

# 情報ネットワーク

(明石高専 電気情報工学科 5年)

---

第9回

2005年1月11日(火)

# 前回の復習

---

## ■ DHCP

- IPアドレスを自動的に割り当てるプロトコル
  - 設定された範囲のIPアドレスを貸し出す(一定期間)
  - TCP/IPの通信に必要な情報も提供

## ■ NAT

- IPアドレスの変換
  - LAN(プライベートIP)とインターネット(グローバルIP)の間
  - IPアドレスの枯渇問題への対応策
- NATPを使えば、
- NAT対応ルーターが必要

# 今回の内容

---

- 第6章 TCPとUDP (p.178 ~ 197)
  - 6.1 トランスポート層の役割
  - 6.2 ポート番号
  - 6.3 UDP  
(User Datagram Protocol)
  - 6.4 TCP  
(Transmission Control Protocol)
    - 6.4.7 ウィンドウ制御と再送制御

# トランスポート層の役割

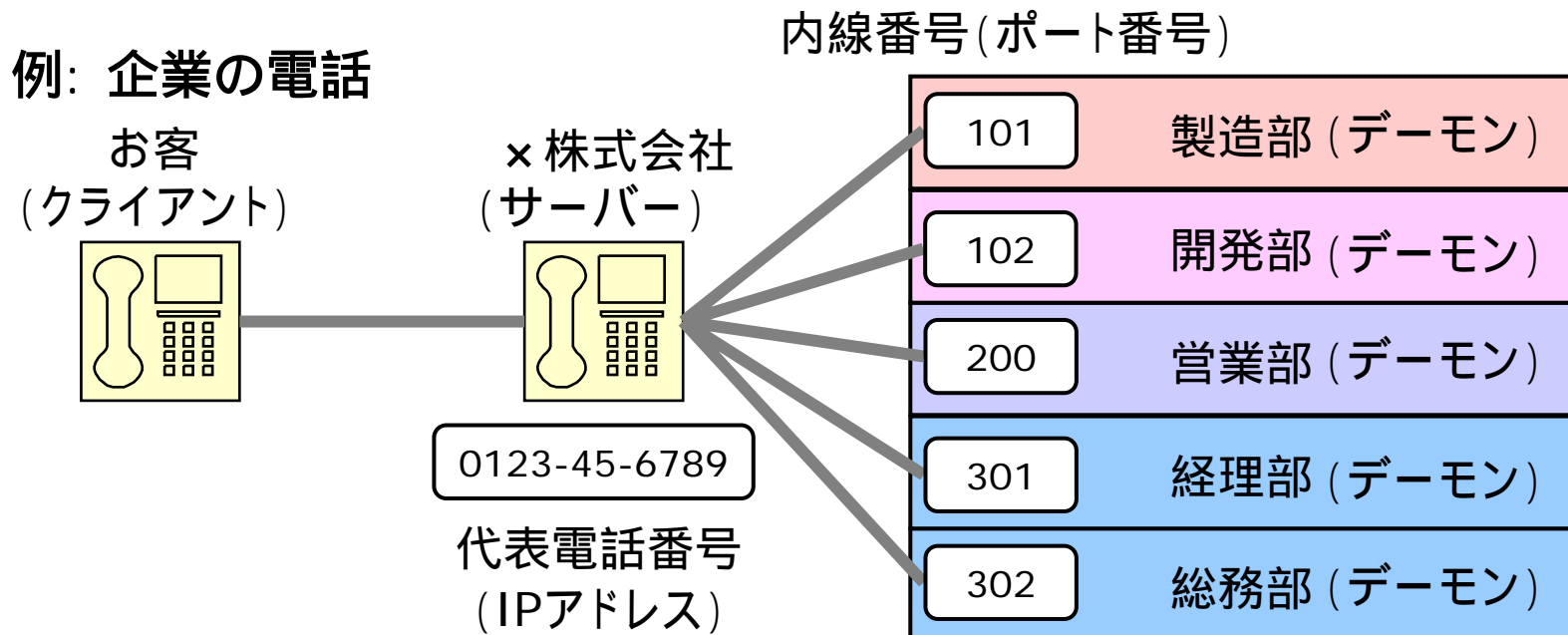
## ■ 通信の信頼性

- トランスポート層のプロトコル: **TCP**と**UDP**

## ■ アプリケーションソフト間の通信

- アプリケーションの指定: **ポート番号**

例: 企業の電話



# TCPとUDP

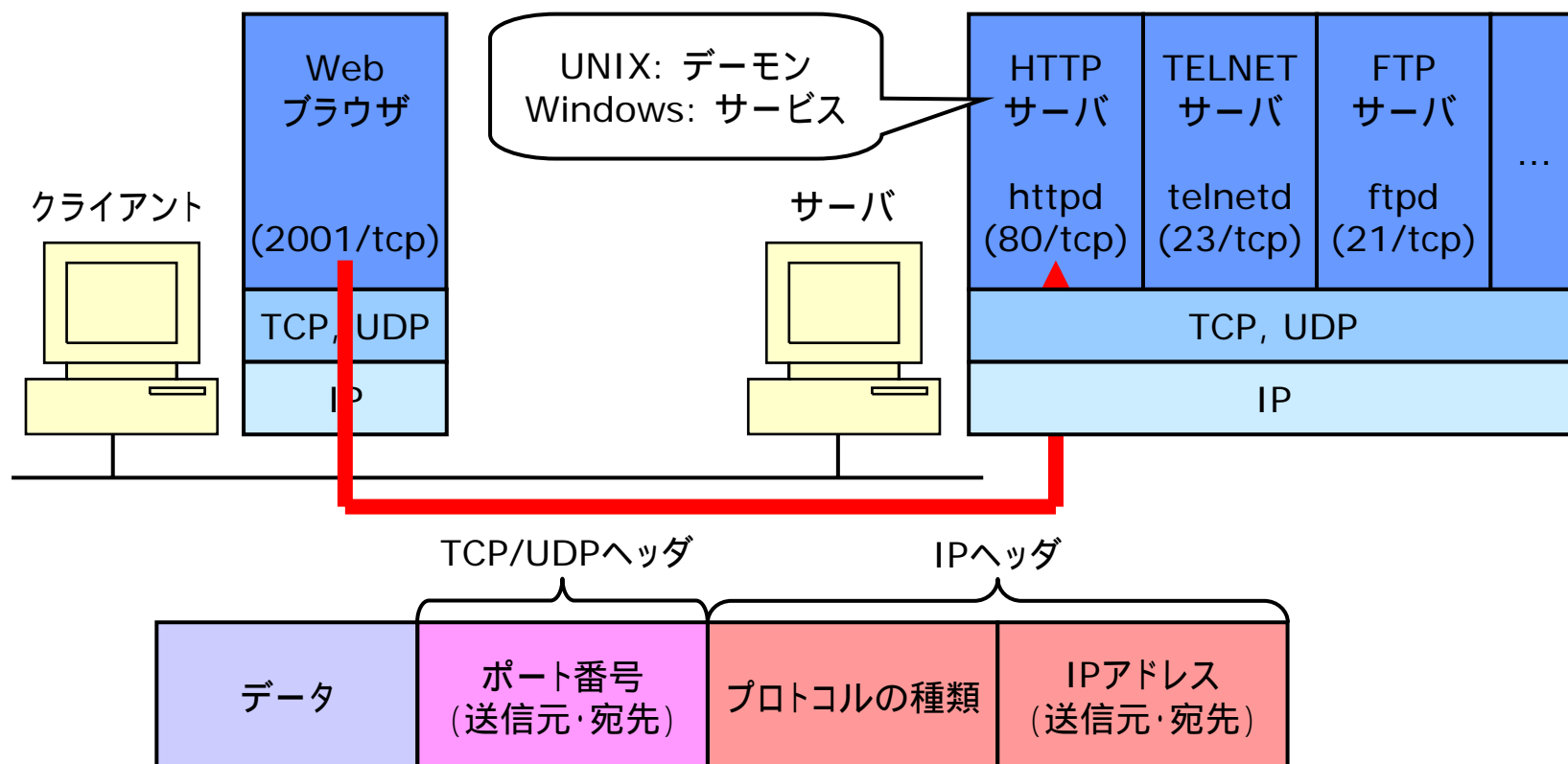
---

- TCP (Transmission Control Protocol)
  - **ストリーム型**プロトコル
    - コネクションを張ってから通信(コネクション志向)
    - **信頼性が高い、高速ではない**(様々な制御処理)
- UDP (User Datagram Protocol)
  - **データグラム型**プロトコル
    - いきなりデータを送信(コネクションレス志向)
    - **信頼性が低い、高速、ブロードキャスト可能**

それぞれのプロトコルの詳細は、後ほど...

# ポート番号

- アプリケーションソフトを識別する番号
  - クライアント / サーバ (C/S) 方式



# ポート番号(つづき)

---

- **ポート番号によるアプリケーション間の通信**
  - 送信元と宛先の両方に必要
  - IPアドレス・ポート番号・プロトコルで通信を識別
- **ポート番号の決め方**
  - 標準として決定
    - Well-known Port Number (0 ~ 1023)  
よく使われるアプリケーション用
  - ダイナミック(動的)な割り当て
    - クライアント側のポート番号 (49152 ~ 65535)
    - 番号が重複しないようにOS側で制御

一般には1024 ~

<http://www.iana.org/assignments/port-numbers>

/etc/services  
(RedHat Linux 7.2)

Port Assignments:

Keyword	Decimal	Description	References
-----	-----	-----	-----
daytime	13/tcp	Daytime (RFC 867)	
daytime	13/udp	Daytime (RFC 867)	
#		Jon Postel <postel@isi.edu>	
#	14/tcp	Unassigned	
#	14/udp	Unassigned	
#	15/tcp	Unassigned [was netstat]	
#	15/udp	Unassigned	
#	16/tcp	Unassigned	
#	16/udp	Unassigned	
qotd	17/tcp	Quote of the Day	
qotd	17/udp	Quote of the Day	
#		Jon Postel <postel@isi.edu>	
msp	18/tcp	Message Send Protocol	
msp	18/udp	Message Send Protocol	
#		Rina Nathaniel <---none--->	
chargen	19/tcp	Character Generator	
chargen	19/udp	Character Generator	
ftp-data	20/tcp	File Transfer [Default Data]	
ftp-data	20/udp	File Transfer [Default Data]	
ftp	21/tcp	File Transfer [Control]	
ftp	21/udp	File Transfer [Control]	
#		Jon Postel <postel@isi.edu>	
ssh	22/tcp	SSH Remote Login Protocol	
ssh	22/udp	SSH Remote Login Protocol	
#		Tatu Ylonen <ylo@cs.hut.fi>	
telnet	23/tcp	Telnet	
telnet	23/udp	Telnet	
#		Jon Postel <postel@isi.edu>	
	24/tcp	any private mail system	
	24/udp	any private mail system	
#		Rick Adams <rick@UUNET.UU.NET>	
smtp	25/tcp	Simple Mail Transfer	
smtp	25/udp	Simple Mail Transfer	

tcpmux	1/tcp		
tcpmux	1/udp		
rje	5/tcp		# Remote Job Entry
rje	5/udp		# Remote Job Entry
echo	7/tcp		
echo	7/udp		
discard	9/tcp		sink null
discard	9/udp		sink null
systat	11/tcp		users
systat	11/udp		users
daytime	13/tcp		
daytime	13/udp		
qotd	17/tcp		quote
qotd	17/udp		quote
msp	18/tcp		
msp	18/udp		
chargen	19/tcp		ttytst source
chargen	19/udp		ttytst source
ftp-data	20/tcp		
ftp-data	20/udp		
ftp	21/tcp		
ftp	21/udp		
ssh	22/tcp		
ssh	22/udp		
telnet	23/tcp		
telnet	23/udp		
#	24 - private mail system		
smtp	25/tcp		mail
smtp	25/udp		mail
time	37/tcp		timserver
time	37/udp		timserver
rlp	39/tcp		resource # resource location
rlp	39/udp		resource # resource location
nameserver	42/tcp		name # IEN 116
nameserver	42/udp		name # IEN 116
nicname	43/tcp		whois
nicname	43/udp		whois



# UDP (User Datagram Protocol)

## ■ アプリケーション側で制御 (ユーザ次第)

- 信頼性が低い
  - 複雑な制御がない (再送、送信量などの制御)
- いきなり通信を開始できる
  - コネクションレス志向
- 高速通信・処理が可能
  - UDPヘッダが小さい (TCPの4割)

UDPヘッダ (64bit)

## ■ 向いている用途

- 総パケット数が少ない通信
- マルチメディア (音声・動画)
- 同報性 (ブロードキャスト、マルチキャスト)

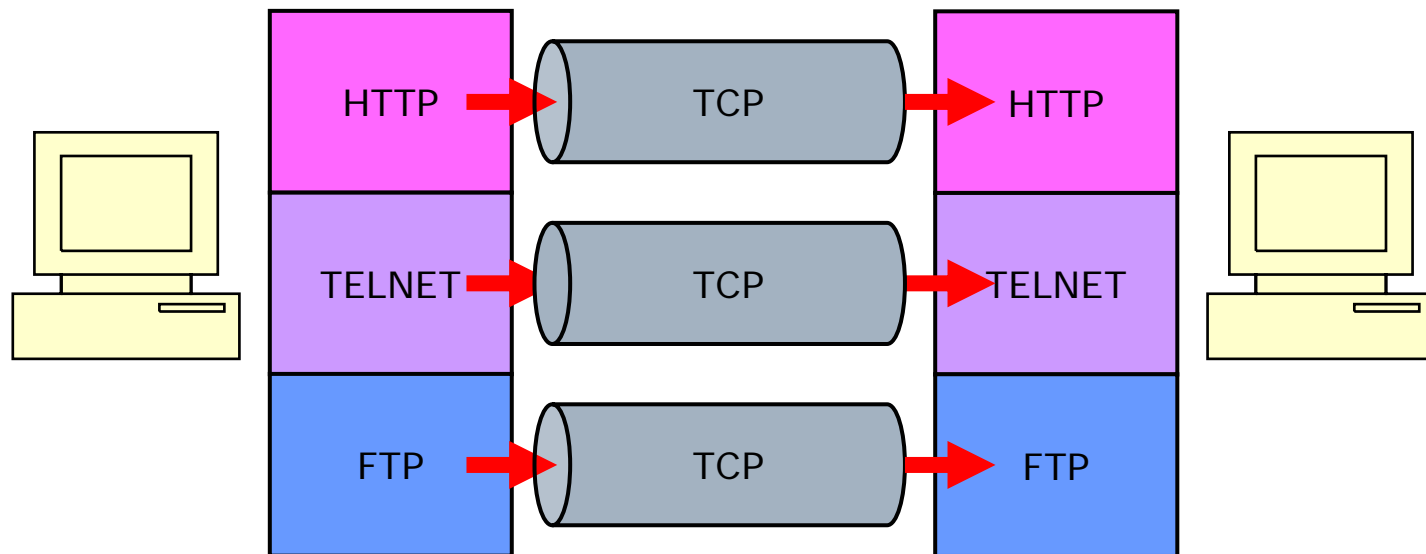
送信元ポート番号 (16bit)	宛先ポート番号 (16bit)
パケット長 (16bit)	チェックサム (16bit)

# TCP (Transmission Control Protocol)

## ■ コネクション志向のプロトコル

### — 信頼性が高い

- アプリケーションごとに**仮想回線**を提供
- **通信制御**(再送、順序入替、確認応答)を行う



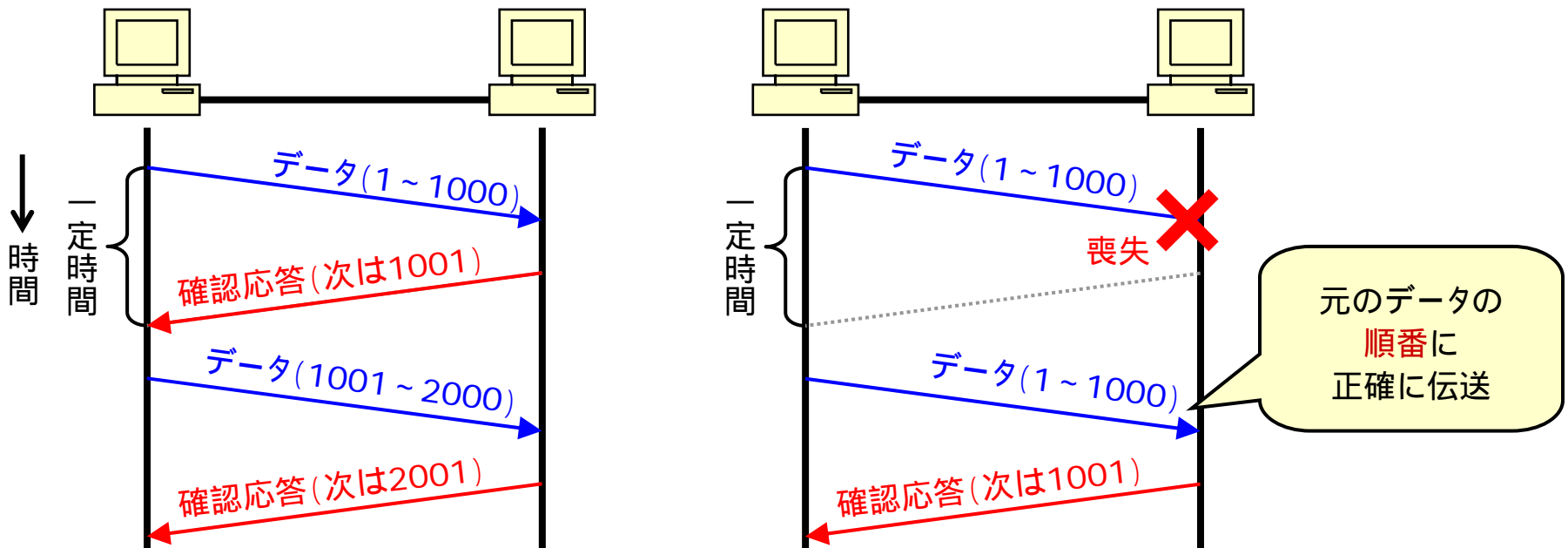
# シーケンス番号と確認応答

## ■ シーケンス番号

- 送信データに**番号**をつける  
(1オクテットごと)
- 初期値: コネクション確立時にランダムに設定

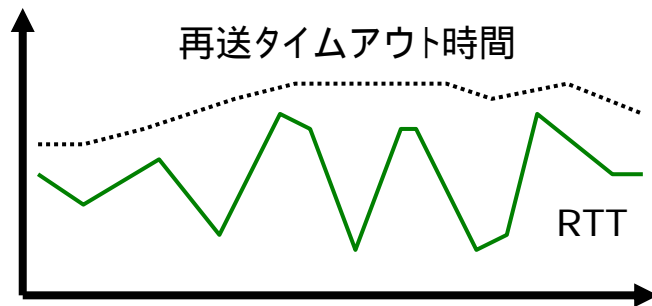
## ■ 確認応答

- データの到着と次のシーケンス番号を通知
- 一定時間内に確認応答がなければ**データを再送**

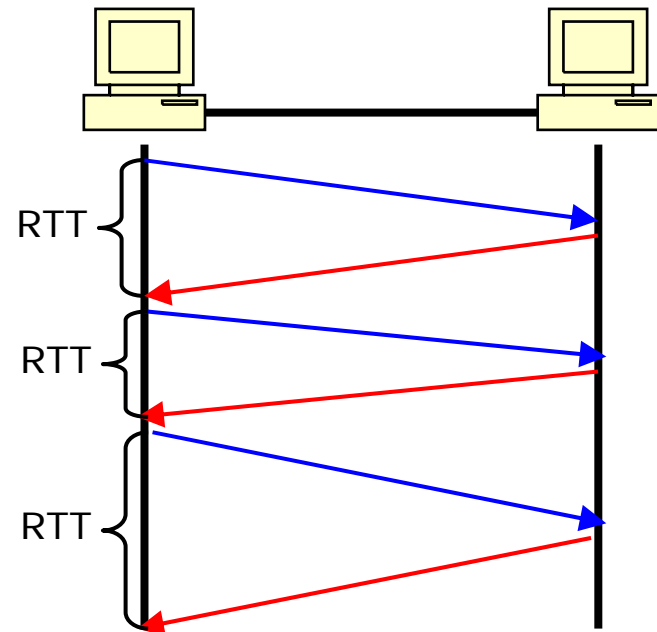


# 再送タイムアウトの決定

- 再送タイムアウト時間で無駄な再送を防ぐ
  - 確認応答を待つ時間
  - RTT: ラウンドトリップ(往復遅延)時間を計測
  - 「再送タイムアウト時間 > RTT + 」となるように決定



UNIXの場合: 0.5秒単位で制御  
(初期値: 6秒程度)

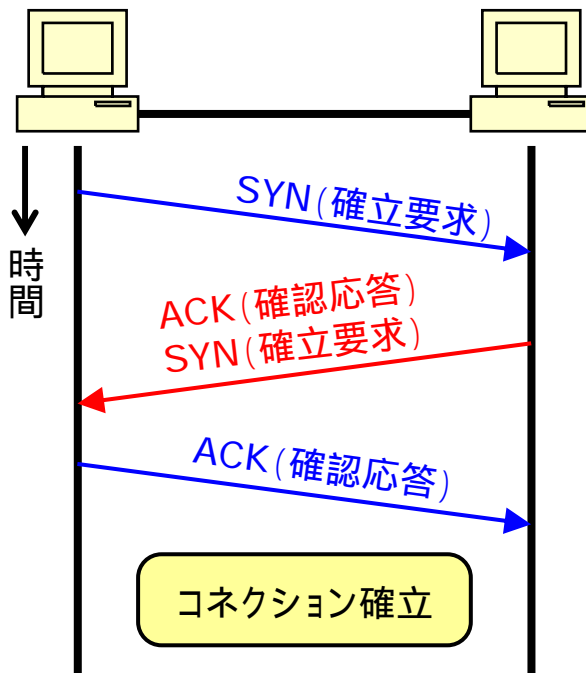


# コネクション管理

## ■ コネクションの確立

SYN    ACK+SYN    ACK

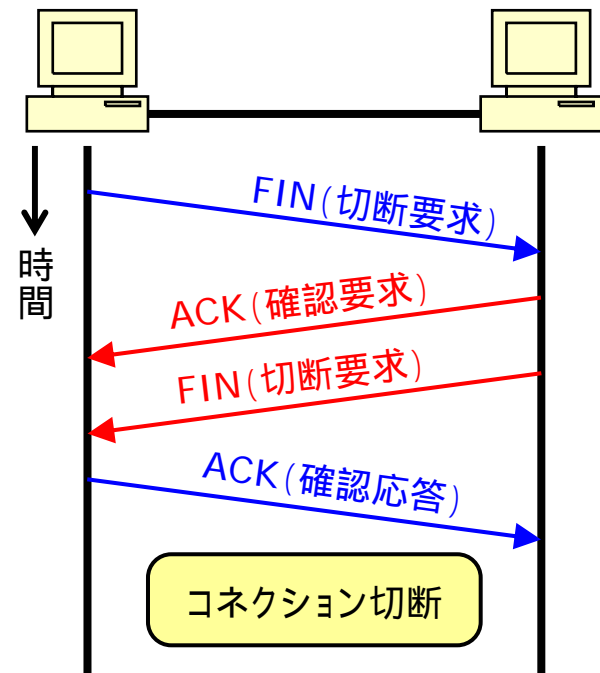
- スリーウェイハンドシェイク  
(3way Handshake)



## ■ コネクションの切断

FIN    ACK、FIN    ACK

- 両方から終了しないと切断できない



# セグメント単位でのデータ送信

---

## ■ IPの「経路MTU探索」機能

- 分割処理が起こらないサイズにしてパケット送信  
(p.134 ~ 135)
- 再構築の負荷が減り、ネットワークの利用効率上がる

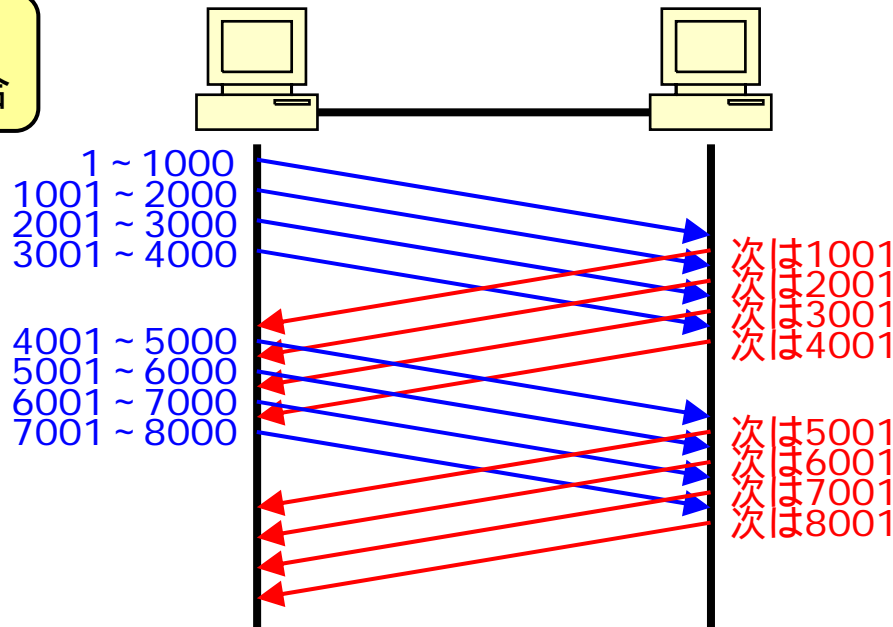
## ■ 最大セグメント長(MSS)

- IPで分割処理されないデータの最大の大きさ
- コネクション確立時に決定
  - SYNパケットを使って送信元に通知

# ウィンドウ制御

- 複数のセグメントをまとめて送信
  - ウィンドウサイズ
    - 確認応答を待たずに送信できるデータの大きさ  
(受信側のバッファの大きさ)

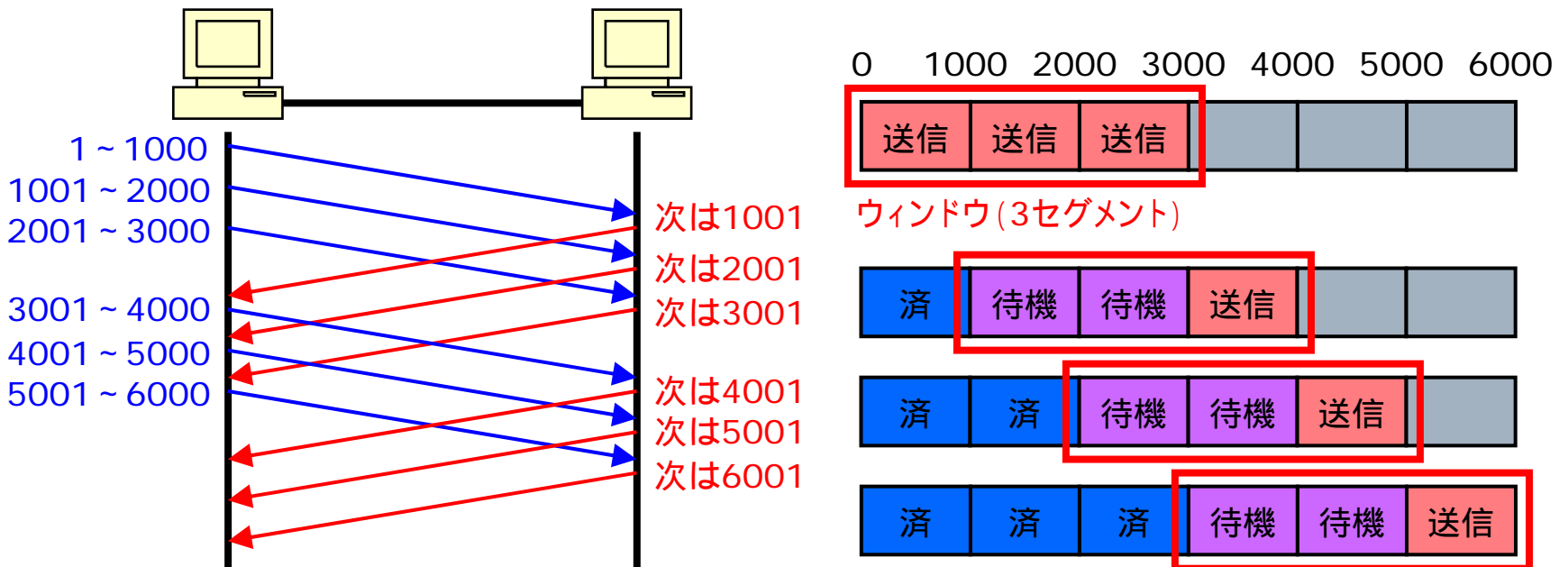
ウィンドウサイズ  
4000オクテットの場合



# ウィンドウ制御(つづき)

## ■ スライディングウィンドウ制御

- 確認応答が到着したらウィンドウをずらす
- ウィンドウの入ったセグメントを送信



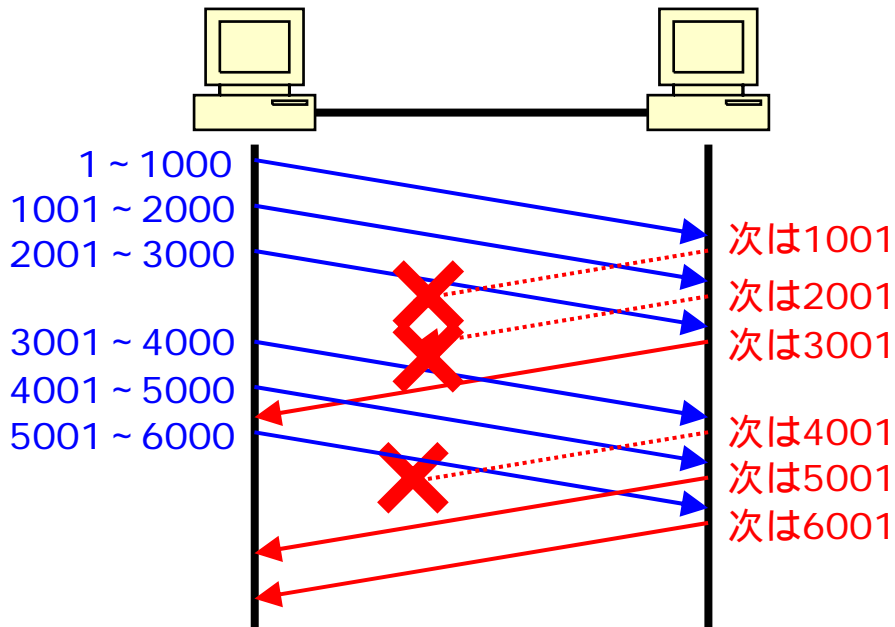


# ウィンドウ制御での再送制御

## ■ 確認応答の喪失

次の確認応答で判断

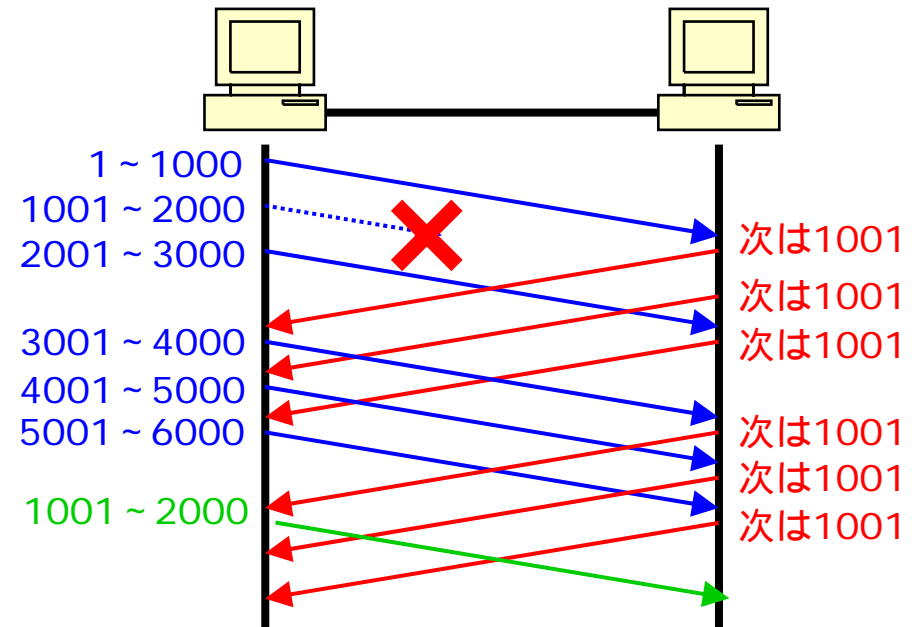
「       までは到着」がわかる



## ■ 送信セグメントの喪失

同じ確認応答を3回連続して受信

確認応答の示すデータを再送



高速再送制御  
(Fast Retransmission)

# 今回のまとめ

---

## ■ ポート番号

- アプリケーション間通信のための識別番号  
(Well-knownポート番号やダイナミックに割り当て)

## ■ UDP

- 信頼性は低い(コネクションレス)
- 高速・同報性(音声・動画などストリーミング向き)

## ■ TCP

- コネクション志向(3ウェイハンドシェイク)
- 信頼性(シーケンス番号、確認応答)
- 再送処理(再送タイムアウト時間)
- 高速化(ウィンドウ制御と再送制御)

# 次回の予定

---

- 次回は1月18日
- 内容
  - 第6章 TCPとUDPの後半
- 連絡事項
  - 第6章の最後 (p.216)まで、通読すること