

第 XV 部

SCTP および DCCP に 関する研究開発

第 15 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、次世代のトランスポートプロトコル：SCTP (Stream Control Transmission Protocol) や DCCP (Datagram Congestion Control Protocol) などに関する研究活動を行うことを目的として 2004 年 12 月に作成された。以下に本年度の活動内容について述べる。

第 2 章 複数アドレスをサポートするトランスポートプロトコル (SCTP) のための Socket API 拡張

2.1 概要

本報告書は、プロトコル独立なプログラミングスタイルにより記述された TCP アプリケーションに一切のコード変更を加えることなく、複数アドレスをサポートするトランスポートプロトコル (SCTP) にも対応させることができるように、`getaddrinfo(3)`、`bind(2)`、`connect(2)` などのソケット API を拡張する手法を提案する。これらの API に対するアドレス情報の受け渡しにおいて、複数のアドレスをまとめて取り扱うための表記方法およびソケットファミリーを定義する。

2.2 目的

従来、TCP による接続は両端に各 1 つずつの IP アドレスを利用するため、アプリケーションが一旦接続を確立した後に、その IP アドレスが利用するネットワークに障害が生じたり、端末が移動して IP アドレスが変更されたりすることに対応することは困難であった。SCTP は、その特徴の 1 つとして、複数アドレスを利用した接続 (アソシエーション) を確立することができる。このように 1 つの接続において複数のアドレスを利用することで、クライア

ント側はアドレスの変更が可能となり、サーバ側では IP アドレスに依存した障害に対する耐性が増すといったメリットがある。

しかしながら、このような SCTP の特徴を活かすためには SCTP 独自のコーディングスタイルを使う必要があり、この点が SCTP 普及に向けての大きな障害となっていた。

そこで本提案では、プロトコル独立なプログラミングスタイルで記述された TCP アプリケーションのコーディングスタイルを変更することなく、ライブラリ (API) を拡張することのみで既存のアプリケーションを SCTP 対応となるようにすることで、複数アドレスを利用する SCTP の特長を活かしたアプリケーションを増やし、SCTP の普及を促すことを目的とする。

2.3 現状

プロトコル独立なプログラミングスタイルをとる TCP のクライアントアプリケーションは、`bind(2)` を明示的に呼ばず内部的に `0.0.0.0` または `::` に対して `bind(2)` を行い、次いで `getaddrinfo(3)` で得たアドレスリストの各アドレスに対して順次 `connect(2)` を行う。また、同スタイルをとる TCP のサーバアプリケーションの多くは `0.0.0.0` または `::` に `bind(2)` した上で `listen(2)` を行っている。ところで、SCTP ソケット API のインターネットドラフトでは `0.0.0.0` または `::` に `bind(2)` すると、そのホストが持つすべてのアドレスを利用してアソシエーションを確立しようとするのが規定されている。すなわち、プロトコル独立なプログラミングスタイルをとる TCP アプリケーションの多くは現状のままでも、プロトコルを SCTP に変更するだけで複数アドレスによるアソシエーションを利用することが可能となっている。

2.4 問題点

複数のアドレスを有するサーバで TCP によるサービスを行う場合で、`0.0.0.0` または `::` で `bind(2)` を行いすべてのアドレスでサービスするのではなく、運用上の理由によりいくつかの限定されたアドレス

でのみサービスをするためには、個々のアドレスに順次 bind(2) を行い複数のソケットで待ち受けるといったプログラミングスタイルをとることになる。しかし、サーバ側で SCTP による複数アドレスによるアソシエーションを利用するためには、各アドレスごとに bind をするのではなく、複数のアドレスをまとめて bind する必要がある。そのために sctp 独自の API を利用する必要があり、コーディングスタイルの変更が必要であった。

また、クライアント側における TCP のコーディングスタイルでは getaddrinfo(3) により複数のアドレスが得られたとき、それらのアドレスに対して接続が成功するまで順次 connect(2) を呼び出す。しかし、この手順ではアドレスを順次試す前に Retry や Timeout 待ちを繰り返すことになるため、接続が成功するまでに何度も不必要な INIT パケットの送出行うだけでなく、接続までに時間がかかるという問題もあり非効率である。

2.5 提案

上で述べた問題点を解決するため、複数のアドレスをまとめて取り扱う際の表記方法、および新たなソケットファミリーを定義し、getaddrinfo(3) や bind(2) などのソケット API で利用できるようにする。このようにソケット API を拡張することで、複数のアドレスを示す文字列を適切な書式で記述し、従来ホスト名として指定していた文字列の代わりに指定することで、ライブラリの差し替えのみで TCP アプリケーションが SCTP による複数アドレスによるアソシエーション対応となる。

まず、次節で述べる複数アドレスをまとめて取り扱う表記方法は、従来の FQDN の表記と同様にそのまま getaddrinfo(3) に渡すことで、複数アドレスをまとめて取り扱うソケットファミリーのデータ構造へと変換される。bind(2) や connect(2) はこのデータ構造を通じて複数のアドレスをまとめて受け取ることで、指定されたアドレス群だけを使って bind(2) することや、connect(2) が一度に複数のアドレスに INIT を送出したり次々と異なるアドレスに INIT を再送することが可能となる。

また、複数のアドレスを扱うアプリケーションやトランスポートプロトコルにおいては、複数のアドレスを 1 つの塊や 1 つのソケットとして扱うことを明示的に示唆する表記法を定義することにより、こ

種のアプリケーションの開発時の作法を標準化することができるものとする。

2.6 複数アドレスをまとめて取り扱う表記方法

- FQDN や IPv4 や IPv6 などの直アドレスを ‘,’ で区切って {} で囲ったもの
- FQDN が含まれている場合、DNS を利用して IPv4 や IPv6 などのアドレスに変換する
例：
(www.example.com が 192.168.0.1 10.0.0.1 を返す場合)
- {www.example.com} は
 {192.168.0.1,10.0.0.1} と等価
- {www.example.com,127.0.0.1,::1} は
 {192.168.0.1,10.0.0.1,127.0.0.1,::1}
と等価
- {www.example.com,fe80::1%fxp0} のように linklocal address も含めてよい
- 複数アドレスをまとめて取り扱うアドレスファミリー

- アドレスファミリーの名前は AF_BUNDLE とする
- ソケットアドレス構造体は次のような形をとる

```
struct sockaddr_bundle {
    unsigned char    sb_len;
    sa_family_t      sb_family;
    socklen_t        sb_addrlen;
    struct sockaddr  *sb_addr;
    struct sockaddr_bundle *sb_next;
};
```

sockaddr_bundle は getaddrinfo(3) で用いられている addrinfo に類似した Linked List の構造をしており、sb_addr に実際のアドレスの構造体へのポインタ、sb_addrlen にそのアドレス構造体の長さ、sb_next に次のアドレスに関連付けられた sockaddr_bundle のポインタが設定される。

第 3 章 SCTP 実装の開発

現在 SCTP ワーキンググループでは CISCO の Randall Stewart らが進めている SCTP 参照実装の

開発プロジェクトに参加し、実装の品質向上に貢献している。

第 4 章 DCCP 実装の開発

現在 SCTP ワーキンググループでは KAME をベースに DCCP 実装の開発を進めている。現在 IETF DCCP ワーキンググループでアクティブに開発が行われている実装として注目されているものは 2 つあり、本ワーキンググループによる実装はその一つである。実装の状況と linux 実装との相互接続試験の結果などについて、64th IETF において発表を行った。

なお本発表で使用したプレゼンテーション資料 (DCCP Implementation on BSDs) は報告書添付の DVD-ROM に収録してあるので参照されたい。

第 5 章 おわりに

本年度の SCTP ワーキンググループの活動は開発に重点が置かれたものとなってしまった感があるが、その成果として実装の成熟度が向上しさまざまな実験が可能となった。この成果を生かし、来年度は開発だけではなく、技術の普及や研究のアウトプットを出すことに注力していく予定である。

