8. ネットワーク・サービスの基礎プロトコル TCP と UDP

要点

- TCP, UDP では「ポート」によって，複数の通信が並行してできるようにしている.
  - 1 個の IP アドレス (1 台のコンピュータ) で複数のポートがつくれる.
- TCP, UDP では「ポート」によって，さまざまなプロトコルをつかいわけることができる.
  - Web のための HTTP, ファイル転送のための FTP, 電子メールのための SMTP などのプロトコルがある.
- TCP では信頼性・性能のたかい通信を実現している.
  - パケットが脱落しても再送する.
  - 通信路を他の通信とうまくわけあって有効につかうしくみがある.
- TCP は複数のパケットにまたがるメッセージを伝送できる.
  - パケットの最大長 (MTU) にしばられない.

ポート番号と複数の通信

- IP アドレスだけでは 1 台のコンピュータで複数の通信しようとしても，くべつできない．
- TCP, UDP では自分と相手のそれぞれの IP アドレスにくわえて，それぞれのポート番号 (1 ～ 65535) が指定できる．
- ポート番号がちがうと，ことなる通信とみなされる．
  - ◆1 台のコンピュータで複数の通信ができる．

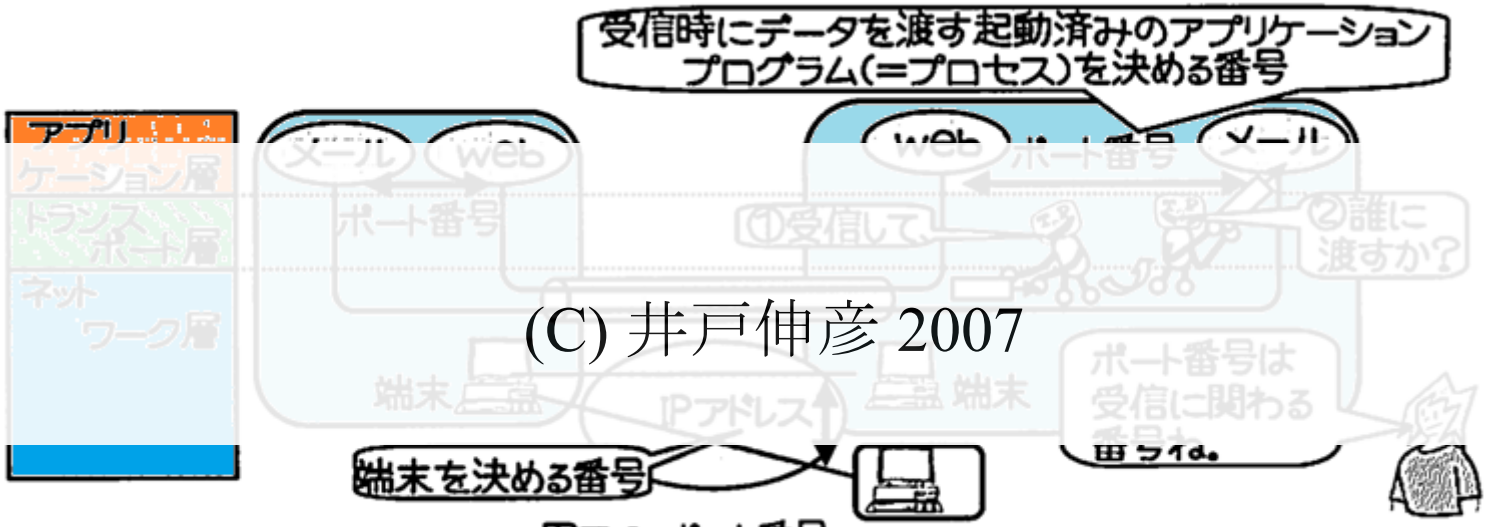


図7-2 ポート番号

ポート番号と複数の通信（つづき）

■ ポートごとに通信相手はことなっていてもよいし，同一でもよい．

織田薫，坪山博貴「図解! よくわかるネットワークの仕組み」，SoftBank Creative

IP ヘッダにおけるポート番号の指定

■「IP プロトコル」は IP ヘッダのプロトコル欄で指定される.

◆ TCP なら 6, UDP なら 17 をここにいれる.

IP プロトコルごとにきめられたポート番号

■ HTTP, FTP などのプロトコルごとに固定のポート番号 (ウェルノウン・ポート) がわりあてられている.

◆ たとえば HTTP では 80 -- Web をつかうときは基本的に 80 を指定.

◆ ウェルノウン・ポート (well-known port) は 1023 まで.

表7・1 主なウェルノンポート番号

アプリケーション	ポート番号	サービス内容			
			DNS	53	ドメインネームの解決(第8章)
FTP	20,21	ファイル転送(第10章)	HTTP	80	webページ(第12章)
SMTP	25	メール転送(第11章)	POP3	110	メール配信(第11章)

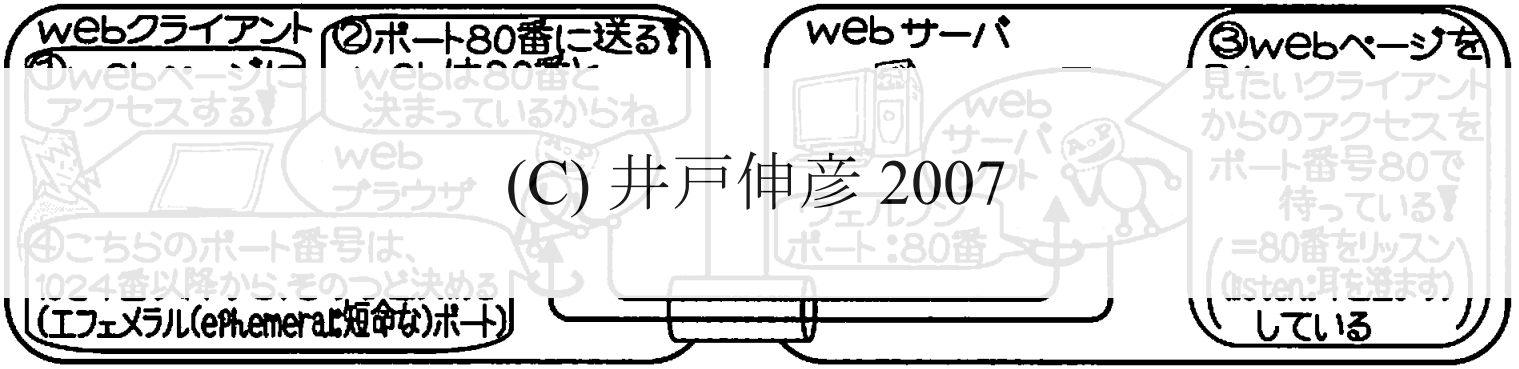


図7・4 ウェルノン(well-known:周知の)ポート

TCP はコネクション型，UDP はコネクションレス型

- TCP は相手と接続してから通信する (= コネクション型).
  - 電話をかけてから話をする，または橋をかけてからとおる ようなもの.
- UDP は接続せずに通信する (= コネクションレス型)
  - 郵便をおくるようなもの.

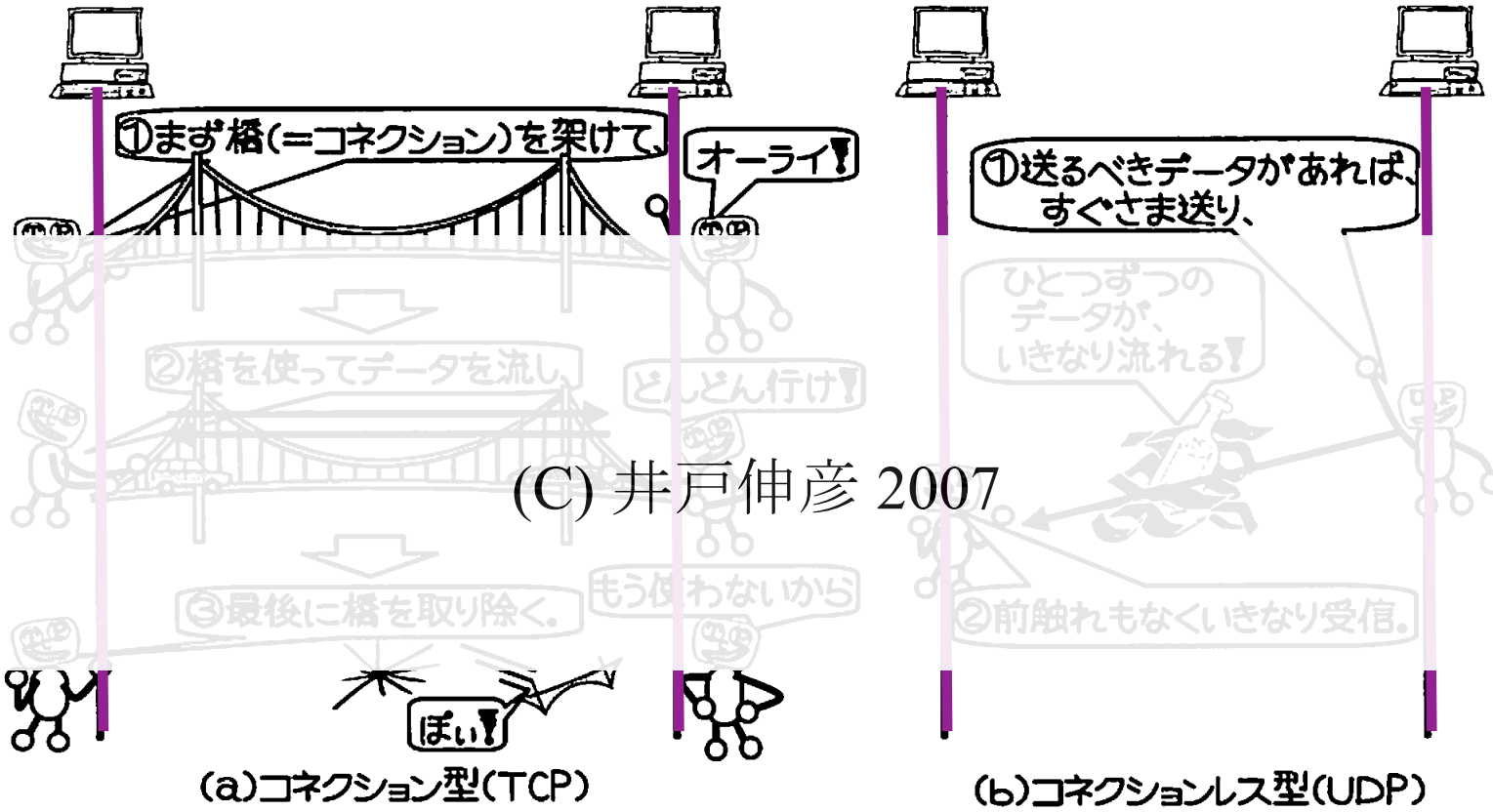


図7-6 コネクション型とコネクションレス型

TCP と UDP のヘッダ・フォーマット

- TCP

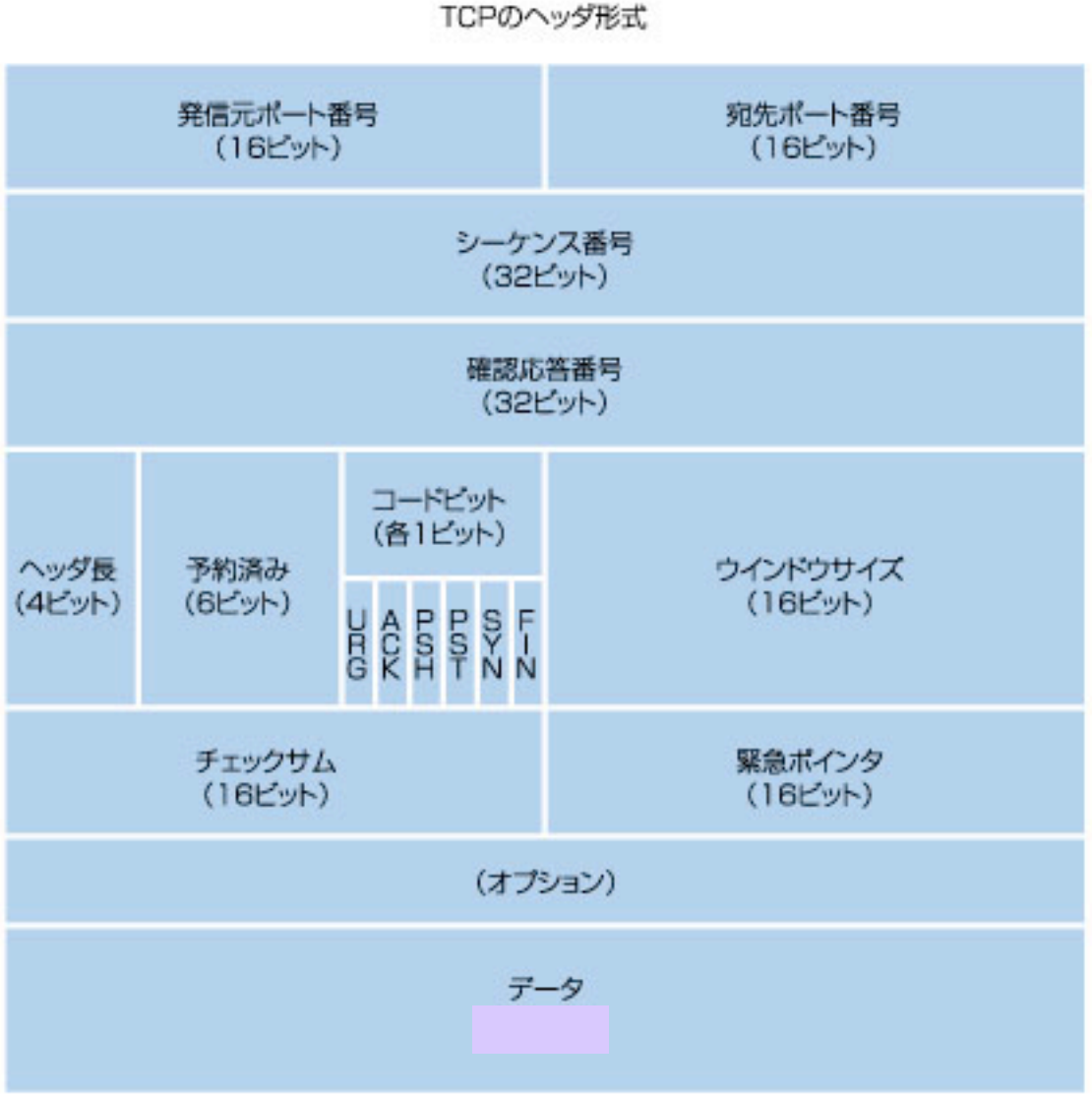
- UDP

図7・7 TCPヘッダ,UDPヘッダ

パケット・フォーマット

TCP と UDP のヘッダ・フォーマット（つづき）

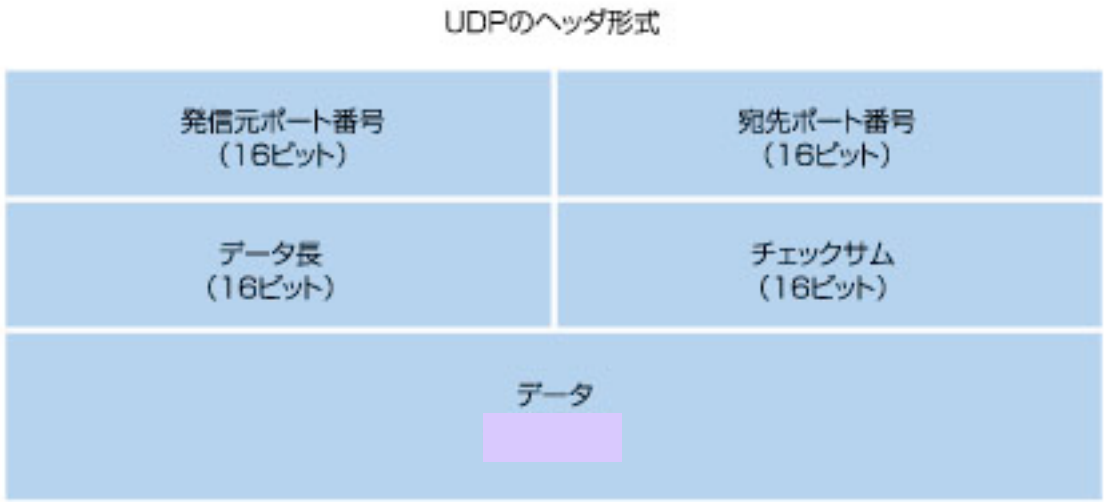
■ TCP の各フィールドの幅を考慮した図



http://www.blwisdom.com/word/key/000623.html

TCP と UDP のヘッダ・フォーマット（つづき）

■ UDP の各フィールドの幅を考慮した図



TCP と UDP のヘッダ・フォーマット（つづき）

■ IETF の標準ドキュメントとパケット・フォーマットの表記

- ◆ TCP は RFC 793, UDP は RFC 768 という IETF の標準ドキュメントで規定されている.
- ◆ RFC では伝統的に図も文字でかく.

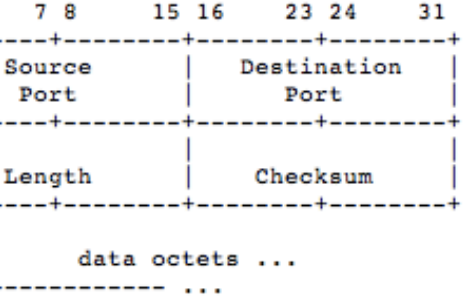
RFC 768 J. Postel, ISI, 28 August 1980

User Datagram Protocol

Introduction

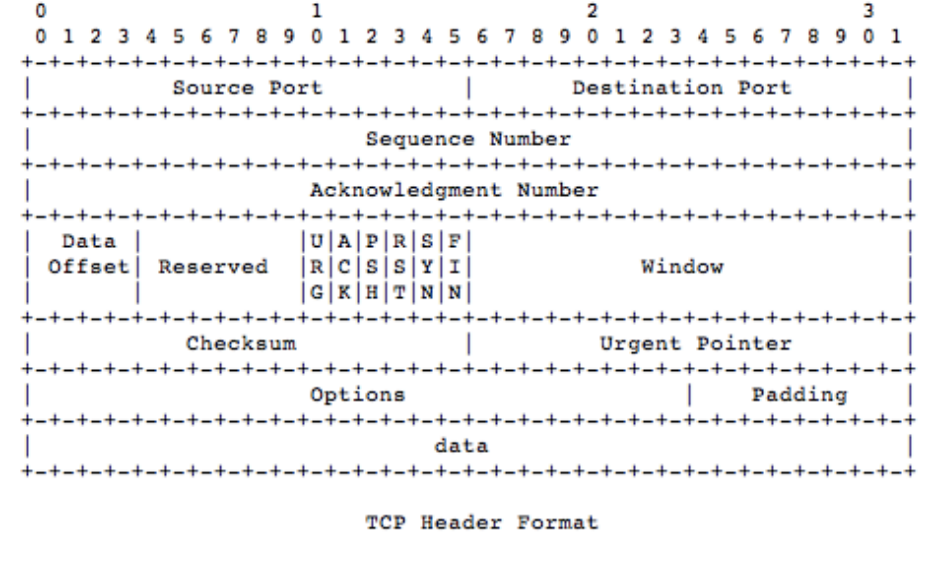
This User Datagram Protocol (UDP) is defined to make available a datagram mode of packet-switched computer communication in the environment of an interconnected set of computer networks. This protocol assumes that the Internet Protocol (IP) [1] is used as the ...otocol.

... provides a procedure for application programs to send ...other programs with a minimum of protocol mechanism. The ...transaction oriented, and delivery and duplicate protection ...nteed. Applications requiring ordered reliable delivery of ...ta should use the Transmission Control Protocol (TCP) [2].



User Datagram Header Format

TCP Header Format



TCP Header Format

Note that one tick mark represents one bit position.

Figure 3.

ネットワーク上での TCP と UDP の比率

■ 日本では最近 UDP の使用が急速にふえている (らしい)

◆ IIJ の総トラフィックにしめる UDP の割合は

2009 年　2.2%

2010 年　6.8%

2011 年　10.0%

(IIJ Internet Infrastructure Review vo. 8, vol. 12).

のこりはほとんど TCP.

◆ 以前はほとんどが TCP だったことがわかる.

TCP の特徴

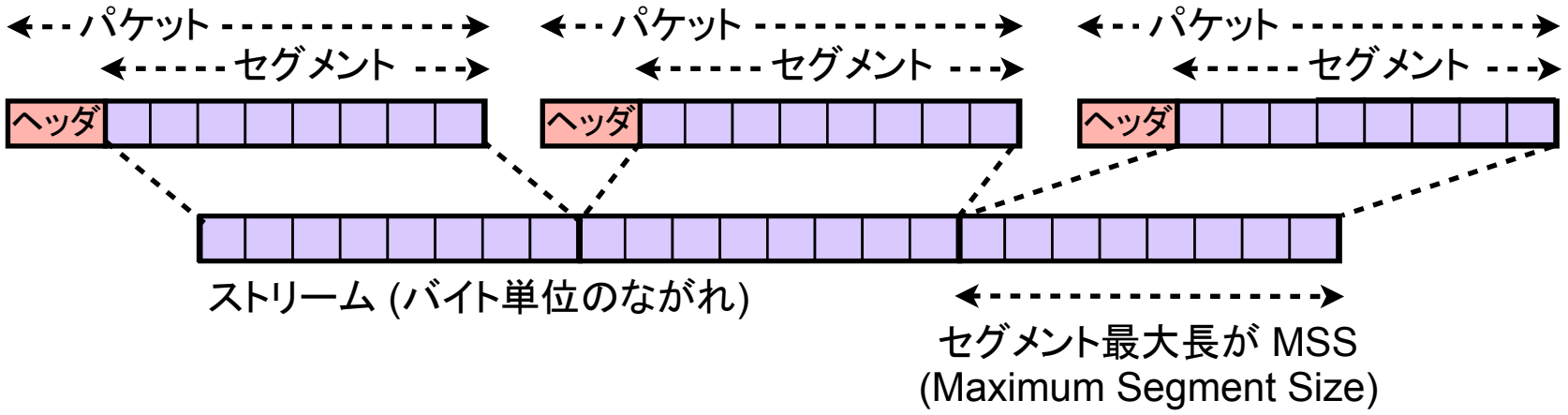
■ 全二重のコネクション型 (connection-oriented) の通信

- ◆ 電話のように接続してから通信する.
- ◆ 電話のように双方向同時に通信できる (全二重通信する).
  - ●双方向でも同時にできないトランシーバのような方式を半二重という.
    (たとえば, イーサネットでは全二重, 半二重のどちらも選択できる.)

全二重と半二重

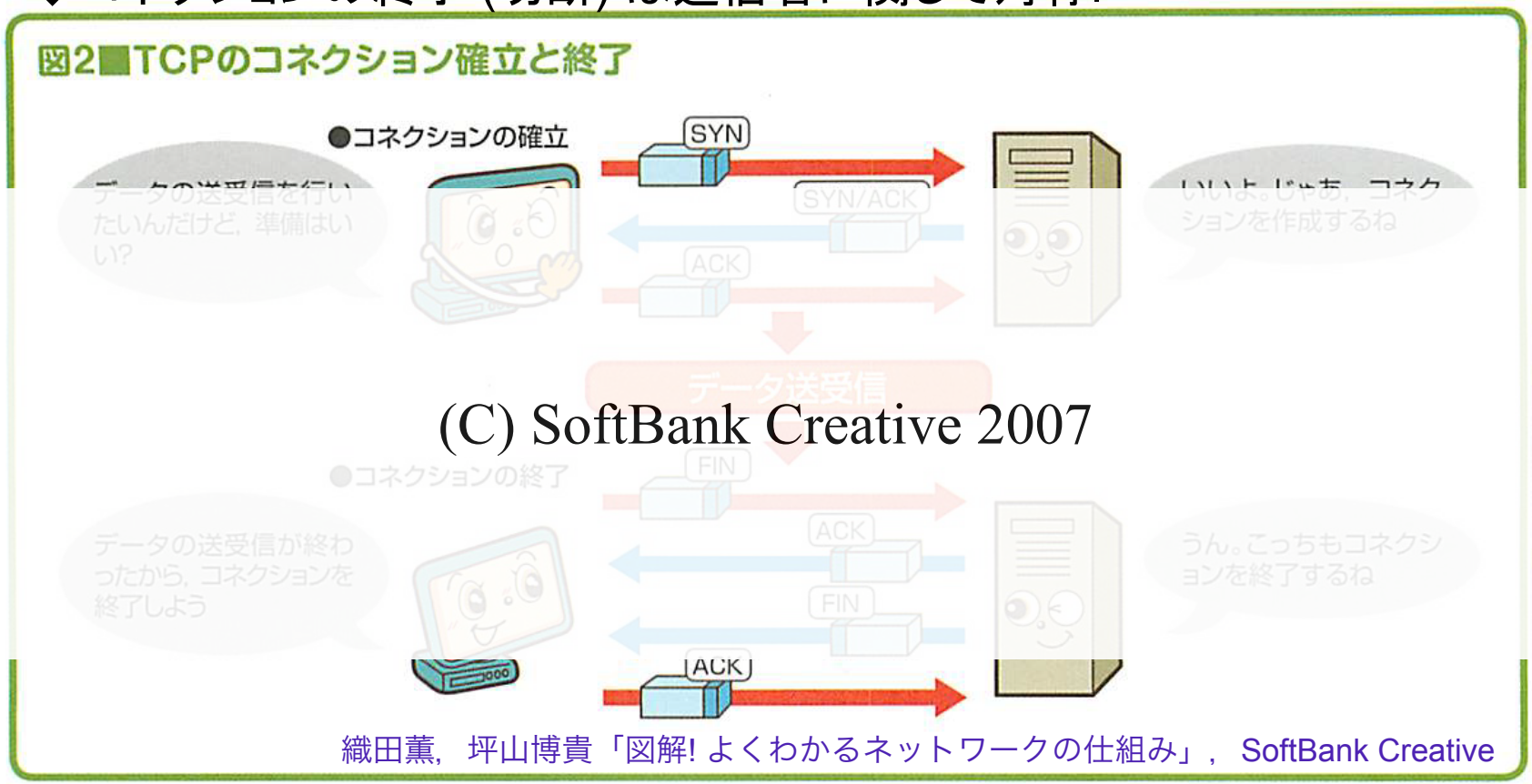
■ ストリーム型の通信 (stream-oriented)

- ◆ バイト単位のストリング (文字列) を通信する.
- ◆ セグメント (パケット) はデータ内容においては意味をもたない.



TCP のコネクション確立と終了

- コネクションの確立には 3 つのメッセージを使用する
  (3 ウェイ・ハンドシェイク)
- 通信の対称性
  - コネクションの確立は通信者に関して非対称.
  - コネクションの終了 (切断) は通信者に関して対称.



(C) SoftBank Creative 2007

織田薫, 坪山博貴「図解! よくわかるネットワークの仕組み」, SoftBank Creative

TCP のコネクション確立と終了（つづき）

■ 3 ウェイ・ハンドシェイクが必要な理由

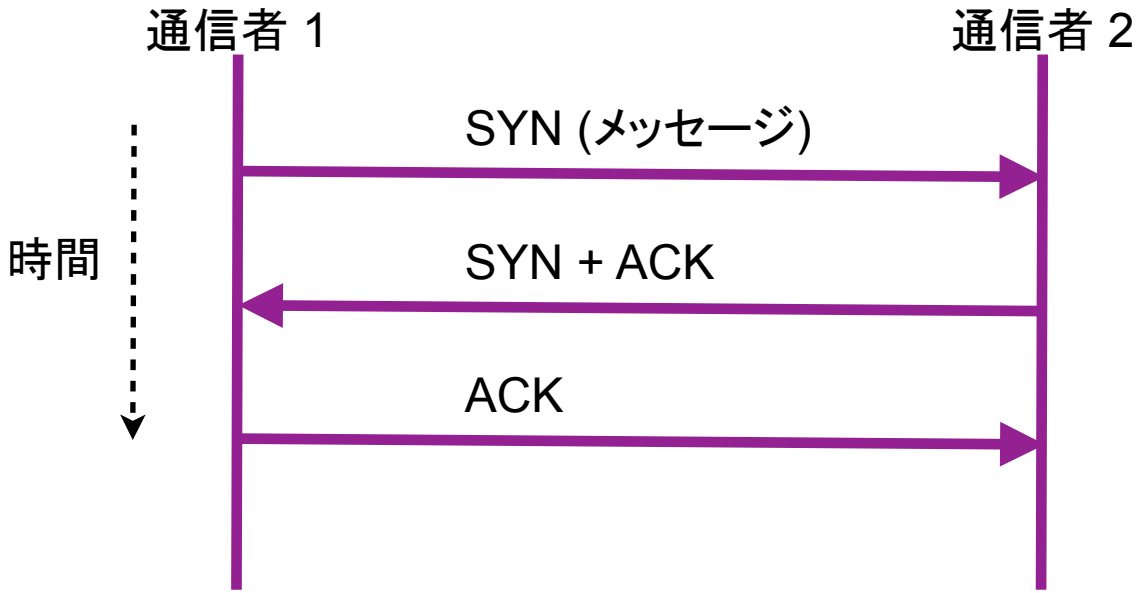
- ◆ 2 ウェイでは，メッセージがうまくとどかなかったときに，SYN の送信者 (A) が，受信者 (B) がコネクションを OK したのかどうかを確認できない．
- ◆ メッセージがうまくとどかないケースにはいろいろあるが，どのケースでも 3 ウェイ・ハンドシェイクならうまくいく．
  - ●SYN がとどかない / 到着がおくれる．
  - ●ACK+SYN がとどかない / 到着がおくれる．
  - ●ACK がとどかない / 到着がおくれる．

(a)2-WAY(2段階)の場合の手順　(b)SYNが彷徨って無意味になってから到着した場合

図7-22 2-way(2段階)では駄目な理由(3-wayが必要な理由)

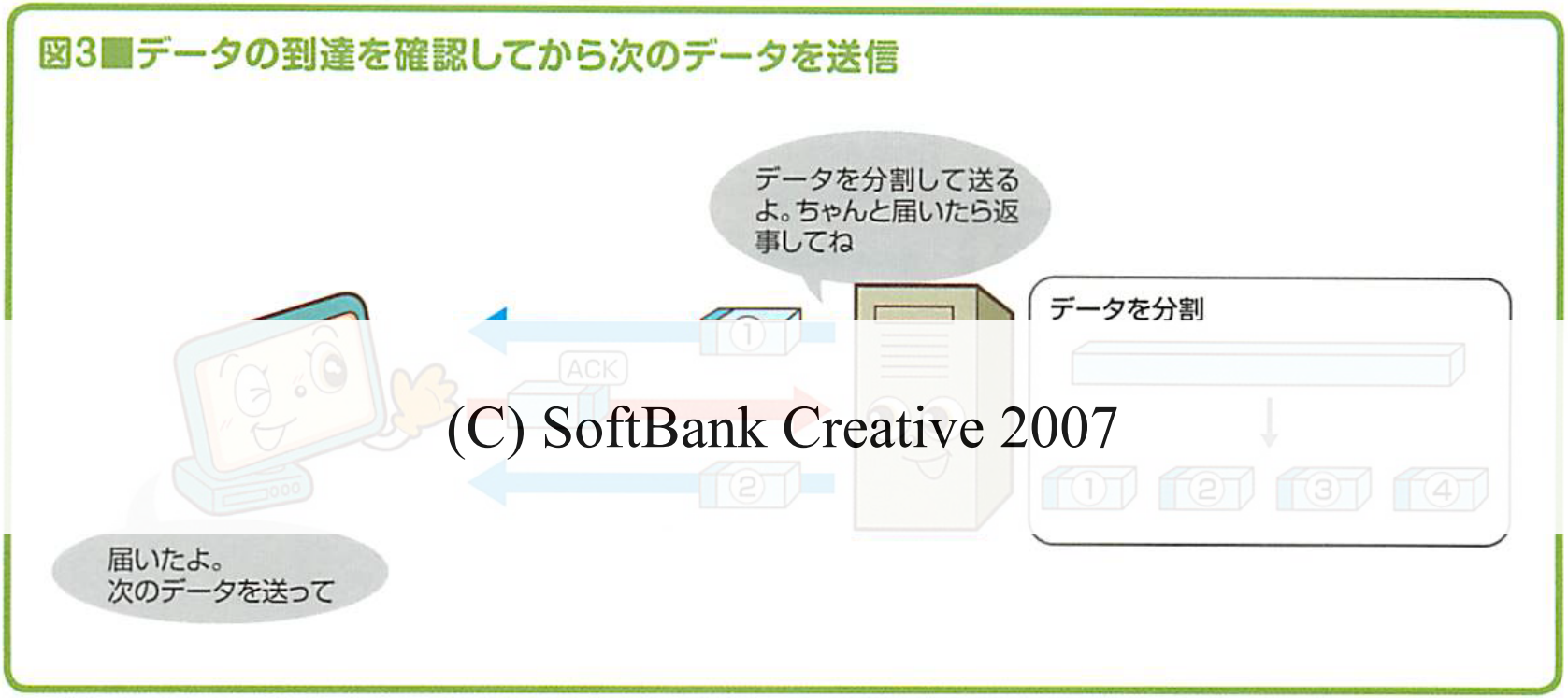
シーケンス図

- 通信のながれを時系列的に図示する.
- UML で定義されている.
  - UML = unified modeling language (統一モデリング言語).
    ソフトウェアの仕様を記述するための標準化された言語.
    OMG によって標準化されている.
  - このスライドのシーケンス図はかならずしも標準に準拠していない.



TCP によるデータ送受信の手順

- データがとどいたかどうかを送達確認 (ACK) する.
- 単純な方法は ACK がとどいてからつぎのデータをおくる方法だが, これでは性能がでない.



織田薫, 坪山博貴「図解! よくわかるネットワークの仕組み」, SoftBank Creative

TCP 高信頼化のためのしくみ: 送達確認

■ TCP ではデータが紛失しても再送される.

◆ パケットにシーケンス番号をつけておき, とどかなかったパケットを識別 (送達確認) して TCP が再送する.

◆ UDP では再送のしくみがないので, 必要ならアプリケーションが紛失を検出して再送する必要がある.

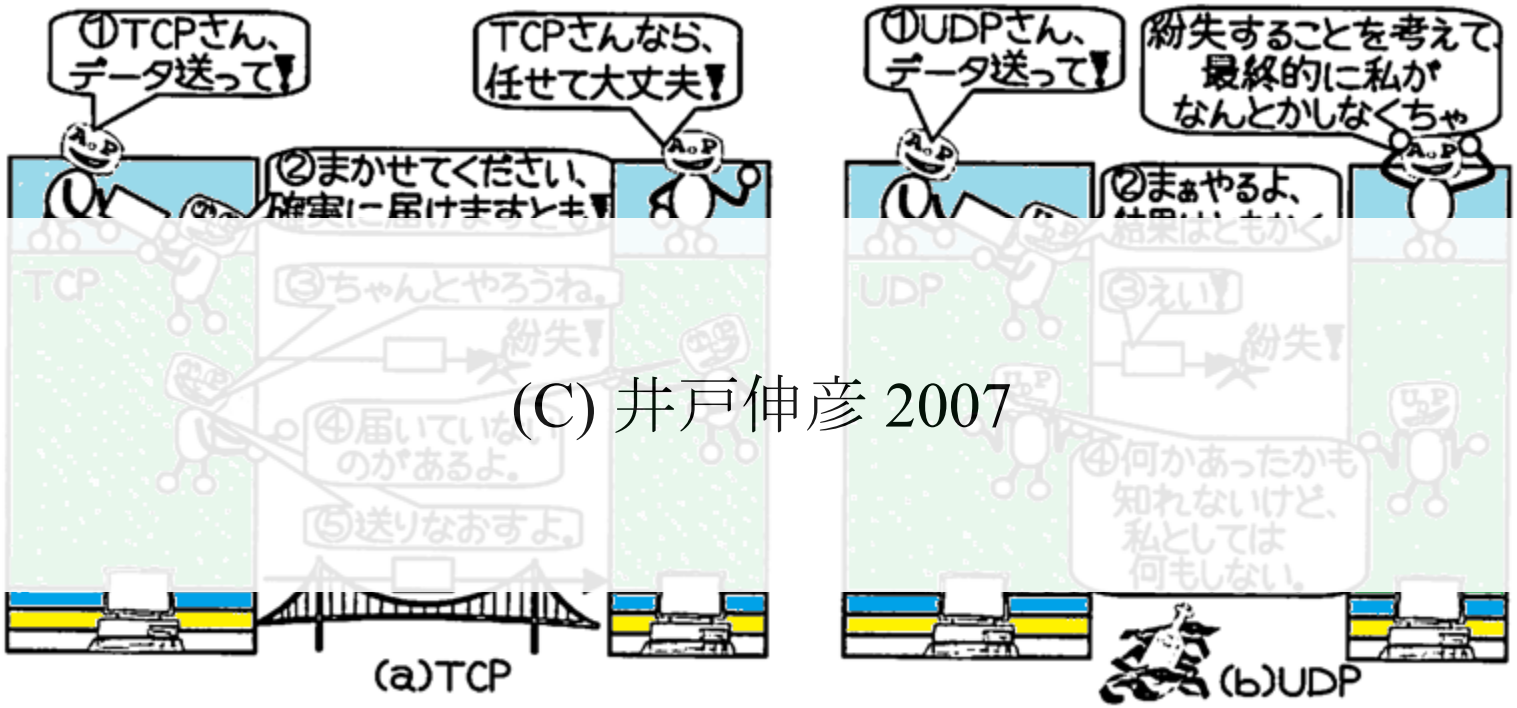


図7・13 データの紛失に対するTCPとUDPの違い

TCP 高信頼化のためのしくみ: 送達確認 (つづき)

■ 送達確認のため，TCP ヘッダ内のつぎのフィールドがつかわれる.

- ◆ シーケンス番号
- ◆ 確認応答番号
- ◆ ACK フラグ

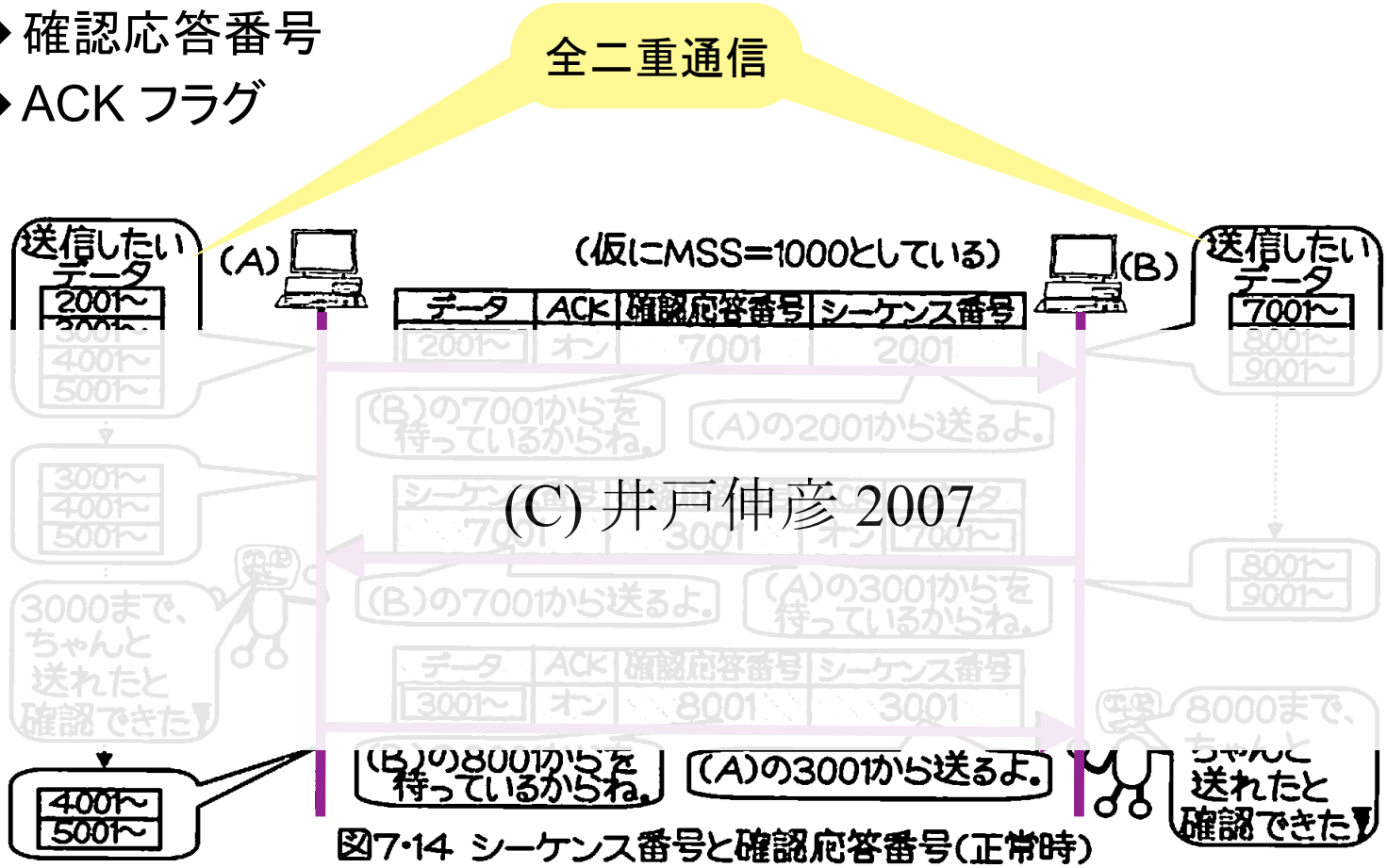


図7-14 シーケンス番号と確認応答番号(正常時)

TCP 高信頼化のためのしくみ: パケットの再送

■ 送信側でタイマーによる再送制御をおこなう.

◆ とどくはずのパケットがとどかないとき (ACK がとどかないとき) は, 再送する.

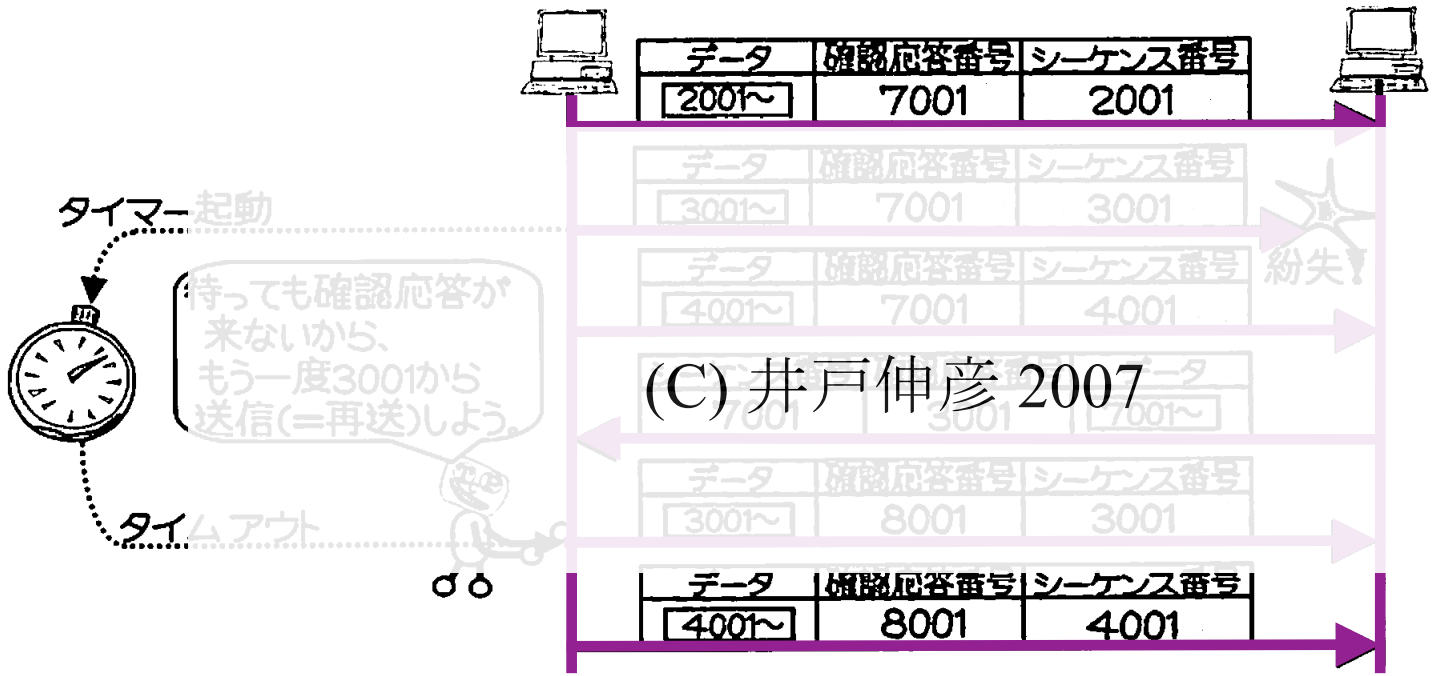


図7·15 セグメント紛失時の再送

TCP 高信頼化のためのしくみ: 転送制御

■ 受信側でシーケンス番号をみて，つぎの処理をする.

- ◆ パケットのならべかえ: シーケンス番号の順にならべる.
- ◆ 重複除去: 同一シーケンス番号のパケットが複数個とどいたら，ひとつだけにする.

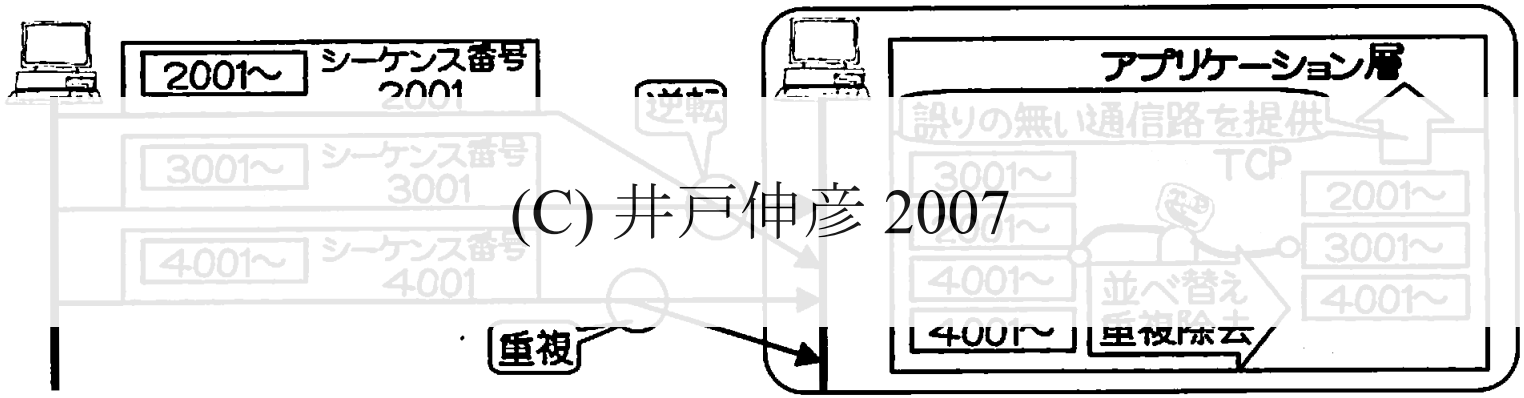
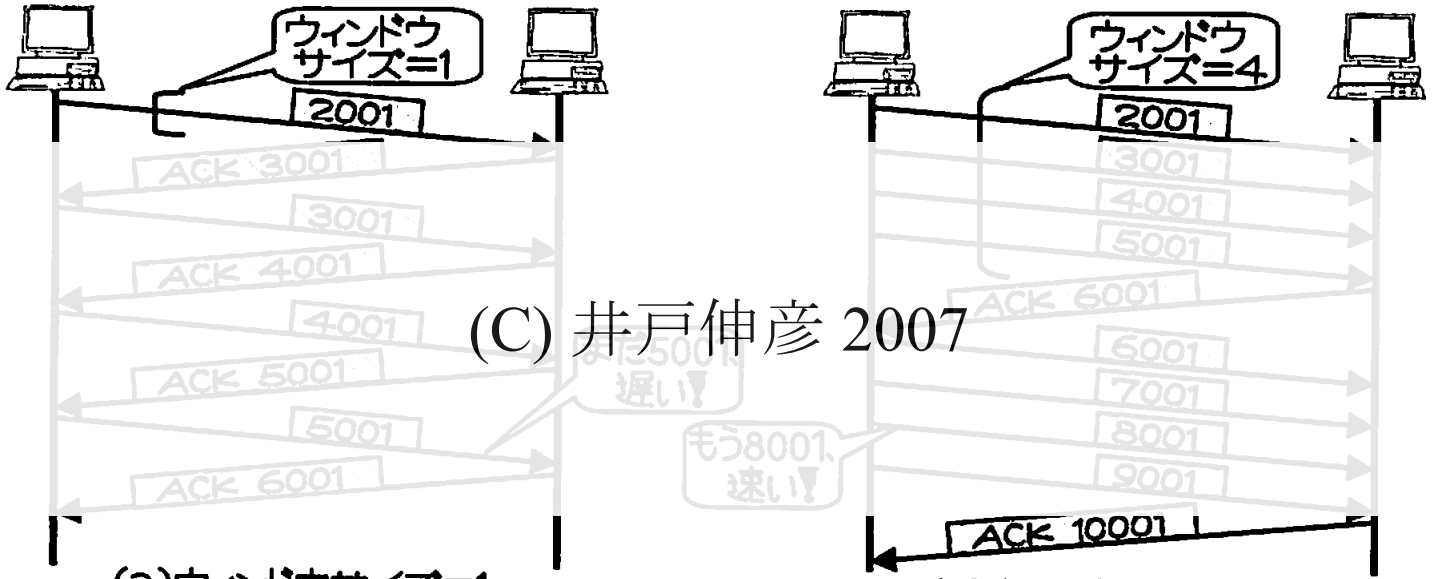


図7・16 並べ替え、重複除去

TCP 高性能化のためのしくみ: ウィンドウ制御

■ 送達確認 (ACK) をパケットごとでなく，まとめておこなうことで高性能化をはかっている.

◆ パケットごとに確認すると，遠距離通信では通信速度が激減する.

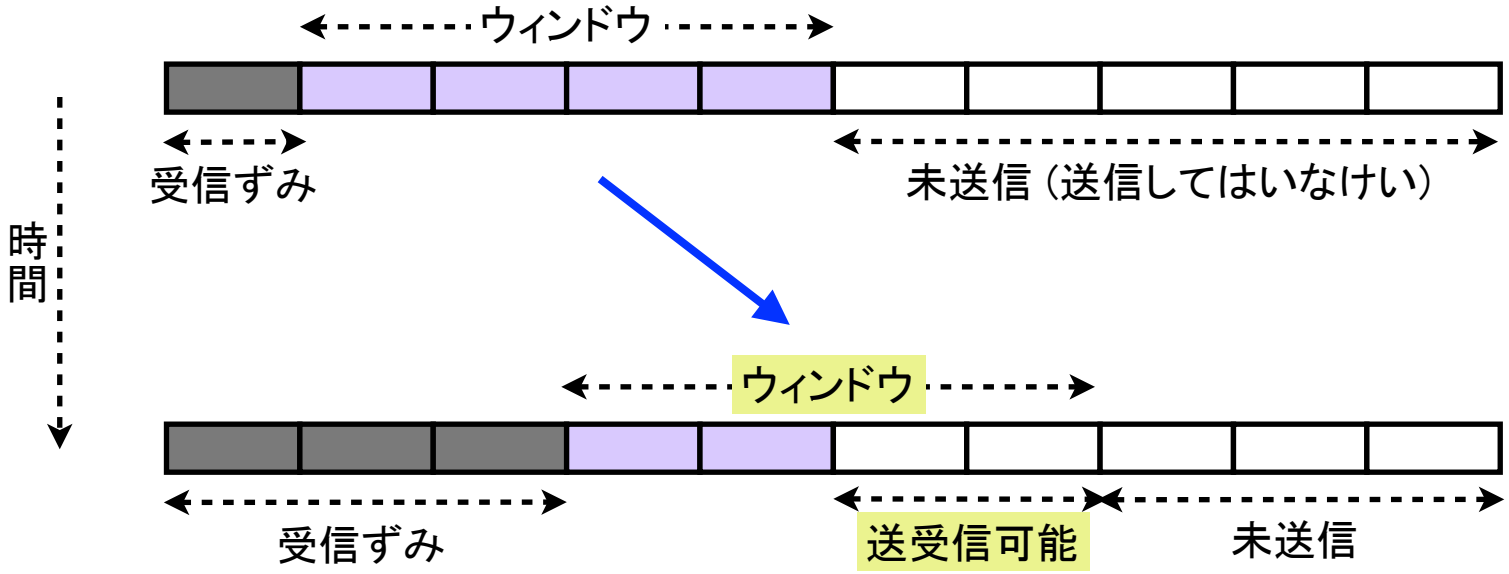


※ ウィンドウサイズ：ここでは「ACKを受け取る前に送信してよいセグメントの数」とする

図7-17 ウィンドウサイズによる転送速度の違い(動作例)

TCP 高性能化のためのしくみ: ウィンドウ制御 (つづき)

■ データ受信が確認されるごとに，送信側でウィンドウをずらす．



TCP 高性能化のためのしくみ: ウィンドウ制御 (つづき)

■ MSS (最大セグメント長) と RWIN (受信ウィンドウ・サイズ) をつかった制御

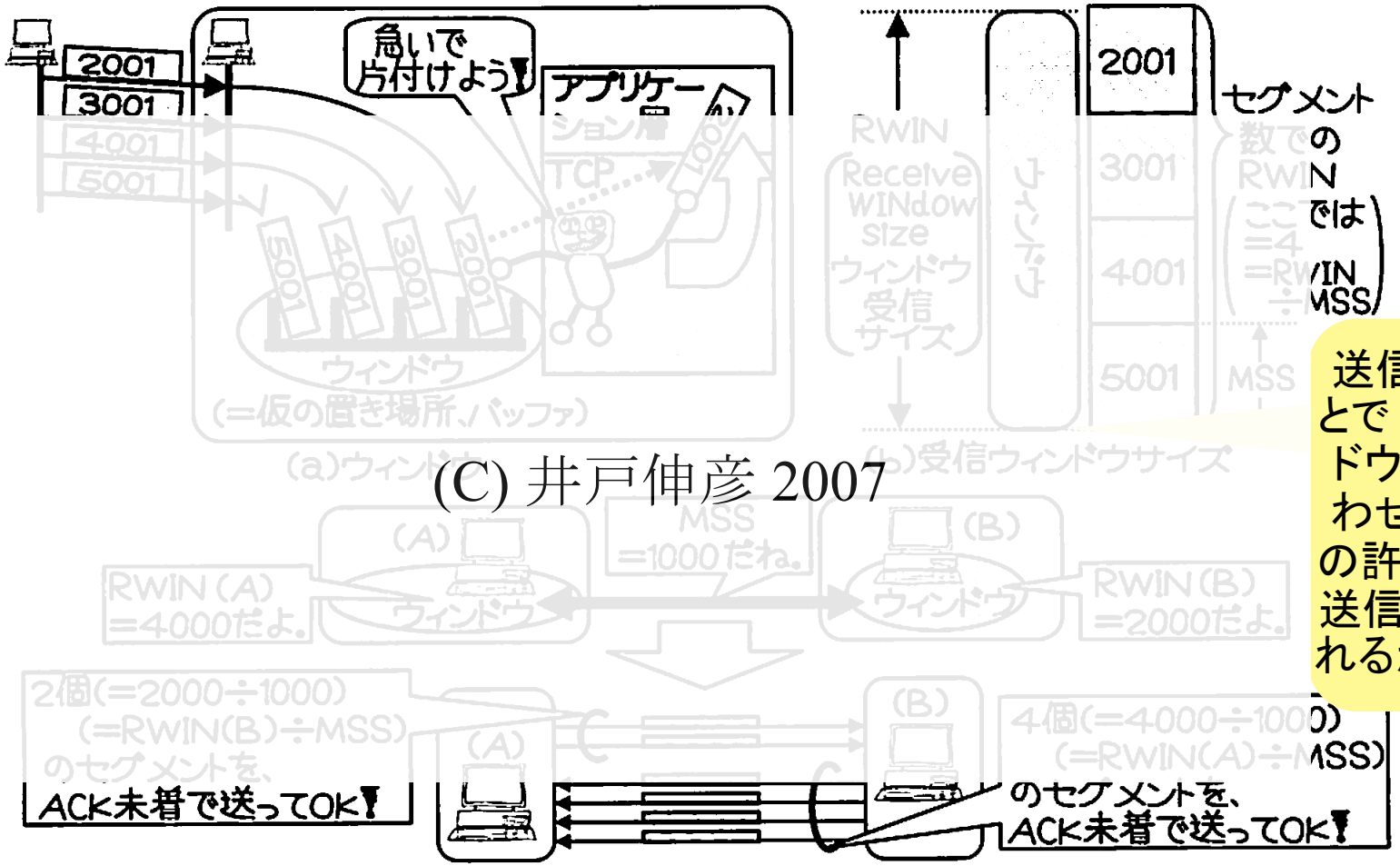


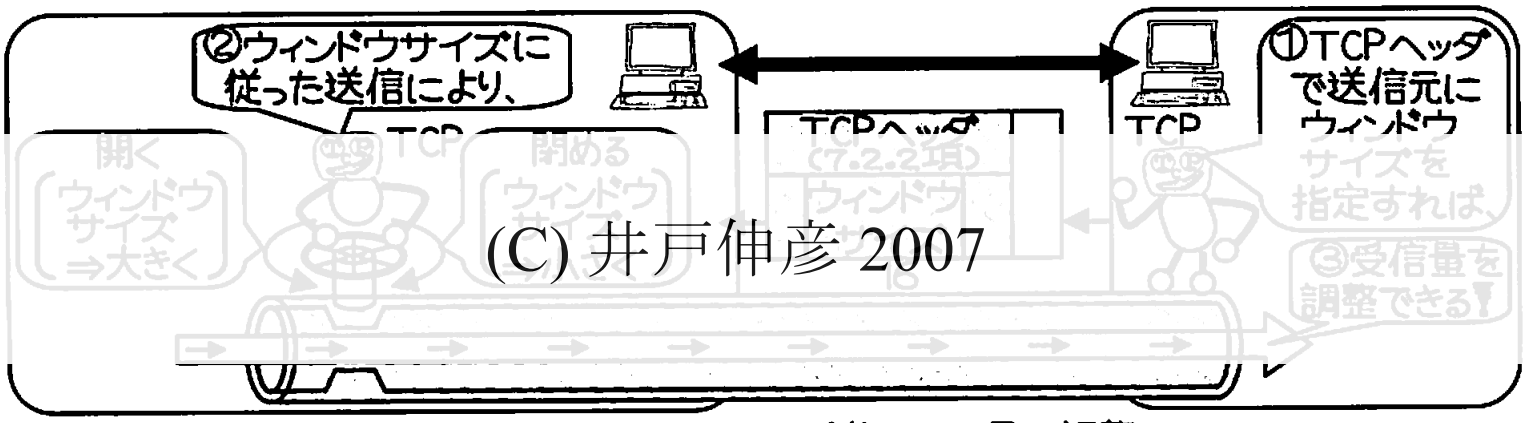
図7・18 RWINとMSS

送信側と受信側とで MSS とウィンドウ・サイズをあわせる. (受信側の許容量をこえて送信すると廃棄されるかもしれない)

ウィンドウを利用した制御: フロー制御・輻輳制御

■ TCP では送信側で通信状態を把握して，ウィンドウ制御によって通信量を加減する.

- ◆ いろいろな制御に利用できる.
- ◆ 受信側でウィンドウ・サイズを指定することもできる.

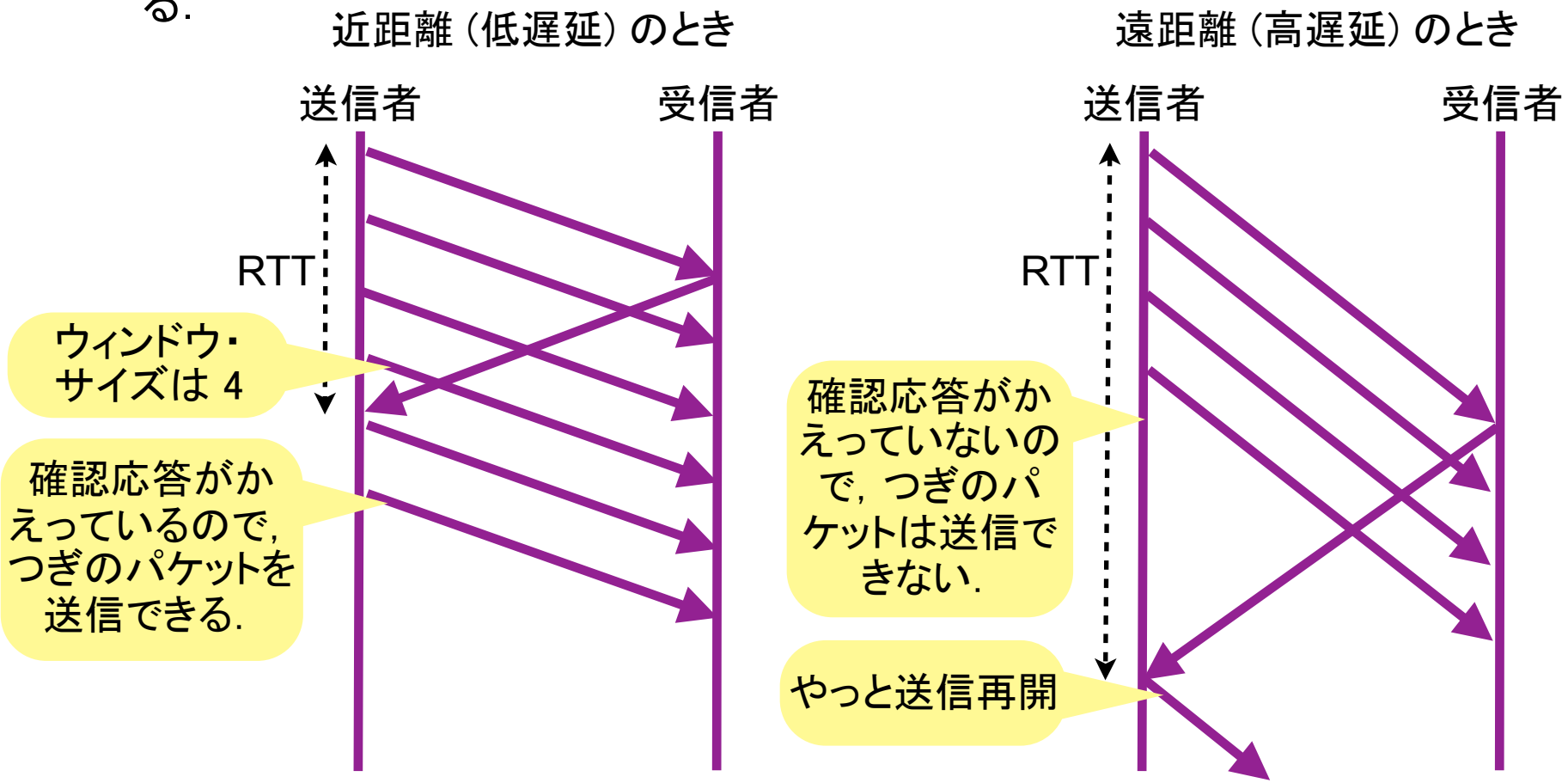


(a)ウィンドウサイズによる受信データ量の調整

ウィンドウを利用した制御: フロー制御・輻輳制御 (つづき)

■ TCP のウィンドウ・サイズと通信量との関係

- ◆ パケットの往復時間を RTT 秒とする. -- 送受信者間を往復する時間
- ◆ ウィンドウ・サイズを w とすると, 1 個のパケットの送信に RTT / w 秒かかる.



ウィンドウを利用した制御: フロー制御・輻輳制御 (つづき)

- フロー制御: TCP では受信側で処理がまにあわなくなりそうなときは, ウィンドウをちぢめる.
- 輻輳制御: TCP ではネットワークが混んでいること (輻輳) を検知するとウィンドウをちぢめて通信量をへらす.
  - ◆どうやって輻輳を知るのか?
    - ●パケットの廃棄を輻輳の兆候とみなす.
  - ◆なぜ輻輳するとウィンドウをちぢめるのか?
    - ●ウィンドウをちぢめると輻輳が軽減されるとかんがえられるから.
    - ●他の通信との共存をめざしている.
  - ◆輻輳制御がうまく動作するとはかぎらない (遠慮しすぎることがある).

(b)フロー制御　(c)輻輳(ふくそう)制御

図7-19 ウィンドウサイズによるフロー制御・輻輳(ふくそう)制御

ウィンドウを利用した制御: フロー制御・輻輳制御 (つづき)

■ TCP の輻輳制御がうまくいくときと，うまくいかないとき

◆ [うまくいく例] 輻輳点における通信がすべて TCP なら，うまくいく.

●輻輳するとみんながウィンドウをちぢめて通信量をへらすから.

◆ [うまくいかない例] UDP があるとうまくいかない.

●UDP 通信では輻輳を検知しないので，UDP が勝つ (通信帯域を独占するかもしれない).

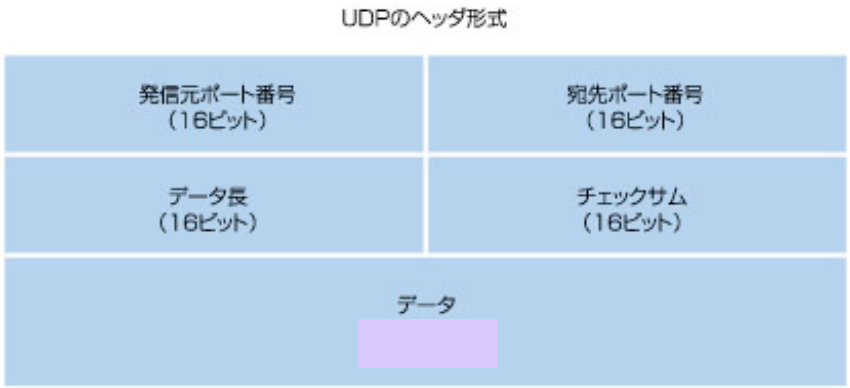
■ TCP 親和性 (TCP friendliness)

◆ TCP 以外のプロトコルでも輻輳時に通信量をへらして TCP と帯域をわけあうこと.

UDP の機能

■ UDP は IP の機能に送受信ポートの指定と誤り検出 (チェックサム) の機能だけをつけくわえる.

- ◆ TCP のような高信頼化や高性能化のための機能はない.
- ◆ UDP の標準ドキュメント RFC 768 は 3 ページしかない.



http://www.blwisdom.com/word/key/000625.html

RFC 768 J. Postel
ISI
28 August 1980

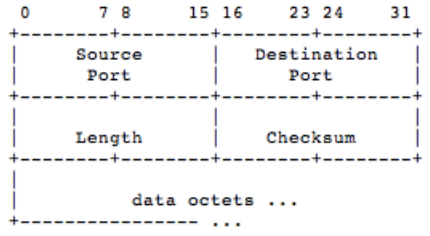
User Datagram Protocol

Introduction

This User Datagram Protocol (UDP) is defined to make available a datagram mode of packet-switched computer communication in the environment of an interconnected set of computer networks. This protocol assumes that the Internet Protocol (IP) [1] is used as the underlying protocol.

This protocol provides a procedure for application programs to send messages to other programs with a minimum of protocol mechanism. The protocol is transaction oriented, and delivery and duplicate protection are not guaranteed. Applications requiring ordered reliable delivery of streams of data should use the Transmission Control Protocol (TCP) [2].

Format



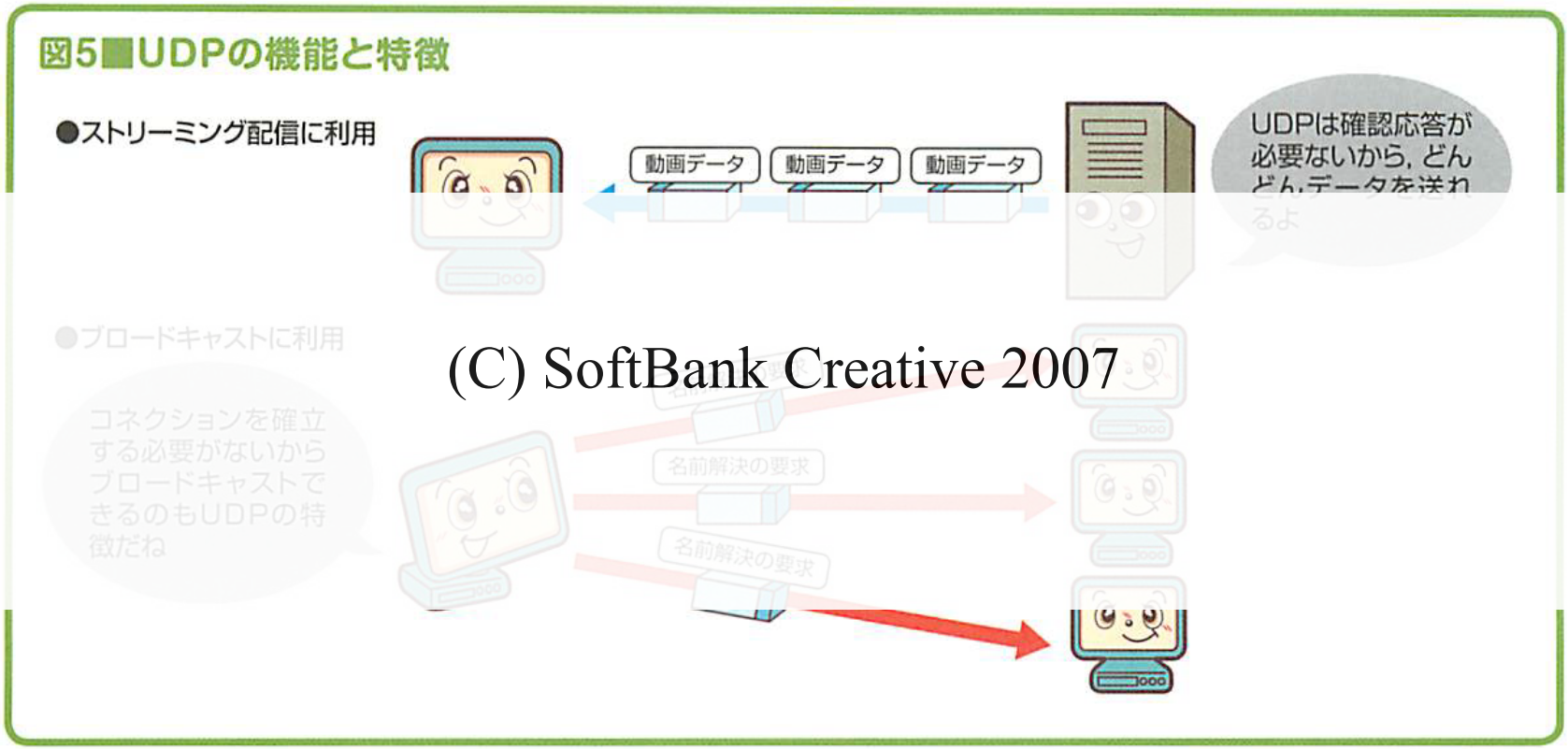
User Datagram Header Format

UDP の特徴

■ リアルタイムの通信に適している.

◆ 高信頼性より低遅延であることが重要な通信に向いている.

■ TCP ではできない 1 対多 や 多対多 の通信ができる.



織田薫, 坪山博貴「図解! よくわかるネットワークの仕組み」, SoftBank Creative

TCP と UDP

■ ファイルを転送するときは，パケットが途中で廃棄されても自動的に回復してくれる TCP のほうが便利である.

◆ UDP をつかうと，アプリケーションがパケットの再送を制御する必要が生じる.

図7·25 UDPで大きなファイルを転送する

TCP と UDP

■ ちいさなメッセージをおくるには UDP のほうが効率がよい.

- ◆ TCP では接続の負荷 (時間, 転送量) がおおきい.
- ◆ TCP ではヘッダがおおきいため, データが少量だと効率がわるい (転送量).

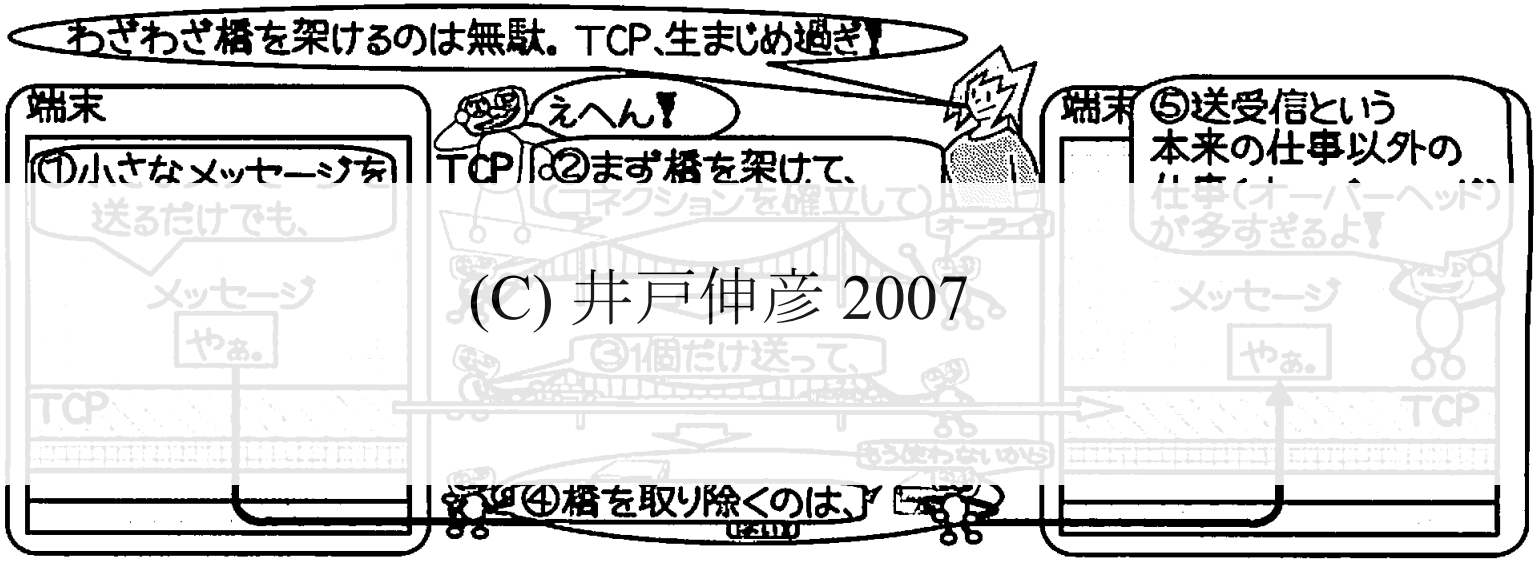


図7・26 TCPで小さなメッセージを送る

TCP と UDP

■ リアルタイムであることが重要なときは，TCP のパケット再送はじゃまになる.

◆ データがそろうまでアプリケーションにわたされないから.

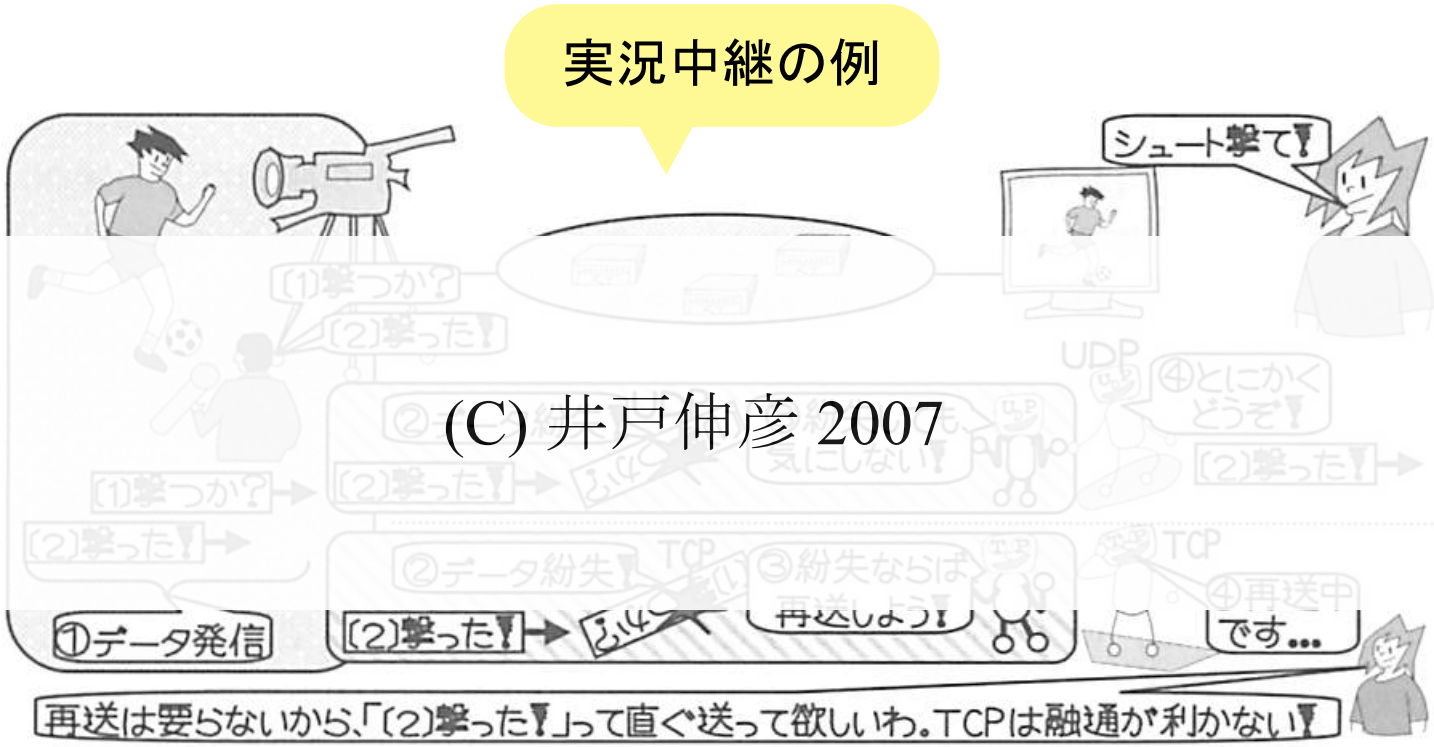


図7･27 TCPで実況中継のデータ(リアルタイムデータ)を送る

TCP と UDP

■ ブロードキャストしたいときは，TCP ではできないので UDP をつかう.

◆ TCP のコネクションは 1 対 1 にかぎられているので，ブロードキャストはできない.

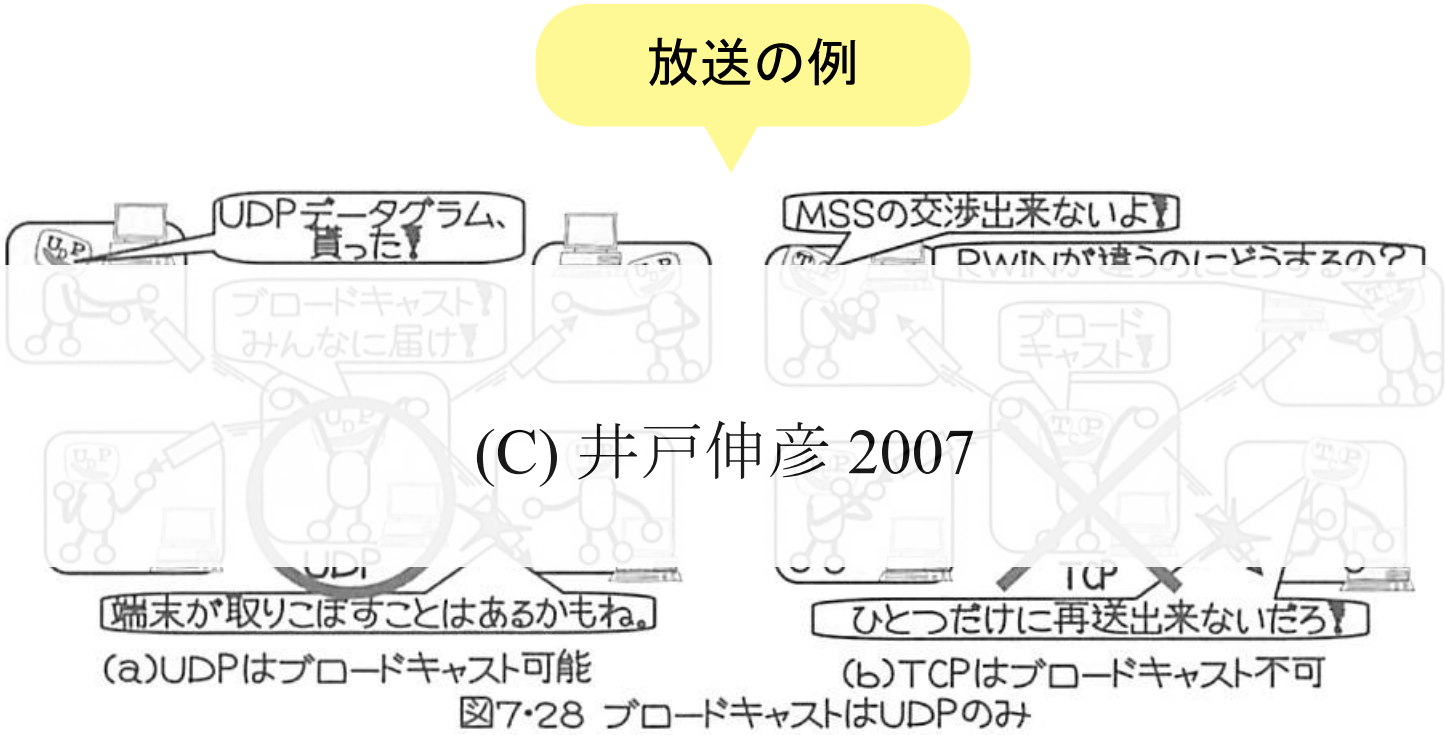


図7･28 ブロードキャストはUDPのみ

TCP と UDP

■ まとめ

表7-1 TCPとUDPの比較

		TCP (Transmission Control Protocol)	UDP (User Datagram Protocol)
コネクション		コネクション型 (コネクションを確立してから送受信)	コネクションレス型 (いきなり送受信)
あだ名	ほめ言葉	多機能、正確、几帳面、まじめ	シンプル、軽快、軽いフットワーク
	悪口	大袈裟、生まじめ、馬鹿丁寧、	機能なし、いい加減、お調子者
ヘッダの大きさ		大きい(20バイト)	小さい(8バイト)
上位とのデータの受け渡し		ストリーム(切れ目の無い流れ)	データの固まり(データグラム)
データの分割		あり(セグメンテーション分割)	なし(必要ならばIPフラグメンテーションに任せる)
信頼性		あり(シーケンス番号による送達確認あり)	なし
効率	大きなファイル	良い(セグメントの紛失を再送で防ぐ)	悪い(データグラムが失われれば失敗)
	小さなメッセージ	悪い(オーバーヘッドが大きい)	良い
リアルタイムデータ、ブロードキャスト		不得意	得意

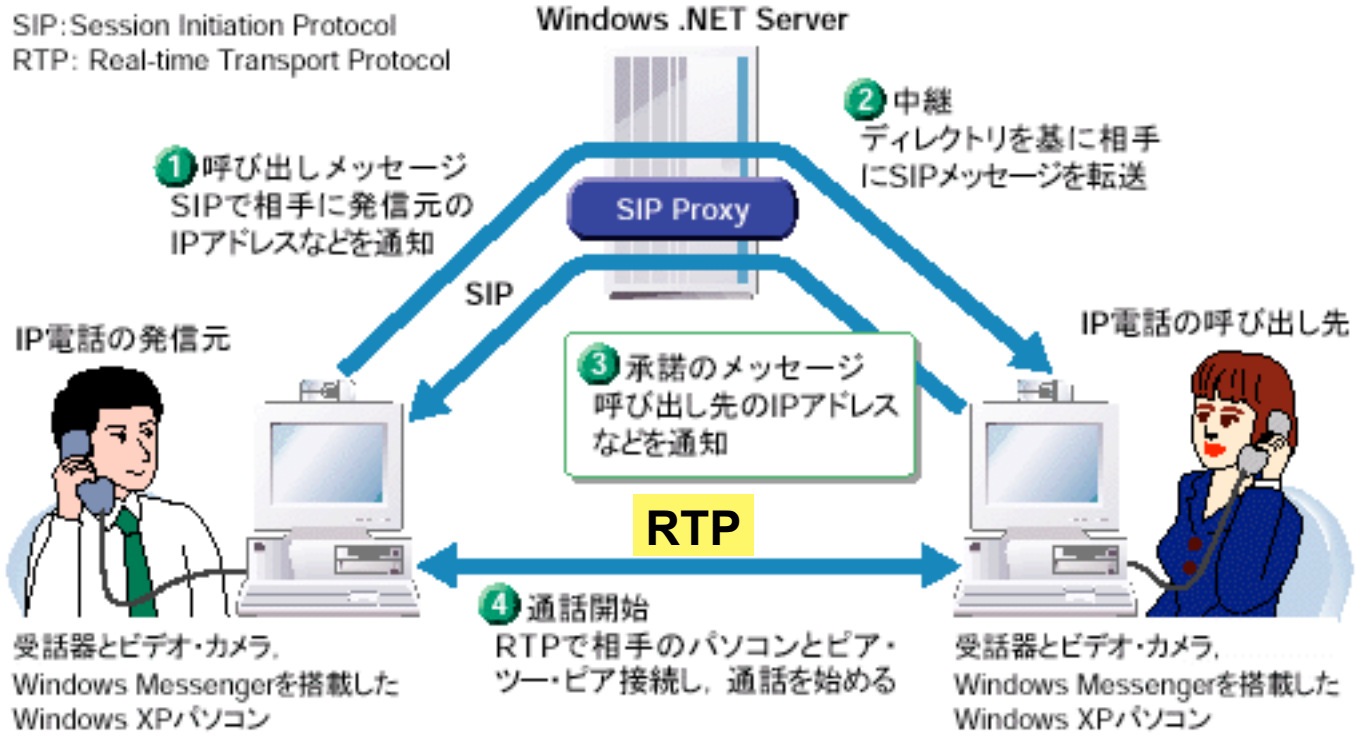
TCP, UDP 以外のトランスポート・プロトコル

- TCP, UDP 以外のプロトコルの必要性
  - TCP の特徴のうちの一部だけを利用したいことがある
    -- たとえば高信頼性はほしいがストリーム型でないほうがよいとき.
  - UDP を高信頼化するのはプログラミングの負荷がたかい.
- つぎのようなプロトコルがある.
  - 音声, 動画などのリアルタイム伝送のための RTP.
  - バイト単位ではないストリーム伝送のための SCTP.
  - ストリームではない単独のメッセージの高信頼伝送のための DCCP (Datagram Congestion Control Protocol)
- TCP, UDP とくらべると, それ以外のトランスポートの使用はずっとすくない.
  - UDP の半分以下 (2008 年で 2% 以下, 2010 年で 4% 以下 (IIJ))

TCP, UDP 以外のトランスポート・プロトコル（つづき）

■ RTP (Real-time Transport Protocol)

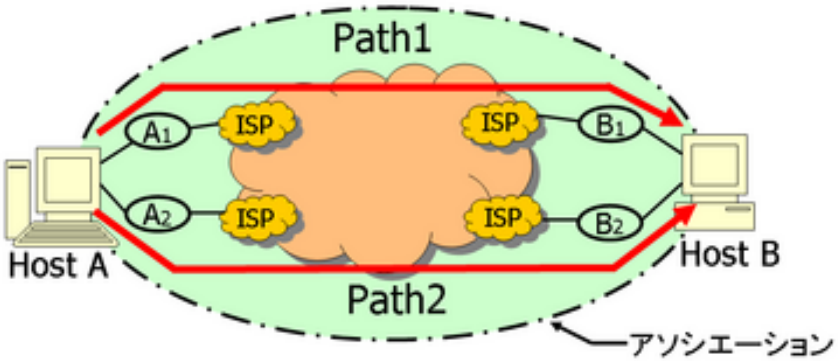
- ◆ 音声，動画などのデータ・ストリームのリアルタイム転送のためのプロトコルである.
- ◆ IP 電話 (光電話など) でつかわれている.
- ◆ 制御のためのプロトコル RTCP (Real-Time Control Protocol) があわせてつかわれる.

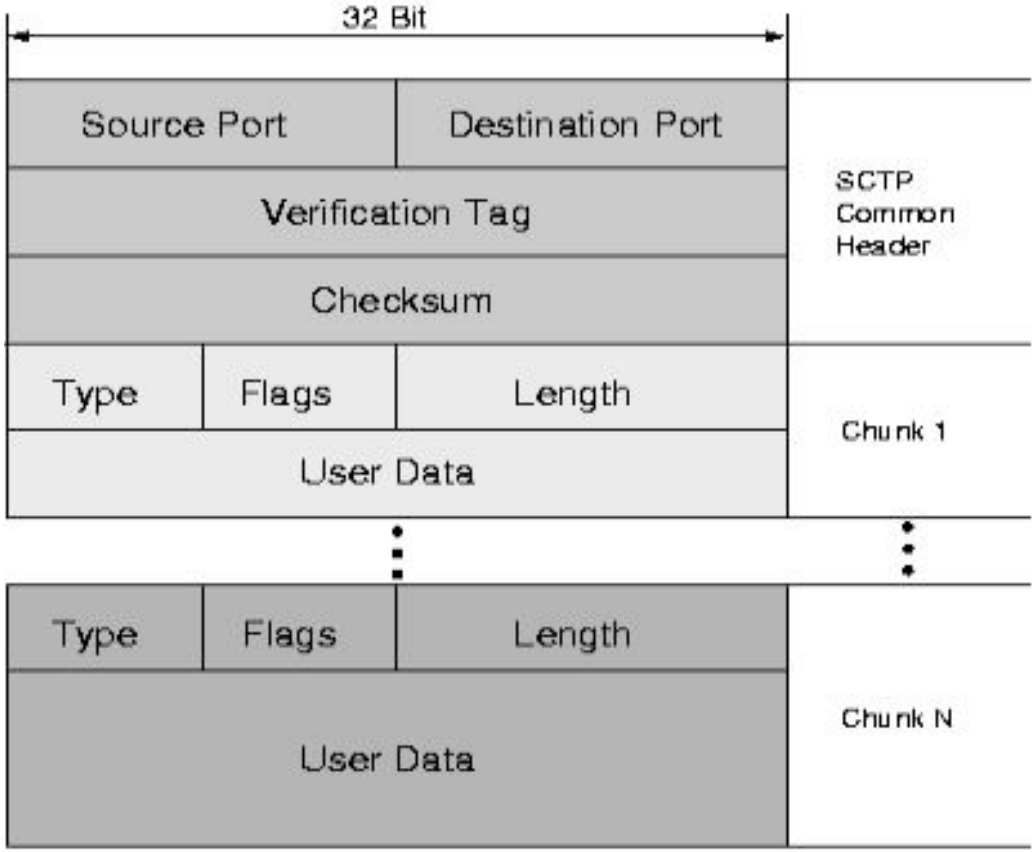


TCP, UDP 以外のトランスポート・プロトコル (つづき)

■ SCTP (Stream Control Transmission Protocol)

- ◆2000 年に IETF で定義された，あたらしいプロトコル.
- ◆つぎの特徴を TCP と共有している: 輻輳制御，到着順序保証 (再送制御)
- ◆TCP とちがって SCTP はフレームの列をあつかう (バイト列ではない).
- ◆TCP にはないいくつかの特徴をもっている.





TCP と UDP のまとめ

- TCP, UDP では「ポート」によって，複数の通信が並行してできるようにしている.
  - 1 個の IP アドレス (1 台のコンピュータ) で複数のポートがつくれる.
- TCP, UDP では「ポート」によって，さまざまなプロトコルをつかいわけることができる.
  - ファイル転送のための FTP, 電子メールのための SMTP, Web のための HTTP などのプロトコルがある.
- TCP では信頼性・性能のたかい通信を実現している.
  - パケットが脱落しても再送する.
  - 通信路を他の通信とうまくわけあって有効につかうしくみがある.
- TCP は複数のパケットにまたがるメッセージを伝送できる.
  - パケットの最大長 (MTU) にしばられない.