。これを実現するためには、(1) 更新すべき SA エントリの特定（どのエントリを更新したらよいのか）、および (2) 更新内容（どのように更新すべきか）が適切に IPsec に知らされる必要がある。

  さらに、K-bit と呼ばれる機能を実現するためには、Internet Key Exchange（IKE）がこれらの情報を知る必要がある。K-bit は、Binding Update/Binding Acknowledgement に含まれるフラグの一部で、MN および HA 上で動作する IKE が互いに張る論理的コネクション（IKEv1 におけるフェーズ 1 コネクション）を MN の移動にともなって維持することが可能かどうかを示すものである。

  K-bit がサポートされている場合、MN と HA 上で動作する IKE（v1）は、MN が移動して IKE のエンドポイントアドレス（すなわち気付アドレス）が変化した場合でも更新されたフェーズ 1 のコネクションを継続して利用することが可能となる。これにより、IKE のシグナリングコストを抑え、帯域の有効利用が可能となる。なお、2.6 系カーネルのケースでは Security Association Database（SADB）に付随して SPD の更新も必要なことがわかっている。2.6 系カー

ネルにおけるSPDエントリは、セレクタにマッチしたパケットに適応すべきIPsec処理（すなわち該当するSAエントリ）を特定する情報をテンプレートとして保持している。このテンプレートには、SAの情報（トンネルモードSAの場合トンネルの入口・出口のアドレスを含む）が含まれているため、これらの情報も同時に更新してやる必要がある。

- IPsec連携機能の設計と実装

我々はKAMEプロジェクトのMobile IPv6開発者およびIPsec WGのracoon開発者、そしてMobile IPv6およびIPsecに造詣の深いFrancis Dupont氏（ENST Bretagne）とMobile IPv6とIPsecの理想的な連携について議論を行った（2004年11月上旬）。

この中でDupont氏は既にPF_KEY Version 2の拡張を用いて上記の機能を実現していることが判明した。議論の参加者の共通の設計目標として、(1) 既存のソフトウェア（Mobile IPv6およびIPsec）に対する変更が少ないこと、(2) 実装が容易であること、(3) システムになるべく依存しないことを挙げ、これを実現しうる最良の方法を検討した。

その結果、Dupont氏のアイデアに基づきPF_KEYの拡張でMobile IPv6からIPsecへのメッセージ通知を行うことで合意した。

このメッセージはPF_KEY MIGRATEと呼ばれる新たなメッセージで、更新すべきSP（Security Policy）およびSAエントリを特定する情報、そして新たなトンネルモードSAの情報が含まれている。Mobile IPv6はこのメッセージを必要に応じて非同期に発行し、システム（IPsecおよびIKE）に移動の事実を知らせる。MNは移動にともない気付アドレスが変化するが、これをシステムにPF_KEY MIGRATEメッセージを用いて通知する。

一方、HAはMNからのBinding Updateを受信することによって、MNの気付アドレスが変わったことを知る。HAはこのタイミングでシステムにMNの移動を通知する。システム（カーネル内のIPsec）は、このメッセージを受信し、処理が適切に済んだ場合にこれを開かれているPF_KEYソケットにブロードキャストする。racoonのようにPF_KEYソケットを見張っているアプリケーションはブロードキャストされたPF_KEY MIGRATEメッセージを聞き、Mobile IPv6で発生した移動の事実を把握することが可能となる。図2.3は、Mobile IPv6とIPsecの連携を示したブロック図である。

- IPsec連携機能のステータス

現在、MIPL-2.0上でPF_KEY MIGRATEおよびこれに対応したracoonの動作テストを行っている。今後は、KAMEプロジェクトおよび慶應義塾大学と協力し、共通のメッセージ通知

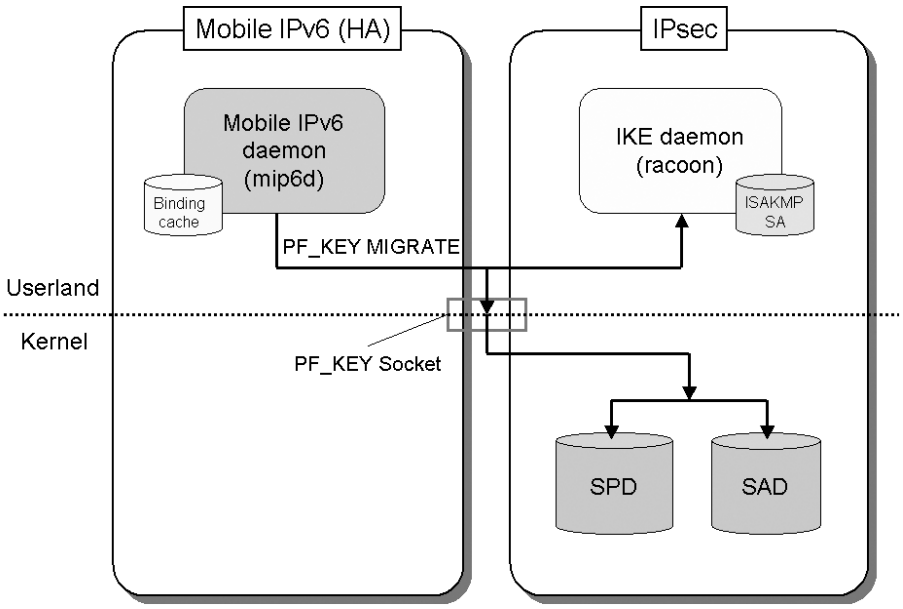

図 2.3. Mobile IPv6とIPsecの連携

機構を実装する予定である。その後 IPsec/IKE に対する Mobile IPv6 の要求事項をまとめ、IPsec WG の開発者にこれを報告する。その結果、MIPL-2.0 および SHISA 上で動作する Mobile 環境に適応可能な IKE デーモンが実現可能となる。

2.1.4 今後の展開

2004 年 11 月、機能制限版のバージョンを MIPL-2.0-rc1 として HUT がテスト公開した。今後は正式リリースの公開を目指して、さらに HUT との共同作業を進めていくとともに、Mobile IPv6 と IPsec との連携機能を強化し、自動鍵交換サポートに注力していく。

2.2 Netfilter

2.2.1 IP バージョン非依存な Connection Tracking の開発

Connection Tracking は、機器に入ってくるパケットすべてを解析し、TCP や UDP のフローの状態の変化を追跡する Linux カーネルの一機能である。Linux には、IPv4 専用の Connection Tracking（以下 ip_conntrack と呼ぶ）が実装されており、Stateful Packet Inspection や NAT の実現に利用されている。一方、IPv6 パケットに対する Stateful Packet Inspection の実現も望まれていたが、IPv6 用の Connection Tracking は存在しなかった。そこで昨年、ip_conntrack のレイヤ 3 部分を一般化した Connection Tracking（以下 nf_conntrack と呼ぶ）の第 1 版を実装した。今年は、より完成度を上げるため、フラグメント化された IPv6 パケットに対する特別な処理の追加やメモリ使用量の低減、現在もメンテナンスが続けられている従来の ip_conntrack に対する変更の追従を行った。以下の項では、nf_conntrack の機能と構成について簡単に説明し、今年行った改善点について説明する。

機能

nf_conntrack の役割は、パケットを走査しフローの状態の変化を追跡することである。現在 nf_conntrack が認識できるフローの状態は以下のいずれかである。

-NEW

新たに検出したフローであり、片方向のパケットしか検出していないことを示す。

-ESTABLISHED

両方向のパケットを検出済みのフローであることを示す。

-RELATED

NEW と同様だが、他のフローと関係のあることが判明したフローであることを示す（例えば、FTP のデータチャネルなど）。

この他にも、特定のカーネルオプションを有効化することにより、フロー中の総パケット数、総バイト数をカウントする機能も持っている。

フローの状態は Stateful Packet Inspection に利用できる。例えば、LAN 外からいきなり来たパケットはすべて破棄するが、過去に LAN 内から LAN 外へ転送されたパケットに対する逆方向のパケットは通過できるようにすることができる。このルールを実現するには、LAN 外から来たパケットを走査した後フローの状態が NEW ならばパケットを破棄し、ESTABLISHED、または RELATED ならば通過を許可すればよい。また、従来の NAT 機能は ip_conntrack が追跡したフローの状態の情報を利用して NAT マッピングを管理している。現在 nf_conntrack を利用した IPv4 NAT 機能は存在しないが、今後対応される方向にある。

構成と処理内容

nf_conntrack は、ネットワークプロトコル、トランスポートプロトコル、アプリケーションデータのそれぞれの走査モジュールと、コアモジュールで構成される。現在、nf_conntrack が対応しているプロトコルは、IPv4、IPv6、TCP、UDP、SCTP、ICMP、ICMPv6、FTP である。nf_conntrack は、パケットをルーティングする前に、ネットワークプロトコルとトランスポートプロトコルの走査モジュールを用いて、パケットがどのフローに属するかを判別し、そのフローの状態を保持するデータベースを更新する。

なお、フローは tuple と呼ばれる識別子で識別される。tuple は、例えば TCP ならば送信元、送信先のネットワークアドレス、ポート番号、ネットワークプロトコルの種別、トランスポートプロトコルの種別からなる。

次に、走査を希望するアプリケーションデータ走査モジュールが存在すれば、そのモジュールはパケットのアプリケーションデータを走査する。これにより、そのパケットが属するフローに関係するフローが、新

たに生成される可能性があるかどうかを判定する。もしその可能性があり、実際に予期したフローを検知した場合、予期されていたフローの状態はRELATEDになる。

フラグメント化されたIPv6パケットの処理

nf_conntrackは、パケット中のアプリケーションデータも走査する可能性がある。そこで、パケットがフラグメント化されている場合は、再構築する必要がある。従来のip_conntrackでは、パケットを走査する前にパケットを再構築する。そして、パケットをネットワークインタフェースから出力する前にパケットを再分割する。このため、パケットサイズが入力時と出力時で異なる可能性がある。RFC 2460によれば、IPv6ではルータがパケットを分割しないとされているため、nf_conntrackが上記のようなパケットのサイズ変更をしない方が好ましい。そこで、パケットを再構築する際は、元々のパケットを保持しておき、そのクローンを再構築に利用する。ここで、クローンとは、パケットのデータをコピーせず、パケットを管理する構造体のみをコピーしたものである。パケットをネットワークインタフェースから出力する際には、保持しておいた元々のパケットを出力し、再構築したパケットは破棄する。以上により、パケットサイズを変更せず、再構築したパケットの走査を可能にした。

メモリ使用量の低減

nf_conntrackが追跡するフロー数の増大は、そのままメモリ使用量に影響することから、フローに関する情報を機能ごとに分類し、その機能ごとに適したメモリ割り当てを行うよう改善した。たとえば、従来のip_conntrackではフローの情報にIPv4 NAT機能のための情報も含まれていたが、IPv6のフロー情報には必要ないためそれを含まない。

従来のIPv4用Connection Trackingに対する変更の追従

nf_conntrackの開発の一方で、メインラインのカーネルには、従来のip_conntrackに対するバグフィクス、機能追加が行われている。そこで、nf_conntrackのコードが機能的に古くならないよう、適宜ip_conntrackに対する変更をバックポートした。2004年12月現在、nf_conntrackは最新の安定バージョンであるLinux 2.6.9に対応しており、ip_conntrackと同等の機能を有している。

今後の予定について

nf_conntrackの開発に際しては、Linuxにおけるパケットフィルタ、NAT機能に関する開発、メンテナンスを行っているNetfilterプロジェクト（`http://www.netfilter.org`）とメーリングリスト上で密に議論しながら進め、9月には同プロジェクトのコアメンバーとドイツ・エアランゲンにおいてミーティングを行った。また同プロジェクトは、メインラインのカーネルに含まれていない拡張機能のパッチ群を公開、管理しており、nf_conntrackもそれに取り込まれた。

今後、Netfilterプロジェクトでは Connection Trackingに関する機能追加を従来のip_conntrackではなくnf_conntrackに対して行っていく方向にあり、安定化などがなされた後、メインラインカーネルに取り入れられることと思われる。

2.2.2 パケットフィルタ機能の品質向上化

Linuxカーネルにおける IPv6パケットフィルタのモジュールを調査したところ、多数のバグが存在しコード品質が低いことが判明したため、これらを修正した。Logging機能や拡張ヘッダ内の情報を使うルールを使用する場合、2.4.29または2.6.10以降のLinuxカーネルを使うことが推奨される。

また、従来のIPv6パケットフィルタ機能では、パケットをフィルタルールと比較する前に、パケットのデータがメモリ上で連続的になるよう、必要ならばメモリコピーを行っていた。Linuxカーネルでは、可能な限りメモリコピーの発生を抑えるため、メモリ上で断片化されたデータをパケットとして扱えるようパケット管理機構に工夫が施されているが、上記の挙動により、その工夫が無駄になっていた。そこで、IPv6パケットフィルタ機能でも、パケット管理機構を最大限に利用して、メモリコピーの発生を抑えるようにした。

2.3 TAHI Automatic Running System

2.3.1 TAHI Conformance Testを利用したRegression Testの開発

USAGIプロジェクトはLinuxを中心によりよいIPv6環境を提供することを目的とし活動を続けてき

た。この活動の一環として、メインラインカーネルに対して開発した USAGI カーネルパッチのマージ作業を行っている。

メインラインカーネルは、改良・修正活動が活発に行われており、ネットワーク周りのコードに関しても、USAGI プロジェクト以外によるパッチが日々取り込まれている。この日々変更されるカーネルコードに対し、メンテナおよび多くの開発者はバグが混入しないよう目を光らせているが、変更によって副作用が生じる可能性は常に付きまとう。

そこで、毎日リリースされているメインラインカーネルのスナップショットに対し、リリースごとに IPv6 環境の機能をテストするシステムを開発した。IPv6 機能のテスト自身には、TAHI プロジェクトの TAHI IPv6 Conformance Test Suite（`http://www.tahi.org`）を利用した。この自動テスト実行システムにより、IPv6 機能に副作用が生じた際にも速やかに修正・対処ができる。

このシステムは現在実験的に一般公開されており、`http://testlab.linux-ipv6.org` に IPv6 接続することにより、デモを見ることができる。

2.3.2 システムの流れ

システムは、新しいカーネルリリースの待ち受け、ビルド、テストの工程を繰り返す。各工程の遷移を図 2.4 に示す。

システムはテストを行っていない間、新たなカーネルリリースを待ち受ける。待ち受けるリリース対象は、各安定バージョンのほかに、安定バージョンの準備段階である rc バージョンおよび毎夜リリースされる bk バージョンである。

新たなリリースを発見すると、システムは発見したカーネルのビルドを開始する。カーネルの configure、build、install の過程は自動で行われる。各過程のログは保管されるため、ビルドエラーなどは後ほど解析することができる。各過程の処理が失敗に終わった際は、ソースに問題があったとして次のリリースを待ち受ける。

カーネルのビルドが終わると、システムはテストの実行に先立ち、ビルドしたカーネルに問題がないか確認するために被テスト対象を再起動する。再起動時のログもテスト結果にあわせて保管される。再起動に失敗した際には、システムはいったん停止し、マニュアルでの確認を待つ。

ビルドが終了し再起動に成功すると、システムはそのカーネルに対し TAHI Conformance Test を実行する。実行したテストの結果を保管する際、システムは前回の結果と変化した点を抽出し別途保管する。この変化した点の確認により、新規パッチによって副作用が起こっていないか確認することができる。

カーネルのテストが終わると、再び新たなカーネルリリースの待ち受け状態に戻る。この待ち受け状態への移行に先立ち、被テスト対象は安定した状態に戻すために再起動され、安定していると確認されているカーネルへと戻される。この再起動のログも保管される。

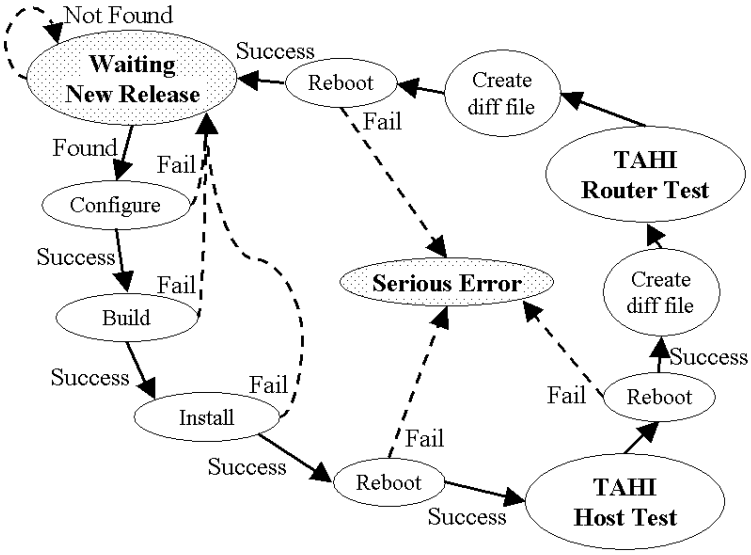

図 2.4.　システムの流れ

2.3.3 収集データと情報へのアクセス

システムは、テストの結果、前結果との差分、テストしたカーネルのソースとそのコンパイル後のバイナリ、カーネルビルドなどテストの各過程におけるログといったデータを収集する。1 回のカーネルリリースに対し、収集されるデータを図 2.5 に示す。

各データには、ウェブブラウザによりアクセスすることができる。閲覧対象データをできるだけ素早く発見できるようにブラウザ画面も工夫した。ウェブブラウザからのアクセス例を図 2.6 に示す。

2.3.4 今後の展開

本システムは 2004 年 12 月現在、メインラインカーネルバージョン 2.6 のみを対象とし、また走らせているテストは TAHI Test Suite のうちの IPv6 Ready Logo Phase I テストのみである。今後の展開として、以下の事項を検討・開発中である。

1) USAGI カーネルなど、異なるカーネルリリースの並列実行
2) IPv6 Ready Logo Phase II テストなど、異なるテストの順次実行

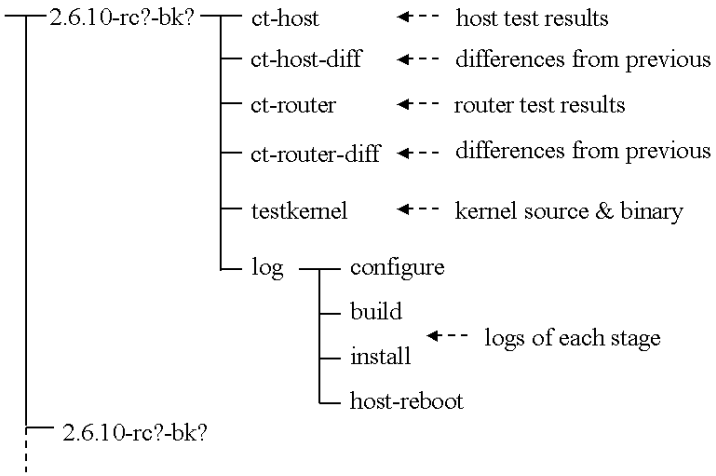

図 2.5. 収集したデータ

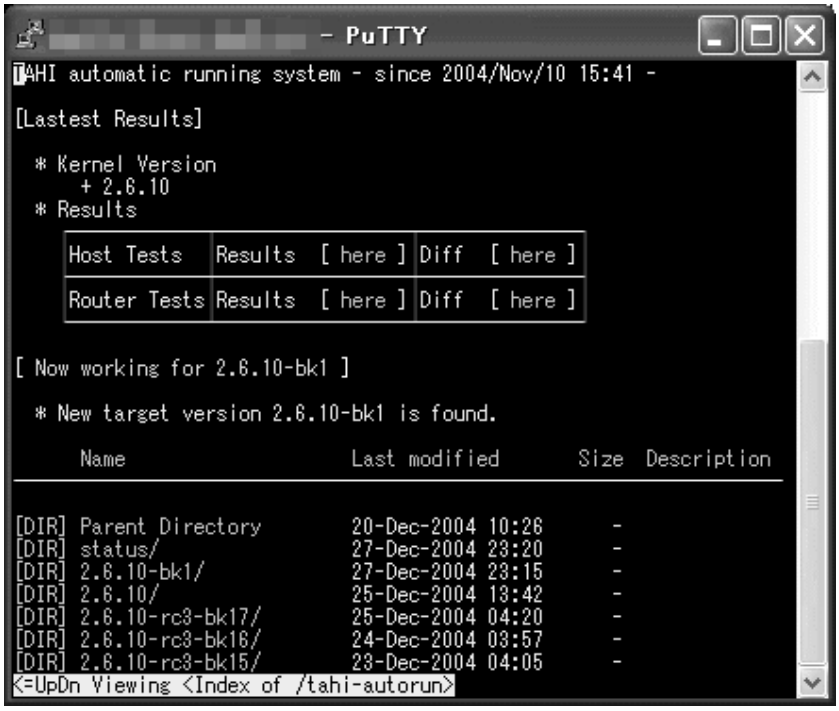

図 2.6. ウェブブラウザよりのアクセス例

2.4 IPsec

IPsec は、IP を用いた通信に対して IP レイヤで機密性、完全性、認証といったセキュリティサービスを提供する。このため既存のアプリケーションに手を加えることなくセキュリティサービスを提供できるという利点を持つ。IPsec の構成は、大きく 2 つに分けることができ、1 つは、パケットを処理する機能であり、もう 1 つは、パケットを処理するために必要な共有鍵や鍵の有効期限などを交換する機能である。IPsec には、Authentication Header（AH）と Encapsulating Security Payload（ESP）があり、それぞれにトンネルモードとトランスポートモードが定義されている。

2.4.1 Linux における IPsec

Linux では、パフォーマンスの観点から IPsec のパケット処理はカーネル内に実装され、鍵交換はデーモンとしてユーザランドに実装される。ユーザランドとカーネル間のインタフェースは KAME と互換性のある PF_KEYv2（RFC 2376）に独自の拡張を加えたものと netlink socket によって実装され、どちらのインタフェースを使ってもセキュリティポリシと IPsec SA の設定ができる。

Linux の IPsec は、Linux-2.4 までは FreeS/WAN プロジェクト（`http://www.freeswan.org/`）によってパッチという形で提供されてきた。しかし、Linux-2.5 からはカーネルメンテナによって FreeS/WAN とは異なる独自のアーキテクチャを用いた IPsec がメインラインカーネルに取り込まれている。このため最新の安定版である Linux-2.6 からは USAGI プロジェクトで行った IPsec の IPv6 サポートとあわせることで、カーネルをビルドする際に設定するだけで IPv4、IPv6 の両方で IPsec の機能を使用することができる。

2.4.2 鍵交換デーモン

鍵交換デーモンに関しては、Linux に対応した racoon が ipsec-tools（`http://ipsec-tools.sourceforge.net/`）プロジェクトから提供されており、使用することができる。また、FreeS/WAN の後継プロジェクトである OpenS/WAN（`http://www.openswan.org`）からは、Linux-2.6 に対応した IKE デーモン pluto が提供されている。

また、IKE の次期バージョンである IKE version 2（IKEv2）や認証に Kerberos を用いた鍵交換プロトコル KINK をサポートする鍵交換デーモンを WIDE プロジェクトの IPsec WG で開発しており、これも Linux-2.6 をサポートする予定である。

2.4.3 xfrm と stackable destination

Linux-2.6 では、IPsec を実装するにあたり、xfrm と stackable destination というアーキテクチャが導入されている。

xfrm は、xfrm_policy、xfrm_tmpl、xfrm_state 構造体からなり、パケット処理のポリシと実際の処理に必要な情報を分離して管理する構造である。IPsec では、xfrm_policy がセキュリティポリシに相当し、xfrm_state が IPsec SA に相当するように実装されている。xfrm_policy と xfrm_state の関連付けは、xfrm_tmpl によって行われる。

入力処理は、パケットを処理するために先に

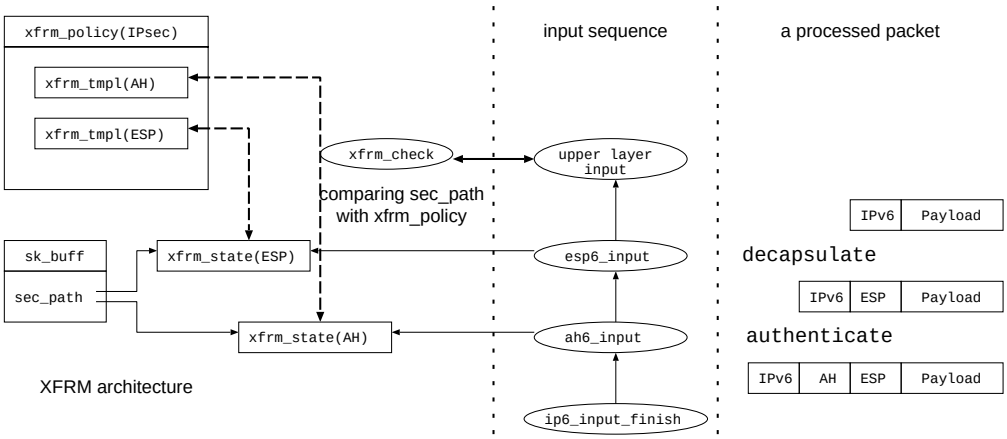

図 2.7.　IPsec の入力処理

xfrm_state が検索され、使用された xfrm_state は、skb の sec_path に保持される。セキュリティポリシとのマッチングはそれらを xfrm_policy の持つ xfrm_tmpl と比較することで行われる。

stackable destination は、出力処理を効率化するための仕組みで、ルーティングテーブル内で最終的な送信先を示す dst_entry 構造体をスタック構造にし、その出力関数（output）を順次呼び出すことで実現されている。dst_entry の output は、通常パケットを送信する処理にあたり、dst_entry の output を呼び出す時点で IP ヘッダを含むパケットは完成している。IP パケットのフラグメント処理はさらにこの後に行われる。

2.4.4 カーネルでの Mobile IPv6 サポート

Mobile IPv6 の実装にあたっては、RFC 3776 にあるように、Mobile IPv6 のシグナリングを IPsec によって保護する必要がある。Mobile IPv6 のシグナリングにおいて、IPsec の保護を必要とするのは、Mobile Node（MN）と Home Agent（HA）間で行われる Home Registration と Mobile Node と Correspondent Node（CN）間で行われる Routing Optimization である。仕様では、Home Registration を MN–HA 間のトランスポートモードで保護し、Routing Optimization を MN–HA 間のトンネルモードで保護する。

USAGI プロジェクトでは、Mobile IPv6 を上記の

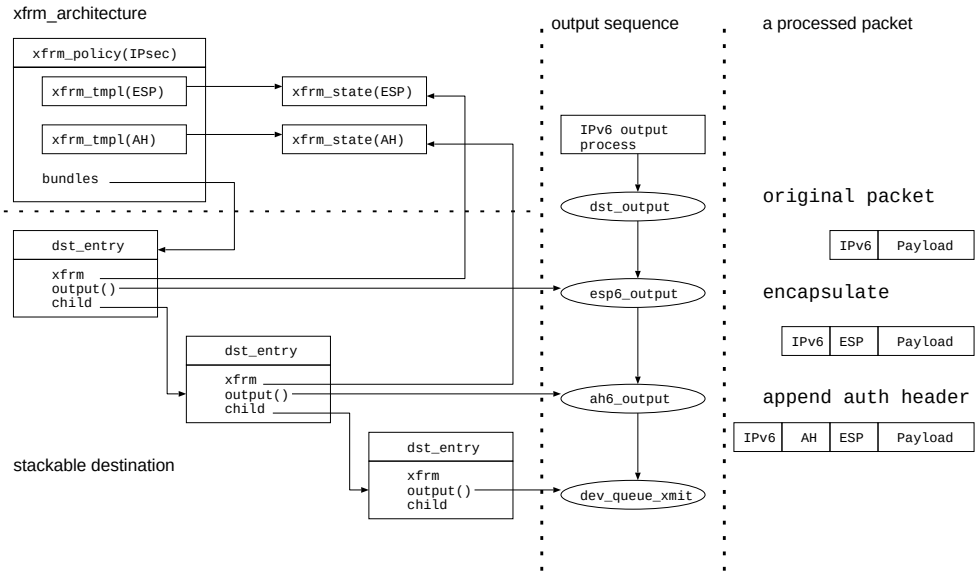

図 2.8.　IPsec の出力処理

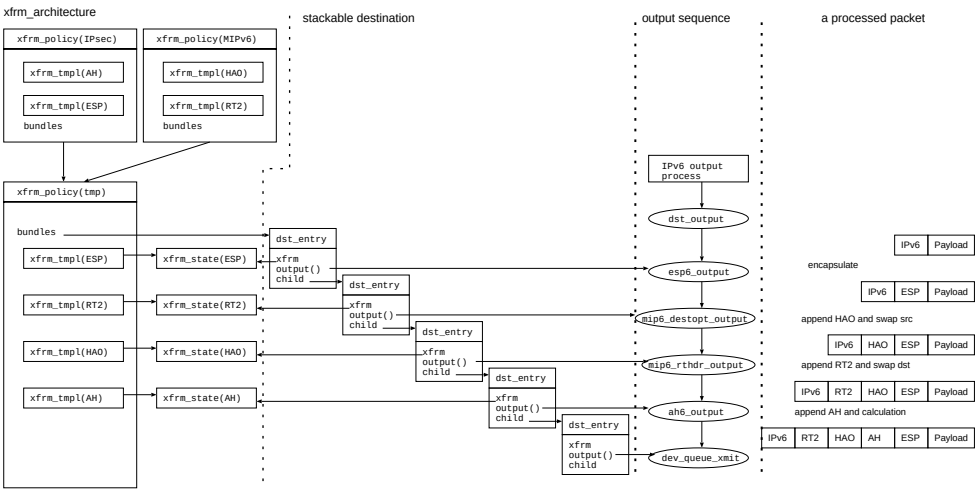

図 2.9.　MIPv6 の xfrm_policy と IPsec の xfrm_policy のマージ

xfrm と stackable destination を用いて実装している（Mobile IPv6 については、2.1 節を参照）。このため、Mobile IPv6 に対して IPsec を適用するためには、Mobile IPv6 と IPsec の両方から使用される xfrm_policy と、xfrm_state を検索する際のパラメータを保持する xfrm_tmpl を適切に保つ必要がある。

現在の実装では、Mobile IPv6 を管理する mipd が、Mobile IPv6 の処理に必要な xfrm_policy と xfrm_state に加え、それらを保護するために必要な IPsec のための xfrm_policy と xfrm_state を設定する。この設計は、Mobile IPv6 関連の設定が一元化されるという利点もある。しかし、一方でユーザが設定したセキュリティポリシによって Mobile IPv6 のパケットが思わぬ影響を受けてしまうことも考えられる。特に CN との通信では、MN–HA 間のトンネル SA に加えて MN のホームアドレスと CN 間にも IPsec を設定する可能性がある。このような設定では、(1) MN のホームアドレスと CN のアドレス間のトンネル、(2) MN の気付アドレスと HA 間のトンネルを正しい順序でパケットにて供する必要がある。そのため今後は、Mobile IPv6 の設定と IPsec の設定を設定時にカーネル内で正しい順序でマージするような設計に変更する予定である。

2.5 IPv6 Ready Logo 取得

2.5.1 IPv6 Ready Logo Program 参加の目的

IPv6 Ready Logo Program とは、国際認証機関である IPv6 Ready Logo Committee（`http://www.ipv6ready.org`）により行われている国際的接続認証活動である。2004 年 12 月現在、IPv6 実装の基礎的な相互接続性を確認対象とした Phase-1 認証が行われており、120 を超える製品が Phase-1 認証を取得している。

USAGI プロジェクトにおいても、提供する成果物の信頼性の高さを示すため、この IPv6 Ready Logo Program に参加し、国際的接続認証を取得することにした。

2.5.2 2004 年度の成果

USAGI プロジェクトは 2004 年度、独自に提供するコードにて IPv6 Ready Logo Program に参加した。2 月には 2.6 系カーネルにおいて、ホスト・ルータ両機能について IPv6 Ready Phase-1 Logo を取得した。さらに 9 月には 2.4 系、2.6 系の両カーネルバージョンにおいて、ホスト・ルータ両機能について IPv6 Ready Phase-1 Logo を取得した。

2.5.3 1CD Linux “KNOPPIX/IPv6” との協業

USAGI プロジェクトは自身が提供するコードによる IPv6 Ready Logo Program への参加のみならず、提供するコードを実際に用いた製品である KNOPPIX/IPv6（`http://www.alpha.co.jp/knoppix/ipv6/`）の IPv6 Ready Logo Program への参加を協業した。

KNOPPIX（`http://unit.aist.go.jp/itri/knoppix/`）は 1 枚の CD のみでブートするインストール作業が不要な Linux ディストリビューションである。KNOPPIX/IPv6 では、USAGI プロジェクトが提供するコードが採用された。これにより、KNOPPIX の「CD-ROM を起動するだけでインストールや設定不要」という特徴と、USAGI プロジェクトの高機能な IPv6 実装が合わさり、初心者でも気軽に IPv6 の世界を体験できる 1 CD OS を実現した。

KNOPPIX/IPv6 の IPv6 対応状況を要約すると、まずアプリケーションでは、Web ブラウザ（Mozilla、Konqueror）、メールソフト（Sylpheed、Kmail）など、デスクトップ用途で一般的に利用するアプリケーションが IPv6 対応し、収録されている。次に、OS としてのネットワークの IPv6 対応に関しては、6to4 機能が内蔵されている。この機能はユーザが常に IPv6 接続環境に接続するとは限らないため内蔵された。6to4 機能は IPv4 接続環境であってもそれを自動検出し、ローカルネットワーク外の IPv6 接続環境に対してトンネル接続を行う。これによりユーザは今から接続しようとするネットワーク環境を、それが IPv4 であるか IPv6 であるかを意識することなく利用することが可能である。

KNOPPIX/IPv6 における具体的な協業内容としては、まず USAGI プロジェクトが開発していた 2.4 系、2.6 系の両カーネルバージョンのコードの提供を行った。また単に提供するだけでなく、KNOPPIX のカーネルコードに対するマージ作業も共同で行った。

KNOPPIX/IPv6 は 2.4 系、2.6 系の両カーネルバージョンにおいて IPv6 Ready Logo Phase-1 を 2004 年 9 月に取得している。この認証取得に必要な相互接続性試験も USAGI プロジェクトと KNOPPIX

開発者の共同作業として慶應義塾大学新川崎 K2 タウンキャンパスにて行った。

2.5.4 今後の展開

USAGI プロジェクトでは相互接続性の品質向上のため、今後も IPv6 Ready Logo Phase-1 Program に参加していく予定である。

また、IPv6 Ready Logo Committee では、実ネットワークでの使用に耐えうるかを検証対象とした IPv6 Ready Logo Phase-2 Program を開始した。IPv6 Ready Logo Phase-2 Program では IPv6 の基本機能のみならず、IPsec、MIPv6、MLD なども検証対象としている。USAGI プロジェクトは IPv6 Ready Logo Phase-2 Program にも積極的に参加し、基本機能のみならず、IPsec、MIPv6、MLD など各機能の品質向上を目指す。