

アドレス情報の変更通知を集約して再送できるようにする mSCTP 拡張

丸山 伸[†] 小塚 真啓^{††} 中村 素典^{†††} 岡部 寿男^{†††}

[†] 京都大学大学院情報学研究科

^{††} 京都大学大学院法学研究科

^{†††} 京都大学学術情報メディアセンター

京都市左京区吉田本町

E-mail: [†]marushin@net.ist.i.kyoto-u.ac.jp, ^{††}kozuka@wide.ad.jp, ^{†††}{motonori,okabe}@media.kyoto-u.ac.jp

あらまし 複数のアドレスを同時に利用して通信を行う SCTP による接続に対して、動的に IP アドレスの増減を行うための拡張である mSCTP は、トランスポート層におけるモビリティ実現技術として知られている。しかし mSCTP は仕様上の制約によりアドレス増減の通知を連続して行うことが出来ないため、ハンドオーバーにおいて本来は必要ないはずの長い時間が必要となったり、相互に通信可能なアドレスを取得したにもかかわらず接続を維持できない場合が発生したりする。本研究においては、アドレス増減の通知の再送を複数個まとめて送信できるように mSCTP を拡張する手法を示し、この変更により複数のアドレスが連続して増減した際においても、通信できるアドレス対が存在する限り実用的な時間内で確実にハンドオーバーできるようになることを示す。

キーワード SCTP, 複数アドレス, モビリティ, ハンドオーバー, msctp

An Extension of mSCTP for cumulative retransmission of Address Configuration Change Chunks

Shin MARUYAMA[†], Masahiro KOZUKA^{††}, Motonori NAKAMURA^{†††}, and Yasuo OKABE^{†††}

[†] Graduate School of Informatics, Kyoto University

^{††} Graduate School of Law, Kyoto University

^{†††} Academic Center for Computing and Media Studies, Kyoto University

Yoshida-honmachi, Sakyo-ku, Kyoto 606-8501, Japan

E-mail: [†]marushin@net.ist.i.kyoto-u.ac.jp, ^{††}kozuka@wide.ad.jp, ^{†††}{motonori,okabe}@media.kyoto-u.ac.jp

Abstract SCTP is a transport protocol which handles multiple IP addresses for each end, and mSCTP, which is an extension of SCTP, supports the dynamic modification of IP addresses of SCTP association after it is established. mSCTP is a realization technology of transport layer IP mobility. When the IP addresses change one by one with a certain interval, the handover of mSCTP is carried out within a practical time. When multiple addresses are deleted and added continuously, there is a case that an unnecessary delay is caused because of the feature of mSCTP definition.

We show in this research that when the IP addresses are modified continuously, as long as there is an address pair which can communicate exists, an extension of mSCTP to bundle multiple retransmission of packets which carry the information of IP addresses change together enables to handover within a practical time.

Key words SCTP, multiple addresses, mobility, handover, msctp

1. はじめに

SCTP “Stream Control Transmission Protocol (RFC2960)” [1] は、通信を行う両端末がそれぞれ複数のアドレスを持つ時に、それらのアドレスを適宜切り替えて通信を行うトランスポート層のプロトコルである。SCTP はその接続 (アソシエ

ション) の確立時に両端末が持つアドレスを相互に交換し、お互いの持つアドレスをアソシエーションに割当てて保持する。端末の移動やインタフェースの追加や削除などによりアソシエーションの確立後にアドレスが増減することがあるが、このような際にアソシエーションのアドレスを増減させるためには SCTP の ADD-IP 拡張 [2] と呼ばれる提案が利用されてい

る。この SCTP の ADD-IP 拡張を用いてアドレスの増減に対応し、トランスポート層によるモビリティを実現したものは mSCTP [3], [4] と呼ばれており、理想的な環境においては端末のアドレス変更が生じた際に端末間の RTT (Round Trip Time) の 4 倍程度の時間でアドレス変更の情報を端末間で共有し、実用的な時間内において IP Handover を実現できることが知られている。[5]~[7]

ところで、この mSCTP におけるアドレス変更は、“Address Configuration (ASCONF) Chunk” と呼ばれるアドレス変更通知と、それを受信した通信相手が返送する ASCONF-ACK チャンクとをやり取りすることにより同期を取りつつ行われる。しかし、連続してアドレスの変更が発生した場合においても、ネットワークの輻輳を避け、またアソシエーションに属するアドレス情報の同期を取るために、送出した ASCONF チャンクに対する ASCONF-ACK チャンクを受けとっていない間は、別の ASCONF チャンクの送出は禁止されている。

そのため、ある ASCONF チャンクの送受信に失敗した上で、さらに別のアドレス更新が生じた時には、ハンドオーバーに長い時間が必要になったり、アソシエーションが切断されるといった不具合が生じることがある。

本論文においては mSCTP で利用される ASCONF チャンクを拡張し、一度に複数の ASCONF チャンクをまとめて送信出来るようにすることで、アドレスの増減が連続して発生した時にも実用上十分に短い時間でアドレスの増減に対応し、適切にハンドオーバーできるようになることを示す。

以下、2 章では mSCTP の挙動に関する既存の研究と、複数のアドレスが連続して増減した時の mSCTP の挙動の問題点について述べ、3 章ではこの問題点を解決するための mSCTP の拡張提案について述べる。4 章では本提案の有効性を示し、5 章で結論を述べる。

2. これまでの研究

SCTP は通信を行う両端に対して複数の IP アドレスを割当て、それらの複数のアドレスを切り替えつつ通信を行うことで、冗長性を持った接続を行うことが出来るトランスポート層のプロトコルである。

2.1 SCTP における複数アドレスの扱い

SCTP の TCP に対する最大の相異点である「1 つの接続において複数のアドレスを利用できる」という特徴を活かすために、SCTP では特定の IP 同士により接続を開始しようとした場合においても、自動的に両端末が持つ利用可能な IP アドレスの情報を相互に交換し、得られた全てのアドレスを利用しようとする。すなわち、通信はこの時に交換されたアドレスのいずれかを、それぞれソース IP アドレスおよび宛先 IP アドレスとして利用する。また、これらの通信相手のアドレスのうちどのアドレスが利用できるかを確認するために HeartBeat パケットが相互に交換され、確認の取れたアドレスは Confirmed Addresses と呼ばれる。このようにして得られた Confirmed Addresses の中から 1 つを選択してプライマリパスと呼び、データを送信す

る際にはプライマリパスの示すアドレスに対して送られる。もしプライマリパスに対するデータ送信に対して適切な ACK が返ってこない場合には、一定回数の再送のあと、別の宛先を選択して再送し、そのアドレスをプライマリパスとする。

2.2 SCTP の ADD-IP 拡張

SCTP による接続が確立したあとに生じたアドレスの増減に対応するために、アソシエーションに属するアドレスの増減を行うための ADD-IP 拡張 [2] と呼ばれている提案がある。この提案ではアドレスの増減が生じた際に“Address Configuration (ASCONF) チャンク”にアドレスの増減に関する情報を含めて送出する。通信相手は ASCONF を受信すると、アドレスの増減を行なった後に ASCONF-ACK を送り返すが、ASCONF の送信者は ASCONF-ACK を受信するまでアソシエーションに属するアドレスの増減を行ってならないと定義されている。

各 ASCONF チャンクには Serial-ID と呼ばれる 32bit のパラメータがあり、これにより各 ASCONF は一意に順序づけされている。また、一旦送出した ASCONF の内容は、ASCONF の再送時にもその内容を変更してはいけない。また、ASCONF の発信者はそれに対応する ASCONF-ACK を受信するまでは、次の(別の) ASCONF を送信してはいけないことになっている。この制限は、アドレスの増減は ASCONF の情報 1 つずつ相互に確認しながら処理することを意味し、続けて 2 つ以上の ASCONF が生成されることを考慮していない。^(注1)

2.3 mSCTP

この SCTP の ADD-IP 拡張を用いてアドレスの変化や増減に対応し、トランスポート層によるモビリティを実現したものが mSCTP (Mobile SCTP) [3], [4] である。mSCTP では、端末のアドレスの増減が発生する度に ASCONF チャンクと ASCONF-ACK チャンクのやりとりを行い、アソシエーションに属するアドレス情報を更新する。

一般に、どのようなアドレスの増減が生じた場合においても、アソシエーション内に通信を行えるアドレス対が 1 組でも存在すれば、そのアドレス対を用いて ASCONF チャンク等のやりとりを行うことが期待されている。また、もし最初に利用したアドレス対が通信できない状況であったとしても、ASCONF の再送を行う際に、ソースアドレス及び宛先アドレスを適切に選

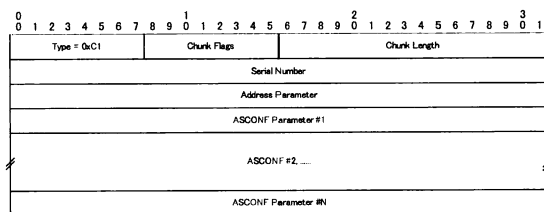


図 1 ASCONF チャンクの構造

Fig. 1 Data structure of ASCONF chunk

(注1) : 1 つの ASCONF チャンクの中にアドレスの増減に関する情報を複数個列挙することは出来る

扱しなおすことにより、再送タイマーの待ち時間が必要となるも適切にアドレス情報の交換を行える。

ところで、無線インタフェース 1 つだけを利用しているローミングなどのように、アドレスの更新時に一旦すべてのアドレスがなくなる場合においても通信を継続できるようにするため、mSCTP は Address Parameter を利用する。^(注2) まず ADD-IP 拡張は、その端末の持つ最後のアドレスを削除してはいけないと規定している。そして、新たにアドレスを得た際に、まずは新たなアドレスをアソシエーションに加え、その後に古いアドレスを消すべきとしている。また、新しいアドレスから送出される ASCONF チャンクのソースアドレスは新たに増えたアドレスであるが、ASCONF を受信した際にどのアソシエーションのものであるかを識別するために従来の Association に含まれていたアドレスのうちのいずれか 1 つを Address Parameter として ASCONF チャンクに含めることとしている。ASCONF を受信した際にはこの情報を元にどのアソシエーションに関する ASCONF なのかを判断する。このように、アドレスが 1 つずつ順次増減する場合には、ADD-IP 拡張を用いたモビリティ技術である mSCTP は概ね適切に動作する。

2.4 mSCTP の問題点

既存の mSCTP の仕様はアドレスは 1 つずつ追加や削除され、そのアドレスの増減情報は主に他のアドレスを利用して即座に peer に通知されることを前提としている。

ところが、以下に示すようなシナリオを検討すると、通信できるアドレス対が存在するにも関わらず、アソシエーションを維持できなくなる。

2.4.1 Dual Stack 環境でアドレスが順次消える時

- 端末につけられている全てのアドレスが順次利用できなくなり、複数の独立したアドレス情報変更イベントが発生し、それらが順次短い時間の間に複数の ASCONF チャンクとして送出される時。

このような状況はある無線インタフェースが IPv4 及び IPv6 の両方を持っているような時に、まず一方のアドレスが無くなり、新しいアドレスを得る前にもう一つのアドレスが無くなるような時に発生する。すなわち、これら両方のアドレスが割当てられたインタフェースが利用できなくなり、Router Advertisement(RA) および DHCP の Lease Time が切れて順次 1 つずつ無効となるような時にこの状況は発生する。

まず、1 つ目のアドレスが無くなる時には、これは最後 1 つのアドレスではないため、端末はこのアドレスを消すために ASCONF を送出しようとする。しかし、この時には既にソースアドレスとして使えるアドレスは存在しないため、ASCONF の送出は失敗し、再送フェイズに入る。そして、2 つ目のアドレスは最後 1 つのアドレスであるため、端末は ASCONF を送出しない。

一旦このような状況になってしまうと、1 つ目の ASCONF に対する ASCONF-ACK を受信するまでは、この端末は例え新し

く別のアドレスを得たとしても新たに ASCONF を送出することが出来ない。そして、ASCONF の再送が行われるまでの長い時間、端末は待つ必要がある。

さらに悪いことに、たとえ再送が行われたとしても、この ASCONF チャンクは通信相手には無視されることになってしまう。なぜならこの端末が新たに得たアドレスは、アソシエーションに含まれているものではないため、このアドレス自体を ADD-IP により追加しようとする時にのみこのアドレスを ASCONF チャンクのソースアドレスとして利用しても良いことになっているからである。そして、以前からある他のアドレスはすでに接続性が失われているため、ソースアドレスとして利用すべきではない。

結果として長い待ち時間のあとに、アソシエーションは CLOSE ステートに移行する。

この問題に対する対策として、1 つ目のアドレスの削除が発生した時に即座に ASCONF を送出しようとするのではなく一定時間を置いてから送出しようとする事で、複数のアドレス更新を 1 つの ASCONF にまとめてしまう手法が考えられる。このようにすれば 2 つのアドレスをまとめて消す ASCONF は端末が持つ最後のアドレスとなるため送出することが出来ず、新たなアドレスを得た時に、まずそのアドレスを追加し、次いで 2 つのアドレスをまとめて消すという手順を取ることが出来るため、アソシエーションを継続することが出来る。

しかし、この手法はどの程度の時間を待てば適切なかを判断する手法がないばかりか、アドレスの更新にかかる時間が延びるという問題もあるので、この方法は採用し難い。

2.4.2 分断されたネットワークのアドレスを得た時

- アソシエーションに有効なアドレスがない状態から、新たに得たアドレスが通信相手に到達性のない無効なものであった時。

このような状況は、端末の持つ最後の有効な IP アドレスが無線が途絶えた等の理由により無効となったあとに、また別のアドレスを新たに得たにもかかわらず、この新しいアドレスが以下のような状態のものであった時に生じる。

- 分断されたネットワークのものであった時
- NAT の下のものであった時
- ファイアウォールなどでフィルタをかけられている時
- 通信相手が IPv4 のみ対応であるにもかかわらず、新たに得たアドレスが IPv6 であるなどの時

ADD-IP 拡張はこのようなアドレス更新をサポートするために、Address Parameter が導入されている。一般に SCTP パケットのソースアドレスはアソシエーションのいずれかのアドレスでなければならないが、新しくアドレスを得てそのアドレスをアソシエーションに追加しようとする時に限り、Address Parameter にアソシエーションに含まれているアドレスの 1 つを含めた上でそのアドレスをソースアドレスとして利用できる。

このような状況が発生したが新しく得たアドレスが NAT の下のネットワークであつたりフィルタがかかっていたり、通信相手に到達出来ない独立したネットワークであつたりした時に

(注2)：この手法に含まれるセキュリティ上のリスクは、Auth-chunk [8] と呼ばれる手法でパケット毎のハッシュを付与することで回避できる。

は、当然 ASCONF の送出は失敗し、ASCONF の再送フェイズに入る。

しかしながら、この接続性のない新しいアドレスは端末の持つ唯一のアドレスであるため、再送はし続けることとなり、結果としてアソシエーションは CLOSED ステートに移行する。

一度この状況に陥ると、例えば端末がこの後に有効なアドレスを得たとしても、そのアドレスはアソシエーションに含まれているものではないため、ASCONF の再送時にソースアドレスとして利用することができない。

この問題に対する対策として、アドレスの変更が発生した即座にアドレス情報の変更をしようと ASCONF の送出を行うのではなく、事前にそのアドレスが相手と通信を行う上で適切かどうかを確認するという手法が考えられる。この手法はこの問題を部分的に解決はするが、ハンドオーバーにかかる時間が増加するだけでなく、もちろん IP アドレスだけから得られる情報で確実な判断をすることは出来ないため、これは適切な解決策とは言えない。

3. 提案手法

これまでの ADD-IP 拡張においては、未解決の ASCONF (すでに送出したにもかかわらず、まだ対応する ASCONF-ACK を受け取っていない ASCONF) が存在するときには、新たな ASCONF を送出してはいけないという制限があった。

そこで ASCONF チャンクの送出に関するルールを既存のものから大きく変更することなく拡張し、複数の ASCONF チャンクをまとめて送出できるようにする手法を提案する。

3.1 集約された ASCONF チャンク

これまでの ASCONF チャンクは図 1 のような構造によるものであるが、この構造は一切変更しない。

その上で、ASCONF の送出ルールを以下のように変更する。

- 端末はアドレスの増減等により新たな ASCONF が作成された時には、それを未解決の ASCONF の末尾に連結した上で、いつでも送出することが出来る。その際には、未解決の ASCONF の再送のために起動されていたタイマーは停止され、新たに送出される ASCONF に対してタイマーが新たに起動される。

- 連結される ASCONF は、その Serial 番号の順に連結して並べられていなければならない。

- 未解決の ASCONF が存在する時に新たに ASCONF チャンクを作成する時には、未解決の ASCONF の全てが成功したと仮定しなければならない。このルールは新たに作られる ASCONF チャンクの Address Parameter フィールド (アソシエーションの識別に用いられる) に入れられるアドレスは、全ての未解決の ASCONF が適切に処理されたと仮定した際にそのアソシエーションに含まれるアドレスのいずれかでなければならないことを意味している。

図 2 は Serial N 番の ASCONF を送ろうとした際に、未解決の ASCONF が 3 つ (すなわち、Serial は N-3 から N-1) 存在し

た場合のものである。

これらの 4 つの ASCONF チャンクは 1 つのメッセージに分割されずに入れられてなければならない、それらの ASCONF チャンクの Serial 番号は上に示す図のように昇順になってなければならない。

それぞれの ASCONF チャンクのフォーマットは ADD-IP 拡張のものと同じである。N-3 番から N-1 番の ASCONF チャンクはそれが最初に作成された時から変更してはならない。

Serial N 番の ASCONF を作成するときには ADD-IP 拡張のフォーマットと同じ手順で作成される。ただし、Address Parameter に 0 以外の値をセットするときには、それ以前の ASCONF の全てが成功したと仮定した時にアソシエーションに含まれるアドレスのいずれかでなければならない。

3.2 集約された ASCONF チャンクに対する ASCONF-ACK

集約された ASCONF チャンクを受信した時には、その中に含まれるそれぞれの ASCONF チャンクが順次届いたと仮定して順次処理し、そのそれぞれに対して ASCONF-ACK を返す。

これらの ASCONF-ACK チャンクは、集約された ASCONF チャンクのソースアドレスに対して返送されなければならない。

それぞれの ASCONF-ACK チャンクは 1 つのパケットの中に Serial の昇順に並べられなければならない。

図 3 は、これまでの ASCONF-ACK チャンクの例である。

図 4 は、本提案に基づく集約された ASCONF-ACK チャンクの構造の例である。

3.3 ASCONF 送信側の変更点

ASCONF 送信側の手順は、ADD-IP 拡張に対して以下のように変更される。

なお、以下の説明において、N 番のシリアルを持つ AS-

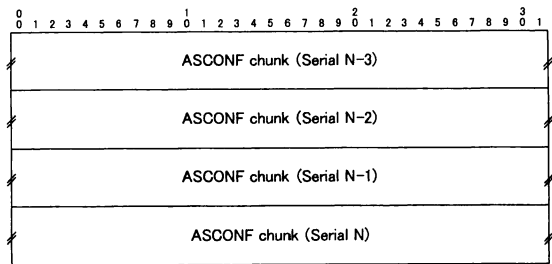


図 2 Cumulative ASCONF チャンクの構造

Fig. 2 Data structure of Cumulative ASCONF chunk

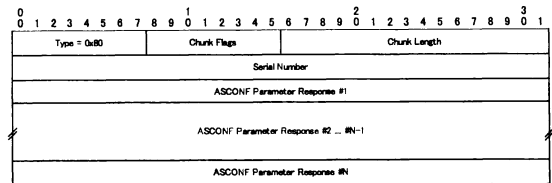


図 3 これまでの ASCONF-ACK チャンクの構造

Fig. 3 Data structure of ASCONF-ACK chunk

CONF チャンクのことを ASCONF(N) と書くことにする。また、ASCONF-ACK(N) も同様の意味である。

(1) ASCONF チャンクを送出する際には、ASCONF 出力キューの中にまだ ASCONF-ACK 応答を得ていないものがあるかどうかを調べる。もし何も残っていない時には、従来の ADD-IP 拡張の手順に従う。逆に、1 つないし複数の ASCONF チャンクが未解決で残されていた時には、ASCONF チャンクを束ねて送信するために、次の手順に進む。

(2) 集約された ASCONF チャンクを作る時には、全ての未解決の ASCONF チャンクは 1 つのパケットに束ねられなければならない。

それらの未解決の ASCONF チャンクに対して書換えを行ってはならず、各 ASCONF が作られた時そのままの状態で保持されなければならない。

未解決の ASCONF チャンクは集約された ASCONF チャンクとして 1 つのパケットに連結されなければならない。

(3) これらの ASCONF チャンクはシリアル番号の昇順に並べなければならない。

(4) 集約された ASCONF チャンクは MTU のサイズを越えるべきではない。もし ASCONF を送信しようとした際に集約された ASCONF のサイズが MTU のサイズを越えることがわかったときには、ASCONF-ACK が到着するのを待つべきである。

(5) 集約された ASCONF チャンクの Address Parameter は 0 にセットされるか、ないしは、既に出した ASCONF チャンクがすべて適切に反映されたと仮定した際にアソシエーションに含まれるアドレスの 1 つにセットされなければならない。

(6) ASCONF パケットは一旦送出されたら、それ移行に修正してはならない。

(7) 集約された ASCONF チャンクを送出する際には、ASCONF チャンクの再送タイマーは再起動されるべきである。

3.4 ASCONF 受信側の変更点

ASCONF を受信した側は、従来の ADD-IP 拡張にあるのと概ね同じ振舞いをする。ASCONF チャンクを受信したときには、端末は以下の手順に従い、そのチャンクの属すべきアソシエーションを決定する。

L1) ソースアドレスと宛先アドレスとポート番号を用いてア

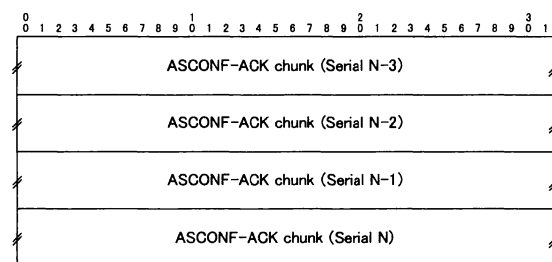


図 4 Cumulative ASCONF-ACK チャンクの構造

Fig. 4 Data structure of Cumulative ASCONF-ACK chunk

ソシエーションを識別する。(これは RFC2960[RFC2960] で規定されているのと同じ手法である) もし見つかった時には L4 に進む。

L2) もしアソシエーションが見つからなかった時には、アドレスパラメータの中に含まれるアドレスと SCTP 共通ヘッダに含まれるポート番号とを組み合わせアソシエーションを識別する。もし見つかった時には L4 に進む。

L2-ext) もし 2 つ以上の ASCONF チャンクが含まれている時には、そのそれぞれの ASCONF チャンクの Address Parameter に含まれるアドレスを用いてアソシエーションを識別しようとする。もし見つかった時には L4 に進む。

L3) もし、L1, L2, L2-ext のいずれにおいてもアソシエーションが見つからない時には、RFC2960 [RFC2960] に定義される “Out Of The Blue” チャンクとして扱う。

L4) RFC2960 [RFC2960] に定義される通常の規則通り、SCTP verification タグの正当性を確認する。

アソシエーションの識別と正当性の確認とが終了したら、以下の手順に従い ASCONF チャンクを順次処理する。

M1) もし 最初の ASCONF チャンクのシリアル番号が ('Peer-Serial-Number'+1) に等しい時には、端末はキャッシュされた ASCONF-ACK 応答を全て消去した上で、ADD-IP 拡張に記述されている手順通りに処理する。処理が終了したら、同じ ASCONF が再送されてきた時 (すなわち ASCONF-ACK がロストした時) に備えて ASCONF-ACK 応答の内容をキャッシュする。
M2) もし、シリアル番号が ('Peer-Serial-Number'+1) よりも小さい時には、単にその ASCONF を飛ばして次の ASCONF に処理を移す。ただし、その場合においてもキャッシュされた ASCONF-ACK の情報から、同じ Serial 番号を持つ ASCONF-ACK を応答パケットに含める。

M3) 続いて、ASCONF を 1 つずつ処理する。もしシリアル番号が ('Peer-Serial-Number'+1) よりも小さい時には、上と同じく、単にその ASCONF を飛ばして応答パケットをこれまでに保存している ASCONF-ACK を応答パケットに追加する。

M4) シリアル番号が次に来るはずの番号に等しくなった時には、ADD-IP 拡張にあるのと同じ手順でその ASCONF チャンクを処理した上で、その処理が終わると、ASCONF-ACK を応答パケットに追加し、その複製をキャッシュする。(後に同じ ASCONF-ACK を再送する必要が生じることがあるから) もし ASCONF を処理する時にエラーが発生した時には、端末は残りの ASCONF に対する処理を中断し、応答パケットを即座に返さなければならない。

M5) 集約された ASCONF チャンクに含まれている全ての ASCONF チャンクに対する処理が終了したとき、集約された ASCONF チャンクに含まれるすべてのシリアルに対する ASCONF-ACK チャンクを 1 つの応答にまとめて送り返す。この ASCONF-ACK チャンクの宛先は、集約された ASCONF チャンクのソースアドレスでなければならない。

M6) 集約された ASCONF を処理する過程において、もし集

約された ASOCNF-ACK パケットのサイズが返送パスの PMTU を越えそうになった時には、それ以上 ASCONF-ACK を応答パケットに追加するのをやめるが、ASCONF チャンクの処理自体は継続しなければならない。そして、全ての ASCONF-ACK 応答は複製されキャッシュされる。ASCONF の送信者は、後に応答を受け取っていない ASCONF を再送してくることになるが、その時にはキャッシュされた ASCONF-ACK 情報を送りかえしても良い。

3.5 ASCONF-ACK 受信側の変更点

集約された ASCONF-ACK パケットの受信者は、それぞれの ASCONF-ACK を順次 ADD-IP 拡張に基いて処理しなければならない。

4. 考 察

本研究では既存の SCTP の ADD-IP 拡張をベースに、複数の ASCONF チャンクを一度にまとめて送出できるようにした。

この変更により ASCONF のサイズは増えることになるが、ASCONF パケットの数自体は増えないため、従来のものに比して輻輳を引き起こすものではない。

ASCONF-ACK パケットも、複数の ASCONF-ACK チャンクを 1 つのパケットに集約して返送することになるため、サイズは増えることになるがパケットの数は増えない。

また、この提案においては集約された ASCONF を受信した時にはその応答として 集約された ASCONF-ACK を返送する際には、それらの複製をキャッシュする必要があるため、従来よりもメモリーを消費することになる。(従来も ASCONF-ACK 1 つ分のキャッシュを行っている)しかし、このサイズは最大でも MTU のサイズ程度であり、著しくメモリ消費量が増えるものではない。

5. お わ り に

この研究では、SCTP のアソシエーションに含まれる IP アドレスを動的に更新する際に利用される ADD-IP 拡張における「アドレスの変更通知 (ASCONF) を連続して送出してはならない」という制限を解消する手法を提案した。

まず、ADD-IP 拡張における ASCONF の送受信手順に存在する ASCONF を連続して送出してはならない制限が存在することを示し、この制限が原因となってハンドオーバーが失敗する事例を明らかにした。次に既存の ASCONF の処理手順の枠組を崩すことなく、この問題を解決するために、複数の ASCONF をまとめて 1 つのチャンクとして送信する手法、及び、そのようなチャンクを処理する手順を詳細に示した。

この改良を行うことで、SCTP のモビリティへの利用である mSCTP において、ネットワーク間の移動をより安定してかつ確実に行えるようになった。

今後改良すべき点として、本提案に対応する端末と対応しない端末との間で問題なく通信できるようにするという点が挙げられる。これにはアソシエーションの開始時に情報交換をするといった手法が考えられる。

なお、SCTP をモビリティにさらに積極的に利用するためには、ハンドオーバーにかかる時間をさらに短縮することが必要となる。そのために、SCTP がトランスポート層以外の各種情報とも連携を取り、NextHop 選択やソースアドレスに配慮しつつ複数のインタフェースを活用することや、無線の強度等を活用してネットワークが切断されるよりも前の段階でハンドオーバーを行うといった方式も模索する必要があるものと考えられる。

謝辞 本論文の作成にあたっては、Cisco Systems 株式会社の Randall Stewart 氏をはじめ、IETF における SCTP 研究チーム、また KAME チーム等、多くの SCTP 研究者に議論に参加して頂きました。この場をお借りして心よりの感謝を申し上げます。

文 献

- [1] R. R. Stewart, Q. Xie, K. Morneault, C. Sharp, H. J. Schwarzbauer, T. Taylor, I. Rytina, M. Kalla, L. Zhang and V. Paxson, "Stream Control Transmission Protocol", RFC2960, <http://www.ietf.org/rfc/rfc2960.txt>, October 2000.
- [2] "Stream Control Transmission Protocol (SCTP) Dynamic Address Reconfiguration", <http://www.ietf.org/internet-drafts/draft-ietf-tsvwg-addip-sctp-14.txt>.
- [3] M. Riegel and M. Tuexen, "Mobile SCTP", <http://www.ietf.org/internet-drafts/draft-riegel-tuexen-mobile-sctp-06.txt>, March 2006.
- [4] S. J. Koh, Q. Xie, S. D. Park, "Mobile SCTP (mSCTP) for IP Handover Support", <http://www.ietf.org/internet-drafts/draft-sjkoh-msctp-01.txt>, October 2005.
- [5] D. P. Kim, J. S. Ha, S. T. Kim and S. J. Koh, "Use of SCTP for IP Handover Support", Proceedings of the Fourth Annual ACIS International Conference on Computer and Information Science (ICIS'05), 2005.
- [6] S. J. Koh, M. J. Chang and M. Lee, "mSCTP for Soft Handover in Transport Layer", IEEE Communications Letter, Vol. 8, No. 3, March 2004.
- [7] A. Kelly, P. Perry and J. Murphy, "A Modified SCTP Handover Scheme for Real Time Traffic", HETNETs Working Conference, Ilkley, England, July 2003.
- [8] M. Tuexen, R. Stewart, P. Lei and E. Rescorla, "Authenticated Chunks for Stream Control Transmission Protocol (SCTP)", draft-ietf-tsvwg-sctp-auth-02.txt, <http://www.ietf.org/internet-drafts/draft-ietf-tsvwg-sctp-auth-02.txt>, March 2006.