

トランスポート層(1)

花田 英輔

1

トランスポート層 (OSIモデル第4層)

- ▶ トランスポート層までは「下位層」と呼ばれる
- ▶ 通信・接続管理よりも通信内容の定義が多い

アプリケーション層	アプリケーション層
プレゼンテーション層	プレゼンテーション層
セッション層	セッション層
トランスポート層	トランスポート層
ネットワーク層	ネットワーク層
データリンク層	データリンク層
物理層	物理層

この話

2

トランスポート層がすること

1. ネットワーク層(第3層)とセッション層(第5層)に対してインターフェースを提供する
2. ネットワーク層で確立された端末間の信頼あるデータ転送(仮想回線)を実現する
3. フロー制御を行う
 - 転送エラーの検出と回復
4. 輻輳(トラフィック)を制御する
5. 重要な伝送プロトコルを定義する
 - TCP、UDP

3

下位層との通信環境の違い

データ・リンク層の環境

ルーター ———— ルーター

1対1で確実な接続

トランスポート層の環境

ネットワーク

- ホストはネットワーク内の様子を知る必要が無い
- 相手を指定して送るのみ

4

相手の指定方法

- ▶ Network Service Access Point
 - ネットワーク層におけるエンドポイント指定
- ▶ Transport Service Access Point
 - トランスポート層におけるエンドポイント指定
- ▶ インターネット(TCP/IP)の場合
 - NSAP: IPアドレス
 - TSAP: ポート(Port)

<http://xxx.yy-u.ac.jp:80>

↑
ポート番号

5

ポートを使用した通信

アプリケーションプロセス

サーバ1

サーバ2

TSAP: 1208

TSAP: 1836

TSAP: 1522

NSAP

NSAP

アプリケーション層

トランスポート層

ネットワーク層

データリンク層

物理層

ホスト1

ホスト2

6

ウェルノウンポート(well-known port)

- ▶ ポート番号の一部(1~1024)は共通的に目的を限定
- ▶ 別の一部(1025~49151)の使用目的はIANAに登録されている

ポート番号	プロトコル	利用目的	ポート番号	プロトコル	利用目的
20, 21	FTP	ファイル転送	161	SNMP	ネットワーク監視
22	SSH	遠隔ログイン(以前はTelnet)	443	HTTPS	Web通信(+SSL/TSL)
25	SMTP	電子メール転送	465	SMTPS	電子メール転送(+SSL/TSL)
53	DNS	DNS	543	RTSP	メディア・プレイヤーコントロール
80	HTTP	Web通信	631	IPP	プリンタ共有
110	POP3	遠隔電子メールアクセス	993	IMAP4s	遠隔電子メールアクセス(+SSL/TSL)
143	IMAP	遠隔電子メールアクセス	995	POP3s	遠隔電子メールアクセス(+SSL/TSL)

7

トランスポート層(2)

花田 英輔

8

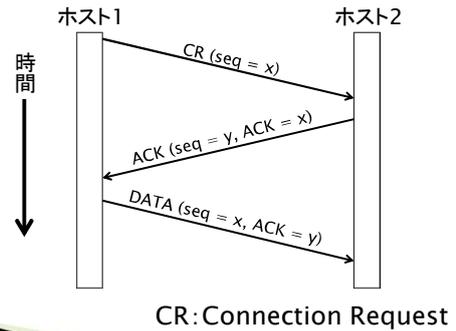
通信(コネクション)の確立と解放

- ▶ パケットは再送される可能性がある
- 送金に関する同じ内容のパケットが2度来ると?
- ▶ 「コネクション」の確立と解放(終了)が重要
 - ホストからネットワーク内の状況は見えないので、遅れてくる通信を排除する必要がある
 - 途中でデータが来なくなったり、途中からデータが再送されてきた場合に通信を終了する必要がある

9

コネクションの確立

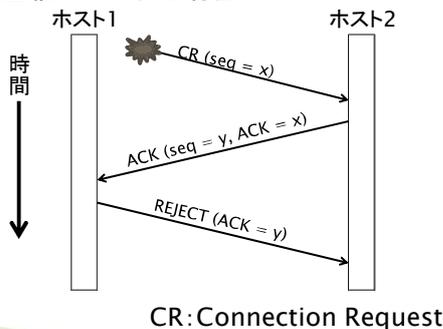
正常な場合



10

コネクションの確立

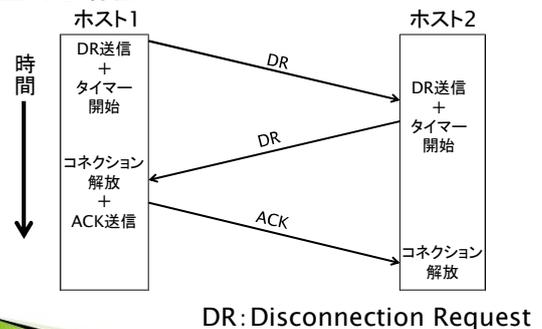
古い重複CRが来た場合



11

コネクションの解放

正常な場合



12

UDPを利用したプロトコルの例(RTP)

- ▶ RTP: Real-time Transport Protocol
 - ネット電話、ビデオ会議、インターネットラジオ、VOD等で利用
 - マルチメディアストリーム向け汎用プロトコル
- ▶ 大きく2つの機能
 - パケット内のメディアデータを送信するプロトコル
 - 受信器での処理

19

RTPの構造

- ▶ アプリケーションはUDP用に用意されたRTPを用いてアプリケーションからデータを送信する
- ▶ RTPヘッダは「順序番号」を持つ

20

RTPの特徴

- ▶ UDPで定義される情報は不要
 - フロー制御やエラー制御、確認応答の機能は無い
 - ・ 二重に定義する必要はない
 - **再送機能は無い**
 - ・ パケットが遅れて来ても、順序通りに再生しないと意味が無い
 - ・ 全部来るのを待つと再生開始が遅れる
- ▶ タイムスタンプ機能が使える
 - 2つ以上のストリームを関連付けて再生可能

21

RTPにおける受信

- ▶ バッファリングとジッター制御
 - 適切な再生速度で再生できる必要がある
 - しかしパケット到着時間は一定ではない
 - ・ 遅延のバラつき = ジッター(Jitter)
 - ・ ジッターがあると、動画がカクカクしたり、音声にノイズが入る
 - バッファリング(Buffering)の活用
 - ・ 一定長の「一時置場(バッファ)」を用意し、貯まったところで再生開始
 - ・ 途中でパケットが遅れた場合の対応が問題

22

バッファリング

▶ バッファリングの例

送信元から出たパケット: 1 2 3 4 5 6 7 8

パケットがバッファに到着した時間

再生ギャップ

時間(秒): 0, 5, 10, 15, 20

- ▶ ギャップの間、音楽やビデオはストップする
 - ・ 動画の場合は最後のフレームを繰り返す
 - ・ 音声の場合は無音にする

23

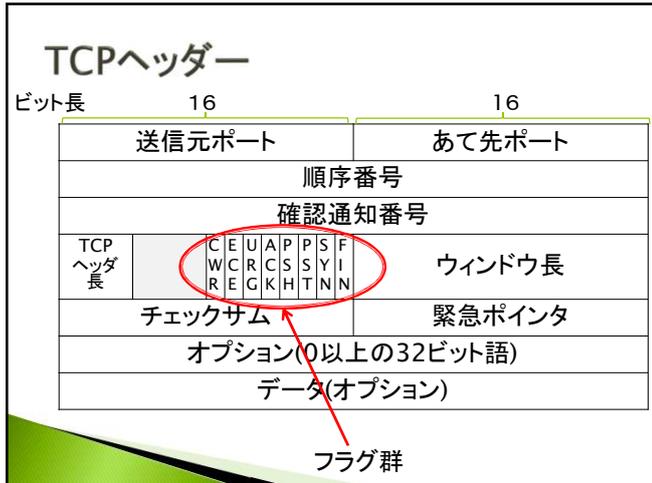
TCP(Transmission Control Protocol)

- ▶ TCP: 伝送制御プロトコル
 - TSAP(ポート)を使用した**コネクションを張っての通信**が原則
 - ポートは理論的には1~65535まで指定可
 - 目的を定めたポート(Well-known Port)有
 - ・ www.iana.orgに掲載

例(再掲)

- 20: FTP
- 22: SSH
- 25: SMTP
- 80: HTTP
- 110: POP3
- 143: IMAP
- 443: HTTPS

24



25

フラグ群の使い方

- ▶ CWRとECE:輻輳の通知(RFC3168使用時)
- ▶ URG:緊急ポインタ使用
- ▶ ACK:確認通知番号が有効か否か
 - 0の場合はセグメントは確認通知を含まないので、確認番号通知は無視される
- ▶ PSH:データがプッシュされたか否か
 - 受け側のバッファ一杯ではなくてもアプリケーションに配送させる
- ▶ RST:コネクションのリセット通知か否か
- ▶ SYN:コネクションの確立に利用
 - コネクション要求時: SYN=1、ACK=0
 - コネクション応答時: SYN=1、ACK=1
- ▶ FIN:コネクションの解放に利用

26

今回の課題

1. TCPとUDPについて説明すると共に、それぞれ代表的な使用例を挙げよ
 2. 動画や音楽の配信を視聴中にデータが「Loading」のまま途切れる原因を説明せよ
 3. 本日の感想
- ▶ 締切: 12月25日(月) 18:00
 - ▶ 本講義に関する情報は(この講義資料も)次のWebpageに掲載するので、時々参照すること
<http://www.ai.is.saga-u.ac.jp/~hanada/DCT/>

27