SIPによるネットワークアプリケーション間のセッション管理の応用について

菊池 裕明[*]　井関 文一[**]　山本 亮介[**]

概要

本論文では，SIPを利用したネットワークアプリケーション間のセッション管理の応用について議論する．ネットワークアプリケーション間のセッション管理にSIPを用いることにより，サーバが固定のIPアドレスを持たない場合（移動体を含む）やハイブリッドP2P型通信でセッションの確立を行う場合に有用性が向上する事を示す．

１．はじめに

近年のインターネットの急激な発展に伴って，その利用形態も多様化し始めており，従来の固定IPを持ったサーバに対してサービスを要求する，クライアント-サーバ(C/S)型の通信の他にP2P型の通信や移動体間の通信などが重要な位置を占めつつある．

ただし，これらの通信手段においては，状況によっては接続の相手を特定する（セッションを確立する）ことが困難な場合がある．特に移動体間などの通信のようにIPアドレスが短時間に動的に変化する場合などでは，DNSなどを利用した従来の方式では，リアルタイムにIPアドレスの変化を検知することは困難であり，そのような状況下で接続を確立することは非常に難しくなる．

２．Session Initiation Protocol

我々は，前述の問題に対して，SIPによるネットワークアプリケーション間のセッション管理を提案する．

SIPという用語はインターネット電話やVoIPの代名詞のように使用される場合があるが，正式名称の Session Initiation Protocol が示すように，アプリケーション間のセッションを管理するプロトコルにすぎない．実際 VoIPなどでは，音声の転送にはSIPとは独立した RTP(Real-time Transport Protocol)，RTCP(RTP Control Protocol) が使用される．

ただし，実際問題として現状では SIPは音声や映像などのストリーム転送用に使用される場合がほとんどで，一般的なアプリケーションで使用される例はあまり無い．

しかしながら，ネットワークアプリケーション間のセッション管理にSIPを利用することによりSIPの持つ様々な機能をそのまま利用することができ，少ないコストで柔軟なシステムを構築することが可能となる．

例えば，SIPの持つ，SIP URIまたは（電話）番号による通信相手の特定，代替サーバへの転送機能，ユーザ認証機能などをそのまま利用する事が可能である．

＊ 三井情報株式会社

＊＊ 東京情報大学　情報システム学科

３．ＳＩＰによるセッション管理

ネットワークアプリケーション間のセッション管理にSIPを利用した場合，非常に利便性が高くなる例として，次のような局面が考えられる．

1) ハイブリッドP2P型で仲介サーバを必要とする場合に仲介サーバをSIPで構成する．
2) C/Sシステムに於いて，サーバのIPアドレスが短い時間で動的に変化する場合．
3) C/Sシステムに於いて，代替サーバへの転送を透過的に行う場合．

これらの局面においてSIPを用いた場合の利点としては，先にも述べたが，次のような点が挙げられる．

1) 新たにセッション管理用のプログラム（プロトコル）を開発する必要がない．
2) SIPの持つ様々な機能（例えば端末の登録や着信の転送機能，ユーザ認証機能など）をそのまま利用可能である．

我々はネットワークアプリケーション間のセッション管理にSIPを使用する場合の手順として以下ようなものを考案した（図１）．

ます，アプリケーション間の接続に必要な情報はSDP(Session Description Protocol)のメディア記述のタイプ a レコード（メディア属性行）に保存する事とする[1][2]．

つまり，SIPのINVITE命令を受け取った（③）マシンは通信用のネットワークアプリケーションを起動し（④），接続に必要な情報を SDPのメディア記述の a レコードに格納して OKレスポンスを返す（⑤）．OK レスポンスを受け取った（⑥）もう一方のマシンはSDPの a レコードの内容に従って対になる通信用のネットワークアプリケーションを起動し（⑦），相手と通信を開始する（⑧）．

SDPのm，o 及びc レコードを使用せずに a レコードを使用するのは，SIP サーバによっては，Proxyを行う必要上，m，o，c レコードの内容を書き換えてしまう物も存在するためである．

この手順の場合，通信を行うマシン間でSDPを適切に扱うプログラムを作成すれば，SIPサーバには一切手を加える必要はない．

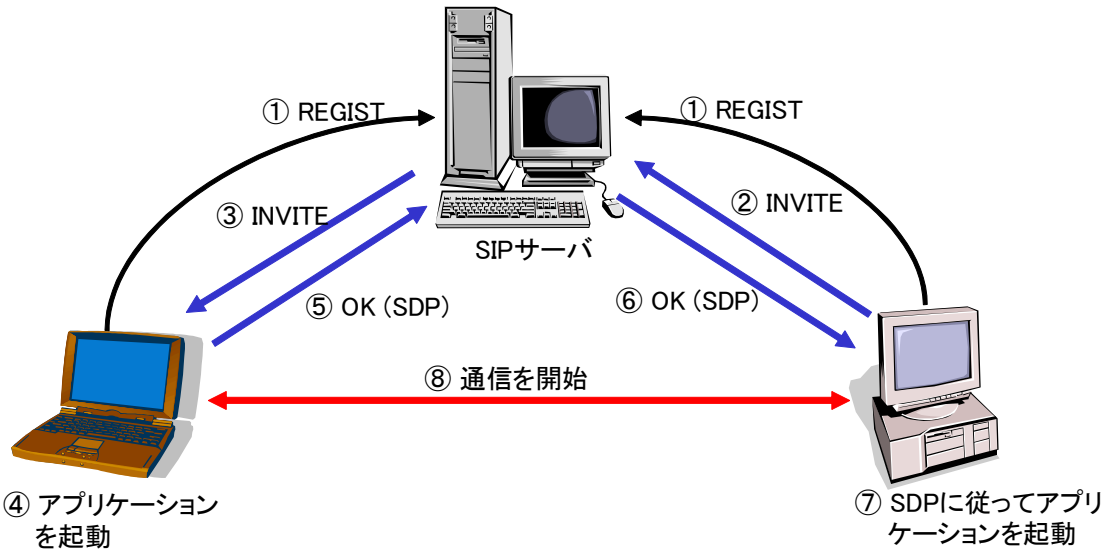

図１ SIPによるセッションの開始．（INVITE,OKレスポンス間のTrying及びRingingは省略）

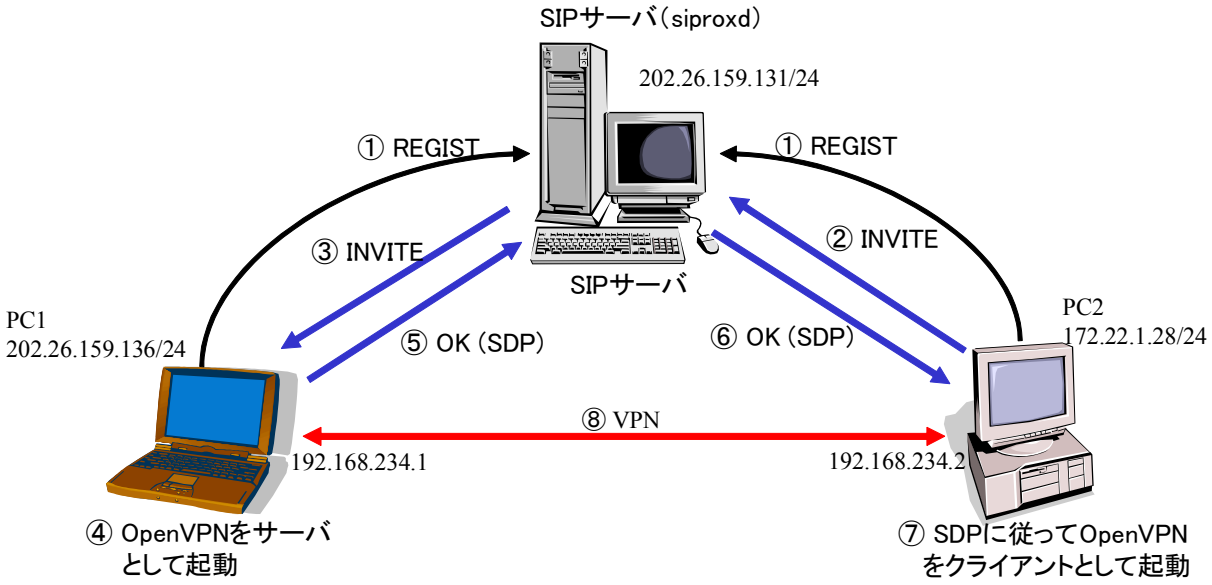

図２ SIPによるOpenVPNセッションの開始．（INVITE,OKレスポンス間のTrying及びRingingは省略）

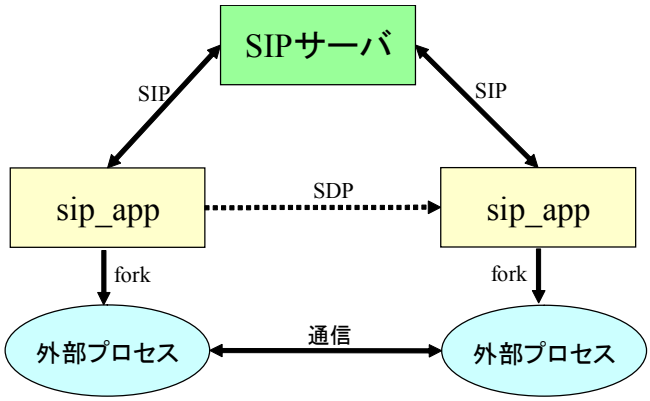

図３ sip_app プログラム

レコードタイプ	値
v	0
o	2500 1169538046 1169538046 IN IP4 202.26.159.131
s	-
t	0 0
m	application/vpn 7084 OpenVPN 0
c	IN IP4 202.26.159.131
a	IP4:202.26.159.136
a	PORT:8000
a	VPN_LOCAL_ADDR:192.168.234.1
a	VPN_REMOTE_ADDR:192.168.234.2

表１ 図2の⑥でのSDPのレコード内容

４．OpenVPN による応用例

SIPによるアプリケーション間のセッション管理の応用例として OpenVPN[3]による VPNの形成を示す（図2)[4].

SIPサーバにsiproxd[5]を使用しているのは,他のSIPサーバと比較してデバッグ情報が取得し易いためであり，他の SIP サーバを使用しても何ら問題はない.

この実験を行うために我々は，SIP クライアントと外部アプリケーション起動の機能を持つ sip_app[4] プログラムの開発を行った.

sip_app は GNU の oSIP2[6]， eXosip2[7]ライブラリを利用して，SIP クライアント機能，SDP 制御機能を実現し，チャイルドプロセスとして外部プロセスを起動することが可能である（図３）.

図２の PC1 では，予め OpenVPN を外部プロセスとして登録し，サーバモードで sip_app を起動しておく．sip_app は起動と同時に SIP のレジストラに登録を行う（①).

PC2 では，クライアントモードで sip_app を起動する．PC2 の sip_app はレジストラに登録（①）を行った後に INVITE 命令で PC1 の sip_app を呼び出す（②，③)．PC1 の呼び出しは，PC1 の IP アドレスではなく，レジストされた PC1 の（電話）番号で行われる.

INVITE 命令を受け取った PC1 の sip_app は登録された OpenVPN をサーバモードで起動し（④)，接続に必要な情報を SDP の a レコードに格納した後，200 OK のレスポンスと共に PC2 へ転送する（⑤，⑥)．なお，今回の実験では OpenVPN はルーティングモードで起動している.

PC2 の sip_app は返された OK レスポンス中の SDP の a レコードから必要な情報を取り出し，OpenVPN をクライアントとして起動する（⑦)．PC2 で起動された OpenVPN は指定されたパラメータに従って，PC1 の OpenVPN と通信を行い，PC1-PC2 間に VPN を形成する（⑧).

sip_app はOpenVPNのプロセスをチャイルドプロセスとして監視しており,OpenVPNプロセスが終了した場合,BYE命令を発行してセッションの切断を行う.

表１に図２の⑥の時点での SDP のレコードのタイプを示す．m レコードでメディアタイプとして application/vpn，プロトコルとして OpenVPN を指定しているが，これは今回の実験に当たり独自に定義した部分である（アプリケーションの実際の動作には影響しない)．m レコードのポート番号（7084)，o,c レコードの IP アドレスは，

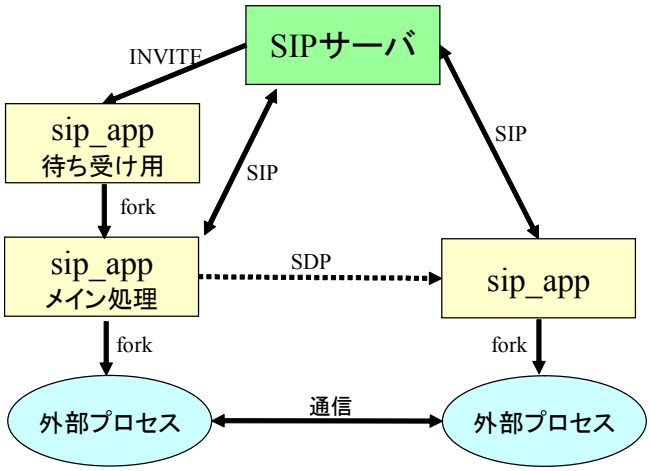

図４　SIPによるスーパーデーモン

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP
202.26.159.131:5060;branch=z9hG4bKd5494712271eafdca196759bbcd82500
Via: SIP/2.0/UDP
202.26.159.131:5060;branch=z9hG4bKdf1ab35628fe284df07a0549b85b5d31
Via: SIP/2.0/UDP 172.22.1.28:5060;rport;branch=z9hG4bK1068437359
Record-Route:<sip:siproxd@202.26.159.131:5060;lr>
From:<sip:2501@202.26.159.131>;tag=1297171609
To:<sip:2500@202.26.159.131>;tag=1988095920
Call-ID: 1815073903@172.22.1.28
CSeq: 20 INVITE
Contact:<sip:2500@202.26.159.136:5060>
User-Agent: SIP for APP b1 rev.45
Allow: INVITE, ACK, OPTIONS, CANCEL, BYE, SUBSCRIBE, NOTIFY,MESSAGE,INFO, REFER, UPDATE
Content-Type: application/sdp
Content-Length: 245

v=0
o=2500 1169538046 1169538046 IN IP4 202.26.159.136
s=-
t=0 0
m=application/vpn 8000 OpenVPN 0
k=DH:crypt code
c=IN IP4 202.26.159.136
a=IP4:202.26.159.136
a=PORT:8000
a=VPN_LOCAL_ADDR:192.168.234.1
a=VPN_REMOTE_ADDR:192.168.234.2
```

メッセージ 1　図2の⑤でPC1が発信したOKレスポンスのSIPメッセージ

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP 172.22.1.28:5060;rport;branch=z9hG4bK1068437359
Record-Route:<sip:siproxd@202.26.159.131:5060;lr>
From:<sip:2501@202.26.159.131>;tag=1297171609
To:<sip:2500@202.26.159.131>;tag=1988095920
Call-ID: 1815073903@172.22.1.28
CSeq: 20 INVITE
Contact:<sip:2500@202.26.159.131>
User-agent: SIP for APP b1 rev.45
Allow: INVITE, ACK, OPTIONS, CANCEL, BYE, SUBSCRIBE, NOTIFY, MESSAGE,INFO, REFER, UPDATE
Content-Type: application/sdp
Content-Length: 245

v=0
o=2500 1169538046 1169538046 IN IP4 202.26.159.131
s=-
t=0 0
m=application/vpn 7084 OpenVPN 0
c=IN IP4 202.26.159.131
k=DH:crypt code
a=IP4:202.26.159.136
a=PORT:8000
a=VPN_LOCAL_ADDR:192.168.234.1
a=VPN_REMOTE_ADDR:192.168.234.2
```

メッセージ２　図2の⑥でPC2が受信したOKレスポンスのSIPメッセージ

siproxd によって書き換えられている．

発信側の sip_app による外部プロセスの起動を制御するものは a レコードであり，表１の IP4 は PC1 の IP アドレス，PORT は PC1 の OpenVPN の接続待ち受け用ポート番号，VPN_LOCAL_ADDR は PC1 の VPN 用 IP アドレス，VPN_REMOTE_ADDR は PC2 の VPN 用 IP アドレスをそれぞれ表す．これらも今回の実験のために独自に拡張した部分である．

参考までに図 2 の⑤，⑥の時点での SIP メッセージの内容をメッセージ 1，2 に示す．なお，メッセージ中では SDP の k レコードを指定しているが，この実験では暗号化は行っていない．

これらのシステムにより，PC2 が PC1 の IP アドレスを知らなくても，何の問題も無く PC1-PC2 間で VPN のセッションの確立，動作及び切断が行える事を確認している．

５．SIP によるスーパーデーモン

前述の OpenVPN の応用例では，sip_app は一度に一つの外部プロセス（OpenVPN）しか起動できないが，着信側（サーバ側）において sip_app のチャイルドプロセスを起動し，そのチャイルドプロセスから外部プログラムを起動すれば，接続を多重化することも可能である（図４）．すなわち，一般の TCP での接続のように，着信側（サーバ側）の sip_app が INVITE 命令を受け取った時点でチャイルドプロセスを起動し，以後の SIP のやり取りはそのチャイルドプロセスが担当し，親プロセスは次の INVITE 命令を待ち受けるようにすることも可能である．

このようにすれば，SIP によるスーパーデーモンを構成することも可能だが，SDP の処理をアプリケーション毎に定義しなければならないため設定がかなり複雑になることが予想される．しかしながら非常に有用なシステムとなる可能性があるので，今後の研究の課題としたい．

６．まとめと今後の課題

SIP によるアプリケーション間のセッション管理の手法を提案し，OpenVPN を使用した応用例を示した．SIP を通常のアプリケーション間のセッション管理に利用した場合，SIP の持つ機能をそのまま利用できるので，非常に有用性が高いと考えられる．

ただし，起動するアプリケーション毎に SDP の a レコードを規定し起動手順を設定しなければならため，それらの規格化が必要である．さらに，SIP の持つ問題点など（NAT 越えなど）もそのまま受け継ぐため，それらの問題の解決も今後の課題である．

また SIP によるスーパーデーモンの実用化の研究も今後の課題としたい．

参考文献

[1] ソフトフロント，「エッセンシャル SIP」，日経 BP 社，2005.

[2] 澤田 拓也，池田 徹，木下 岳人， 西澤 哲夫，「実践 SIP 詳解テキスト」，リックテレコム，2005.

[3] OpenVPN, http://openvpn.net/

[4] 菊池 裕明，「ハイブリッド P2P 型 VPN に関する研究」，東京情報大学大学院修士論文，Mar. 2007.

[5] siproxd, http://sipoxd.sourceforge.net/

[6] oSIP2, http://www.gnu.org/software/osip/

[7] eXosip2, http://www.antisip.com/as/en/products.php