

学習情報ネットワーク論 (第2回)

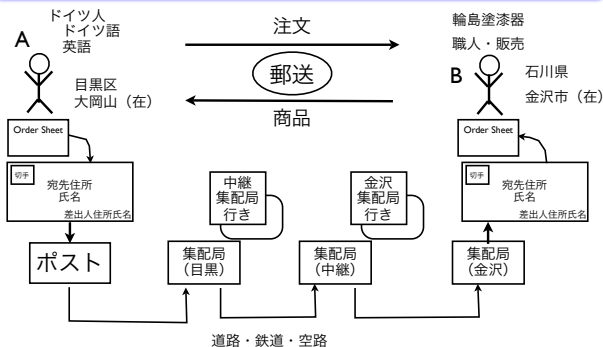
室田真男
大学院社会理工学研究科 人間行動システム専攻
murota@hum.titech.ac.jp

今日の内容（2008年1月15日）

- プロトコル階層（復習をかねて）
- World Wide Web
 - ▶ HTTP (Hyper Text Transfer Protocol)
 - ▶ URI (Uniform Resource Identifier)
- キーワード
 - ▶ プロトコル, 階層構造
 - ▶ World Wide Web, W3C
 - ▶ HTTP, URI, URL

2

郵送の仕組みの例（復習）



3

手順の回答例（1）

手順	誰が	どこで	何をするか	利用する情報
注文書を書く	A	自宅で	英語or日本語で注文書を記述 注文者情報を記述	カタログ 日本語で書く場合は、通訳等を利用
封筒に入れる	A	自宅で	宛先（住所・氏名） 差出人（住所・氏名）を記述 切手を貼る	カタログなど
投函する	A	最寄りの ポスト・ 郵便局	投函する	—
ポスト等から 集める	郵便局員C	ポスト ↓ 目黒の 集配局	ポストから手紙等の束を 集配局に届ける	ポストと集配局は あらかじめ 決まっている
次の集配局 ごとに分類	郵便局員D	目黒の集配局	封筒に書かれている住所・郵便番 号を元に、次の集配局を決める 同じ集配局宛の郵便物を集約し て、宛先札（集配局宛）を付ける	封筒に書かれている 郵便番号や住所 (ただし、県や市レベルまで)

4

手順の回答例（2）

手順	誰が	どこで	何をするか	利用する情報
次の集配局へ 運搬	運搬担当E	道路・鉄道・ 空路	次の集配局へ運搬	宛先集配局
：	：	：	：	：
金沢の集配局へ 運搬	運搬担当F	道路	金沢の集配局へ運搬	宛先集配局
家に配送	郵便局員G	金沢集配局 ↓ 職人Bの家	職人Bの家のポストへ配送	封筒に書かれている住所 (市、町、番地)
Bさんへ渡す	家の誰か	Bさんの家	宛先人を元にBさんへ渡す	封筒に書かれている宛先人
注文書を確認	Bさん	Bさんの家	封筒を開封し、注文書を確認 (何語で書かれているか、何を注 文しているか)	注文書の内容

5

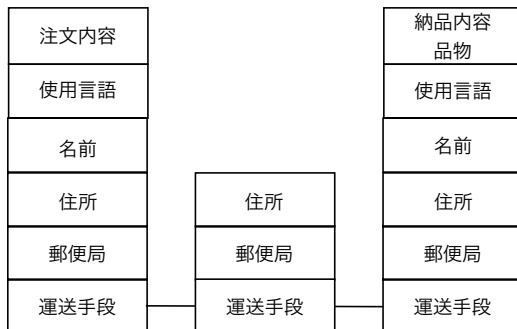
反対向きは省略

プロトコル

- プロトコルとは
 - ▶ コンピュータシステムで、データ通信を行うために定められた規約。情報フォーマット、送信手順、誤り検出法などを定める。（広辞苑第5版）
 - ▶ Computers, a set of rules governing the format of messages that are exchanged between computers. (Random House Webster's Unabridged Dictionary)
 - ▶ 互いにコミュニケーションするための取り決め

6

郵送のプロトコル階層



7

階層化 (Layering)

- プロトコル設計デザインの複雑さを減らす
- 機能を分離し、各層の独立性を高める
- n層どうしが通信を行っているつもり
- 実際には下の層が提供する機能（サービス）を利用して、相手側の同じ層と通信する
- プロトコル群（Protocol Suite）
- プロトコルスタック（Protocol Stack）

8

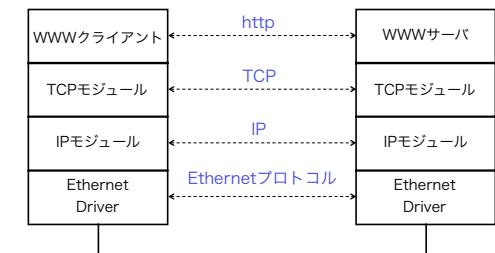
インターネットプロトコル階層

プロトコルスタック

アプリケーション	HTTP, SMTP, DNS, FTP
トランスポート	TCP/UDP
ネットワーク	IP/ルーティングプロトコル
データリンク	イーサネット
物理	ペア対線、光ファイバ

9

World Wide Webでは



10

Webの誕生



Web2.0への道（インプレスムック）より

- Tim Berners-Lee
 - ▶ イギリス出身の物理学者
 - ▶ Web生みの親（考案・開発）
 - ▶ URL, HTML, HTTP開発者
 - ▶ Web誕生の日
 - 1991年8月6日
 - Webの仕組みを公開
 - ▶ W3Cディレクタ
 - ▶ セマンティックWebを提唱

<http://www.w3.org/People/Berners-Lee/>

11

従来との違い

- オープン
 - ▶ 仕様を公開
- インターネットとハイパーテキストの融合
 - ▶ URI
- 簡単
 - ▶ HTMLというシンプルな記述方式
 - ▶ HTTPというシンプルなプロトコル
 - ▶ URIというシンプルなリンク記述
 - 一方的にリンクを張ればよい

12

World Wide Webの初期

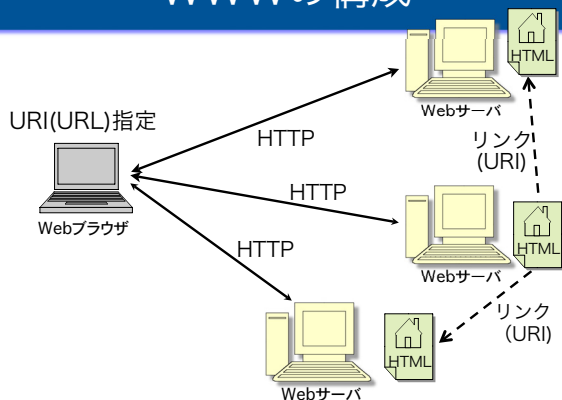
1989年	Tim Berners-Leeが、CERN (European center for nuclear research, ヨーロッパ素粒子物理研究所)で提案
1991年8月6日	World Wide Web プロジェクトに関する簡単な要約をalt.hypertextニュースグループに投稿
1991年12月	Hypertext'91会議で公式デモンストレーション
1993年2月	最初のグラフィックインタフェースMosaicがNCSAからリリース (ソースコードを無料で公開)
1994年	World Wide Web Consortium (W3C)設立 -Tim Berners-Leeが初代会長 (現在顧問) -http://www.w3.org/
1995~2000	ブラウザ戦争 (Internet Explorer v.s. Netscape Navigator)

World Wide Webの特徴

- 情報提供サービス
- 情報：HTML文書
- 提供：WebサーバとWebブラウザ
 - ▶ サーバ・クライアントモデル
- 情報の関連づけ
 - ▶ ハイパーテキストリンク (Hypertext Link)
 - 別のサーバ上の情報を「指す」 (一方向)

14

WWWの構成



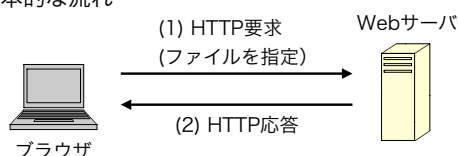
WWWの構成要素

- コンテンツへのリンク (URI/URL)
- コンテンツへのアクセス (HTTP)
- コンテンツ (HTML)
- Webサーバ
 - ▶ Apache, Microsoft IIS など
- ブラウザと呼ばれるクライアント
 - ▶ Firefox, Internet Explorer, Netscape, Mozilla, Opera, Safariなど

16

HTTP

- Hyper Text Transfer Protocol
- 現在は、バージョン1.1
- HTTPに求められる最小限の機能
 - ▶ ファイルの転送
- 基本的な流れ



17

HTTPの要求

- 典型的な要求メッセージの例

```
GET /somedir/page.html HTTP/1.1
Host: www.someschool.ac.jp
Connection: close
User-agent: Mozilla/5.0
Accept-language: ja
```

- 要求フォーマット

```
メソッド<SPC>URL<SPC>HTTPバージョン<CR><LF>
ヘッダ名:<SPC>値 <CR><LF> (0行以上)
<CR><LF>
ボディ(もしあれば) <CR><LF>
```

要求行
ヘッダ行
空白行

HTTP要求メソッド

GET	サーバに対してHTML文書や画像データなどを要求
HEAD	サーバに対してヘッダ情報（メタデータ）を要求
POST	サーバ側のCGIへ、パラメータの引き渡し
PUT	サーバに対してデータを送信する
DELETE	ユーザ認証後、サーバのデータを削除する

HTTPの応答

- 一般的なHTTP応答メッセージ

```
HTTP/1.1 200 OK
Connection: close
Date: Fri, 15 Oct 2004 12:00:10 GMT
Server: Apache/1.3.29 (Unix)
Last-Modified: Thu, 10 Jun 2004 05:28:20 GMT
Content-Length: 5556
Content-Type: text/html

(data data data)
```

- 応答のフォーマット

バージョン<SPC>ステータスコード<SPC>ステータス文<CR><LF>
ヘッダ(0行以上) <CR><LF>
<CR><LF>
ボディ(もしあれば: テキストorバイナリデータなど) <CR><LF>

ステータス行
ヘッダ行
空白行

HTTP/1.1ステータスコードの例

200 OK	要求が満たされ、要求された情報が返送される
301 Moved Permanently	要求されたリソースには新しいURLが割り当てられている。新しいURLが応答メッセージ内のLocation:行に記載される。クライアントは自動的に新しいURLにアクセスに行く。
304 Not Modified	ドキュメントは更新されていない(条件付きGET) クライアントはキャッシュのデータを利用できる
404 Not Found	見つからない
505 HTTP Version Not Supported	HTTPのバージョンをサポートしていない

実際のデータを見てみよう

- telnetを利用
 - 80番ポートに接続
- フリーソフトWireshark (旧 Ethereal) を利用
 - Wiresharkのオリジナルサイト
 - <http://www.wireshark.org/>
 - Etherealのオリジナルサイト
 - <http://www.ethereal.com/>
 - Etherealの日本語解説サイト
 - <http://www.space-peace.com/ethereal/ethereal.htm>

HTTP/1.1ヘッダの例（１）

ヘッダ名	意味	要求	応答	本体
Connection	接続制御	○	○	
Date	日時	○	○	
Cache-Control	キャッシュ制御情報	○	○	
MIME-Version	マイムバージョン	○	○	
Accept	対応するメディアの形式	○		
Accept-Charset	対応する言語セット	○		
Accept-Language	対応する言語	○		
Accept-Encoding	対応するエンコード方式	○		
Host	サーバのホスト名	○		
Cookie	前回アクセス時のサーバからのクッキー情報	○		
If-Modified-Since	ある日時以降に更新されていたらリクエストを実行	○		

HTTP/1.1ヘッダの例（２）

ヘッダ名	意味	要求	応答	本体
From	電子メールアドレス	○		
Referer	リクエスト元のURI	○		
User-Agent	ユーザエージェント	○		
Location	違う場所へのリダイレクト		○	
Server	サーバソフトウェアの名前		○	
Set-Cookie	クッキー情報		○	
ETag	本体の識別情報		○	
Content-Length	データ長（バイト）			○
Content-Type	本体のメディアタイプ			○
Content-Encoding	エンコーディング方式			○
Content-Language	本体の言語			○
Last-Modified	最終更新日時			○
Expires	無効になる日時			○

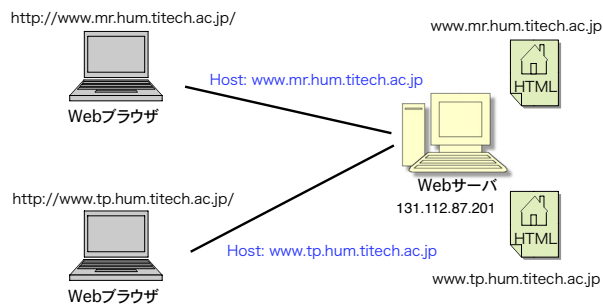
HTTPヘッダの用途

- 送信ファイル内容の説明
 - ▶ Content-Length
 - ▶ Content-Type
 - ▶ Content-Language
 - ▶ Last-Modified
- キャッシュファイルの利用
 - ▶ If-Modified-Since
 - ▶ Expires
- 連続したリクエスト
 - ▶ Connection
- バーチャルホスト
 - ▶ Host

25

バーチャルホスト

- 一つのWebサーバソフトウェアが複数のドメインのサービスを提供可能



26

URIとは

- URI (Uniform Resource Identifier)
 - ▶ インターネット上の資源を一意に識別するアドレス記述法
 - ▶ 先頭にスキーム、そしてコロン、スキームごとのリソース指定方法
 - <scheme> : <scheme-specific-part>
 - ▶ スキーム：一般的には計画・枠組みという意味、ここでは通信言語や通信方法をさす
- URIスキーム
 - ▶ <http://www.iana.org/assignments/uri-schemes>
 - http, ftp, https, mailto, file, telnet...

27

URL?

- URLのほうがよく聞くけど…
 - ▶ Uniform Resource Locator
 - ▶ <http://www.w3.org/TR/uri-clarification/>
 - ▶ 現在はURIスキームのひとつと捉えられている
 - ▶ リソースの場所を表す
 - ▶ 「便利だけれども非公式の概念」

28

http: URI

- http: URIの形式
 - ▶ http://ホスト名 (アドレス) /パス名
 - ▶ ホストアドレス
 - IPアドレス
 - フルドメイン名 FQDN (Fully Qualified Domain Name)
 - ▶ パス名
 - Webサーバ上のファイル名あるいはディレクトリ名
 - ディレクトリを指定する場合は最後に必ず「/」をつける

29