ネットワークアプリケーション

第3回 アプリケーションプロトコルの設計（1）

石井 健太郎
（423研究室・オフィスアワー水3限）

スケジュール

9月15日　第1回「TCP/IPプロトコルスイート」
9月29日　第2回「ネットワークアプリケーションのプログラミングモデル」
10月6日　第3回「アプリケーションプロトコルの設計（1）」
10月13日　第4回「アプリケーションプロトコルの設計（2）」
10月20日　第5回「アプリケーションプロトコルの設計（3）」
10月27日　第6回「アプリケーションプロトコルの設計（4）」　演習（第3演習室）
11月10日　第7回「サーバサイドウェブプログラミング（1）」
11月17日　第8回「サーバサイドウェブプログラミング（2）」

スケジュール

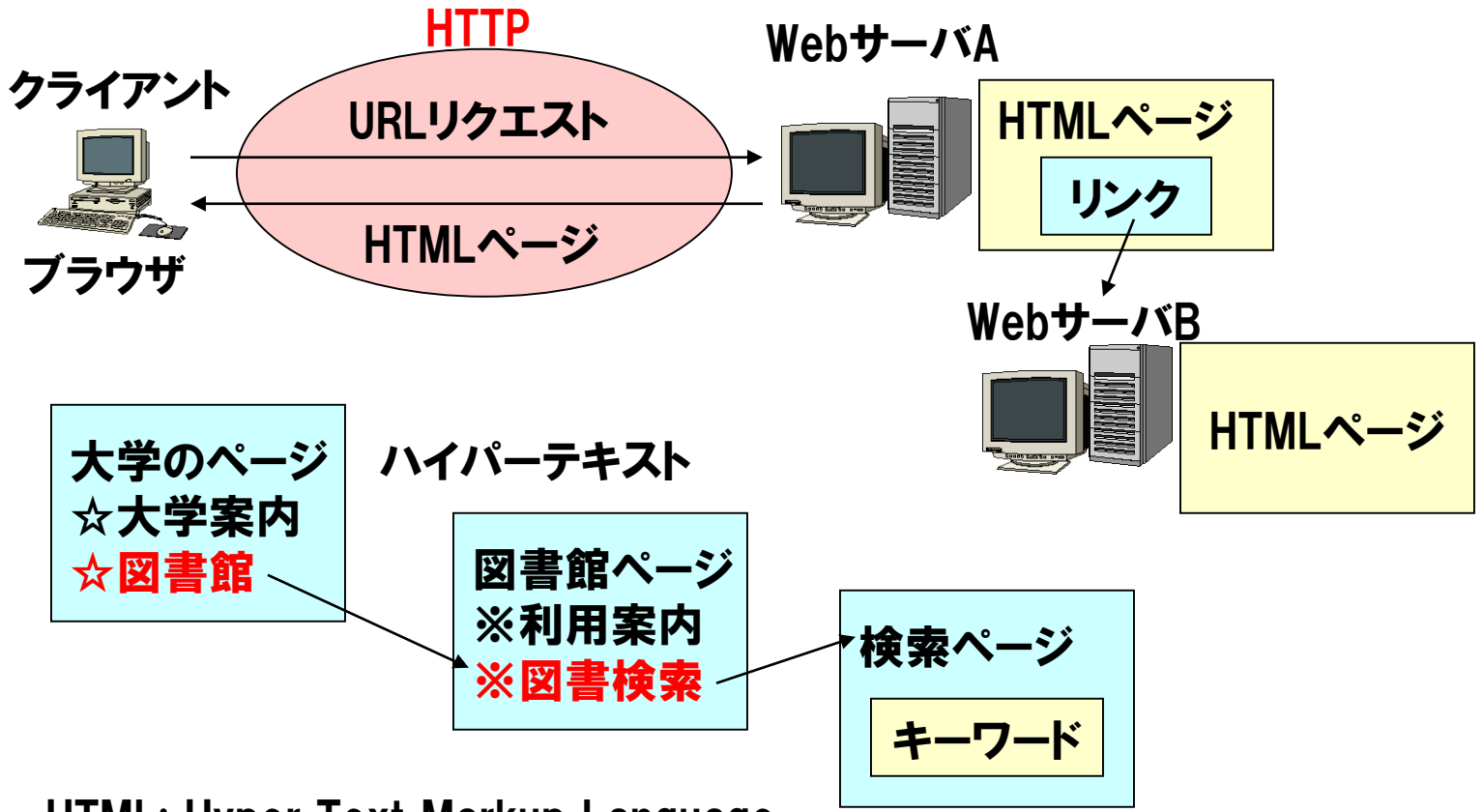
11月24日　第9回「サーバサイドウェブプログラミング（3）」
12月1日　第10回「サーバサイドウェブプログラミング（4）」　演習（第3演習室）
12月8日　第11回「クライアントサイドウェブプログラミング（1）」
12月15日　第12回「クライアントサイドウェブプログラミング（2）」
12月22日　第13回「クライアントサイドウェブプログラミング（3）」
1月12日　第14回「クライアントサイドウェブプログラミング（4）」　演習（第3演習室）
1月19日　第15回「まとめと演習」　演習（第3演習室）

- レポートを提出してください
  - 講義終了時でも可

- 今日は既存のアプリケーションプロトコルについて学びます
  - HTTP（Hypertext Transfer Protocol）
  - SMTP（Simple Mail Transfer Protocol）

WWW（World Wide Web）

ハイパーテキストに基づく、インターネットでの情報共有、サービス提供のシステム



HTML: Hyper Text Markup Language
HTTP: Hyper Text Transfer Protocol

WWW（World Wide Web）

URI（Uniform Resource Identifier）またはURL（Uniform Resource Locator）

サーバリソース(ドキュメント、プログラム)のアドレス

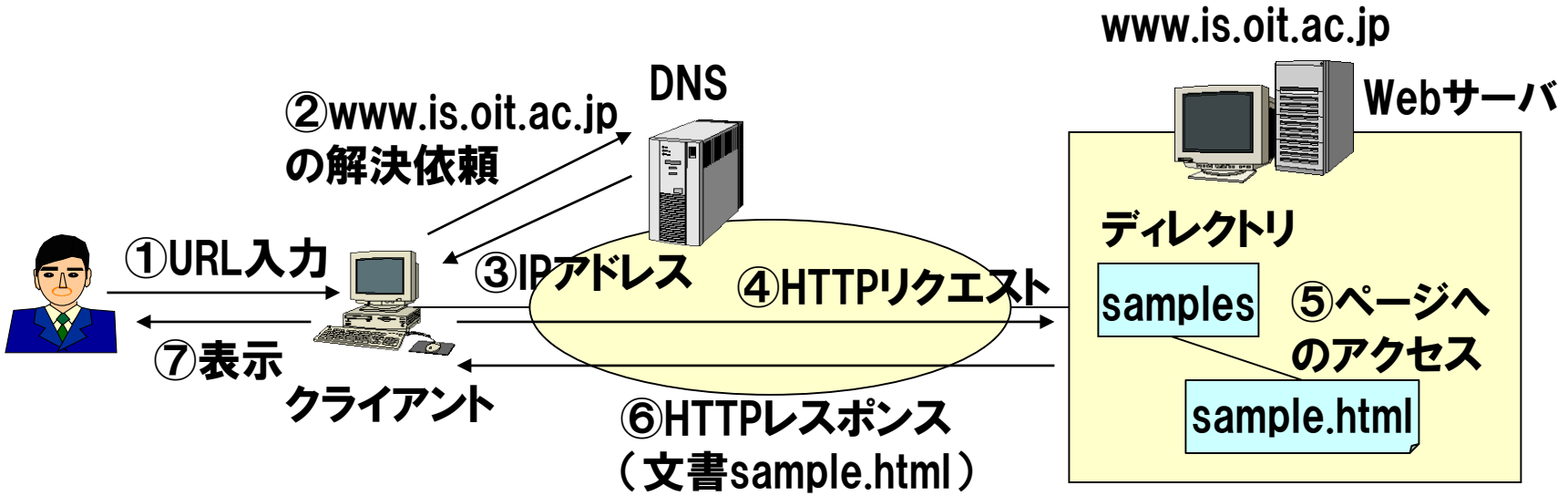
形式　http://サーバの場所:ポート番号/ファイルの場所

http のウエルノウンポート80番は省略可能

ホスト名　　ファイル名

（例） http://www.is.oit.ac.jp/samples/sample.html

ディレクトリ



ウェブページが表示されるまで

HTMLに引き続き，
画像データなど付属データも，
1つずつダウンロードされる

クライアント
（ブラウザ）

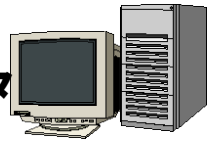
①index.htmlの要求

②index.htmlのテキスト

③header_logo.gifの要求

④header_logo.gifの画像データ

Webサーバ



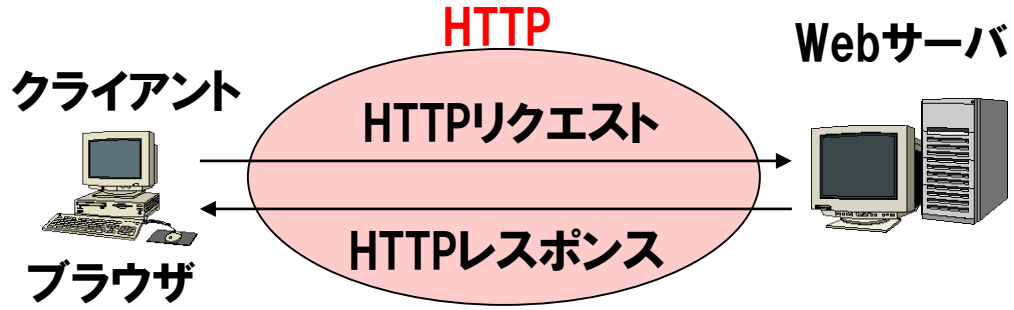
⑤oit.jpgの要求

⑥oit.jpgの画像データ

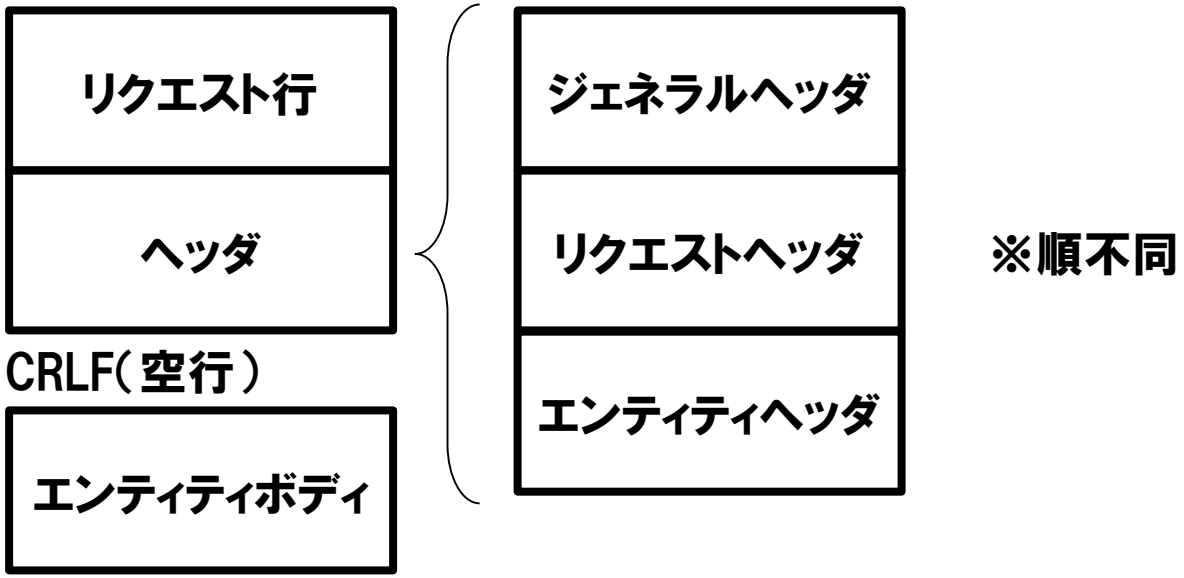
HTTP（Hypertext Transfer Protocol）

- ハイパーテキスト（HTMLなど）転送プロトコル
  - ブラウザがウェブサーバからファイルを取得するために使用
  - メッセージパッシング
  - ヘッダはテキストデータ（ASCII文字列），
    ボディはテキストデータもバイナリデータも扱う
  - TCP
    - ポート番号80番

HTTPリクエストメッセージ



- 1行目はリクエスト行
- 2行目から空行まではヘッダ
- 空行以降がエンティティボディ（実際のデータ（ある場合））

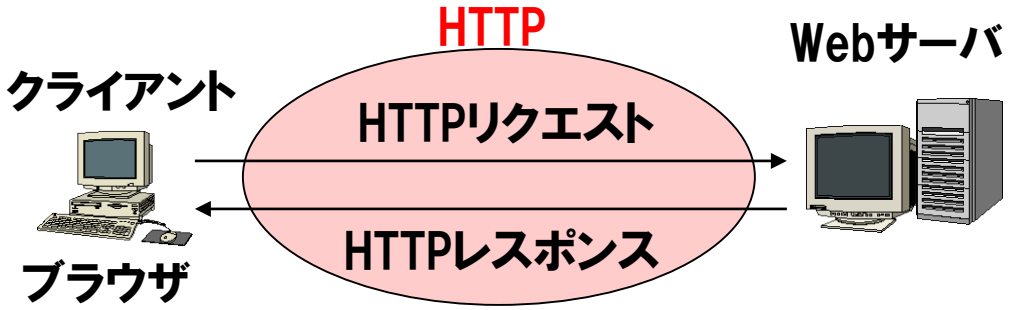


リクエスト行

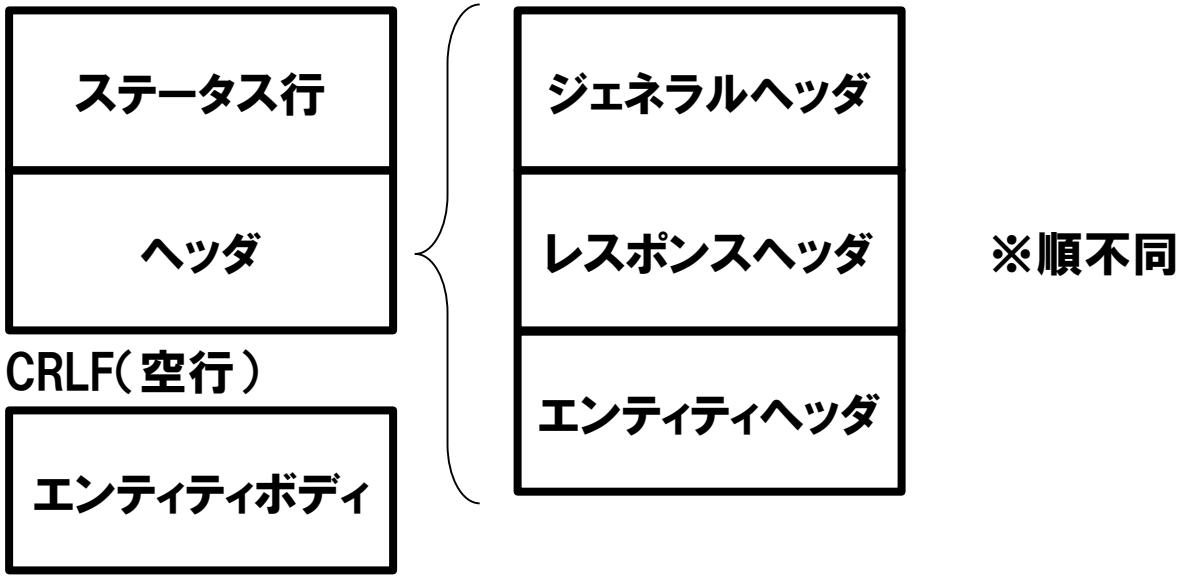
- メソッド（空白）URI（空白）HTTPバージョン
  - GET /index.html HTTP/1.1

メソッド	概要	備考
OPTIONS	Webサーバがサポートするリクエストメソッドなど、機能の照会	
GET	リソース（Webページ、イメージ、オーディオファイルなど）の要求	リソース名の直後に？を付加し、パラメータを渡せる
HEAD	リンクの確認やリソースの更新のチェック	
POST	Webサーバにデータ（エンティテイボディ）を渡し、処理を要求（DBの更新、注文など）	
PUT	指定したリソースへのデータの保存（更新）要求	認証によるアクセス制御が必要
DELETE	リソースの削除要求	同上
TRACE	HTTPリクエストの通過ノードの確認	

HTTPレスポンスメッセージ



- 1行目はステータス行
- 2行目から空行まではヘッダ
- 空行以降がエンティティボディ（実際のデータ）



ステータス行

- HTTPバージョン（空白）ステータスコード（空白）メッセージ
  - HTTP/1.1 200 OK

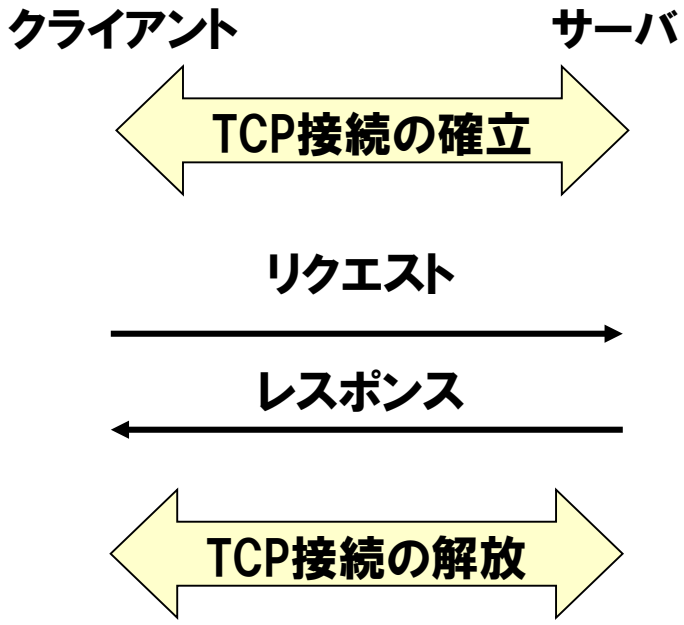
コード	メッセージ
200	OK
301	Moved Permanently
307（302）	Temporary Redirect （Moved Temporarily）
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error
503	Service Unavailable

接続形態

（１）リクエスト単位の接続

○リクエスト毎にTCP接続を確立・解放

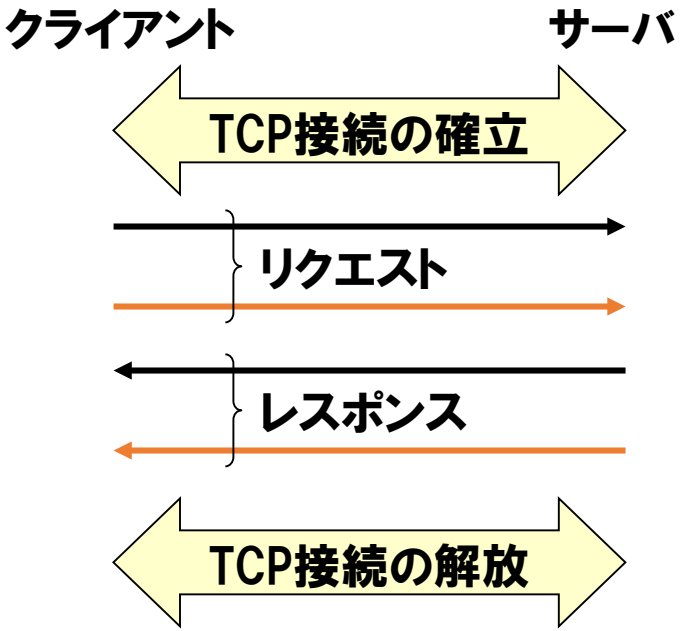
ヘッダ
Connection: close
で明示的に指定できる



（２）持続型接続（HTTP/1.1から可能）

○一度TCP接続を確立したら、どちらか一方が接続解放を要求するまで接続を継続（Keep Alive）

○クライアントはレスポンスを待たず、複数のリクエストを送信可能（パイプライン）

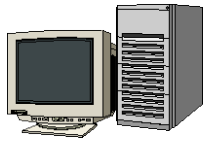


GETリクエスト

・URLで指定したリソース(ページ、映像等)の取得要求
・パラメータ長が短い場合(検索等)に使用

クライアント

Webサーバ



GET /index.html

index.htmlのページ

GET /index.html
If-Modified-Since: Tue, 29 Sep 2015 09:45:23

index.htmlのページ

Index.htmlの
最終更新日時:
2015年10月1日

GET /index.html
If-Modified-Since: Wed, 21 Oct 2015 14:09:34

304 not modified(更新されていない)

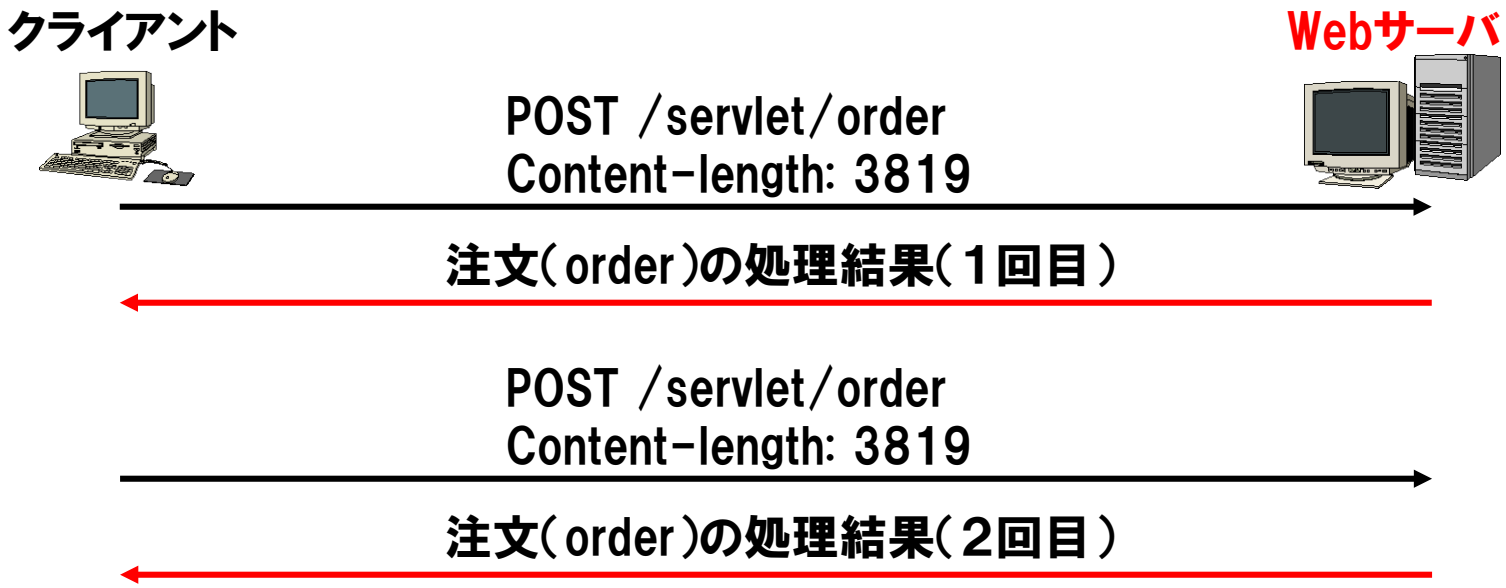
GET /servlet/search?map+book

空白は+で置換

プログラムsearchにパラメータ"map book"を与えた場合の結果

POSTリクエスト

・URLで指定したリソース(プログラム等)へデータを渡し、
　処理を要求(DBの更新や注文等)
・パラメータ長が長い場合に使用
・データを第三者に見られたくない場合(要https)に使用



データサイズをContent-lengthで指定

HTTPヘッダの例

リクエスト

```
GET / HTTP/1.1
Host: is.is.oit.ac.jp
Connection: keep-alive
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/45.0.2454.101 Safari/537.36
Accept-Encoding: gzip, deflate, sdch
Accept-Language: ja,en-US;q=0.8,en;q=0.6
Cookie: [216 bytes were stripped]
```

レスポンス

```
HTTP/1.1 200 OK
Date: Fri, 02 Oct 2015 06:52:33 GMT
Server: Apache/2.4.7 (Ubuntu)
Last-Modified: Fri, 03 Jul 2015 04:00:08 GMT
ETag: "4e1-519f0978843f5-gzip"
Accept-Ranges: bytes
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 555
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html
```

HTTPヘッダの例

リクエスト

```
GET / HTTP/1.1
Host: www.yahoo.co.jp
Connection: keep-alive
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/45.0.2454.101 Safari/537.36
Accept-Encoding: gzip, deflate, sdch
Accept-Language: ja,en-US;q=0.8,en;q=0.6
Cookie: [499 bytes were stripped]
```

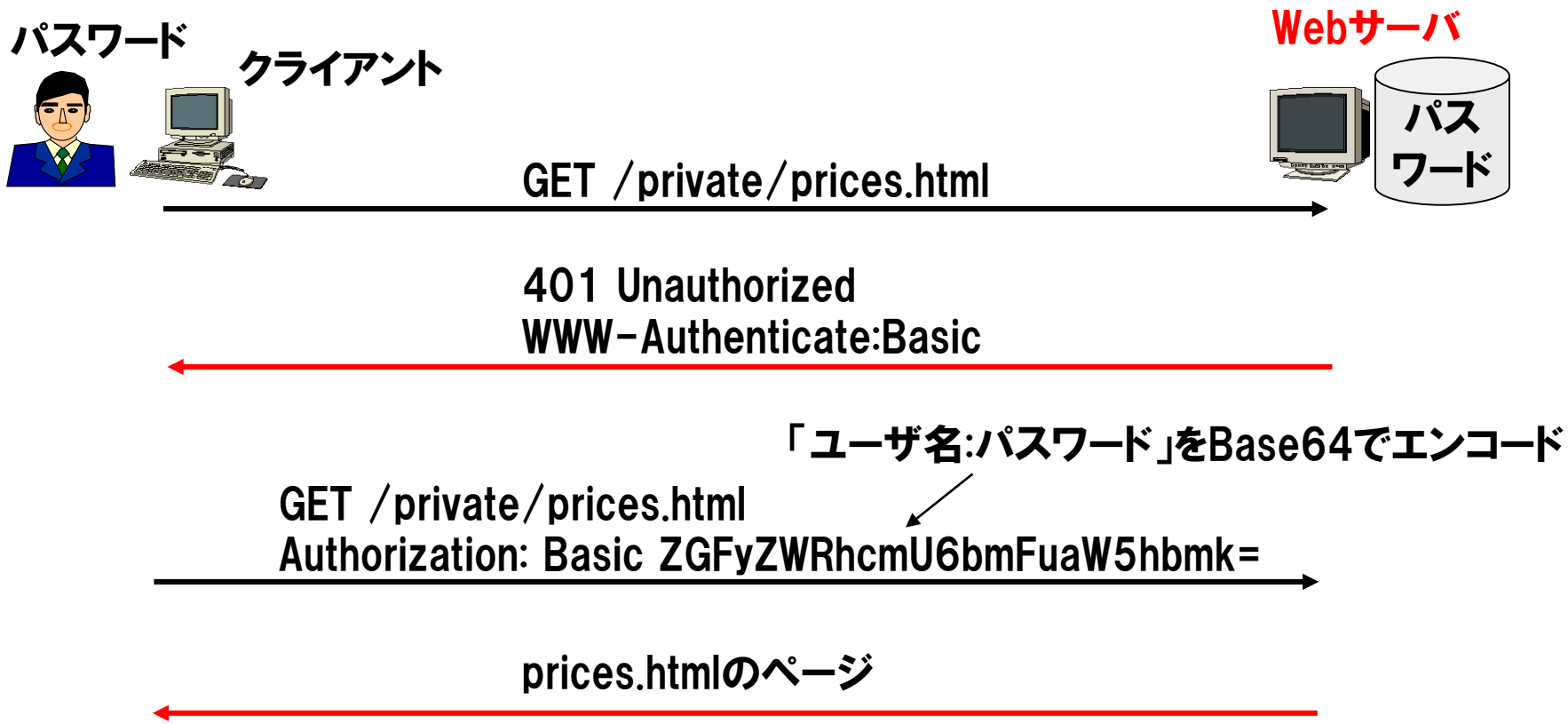
レスポンス

```
HTTP/1.1 200 OK
Server: nginx
Date: Fri, 02 Oct 2015 07:02:53 GMT
Content-Type: text/html; charset=UTF-8
P3P: policyref="http://privacy.yahoo.co.jp/w3c/p3p_jp.xml", CP="CAO DSP COR CUR ADM DEV TAI (略)"
Cache-Control: private, no-cache, no-store, must-revalidate
Expires: -1
Pragma: no-cache
X-XRDS-Location: https://open.login.yahooapis.jp/openid20/www.yahoo.co.jp/xrds
Vary: Accept-Encoding
Content-Encoding: gzip
X-Frame-Options: SAMEORIGIN
X-Cache: MISS from h-cache2
X-Cache-Lookup: MISS from h-cache2:8080
Transfer-Encoding: chunked
Connection: keep-alive
```

HTTPでの認証

基本認証(Basic Authentication)

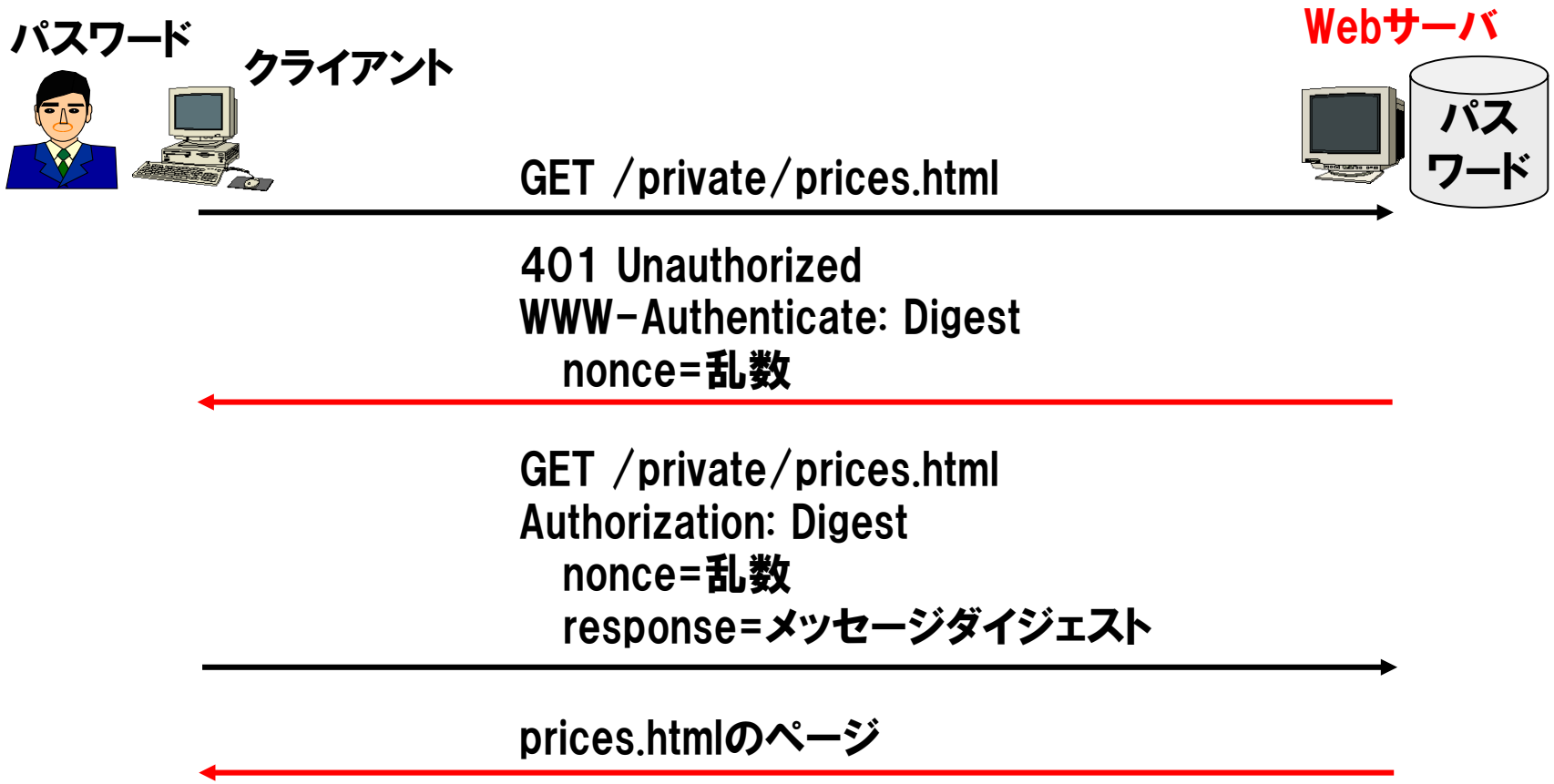
○パスワードによるユーザ認証
○ユーザ名、パスワードをBase64でエンコードし、返却



HTTPでの認証

ダイジェスト認証(Digest Access Authentication)

○メッセージダイジェストを使用したユーザ認証
○サーバから送信されたナンス値(乱数)やユーザ名、パスワード等
からMD5でメッセージダイジェストを作成し、返却

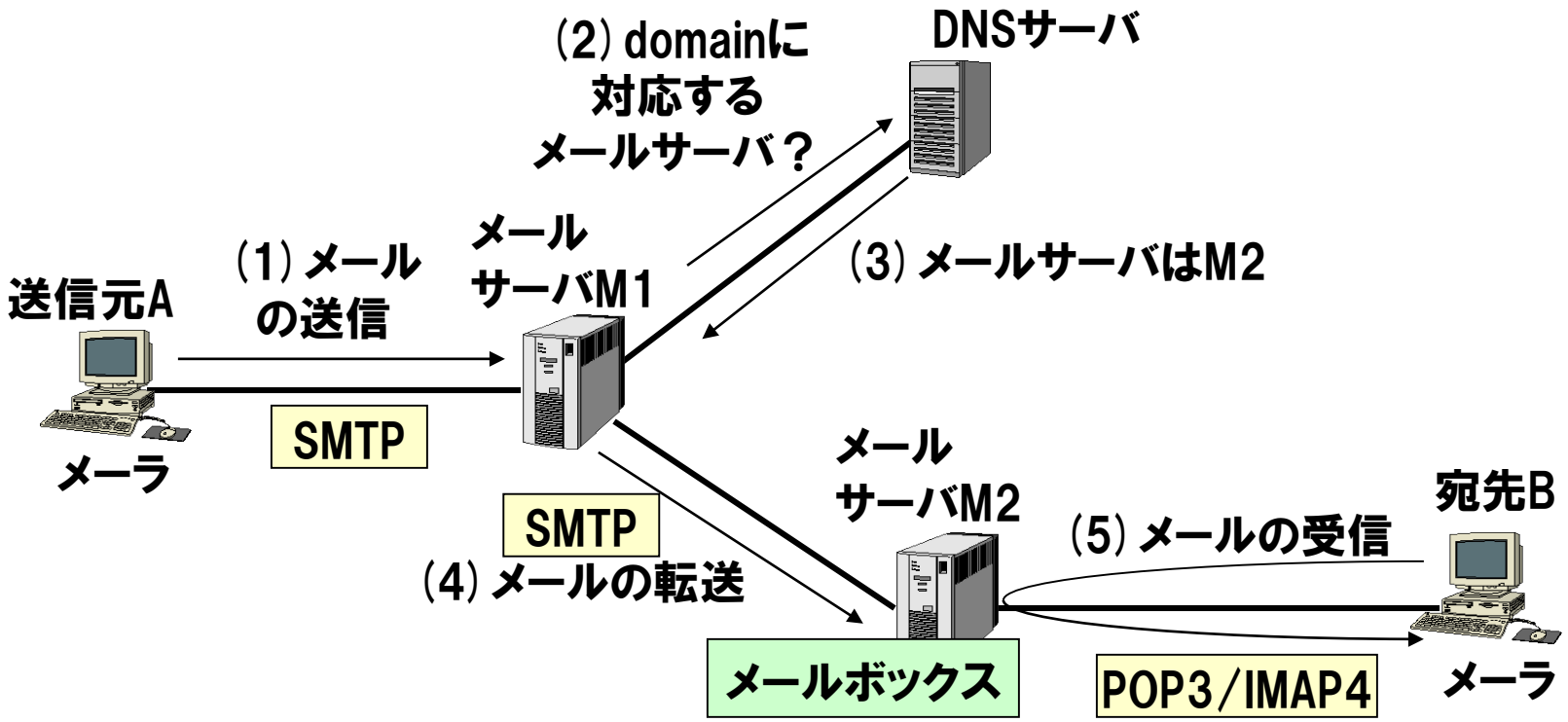


メールシステム

SMTP: Simple Mail Transfer Protocol
POP: Post Office Protocol
IMAP: Internet Message Access Protocol

電子メールアドレス　user@domain

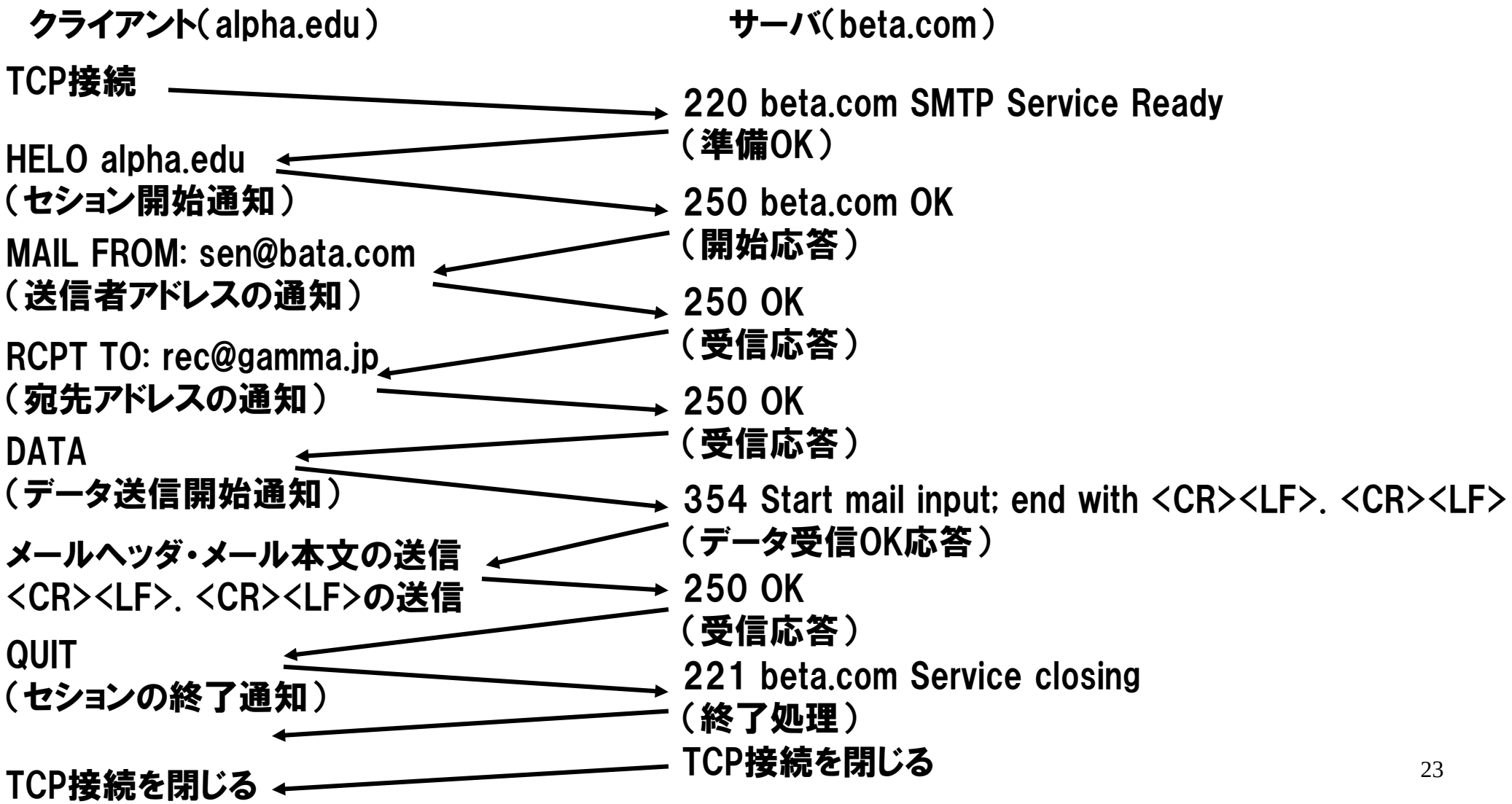
user：ユーザ名（メールボックスの識別名）
domain：メールの終点サーバのドメイン名



SMTP（Simple Mail Transfer Protocol）

- メール転送プロトコル
  - クライアントからメールサーバへの送信、メールサーバ間の転送に使用
  - メッセージパッシング
  - テキストデータ（ASCII文字列）
    - 7bit文字列でなければならない
  - TCP
    - ポート番号25番，ただし，迷惑メール対策で，
      クライアントからメールサーバへの送信には587番が使われていることが多い

SMTPのメッセージパッシング



SMTPコマンド

コマンド	パラメータ	説明
EHLO (HELO)	クライアントのドメイン名	セションの開始 サーバへのクライアントの通知
MAIL FROM	送信者メールアドレス	サーバへの送信者アドレスの通知
RCPT TO	宛先メールアドレス	サーバへの宛先アドレスの通知
DATA	なし	サーバへのメッセージ送信
RSET	なし	相手の内部状態のリセット
NOOP	なし	操作なし
QUIT	なし	セションの終了通知

SMTP実行例

（※赤字は入力）

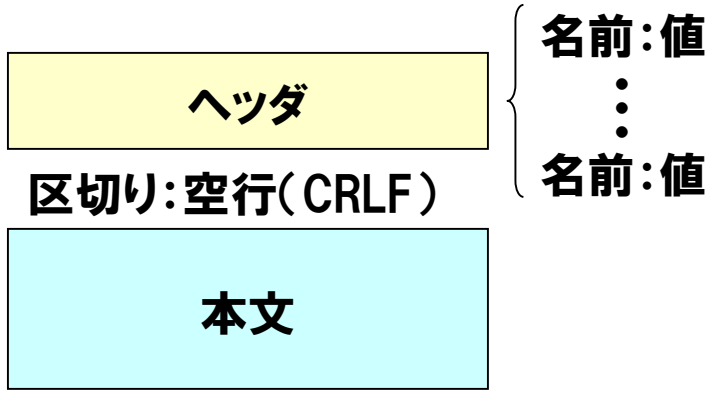
```
> telnet is.is.oit.ac.jp 587
220 is.is.oit.ac.jp ESMTP Postfix (Ubuntu)
EHLO is.is.oit.ac.jp
250-is.is.oit.ac.jp
250-PIPELINING
250-SIZE 10240000
250-VRFY
250-ETRN
250-STARTTLS
250-ENHANCEDSTATUSCODES
250-8BITMIME
250 DSN
MAIL FROM: <kenta@is.is.oit.ac.jp>
250 2.1.0 Ok
RCPT TO: <kenta@is.is.oit.ac.jp>
250 2.1.5 Ok
DATA
354 End data with <CR><LF>.<CR><LF>
From: kenta@is.is.oit.ac.jp
To: kenta@is.is.oit.ac.jp
Subject: test

This is a test mail.
.
250 2.0.0 Ok: queued as 22BD81209C9
QUIT
221 2.0.0 Bye
```

ホストとの接続が切断されました。

メールメッセージ
（DATAコマンド後のデータの中身）

メールメッセージの構成



メールメッセージの例

```
Return-Path: <kenta@is.is.oit.ac.jp>
X-Original-To: kenta@is.is.oit.ac.jp
Delivered-To: kenta@is.is.oit.ac.jp
Received: from [192.168.241.193] (unknown [150.89.250.195])
        (using TLSv1.2 with cipher ECDHE-RSA-AES128-GCM-SHA256 (128/128 bits))
        (No client certificate requested)
        by is.is.oit.ac.jp (Postfix) with ESMTPS id A224F1209C9
        for <kenta@is.is.oit.ac.jp>; Mon,  5 Oct 2015 13:20:52 +0900 (JST)
To: kenta@is.is.oit.ac.jp
From: Kentaro Ishii <kenta@is.is.oit.ac.jp>
Subject: test
Message-ID: <5611FAB4.9040109@is.is.oit.ac.jp>
Date: Mon, 5 Oct 2015 13:21:08 +0900
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:38.0) Gecko/20100101
 Thunderbird/38.3.0
MIME-Version: 1.0
Content-Type: text/plain; charset=iso-2022-jp
Content-Transfer-Encoding: 7bit
```

```
This is a test mail.

--
ESC$B@P0f7rB@O:ESC(B
kenta@is.is.oit.ac.jp
ESC$BBg:e9)6HBg3XESC(B ESC$B>pJs2J3XItESC(B ESC$BFCG$9V:UESC(B
http://www.oit.ac.jp/is/~is/server/~kenta/index.html
```

メールヘッダ

ヘッダ	意　味
To	宛先のメールアドレス。複数のアドレス、複数のToヘッダも可
Cc	メールの写しの宛先アドレス
Bcc	メールの写しの宛先アドレス。他の宛先には送信されないヘッダ
Reply-To	返信先のメールアドレス（FromやSender以外に返信する場合）
Subject	標題
Received	メール配送時の経路情報
Date	送信日時
From	送信者（メッセージ作成者）のメールアドレス
Sender	送信者（メーリングリストマネジャ等の代理送信者）のメールアドレス
Message-ID	メッセージ識別子。世界中で一意の値（シリアル番号＋host.domain）
Return-Path	MAIL FROM コマンドに基づき、メールサーバが付加する返信先のメールアドレス
X-	ユーザ定義ヘッダ

（参考） 送信アドレスと受信アドレスについて

MAIL FROM
RCPT TO

SMTPのコマンドで使用する情報
転送サーバに伝える情報

DATA
　メールヘッダ
　　メーラで見える送信元
　　メーラで見える宛先
　メール本文

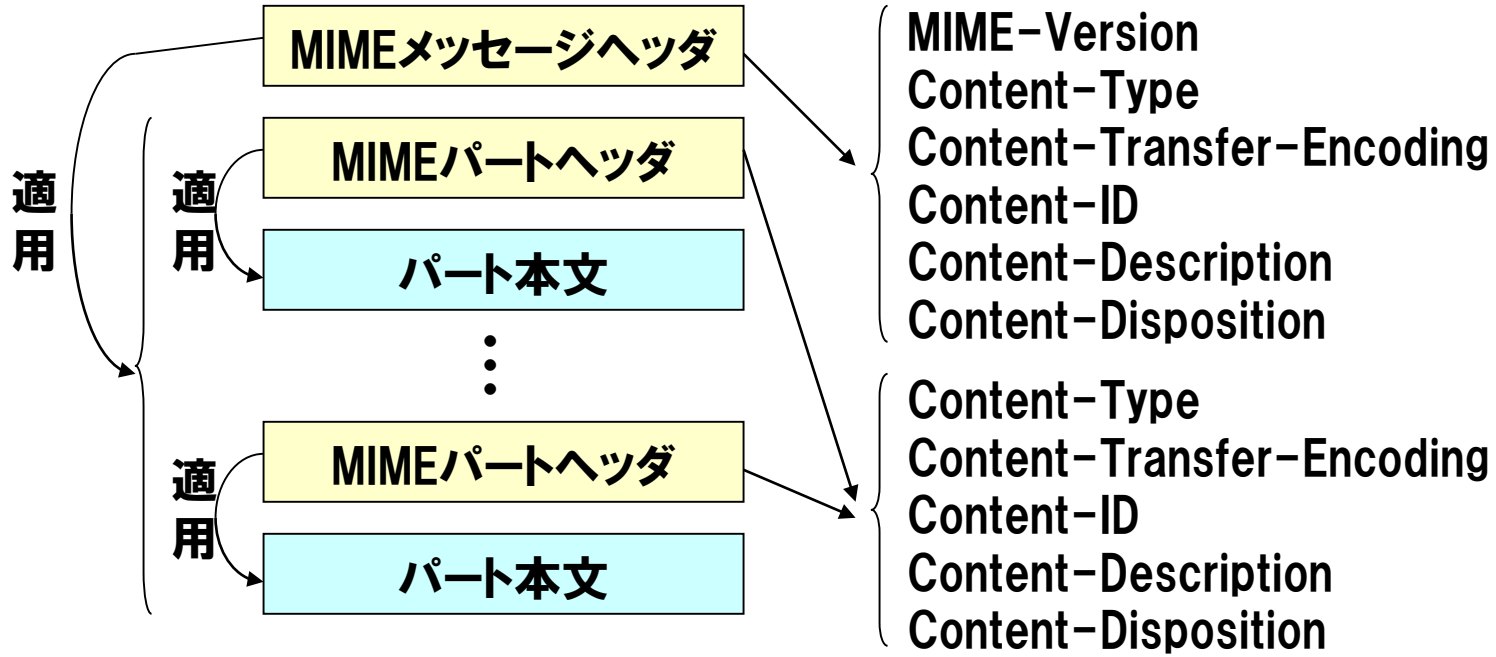
届いたメールに残る情報
受信メールでの表示に使用

アドレス	エンベロープ	メールヘッダ
送信元アドレス	MAIL FROM コマンドに設定	From フィールドに設定
宛先アドレス	RCPT TO コマンドに設定	To フィールドに設定
CCアドレス	RCPT TO コマンドに設定	Cc フィールドに設定
BCCアドレス	RCPT TO コマンドに設定	設定しない

（参考） MIME （Multipurpose Internet Mail Extensions）

・英字以外の文字データ、バイナリデータ（プログラム、画像、音楽等）をテキスト（ASCII文字コード）として表現すると共に、複数の本文を一つの本文として扱う（マルチパート形式）ための仕様
・MIMEにより、任意のコードを添付ファイルとして送信できる

MIMEメッセージの構造



（参考） MIMEヘッダ

ヘッダフィールド	意味
MIME-Version	MIMEのバージョン
Content-Type	メディアタイプ（text, image, audio, video, application, multipart, message）、MIMEバウンダリ（*）
Content-Transfer-Encoding	エンコード方式（7bit, 8bit, binary, quoted-printable, base64）
Content-ID	メッセージ間での本文参照に使用（IDは一意の値）
Content-Description	コンテンツに関する注釈
Content-Disposition	添付ファイルの表示方法（インライン表示等）

（*）MIMEバウンダリ

2つ以上のパートから構成される場合に、パート間の区切りを示す

Content-Type: multipart/mixed; boundary="----------B164240059B29C0E4EFEC397"

MIMEバウンダリ

宿題

- telnetコマンドを使うと，
  手入力でHTTP/SMTPサーバとの通信を試してみることができる
  これを用いて，任意のサーバとのHTTP・SMTPによる対話をそれぞれ報告せよ
  - 表示された画面をコピーして掲載する
    - ただし，パスワードはつぶしてもよい（というか，認証を必要とする通信は極力避ける）
    - サーバ・ポートがわかるように，コマンドラインも含めてコピーすること
  - 正しくHTMLファイルが取得できなくても，メール送信までいけなくても，
    それを理由に減点することはない．結果を正確に報告すること．
  - Windows, Mac, Linux のどれでも telnet は使えるが，
    （最近の）Windowsは機能の有効化が必要なので注意
  - SMTPサーバのポートは25番ではなく587番であることが多い
  - 学内サーバには，学外ネットワークからのアクセスが禁止されている（要VPN）

- レポート（A4用紙1～2枚）にして，次回の講義終了時までに提出
  - 様式自由，ただし，学生番号と名前をわかる場所に書くこと