

第 VIII 部

Linux における IPv6/IPsec スタックの研究開発

第 8 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 を中心により良い IPv6 環境を提供することを目的としたプロジェクトである。本プロジェクトは WIDE プロジェクトを中心として、有志によって構成されている。USAGI プロジェクトの活動内容は、Linux の IPv6 スタックや IPv6 に関するライブラリ・アプリケーションの改良および公開、成果物のコミュニティへの還元といった開発活動、また機会を捉えての啓蒙活動などである。これらの活動は KAME プロジェクト、TAHI プロジェクトと連携をとりつつ行っている。

USAGI プロジェクトでは、成果物を 2 週に 1 度の snapshot と、年数回を目処とした stable release を公開している。これら成果物は、前述のコミュニティへの還元活動の結果、メインラインカーネルに新たな機能や改善として多く取り込まれた。USAGI プロジェクトは、今後も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 2005 年の主な活動

2.1 IPv6 Mobility の設計と開発活動

2.1.1 はじめに

USAGI プロジェクトは、RFC 3775 と RFC 3776 を実現する Mobile IPv6 プロトコルスタックが Linux の機能として取り入れられることを目標として、ヘルシンキ工科大学 (HUT) の GO-Core プロジェクトと本格的な協力体制を 2003 年より開始し、MIPL2 の開発に取り組んでいる。TAHI プロジェクトで公開されている MIPv6 Conformance Test の 100% PASS および IPv6 Ready Logo Program Phase-2 の取得による品質の安定化を確保した上での MIPL2 正式リリースを目指している。また、Mobile IPv6 と IPsec/IKE 連携機能を強化し、racoon の改造や、PFKEYv2 インタフェースの拡張の提案を行っている。

2.1.2 2005 年度の活動

2.1.2.1 実績

本年度の主な実績を図 2.1 に示す。

2.1.2.2 ソフトウェアの特徴

MIPL 2.0 RC2 は、カーネルが Linux 2.6.8.1 ベースであり、以下が実現されている。

- RFC 3775 のすべての機能
- RFC 3776 のうち、Mobile Node (MN) と Home Agent (HA) 間のモビリティヘッダメッセージ

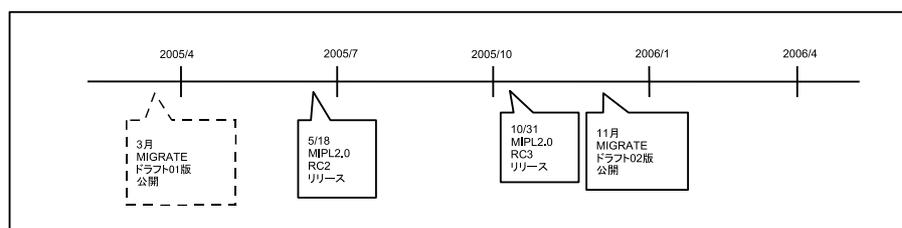


図 2.1. 実績 (2005 年度)

の IPsec による保護機能 (手動鍵設定のみ)

RC3 は、カーネルが Linux 2.6.11 ベースであり、RC2 までの実現機能に加えて、MIPv6 Conformance Test の Correspondent Node (CN)、HA、MN で 100% PASS の品質を達成している。

RC3 リリース以降は、カーネルを Linux 2.6.14 ベースへとアップデートし、IPv6 Ready Logo Phase-2 MIPv6 の取得を目指している。

2.1.2.3 ソフトウェアの品質 (Mobile IPv6)

USAGI プロジェクトでは、MIPL 2.0 の品質調査および回帰試験として MIPv6 Conformance Test を定期的に行っている。MIPv6 Conformance Test では、項目ごとに、PASS、FAIL、WARN、SKIP、Not Available (N/A) のいずれかが結果として得られる。Tester が想定する試験が実施された場合は PASS、FAIL、WARN となり、試験設定により試験を行わなかった場合や、Tester 側の内部エラーにより試験が行われなかった場合等は SKIP や、N/A となる¹。

CN の MIPv6 Conformance Test の結果の遷移を図 2.2 に示す。CN は、N/A 項目が 1 件 (IPsec サポート) あるだけで、それ以外は PASS である。

HA (IPsec 有り) の MIPv6 Conformance Test の結果の遷移を図 2.3 に示す。ここでは、IPsec 無し項目や、外部網接続モード (2 つの異なるリンクを HA が保持する状態) での試験項目を無効としているため、SKIP が 66 項目程度あるが、それ以外の多くは比較的安定して PASS していることがわかる。

CN、HA については、すでに MIPv6 Conformance Test を 100% PASS する品質に達していると言えるが、MN については、RC3 リリースでは 100% PASS に成功したものの、年度当初より FAIL が散見され、MN において FAIL となっている試験項目を PASS にすることが、基本機能の実装における最大のテーマとなっている。MN の MIPv6 Conformance Test の結果の遷移を図 2.4 に示す。

MN で FAIL が生じている原因としては、以下の要素があげられる。

- MN がリンク間を移動する前に Neighbor Unreachability Detection (NUD) を実施しない不具合
- ホーム網に戻ったときに、正常に De-registration 出来ない不具合

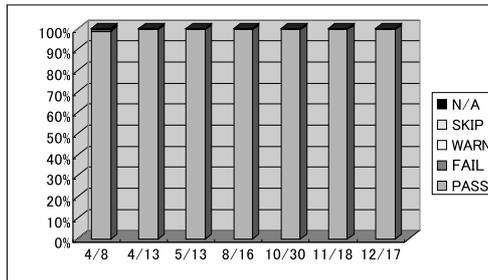


図 2.2. CN の MIPv6 Conformance Test 結果遷移 (2005 年度)

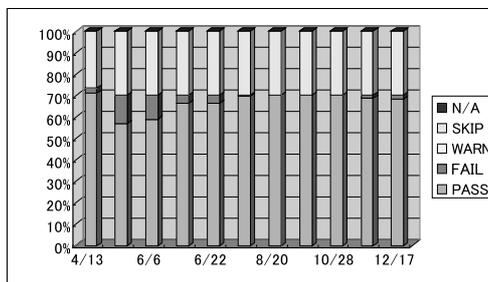


図 2.3. HA (IPsec 有り) の MIPv6 Conformance Test 結果遷移 (2005 年度)

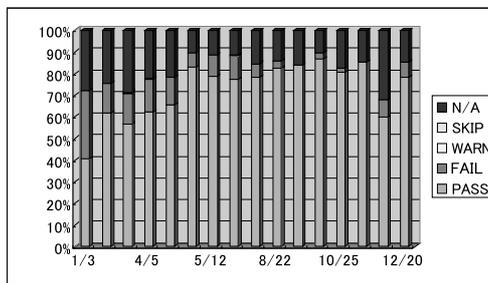


図 2.4. MN の MIPv6 Conformance Test 結果遷移 (2005 年度)

- Return Routability の Home Test Init (HoTI) 及び Care-of Test Init (CoTI) メッセージの再送タイマと Binding Update (BU) メッセージの再送タイマの不整合により BU が送信できない不具合

上記修正をほどこしたあと、RC3 をリリースし、この時点で CN、HA、MN すべてで MIPv6 Conformance Test を 100% PASS を確認した。

2.1.2.4 ソフトウェアの品質 (IPv6 Core Protocol)

IPv6 Ready Logo Program Phase-2 MIPv6 を取得するためには、MIPv6 Conformance Test による

1 Tester の詳細については TAHI プロジェクトのホームページ <http://www.tahi.org> を参照のこと。

検証に加え、IPv6 自体の動作品質を検証する IPv6 Ready Logo Phase-2 Conformance Test での検証も必要となる。

カーネルが Linux 2.6.14.5 であれば、Host 機能、Router 機能共に IPv6 Ready Logo Phase-2 Conformance Test を 100% PASS できる品質を持ち合わせているため、現在の MIPL2 では、Mobile IPv6 機能を追加したことによって IPv6 Core 機能の品質劣化が起きていないことを確認することが必要となる。

CN における Host 機能および HA における Router 機能の検証においては特に問題が生じることはなかった。しかし、MN における Host 機能ではカーネルにおける RA 受信機能を停止し、ユーザランドが RA 受信処理を行うことに起因した問題によって FAIL する項目が多数生じた。

カーネルでは RA 受信を停止することができ、MN はユーザランドによってカーネルに換わって RA を受信する。RA 受信処理は、default router の選択、prefix 情報の管理および近隣情報(Neighbor Cache)の管理等を中心とした処理であるが、IPv6 Ready Logo Phase-2 Conformance Test の詳細な要件を満たす品質には至っていなかった。

そこで、カーネルに RA 受信可否に加えて、RA 受信ともなう default router の選択の可否、reachable time の設定の可否、prefix information の設定の可否を選択する機能を新たに追加した。

これらの機能により、ユーザランドでは default router の選択と prefix 情報の設定を行い、reachable time の設定や近隣情報の登録などの機能をカーネル

でおこなうよう変更した。ユーザランドでは、MN が移動検知とともに default router の選択等の最低限の機能を有するに留めることができ、その他の処理は元々動作実績のあるカーネルへ委譲することができた。

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ている。

2.1.3.1 カーネルの設計

カーネルの機能ブロック図を図 2.5 に示す。

カーネル機能の詳細は、2004 年度 WIDE 報告書を参照のこと。本年度の差分は、IPsec のエンドポイントアドレス書き換え機能として、MIGRATE 機能をサポートしたことである。MIGRATE 機能に関しては 2.1.3 節 Mobile IPv6 と IPsec の連携を参照のこと。

2.1.3.2 ユーザランドの設計

Mobile IPv6 のプロトコル処理のデータ管理や状態遷移などのロジックは、すべてデーモンで行うように実装されている。詳細は、2004 年度 WIDE 報告書を参照のこと。

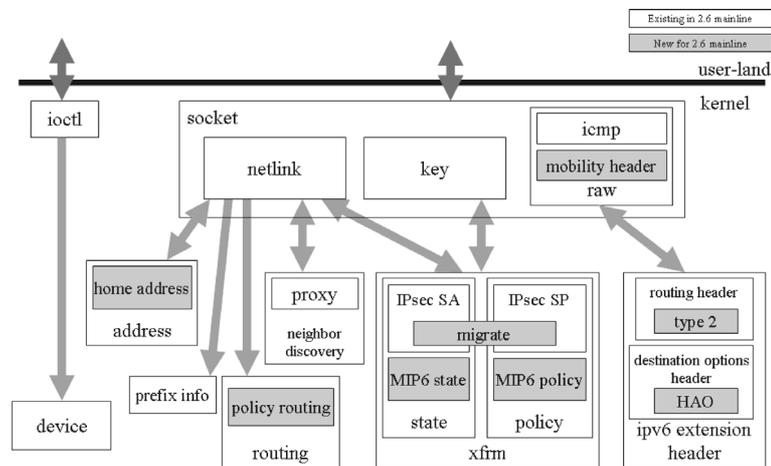


図 2.5. カーネルの機能ブロック図

2.1.3.3 Mobile IPv6 と IPsec の連携

Mobile IPv6 と IPsec/IKE の連携に関しては若干の進捗があった。MIPL 2.0 では、昨年より PF_KEY の拡張機能(MIGRATE 機能)によって Mobile IPv6 と IPsec/IKE のスムーズな協調動作を実現してきた。この拡張機能をサポートするためには、Mobile IPv6 デモン (mip6d) の PF_KEY ソケットへのアクセスが必要となり、結果として ipsec-tools 由来の libipsec ライブラリをソフトウェアコンポーネントの一部に取り込む形となっていた。Mobile IPv6 と IPsec/IKE の連携の観点からの要求事項は満たされたものの、副作用としてソースコードの肥大化を引き起こしていた。

これを受けて GO-Core プロジェクトと協議した結果、Linux との親和性が高い Netlink²によって MIGRATE 機能を実現することによって、ソースコードの肥大化を回避する選択を行った。すなわち、mip6d は PF_KEY ソケットの代わりに Netlink ソケットに対して MIGRATE メッセージを発行し、移動を通知する。MIGRATE メッセージはカーネルによって処理され、SPD および SAD が更新される。次に、MIGRATE メッセージはすべての開かれた PF_KEY ソケットおよび Netlink ソケットに対してブロードキャストされる。これにより、PF_KEY にもとづいて実装された既存の IPsec ソフトウェア(例: racoon、racoon2)はこれまでと同様に MIGRATE 機能を実現することが可能である。最新のコードでは、これまで libipsec に依存していた部分がすべて Netlink ベースのものに置き換えられている。カーネルからユーザランドへの MIGRATE メッセージの広告は現時点では未サポートであり、手動鍵交換のみがサポートされている。カーネルコードは 2006 年初頭にサポートが予定されており、これにより自動鍵交換のサポートが可能となる。

PF_KEY は RFC2367、Netlink は RFC3549 として、いずれも IETF より仕様が Informational RFC として提案されている。Linux-2.6 では多くのユーザスペース カーネル間の情報交換が Netlink で実現されている。これまで PF_KEY で実現されていた IPsec データベースへのアクセスを Netlink で行うことも可能である。Linux-2.6 開発のおおまかな流れとしては、Netlink によるインタフェースの統

一化に向かっているが、既存のソフトウェアとの下位互換性を保つことが重要な課題となっている。

2.1.4 Linux Zaurus への対応

USAGI プロジェクトでは、これまでににおいても USAGI スタックを Linux Zaurus へポーティングして動作検証を行ってきた。特に、PDA の特性から MIPv6 の MN の機能を搭載して検証を続けてきた。

昨年度以前は、SHARP から提供されている Zaurus スタックのカーネルバージョンが 2.4 をベースにしたものであったため、現在開発を進めている 2.6 をベースにした MIPv6 の実装をポーティングすることはできなかった。

今年度は、SHARP の研究部門が実装した Zaurus SL6000 に対応した 2.6.11 対応カーネルの提供により、Zaurus SL6000 に対して 2.6.11 をベースとした RC3 のポーティング作業を開始した。

ポーティング作業の中心は、コンパイラが異なることによる解釈の違いに起因する問題を解決する作業となった。現時点においては PC の動作と同等の品質には至っていないが、MIPv6 Conformance Test の 100%の消化を目指して作業をすすめ、PDA における Mobile IPv6 の有効性を検証していく。

2.1.5 今後の展開

現在は、ベースとなっているカーネルが 2.6.14 に上がったことにより生じた問題を修正する作業に取り組んでおり、2005 年度 1 月開催予定の TAHI 相互接続テストイベントにおいて、CN、HA、MN 各ノードおよび、Linux Zaurus による IPv6 Ready Logo Program Phase-2 MIPv6 の取得に対して備えている。また、Mobile IPv6 と IPsec/IKE 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 IPv4/IPv6 に対応した Connection Tracking の開発活動

2.2.1 概要

USAGI プロジェクトは、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おり、パケットフィルタ機能もそのひとつとなっている。今年度、USAGI プロジェクトは昨年度に引き続き、IPv4/IPv6 に対

2 NETLINKについては、NETLINKのmanページ<http://www.linux.or.jp/JM/html/LDP_man-pages/man7/netlink.7.html>参照。

応した Connection Tracking を開発した。以下では、その内容について報告する。

2.2.2 背景

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 と記す) 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 パケットを扱える Connection Tracking は存在しなかった。そこで 2003 年、ip_conntrack のレイヤ 3 部分を一般化した Connection Tracking (以下 nf_conntrack と記す) を実装し、その後メインラインカーネルへの採用を目指して安定化と効率化に努めてきた。

2.2.3 2005 年度の開発内容

USAGI プロジェクトは今年度、ip_conntrack から nf_conntrack への移行を容易にするため互換性に関する課題に対処した。また、ip_conntrack に対する変更の追従を行った。これらにより nf_conntrack に対する課題がすべて消化されたため、11 月メインラインカーネルに採用された。以下ではその内容について報告する。

2.2.3.1 Connection Tracking を利用するモジュールの互換性対応

nf_conntrack は ip_conntrack を置き換える機能である。そのため、これまで ip_conntrack の機能を利用していた IPv4 パケットフィルタのカーネルモジュールを、nf_conntrack に対応させる必要がある。ただし、ip_conntrack から nf_conntrack への緩やかな移行を可能にするため、ip_conntrack がしばらく Linux カーネルに残されることを考慮する必要がある。そこで、ip_conntrack と nf_conntrack に対し両者に共通の API を追加した。これにより、IPv4 パケットフィルタの開発者は、ip_conntrack、nf_conntrack のどちらがカーネルに組み込まれるかを意識せずに開発できるようになった。

2.2.3.2 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 に対する変更を追従した。最も大きな追加機能は、ユーザプロセスからフローの状態を取得、変更可能にする機能であり、現在も USAGI プロジェクトは同機能の著者と共同で移植作業にあっている。

2.2.4 開発体制

nf_conntrack の開発に際しては、Linux におけるパケットフィルタ、NAT 機能の開発、メンテナンスを行っている Netfilter Project (<http://www.netfilter.org>) と密に連携しながら進めた。また、10 月には同 Project のコアメンバや主要な Netfilter の開発者で行われる workshop に USAGI プロジェクトの nf_conntrack 開発者が招待され、課題等を議論した。さらに、これまでの nf_conntrack や IPv6 パケットフィルタ機能のメンテナンス等の貢献から、同開発者が Netfilter Project のコアメンバに招かれた。これにより、USAGI プロジェクトはパケットフィルタに関する開発をよりスムーズに行える体制をとることができた。

2.2.5 現在の状況

nf_conntrack は Linux 2.6.15 に取り込まれた。ユーザは、nf_conntrack が追跡したフローの状態を利用する IPv4 のフィルタルールを設定可能となった。ただし、IPv6 に対応した同様のパケットフィルタ機能は未だ導入されていない。それらは 2.6.16 で導入される予定である。導入を遅らせた理由は、別々に実装された IPv4 パケットフィルタと IPv6 パケットフィルタが 2.6.16 で統合される予定であり、2.6.15 で上記の機能を導入してもすぐに変更が必要になるためである。

2.2.6 今後の予定

nf_conntrack が Linux のメインラインカーネルに採用されたことで、開発は一段落したが、今後も USAGI プロジェクトは nf_conntrack のメンテナンスを続ける予定である。

2.3 品質向上活動

2.3.1 品質向上活動について

USAGI プロジェクトは、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、より高機能なコードを提供してきた。本プロジェクトの活動の成果物は Linux コミュニティに広く受け入れられている。たとえば IPv6 プロトコルスタックの改良コードは、既にその多くがメインラインカーネルに採り入れられている。また、USAGI プロジェクトが提供するコードを用いた組み込みデバイス等の製品も登場している。

広く Linux コミュニティにコードが受け入れられるようになった現在、更なる開発活動に加え、既に開発したコードの品質維持・向上も重要である。この品質維持・向上に対する USAGI プロジェクトの活動をここに報告する。

2.3.2 TAHI Automatic Running System

2.3.2.1 TAHI Conformance Test を利用した

Regression Test システムの開発

USAGI プロジェクトは Linux を中心により良い 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高品質な IPv6 プロトコルスタックのコードの提供を継続して行っている。

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数多く取り込まれている。この日々変更されるカーネルコードに対し、メンテナ及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、IPv6 プロトコルスタックをテストする Regression Test システムを開発した。IPv6 プロトコルスタックのテスト自身には、TAHI プロジェクトの TAHI IPv6 Conformance Test Suite (<http://www.tahi.org>) を利用している。この自動テスト実行システムにより、IPv6 プロトコルスタックに副作用が生じた際に速やかに発見および対処・修正ができる。

本システムは一般公開されており、<http://testlab.linux-ipv6.org> にて閲覧可能である。

2.3.2.2 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。本システムでは、IPv6 Ready Logo Phase-1 Conformance Test を始め TAHI プロジェクトがリリースしている IPv6 Conformance Test Tool を利用した Conformance Test を使用することが可能である。また、一度ビルドしたカーネルに対し、複数の Conformance Test を順次実行することも可能である。

システムの流れを図 2.6 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

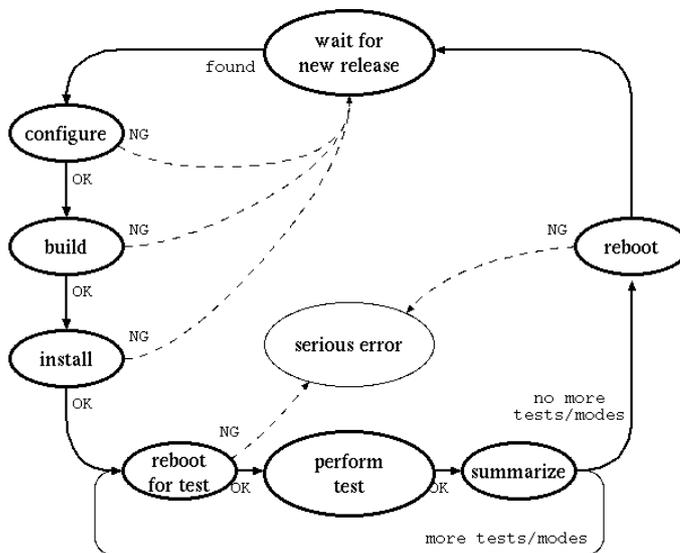


図 2.6. システムの流れ

受ける。待ち受けるリリース対象は、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、および毎夜開発コードレポジトリより切り出してリリースされる git バージョン（2005 年 6 月以前は 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をダウンロードし、ビルドを開始する。カーネルの configure、build、install は順次行われ、各過程のログは保管される。ログを参照することにより、ビルドエラーなどが生じた際にも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被テスト対象を再起動する。この再起動はビルドしたカーネルが正しく起動できるのが確認するために行われる。再起動時のログもテスト結果にあわせて保管される。再起動に失敗した場合、システムは一旦停止し、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 TAHI Conformance Test を実行する。テスト終了後に結果を保管する際、システムは既に行った実行結果との比較一覧を作成し別途保管する。この比較一覧の確認により、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。

TAHI Conformance Test が 1 つ終了した後、別の Conformance Test を実行するよう設定されてい

れば、システムは被テスト対象を再起動し、次に設定された Conformance Test を実行する。一旦システムを再起動するのは、前回のテスト時にカーネルが異常停止していないか検出するためである。再起動時のログは保管される。

カーネルへのテストが全て終わると、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、被テスト対象は再起動し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。

2.3.2.3 データの収集および収集データの確認方法

システムは、テストの結果、前結果との比較一覧、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、カーネルビルド等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。収集したデータはウェブブラウザから確認することができる。ウェブブラウザからのアクセス例を図 2.7 に示す。また結果比較の表示例を図 2.8 に示す。

2.3.2.4 2005 年度の成果

2004 年 12 月時点では、本システムは TAHI Test Suite のうちの IPv6 Ready Logo Phase-1 Conformance Test しか実行することができなかった。2005 年度には複数のテストを順次実行することを可能とした。現在以下のテストをシステム上で走らせ、結果を公開している。

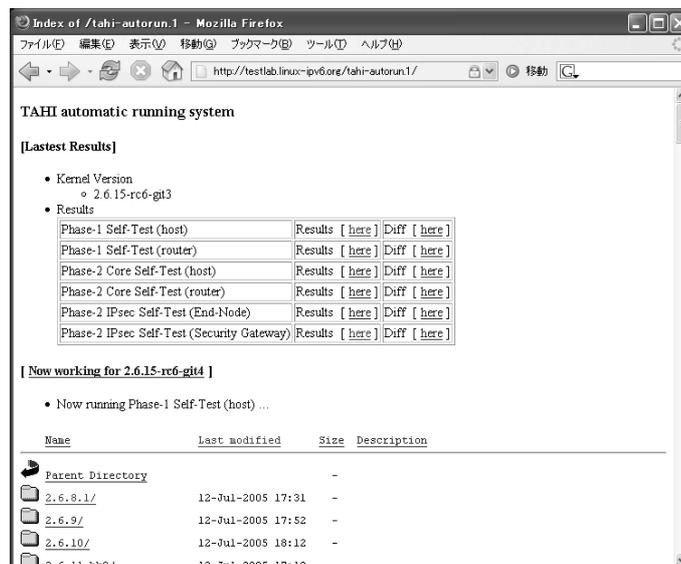


図 2.7. ウェブブラウザよりのアクセス例

Index	2.6.14	2.6.15-rr6	2.6.15-rr6-gi1	2.6.15-rr6-gi2
Initializing the NUT				
Initialization (please ignore)				
5 Test for ICMPv6 Specification (RFC 2463)				
5.1.1 Transmitting Echo Requests				
Transmitting Echo Requests	PASS	PASS	PASS	PASS
5.1.2 Replying to Echo Requests				
A Request sent to Link-Local address	PASS	PASS	PASS	PASS
B Request sent to global address	FAIL	PASS	PASS	PASS
C Request sent to multicast address	PASS	PASS	PASS	PASS
5.1.3 Destination Unreachable Message Generation				
C.Port Unreachable - Link-Local Address	PASS	PASS	PASS	PASS
D.Port Unreachable - Global Address	PASS	PASS	PASS	PASS
5.1.6 Erroneous Header Field (Parameter Problem Generation)				
... Fragment Test Preparation	PASS	PASS	PASS	PASS
Erroneous Payload	PASS	PASS	PASS	PASS
5.1.7 Unrecognized Next Header (Parameter Problem Generation)				
Next Header 128	PASS	PASS	PASS	PASS
5.1.8 Unknown Informational Message Type				
Message Type 255	PASS	PASS	PASS	PASS
5.1.10 Error Condition With Multicast Destination				
A UDP Port Unreachable	PASS	FAIL	PASS	PASS

図 2.8. 結果比較の表示例

- IPv6 Ready Logo Phase-1 Conformance Test (Host)
- IPv6 Ready Logo Phase-1 Conformance Test (Router)
- IPv6 Ready Logo Phase-2 Conformance Test (Host)
- IPv6 Ready Logo Phase-2 Conformance Test (Router)
- IPsec for IPv6 Ready Logo Phase-2 Conformance Test (End-node)
- IPsec for IPv6 Ready Logo Phase-2 Conformance Test (Security Gateway)

2005 年度、本システムにより kernel のバグの混入や TAHI Conformance Test のバグを 10 件近く発見することができた。また、差分作成方式の改善により、変更点の確認と追跡がより容易となった。

今後、更なる改善・利便性向上において以下の項目を検討している。

- Mobile IPv6 に関する試験のサポート
- TAHI 相互接続性試験ツール (vel) を含む、より多様な試験の自動化に関する検討
- より多様な環境 (CPU Architecture、SMP)、設定での試験の実施
- より効率的な問題追跡手法に関する検討

2.3.3 IPv6 Ready Logo 取得

2.3.3.1 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。2005 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、実運用性を主眼としより高度な機能を確認対象とした Phase-2 認証が行われている。Phase-2 認証においては IPv6 の中心機能に加えて IPsec、MIPv6 などの認証も行われている。

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、国際的接続認証の取得に努めている。

2.3.3.2 2005 年度の成果

USAGI プロジェクトは 2005 年 1 月、TAHI プロジェクトが主催する IPv6 Interoperability Test Event に参加し、Linux のメインラインカーネルバージョン 2.6、および本プロジェクトが独自にメインラインのバージョン 2.4 および 2.6 を拡張して一般に公開しているカーネルにおいて IPv6 Ready Logo Phase-1 認証を取得した。認証を受けたメインラインカーネルのバージョンは 2.6.11-rc2 である。また、USAGI が拡張して提供しているカーネルは USAGI STABLE RELEASE 6 として一般に公開している。

第3章 論文リスト

本年度の USAGI プロジェクトの論文リストを以下に示す。

- 小堺康之, 吉藤英明, 江崎浩, 村井純. Linux における IP バージョン非依存な接続トラッキングの設計と実装. 電子情報通信学会. 第21回 NS・IN 研究会, 104 (信学技報 IN2004-248), pp. 29-33, 2005.
- Hideaki Yoshifuji. Linux Is Now IPv6 Ready. In Proceedings of Linux Symposium 2005, pp. 283-290, 2005.

