知っていそうで案外知らない インターネットの仕組み (2)

2014年2月17日

長岡技術科学大学 経営情報系 湯川 高志

インターネット・プロトコル(続き)

OSIの階層モデルとインターネット・プロトコルの対応

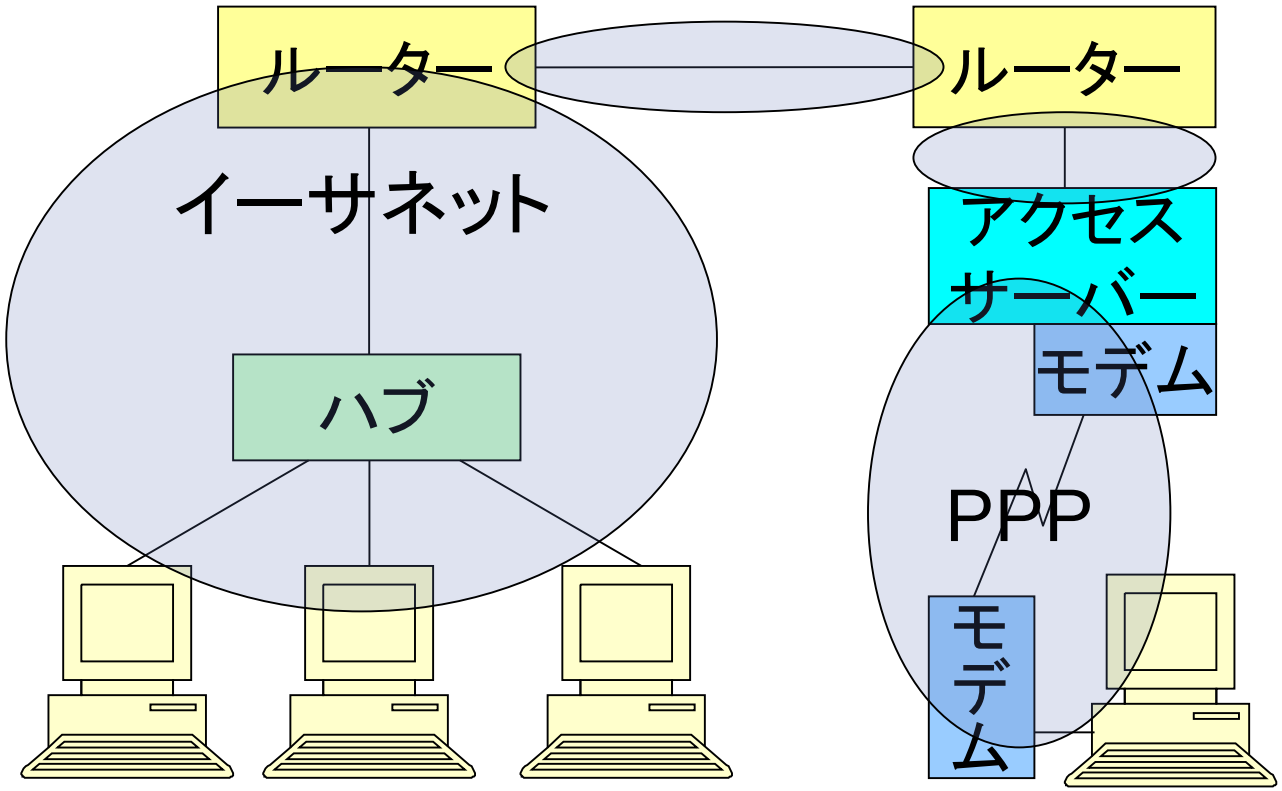
インターネット・プロトコルの階層	プロトコル	OSIの階層
アプリケーション層	HTTP, SMTP, POP, TELNET, FTP, …	7. アプリケーション層
		6. プレゼンテーション層
		5. セッション層
トランスポート層	TCP, UDP	4. トランスポート層
ネットワーク層	IP (Ineternet Protocol)	3. ネットワーク層
ネットワーク・インタフェース層またはMAC（メディアアクセス制御）層	イーサネット, PPP (モデム, ISDN)	2. データリンク層
		1. 物理層

TCP/IP

物理層とデータリンク層

- 物理層 (レイヤー1): 媒体の物理的(電気的)特性や信号の解釈を規定
  - データ・ビット(0と1)の信号としての表現
  - ケーブル
  - コネクタのピン配置
- データリンク層(レイヤー2): 同一媒体上で複数のコンピュータが相互にデータを伝達するための規定
  - 媒体上の個々のコンピュータの識別
  - 宛先，送信元，データの区切り(境界)の認識
  - データ誤りや異常が起こった際の対処（訂正，再送，破棄等）

物理層とデータリンク層（続き）



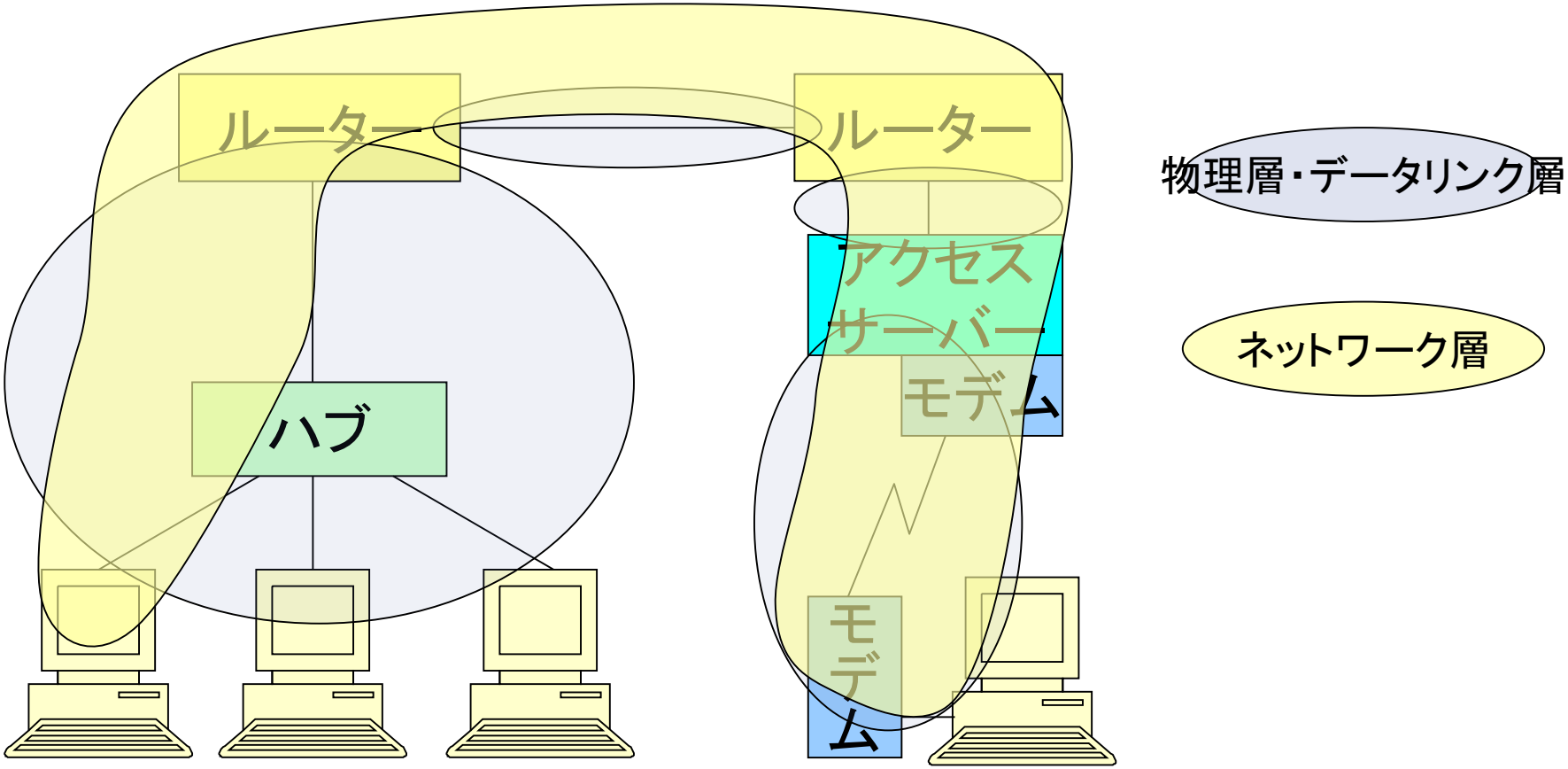
イーサーネット（ethernet)

- Xeroxのパロアルト研究所で開発，IEEEで規格化（IEEE802.xx）
- ツイストペア(撚り対線)→今，よく使われているイーサーネット: 10BASE-T, 100BASE-T
- もともとは同軸ケーブル: 10BASE5, 10BASE2
- ベースバンド伝送
- CSMA/CD
  - 他のノードが送信していない時を見計らって送信，運悪く衝突したら適当に待って再送
- ノードには世界で唯一からアドレスが付与される→MACアドレス
  - 6バイト: 3バイトがベンダー固有のアドレス，3バイトがベンダーで決める個体固有のアドレス
    (例: 00:02:8a:01:02:03)

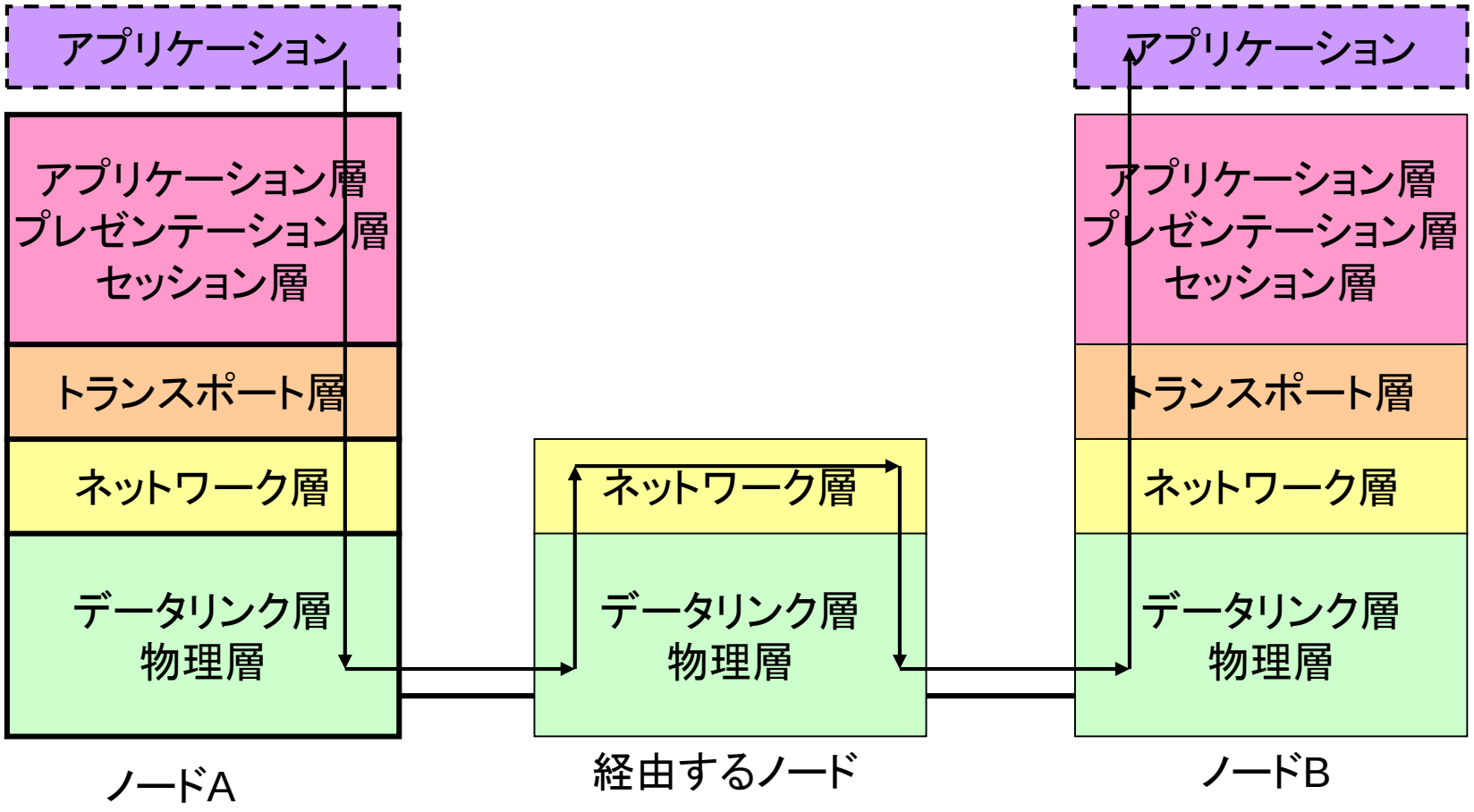
ネットワーク層 IP (Internet Protocol)

- 同一媒体で接続されたノードのデータ転送を規定するデータリンク層を利用して
- いくつのも媒体を介し
- ２つのノード間でのデータの転送を行うためのプロトコル
  - ネットワーク内でのノードの識別
  - データリンク層が扱うことのできるパケットサイズが異なる場合のデータブロックの分割・組立て
  - どのデータリンク層へパケットを送り出すかの選択＝ルーティング

ネットワーク層（続き）

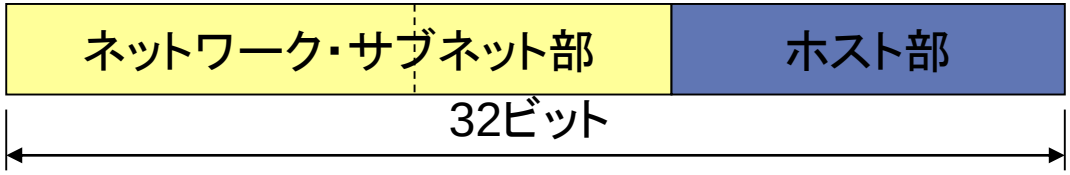


ネットワーク層


アプリケーション
アプリケーション層
プレゼンテーション層
セッション層
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
ノードA
ネットワーク層
データリンク層
物理層
経由するノード
アプリケーション
アプリケーション層
プレゼンテーション層
セッション層
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
ノードB

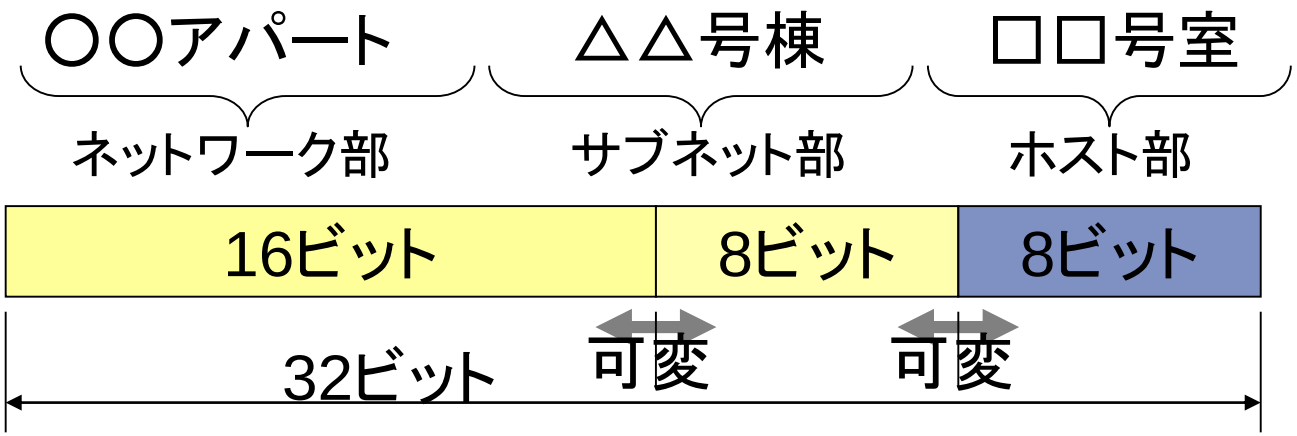
IPアドレス

- ネットワーク上のノードを識別するアドレス
  - ノードごとに付与されるインターネット上（世界中）で一意な番号
  - 4バイト (1バイト=8ビットなので32ビット)
  - ルーティングしやすいように，32ビットをネットワーク・サブネット部とホスト部に分けて使う



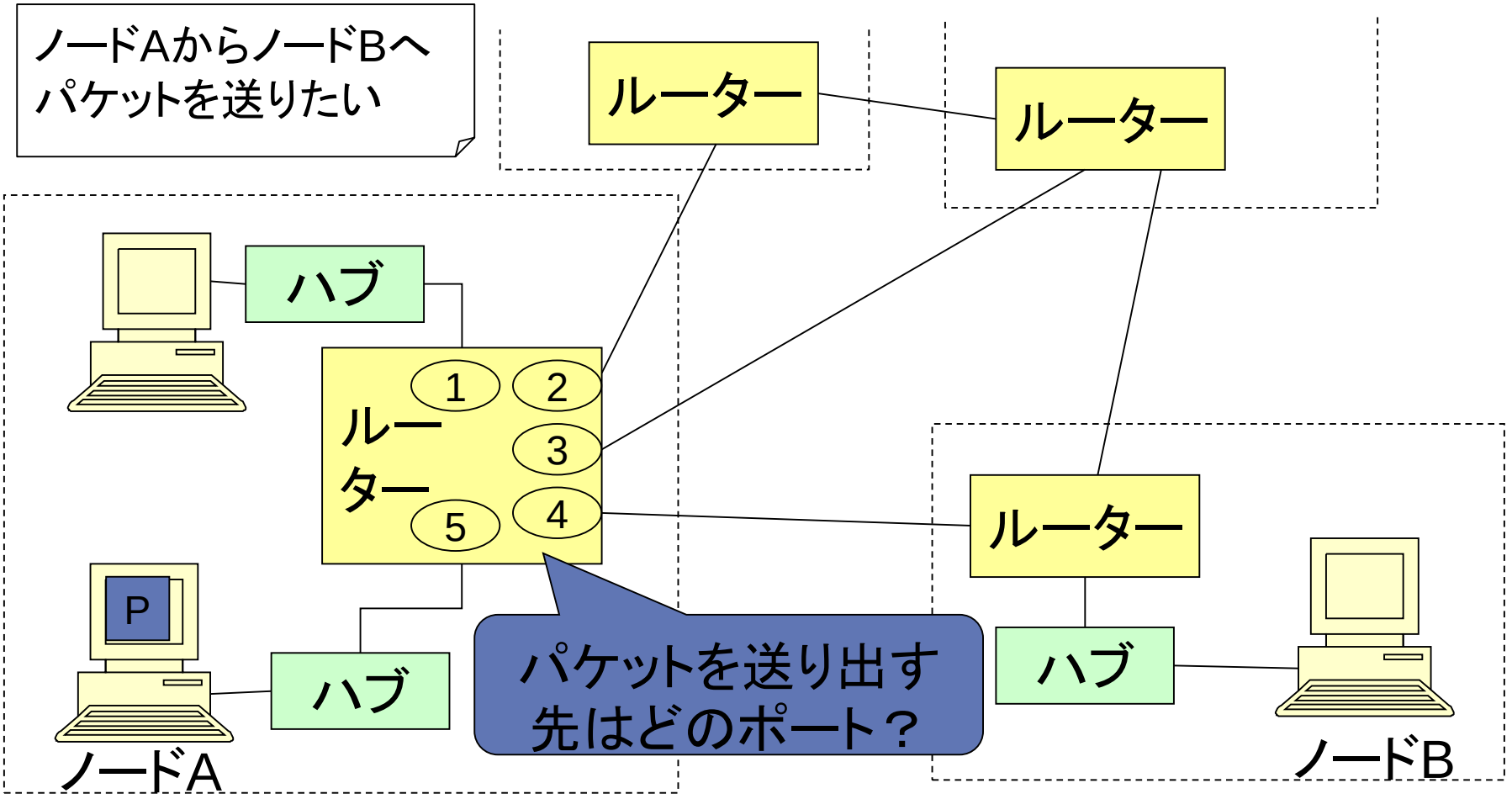
  - 各バイトの10進表記を'.'で区切って記述
    例: 133.44.72.27

IPアドレス（続き）



- ホスト部がすべて0:ネットワーク全体を表現
  - 表記: ネットワークアドレス/ネットワーク・サブネット部のビット数
- 例
  - 長岡技大のネットワーク: 133.44.0.0/16
  - 電気△号棟▲階: 133.44.72.0/24
  - そこにある，あるコンピュータ: 133.44.72.27

経路制御(ルーティング)


ノードAからノードBへ
パケットを送りたい
ルーター
ルーター
ハブ
1
2
ルー
ター
3
5
4
ルーター
P
ハブ
パケットを送り出す
先はどのポート？
ハブ
ノードA
ノードB

経路制御 (続き)

- ルーターの動作
  - 到着したパケットの宛先IPアドレスに基づいて
  - そのパケットを次にどのポートに出すか決め
  - 送出する

全体のことは気にしない

自分のまわりのことだけ考えている

- 宛先IPアドレスとそれを送出するポートの対応表:
  経路制御表(ルーティング・テーブル)

経路制御 (続き)

- 経路制御表をどうやって作る?
  - 人間が手作業で
  - ルーター同士が経路情報を交換して自動的に構築 → ルーティング・プロトコル
- 手作業で作るのは大変では?
  - 経路を良く知っているルーターが他組織(一般にプロバイダー等の上位組織)にあれば，とりあえず，そこのルーターに全部渡してしまえば良い: デフォルトルート
  - 手作業で作成するのは
  - 自組織内の経路
  - 明示的に指定したい経路

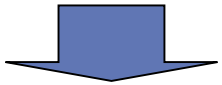
グローバルアドレスとプライベートアドレス

- IPアドレスは世界で一意でなければならない＝グローバルアドレス
  - 32ビット → 約40億ー様々なオーバヘッド
- 世界中で多数のコンピュータがIPを使い出した
  → IPアドレスが足りない！！
- 実はインターネットに接続していない(社内や組織内のLANとしてだけ利用する)コンピュータも多い

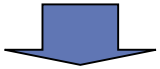
- 社内や組織内だけで利用するアドレス領域を設定＝プライベートアドレス → 内部で勝手に使って良い
  - 10.0.0.0～10.255.255.255
  - 172.16.0.0～172.31.255.255
  - 192.168.0.0～192.168.255.255
- インターネット上では使えない(ルーティングされない)

グローバルアドレスとプライベートアドレス (続き)

- インターネット接続しないつもりでLANを構築したが，やはりインターネットにつなぎたい
- インターネットに接続したいが，IPアドレスが足りない



- 社内・組織内はプライベートアドレス
- インターネット接続部分だけグローバルアドレス
  - 社内のノードからどうやってインターネットに接続する？



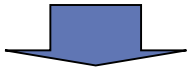
  - NAT (Network Address Translation)
  - アプリケーション・ゲートウェイ: proxy, socks, Delegate

トランスポート層：TCP，UDP

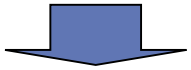
- TCP (Transmission Control Protocol)
  - 2つのノード間のデータ伝送の信頼性を保証
    - パケット交換の場合，(データのかたまりを分割した)パケットが消失したり順番が入れ替わったりする
    - データが正しく相手まで届いたか確認し，問題(消失や誤り)があれば再送信する
  - ノード間に，アプリケーションごとに，あたかも回線が張られたかのようになる
    →バーチャル・サーキット (Virtual Circuit)
- UDP (User Datagram Protocol)
  - 同じレイヤーで，信頼性が保証されないプロトコル
  - そのかわり，1対Nでデータを送ることができたり，相手の負荷にかかわらずどんどんデータを送りつけることができる

TCP: ポート番号

- IPアドレス=ノード(コンピュータ)に1個
- ひとつのノードで複数のアプリケーションが通信をする場合あり
  - Webを見ながら，電子メールを受信する



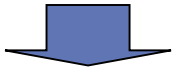
- どのアプリケーションの通信か識別が必要



- **ポート番号**(両ノードそれぞれに付与)
- ノードAのIPアドレス:ポート番号，ノードBのIPアドレス:ポート番号の組でバーチャルサーキットを識別

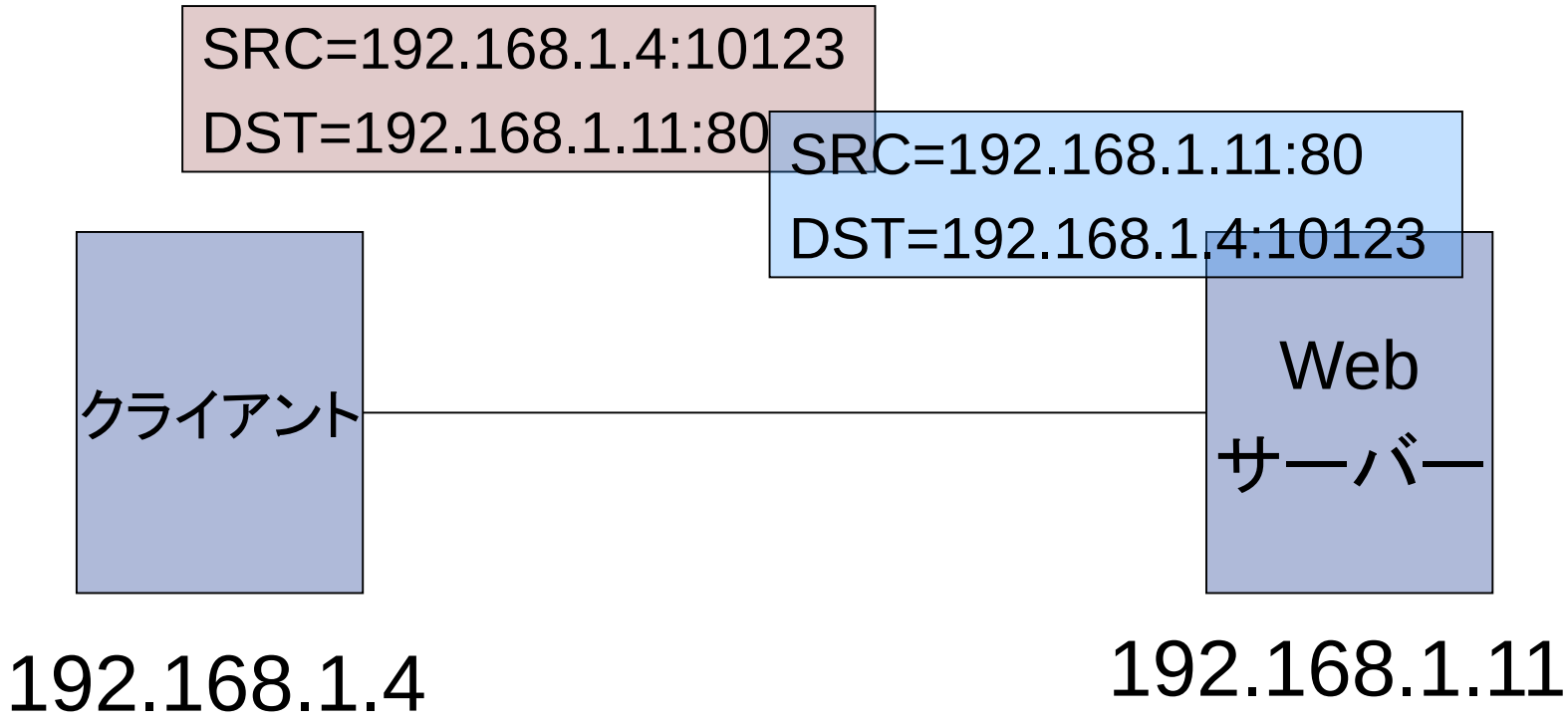
TCP: ポート番号 (続き)

- 接続を開始する際にどの(何番の)ポートを使う？



- 接続を開始する側
  - どのポートを使っているか自分でわかる
  - 空いているポートを使う (一般に10000以上の大きい番号)
- 接続を受ける側
  - 接続を要求されるポート番号によってアプリケーションを切り分けたい
  - アプリケーションによって決まったポート: Well-Known Port
    - Web=80, SMTP=25, POP=110, TELNET=23 (1024未満)

TCP: ポート番号 (続き)


SRC=192.168.1.4:10123
DST=192.168.1.11:80
SRC=192.168.1.11:80
DST=192.168.1.4:10123
クライアント
Web
サーバー
192.168.1.4
192.168.1.11

名前からIPアドレスを見つける：DNS

- ノードはIPアドレスで識別される
  - 4つの数字の並び: おぼえにくい，ノードの役割をイメージしにくい
  - 人間にとってもっとわかりやすい識別方法が欲しい!!
- 人間にとって意味のある語で名前をつけ，それにIPアドレスを対応させる
  - 好き勝手につけたら収拾がつかないので → Domain Name System
  - 名前とIPアドレスとの対応を見つける: 名前解決 (Name Resolution Service)

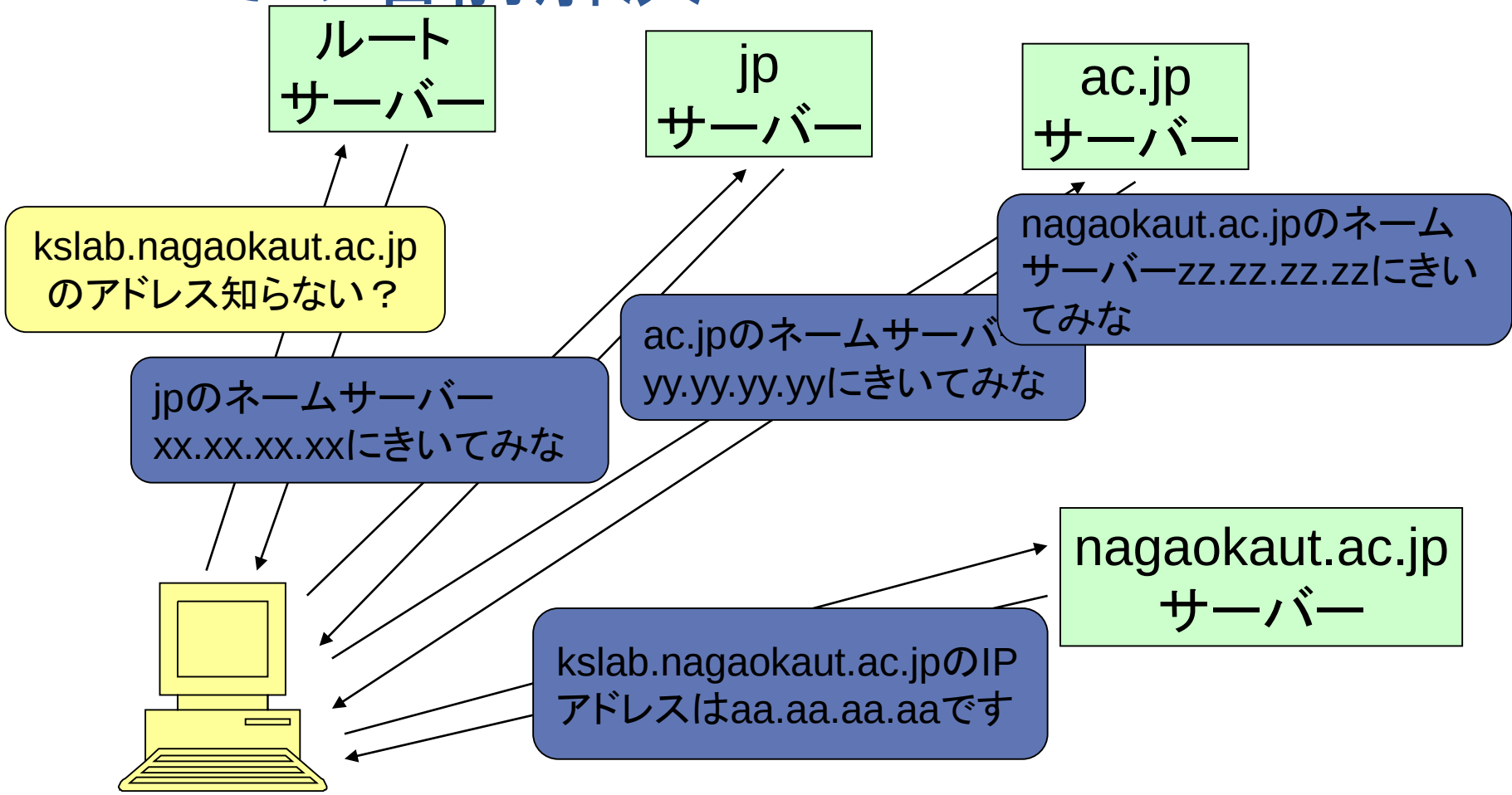
DNSの名前の構造

- 階層構造になっている

  kslab.nagaokaut.ac.jp

  Second-Level Domain　　Top-Level Domain
- Top-Level Domain (TLD)
  - ccTLD: 国や地域別のTLD (日本はjp)
  - gTLD: 国に関わらず使えるTLD (com, org, net, …)
- Second-Level Domain (SLD)
  - ccTLDに対するSLDは，TLDを持つ国や地域で決める
  - jpの場合: co(企業), ac(高等教育機関), go(政府機関), or(団体), …
- その次のレベル
  - ドメイン名を管理するレジストラに登録する(早いもの勝ち)
- さらにその次のレベル
  - 組織内で勝手に決める

DNSでの名前解決

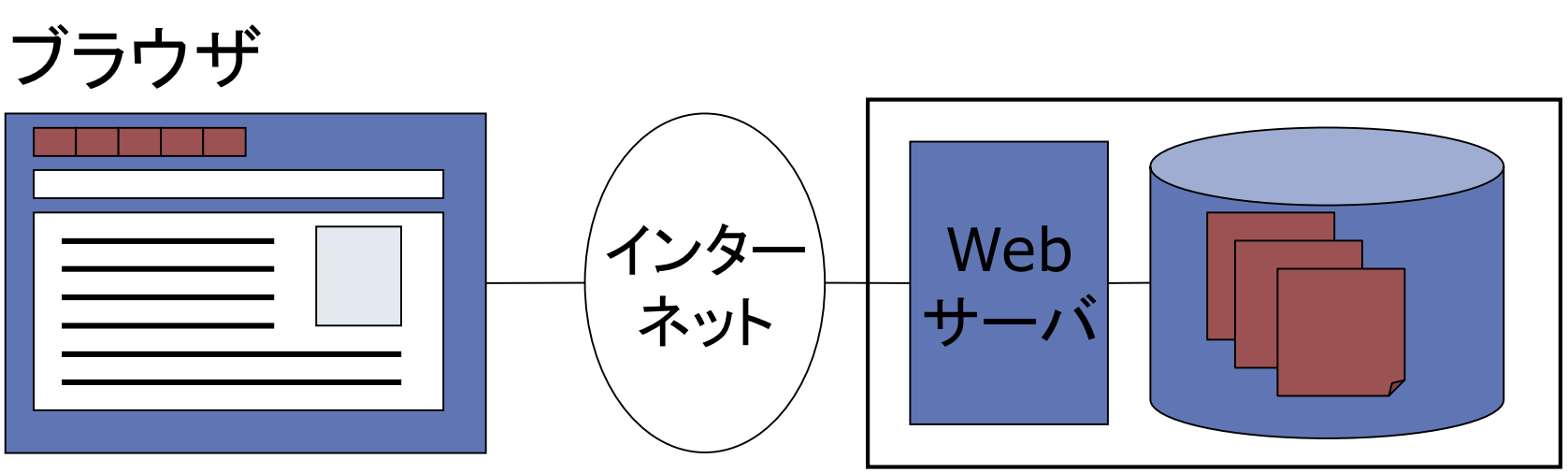

ルート
サーバー
jp
サーバー
ac.jp
サーバー
kslab.nagaokaut.ac.jp
のアドレス知らない？
nagaokaut.ac.jpのネーム
サーバーzz.zz.zz.zzにきい
てみな
ac.jpのネームサーバ
yy.yy.yy.yyにきいてみな
jpのネームサーバー
xx.xx.xx.xxにきいてみな
nagaokaut.ac.jp
サーバー
kslab.nagaokaut.ac.jpのIP
アドレスはaa.aa.aa.aaです

アプリケーション層

- アプリケーションごとに，その動作に必要なデータの形式と通信の手順を規定
  - Web
    - Webページを記述したデータを取出す手順
    - Webページの指定方法
    - Webページ記述する言語
  - 電子メール
    - 宛先の指定方法
    - ヘッダ部の記述形式
    - 本文の記述形式
  - . . .

Web（WWW; World Wide Web)

- WebサーバとWebブラウザ
  - サーバ: 文書を格納し，ブラウザの要求に応じて文書のデータを提供
  - ブラウザ:
  - サーバから得たデータを画面上に整形して表示
  - リンクをクリックしたら，リンク先の文書を取得



HTTPとHTML

- Webページが表示されるまで

1. ブラウザが要求し，サーバは文書データを提供
   - 文書データをやりとりする手順: HTTP (ハイパーテキスト・トランスファー・プロトコル)
   - ブラウザからデータをサーバに送り込むことも可能
2. サーバ来た文書データを，データ形式の取り決めに基づいて，整形して表示
   - データ形式の取り決め: HTML (ハイパーテキストマークアップランゲージ)

Webのためのプロトコル：HTTP (Hyper-Text Transfer Protocol)

```
GET /test.html HTTP/1.1
Host: kslab.nagaokaut.ac.jp

HTTP/1.1 200 OK
Date: Sun, 14 Mar 2004 01:50:48 GMT
        :
Content-Type: text/html

<html>
        :
<body>
  <h1>テストページの第1章</h1>
  <p>第1章の内容が書いてあります。</p>
        :
</body>
</html>
```

クライアントから
サーバーへの要求

サーバーからの
返答

HTML (Hyper-Text Markup Language)

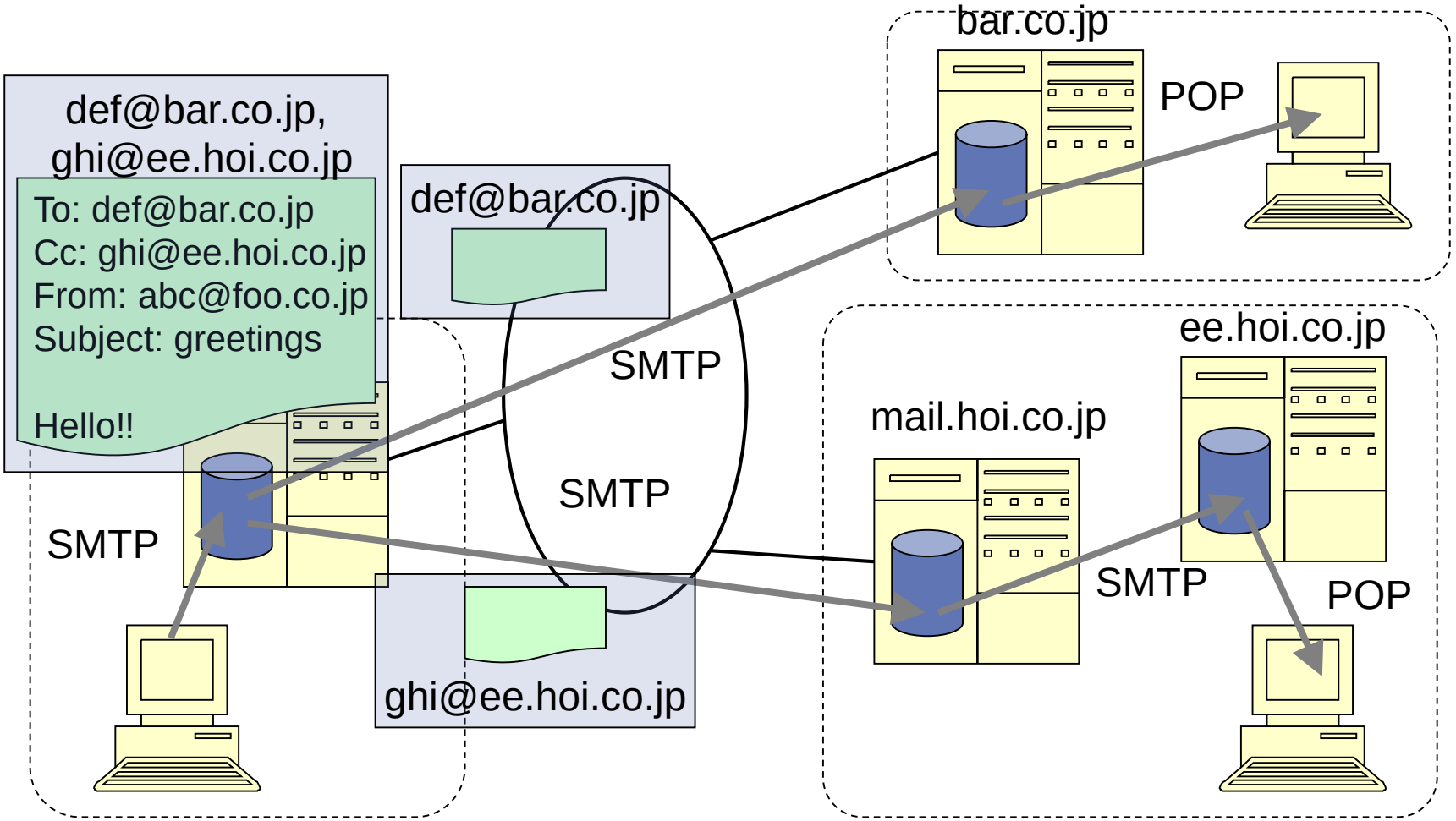
- タグ: Webページとして書く情報(文書)の構造と見た目を制御
  - 構造: 文書の標題，著者，章だて等
  - 見た目: 文字の大きさや色，箇条書きの形式

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/REC-html401/loose.dtd">
<html>
<head>
<title>テストページ</title>
</head>
<body>
<h1>テストページの第1章</h1>
<p>第1章の内容が書いてあります。</p>
```

電子メールのためのプロトコル：SMTP, POP

- SMTP (Simple Mail Transfer Protocol)
  - メールサーバー間での電子メイル配送プロトコル
  - メールをやりとりする機械が常時稼動していることが前提
  - 電子メールのヘッダに基づき配送が制御される
    - ヘッダ記述の規約はインターネット以外も配送経路に含むことができるように設計されている
- POP (Post Office Protocol)
  - メイルサーバに届いたメールを端末(PC)から読み出すためのプロトコル
  - 常時稼動しているサーバーから必要な時に読み出す

電子メールが届くまで


bar.co.jp
POP
def@bar.co.jp,
ghi@ee.hoi.co.jp
To: def@bar.co.jp
Cc: ghi@ee.hoi.co.jp
From: abc@foo.co.jp
Subject: greetings
Hello!!
def@bar.co.jp
SMTP
SMTP
SMTP
ee.hoi.co.jp
mail.hoi.co.jp
SMTP
POP
ghi@ee.hoi.co.jp

電子メールのプロトコルから言えること

- 差出人のところに書いてあるアドレスが本当の差出人とは限らない
- 宛先のところに書いてあるアドレスが本当の宛先とは限らない
- メールが確実に届くとは限らない(届いたことを確認する手段はない)
  - 届いても，受信者が読んだことを確認(開封確認)する手段はない
- メールが中継点で改竄される可能性がある
  - 改竄されたことを検出する手段はない

インターネット・プロトコルの特徴

- 「みんな正直者」が前提のプロトコル
  - 悪いことをしようという要素があった時に，それを防いだり検出する仕組みが存在しない
- 階層性
  - データがすべての階層を通るので，どの階層でも覗き見たり改竄することが可能

- 実際のシステムでは，後づけ or 上からかぶせる形で，窃用・改竄を防止しようとしている
  - SSL：Webの通信を暗号化し，途中で覗き見られたり改竄されたりを避ける
  - システム(サービス)利用のためのユーザ名/パスワードによる認証
  - 他のメディア (携帯電話のSMS)との組合せによる認証

第2部のまとめ

- インターネットは原理的に「みんなが正直者」システム
  - 悪いことを防いだり検知する仕組みが，基本部分には備わっていない

- 悪いことをしようという時に，いくらでも攻めるポイントがある

- 実際のシステムやサービスでは，後づけやアプリケーション層にかぶせる形で，セキュリティを実現

- 元が脆弱なので，それでも，攻めようがいろいろある

全体のまとめ

- インターネットは自律分散型，オープン，世界中に普及

- トップダウン型・地域型の犯罪対策では対応できない場合も多い
- 犯罪者の思考様式を理解する / 広域連携

- インターネット・プロトコルは，「みんな正直者」が前提

- 日々，インターネットを悪用した新たな犯罪手口が発明される
- 常に情報収集を / 専門家の力を借りる