

# IP電話の基礎と応用

～ 音声符号化、SIP、RTPからIP-PBX、NGNまで～

波多浩昭  
hata@nttpc.co.jp

## 目的

- ◆ SIPプロトコルの基本動作を理解する
- ◆ VoIPをサービスとして実現するための問題整理
- ◆ SIPを利用したVoIPサービスを開発を理解する
- ◆ 次世代電話におけるSIPの重要性を理解する

# 目次

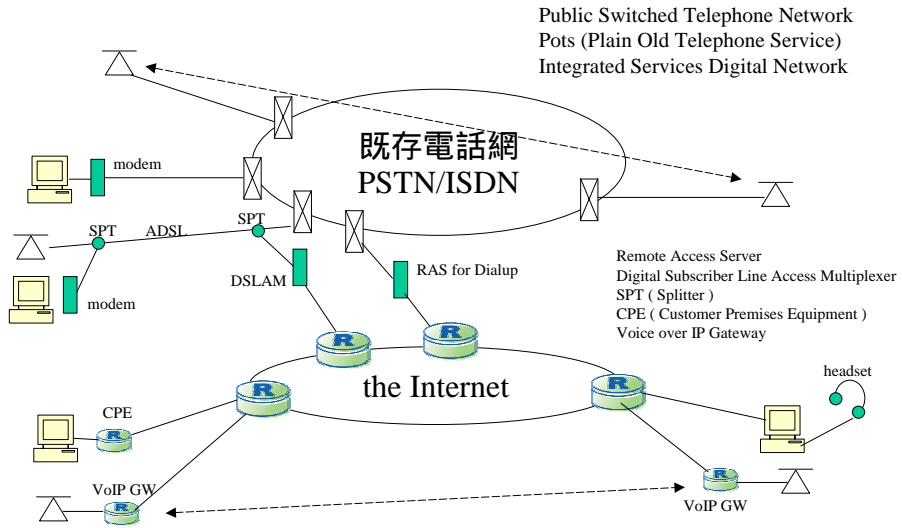
- **パート0 VoIP概要と本コースの内容**
  - VoIPとは
- **パート1 SIPの基本**
  - SIPの歴史、どうして注目されたのか、その背景
  - SIPの概要
  - シーケンス
  - ルーティング
- **パート2 RTPと信号処理**
  - RTP
  - サンプリングと符号化
  - ソフトフォン実装概要
- **パート3 SIPをとりまく諸問題**
  - NAT
  - 端末の音響
  - 品質、信頼性
- **パート4 サービス開発とSIPシーケンス**
  - システム化(PSTNとの接続)
  - IPPBX
  - コールピックアップ
  - 3PCC
  - 着信転送
  - プレゼンス、IM
  - その他のアプリケーション
- **パート5 3GPP/IMS/SIP/NGN/RFC3261**
  - 3GPP NGN
  - IMS

# パート0

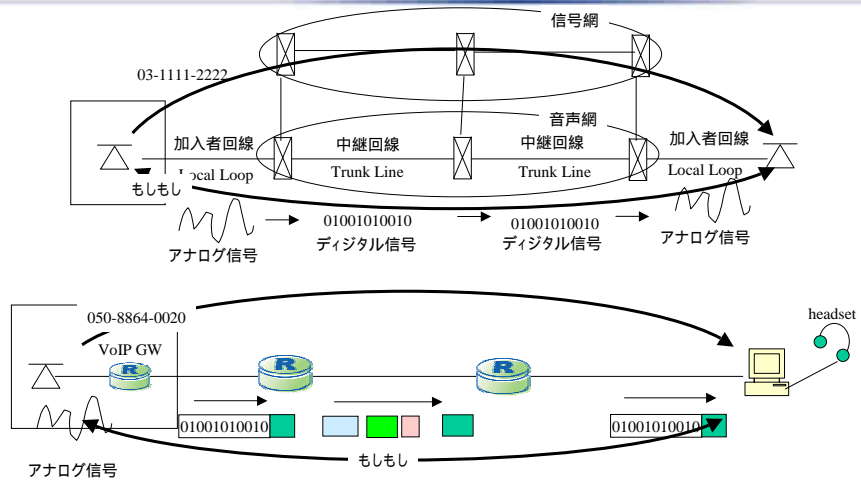
## パート0

### VoIPの概要と 本コースの内容

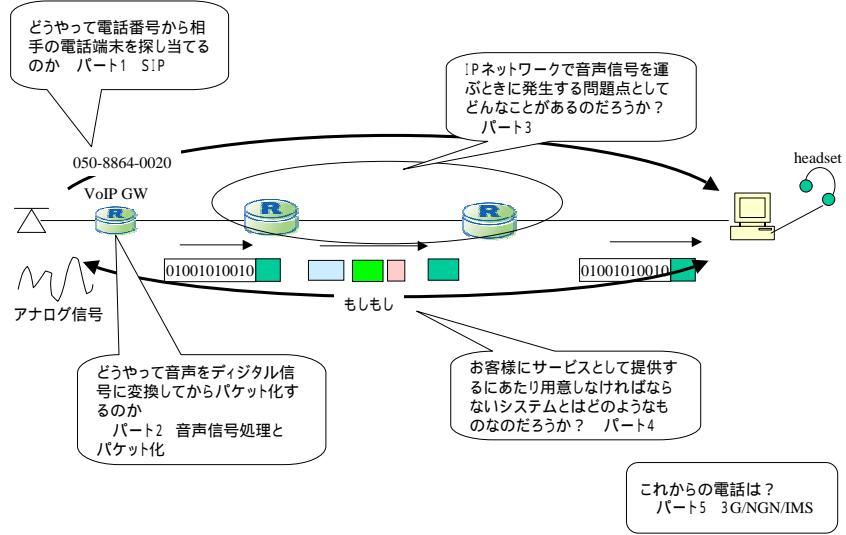
# インターネットと電話網の関係



# 技術的観点での比較



# VoIPを実現する技術要素



# パート1

## パート1

### SIPの基本知識

## 関連イベント

- ◆RFC 2543 (1999年3月)
- ◆Windows XP発売 (2001年10月)  
添付のWindows MessengerがSIP対応
- ◆Windows 2000、Windows 98/Me対応  
MSN Messenger ver4.xからSIP対応
- ◆YAMAHA RT60wルータがSIP対応(2001年12月)
- ◆BB phoneサービス開始(0AB-J)
- ◆RFC 3261 (2002年6月)
- ◆050電話サービス開始(2002年12月)

## RFC(1/2)

### 当初

RFC 2543 (153枚)

### 改定後

RFC 3261 (269枚) SIPコア

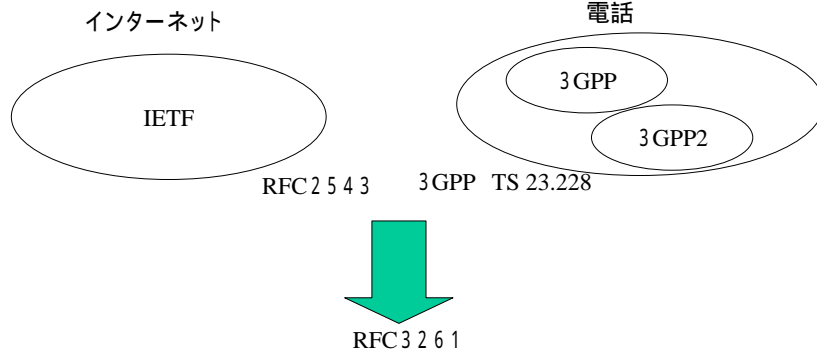
RFC 3262 (14) 応答メッセージの転送の信頼性強化

RFC 3263 (17) SIPサーバがDNSを使う場合の処理

RFC 3264 (25) SDPのオファーアンサーモデル

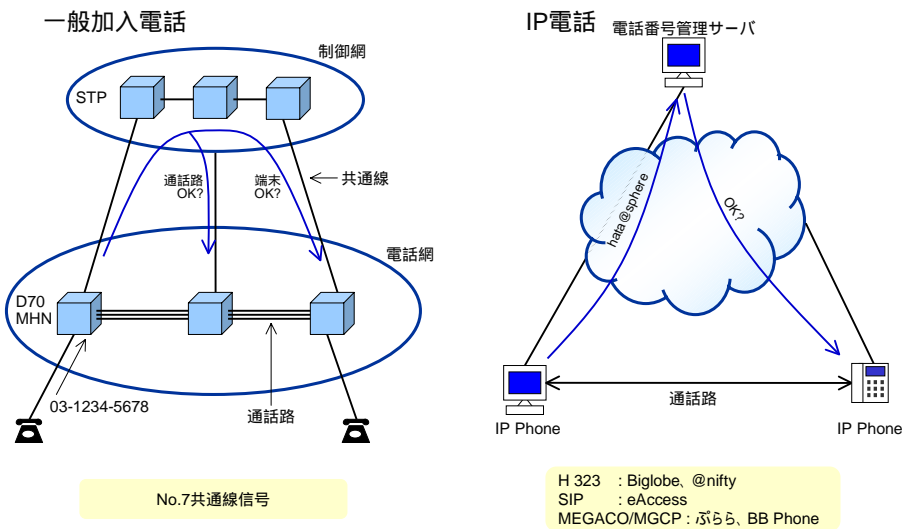
RFC 3265 (38) イベント通知処理

# RFC(2/2)



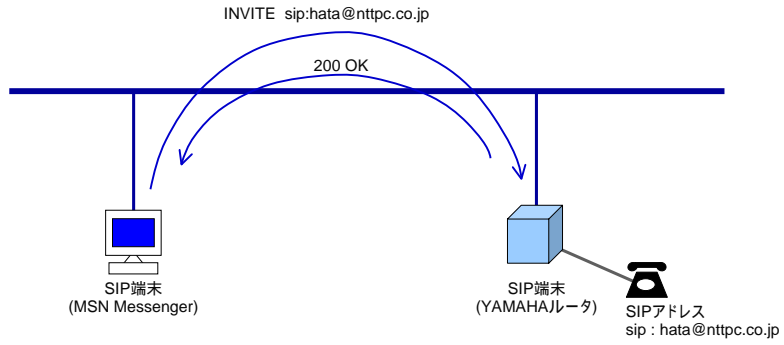
IETFと3GPPの共同作業により、次世代電話網制御プロトコルとして制定

# 呼制御プロトコル

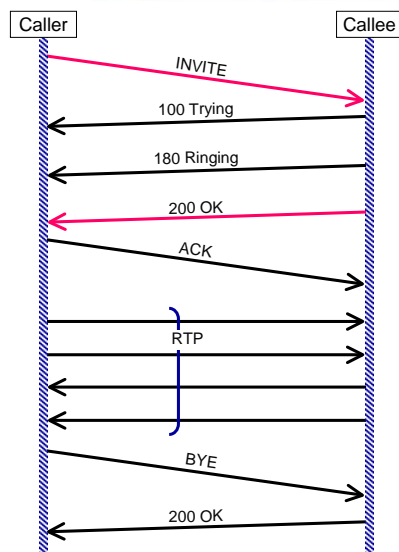


## SIPを使った簡単な通話(とりあえず試してみよう)

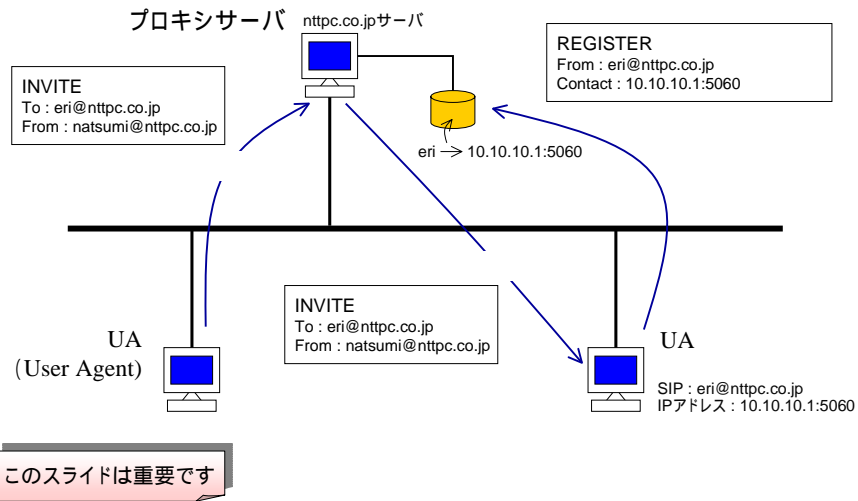
相手IPアドレスがわかっており、  
ネットワークに接続されているとき



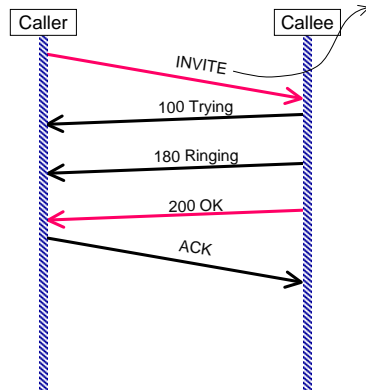
## シーケンス概要



# プロキシサーバとUA



# INVITE ダンプ

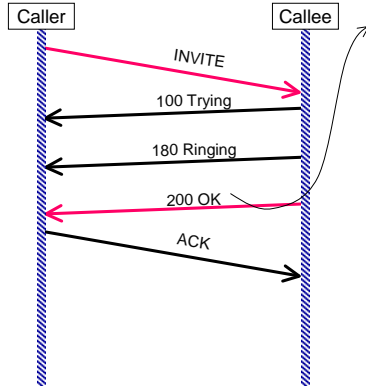


```

INVITE sip:bob@biloxi.example.com SIP/2.0
Via: SIP/2.0/TCP client.atlanta.example.com
Max-Forwards: 70
Route: <sip:ss1.atlanta.example.com;lr>
From: Alice <sip:alice@atlanta.example.com>
To: Bob <sip:bob@biloxi.example.com>
Call-ID: 38482762982@atlanta.example.com
CSeq: 1 INVITE
Contact: <sip:alice@client.atlanta.example.com>
Content-Type: application/sdp
Content-Length: 151

v=0
o=alice 2890 28526 IN IP4 cnt.ata.example.com
s=-
c=IN IP4 192.0.2.101
t=0 0
m=audio 49172 RTP/AVP 0
a=rtpmap:0 PCMU/8000
    
```

# 200OKダンプ

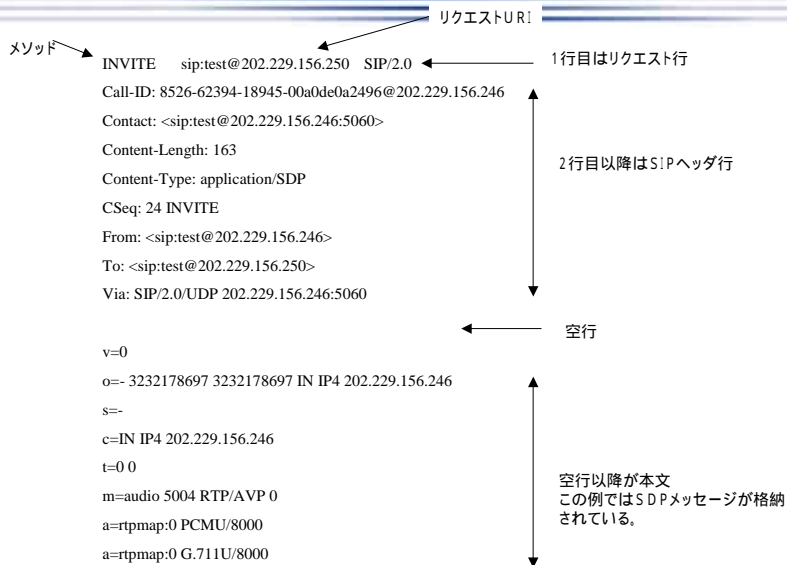


```

SIP/2.0 200 OK
Via: SIP/2.0/TCP ss2.biloxi.example.com
Via: SIP/2.0/TCP ss1.atlanta.example.com
Via: SIP/2.0/TCP client.atlanta.example.com
Record-Route: <sip:ss2.biloxi.example.com;lr>,
<sip:ss1.atlanta.example.com;lr>
From: Alice <sip:alice@atlanta.example.com>
To: Bob <sip:bob@biloxi.example.com>
Call-ID: 3848276298220188511@atlanta.example.com
CSeq: 2 INVITE
Contact: <sip:bob@client.biloxi.example.com;>
Content-Type: application/sdp
Content-Length: 147

v=0
o=bob 2890847 2890847 IN IP4 client.bexample.com
s=-
c=IN IP4 192.0.2.201
t=0 0
m=audio 3456 RTP/AVP 0
a=rtpmap:0 PCMU/8000
    
```

# メッセージフォーマット

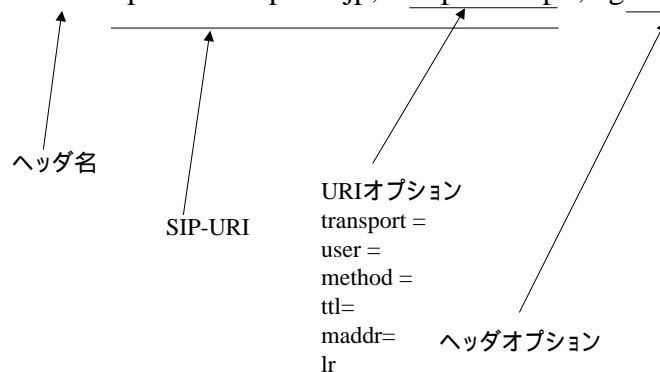


## 代表的なヘッダ

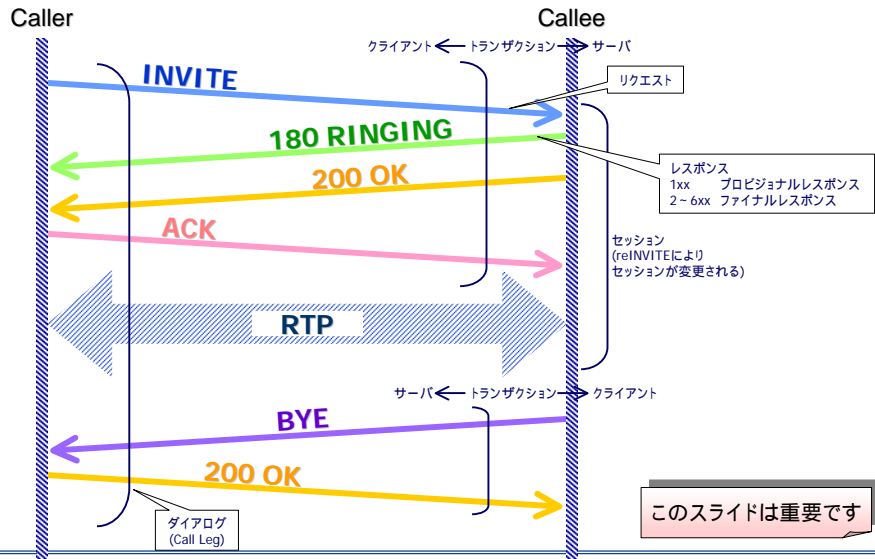
- To** : あて先のSIP-URL
- From** : 発信元のSIP-URL
- Call-ID** : ダイアログを識別
- CSeq** : 同一Call-IDで何個目のトランザクションかを表示  
(ACKとCANCELは対応するINVITEとあわせる)
- Via** : 本リクエストに対するレスポンスはここへ送ってほしい旨通知
- Contact** : 以後、自分に対するリクエストはここへ送ってほしい旨通知
- Content-Type** : メッセージボディのMIMEタイプ
- (例)
- |         |                           |
|---------|---------------------------|
| INVITE  | application/SDP           |
| NOTIFY  | application/xpidftxml     |
|         | application/cpim-pidftxml |
| MESSAGE | text/plain                |

## SIP-URIとオプション

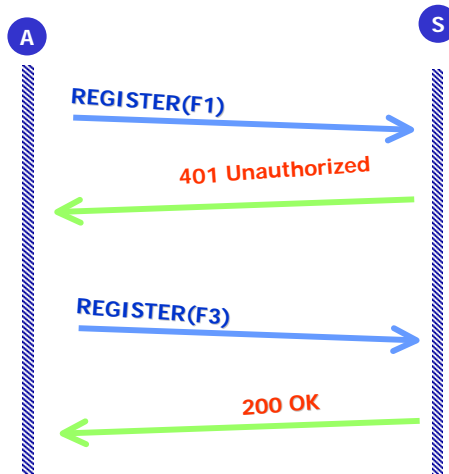
From: <sip:hata@nttpc.co.jp;transport=udp>;tag=24521



# 用語



# REGISTERシーケンス



## REGISTER(F1)

```
REGISTER sips:ss2.biloxi.example.com SIP/2.0
Via: SIP/2.0/TLS client.biloxi.example.com:5061;branch=z9hG4bKnashds7
Max-Forwards: 70
From: Bob <sips:bob@biloxi.example.com>;tag=a73kszlfl
To: Bob <sips:bob@biloxi.example.com>
Call-ID: 1j9FpLxk3uxtm8tn@biloxi.example.com
CSeq: 1 REGISTER
Contact: <sips:bob@client.biloxi.example.com>
Content-Length: 0
```

## REGISTER(F2)

```
SIP/2.0 401 Unauthorized
Via: SIP/2.0/TLS client.biloxi.example.com:5061;branch=z9hG4bKnashds7
      ;received=192.0.2.201
From: Bob <sips:bob@biloxi.example.com>;tag=a73kszlfl
To: Bob <sips:bob@biloxi.example.com>;tag=1410948204
Call-ID: 1j9FpLxk3uxtm8tn@biloxi.example.com
CSeq: 1 REGISTER
WWW-Authenticate: Digest realm="atlanta.example.com", qop="auth",
                   nonce="ea9c8e88df84f1cec4341ae6cbe5a359",
                   opaque="", stale=FALSE, algorithm=MD5
Content-Length: 0
```

チャレンジ

## REGISTER(F3)

```
REGISTER sips:ss2.biloxi.example.com SIP/2.0
Via: SIP/2.0/TLS client.biloxi.example.com:5061;branch=z9hG4bKnashd92
Max-Forwards: 70
From: Bob <sips:bob@biloxi.example.com>;tag=ja743ks76zlfIH
To: Bob <sips:bob@biloxi.example.com>
Call-ID: 1j9FpLxk3uxtm8tn@biloxi.example.com
CSeq: 2 REGISTER
Contact: <sips:bob@client.biloxi.example.com>
Authorization: Digest username="bob", realm="atlanta.example.com"
                nonce="ea9c8e88df84f1cec4341ae6cbe5a359", opaque="",
                uri="sips:ss2.biloxi.example.com",
                response="dfe56131d1958046689d83306477ecc"
Content-Length: 0
```

レスポンス

## REGISTER(F4)

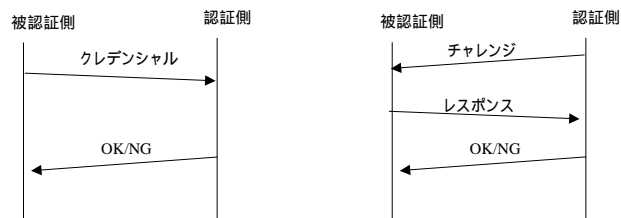
```
SIP/2.0 200 OK
Via: SIP/2.0/TLS
client.biloxi.example.com:5061;branch=z9hG4bKnashd92
;received=192.0.2.201
From: Bob <sips:bob@biloxi.example.com>;tag=ja743ks76zlfIH
To: Bob <sips:bob@biloxi.example.com>;tag=37GkEhwI6
Call-ID: 1j9FpLxk3uxtm8tn@biloxi.example.com
CSeq: 2 REGISTER
Contact: <sips:bob@client.biloxi.example.com>;expires=3600
Content-Length: 0
```

## AAA 認証

- 認証(Authentication)
  - 入力がクレデンシャル(ID+PASSWD)、出力はOK / NG
- 承認(Authorization)
  - 入力ID、出力は権限
- 課金(Accounting)
  - 入力ID、出力はリソース使用状況

## 認証方式

- PAP (Password Authentication Protocol)
  - ネットワーク上のセキュリティに弱い: サーバ上のセキュリティに強い
- CHAP (Challenge Handshake Authentication Protocol)
  - ネットワーク上のセキュリティに強い: サーバ上のセキュリティに弱い



# 認証



```
SIP/2.0 407 Proxy Authentication Required
Call-ID: 53409-55514-55500-00a0de0be46e@10.224.228.1
CSeq: 82 INVITE
From: <slp:0352065369@voip.sphere.ne.jp>;tag=2389310069
To: <slp:0352106434@sphere.ne.jp:user=phone>
User-Agent: YAMAHA RTA54i
Via: SIP/2.0/UDP 10.224.228.1:5060
Date: Thu, 27 Feb 2003 12:17:33 GMT
Proxy-Authenticate: Digest realm="nttpc.com", domain="10.227.109.134",
nonce="1046347682", opaque="", stale=FALSE, algorithm=MD5

INVITE slp:0352106434@ocn.ne.jp:user=phone SIP/2.0
Call-ID: 53409-55514-55500-00a0de0be46e@10.224.228.1
Contact: <slp:0352065369@10.224.228.1:5060>
Content-Length: 135
Content-Type: application/sdp
CSeq: 83 INVITE
From: <slp:0352065369@voip-ca.sphere.ne.jp>;tag=2389310069
Proxy-Authorization: Digest username="nttpc5369", realm="nttpc.com",
nonce="1046347682", opaque="", uri="10.227.109.134",
response="d2b7b1cbfa020aab7ce6c728b00238d1"
To: <slp:0352106434@sphere.ne.jp>
User-Agent: YAMAHA RTA54i
Via: SIP/2.0/UDP 10.224.228.1:5060
```

# 認証条件

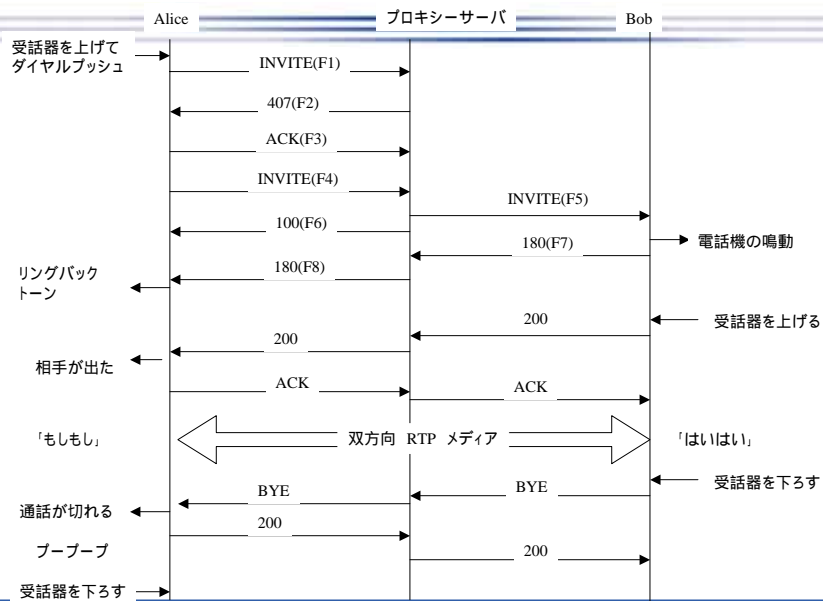
```
char *username="bob";
char *nonce="1121030407281420";
char *realm="atlanta.example.com";
char *passwd="mypasswd";
char *uri="sips:ss2.biloxi.example.com ";
char *cnonce="0D03C005";
char nc[9]="00000001";
char *qop="auth";
char *method="INVITE";
char HA1[36];
char HA2[36];
char response[36];
```

# Digest認証の仕組み

HA1=MD5(Username:Realm:Password)  
 HA2=MD5(Method:URI)  
 Response=MD5(HA:Nonce:HA2)



# ワンコールシーケンス



## INVITE(INV) F1

INVITE sip:bob@biloxi.example.com SIP/2.0  
Via: SIP/2.0/TCP client.atlanta.example.com:5060;branch=z9hG4bK74b43  
Max-Forwards: 70  
Route: <sip:ss1.atlanta.example.com;lr>  
From: Alice <sip:alice@atlanta.example.com>;tag=9fxcd76sl  
To: Bob <sip:bob@biloxi.example.com>  
Call-ID: 3848276298220188511@atlanta.example.com  
CSeq: 1 INVITE  
Contact: <sip:alice@client.atlanta.example.com;transport=tcp>  
Content-Type: application/sdp  
Content-Length: 151

v=0  
o=alice 2890844526 2890844526 IN IP4 client.atlanta.example.com  
s=-  
c=IN IP4 192.0.2.101  
t=0 0  
m=audio 49172 RTP/AVP 0  
a=rtpmap:0 PCMU/8000

見どころ  
SDP

## INVITE(407) F2

SIP/2.0 407 Proxy Authorization Required  
Via: SIP/2.0/TCP client.atlanta.example.com:5060;branch=z9hG4bK74b43  
From: Alice <sip:alice@atlanta.example.com>;tag=9fxcd76sl  
To: Bob <sip:bob@biloxi.example.com>;tag=3flal12sf  
Call-ID: 3848276298220188511@atlanta.example.com  
CSeq: 1 INVITE  
Proxy-Authenticate: Digest realm="atlanta.example.com", qop="auth",  
nonce="f84f1cec41e6cbe5aea9c8e88d359",  
opaque="", stale=FALSE, algorithm=MD5  
Content-Length: 0

見どころ  
From-tag  
Challenge

## INVITE(ACK)F3

```
ACK sip:bob@biloxi.example.com SIP/2.0
Via: SIP/2.0/TCP client.atlanta.example.com:5060;branch=z9hG4bK74b43
Max-Forwards: 70
From: Alice <sip:alice@atlanta.example.com>;tag=9fxced76sl
To: Bob <sip:bob@biloxi.example.com>;tag=3flal12sf
Call-ID: 3848276298220188511@atlanta.example.com
CSeq: 1 ACK
Content-Length: 0
```

## INVITE(INV)F4

```
INVITE sip:bob@biloxi.example.com SIP/2.0
Via: SIP/2.0/TCP client.atlanta.example.com:5060;branch=z9hG4bK74bf9
Max-Forwards: 70
Route: <sip:ss1.atlanta.example.com;lr>
From: Alice <sip:alice@atlanta.example.com>;tag=9fxced76sl
To: Bob <sip:bob@biloxi.example.com>
Call-ID: 3848276298220188511@atlanta.example.com
CSeq: 2 INVITE
Contact: <sip:alice@client.atlanta.example.com;transport=tcp>
Proxy-Authorization: Digest username="alice", realm="atlanta.example.com",
    nonce="wf84f1ceczx41ae6cbe5aea9c8e88d359",
    opaque="", uri="sip:bob@biloxi.example.com",response="42ce3cef44b22f50c6a6071bc8"
Content-Type: application/sdp
Content-Length: 151

v=0
o=alice 2890844526 2890844526 IN IP4 client.atlanta.example.com
s=-
c=IN IP4 192.0.2.101
t=0 0
m=audio 49172 RTP/AVP 0
a=rtptime:0 PCMU/8000
```

見どころ

CSeq  
response

## INVITE(180) F7

SIP/2.0 180 Ringing  
Via: SIP/2.0/TCP ss1.atlanta.example.com:5060;branch=z9hG4bK2d4790  
Via: SIP/2.0/TCP client.atlanta.example.com:5060;branch=z9hG4bK74bf9  
Record-Route: <sip:ss2.biloxi.example.com>, <sip:ss1.atlanta.example.com>  
From: Alice <sip:alice@atlanta.example.com>;tag=9fxcde76sl  
To: Bob <sip:bob@biloxi.example.com>;tag=314159  
Call-ID: 3848276298220188511@atlanta.example.com  
Contact: <sip:bob@client.biloxi.example.com;transport=tcp>  
CSeq: 2 INVITE  
Content-Length: 0

見どころ

Via  
From  
Toヘッダ

## INVITE(180)F8

SIP/2.0 180 Ringing  
Via: SIP/2.0/TCP client.atlanta.example.com:5060;branch=z9hG4bK74bf9  
Record-Route: <sip:ss2.biloxi.example.com>, <sip:ss1.atlanta.example.com>  
From: Alice <sip:alice@atlanta.example.com>;tag=9fxcde76sl  
To: Bob <sip:bob@biloxi.example.com>;tag=314159  
Call-ID: 3848276298220188511@atlanta.example.com  
Contact: <sip:bob@client.biloxi.example.com;transport=tcp>  
CSeq: 2 INVITE  
Content-Length: 0

見どころ

Via

## INVITE(200)F9

```
SIP/2.0 200 OK
Via: SIP/2.0/TCP ss2.biloxi.example.com:5060;branch=z9hG4bK721e4.1
Via: SIP/2.0/TCP ss1.atlanta.example.com:5060;branch=z9hG4bK2d4790
Via: SIP/2.0/TCP client.atlanta.example.com:5060;branch=z9hG4bK74bf9
Record-Route: <sip:ss2.biloxi.example.com>,<sip:ss1.atlanta.example.com>
From: Alice <sip:alice@atlanta.example.com>;tag=9fxced76sl
To: Bob <sip:bob@biloxi.example.com>;tag=314159
Call-ID: 3848276298220188511@atlanta.example.com
CSeq: 2 INVITE
Contact: <sip:bob@client.biloxi.example.com;transport=tcp>
Content-Type: application/sdp
Content-Length: 147

v=0
o=bob 2890844527 2890844527 IN IP4 client.biloxi.example.com
s=-
c=IN IP4 192.0.2.201
t=0 0
m=audio 3456 RTP/AVP 0
a=rtmap:0 PCMU/8000
```

見どころ

SDP  
Contactヘッダ

## INVITE(ACK)

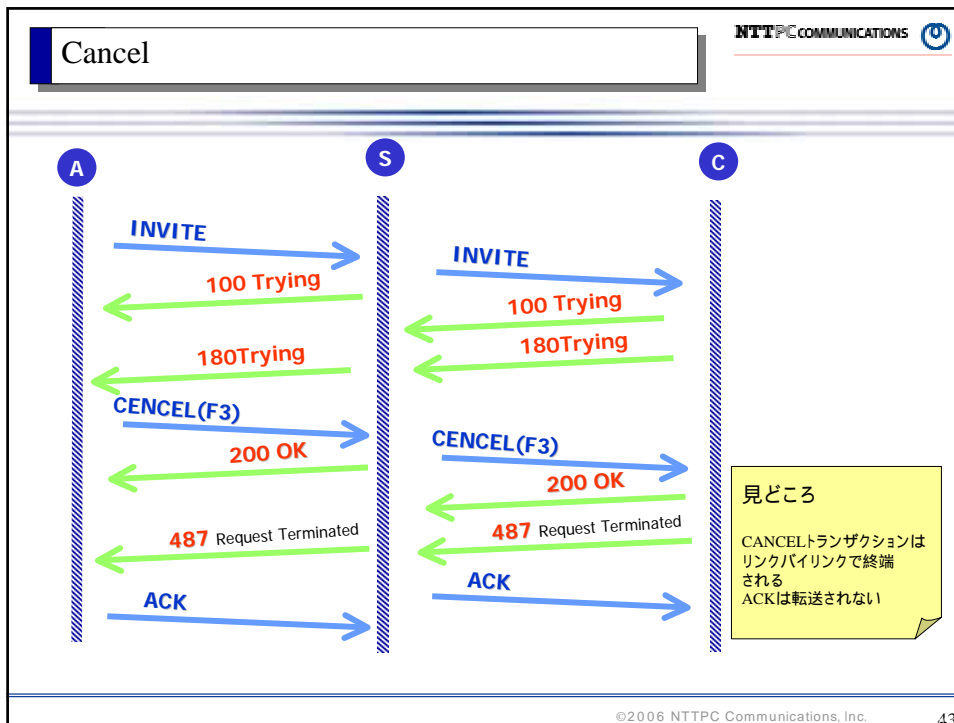
```
ACK sip:bob@client.biloxi.example.com SIP/2.0
Via: SIP/2.0/TCP
client.atlanta.example.com:5060;branch=z9hG4bK74b76
Max-Forwards: 70
Route: <sip:ss1.atlanta.example.com>,<sip:ss2.biloxi.example.com>
From: Alice <sip:alice@atlanta.example.com>;tag=9fxced76sl
To: Bob <sip:bob@biloxi.example.com>;tag=314159
Call-ID: 3848276298220188511@atlanta.example.com
CSeq: 2 ACK
Content-Length: 0
```

## INVITE(BYE)


```
BYE sip:alice@client.atlanta.example.com SIP/2.0
Via: SIP/2.0/TCP client.biloxi.example.com:5060;branch=z9hG4bKnashds7
Max-Forwards: 70
Route: <sip:ss2.biloxi.example.com;lr>, <sip:ss1.atlanta.example.com;lr>
From: Bob <sip:bob@biloxi.example.com>;tag=314159
To: Alice <sip:alice@atlanta.example.com>;tag=9fxced76sl
Call-ID: 3848276298220188511@atlanta.example.com
CSeq: 1 BYE
Content-Length: 0
```

## INVITE(200)

```
SIP/2.0 200 OK
Via: SIP/2.0/TCP client.biloxi.example.com:5060;branch=z9hG4bKnashds7
From: Bob <sip:bob@biloxi.example.com>;tag=314159
To: Alice <sip:alice@atlanta.example.com>;tag=9fxced76sl
Call-ID: 3848276298220188511@atlanta.example.com
CSeq: 1 BYE
Content-Length: 0
```



## CANCEL(F1)



CANCEL sip:bob@biloxi.example.com SIP/2.0  
 Via: SIP/2.0/UDP client.atlanta.example.com:5060;branch=z9hG4bK74bf9  
 Max-Forwards: 70  
 From: Alice <sip:alice@atlanta.example.com>;tag=9fxcde76sl  
 To: Bob <sip:bob@biloxi.example.com>  
 Route: <sip:ss1.atlanta.example.com;lr>  
 Call-ID: 2xTb9vxSit55XU7p8@atlanta.example.com  
 CSeq: 1 CANCEL  
 Content-Length: 0

©2006 NTT PC Communications, Inc. 44

## CANCEL(F2)

SIP/2.0 200 OK  
Via: SIP/2.0/UDP client.atlanta.example.com:5060;branch=z9hG4bK74bf9  
From: Alice <sip:alice@atlanta.example.com>;tag=9fxced76sl  
To: Bob <sip:bob@biloxi.example.com>  
Call-ID: 2xTb9vxSit55XU7p8@atlanta.example.com  
CSeq: 1 CANCEL  
Content-Length: 0

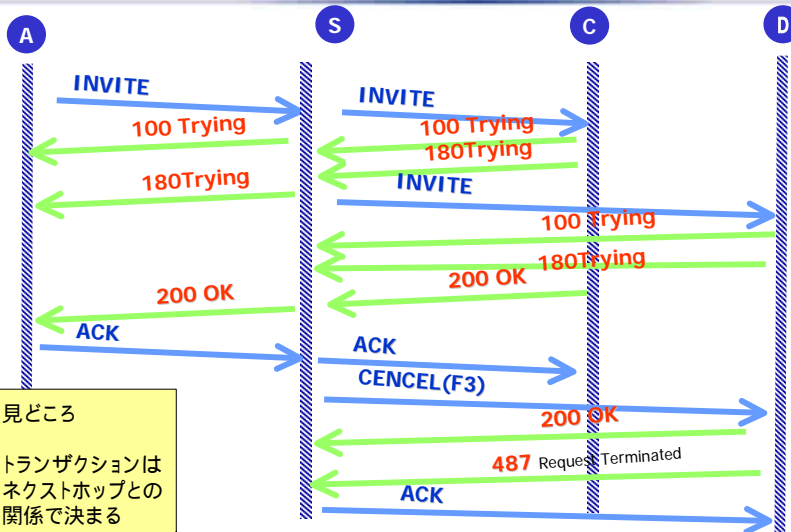
## CANCEL(F3)

SIP/2.0 487 Request Terminated  
Via: SIP/2.0/UDP client.atlanta.example.com:5060;branch=z9hG4bK74bf9  
From: Alice <sip:alice@atlanta.example.com>;tag=9fxced76sl  
To: Bob <sip:bob@biloxi.example.com>;tag=314159  
Call-ID: 2xTb9vxSit55XU7p8@atlanta.example.com  
CSeq: 1 INVITE

# CANCEL(F4)

ACK sip:bob@biloxi.example.com SIP/2.0  
Via: SIP/2.0/UDP client.atlanta.example.com:5060;branch=z9hG4bK74bf9  
Max-Forwards: 70  
From: Alice <sip:alice@atlanta.example.com>;tag=9fxcde76sl  
To: Bob <sip:bob@biloxi.example.com>;tag=314159  
Call-ID: 2xTb9vxSit55XU7p8@atlanta.example.com  
Proxy-Authorization: Digest username="alice",  
realm="atlanta.example.com",  
nonce="ze7k1ee88df84f1cec431ae6cbe5a359", opaque="",  
uri="sip:bob@biloxi.example.com",  
response="b00b416324679d7e243f55708d44be7b"  
CSeq: 1 ACK  
Content-Length: 0

# Cancel (親子電話)

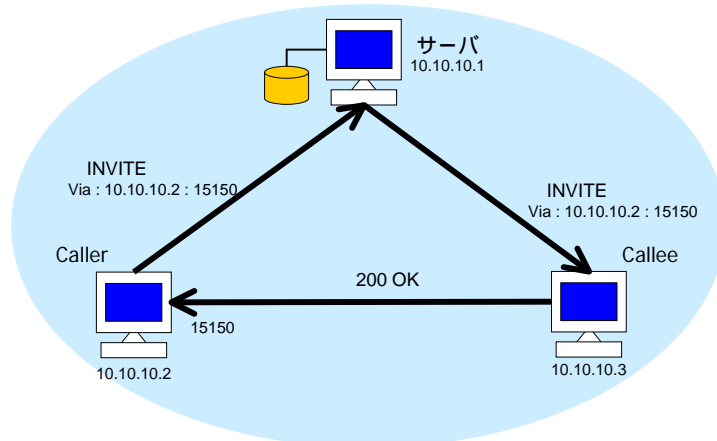


見どころ  
トランザクションは  
ネクストホップとの  
関係で決まる

## ルーティング

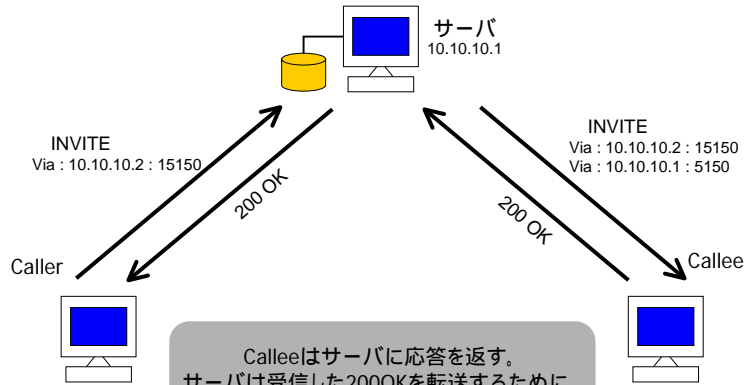
- ルーティングの必要性
  - どうしてトランスポートプロトコルにルーティングが必要なのか？
- 中継されるUDP
  - UDP上のトランスポートレイヤ(SNMP、RADIUSなどはルーティング機能はあるのか？)

## ルーティングの必要性 1



サーバは状態管理をしない。Calleeは直接Callerに応答を返す

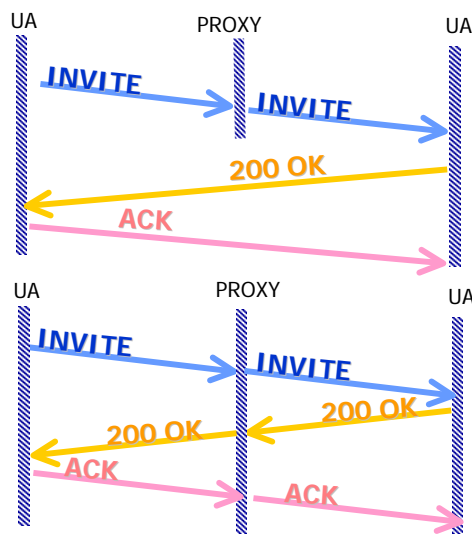
## ルーティングの必要性2



Calleeはサーバに応答を返す。  
サーバは受信した200OKを転送するために、  
CallerのINVITEに自分のアドレスを追記する。

- 結果をサーバで管理したい場合
- 呼切断(BYE)もサーバで受信したい場合はRouteヘッダを使う
- Caller/Callee間でIPアドレスを隠蔽する効果はない SDPの中に入っているから

## ルーティング



INVITEだけ転送するので  
あとは勝手にやって頂戴

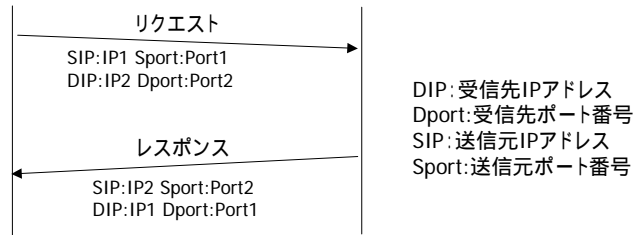
通過するパケットはプロキシ  
に転送してくださいね。

- ・Viaスタック
- ・Record-Route

## UDPの基本1

クライアント IP1/Port1

サーバ IP2/Port2



DIP: 受信先IPアドレス  
Dport: 受信先ポート番号  
SIP: 送信元IPアドレス  
Sport: 送信元ポート番号

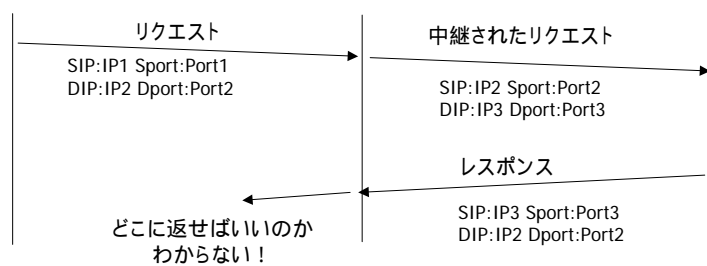
RadiusやSNMPなどのUDP上のトランスポートプロトコルは、  
リクエストのIPヘッダの送信先と送信元を逆転してレスポンスを生成していた。

## UDPの基本2

発信元IP1/Port1

中継IP2/Port2

サーバIP3/Port3



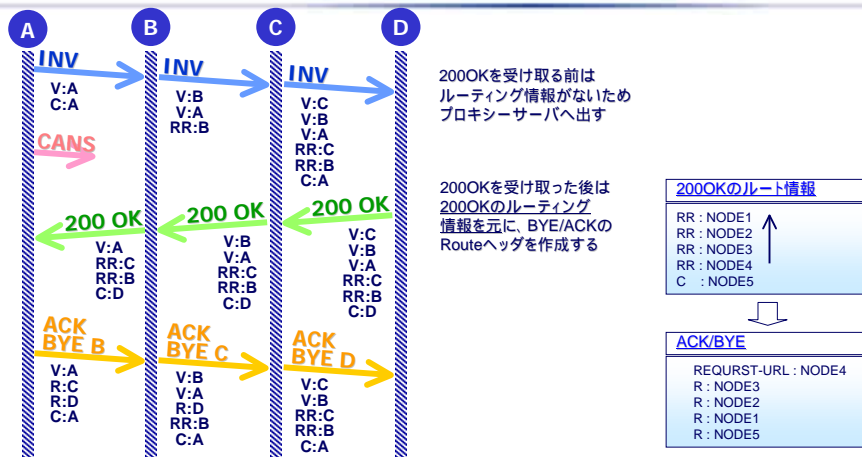
中継サーバIP2はプロキシ - サーバであり、リクエストは本当のサーバ3に  
転送される。しかし、返されたレスポンスにはクライアント情報が入っていない  
ためどこにこのレスポンスを返すべきが判断できない。

## ヘッダ(Via,Route,R-R)

- Via
- Record-Route
- Route
- Contact
- Request-URI

## STRICT ROUTING

パターン1 (Callerから切断)

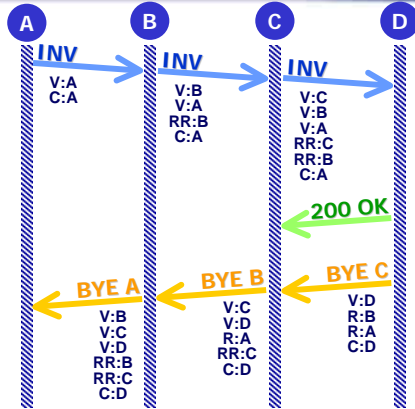


### ルーティング情報の作り方

- Record-Routeの最下位のノードをリクエストURLにする。
- Record-RouteがないときはContactのノードをリクエストURLとする。 処理終わり
- Record-Routeヘッダの最下位を除き、下から順に上へ向かって Recordヘッダへコピーする。
- ContactノードをRecordヘッダの最下位へコピーする。

# STRICT ROUTING

パターン2 (Calleeから切断)



INVを受け取ったなら  
そのINVITEリクエストを元に  
BYEのルーティング情報を  
作成する

INVITEのルート情報
RR : NODE1
RR : NODE2
RR : NODE3
RR : NODE4
C : NODE5

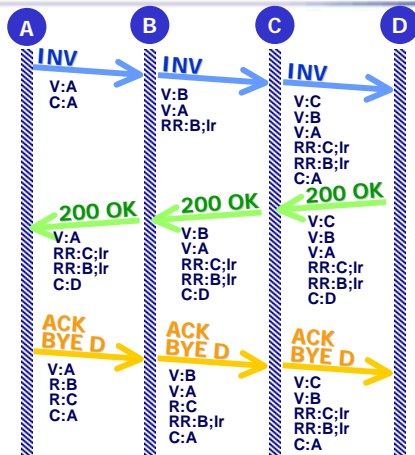
BYEのルーティング情報
REQUST-URL : NODE1
R : NODE2
R : NODE3
R : NODE4
R : NODE5

### ルーティング情報の作り方

- Record-Routeの最上位のノードをリクエストURLにする。
- Record-RouteがないときはContactをリクエストURLとする。 処理終わり
- Record-Routeの2番目のノードから、上から順に下へ向かって
- Recordヘッダへコピーする。
- ContactヘッダをRecordヘッダの最下位へコピーする。

# LOOSE ROUTING

パターン1 (Callerから切断)



200OKを受け取った後は  
200OKのルーティング  
情報を元に、BYE/ACKの  
Routeヘッダを作成する

200OKのルート情報
RR : NODE1
RR : NODE2
RR : NODE3
RR : NODE4
C : NODE0

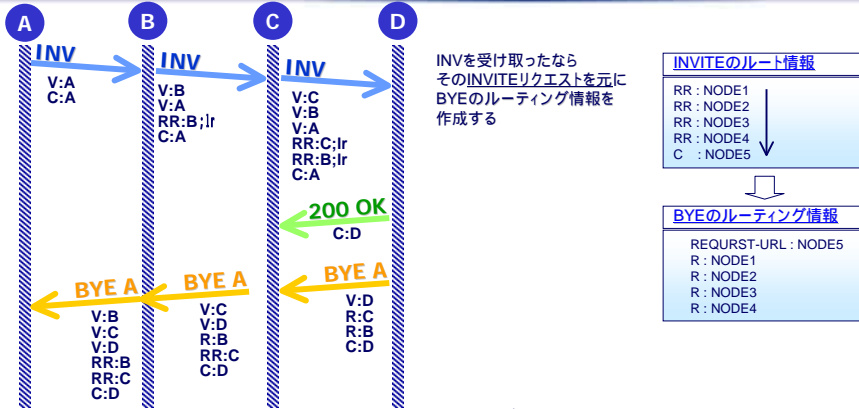
ACK/BYE
REQUST-URL : NODE0
R : NODE4
R : NODE3
R : NODE2
R : NODE1

### ルーティング情報の作り方

- ContactノードをリクエストURLにする。
- Record-Routeヘッダの下から順に上へ向かってRecordヘッダへコピーする。

# LOOSE ROUTING

パターン2 (Calleeから切断)



ルーティング情報の作り方  
ContactノードをリクエストURLにする。  
Record-Routeヘッダの上から順に下へ向かってRecordヘッダへコピーする。

# パート2

## パート2

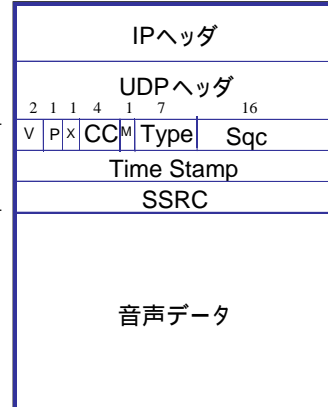
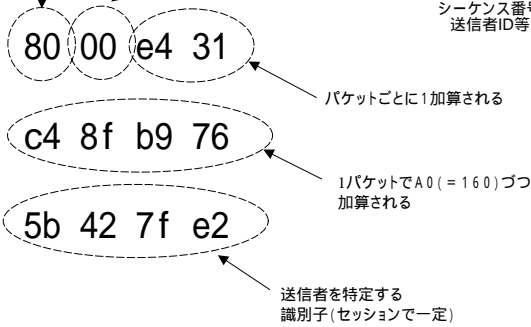
音声信号処理とパケット化



# RTP

## RTPヘッダ12バイト

Ver=2.0  
 Padding=0  
 Extension=0  
 Marker=0(フレームの切れ目ではない)  
 Type=0 PCM μ-law



- Type: 0 PCM μ-law
- 2 G721
- 8 PCM a-law
- 9 G722
- 26 JPEG Video
- ...

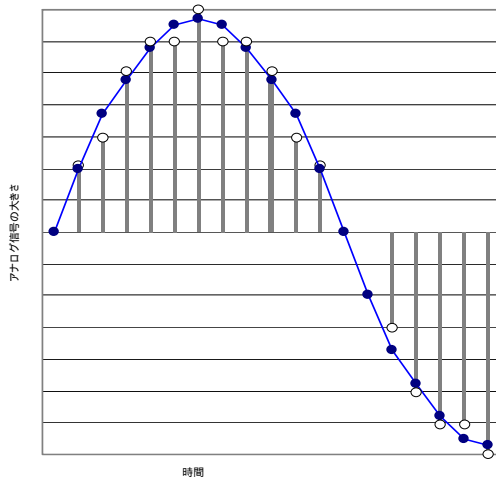
# RTPパケットダンプ

```

0000 00 90 99 20 35 f7 00 a0 de 0a 24 96 08 00 45 00 ... 5...$....E.
0010 00 c8 ab 28 00 00 40 11 ff 40 ca e5 9c f6 ca e5 ...(.@.@.....
0020 9c fa 13 8c 25 be 00 b4 b5 9c 80 00 e4 31 c4 8f ...%.1..
0030 b9 76 5b 42 7f e2 4d 4e 4f 51 53 56 58 5b 5d 60 .v[B..MNOQSVX[`]
0040 64 68 6c 70 78 7d fa f7 ef ee ec eb ea e9 ea e9 dh lpx}.....
0050 eb ea ec ec ee ef f1 f3 f5 f8 f8 fc fe 7e 7d 7d .....~}}
0060 7b 7a 78 79 76 78 76 77 77 75 78 77 79 77 77 79 {zxyvxxvwxwxywxy
0070 79 79 78 7a 7a 7d 7a 7c 7b 7c 7c 7e 7c 7e 7d 7c yxzcz|{|~|~|
0080 7e 7b 7c 7a 7a 79 79 78 75 74 6f 6e 6c 68 64 5f ~{|zzyxutonlhd_
0090 5c 59 55 53 52 52 54 56 59 5e 63 6d 7a f3 e9 e2 ¥YUSRRTVY^cmz...
00a0 dd da d7 d5 d4 d3 d1 d1 d1 d2 d2 d3 d4 d5 d7 d8 .....
00b0 da dc de e0 e3 e7 ea ed f0 f6 fa fd ff ff f9 e9 .....
00c0 db d3 d0 cf ce cf d1 d5 d9 de e6 ef 7b 6c 63 5d .....{lc}
00d0 5a 57 54 52 50 4f                               ZWTRPO
    
```

無圧縮だが、見慣れたパターンではない、wavファイルの符号化方法とはちがう。

# サンプリング



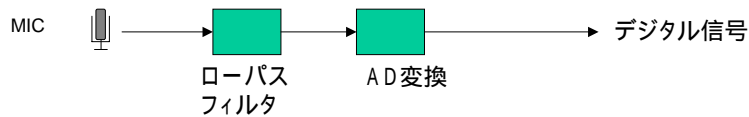
横軸のきめの細かさ(サンプリングレート)  
 1秒間あたり何データサンプルするか  
 電話 8000サンプル毎秒  
 CD 44100サンプル毎秒  
 サンプリング周波数の半分の周波数成分  
 まで再現できる(ナイキスト、シャノン、築谷)

縦軸のきめの細かさ(語長)  
 1サンプルあたり何ビット割り当てるか  
 電話 8ビット毎サンプル  
 CD 16ビット毎サンプル

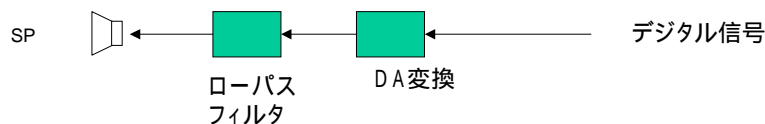
1秒間に何ビット伝送しなければならないか  
 電話  
 8Kサンプル毎秒 × 8ビット毎サンプル  
 = 64Kビット毎秒  
 CD  
 44.1Kサンプル × 16ビット × 2チャンネル  
 = 約1.5Mbps(1倍速CD)

# AD変換

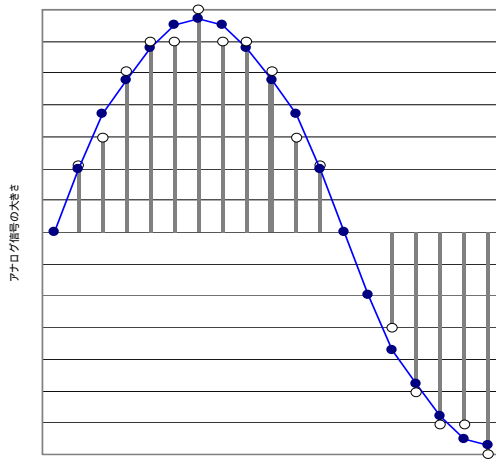
サンプリング周波数の1/2の周波数以上の成分が含まれると  
 雑音になる(エイリアシング)ので、事前に高域成分をLPFで除去する



デジタル信号をアナログ電圧に変換しただけでは、がたがたの階段  
 状の波形になり音質が悪い。波形を平滑化するために高域成分を  
 LPFで除去する



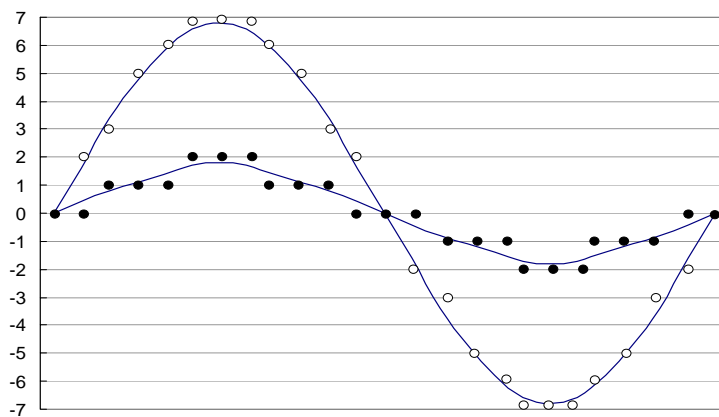
# 符号の種類



オフセット 2進	2の 補数	折り返し 2進	折り返し2進 (反転)
1111	0111	0111	1000
1110	0110	0110	1001
1101	0101	0101	1010
1100	0100	0100	1011
1011	0011	0011	1100
1010	0010	0010	1101
1001	0001	0001	1110
1000	0000	<u>0000</u>	<u>1111</u>
		1000	0111
0111	1111	1001	0110
0110	1110	1010	0101
0101	1101	1011	0100
0100	1100	1100	0011
0011	1011	1101	0010
0010	1010	1110	0001
0001	1001	1111	0000

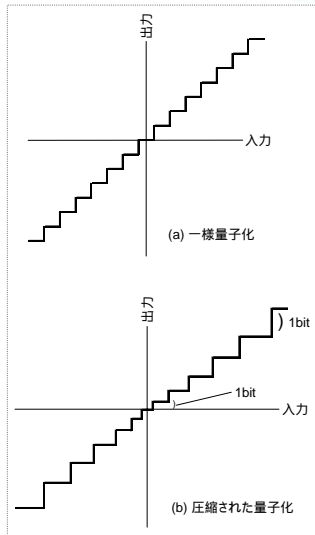
WAVファイル オフセット2進  
PCM伝送 折り返し2進(反転)

# 量子化雑音

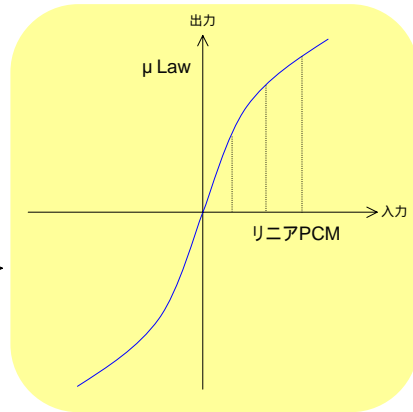


量子化雑音により弱信号ほど  
S/N比が悪くなる

# 一様量子化と圧伸量子化



μ-law (日・米)  
a-law (欧)



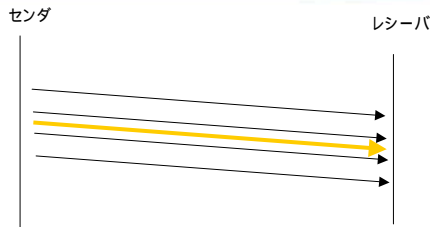
# μ-law

入力レベル範囲	ステップ サイズ	折 線 番 号	符号パターン 12345678	出力レベル
7903 - 8159	256		10000000	8031
7647 - 7903	"	"	10000001	7775
7391 - 7647	"	"	10000010	7519
7135 - 7391	"	"	10000011	7263
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
35 - 39	4		11101110	37
31 - 35	"	"	11101111	33
29 - 31	2		11110000	30
27 - 29	"	"	11110001	28
25 - 27	"	"	11110010	26
23 - 25	"	"	11110011	24
21 - 23	"	"	11110100	22
19 - 21	"	"	11110101	20
17 - 19	"	"	11110110	18
15 - 17	"	"	11110111	16
13 - 15	"	"	11111000	14
11 - 13	"	"	11111001	12
9 - 11	"	"	11111010	10
7 - 9	"	"	11111011	8
5 - 7	"	"	11111100	6
3 - 5	"	"	11111101	4
1 - 3	"	"	11111110	2
0 - 1	1	"	11111111	0

**実装**  
16bitのリニアPCMでサンプリング  
8bit符号に圧縮

語長14ビットで一様に量子化したものを語長8ビットに圧縮された符号に変換する。

# RTCP



RTPで使っているポート(ソース、ディスタネーション)にそれぞれ1加算したポート番号がRTCPのポート番号

定期的にRTCPパケットを送り出す。RTCPには

- ・センドレポートSR
- ・レシーバレポートRR
- ・ソースディスクリプションSDES
- ・BYE
- ・アプリケーションスペック

などの種類がある。ひとつのパケットに複数のRTCPを含めることもできる

IP電話では、定期的に飛び交う(5秒に一度程度)RTCPには、SR/RR/SDESが含まれている。

IPヘッダ
UDPヘッダ
RTCPセンドレポート
RTCPレシーバレポート
RTCPソースディスクリプション

# RTCP

- SR

これまで送信したパケット数  
これまで送信したオクテット数  
その実時間 (NTPタイムスタンプ)

- RR

欠落パケット数、欠落率  
受信済み最大シーケンス番号  
ジッタの見積もり  
» など

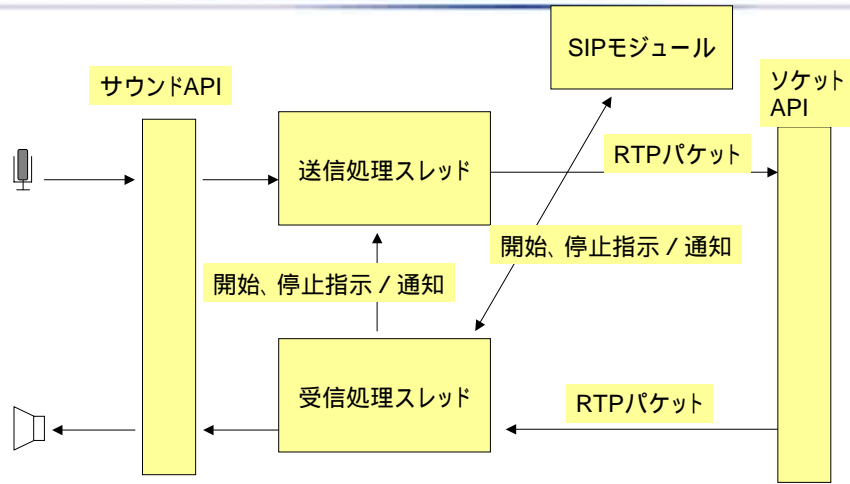
- SDES

参加者のCNAME (表示可能なアドレス)。例  
term183@202.229.156.221

- BYE

セッションを終了する。終了理由が設定できるために、SSRCが衝突したらBYEを送って、新しいSSRCに更新してもらうこともできる。

## ソフトフォン実装



## サウンドAPI (Win32の例)

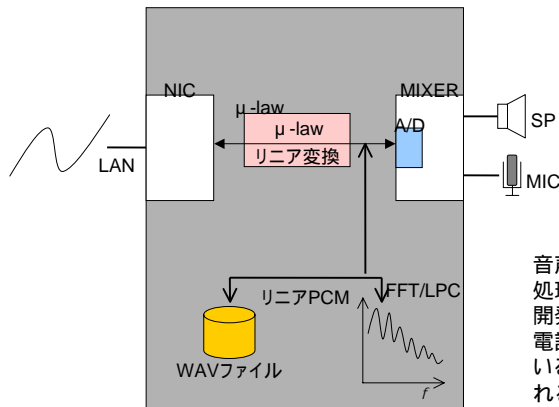
### • 録音

- デバイスの生成と例外ハンドラの登録waveInOpen
- リングバッファの登録waveInPrepareHeader
- 書き込み可能バッファエリアの指定waveInAddBuffer
- 録音の開始 waveInStart
- バッファに音声を書き込まれると、例外が発生
- 例外ハンドラでデータを引き上げた後、waveInUnprepareHeaderで再利用可能領域に指定

### • 再生

- デバイスの生成と例外ハンドラの登録waveOutOpen
- 音声データをバッファに書き込みwaveOutPrepareHeader
- 再生スタート(キューに積む) waveOutWrite
- 再生が完了すると例外ハンドラが起動する
- 例外ハンドラで、バッファを無データ領域に指定するwaveOutUnprepareHeader

## 端末内での音声情報の扱い



音声信号が無圧縮なので、音声信号処理技術を使ったアプリケーション開発が容易になります (ex. リカちゃん電話)。しかし、パソコンで用いられているデジタル音声信号とVoIPで使われる音声信号は異なるフォーマットです。両者間で音声コンテンツをやりとりする場合にはコンバータの実装が必要です。

## まとめ

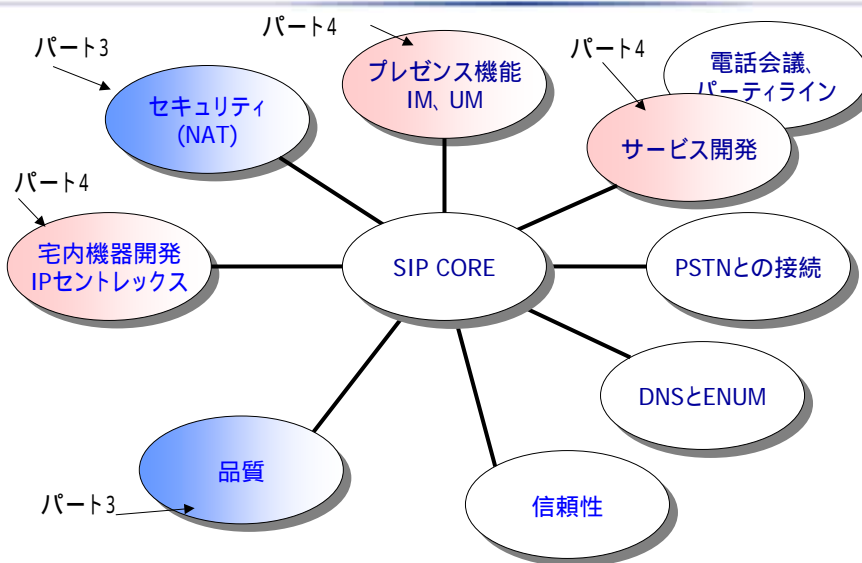
- デジタル化
  - 標本化
  - 量子化
  - 符号化
- 実装
  - Win32 APIの使い方
- バッファリング
  - ジッタの影響を回避

## パート3

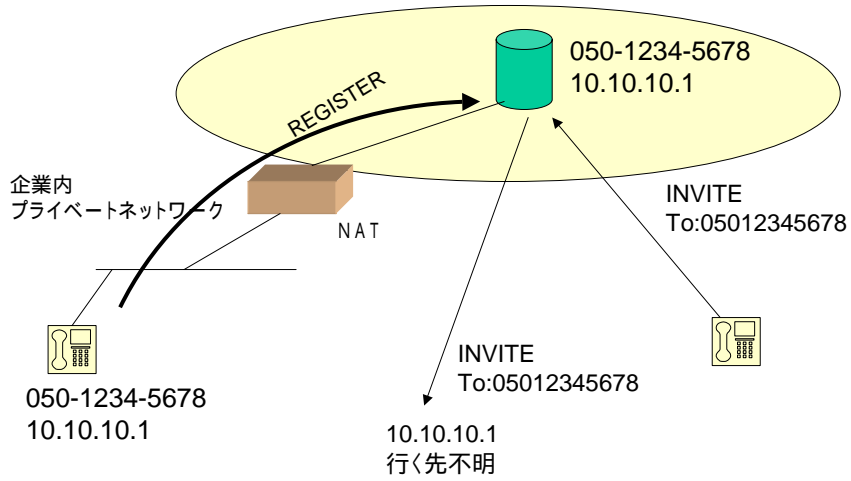
### パート3

## SIPとVoIPをとりまく諸問題

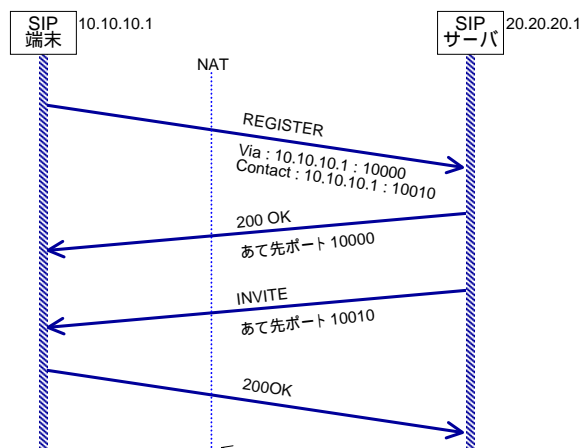
## 諸問題一覧



# NAT問題

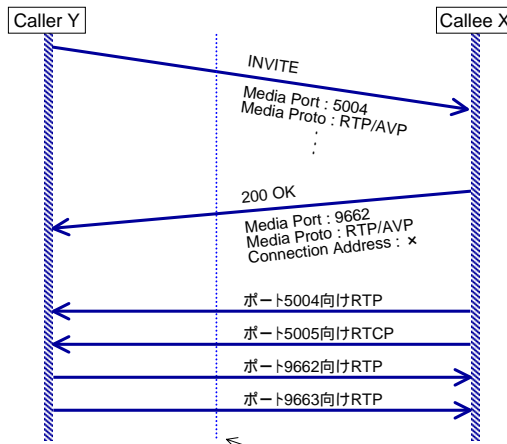


# NAT1



NATで200 OKやINVITEを通過させたいなら、  
事前のREGISTERのVia : ヘッダや  
Contact : ヘッダを認識できる機能が必要

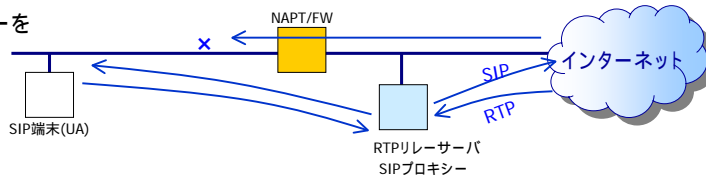
# NAT2



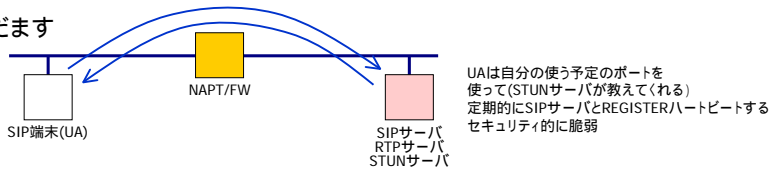
NATでRTPを通過させたいなら  
 事前に通過したINVITEと200 OKのSDPを  
 認識し、相手IPアドレス(200 OKの発元ではないかも?)  
 と自サイト側ポート番号を知っておく必要がある。

# NAT3

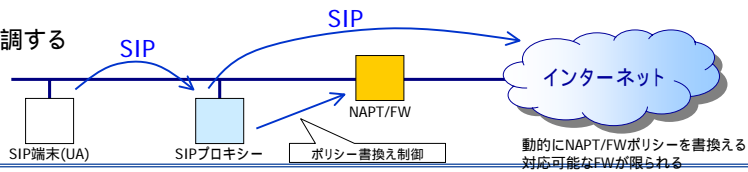
## ◆プロキシを使う方法



## ◆NAPTをだます方法

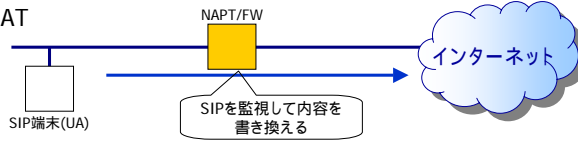


## ◆FWと協調する方法

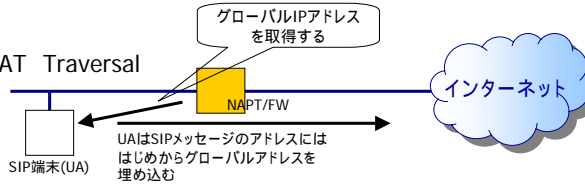


# NAT4

## SIP対応NAT

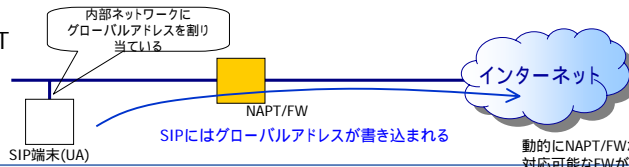


## UPnP NAT Traversal

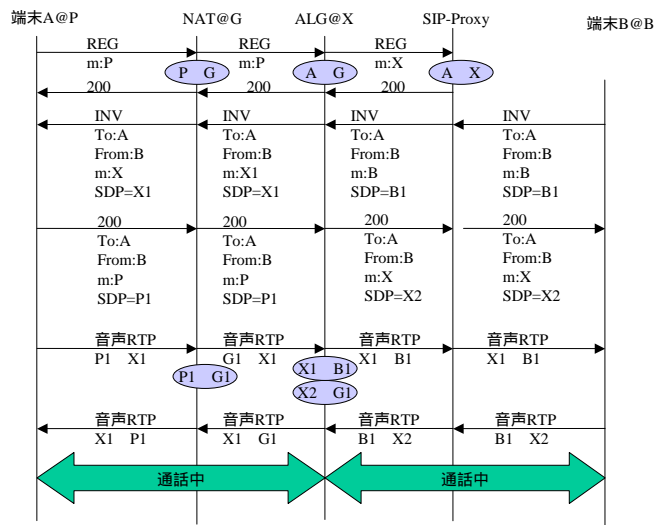


WARP STARTが NAT Traversalを対応した

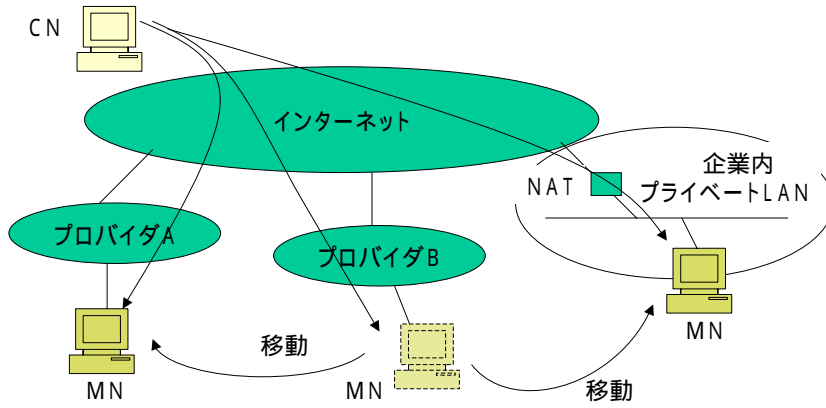
## GapNAT



# ALG SBCによるNAT越え



## Mobile IPによるNAT越え

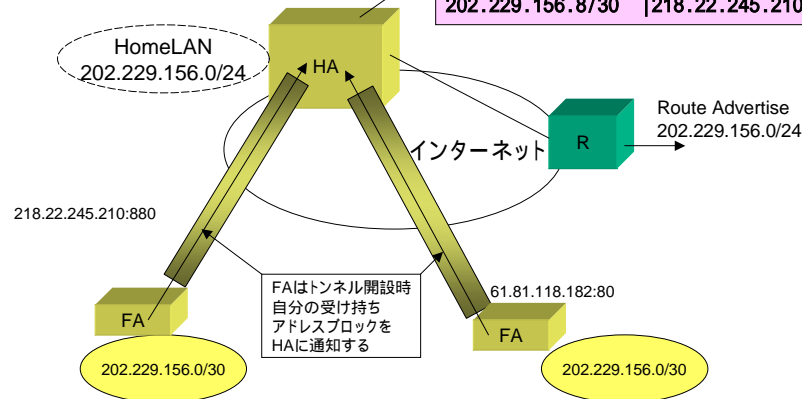


## トンネリングプロトコル

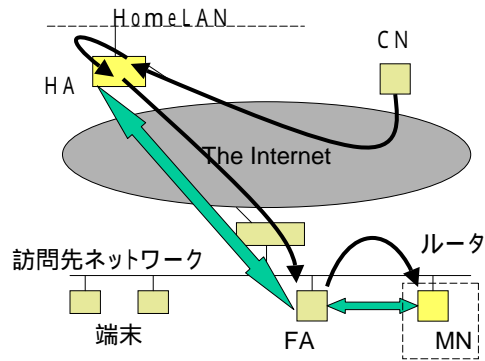
トンネル端点とCoAから  
 フォワーディングキャッシュを生成  
 グローバルから受け取ったパケットは  
 フォワーディングキャッシュに登録されていれば  
 該当トンネルに押し込められる

Forwarding Cache

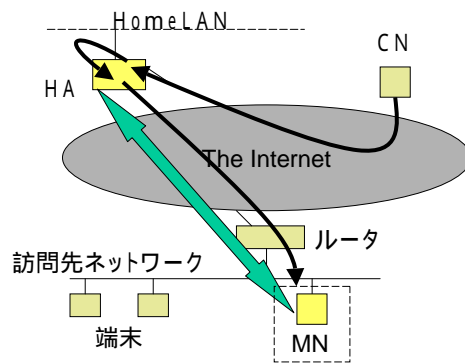
Destination	Care of Address	state
202.229.156.0/30	61.81.118.182:80	alive
202.229.156.8/30	218.22.245.210:880	alive



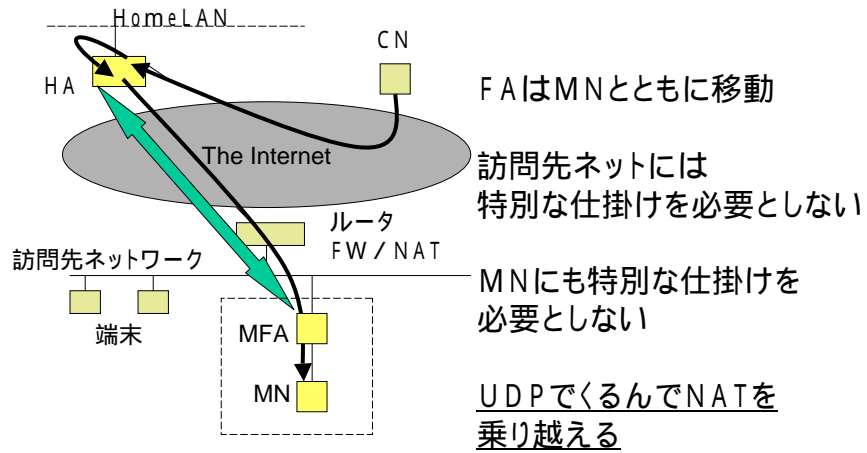
# システムフレームワーク FAモード



# システムフレームワーク CoAモード

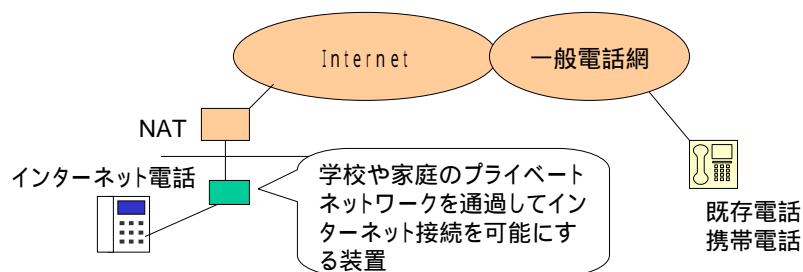


## Proxy MIP ( 怪しいNTTPC方式 IP Warp® )



## ネットワークSI~モバイルIPとの組み合わせ

### • モバイルIPとの組み合わせ



## 品質、信頼性

### ◆品質

- 通話品質を保证するために、ネットワーク品質はいかにあるべきか

### ◆信頼性

- 宅内サイトからソフトスイッチまでの経路が切れた場合、ソフトスイッチがフェイルした場合、誰が通話を救済すべきか

## 品質分析

### ◆パケットロス

- ビットエラーによるパケット廃棄(ランダムエラー)
- 輻輳によるパケット廃棄(バーストエラー、ジッタをとまなう)

### ◆遅延

- 音声おくれ

### ◆ジッタ(含パケット逆転)

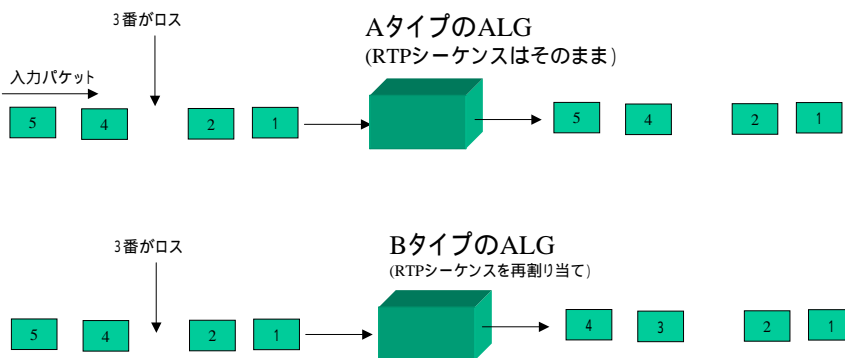
- 音声とぎれ
- パケットロスに似ている

## 音質劣化の原因

- ネットワーク輻輳
  - 帯域不足
    - ATMなどのQoS(最大6Mbpsの50%保証)でVoIPは1ch=100Kbpsだとすれば、同時通話可能数は30人?それとも60人?
  - SIP高負荷によるRTPパケット廃棄
    - REGISTERの集中によるデータベース高負荷
    - OPTIONの集中によるネットワーク高負荷
- SIPとRTPの関係
  - QoSつきネットワークでは、SIP>RTP>その他という優先順位となるが、優先順位の高いプロトコルに障害があると、低順位のプロトコルにも影響が及ぶ。(例)SIPが高負荷のためにRTPが廃棄
- パケットロス保証のバグ
  - ALGではパケットロスが検出されるとどうするべきか?
    - ALGではスキップした番号はスキップしたままにすべき。

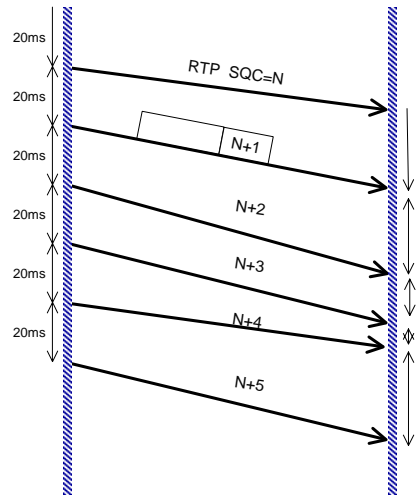
## ALG越えのRTPシーケンス番号

Aタイプ、Bタイプのどちらが望ましいだろうか？



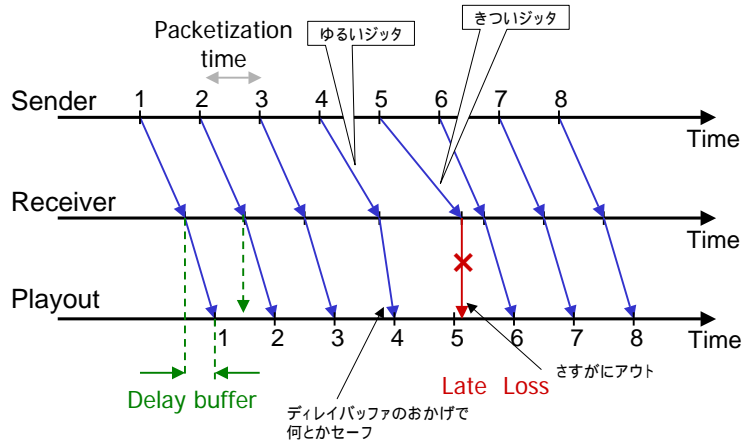
BタイプのALGではパケットロスの保障が効かず、音質の劣化になる  
わかりやすくするためにロス数1パケットとしているが、実際には  
バースト的に大量ロスが起こる。

# パケットジッタ



遅延のゆらぎ(パケットジッタ)により  
受信側でのパケット到着間隔が  
不規則になる

# 受信バッファの必要性



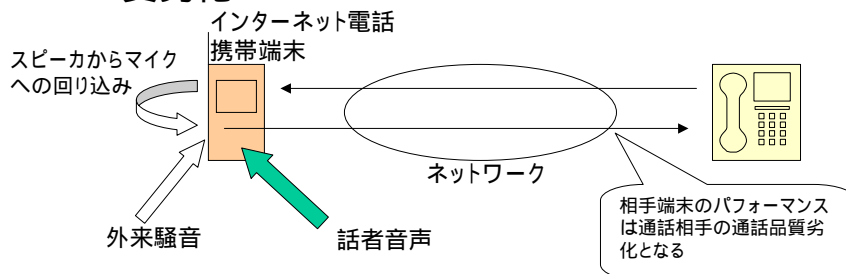
ネットワークの遅延揺らぎのために、前のパケットの再送が完了するまでに次のパケットが到着  
していなければならない。

## バッファリングの実装例

- 遅延再生
  - 最初のパケットが到着しても、すぐにはキューに入れず、次のパケットが到着してから再生を開始させる
  - その後は、パケットが到着するたびに即座にバッファに音声パケットを積んでいく
- 割り込みハンドラを利用する方法
  - リングバッファに音声を書き込んだ後に最初のバッファの再生を開始する。
  - バッファの再生が完了するたびに呼び出される例外ハンドラがソケットからひとつパケットを読み出して、再生バッファを埋める
- ダイナミックにバッファの深さを変化させる
  - 深いバッファは音声遅延の原因になる
  - 浅いバッファは軽いジッタにも耐え切れない
  - パケット到着間隔を常時計測し、揺らぎにより遅延バッファ時間を変化させる

## 音響問題

- 現在の電話端末の問題点
  - 遠端漏話による通話品質の劣化
  - 騒音環境下におけるモバイル端末での通話品質劣化

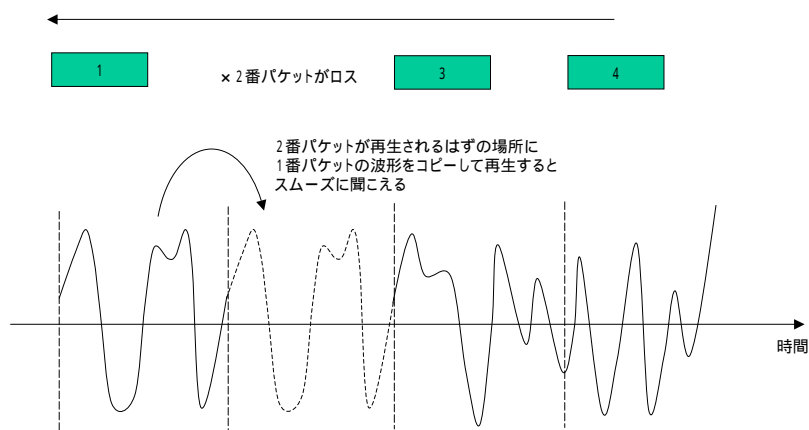




## 音質劣化の原因

- ネットワーク輻輳
  - 帯域不足
  - SIP高負荷によるRTPパケット廃棄
    - REGISTERの集中によるデータベース高負荷
    - OPTIONの集中によるネットワーク高負荷
- SIPとRTPの関係
  - QoSつきネットワークでは、SIP>RTP>その他という優先順位となるが、優先順位の高いプロトコルに障害があると、低順位のプロトコルにも影響が及ぶ
  - SIPが高負荷のためにRTPが廃棄される
- パケットロス保証のバグ
  - ALGではパケットロスが検出されるとどうするべきか？
    - ALGではスキップした番号はスキップしたままにすべき。

## パケットロス保障 (PLC)



- 口頭説明

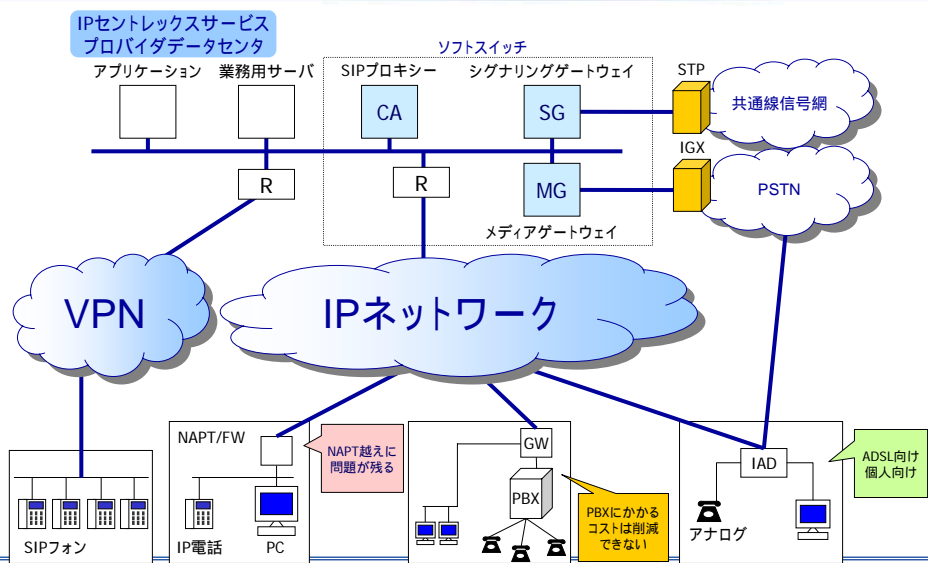
パート4

サービス化のためのシステム

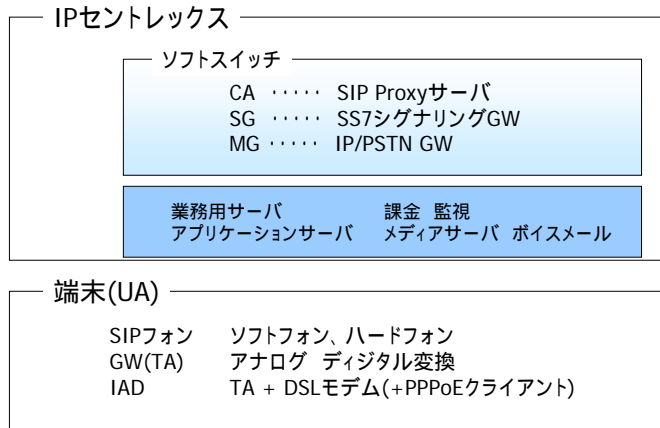
## サービス要件

- 単なる電話として使いたい
  - 認証、顧客管理
- 一般電話との通信
  - 一般電話へ
  - 一般電話から
- PBXの代わりになるのか？
  - IP-PBX
- IPならではのサービスは
  - プレゼンス
  - メッセンジャー
  - 音声サーバ
  - CTI

## システムアーキテクチャ

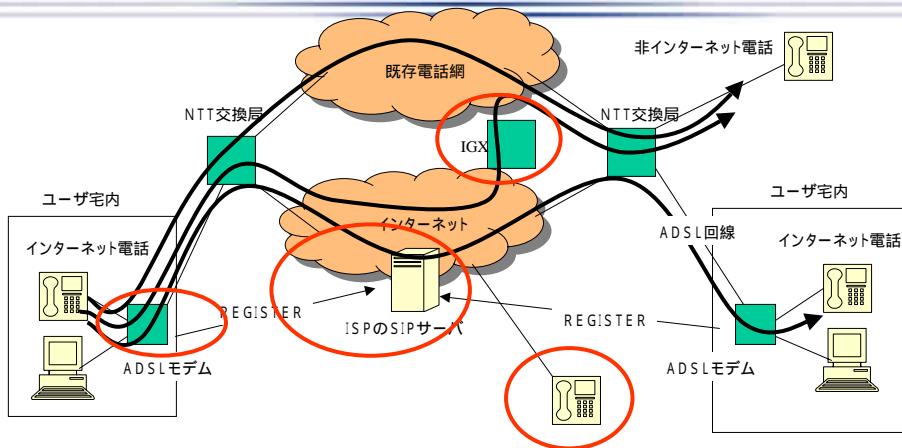


# IPPBXシステムアーキテクチャ



CA Call Agent  
 UA User Agent  
 IAD Integrated Access Device

# インターネット電話システム構成例



: 既存電話同士の従来型の通話。インターネットを利用しない。  
 : インターネット電話から従来型電話への接続。着信最寄のエリアまでインターネットで接続する  
 : インターネット電話同士の通話。既存電話網を利用しない。

### 必要とされる機能例

- 外線ゲートウェイ発信
- グループ鳴動
- コールピックアップ
- 不在転送
- 話中転送
- 不応答転送
- 仲介転送
- 短縮ダイヤル発信
- 識別リング
- 外線への発信/着信
- 発ID通知
- 発ID非通知
- 呼転送(仲介転送)
- コールピックアップ後の呼転送
- 内線キャンブオン
- アドオン会議(三者)
- コールピックアップ後のアドオン会議
- 利用停止(有効フラグOFF)
- 表示名称(ディスプレイネーム)
- 発信規制
- 最大コンタクト数制限
- 最大同時発信呼数制限
- 障害時アナウンス切替

- REFERの目的
  - REFER受信先があらたなリクエスト(デフォルトはINVITE)を発行するように促す。
  - 暗黙的にSUBSCRIBEを発行した効果ももつ
- 主な利用
  - コール転送
    - 不応答転送
    - 応答転送

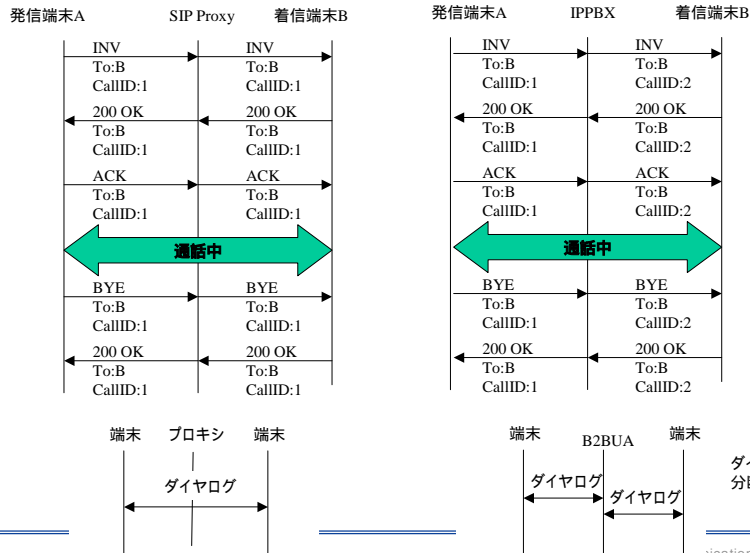
REFERを受信したら、  
新たなINVITEをするかどうか  
の判断は重要な問題である。



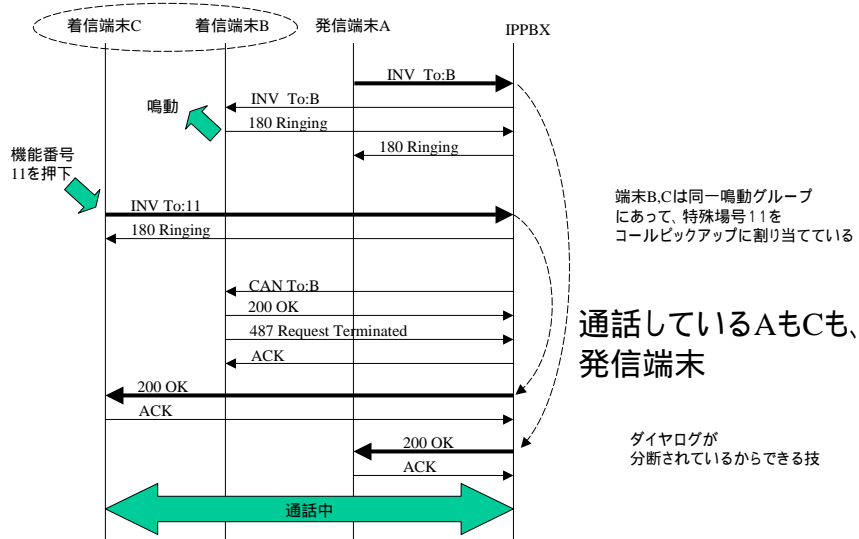
# IPPBXはB2BUA

SIP Proxyサーバの  
ワンコールシーケンス

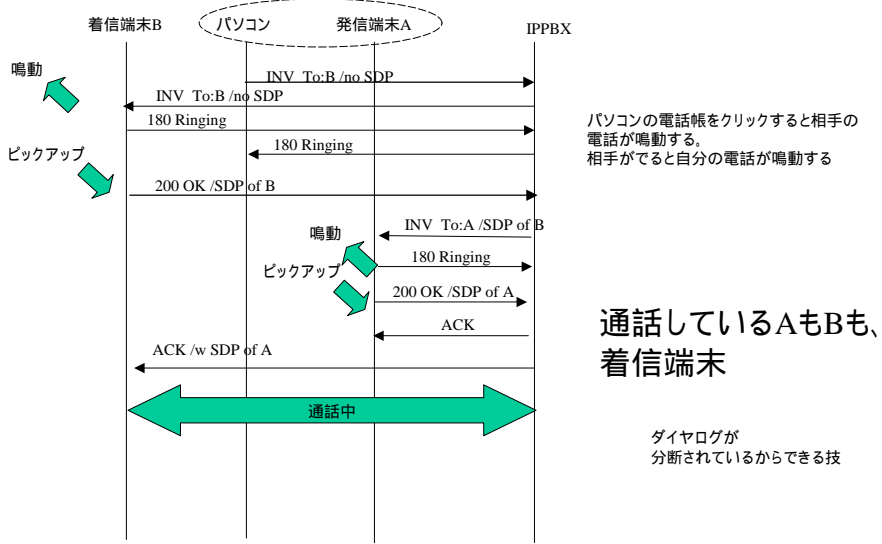
IPPBXの  
ワンコールシーケンス



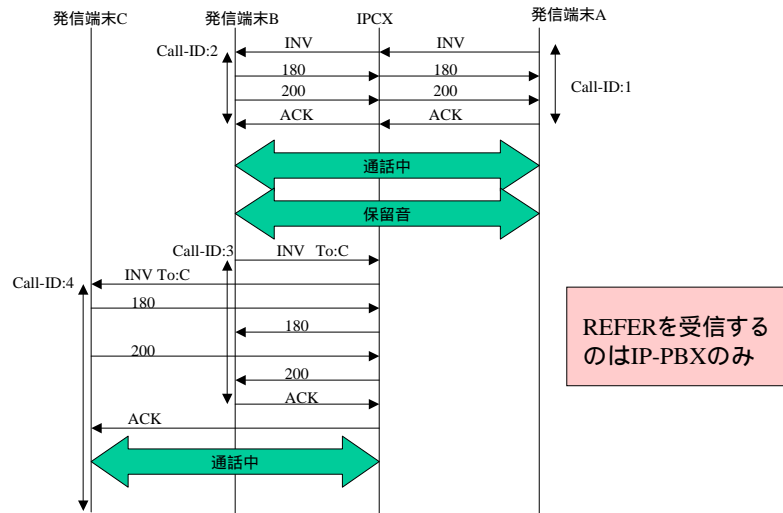
# コールピックアップ



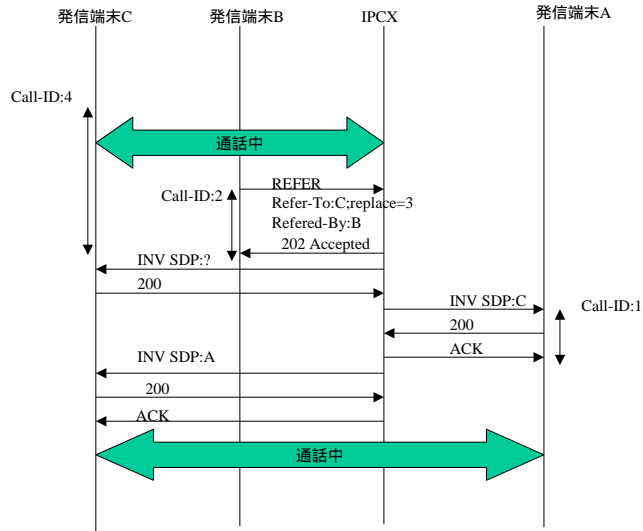
### 3 PCC



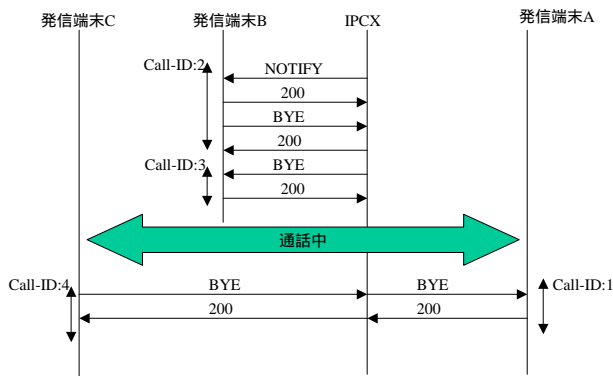
### IPPBXの呼転送 (仲介転送)



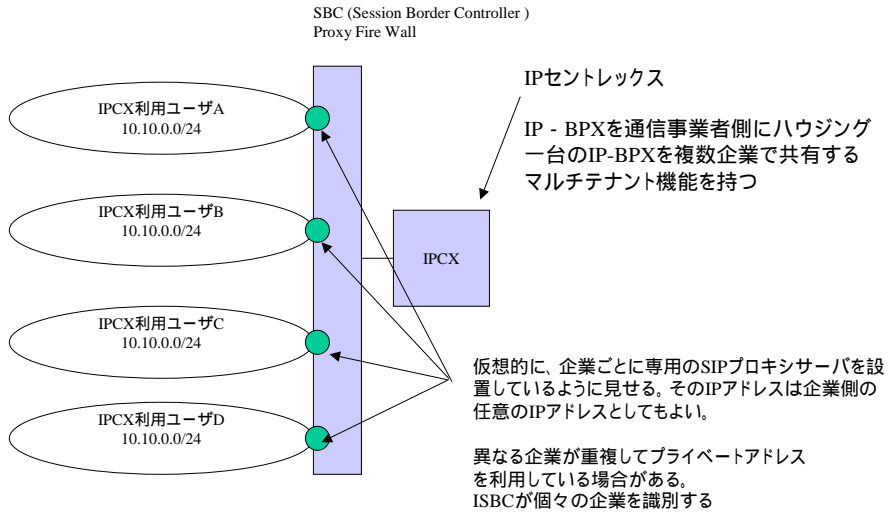
# 呼転送 (仲介転送) つづき



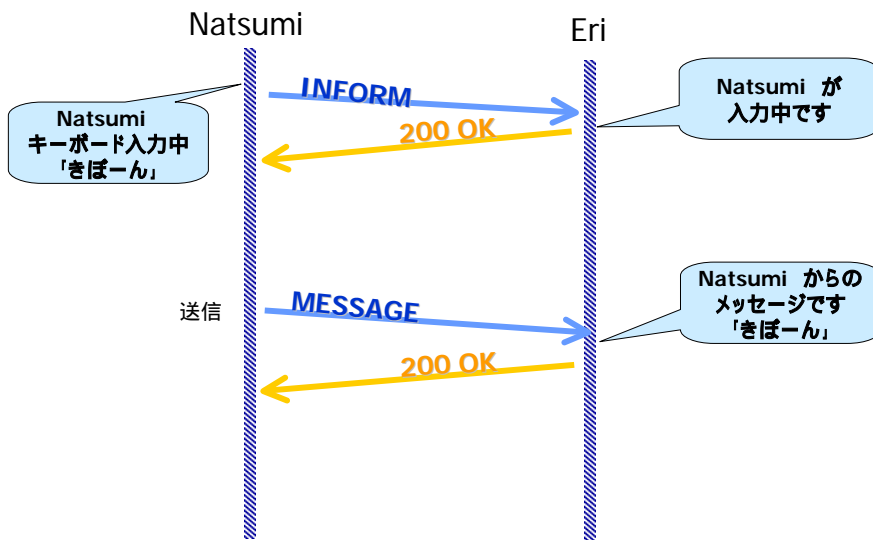
# 呼転送 (仲介転送) つづき 2



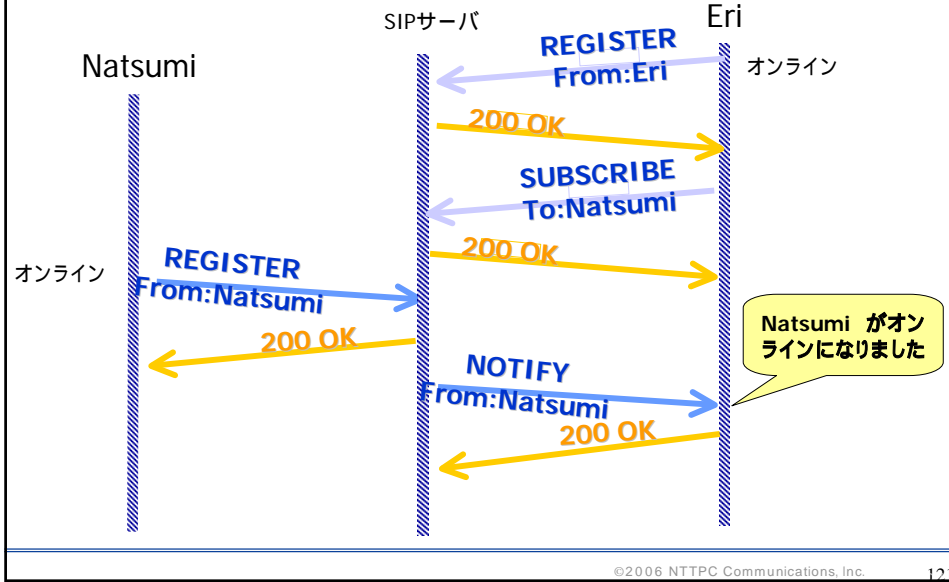
# Session Border Controller



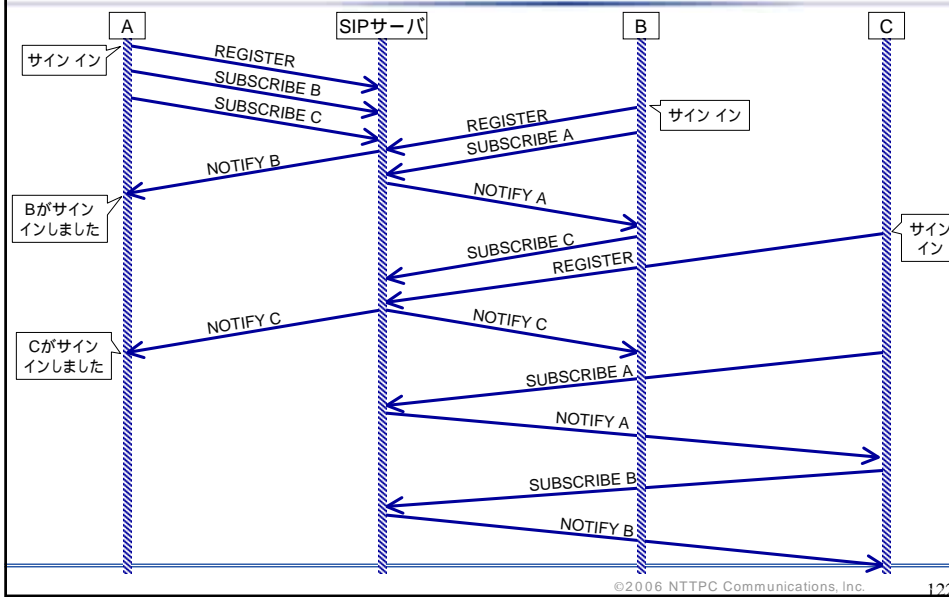
# その他のSIPメッセージ 代表的なシーケンス



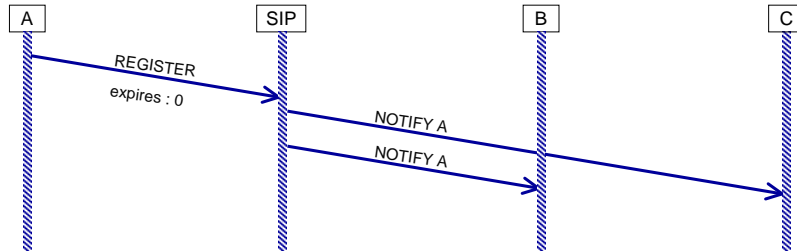
## 代表的なシーケンス(プレゼンス)



## プレゼンス機能(1)



## プレゼンス機能(2)



一体NOTIFYにどのような情報を入れてやれば  
MSN Messengerは反応してくれるのだろうか?



REGISTERやSUBSCRIBEはMSN Messengerが  
発信していたのでフォーマットをまねていけばよかった。  
NOTIFY(そもそもNOTIFYで良いのか?)はSIPサーバが  
発信するもので、Messengerからは送信されない。

## プレゼンス機能(3)

### 参考文献

- draft-ietf-simple-presence-01.txt (2001年7月)
  - ◆プレゼンス通知はNOTIFYリクエストを使う
  - ◆SUBSCRIBEリクエストの中に、NOTIFYにより自分が受け入れられるMIMEタイプをACCEPTヘッダに格納する
  - ◆例題ではapplication/xpidf+xmlが使われている

MSN MessengerのSUBSCRIBEにはAcceptヘッダがなかった
- draft-ietf-simple-presence-07.txt (2002年5月)
  - ◆NOTIFYのMIMEタイプがapplication/cpim-pidf+xmlに変わっている
- draft-ietf-impp-cpim-pidf-05.txt
  - ◆cpim-pidf+xmlの使い方

結局MSN Messengerはcpim-pidf+xmlにどうにも反応せず

## サインインの通知方法

http://www.cs.columbia.edu/sip/drafts/messenger.txt

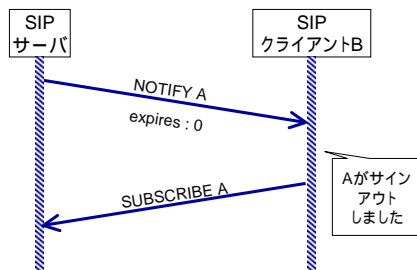
```
NOTIFY sip:xiaotaow@128.59.19.27:5060 SIP/2.0
Via: SIP/2.0/UDP 128.59.19.251:13170
From: sip:william@conductor.cs.columbia.edu;tag=d2dd3c15-b762-486b-8911-25ed3a3bd975
To: sip:xiaotaow@cs.columbia.edu
Call-ID: 330938581@128.59.19.27
CSeq: 1 NOTIFY
Contact: <sip:128.59.19.251:13170>
User-Agent: Windows RTC/1.0
Content-Type: application/xpidf+xml
Content-Length: 353

<?xml version="1.0"?>
<DOCTYPE presence
PUBLIC "-//IETF//DTD RFC:xxxx XPIDF 1.0//EN" "xpidf.dtd">
<presence>
<presentity uri="sip:xiaotaow@cs.columbia.edu;method=SUBSCRIBE" />
<atom id="1003">
<address uri="sip:128.59.19.251:13170;user=ip" priority="0.800000">
<status status="open" />
<msnsubstatus substatus="online" />
</address>
</atom>
</presence>
```

Messengerが発したNOTIFYリクエストのダンプ  
(どうやればNOTIFYが出るのかは不明)

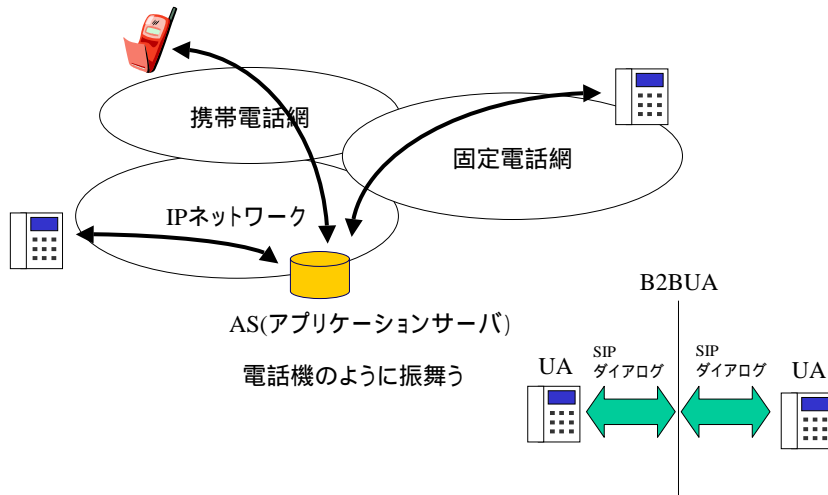
## サインアウトの通知方法

<status status = "closed">ではダメでした。  
結局以下の方法をとりました。

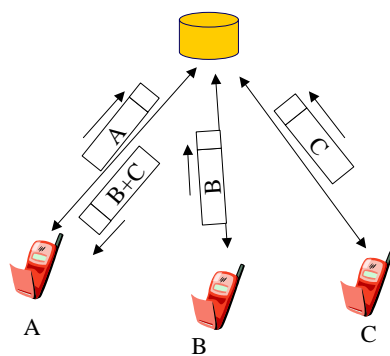


NOTIFYのメッセージボディではなく  
メッセージヘッダのexpiresを0にセットした  
ものを送ります。

## 電話機以外の端末 (AS)



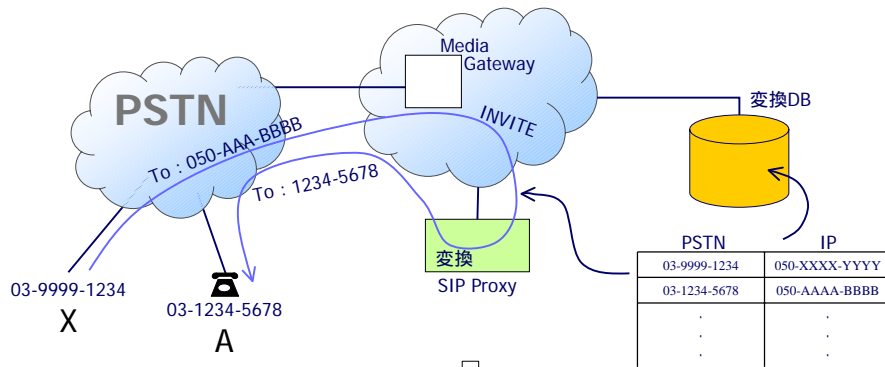
## (例) 電話会議システム(Party Line)



端末ABCそれぞれからやってきたパケットの音声を合成して、各端末に送り返す。

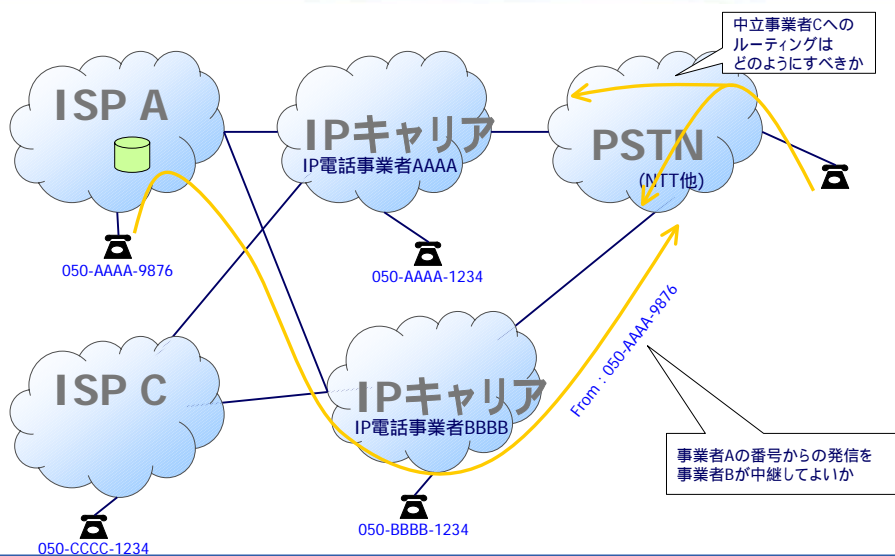
自分の声が返ってこないように、Aに送り出す音声パケットには、Aの音声成分が含まれないように合成する。

## (例) 番号変換アプリケーション

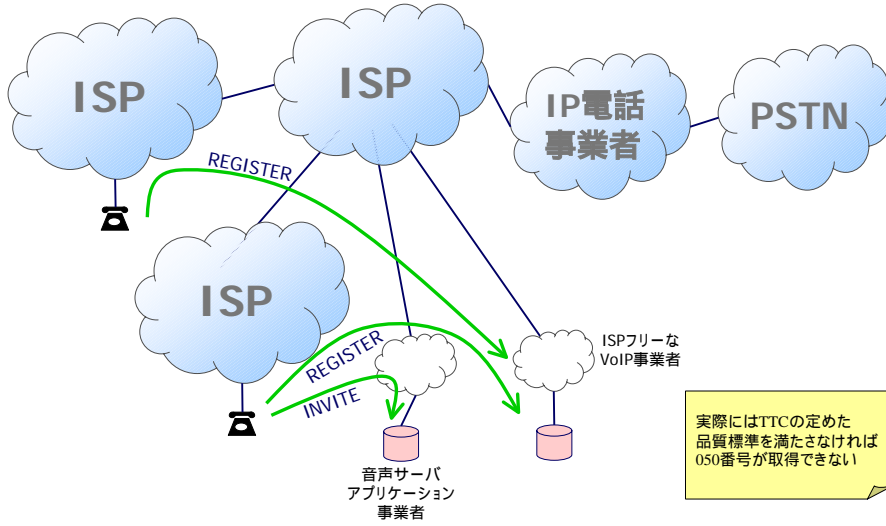


XはAの本当の電話番号を知らなくても  
Aに電話をかけられるサービス

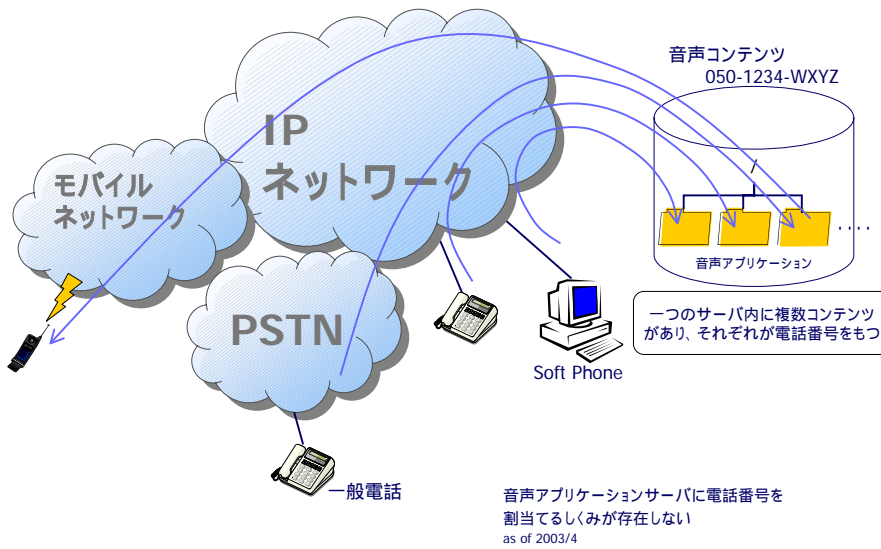
## 事業者間接続



# ISPフリーなVoIP事業者



# 音声サーバアプリ

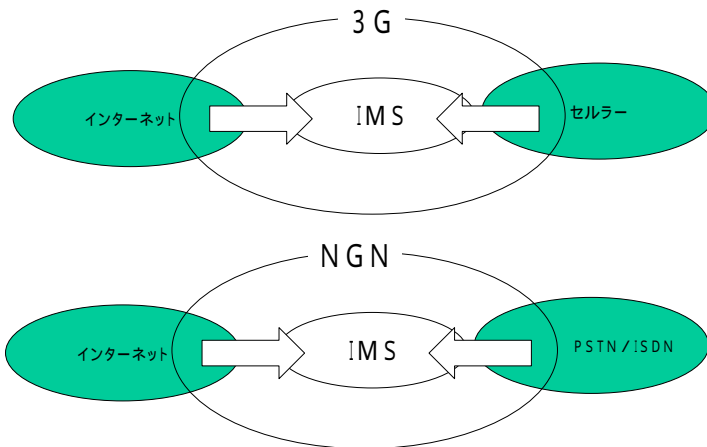


# パート5

## パート5

3GPP/IMS/SIP/NGN/RFC3261

# 3GとNGN

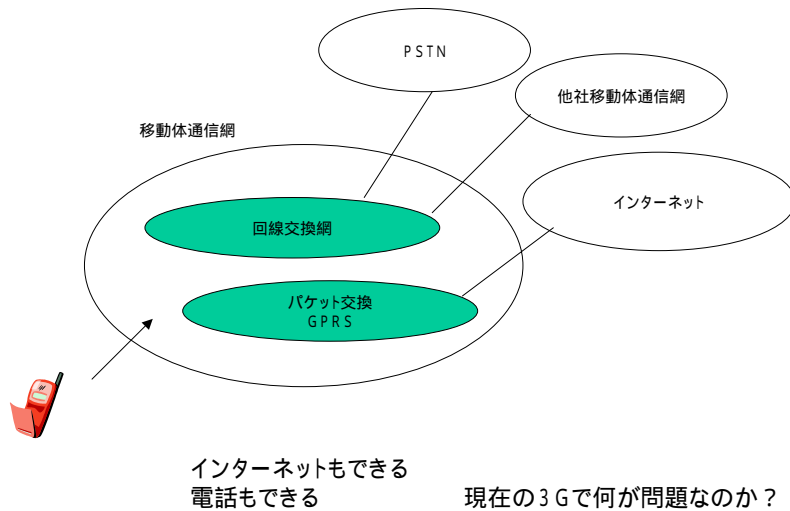


- 携帯端末ひとつで
    - 電話
      - 音声通話
      - テレビ電話
    - インターネットサービス
      - 電子メール
      - Web閲覧
      - 電話会議
      - ストリーム閲覧
- すべてを実現したい

- 3GPP
  - GSMをベースに3Gへの移行技術仕様を策定する団体
  - 日本からTTC、ARIBが加盟
- 3GPP2
  - CDMA2000をベースに3Gへの移行技術仕様を策定する団体
  - 日本からTTC、ARIBが加盟

標準化は各国の標準化組織が国内標準を策定する

## 現在のシステム



## IMSへの期待

- IPメディア統合
  - GPRSではなく回線交換だけでマルチメディア通信が可能(IPベースの電話会議、テレビ電話)
  - 端末へのアプリケーションの組み込み
  - APIの不足
- QoS
  - インターネットはベストエフォート
- 課金
  - サービスに対し課金するのに必要な課金情報の収集機能
- 相互接続性(ローミング)
- サードパーティによるアプリケーション開発のスピード化など



インターネットプロトコルを取り込みながら、

- IMSはIPベースのネットワークにする
  - IETFの成果を3GPPに焼きなおすか？
  - それともIETFで共同作業で仕様を策定するか？



後者が選択された

プロトコルは次々と生産されるために、3GPP/3GPP2はIETFと共同作業を開始した 2001年  
RFC3113/RFC3131

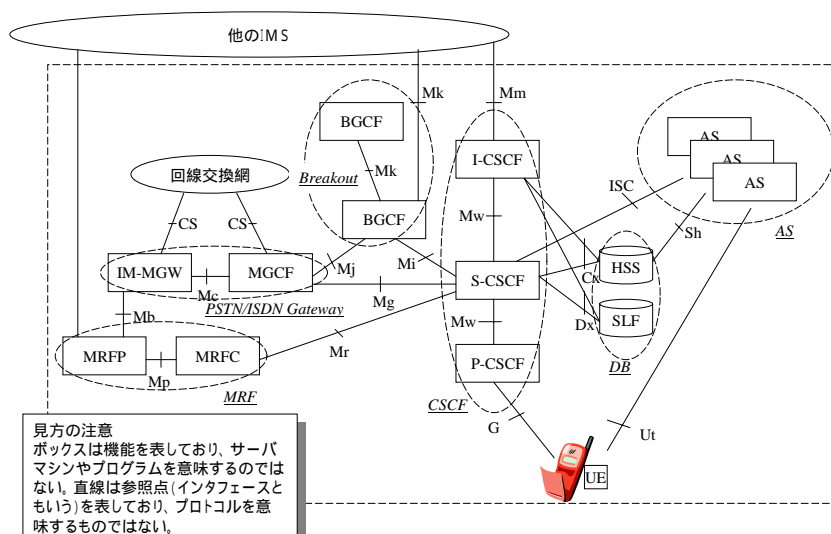
IETF,3GPP,3GPP2はリエゾンと呼ばれる代表者を立てて共同してプロトコル標準化作業に当たる  
ドキュメントはそれぞれのホームページで誰に対しても公開する

- 呼制御 (SIP)
  - RFC2543からRFC3261へ
- 認証認可課金 (DIAMETER)
  - RADIUSからDIAMETER
- QoS (COPS-PR)
  - PIB(ポリシー情報DB)とネットワークのプロトコル

## 強化されたSIP

- 認証
  - チャレンジレスポンス
- 信頼性
  - アップデートタイマー
  - PRACK
- QOS (無線区間でも利用可能な配慮)
  - 音声帯域確保
  - SIP圧縮フォーマット

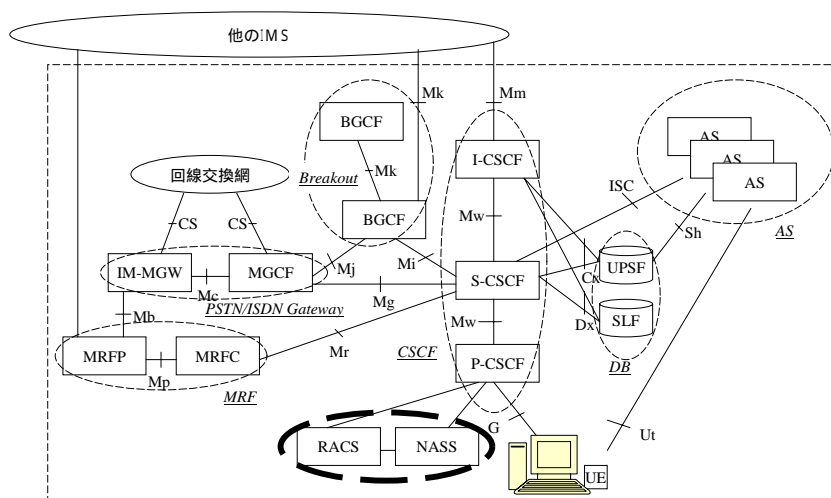
## 3G IMSアーキテクチャ



## IMSのコンポーネント

- DB(データベース)
  - HSS:加入者情報
  - SLF:加入者の現在位置、アドレスを保持
- CSCF(SIPプロキシサーバ)
  - P-CSCF:IMS端末が最初に接続するところ
  - S-CSCF:レジストラサーバ
  - I-CSCF:SIPサーバとしての対外ネットワークGW
- AS
  - SIP-AS:B2BUA,番号変換等
- MRF(メディア保存)
  - トーキー音声
- Breakout Gateway
  - BGFC:他IP網との接続点
- PSTN/ISDN Gateway
  - MGCF:共通線信号の相互接続機能
  - MGW:音声チャンネルの相互接続機能

## NGNのIMSアーキテクチャ



## NGNのレイヤ

- トランスポートレイヤ
  - NASS
    - ユーザ認証、ユーザプロフィールに基づく接続設定
  - RACS
    - ユーザプロフィールに基づくアドミッション制御、リソース制御
- サービスレイヤ
  - ISDN/PSTNシミュレーション
    - 電話サービスをそのまま実現する
  - ISDN/PSTNエミュレーション(PES)
    - 既存のネットワークサービス
      - コールウェイトイング、発番通知、電話会議、着信転送等
    - 新しいタイプの電話機
    - 新しいタイプのネットワークサービス

## NGNは必要なのか

- インターネットではだめなのか？
- NGNとは次世代フレッツ網のことなのか？
- サードパーティが端末でアプリケーションを実現できるのか
- NGNのメリットは誰が享受するのか
  - エンドユーザ
  - NGN事業者
  - xSP事業者
  - コンテンツプロバイダー
  - アプリケーション開発者

- SIPの基礎
- 音声伝送圧縮の仕組み
- サービスを実現するために乗り越えなければならない問題
  - NAT
  - 音声品質
- サービス開発
  - IP-BPX
  - AS
- NGNと3G
  - 次世代の携帯電話、固定電話ネットワークはIP化され、制御プロトコルとしてSIPが採用される