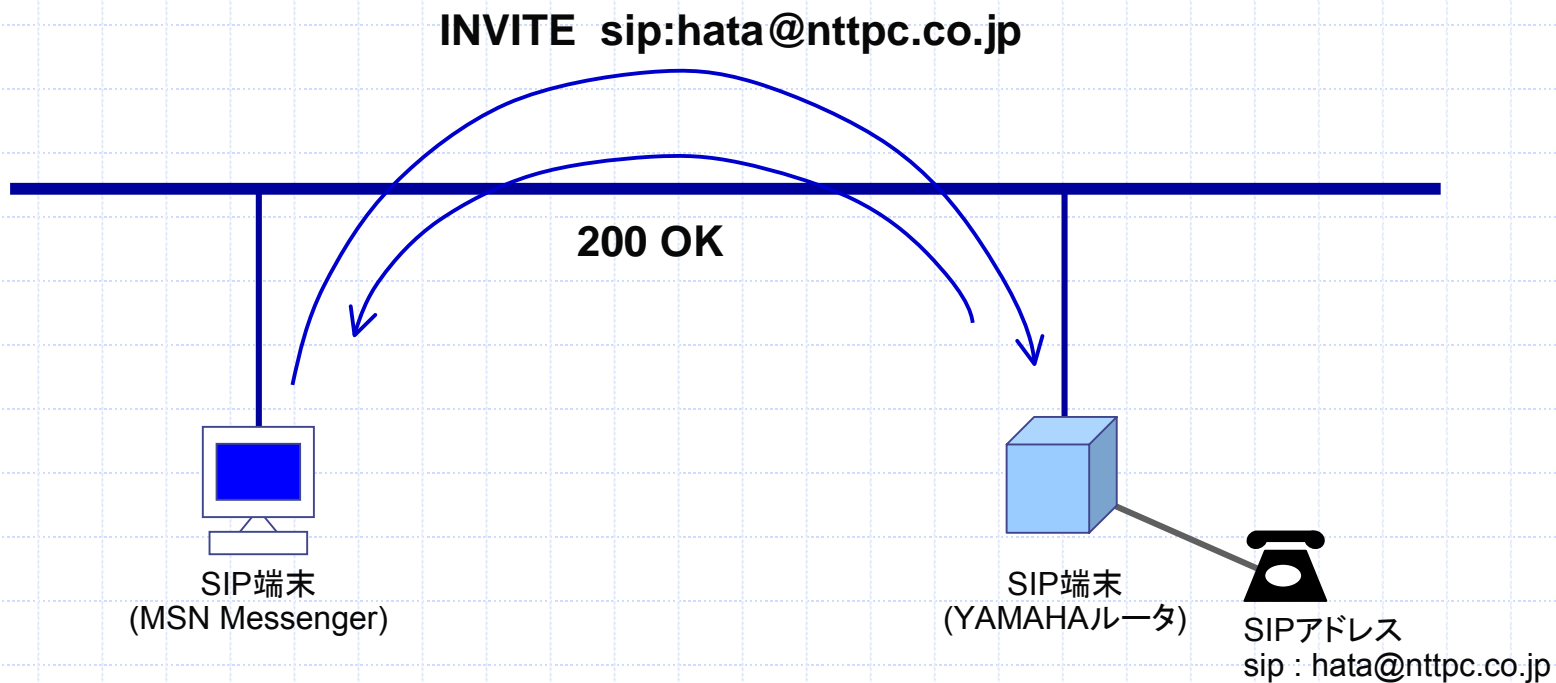


# SIPとVoIP

NTTPC Communications, Inc. 波多 浩昭

# 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



# SIPリクエストメッセージ

```
INVITE sip:1003@nttpc.ne.jp SIP/2.0
Call-ID:238bc-25adc-1532b5@192.168.0.32
From:1001<sip:1001@nttpc.ne.jp>;tag=176866
To:sip:1003@nttpc.ne.jp
CSeq:100 INVITE
Via:SIP/2.0/UDP 192.168.0.32
Contact:1001<sip:1001@192.168.0.32>
Content-Type:application/sdp
Content-Length:110
```

```
v=0..o=1001 0 0 IN IP4 192.168.0.32
s=-
c=IN IP4 192.168.0.32
t=0 0
m=audio 8002 RTP/AVP 0
a=ptime:30
```

ファーストラインメソッドとあて先

ヘッダ部 SIPで規定

空行をあけてヘッダと本体を区切る

SDP

192.168.0.32のポート8002で  
RTPの上のG.711を受信できますよ  
パケットは30ミリ秒ごとにでます

# SIPリクエストメッセージ

200 OK

リザルトコード

Via: SIP/2.0/UDP 192.168.0.1:5060;branch=z9hG4bK135

Via: SIP/2.0/UDP 192.168.0.32:5060

From: <sip:1001@nttpc.ne.jp>;tag=176866

To: <sip:1003@nttpc.ne.jp>;tag=b8b9ba94-ea53-4ed2-84b9-6e9d730c605d

Call-ID: 238bc-25adc-1532b5@192.168.0.32

CSeq: 100 INVITE

Record-Route: <sip:NTTPC135@192.168.0.1:5060>

Contact: <sip:192.168.0.10:9063>

User-Agent: Windows RTC/1.0

Content-Type: application/sdp

Content-Length: 154

v=0

o=hata 0 0 IN IP4 192.168.0.10

s=-

c=IN IP4 192.168.0.10

b=CT:1000

t=0 0

m=audio 41794 RTP/AVP 0

a=rtpmap:0 PCMU/8000

SDP

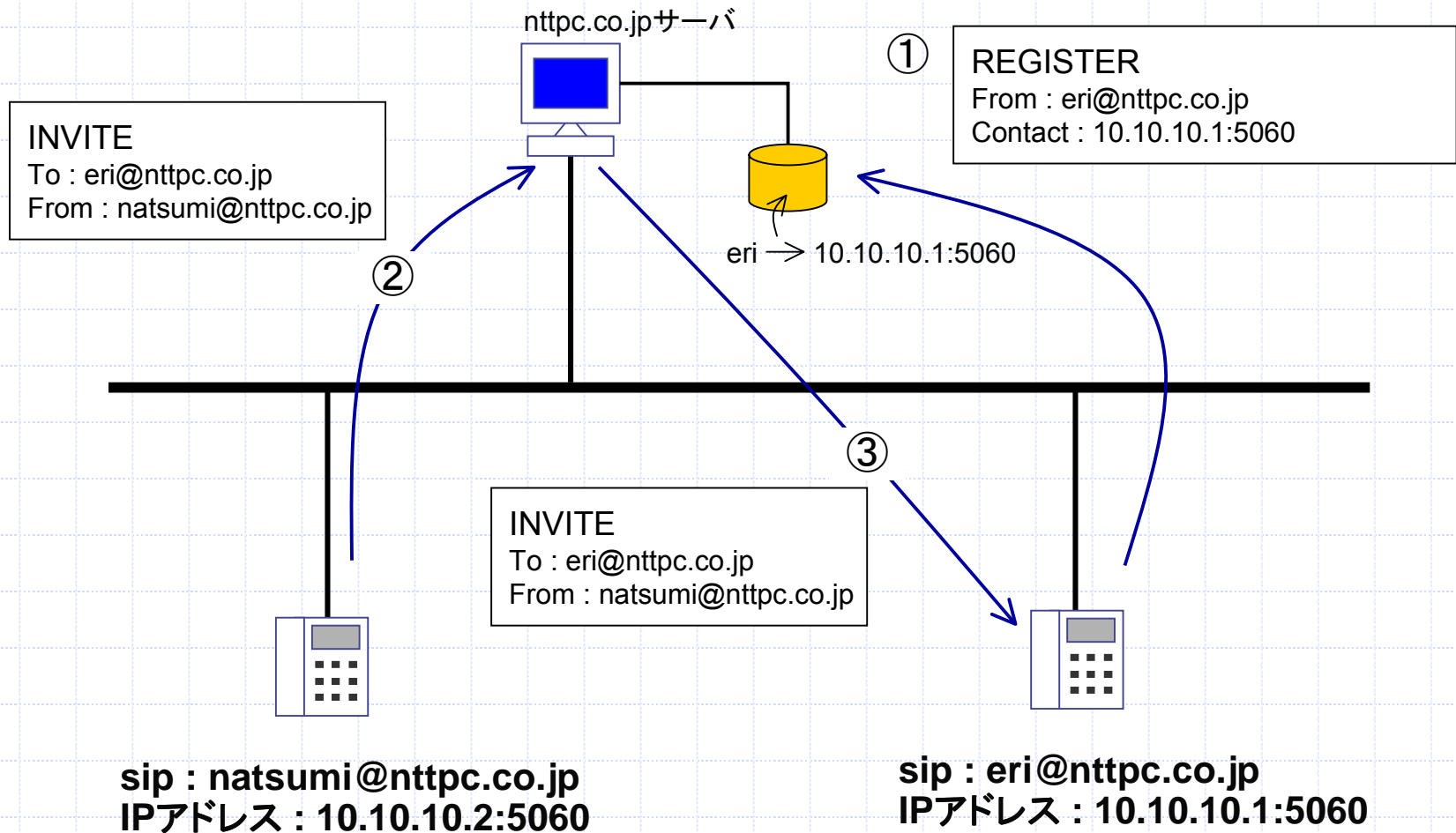
192.168.0.10のポート41794で  
RTPの上のG.711を受信でOKです

## 代表的なヘッダ

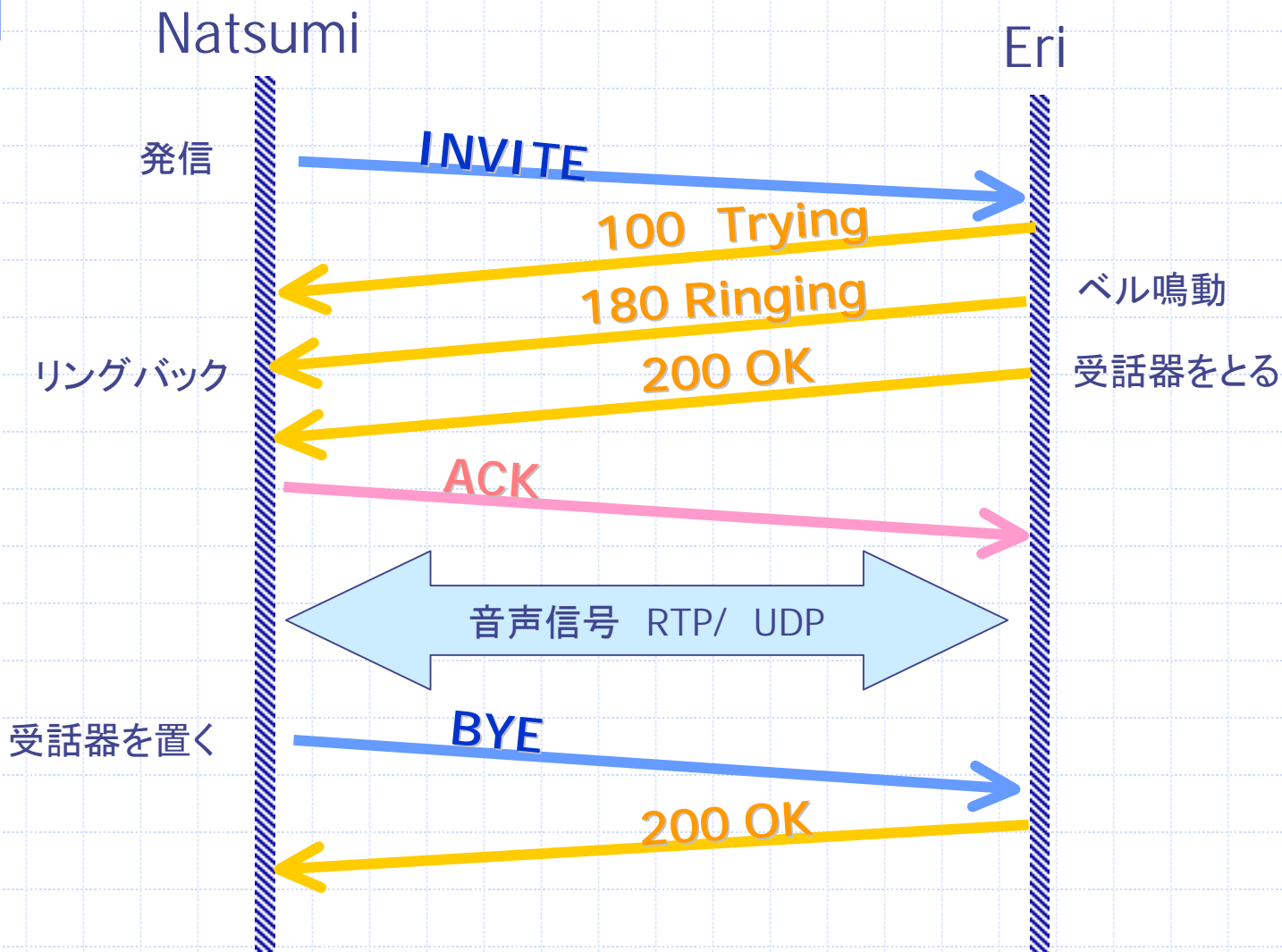
- To** : あて先のSIP-URL
- From** : 発信元のSIP-URL
- Call-ID** : 1回Registerしてから解除するまでの一連のセッションを識別
- CSeq** : 同一Call-IDで何個目のリクエストかを表示
- Via** : 本リクエストに対するレスポンスはここへ送ってほしい旨通知
- Contact** : 以後、自分に対するリクエストはここへ送ってほしい旨通知
- Content-Type** : メッセージボディのMIMEタイプ

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

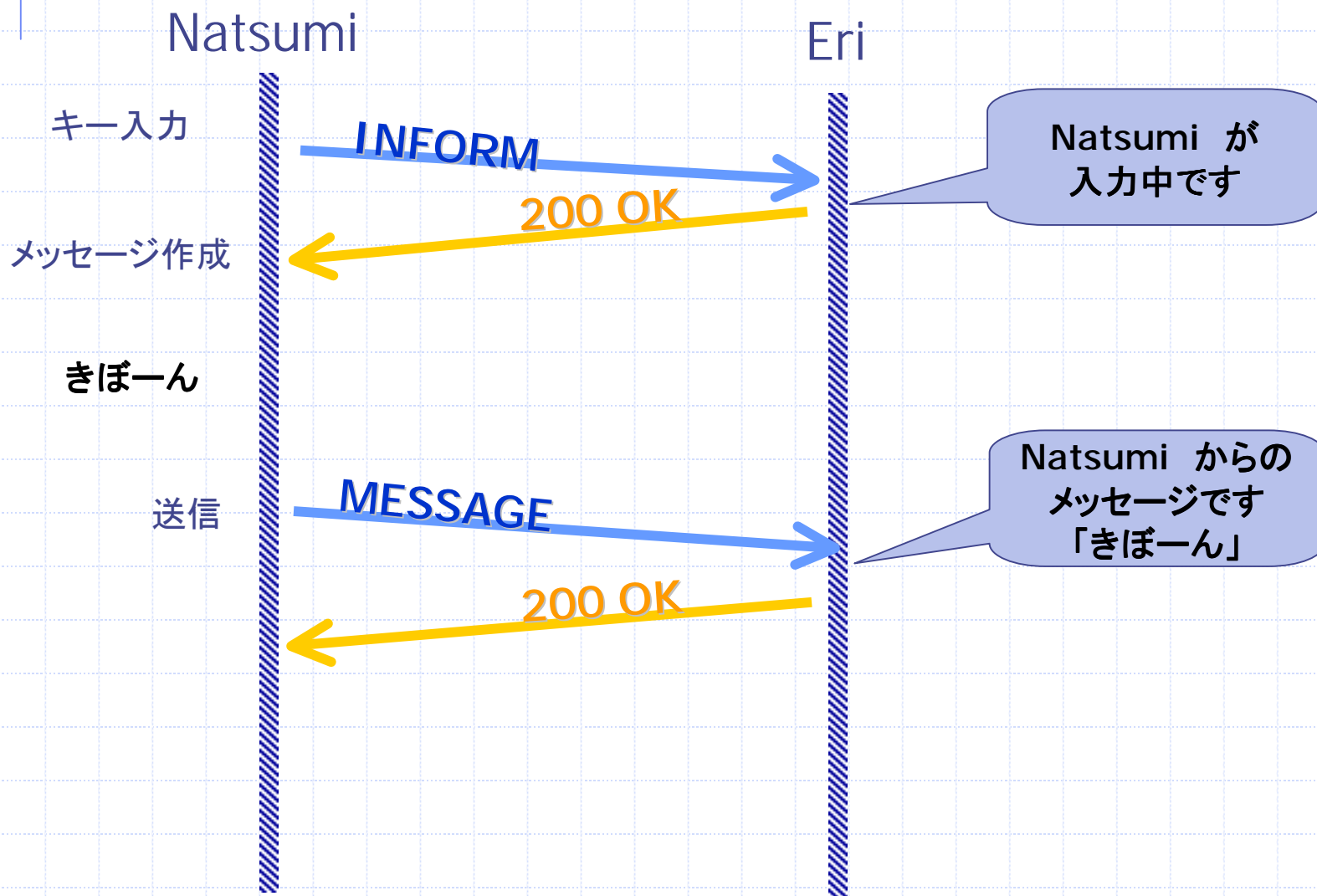
# SIPサーバ(P2PのSIPじゃあつまらない)



# 代表的なシーケンス(音声通話)

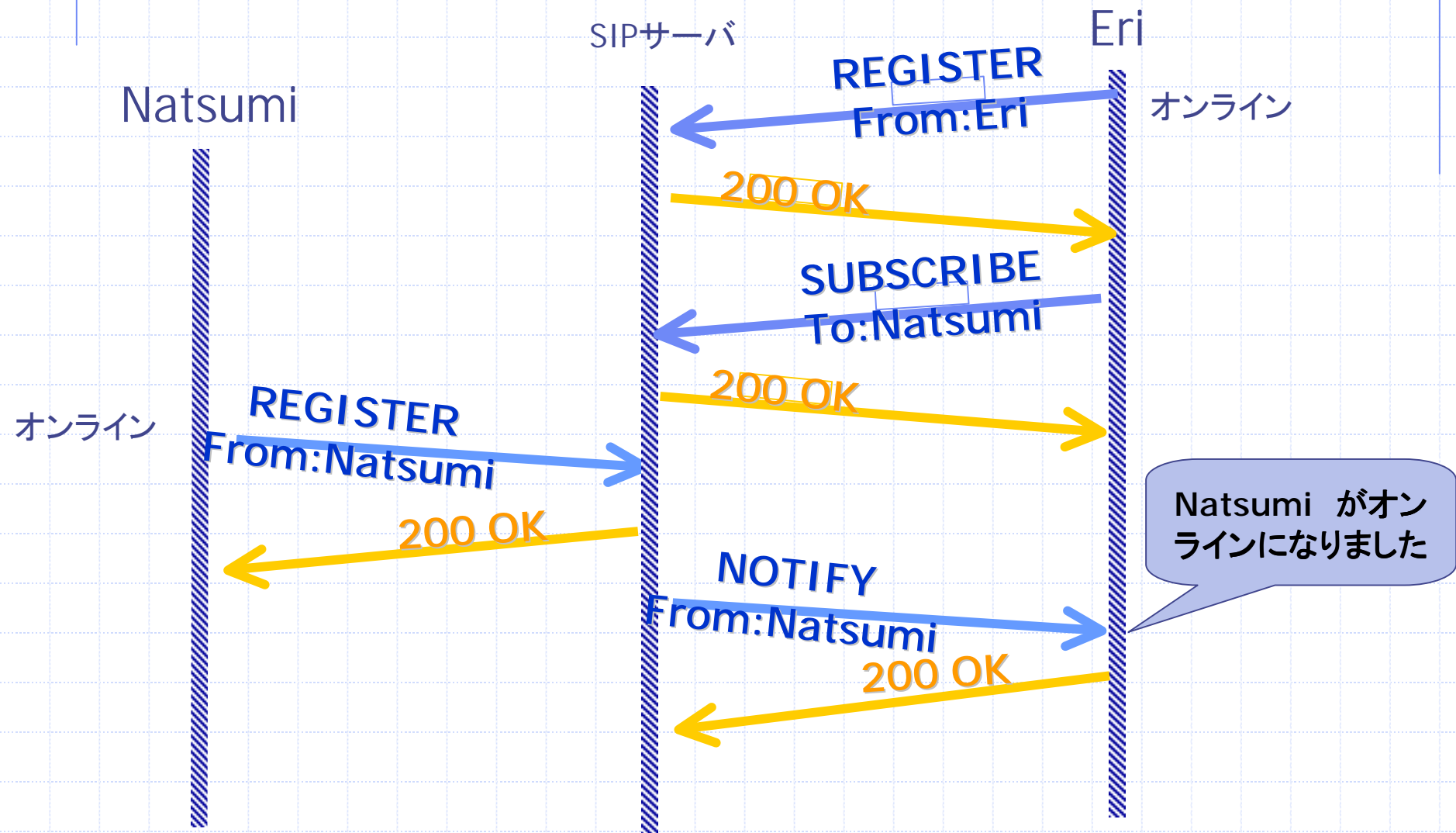


## 代表的なシーケンス(インスタントメッセージ IM)





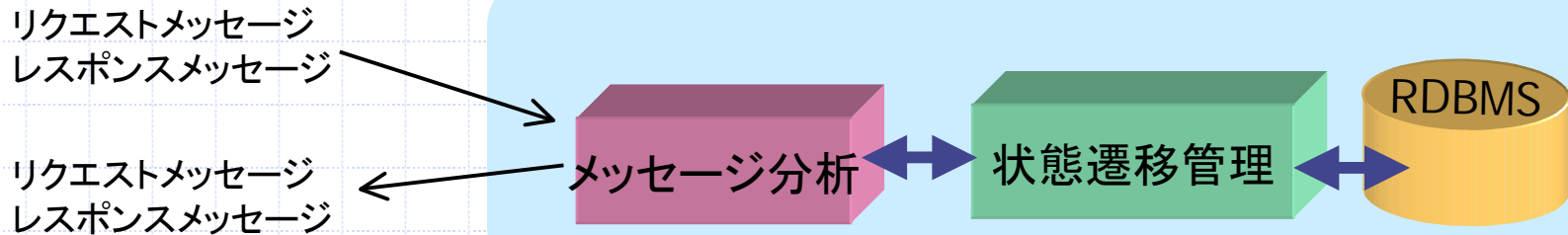
# 代表的なシーケンス(プレゼンス)



# SIPサーバアーキテクチャと設計概要

こんな感じでやれば、動くんじゃないかな？

## SIPサーバ



### SIPサーバとはなにか？

何かメッセージを受信してそれを入力とする。  
入力を分析し、適切な出力をメッセージとして送信する。

### メッセージ分析モジュール

メッセージを受けて、それが何を意味しているのか解釈する。  
返送メッセージを適切なメッセージフォーマットに整形し、ネットワークに送出する

### 状態遷移管理モジュール

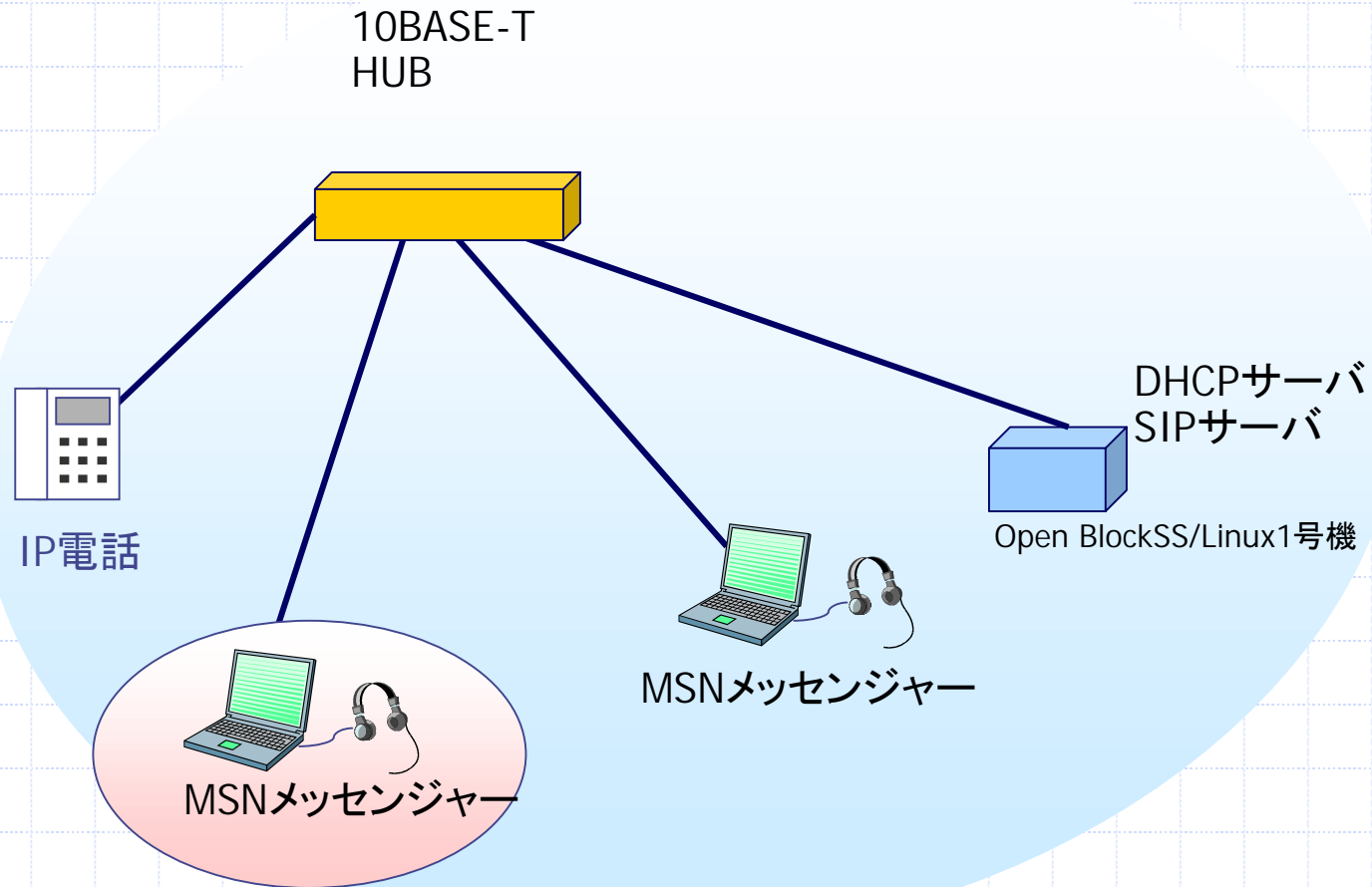
受け取った入力から、誰に何を返すべきかを決定する判断エンジン

### RDBMSモジュール

現在どのような状態にあるのか(誰が、オンライン中で誰が通話中か)を保持する

というわけで作って動かしてみました。  
クライアントは只でもらえるmsnメッセンジャーです

# デモ環境



# IP電話機のインタフェース

## IP電話機



## SIPサーバ

# 品質

## ◆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)

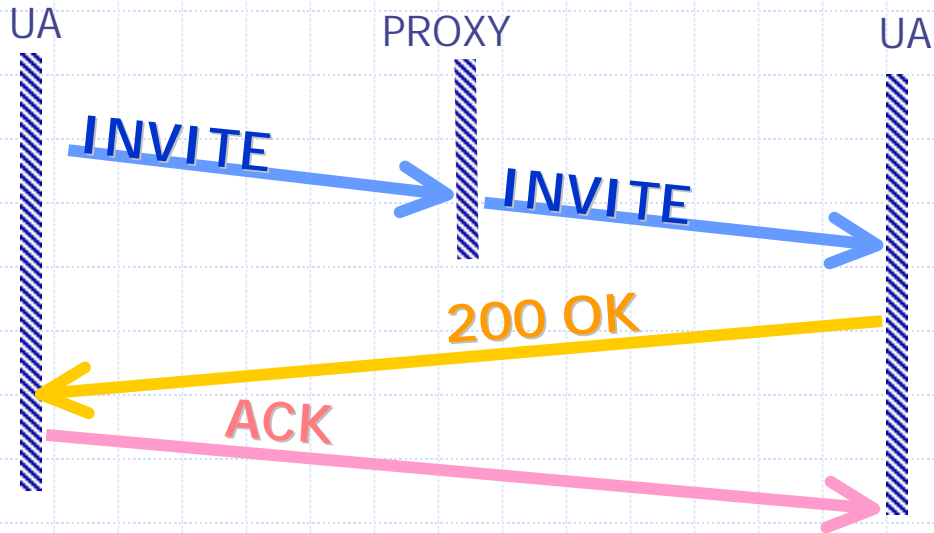
## ◆遅延

- 音声おくれ

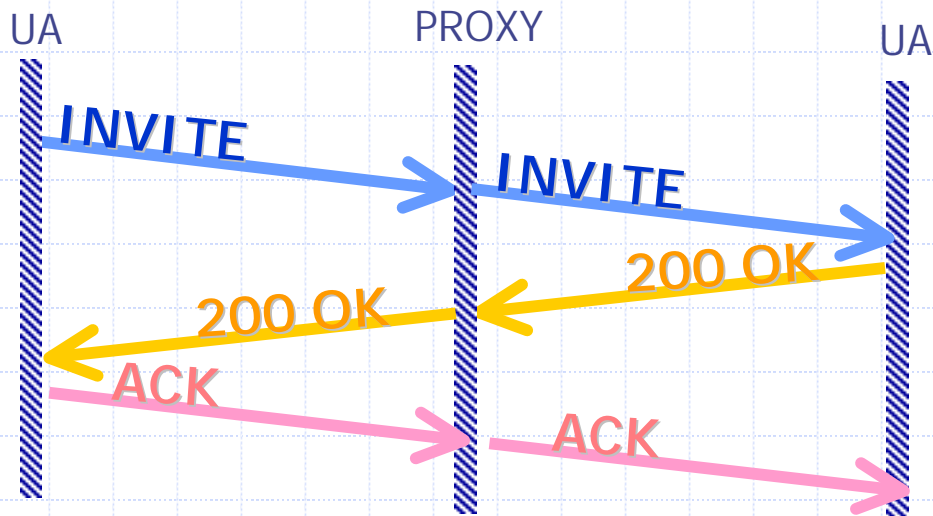
## ◆ジッタ(含パケット逆転)

- 音声とぎれ
- パケットロスに似ている

# 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

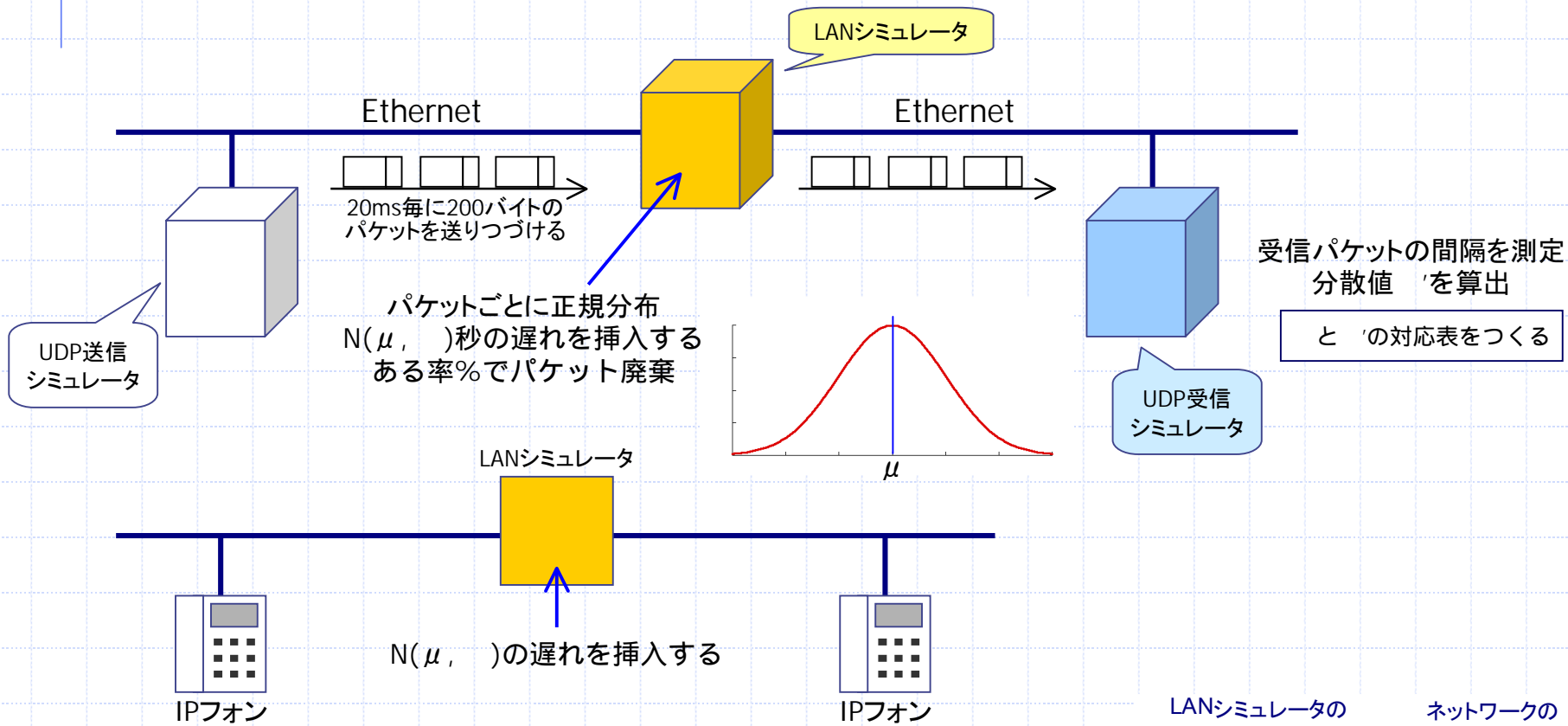


通過するパケットはプロキシ  
に転送してくださいね。

- ・Viaスタック
- ・Record-Route

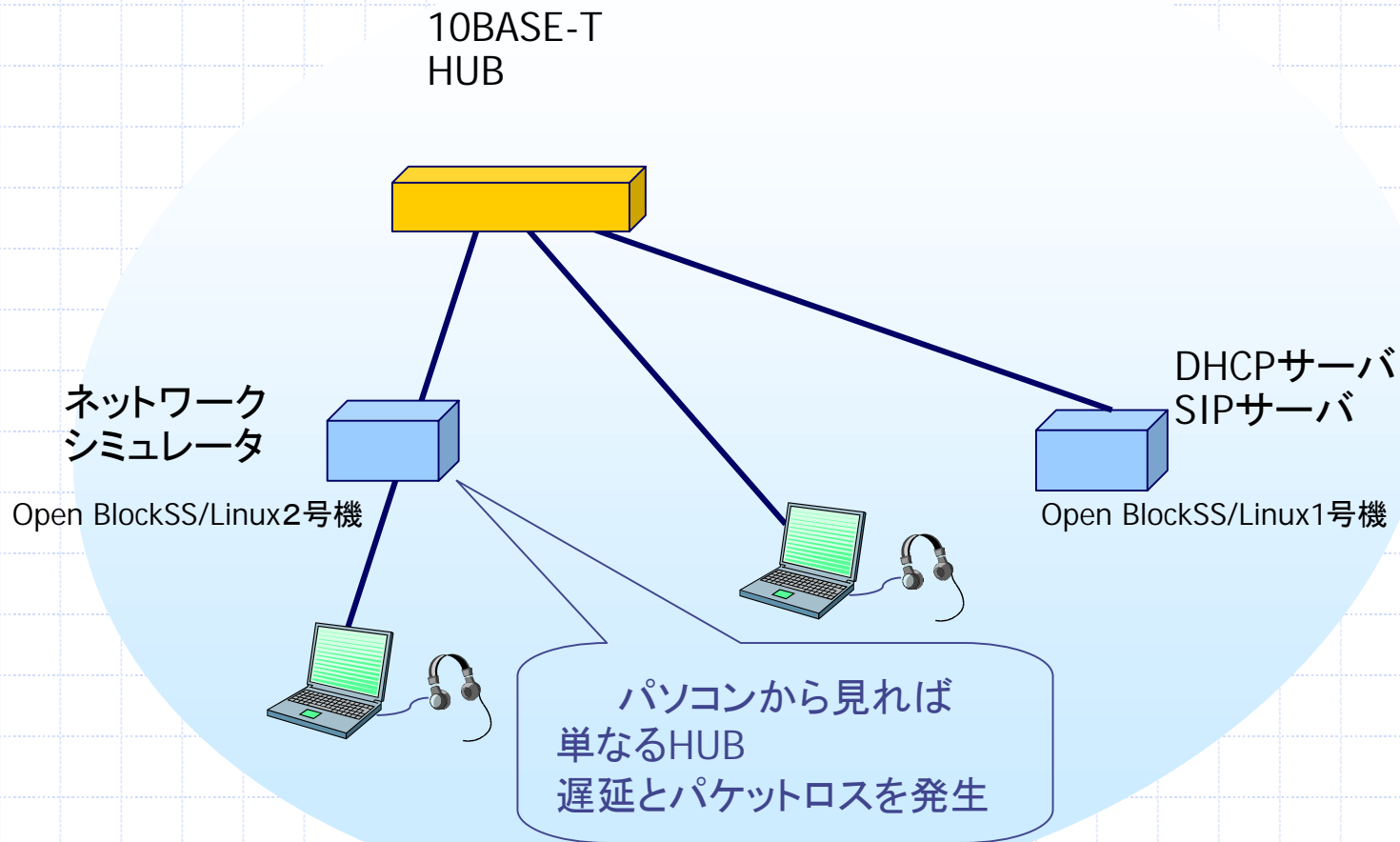
# シミュレーション

～網品質をどこまで保証すればよいのだろう～



	LANシミュレータのエラーパラメータ	ネットワークの推定値
良好	$\mu, \sigma, \%$	$\mu, \sigma, \%$
あれ?変だな		
こりやまずい		
使えない		

## デモ環境2





## 本セッションで利用した機材・ソフト・ドキュメント類

### SIPサーバ

<http://www.hata.cc/hata/sipd20030707.tar.gz>

### イーサネットシミュレータ

<http://www.hata.cc/hata/linee.tar.gz>

### イーサネットシミュレータの詳細解説 (PC-View)

<http://www.pc-view.net/Column/030110/index.html>

### 揺らぎ測定ツール

<http://www.hata.cc/hata/netest.tar.gz>

### MSNメッセージャー

Version4. 6もしくは4. 7だけがSIP対応

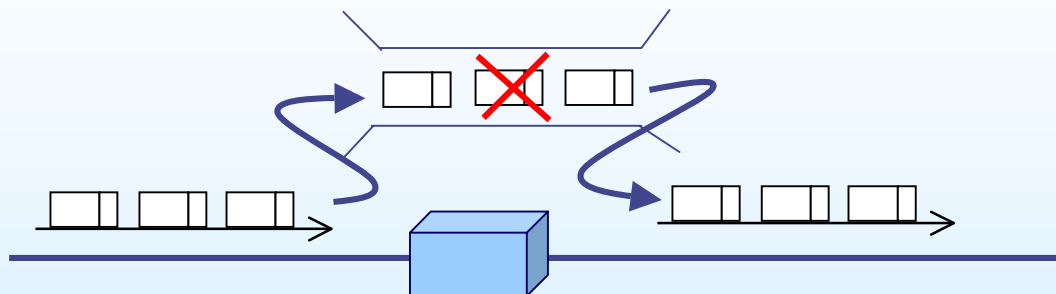
### IP電話機

東亜通信工業股分有限公司  
日本弁事処 045-474-3222

# ネットワークシミュレータの動作

## 基本動作

イーサポートに現れたフレームはMACアドレスやIPアドレスが何であろうともう一方のインタフェースに送出する



## オプション動作

フレーム毎にキューに滞留する時間をランダム、正規分布、固定のいずれかで指定できる

たまにフレームを廃棄する(他方のインタフェースに送らない)。バースト的に廃棄するか、ランダムに廃棄するかの指定ができる