第 XI 部

SCTP および DCCP に関する研究開発

第 11 部
SCTP および DCCP に関する研究開発

第 1 章　はじめに

SCTP ワーキンググループは、SCTP（Stream Control Transmission Protocol）や DCCP（Datagram Congestion Control Protocol）などの次世代トランスポートプロトコルに関する研究を中心に活動を行っている。本年度の主な活動内容は以下の通りである。

- SCTP 相互接続検証の開催と参加
- Windows 用 SCTP ドライバの開発
- SCTP の ADD-IP 拡張に対する改善の提案
- SCTP による高速ハンドオーバ手法の研究

以下、各章において詳細な内容を報告する

第 2 章　SCTP 相互接続検証の開催と参加

2.1 目的と概要

SCTP Interop は、SCTP の実装を行っているメーカや団体がそれぞれの実装を持ち寄り、相互に接続や通信を行えることを確認するイベントである。第 1 回は 2000 年にドイツで開催され、今年で 9 回目となる。今回は初のアジアでの開催となった。

日時は WIDE の活動と IETF の活動を考慮し、2007 年 8 月 19 日–25 日となった。

ちょうど京都では五山送り火の日に近かったこともあり、準備に関わった者はこの送り火を見たあと作業の開始となった。

会場は京都大学学術情報メディアセンター北館 3F を利用した。従来の開催規模から考え、60 人収容の会議室を確保した。6 回路分の電源の増設が必要となった。

参加者の募集は 6 月 1 日より開始した。この種のイベントにおいては毎回継続して参加する団体が多いことを考慮し、これまでの参加者に対して重点的にアナウンスした。その他、SCTP 実装を行うものが質問する ML でアナウンスした。

その結果、参加者募集期間が短期間であったにも関わらず、8 種類の実装に対して 15 人が集まった。

以下に詳細を述べる。

2.2 実装

集まった実装は以下の通りである。それぞれの実装に 1 ないし 2 名ずつの参加があった。

参加組織	OS
Kyoto University	Windows
Cisco Systems Inc.	FreeBSD and IOS
Muster University of Applied Sciences	Mac OS X
University of Duisburg-Essen	userland (sctplib)
Emerson	Linux
Sun Microsystems, Inc.	Solaris
HP	Linux
Continuous Computing India Pvt. Ltd	Linux
Korea National University of Education	—

SCTP ワーキンググループとしては、現在開発中の Windows 用実装である SCTP Driver で検証に参加した。なお、今回は実装を持ち寄ることは出来なかったが、韓国からの参加者があった。アジア地域における SCTP への興味が高まっていることの現れと考えられる。

2.3 ネットワーク

各実装に対して 2 つの独立したネットワークを提供し、IPv4、IPv6 のアドレスを付与した。実装ごとに割り当てられるネットワークは、それぞれ異なるブロードキャストネットワークであり、各実装が別の実装と通信を行う場合、ルータを経由するようになっている。ルータには、FreeBSD 6.2R を利用した。また、遅延がある場合や、パケットロスが発生する場合において正常な通信が行えるかどうかを試

験する際には、ルータ上において、dummynet 機能を利用した。

2.4 テスト項目

テスト項目は Basic テストと Advanced テストに分けられて、各実装は Basic テストをクオリファイすることが第一段階の目標とされた。Basic テストは、互いの実装が Initiator または Responder となり、SCTP アソシエーションの確立ができるかどうか、互いにデータの転送・受信が行えるのかどうかを確認するものである。SCTP Driver を含め、あらゆる実装の組み合わせにおいて Basic テストをクリアすることができた。

Advanced テストは、1 Gbps のネットワークを用いた際のデータ転送のパフォーマンスや、遅延やパケットロスのあるネットワークでの SCTP の挙動をテストするものである。SCTP Driver は、実装の初期段階であったため、Advanced テストには参加することはできなかった。

2.5 まとめ

SCTP が多くの OS に対して標準的に導入されるようになっている中、今回のような相互接続性試験においてどの実装もほぼ満点のスコアで試験を終えることが出来た点はすばらしい成果であったと考える。

また、今回 SCTP ワーキンググループとして新たに持ち込んだ Windows 向け実装である SCTP Driver も既存の実装と同程度以上のスコアを出すことが出来た。

今後もこのような相互接続性試験は毎年ないしは隔年の頻度にて開催が続けられるとのことであるので、今後も継続して参加し SCTP の普及に貢献していきたい。

第 3 章　Windows 用 SCTP ドライバの開発

3.1 SCTP ドライバの意義

SCTP ワーキンググループは、トランスポート層に関する研究を行うために設立されたワーキンググループであり、TCP や UDP に代わる次世代のトランスポートプロトコルである SCTP（Stream Control Transmission Protocol[155]）に関する調査研究や機能拡張を中心に活動しており、各種 OS に対する SCTP の実装活動にも積極的に参加してきた。また、一昨年の春と秋の WIDE 合宿において SCTP を利用した移動通信の実験を行い、マルチホームなど SCTP の新機能の有用性の検証も行っている。これらの実験によって、SCTP の利用にはメリットがあることが明らかとなった。

各種 OS で利用可能な SCTP 実装は、Linux においてバージョン 2.6 から本家カーネルのソースコードに含まれるようになったのを皮切りに、Solaris、AIX、HP-UX などの UNIX 系 OS において標準で搭載されるようになった。SCTP ワーキンググループが実装活動に積極的にかかわってきた FreeBSD 上の SCTP 実装も、7.0 RELEASE 以降のリリースにおいてはデフォルトで有効にされており、インストール直後からとくにカーネルの再構築等を行わなくとも利用できる。コンシューマ向け OS である Mac OS X においても、FreeBSD 上の SCTP 実装をベースとした SCTP カーネルモジュールが作成され、公開されている[1]。

一方、コンシューマ向け PC の大多数において使われている Windows 系 OS においては、Mac OS X 上の SCTP カーネルモジュールのような SCTP 実装はこれまで存在しなかった。WIDE 合宿で実施した実験では、ユーザランドにおいて動作する sctplib[169] を利用して SOCKS トレンスレータを開発し、実験に用いた。SOCKS トランスレータによって、Windows 系 OS においても SCTP の有用性を体感することはできたが、SOCKS を経由した通信となるためオーバヘッドが大きく、また、マルチホーム機能以外のマルチストリーム機能やデータ受信・送信の際のメッセージ境界の維持といった SCTP の機能を利用することができなかった。

SCTP ドライバは、SOCKS トランスレータとは異なり、TCP や UDP などのドライバと同様に、カーネル空間で動作し、SCTP パケットを直接処理するため、複数のアプリケーションに対して同時に SCTP をサービスできるだけでなく、TCP から SCTP へ変換するといったオーバヘッドが存在しない。また、Windows 上の BSD ソケット実装である Winsock に対応しており、TCP や UDP と類似したプログラミ

1　各種 OS の SCTP 実装の一覧については、`http://www.sctp.org/implementations.html` を参照。

ングスタイルで SCTP を利用できる。

以下では、SCTP ドライバの動作概要と機能を紹介する。

3.2 SCTP ドライバの動作概要

PC 向け Windows 系 OS において TCP/IP の機能は、カーネル空間で動作しパケットの送受信やタイマの管理などを行う TCP/IP ドライバと、アプリケーションと TCP/IP ドライバとの間を中継するライブラリ（ws2_32.dll）から構成される。ws2_32.dll は、アプリケーションから呼び出されるソケット API（Winsock API）と TCP/IP ドライバとの間の関連付けを自由に追加・変更できるように設計されている。SCTP ドライバでは、この機能を利用して、SCTP のソケット API を実装した。

SCTP ドライバは、TCP/IP ドライバとは独立した別個のドライバとなっており、カーネル空間で動作する。SCTP パケットの送受信には、TCP/IP ドライバが実装している RawIP 機能を用いた。ホストに入ってきた SCTP パケット（IPv4 または IPv6）は、TCP/IP ドライバによって IP フラグメントの再構成などの Layer 3 での処理が行われた後、SCTP ドライバによって処理される。また、SCTP ドライバによる SCTP パケットの送出は、TCP/IP ドライバを通じて（近接探索やフラグメント化などが行われた後）行われる。

SCTP ドライバが行う SCTP パケットの処理やタイマの処理は、SCTP.org で公開されている SCTP 実装を移植して実装した[2]。また、中継ライブラリとドライバの間のインターフェースに用いるために FreeBSD のソケットインターフェースも同時に移植した。

3.3 SCTP ドライバの機能

SCTP ドライバは、動的にシステムに組み込むことが可能であり、再起動することなく、導入・除去することができる。

SCTP ドライバは SCTP.org の SCTP 実装を移植したものであるため、ADDIP 拡張などの最新の SCTP 拡張に対応している。また、SCTP 実装の移植に際してソースコードに加えた変更はごくわずかであり、その変更も SCTP.org の SCTP 実装に取り込まれているため、SCTP.org 上の SCTP 実装に新たな SCTP 拡張が実装されれば、ごくわずかな作業量で SCTP ドライバにおいても同じ機能を利用できるようになる。

アプリケーションから用いる API については、現在標準化が行われている SCTP 用のソケット API[156] に対応しており、TCP 互換の one-to-one ソケットだけでなく、新たに定義された one-to-many ソケットも利用できる。Winsock API には大規模サーバアプリケーションの実装において性能改善に寄与する Overlapped I/O という POSIX Asynchronous I/O（AIO）と類似の機能があるが、SCTP ドライバにおいても、Overlapped I/O を利用できる。

3.4 今後の開発予定

SCTP 機能の実装（移植）はほぼ完了しており、API も基本的な機能については実装が完了している。今後は動作の安定化及び最適化を図る予定である。また、SCTP ワーキンググループにおいて行っている SCTP 拡張の SCTP ドライバへの導入を積極的に行い、SCTP ワーキンググループでの研究活動にも役立てていきたい。

第 4 章　SCTP の ADD-IP 拡張に対する改善の提案

SCTP の ADD-IP 拡張は、アソシエーション確立後に各エンドポイントにおいてアドレスの動的な追加や削除を行うことを可能にするものであり、中継ノードに依存しないハンドオーバーなどに利用することができる。しかしながら、以前の ADD-IP 拡張では、同時に 2 つ以上のリクエストを送ることができず、アドレスの変化が生じたときに通信を再開できなくなる可能性があった。そのため我々は、2006 年度において、同時に 2 つ以上のリクエストの送信を可能とする改善を提案した。その後、この改善は ADD-IP 拡張の提案に取り込まれ、2007 年度末には改善を含んだ形で、ADD-IP 拡張の提案が RFC 5061 となった。

2　この実装のソースコードの取得方法は、http://lakerest.net/pipermail/sctp-coders/2007-August/005312.html を参照。

第 5 章　SCTP による高速ハンドオーバ手法の研究

SCTP の Multi-home 機能及び Dynamic Address Reconfiguration 機能（ADD-IP 拡張）を用いたモビリティサポートの研究開発を行った。本年度はシングルホームの端末が異なる IP プレフィックスのネットワークに移動した際、多くの遅延が発生する問題を解決した。

シングルホームの端末が異なるネットワークに接続する際は、デバイスのオーバヘッド及びレイヤ 2、レイヤ 3 の接続確立の間、ネットワーク到達性が失われる。また、端末が無線の電波強度が著しく弱いエリアやどの無線基地局にも接続できないエリアを通過する際にもネットワーク到達性が失われる。その間には双方のノードにおいてパケットロスが発生し、SCTP においては、それらのパケットは Retransmission Time Out（RTO）にもとづいて再送される。そのため、ネットワーク到達性が復旧した後も次の RTO までパケットが再送されず余計な遅延が発生する。また、RTO の値は 1 秒から 60 秒の間で expire ごとに倍になるため、ネットワーク到達性の喪失時間が長くなるにつれて到達性復旧後の遅延が長くなる可能性がある。本研究ではこの問題を解決するため、SCTP Association 内でアドレスの変更を通信相手に伝達する ASCONF Chunk のやりとりをトリガーにした再送機構を提案した。研究成果は、論文誌（`wide-paper-sctp-micchie-ieicetran2007-00.txt`）、国際カンファレンス（`wide-paper-sctp-micchie-iscc2007-00.txt`）にて発表した。実装は FreeBSD の SCTP スタックおよび Windows の SCTP ドライバに取りこまれている。来年度は、標準化活動及び他の OS への実装を進めていく予定である。

第 6 章　まとめ

本年度は、昨年度の実験結果をベースにして主に Windows ドライバの実装及びモビリティサポートの強化に関する研究を行った。来年度はそれらをより進める他にも、マルチホーム機能による複数パスの同時利用や、他のレイヤの情報を用いたモビリティサポートの改善に関する研究開発も行う予定である。