

クライアントサイドモジュール型 SIP-B2BUA ミドルウェア 「SIPProp」の開発

1. 背景及び目的

「IP 電話が相互接続できない！」これは IP 電話に使用されている SIP プロトコルの問題で、SIP の仕様が曖昧であるため、実装者による解釈の相違が大きく、相互接続性が低いというのが現状である。

一方で、SIP には電話だけでなく、色々な機器に適用できるという利点があり、またオープンな規格であり、誰でも使用できることから今後さらに有望な技術である。

最近では、Asterisk というオープンソースの IP-PBX の登場により、個人レベルで SIP サーバを持っている人が出てくるなど SIP も急速に普及し始めている。しかしながら、例えば、プロバイダの提供する IP 電話サービスと接続しようとした場合、SIP 通信でエラーが出て接続できないなど、まだまだ課題が山積みである。

本プロジェクトでは、世界的に問題となっている VoIP (SIP-UA) 相互接続問題を解決するミドルウェア SIPProp を開発し、SIP が本来目指していたセッション層を汎用プロトコル化することで、世界中の SIP サーバを相互接続することを目指す。これにより、キャリア版ではない、インターネット版の IMS を実現する。

2. プロジェクトの概要

本プロジェクトでは、世界的に問題となっている VoIP (SIP-UA) 相互接続問題を解決するミドルウェア SIPProp を開発した。

従来、VoIP 事業者ごとに独立したクローズドなネットワーク上に構成されてきたため、同じベンダーの SIP 端末同士、SIP サーバの推奨する特定端末同士であればつながるが、それ以外ではうまくつながらないということが多々あった。これを SIPProp では、SIP が本来目指していたセッション層を汎用プロトコル化することで解決し、相互接続を実現した。

具体的には、「SIP の方言」(SIP の仕様が曖昧であり、実装者の解釈の相違による実装の差異)を吸収する B2BUA (Back To Back User Agent)、SIP メッセージのヘッダ部分の変換によるシーケンスの違いの吸収、通信相手に合わせてモジュール化した UASet を整備し、これらをクライアントサイドで動作させることで、実現した。

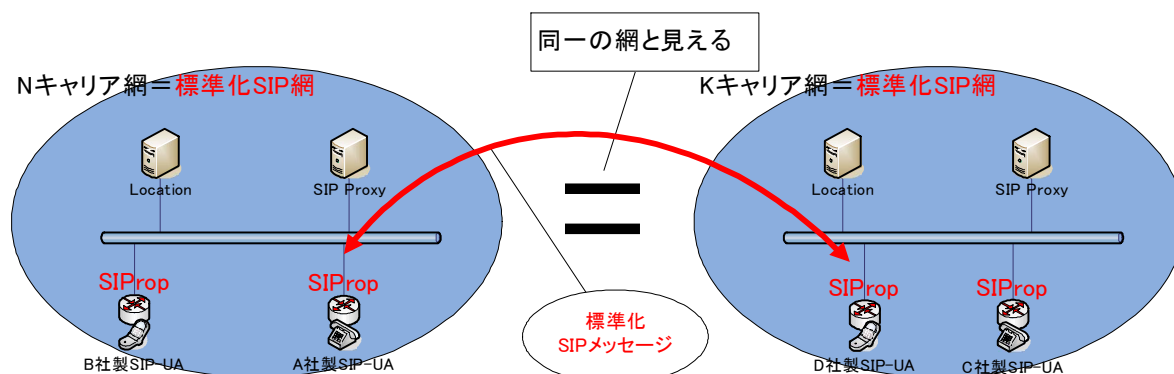


図 2-1 VoIP (SIP-UA) 相互接続問題を解決するミドルウェア SIPProp

3. 開発内容

3. 1. 開発環境

開発環境としては、下記を用意し、SIProp がそれぞれの環境で動作すること、相互接続できることを確認した。

- ・ 開発言語 : J2ME-PP (J2SE1.3 相当、Jain-SIP ライブラリ)
- ・ プラットフォーム : Windows, Linux
- ・ 電話機
 - ソフトフォン : KPhone (Zaurus)、X-lite (Windows)
 - ハードフォン : GrandStream 製 IP 電話機
- ・ サーバ : Asterisk

3. 2. B2BUA (Back To Back User Agent)

SIProp の核となるのが、SIP の方言を吸収するための B2BUA (Back To Back User Agent) である。

B2BUA を SIP 端末や SIP サーバなど、通信する機器の間に配置し、B2BUA がプロトコルを変換することで仕様が違う SIP 機器を接続できるようにした。

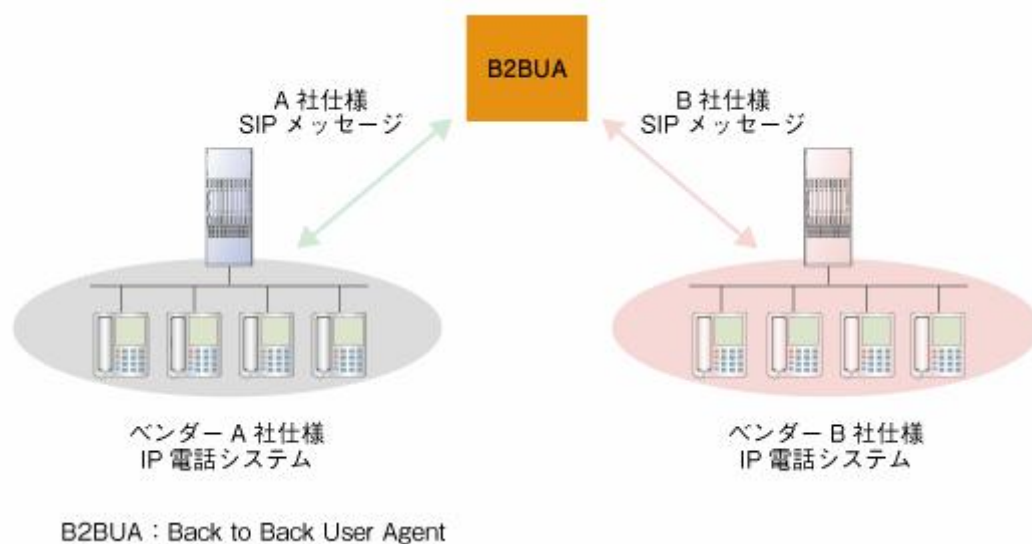


図 3-2-1 B2BUA による SIP の方言の吸収

3. 3. SIP メッセージのヘッダ部分の変換

「SIP メッセージのヘッダ部分の変換」とは、SIP の解釈が異なるためにヘッダ部分の記述方法が異なっている場合に、どの端末でも処理できるような標準的なものに書き換える機能 (図 3-3-1) である。

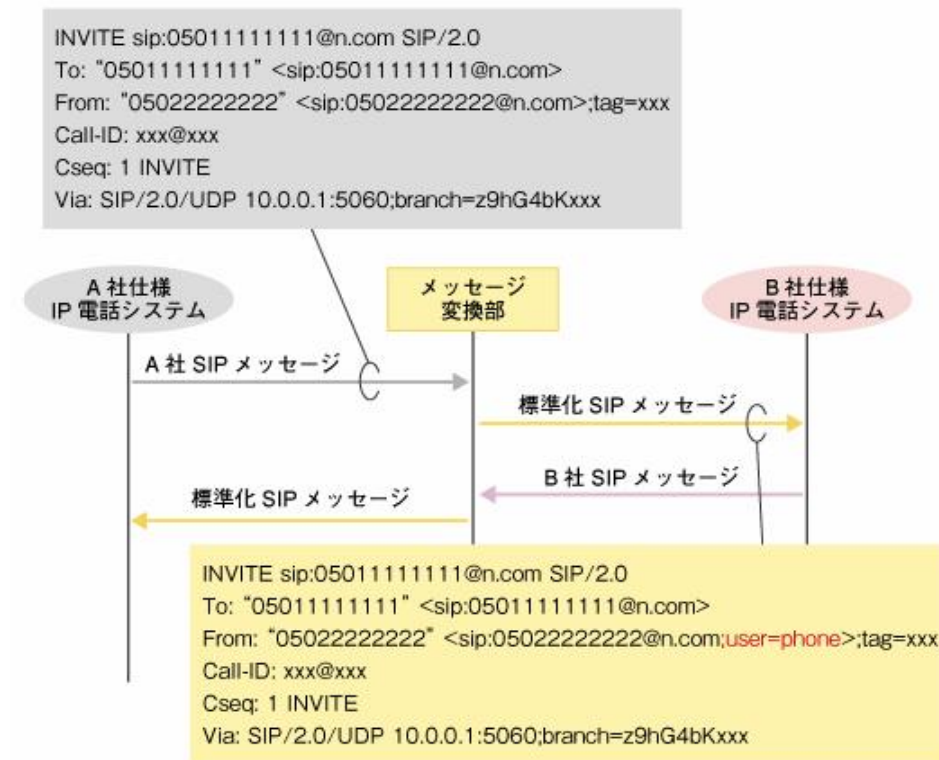


図 3-3-1 SIP メッセージのヘッダ部分の簡単な変換

3. 4. FlatSIP と UASet

SIPprop では、「似た動作をする端末群」（これを UASet と呼ぶ）ごとに、機能単位で SIP メッセージやシーケンスをベストプラクティス集という形で定義しており、これを標準化 SIP 「FlatSIP」と呼ぶ。

このモジュール化によって、異なる仕様の SIP システム間であってもその接続を容易にしている。まず、FlatSIP と対象端末の SIP メッセージやシーケンスを比較することで、対象端末がどの UASet に属し、どの FlatSIP が適しているかを判定する。続いて、対象端末がどのような SIP メッセージやシーケンスであるべきかを判断し、FlatSIP との差分を端末ごとにモジュールとして実装している。

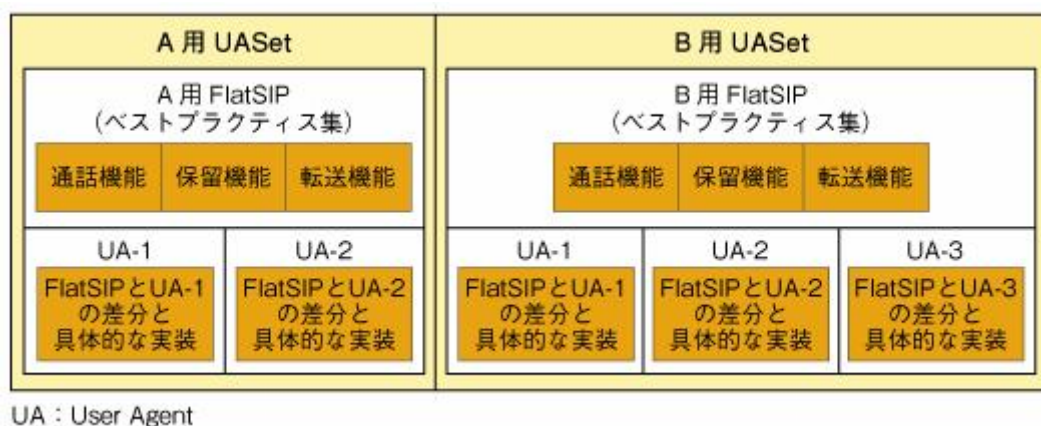


図 4.3-1 FlatSIP と UASet

4. 今後の課題、展望

今回の開発で、相互接続を実現できた。しかし、UASet の充実は、法律やその他の事情があり、これ以上は不可能な状態である。そこで、今後展望としては、本格的な普及へ向けて「SIP IX」や「Asterisk」等との連携が重要となる。具体的には、「SIP IX」で相互接続性を確保するための基盤を SIProp で提供するために、サーバ側での使用を重視した「SIProp バージョン 2.0」を新規に開発中である。

また、SIProp の新プロジェクトとして Web と SIP とのアプリの融合を支援するミドルウェア（アプリケーションサーバ）「雷電」を検討中である。雷電は、UA ベースフレームワークの実装とし、柔軟性の高いアプリ構築やクライアント間マッシュアップが可能である。Web と SIP の相互接続を目指している。

5. 開発者

- 代表

今村謙之

- メンバー

<http://www.siprop.org/ja/1.0/index.php?SIProp%2FPJ%C2%CE%C0%A9>

6. 関連 Web サイト

- SIProp 開発者サイト

<http://www.siprop.org/>

- SIProp 勉強会告知サイト

<http://www.siprop.org/ja/2.0/index.php?%B3%AB%C8%AF%2F%A5%B3%A5%DF%A5%E5%A5%CB%A5%C6%A5%A3%A1%BC%2F%CA%D9%B6%AF%B2%F1>

- 新プロジェクト「雷電」のサイト

<http://raiden.siprop.org/>

- 書籍「俺流 プロトコル実装入門」のサイト

<http://pipbook.siprop.org/>