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

NTTPC COMMUNICATIONS

パート0

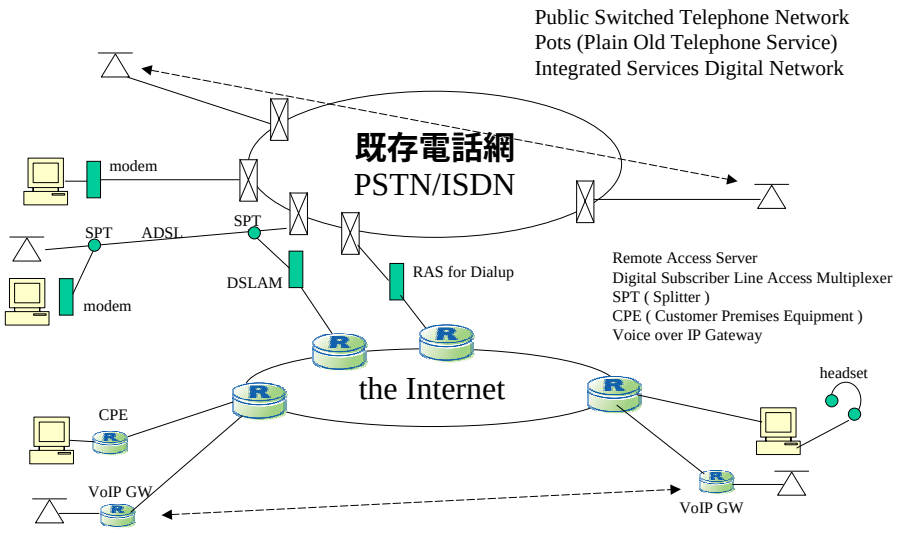
NTTPC COMMUNICATIONS

パート0

VoIPの概要と本コースの内容

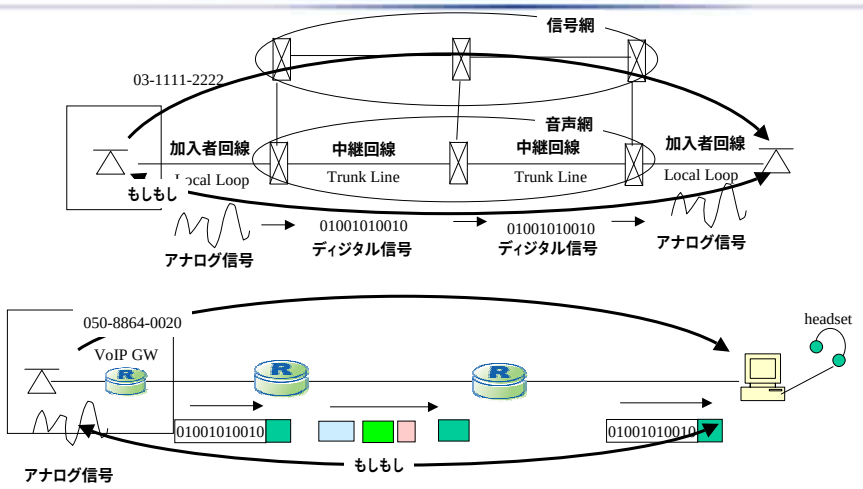
インターネットと電話網の関係

NTTPC COMMUNICATIONS

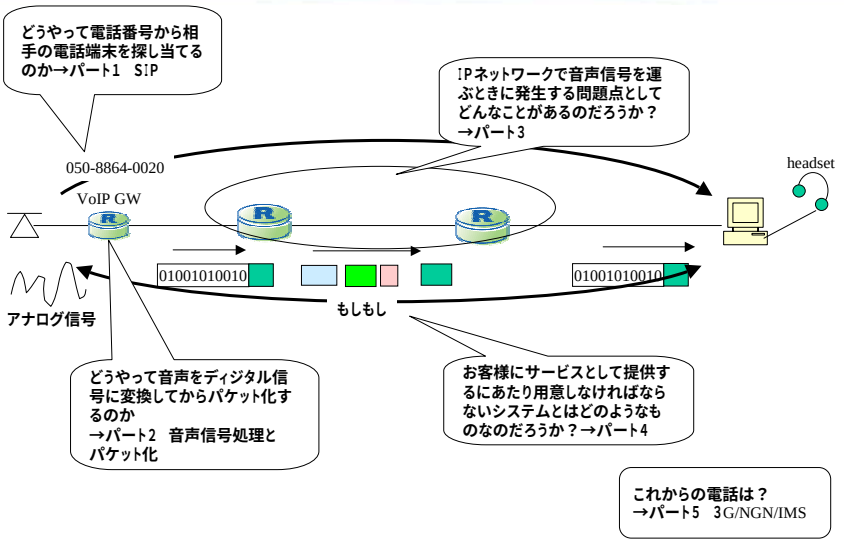


技術的観点での比較

NTTPC COMMUNICATIONS



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

NTTPC COMMUNICATIONS

- 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)

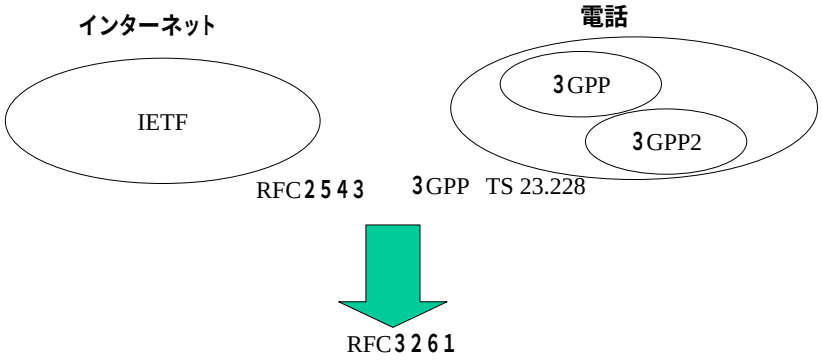
RFC(1/2)

NTTPC COMMUNICATIONS

当初
RFC2543（153枚）

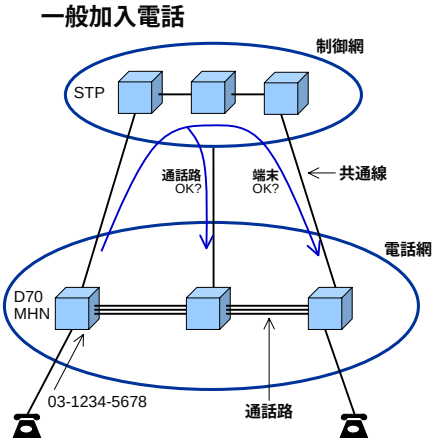
改定後
RFC3261(269枚)　SIPコア
RFC3262（14）応答メッセージの転送の信頼性強化
RFC3263（17）SIPサーバがDNSを使う場合の処理
RFC3264（25）SDPのオファーアンサーモデル
RFC3265（38）イベント通知処理

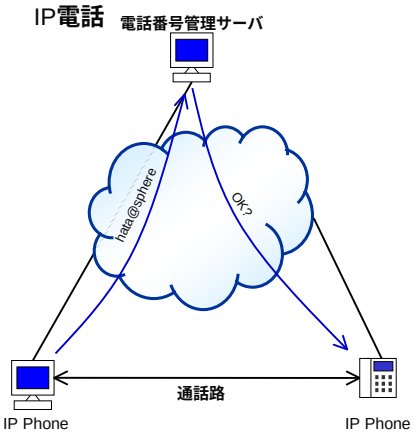
RFC(2/2)



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





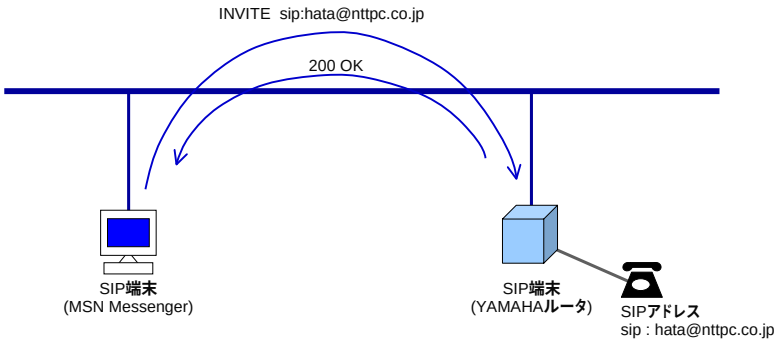
No.7共通線信号

H 323 : Biglobe、@nifty
SIP : eAccess
MEGACO/MGCP : ぷらら、BB Phone

SIPを使った簡単な通話（とりあえず試してみよう）

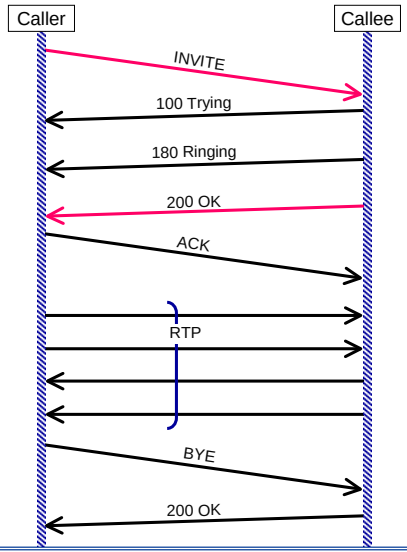
NTTPC COMMUNICATIONS

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

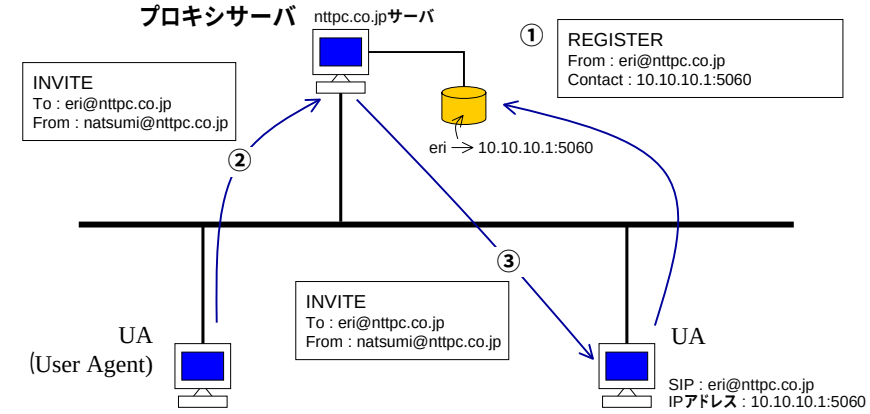


シーケンス概要

NTTPC COMMUNICATIONS

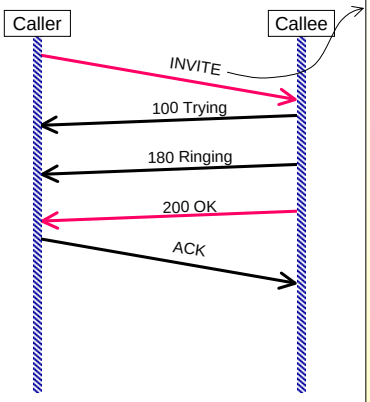


プロキシサーバとUA



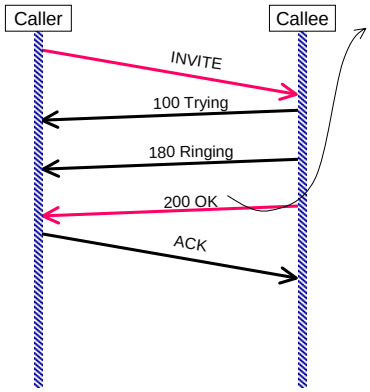
このスライドは重要です

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

リクエストURI

メソッド

```
INVITE      sip:test@202.229.156.250   SIP/2.0
Call-ID: 8526-62394-18945-00a0de0a2496@202.229.156.246
Contact: <sip:test@202.229.156.246:5060>
Content-Length: 163
Content-Type: application/SDP
CSeq: 24 INVITE
From: <sip:test@202.229.156.246>
To: <sip:test@202.229.156.250>
Via: SIP/2.0/UDP 202.229.156.246:5060

v=0
o=- 3232178697 3232178697 IN IP4 202.229.156.246
s=-
c=IN IP4 202.229.156.246
t=0 0
m=audio 5004 RTP/AVP 0
a=rtpmap:0 PCMU/8000
a=rtpmap:0 G.711U/8000
```

1行目はリクエスト行

2行目以降はSIPヘッダ行

空行

空行以降が本文
この例ではSDPメッセージが格納されている。

代表的なヘッダ

NTTPC COMMUNICATIONS

To ：あて先のSIP-URL

From ：発信元のSIP-URL

Call-ID ：ダイヤログを識別

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

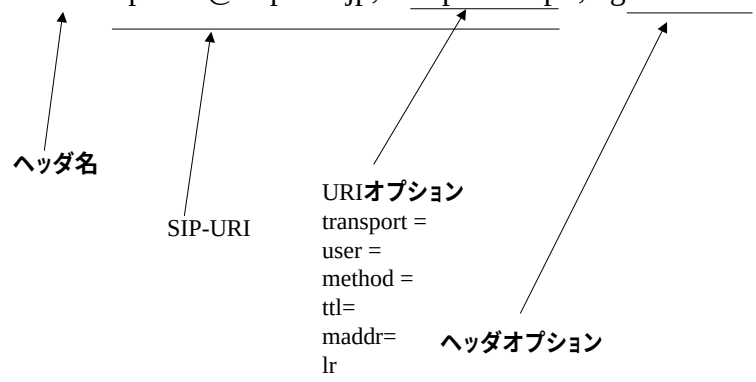
（例）

INVITE	application/SDP
NOTIFY	application/xpidftxml
	application/cpim-pidftxml
MESSAGE	text/plain

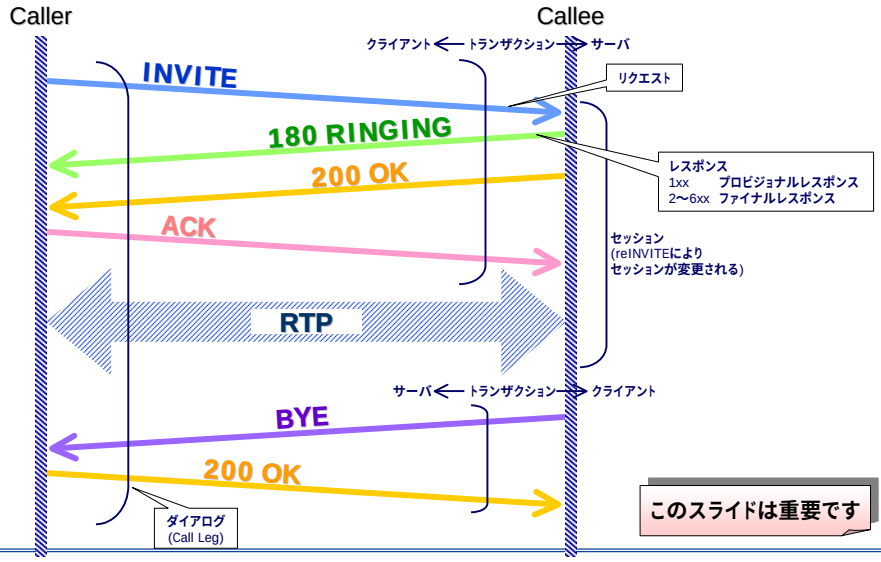
SIP-URIとオプション

NTTPC COMMUNICATIONS

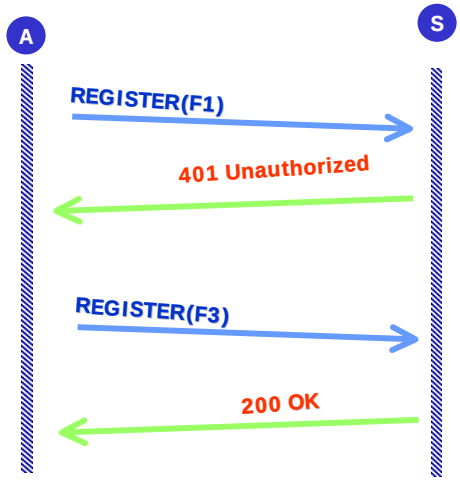
From: <sip:hata@nttpc.co.jp;transport=udp>;tag=24521



用語



REGISTERシーケンス



REGISTER(F1)

NTTPC COMMUNICATIONS

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)

NTTPC COMMUNICATIONS

```
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)

NTTPC COMMUNICATIONS

```
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=ja743ks76zlflH
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)

NTTPC COMMUNICATIONS

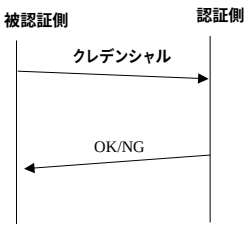
```
SIP/2.0 200 OK
 Via: SIP/2.0/TLS
client.biloxi.example.com:5061;branch=z9hG4bKnashd92
   ;received=192.0.2.201
From: Bob <sips:bob@biloxi.example.com>;tag=ja743ks76zlflH
To: Bob <sips:bob@biloxi.example.com>;tag=37GkEhwl6
Call-ID: 1j9FpLxk3uxtm8tn@biloxi.example.com
CSeq: 2 REGISTER
Contact: <sips:bob@client.biloxi.example.com>;expires=3600
Content-Length: 0
```

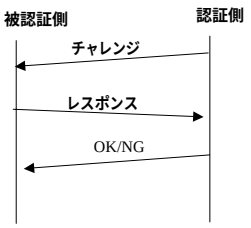
AAA 認証

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

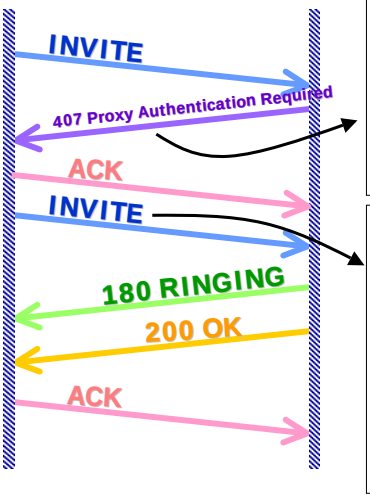
認証方式

- 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: <sip:0352065369@voip.sphere.ne.jp>;tag=2389310069
To: <sip: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 sip:0352106434@ocn.ne.jp;user=phone SIP/2.0
Call-ID: 53409-55514-55500-00a0de0be46e@10.224.228.1
Contact: <sip:0352065369@10.224.228.1:5060>
Content-Length: 135
Content-Type: application/sdp
CSeq: 83 INVITE
From: <sip: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: <sip: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

NTTPC COMMUNICATIONS

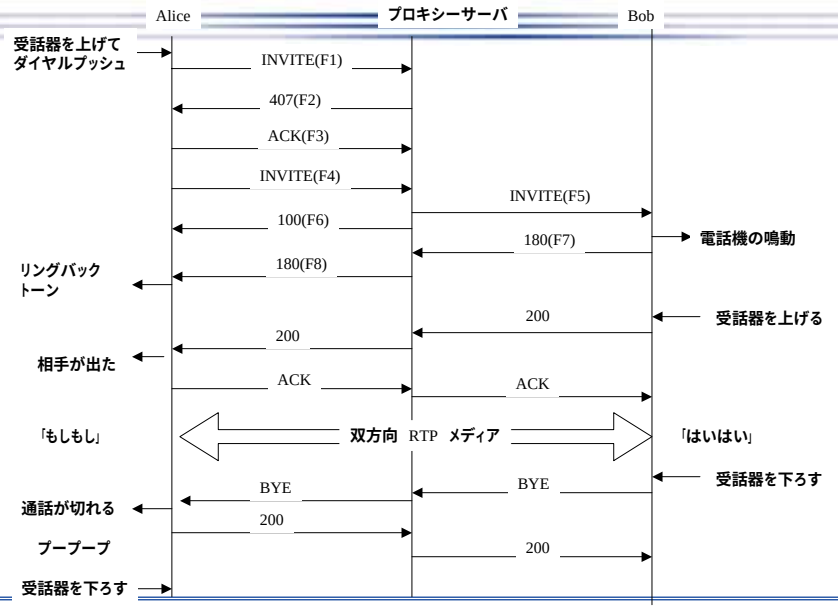
HA1=MD5(Username:Realm:Password)

HA2=MD5(Method:URI)

Response=MD5(HA:Nonce:HA2)

ワンコールシーケンス

NTTPC COMMUNICATIONS



INVITE(INV) F1

NTTPC COMMUNICATIONS

```
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=9fxced76sl
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

NTTPC COMMUNICATIONS

```
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=9fxced76sl
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

NTTPC COMMUNICATIONS

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

NTTPC COMMUNICATIONS

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=rtpmap:0 PCMU/8000

見どころ

CSeq
response

INVITE(180) F7

NTTPC COMMUNICATIONS

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=9fxced76sl
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

NTTPC COMMUNICATIONS

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=9fxced76sl
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

NTTPC COMMUNICATIONS

```
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=rtpmap:0 PCMU/8000
```

見どころ

SDP
Contactヘッダ

INVITE(ACK)

NTTPC COMMUNICATIONS

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)

NTTPC COMMUNICATIONS

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)

NTTPC COMMUNICATIONS

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

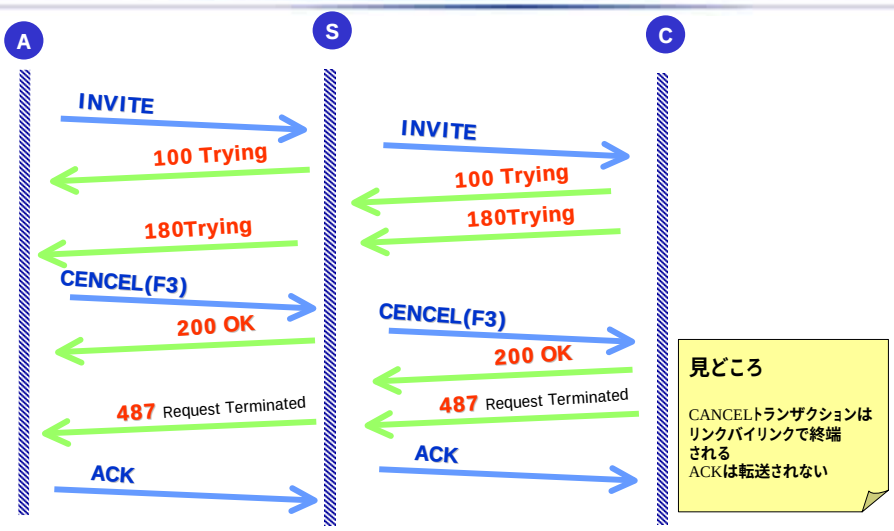
To: Alice <sip:alice@atlanta.example.com>;tag=9fxced76sl

Call-ID: 3848276298220188511@atlanta.example.com

CSeq: 1 BYE

Content-Length: 0

Cancel



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=9fxced76sl
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

CANCEL(F2)

NTTPC COMMUNICATIONS

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)

NTTPC COMMUNICATIONS

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

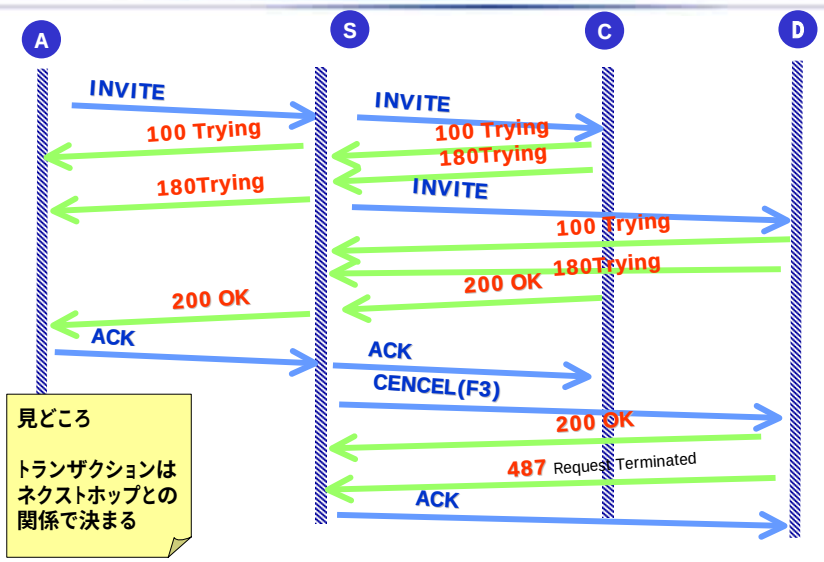
CANCEL(F4)

NTTPC COMMUNICATIONS

```
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=9fxced76sl
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)

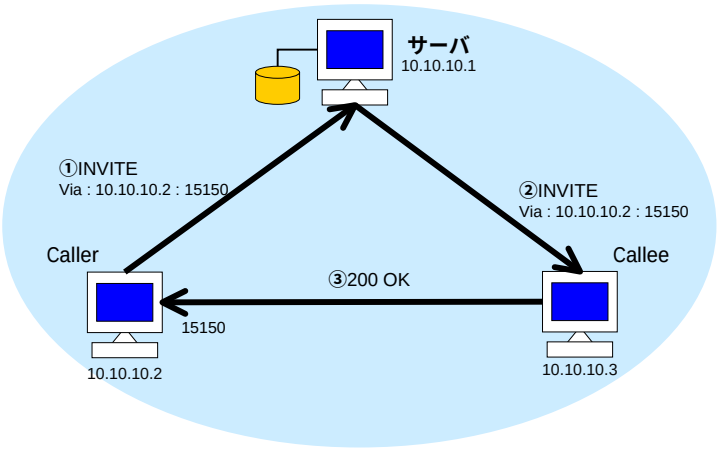
NTTPC COMMUNICATIONS



ルーティング

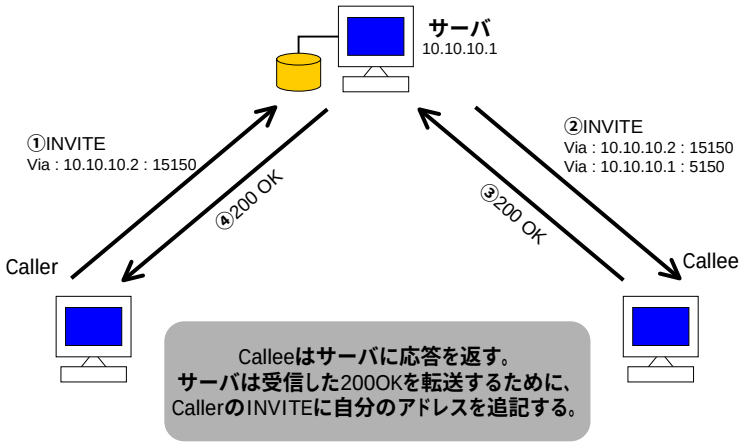
- ルーティングの必要性
  - どうしてトランスポートプロトコルにルーティングが必要なのか？
- 中継されるUDP
  - UDP上のトランスポートレイヤ（SNMP、RADIUSなどはルーティング機能はあるのか？）

ルーティングの必要性1



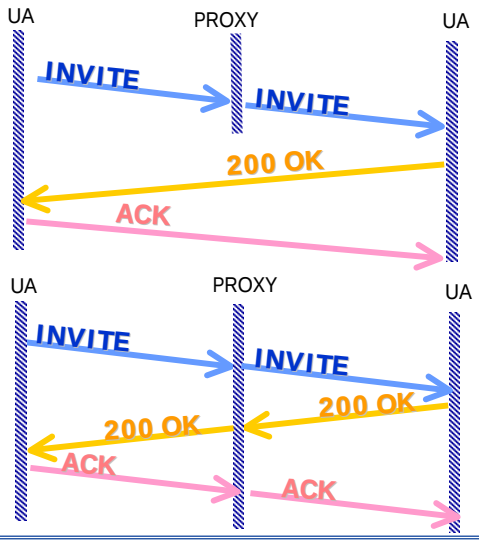
サーバは状態管理をしない。Calleeは直接Callerに応答を返す

ルーティングの必要性2



- 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



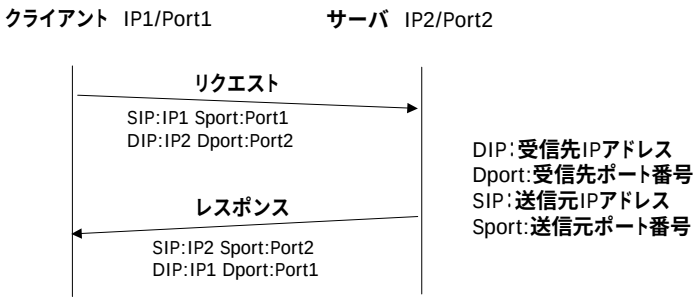
INVITEだけ転送するので
あとは勝手にやって頂戴

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

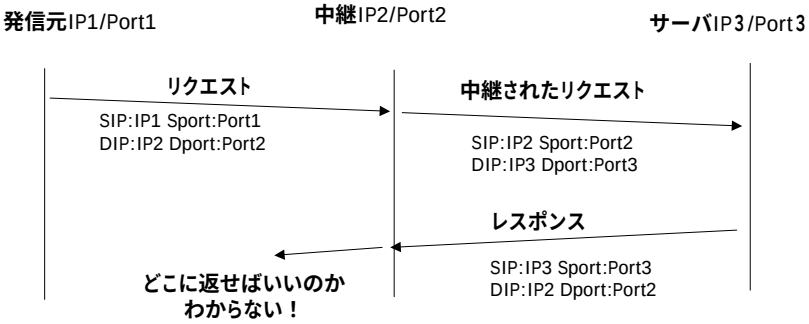
NTTPC COMMUNICATIONS



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

NTTPC COMMUNICATIONS



中継サーバIP2はプロキシーサーバであり、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
ために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)

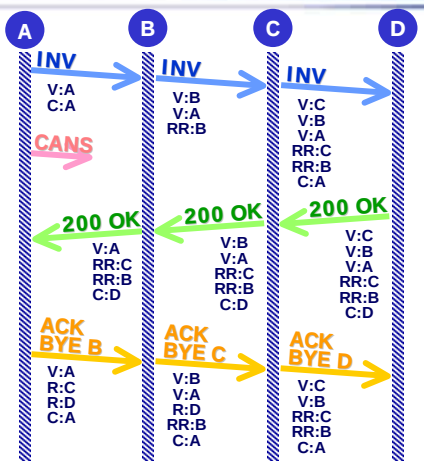
NTTPC COMMUNICATIONS

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

STRICT ROUTING

パターン1 (Callerから切断)

NTTPC COMMUNICATIONS



200OKを受け取る前は
ルーティング情報がないため
プロキシーサーバへ出す

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 ④
RR : NODE2 ③
RR : NODE3 ②
RR : NODE4 ①
C : NODE5 ⑤

ACK/BYE

① REQURST-URL : NODE4
② R : NODE3
③ R : NODE2
④ R : NODE1
⑤ R : NODE5

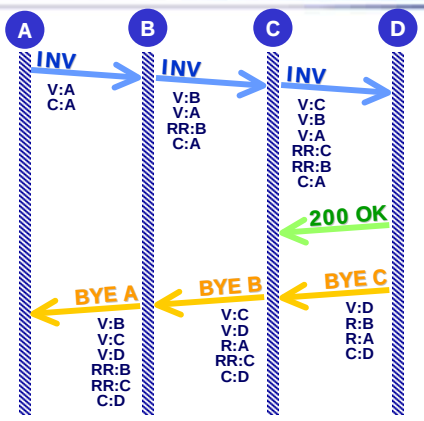
ルーティング情報の作り方

① Record-Routeの最下位のノードをリクエストURLにする。
①' 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
② 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。
③ 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)

NTTPC COMMUNICATIONS



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	①
RR : NODE2	②
RR : NODE3	③
RR : NODE4	④
C : NODE5	⑤

BYEのルーティング情報

① REQURST-URL : NODE1
② R : NODE2
③ R : NODE3
④ R : NODE4
⑤ R : NODE5

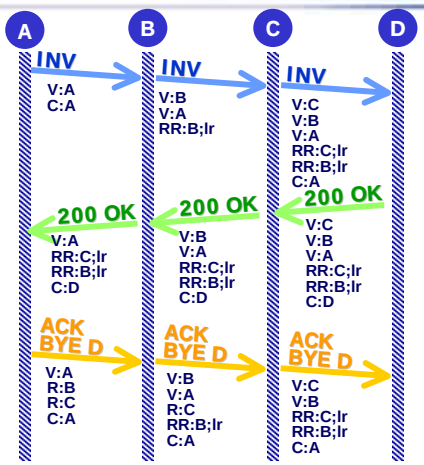
ルーティング情報の作り方

① Record-Routeの最上位のノードをリクエストURLにする。
①' Record-Routeがないときは Contactをリクエスト URLとする。→処理終わり
② 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。
③ 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)

NTTPC COMMUNICATIONS



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	⑤
RR : NODE2	④
RR : NODE3	③
RR : NODE4	②
C : NODE0	①

ACK/BYE

① REQURST-URL : NODE0
② R : NODE4
③ R : NODE3
④ R : NODE2
⑤ R : NODE1

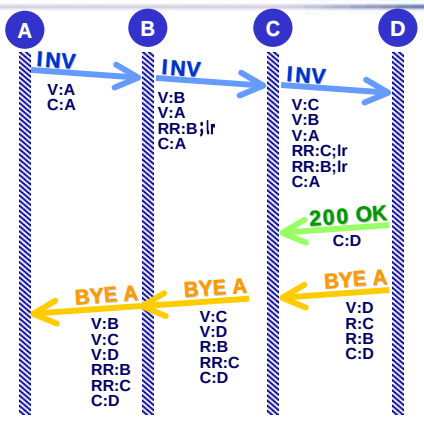
ルーティング情報の作り方

① Contactノードをリクエスト URLにする。
② 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)

NTTPC COMMUNICATIONS



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 ②
RR : NODE2 ③
RR : NODE3 ④
RR : NODE4 ⑤
C : NODE5 ①

BYEのルーティング情報

① REQURST-URL : NODE5
② R : NODE1
③ R : NODE2
④ R : NODE3
⑤ R : NODE4

ルーティング情報の作り方

① Contactノードをリクエスト URLにする。
② 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。

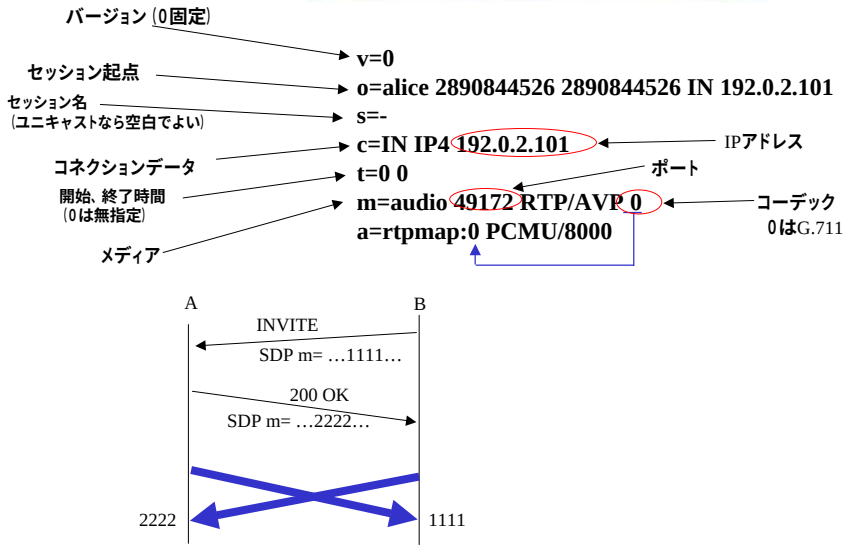
パート2

NTTPC COMMUNICATIONS

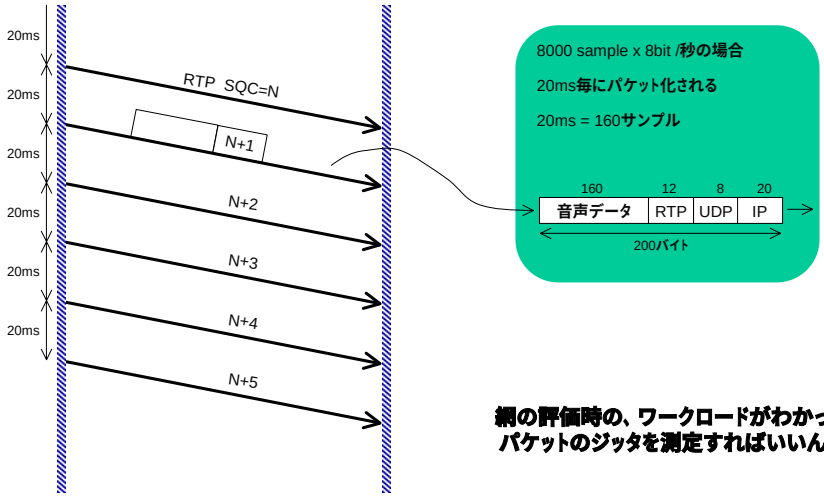
パート2

音声信号処理とパケット化

SDP



音声伝送

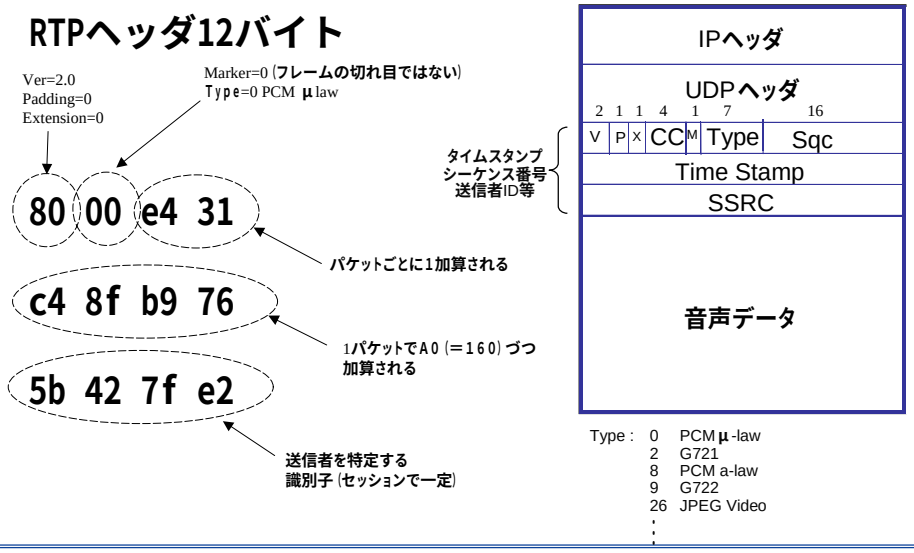


網の評価時の、ワークロードがわかった
パケットのジッタを測定すればいいんだ

RTP

NTTPC COMMUNICATIONS

RTPヘッダ12バイト



Type : 0 PCMμ-law
2 G721
8 PCM a-law
9 G722
26 JPEG Video
⋮

RTPパケットダンプ

NTTPC COMMUNICATIONS

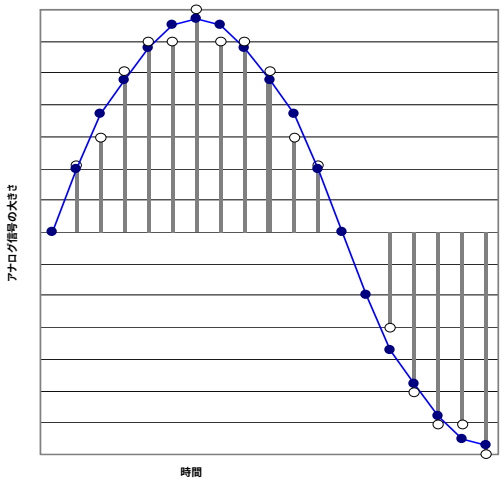
RTPヘッダ

```
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   dhlpx}..........
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   {zxyvxvwwuxwywwy
0070  79 79 78 7a 7a 7d 7a 7c 7b 7c 7c 7e 7c 7e 7d 7c   yyxzz}z|{||~|~}|
0080  7e 7b 7c 7a 7a 79 79 78 75 74 6f 6e 6c 68 64 5f   ~{|zzyyxutonlhd_
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
```

音声信号160バイト

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



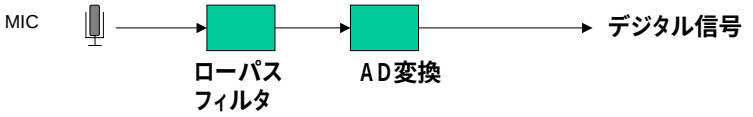
横軸のきめの細かさ（サンプリングレート）
　1秒間あたり何データサンプルするか
　電話　8000サンプル毎秒
　CD　44100サンプル毎秒
　サンプリング周波数の半分の周波数成分
　まで再現できる（ナイキスト、シャノン、染谷）

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

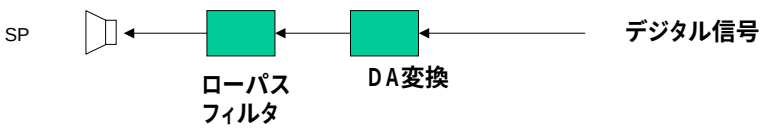
1秒間に何ビット伝送しなければならないか
電話
　8Kサンプル毎秒×8ビット毎サンプル
　＝64Kビット毎秒
CD
　44．1Kサンプル×16ビット×2チャンネル
　＝約1．5Mbps（1倍速CD）

AD変換

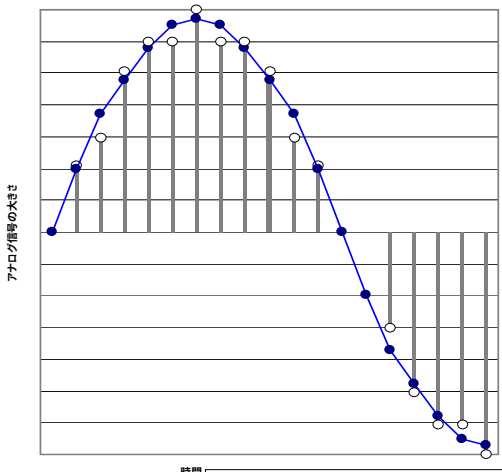
サンプリング周波数の1／2の周波数以上の成分が含まれると
雑音になる（エイリアシング）ので、事前に高域成分をLPFで除去する



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



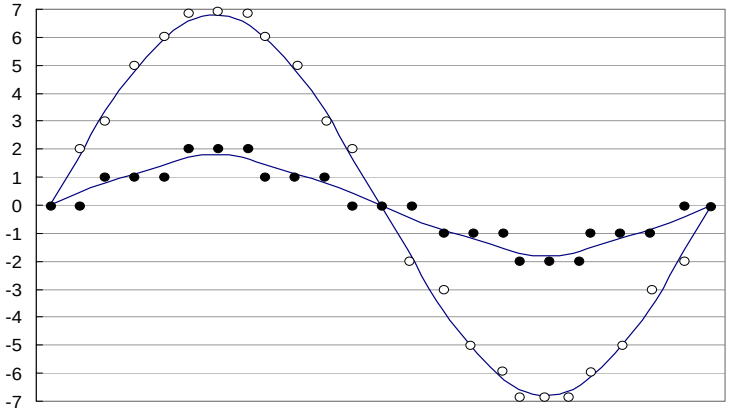
符号の種類



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	0000 1000	1111 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

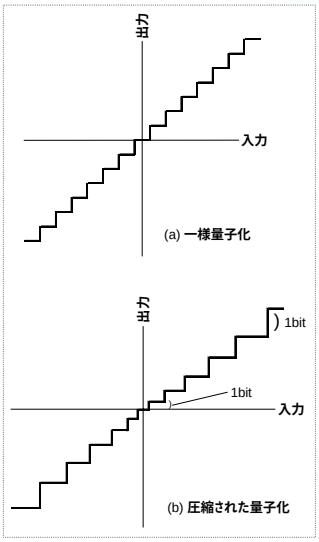
WAVファイル オフセット2進
PCM伝送　　折り返し2進(反転)

量子化雑音

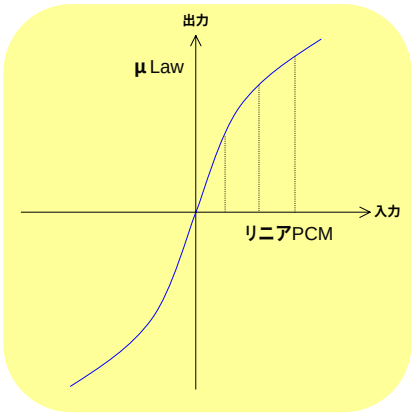


量子化雑音により弱信号ほど
S/N比が悪くなる

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	Ⅷ	10000000	8031
7647～7903	〃	〃	10000001	7775
7391～7647	〃	〃	10000010	7519
7135～7391	〃	〃	10000011	7263
・ ・	・	・	・	・
・ ・	・	・	・	・
・ ・	・	・	・	・
35～39	4	Ⅱ	11101110	37
31～35	〃	〃	11101111	33
29～31	2	Ⅰ	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

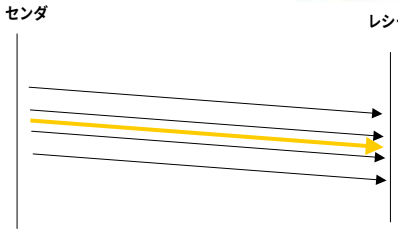
実装

16bitのリニアPCMでサンプリング
↓
8bi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

NTTPC COMMUNICATIONS



RTPで使っているポート（ソース、ディスティネーション）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
・センダーレポート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

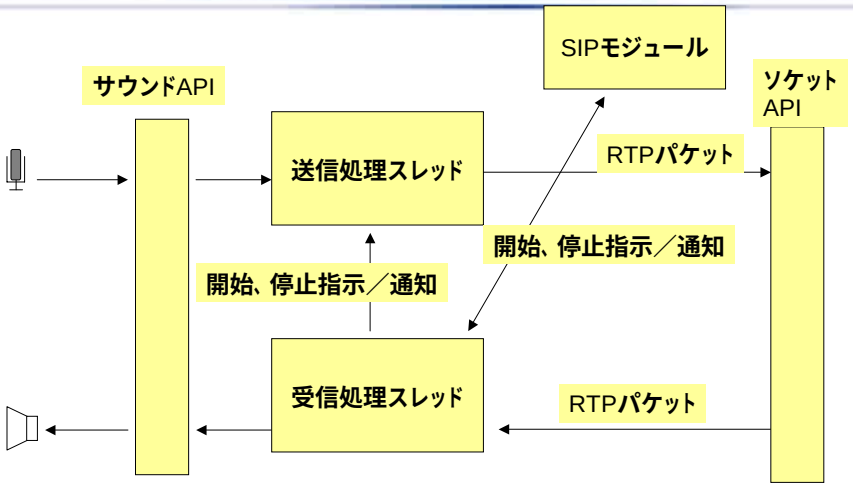
IP電話では、定期的に飛び交う（5秒に一度程度）ＲＴＣＰには、
ＳＲ／ＲＲ／ＳＤＥＳが含まれている。

ＲＴＣＰ

NTTPC COMMUNICATIONS

- 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。

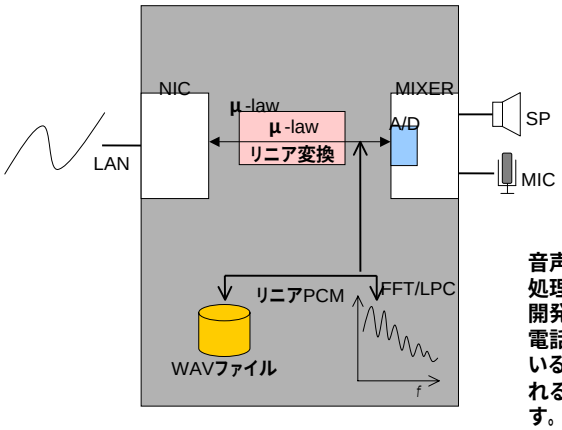
ソフトフォン実装



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
  - バッファに音声が書き込まれると、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ィジタル音声信号と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

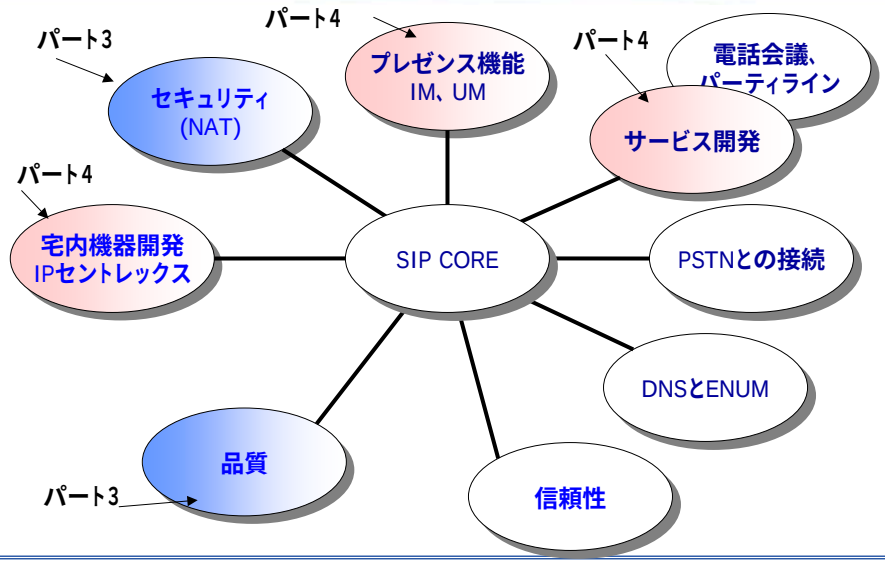
NTTPC COMMUNICATIONS

パート3

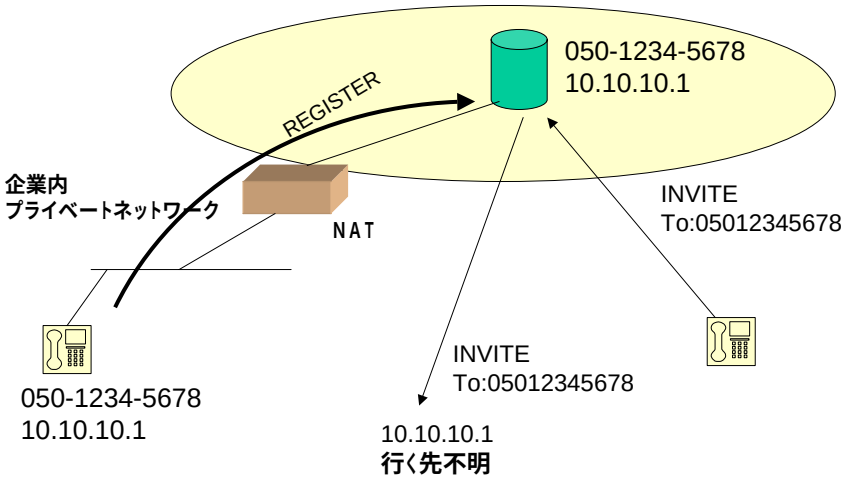
SIPとVoIPをとりまく諸問題

諸問題一覧

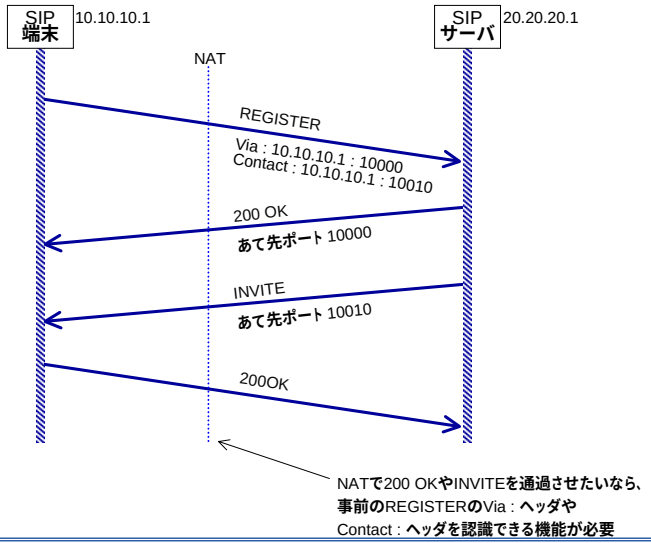
NTTPC COMMUNICATIONS



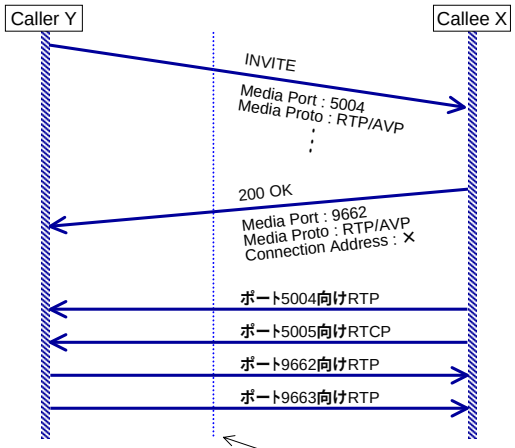
NAT問題



NAT1



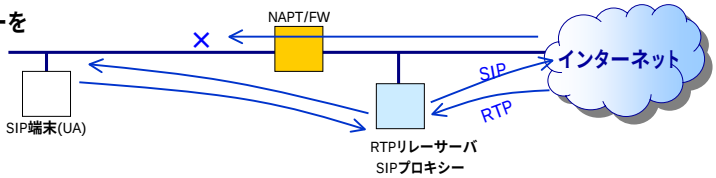
NAT2



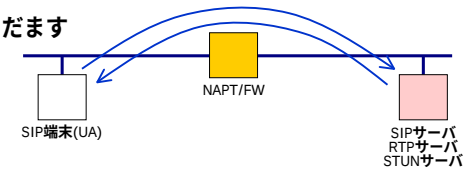
NATでRTPを通過させたいなら
事前に通過したINVITEと200 OKのSDPを
認識し、相手IPアドレス(200 OKの発元ではないかも?)
と自サイト側ポート番号を知っておく必要がある。

NAT3

◆プロキシーを使う方法

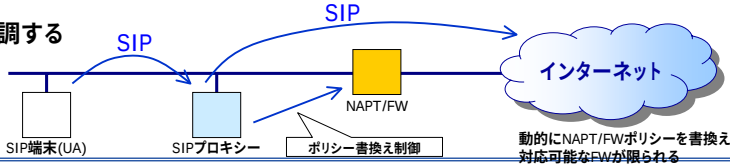


◆NAPTをだます方法



UAは自分の使う予定のポートを
使って(STUNサーバが教えてくれる)
定期的にSIPサーバとREGISTERハートビートする
セキュリティ的に脆弱

◆FWと協調する方法

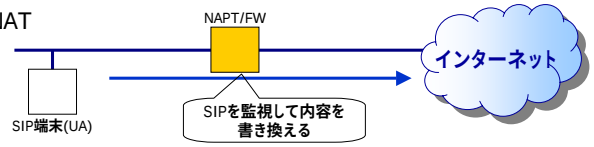


動的にNAPT/FWポリシーを書換える
対応可能なFWが限られる

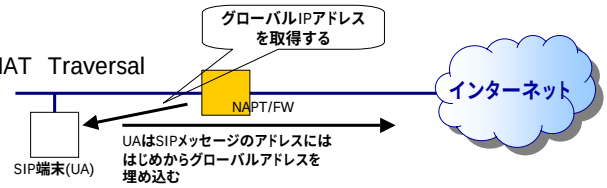
NAT4

NTT PC COMMUNICATIONS

◆SIP対応NAT

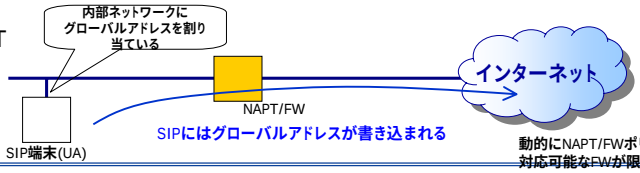


◆UPnP NAT Traversal



WARP STARTが
NAT Traversalを
対応した

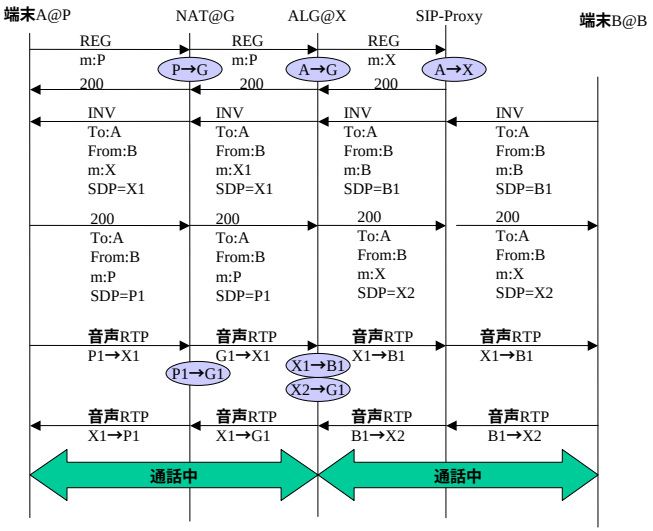
◆GapNAT



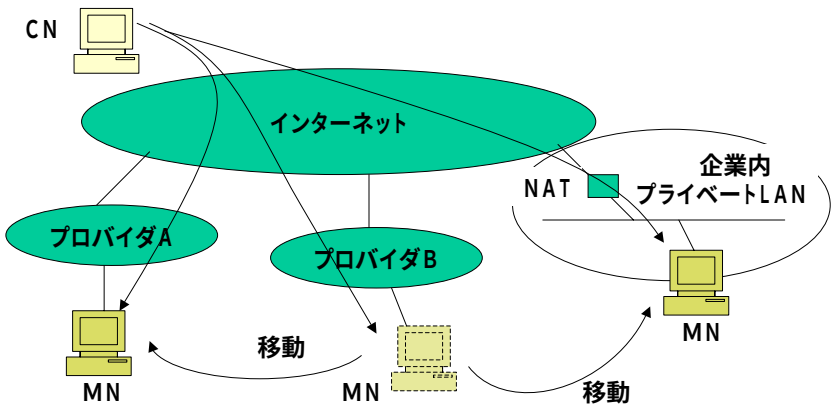
動的にNAPT/FWポリシーを書換える
対応可能なFWが限られる

ALG SBCによるNAT越え

NTT PC COMMUNICATIONS



Mobile IPによるNAT越え

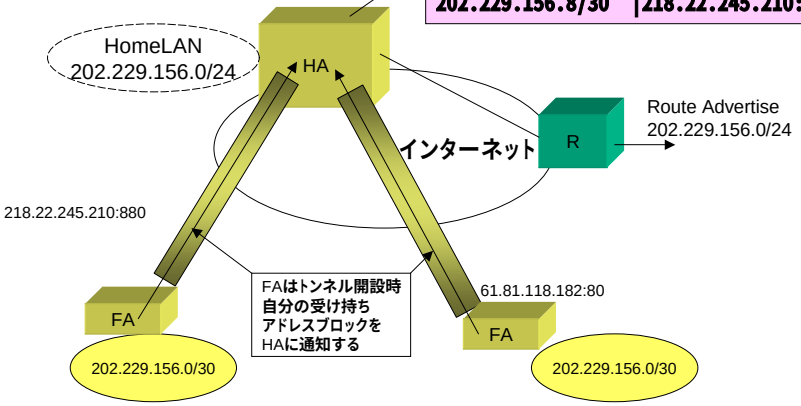


トンネリングプロトコル

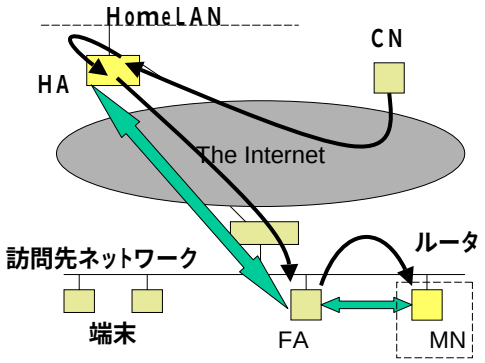
トンネル端点とCoAから
フォワーディングキャッシュを生成
グローバルから受け取ったパケットは
フォワーディングキャッシュに登録されていれば
該当トンネルに押し込められる

Forwarding Cache

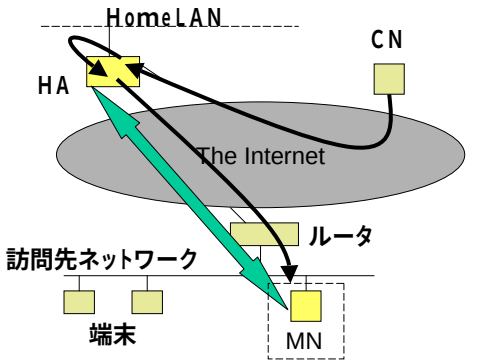
Destination	Care of Address	state
202.229.156.0/30	61.81.118.182:80	alive
202.229.156.8/30	218.22.245.210:880	alive



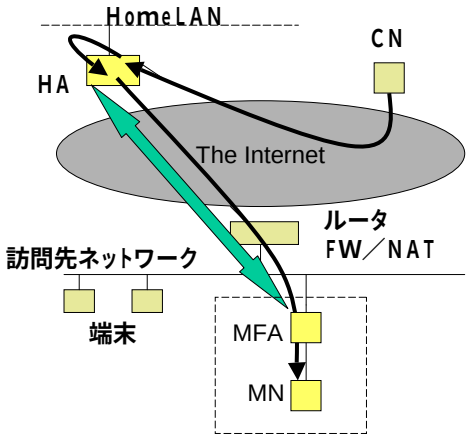
システムフレームワーク　FAモード



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 ®）



FAはMNとともに移動

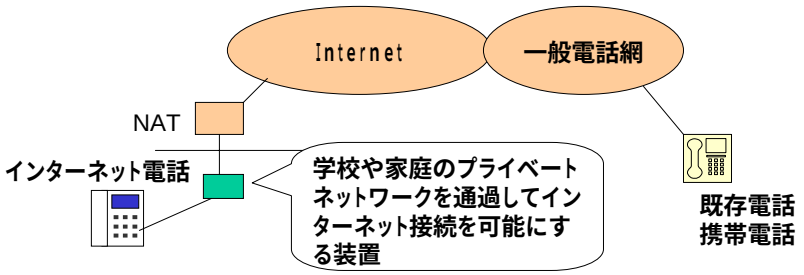
訪問先ネットには
特別な仕掛けを必要としない

MNにも特別な仕掛けを
必要としない

UDPでくるんでNATを
乗り越える

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

NTTPC COMMUNICATIONS

◆品質
- ■通話品質を保証するために、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

NTTPC COMMUNICATIONS

◆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、ジッタをともなう)

◆遅延
- ■音声おくれ

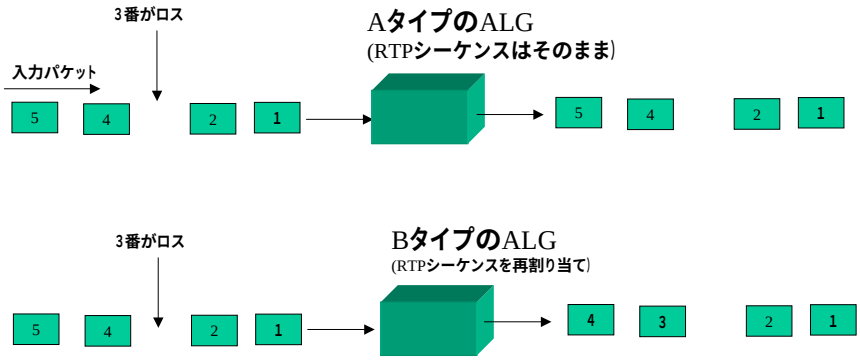
◆ジッタ(含パケット逆転)
- ■音声とぎれ
- ■パケットロスに似ている

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。

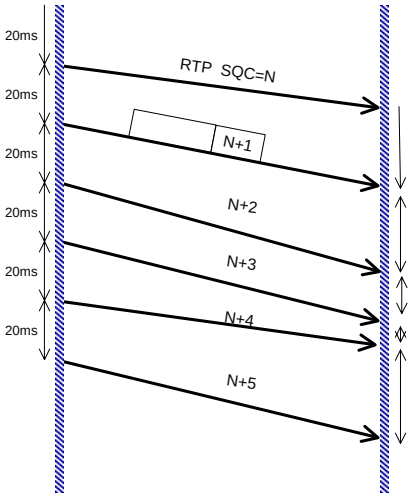
ALG越えのRTPシーケンス番号

Aタイプ、Bタイプのどちらが望ましいのだろうか？



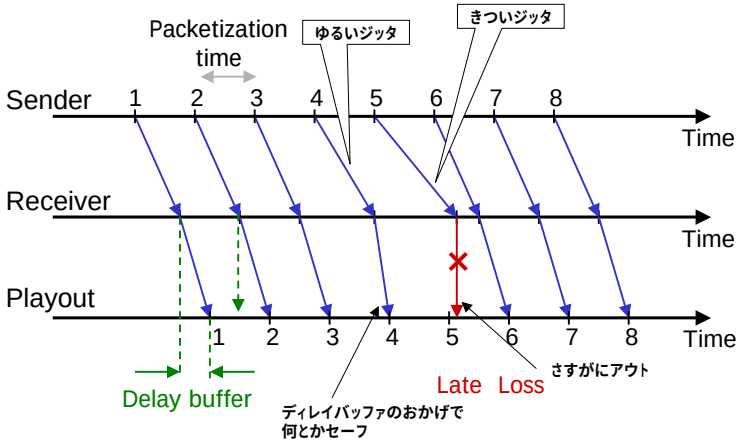
Bタイプの ALGではパケットロスの保障が効かず、音質の劣化になる
わかりやすくするためにロス数1パケットとしているが、実際には
バースト的に大量ロスが起こる。

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
不定期になる

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

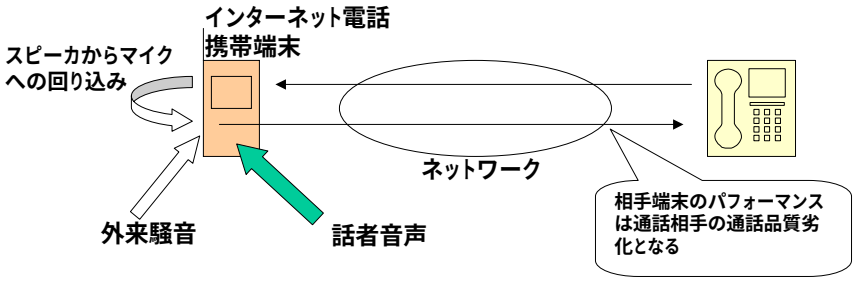
NTTPC COMMUNICATIONS

- 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

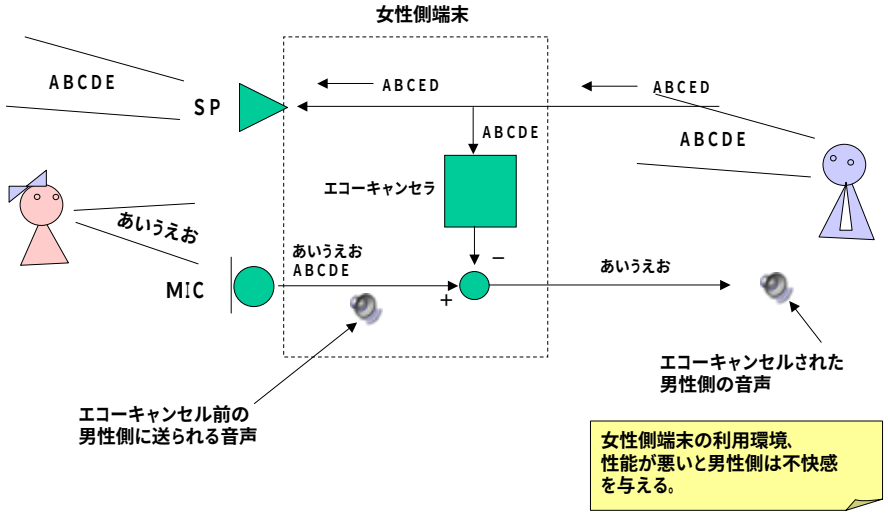
音響問題

NTTPC COMMUNICATIONS

- 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



エコーキャンセラ



サプレッサ効果

処理前

- 高速走行中携帯会話
- オーディオ音

処理後

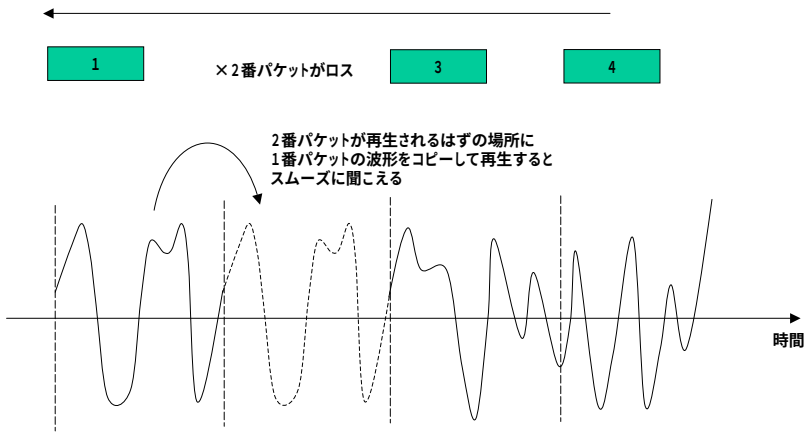
- 高速走行中携帯会話
- オーディオ音

協力：旭化成株式会社様

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)



その他セキュリティの問題

NTTPC COMMUNICATIONS

- 口頭説明

パート4

NTTPC COMMUNICATIONS

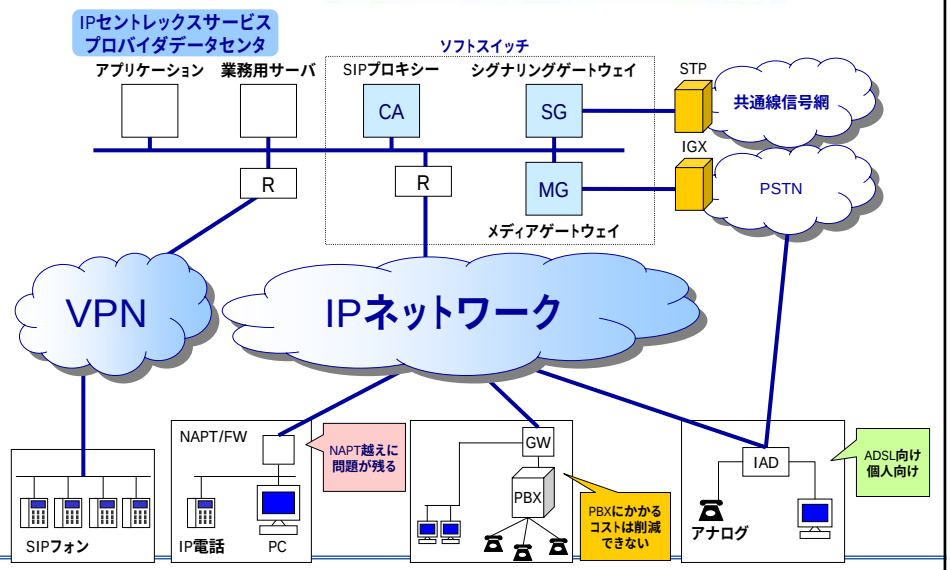
パート4

サービス化のためのシステム

サービス用件

- 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

NTTPC COMMUNICATIONS

IPセントレックス

ソフトスイッチ

CA ・・・・・ SIP Proxyサーバ
SG ・・・・・ SS7シグナリングGW
MG ・・・・・ IP/PSTN GW

業務用サーバ　　課金　監視
アプリケーションサーバ　　メディアサーバ　ボイスメール

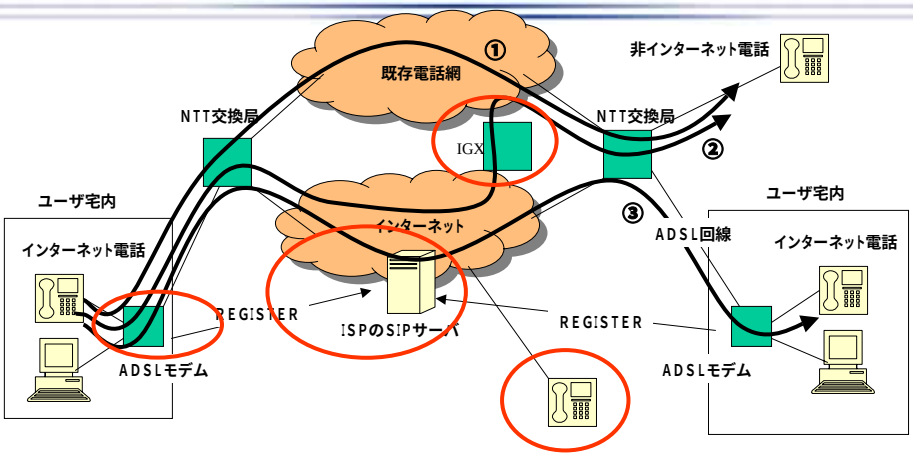
端末(UA)

SIPフォン	ソフトフォン、ハードフォン
GW(TA)	アナログ　ディジタル変換
IAD	TA + DSLモデム(+PPPoEクライアント)

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

NTTPC COMMUNICATIONS



①: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。
②: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
③: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。

IP PBXの機能

NTTPC COMMUNICATIONS

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
- 識別リンギング
- 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
- 内線キャンプオン
- アッドオン会議（三者）
- コールピックアップ後のアッドオン会議
- 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

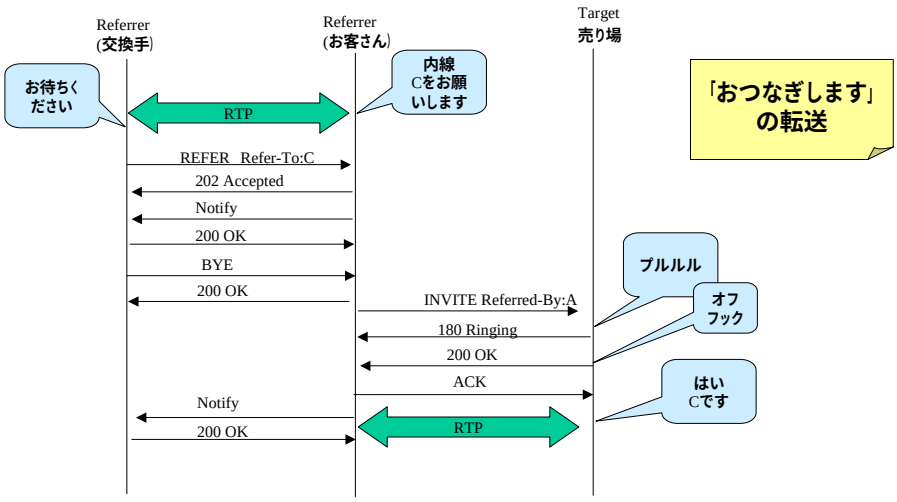
REFERの使い方

NTTPC COMMUNICATIONS

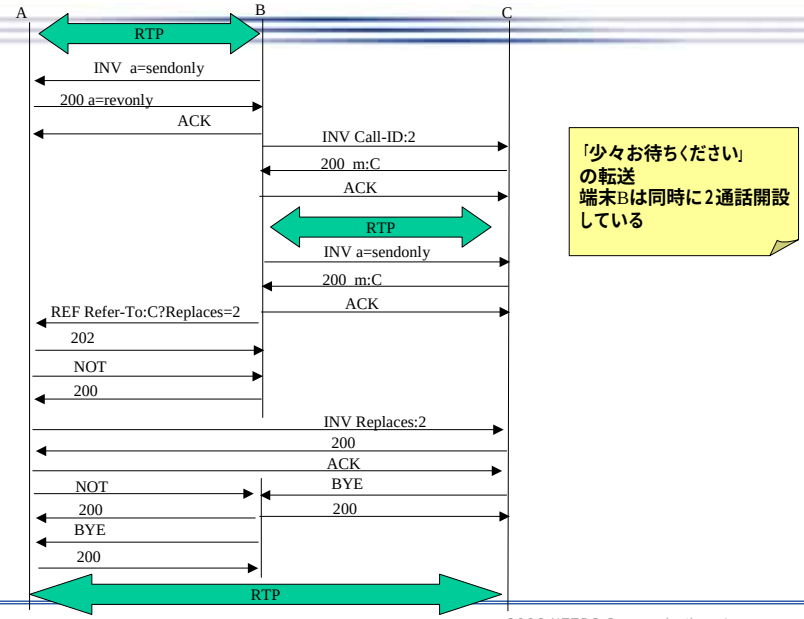
- REFERの目的
  - REFER受信先があらたなリクエスト（デフォルトはINVITE)を発行するように促す。
  - 暗黙的にSUBSCRIBEを発行した効果ももつ
- 主な利用
  - コール転送
    - 不応答転送
    - 応答転送

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。

不応答転送(Unattended Transfer)



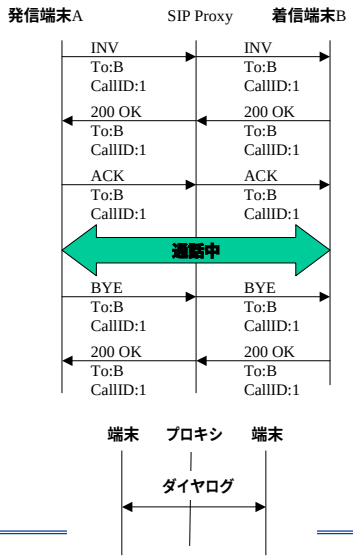
仲介転送(Attended Transfer)



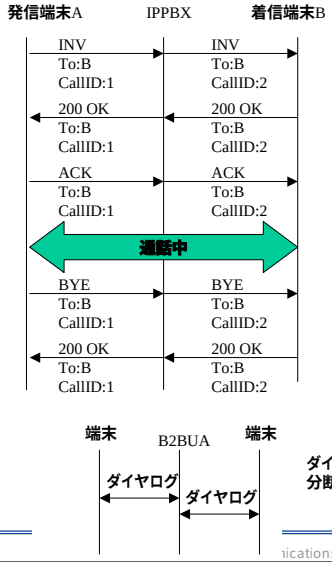
IPPBXはB2BUA

NTTPC COMMUNICATIONS

SIP Proxyサーバの
ワンコールシーケンス



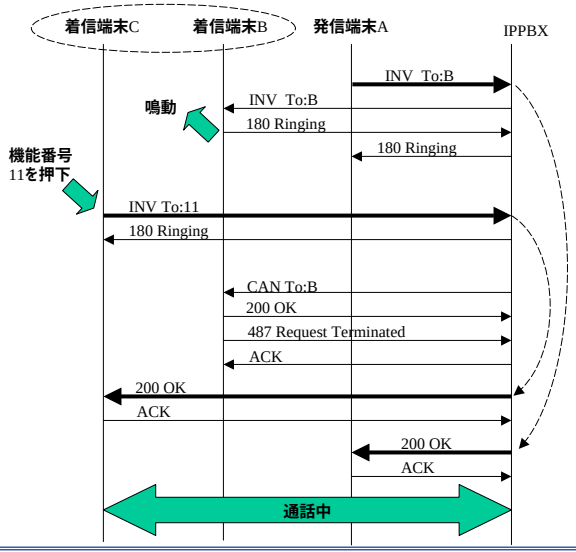
IPPBXの
ワンコールシーケンス



ダイヤログが
分断されている

コールピックアップ

NTTPC COMMUNICATIONS

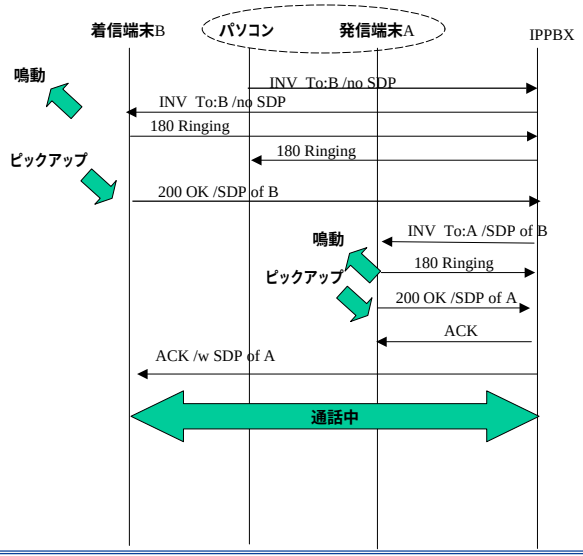


端末B,Cは同一鳴動グループ
にあって、特殊場号11を
コールピックアップに割り当てている

通話しているAもCも、発信端末

ダイヤログが
分断されているからできる技

3PCC

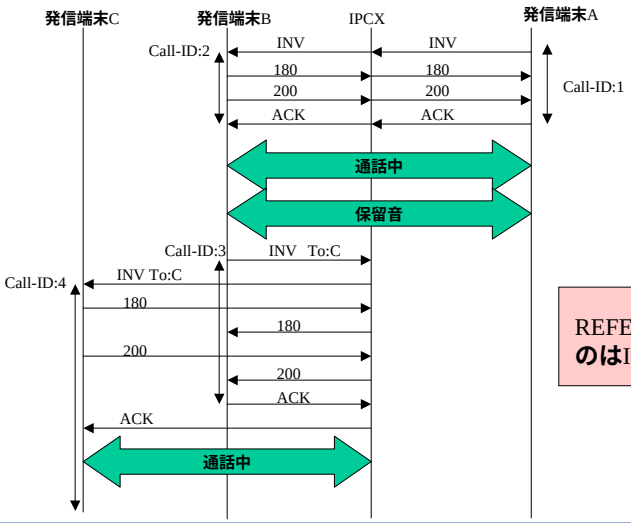


パソコンの電話帳をクリックすると相手の電話が鳴動する。
相手がでると自分の電話が鳴動する

通話しているAもBも、着信端末

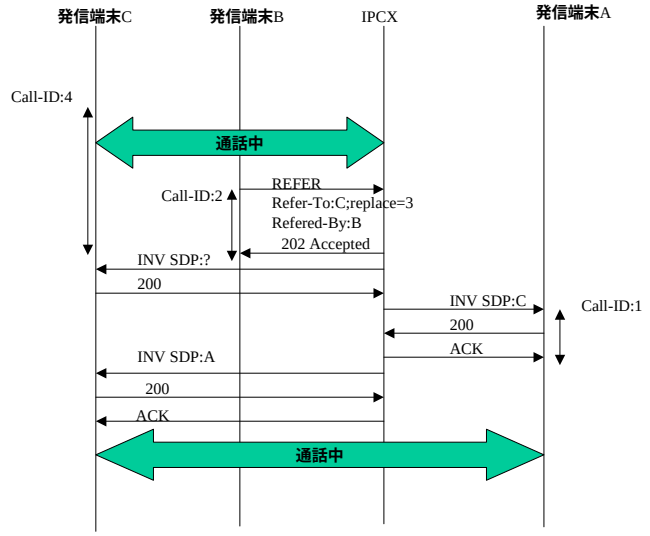
ダイヤログが
分断されているからできる技

IPPBXの呼転送（仲介転送）

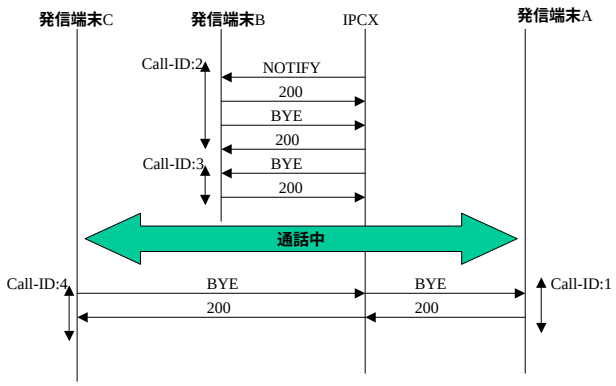


REFERを受信するのはIP-PBXのみ

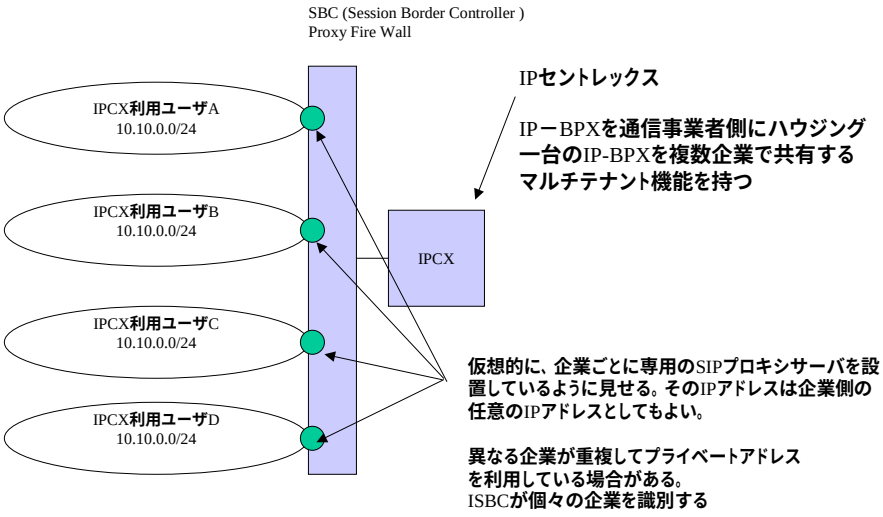
呼転送（仲介転送）つづき



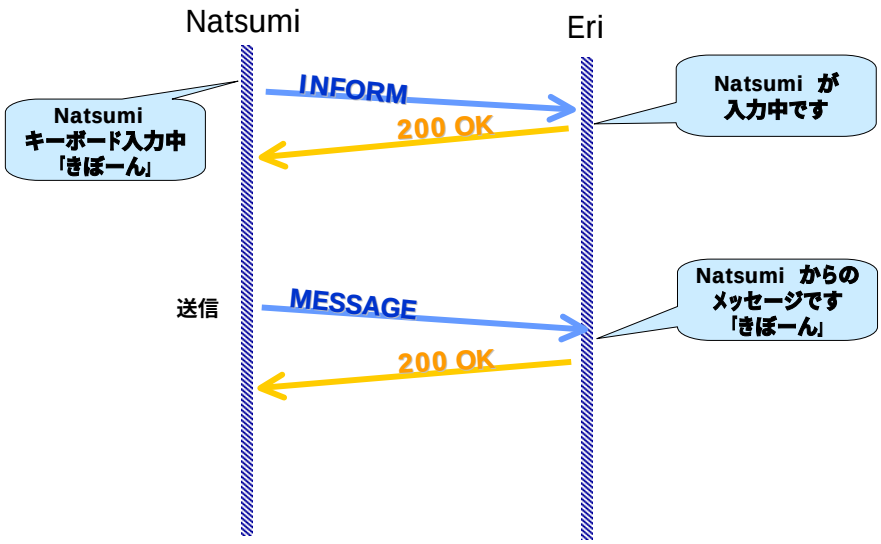
呼転送（仲介転送）つづき2

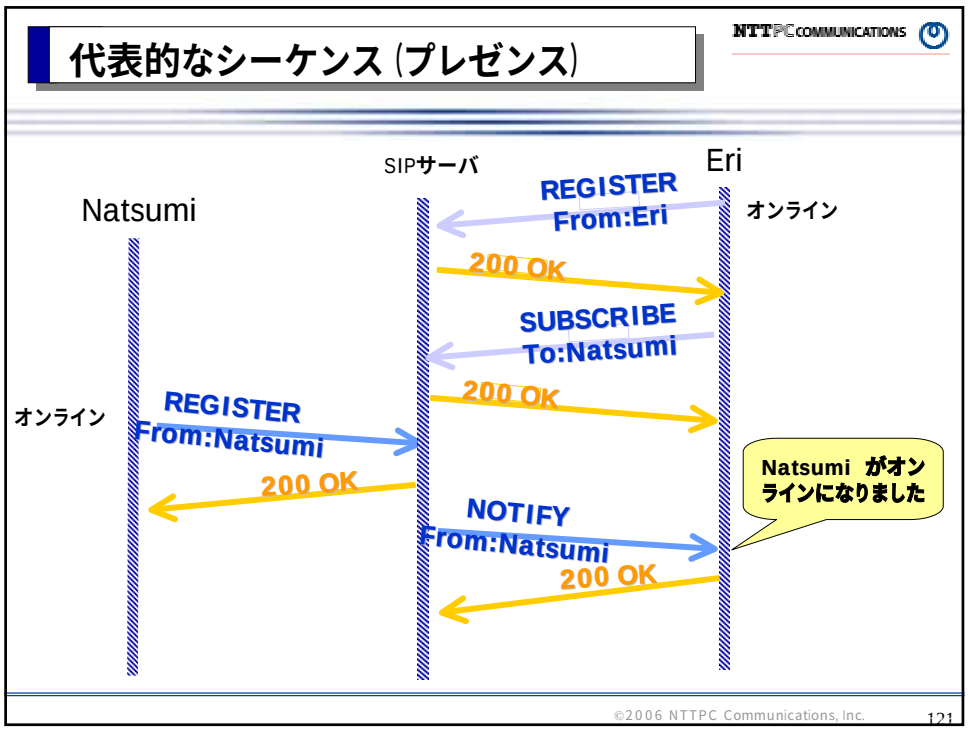


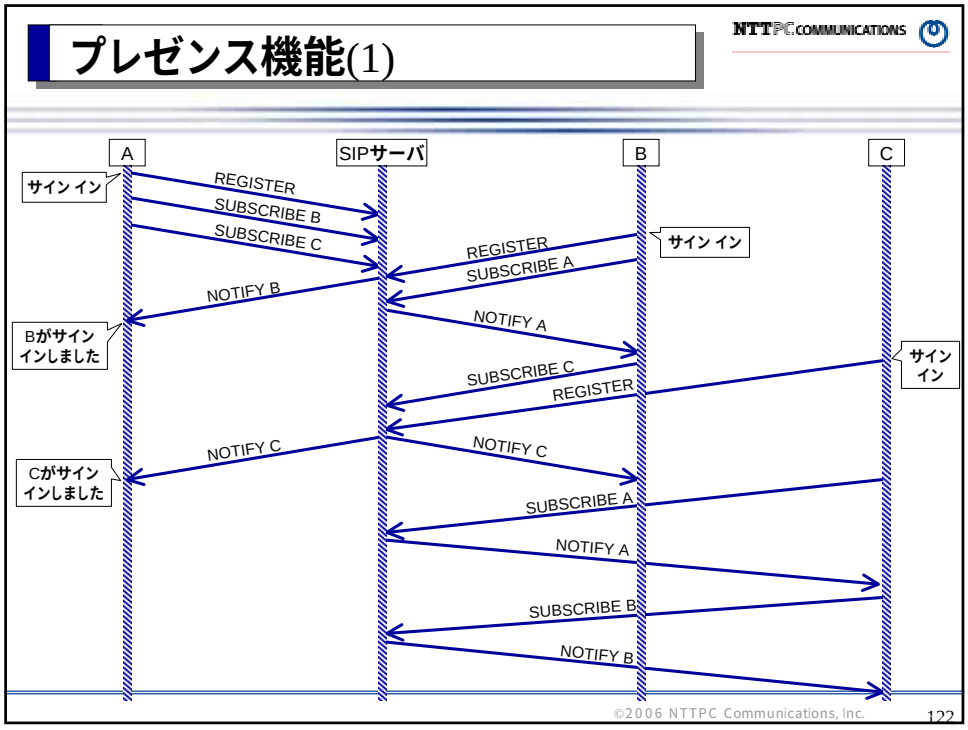
Session Border Controller



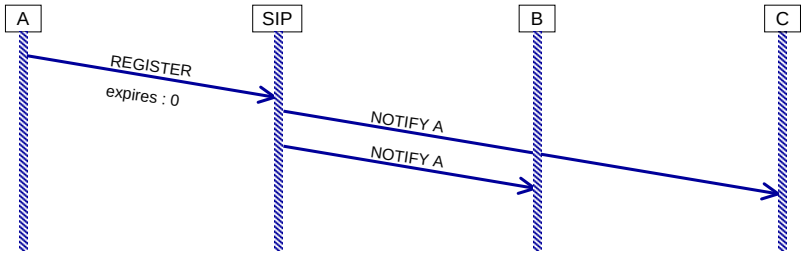
その他のSIPメッセージ　代表的なシーケンス




代表的なシーケンス（プレゼンス）
NTTPC COMMUNICATIONS
SIPサーバ
Eri
Natsumi
REGISTER
From:Eri
オンライン
200 OK
SUBSCRIBE
To:Natsumi
200 OK
オンライン
REGISTER
From:Natsumi
200 OK
Natsumi がオンラインになりました
NOTIFY
From:Natsumi
200 OK
©2006 NTTPC Communications, Inc.
121


プレゼンス機能(1)
NTTPC COMMUNICATIONS
A
SIPサーバ
B
C
サイン イン
REGISTER
SUBSCRIBE B
SUBSCRIBE C
REGISTER
SUBSCRIBE A
サイン イン
NOTIFY B
NOTIFY A
Bがサイン インしました
SUBSCRIBE C
REGISTER
サイン イン
NOTIFY C
NOTIFY C
Cがサイン インしました
SUBSCRIBE A
NOTIFY A
SUBSCRIBE B
NOTIFY B
©2006 NTTPC Communications, Inc.
122

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していればよかった。
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/xpidftxml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-pidftxmlに変わっている

●draft-ietf-impp-cpim-pidf-05.txt
- ◆cpim-pidftxmlの使い方

結局MSN Messengeはcipm-pidftxmlにどうにも反応せず

サイン インの通知方法

NTTPC COMMUNICATIONS

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

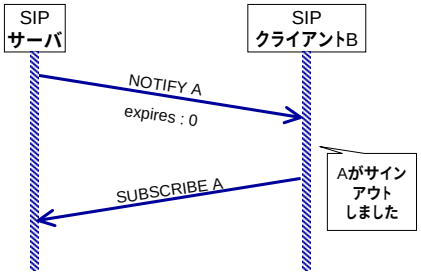
```
NOTIFY sip:xiaotaow@128.59.19.27:5060 SIP/2.0
Via: SIP/2.0/UDP 128.59.19.251:13170
From: sip:william@conductor.cs.columbia.edu;tag=d2dd3c15-b762-486b-8911-25ed3a3bd975
To: sip:xiaotaow@cs.columbia.edu
Call-ID: 330938581@128.59.19.27
CSeq: 1 NOTIFY
Contact: <sip:128.59.19.251:13170>
User-Agent: Windows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

<?xml version="1.0"?>
<!DOCTYPE presence
PUBLIC "-//IETF//DTD RFCxxxx 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

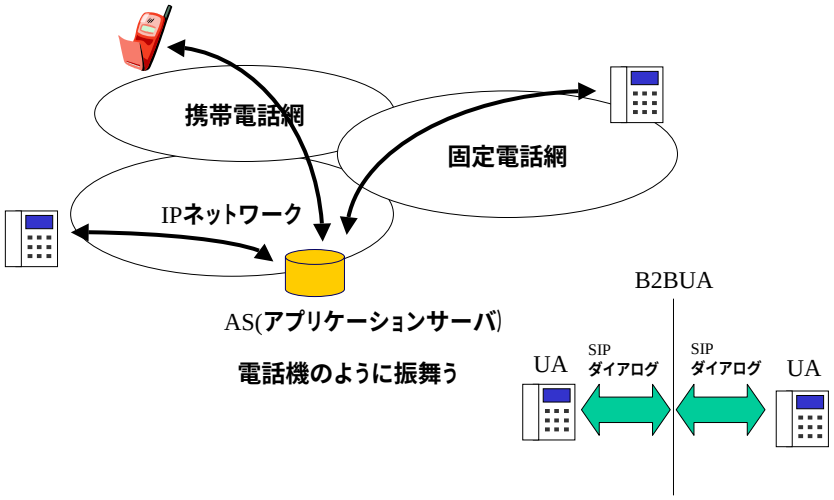
NTTPC COMMUNICATIONS

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

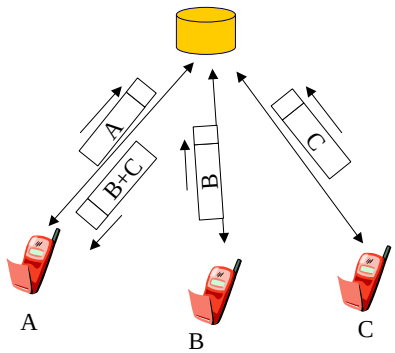


NOTIFYのメッセージボディではなく
メッセージヘッダのexpiresを0にセットした
ものを送ります。

電話機以外の端末（AS）



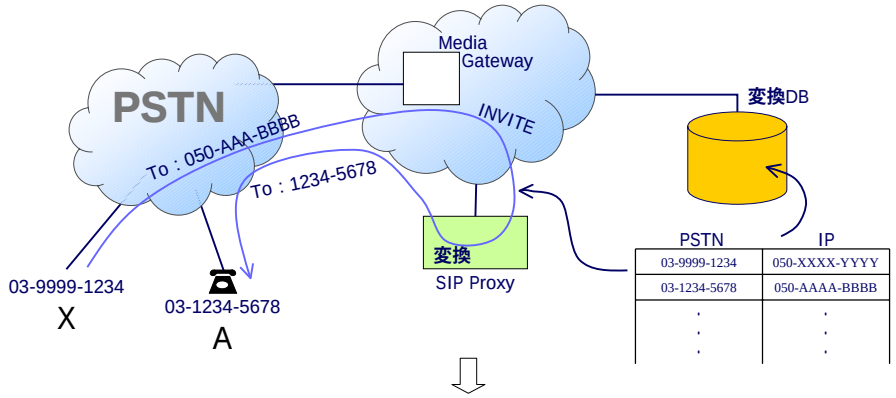
（例）電話会議システム(Party Line)



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。

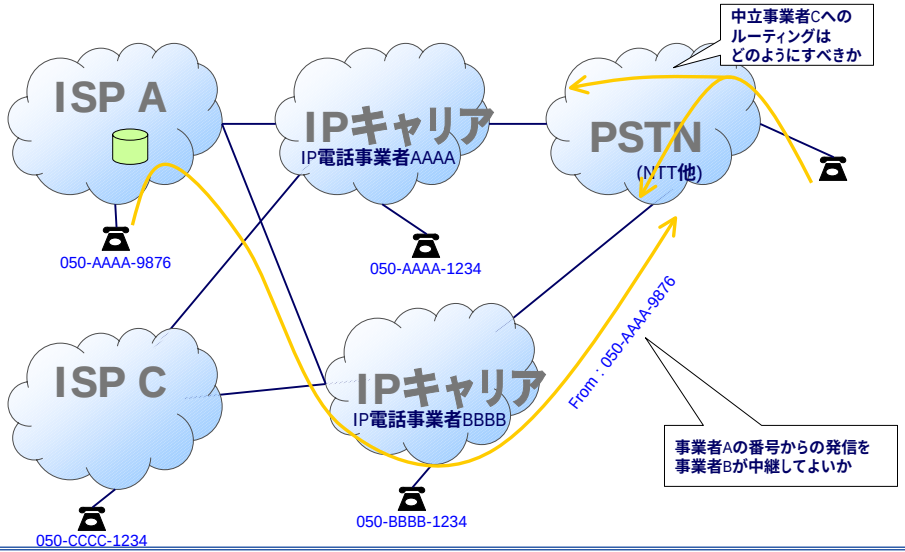
自分の声が返ってこないように、Aに送り出す音声パケットには、Aの音声成分が含まれないように合成する。

(例) 番号変換アプリケーション

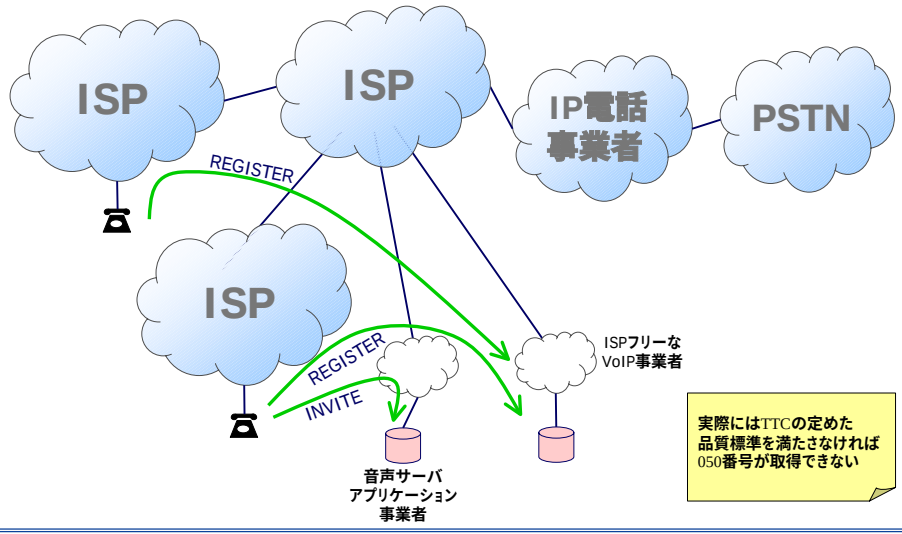


XはAの本当の電話番号を知らなくても
Aに電話をかけられるサービス

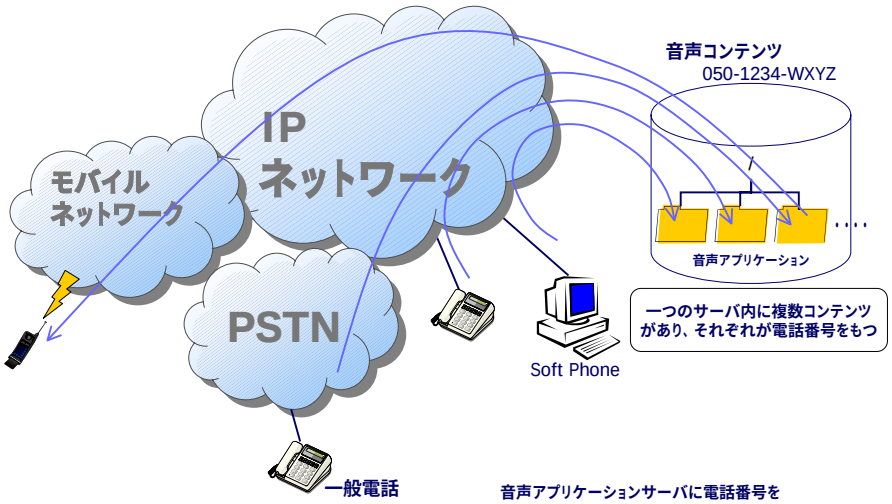
事業者間接続



ISPフリーなVoIP事業者



音声サーバアプリ



音声アプリケーションサーバに電話番号を
割当てるしくみが存在しない
as of 2003/4

パート5

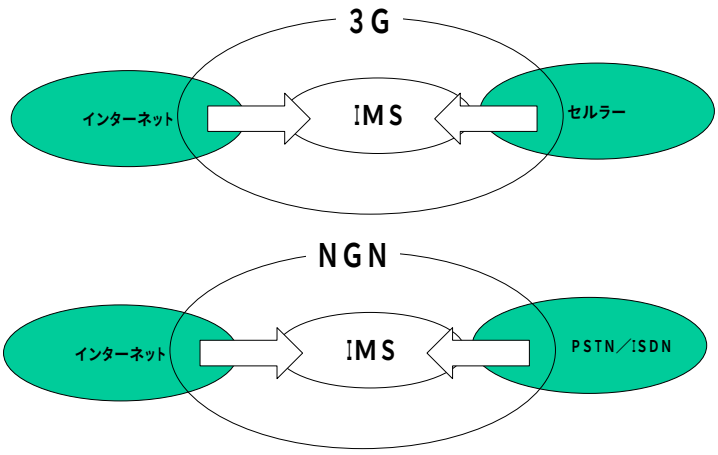
NTTPC COMMUNICATIONS

パート5

3GPP/IMS/SIP/NGN/RFC3261

3GとNGN

NTTPC COMMUNICATIONS



第3世代携帯

- 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

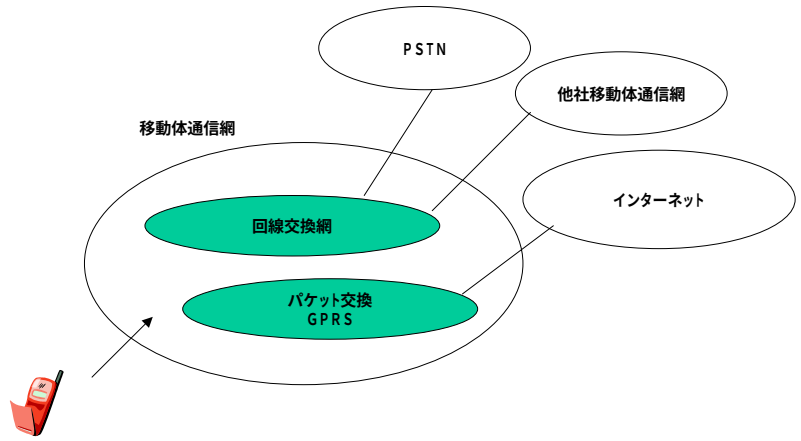
すべてを実現したい

3GPP と 3GPP2

- 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



インターネットもできる
電話もできる

現在の３Ｇで何が問題なのか？

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、

3GPPとIETFの協調

NTTPC COMMUNICATIONS

- 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

IMSのためのIETFプロトコル

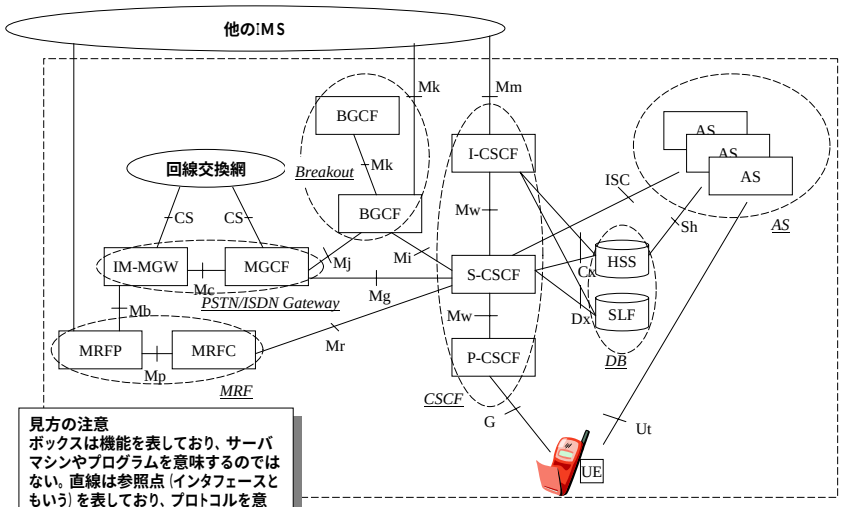
NTTPC COMMUNICATIONS

- 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

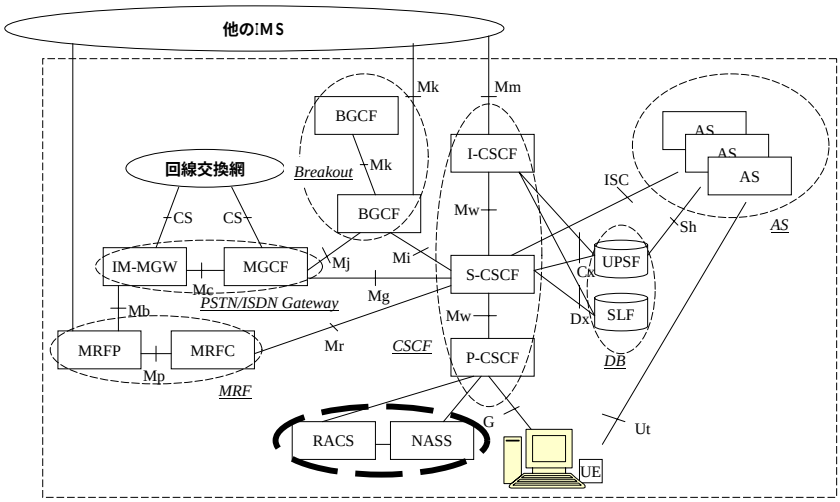
3G IMSアーキテクチャ



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プロキシーサーバ)
  - 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、ユーザプロファイルに基づく接続設定
  - RACS
    - ユーザプロファイルに基づくアドミション制御、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
      - コールウェイティング、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

まとめ

NTTPC COMMUNICATIONS

- 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