

Internet Servers

March 24th ,2005
(lecture)

In This Lesson

Purpose;Preparing Build-up Server

◆ Lecture

1. Categorized Internet System

- 1-1. Client Server
- 1-2. DNS System
- 1-3. Web System
- 1-4. Mail System

◆ Practice (option)

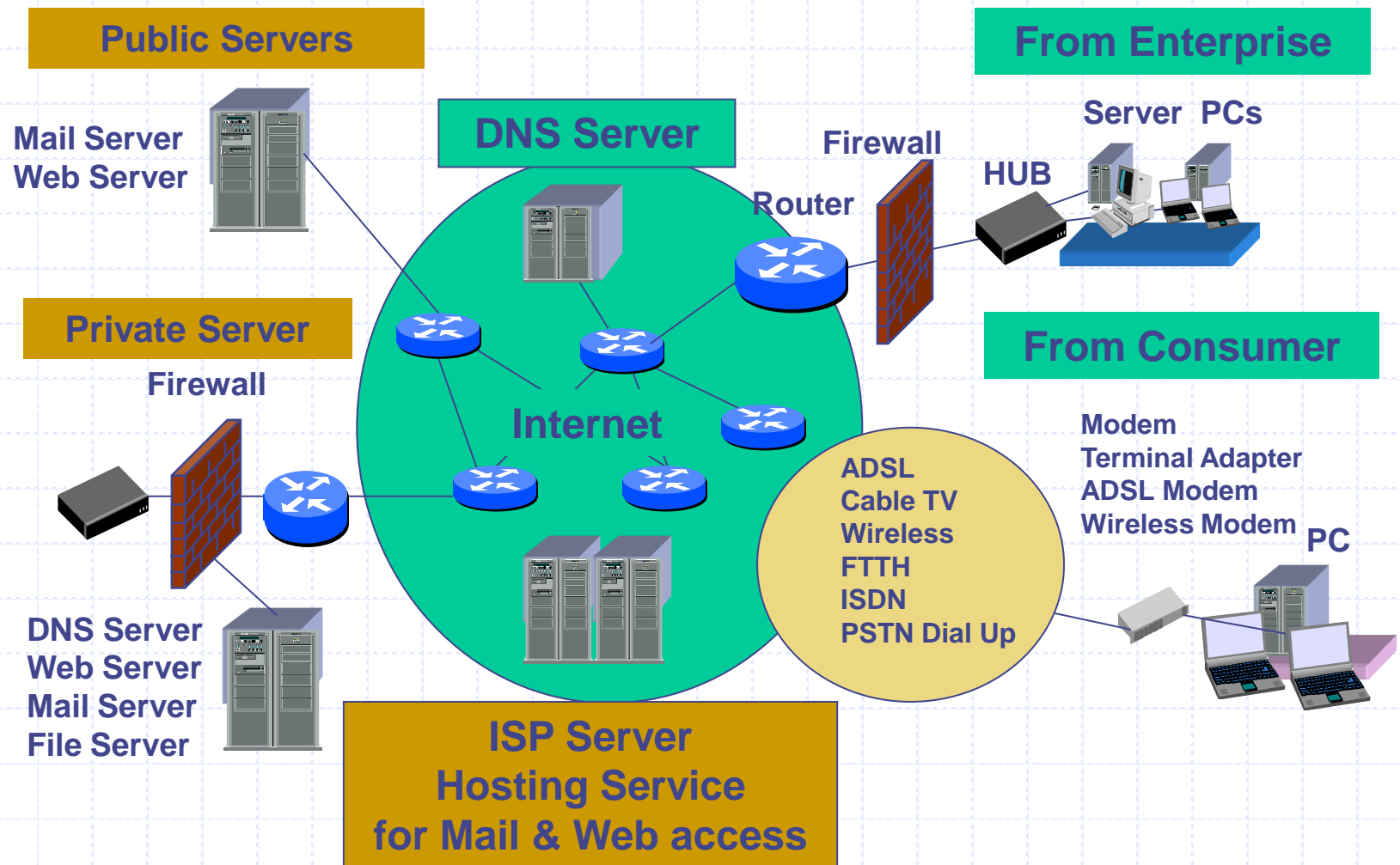
1. Server Configuration

- 1-1 DNS Server (BIND)
- 1-2 Web Server (Apache)
- 1-3 Mail Server (Sendmail)

Clients Server

- ◆ A “server” is a computer or system that provides a specific service (mail ,dns . . .) to clients
- ◆ A client is an application that runs on a personal computer or workstation and relies on a server to perform some operations.(ex. IE(www),outlook(mail))

Internet Configuration



Domain Name System(DNS)

- ◆ The domain name system is a global network of servers that translate host names like `www.hotwired.com` into numerical IP (Internet Protocol) addresses, like `204.62.131.129`, which computers on the Net use to communicate with each other.
- ◆ Without DNS, we'd all be memorizing long numbers instead of intuitive URLs or email addresses. And that wouldn't be much fun.



DNS Makes networks human friendly

Domain Name System(DNS)

◆Composition of a domain name

Domain name

www . example . co . jp

— Top Level Domain(TLD)

— Second Level Domain

— Third Level Domain

⋮

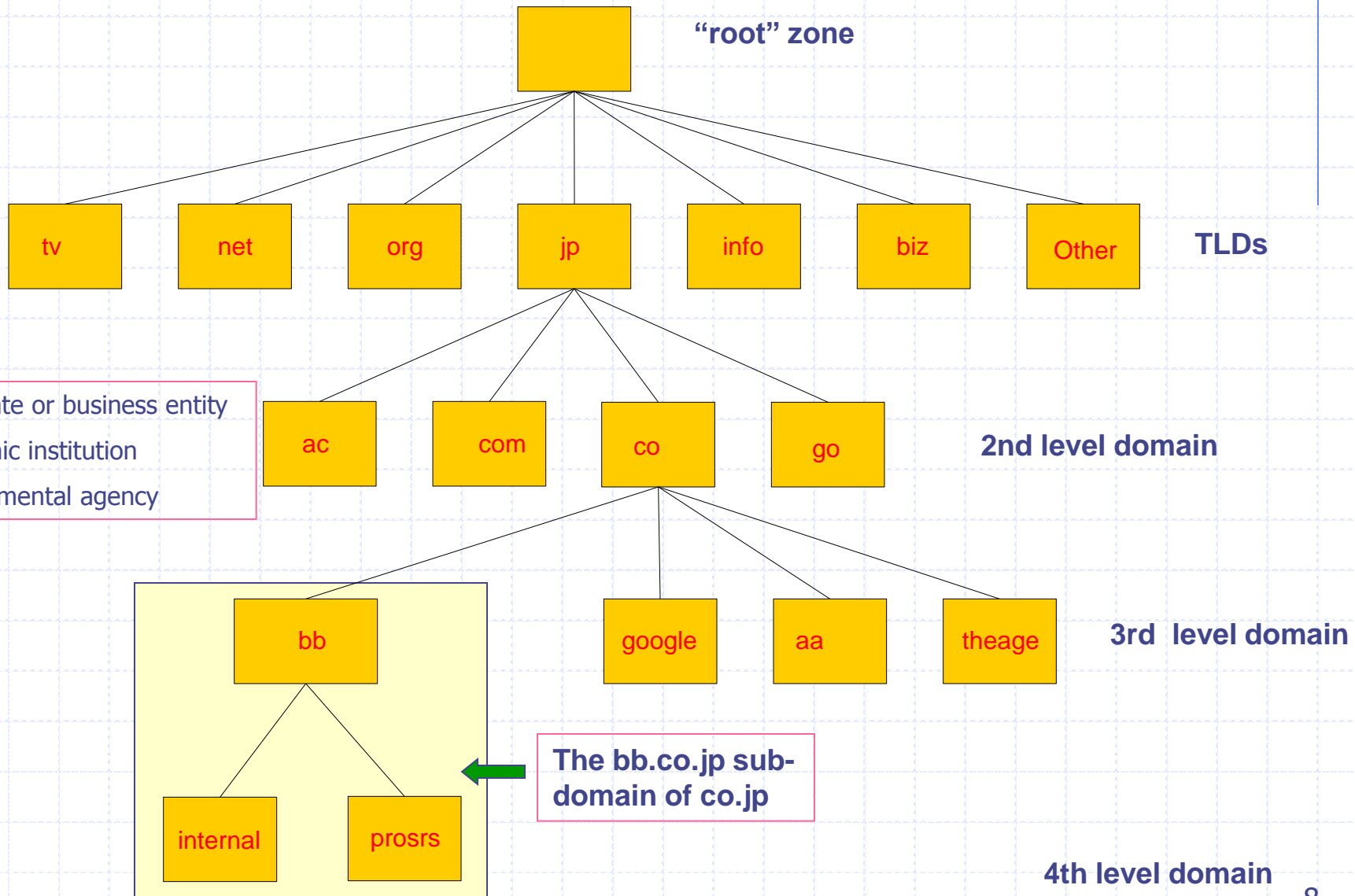
Kind of TLDs

- TLDs with two letters (such as .de, .mx, and .jp) have been established for over 240 countries and external territories and are referred to as "country-code" TLDs or "ccTLDs".

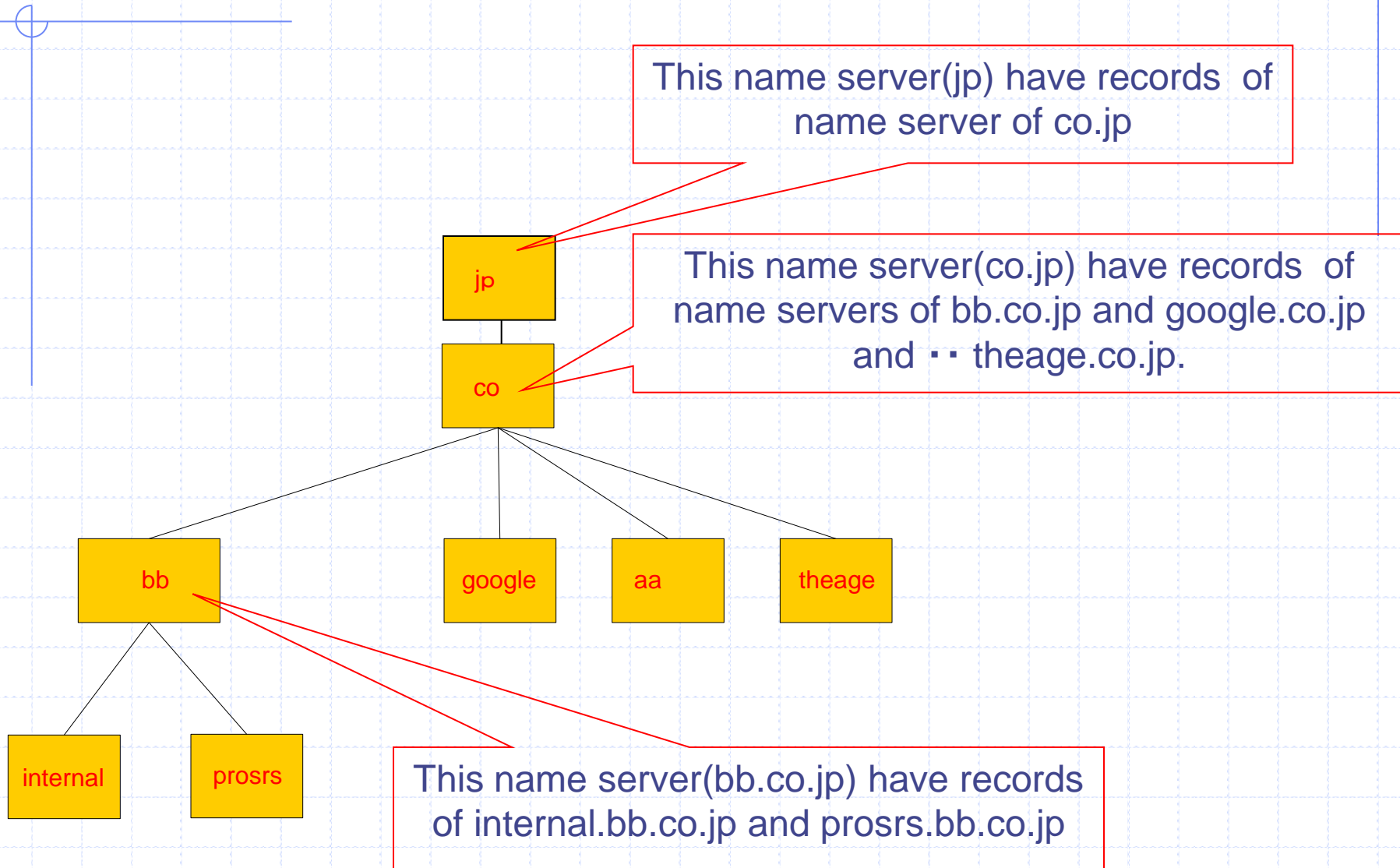
Reference <http://www.iana.org/cctld/cctld-whois.htm>

- Most TLDs with three or more characters are referred to as "generic" TLDs, or "gTLDs"(com, .edu, .gov, .int, .mil, .net, and .org.....)

DNS Structure



DNS Structure



Summary of Domain

- ◆ A domain is a sub tree of a larger tree identified by a domain name .
- ◆ Name server have records of name server of sub-domain
- ◆ the domain name system is that no single name server manage all the information . Name server manage It's domain information.It's what's known as a distributed database.

Reference site

<http://www.internic.net/>
<http://www.netsol.com/>
<http://www.whois.net/>

Server and Resolver

◆ What is a DNS server

- just a computer that's running DNS software.
- most servers are Unix machines, the most popular program is BIND (Berkeley Internet Name Domain)

◆ Components DNS server

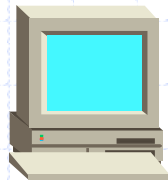
- the actual name server
 - The name server responds to browser requests by supplying name-to-address conversions.
- Resolver
 - When name server doesn't know the answer, the resolver will ask another name server for the information.

Name Resolution

- You type `http://www.google.com` into your web browser and hit enter !

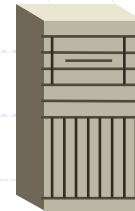
Step 1: Resolver of your PC sends a resolution request to its configured DNS Server, typically at your ISP.

resolver



Client PC

Tell me the Address of
“www.google.com”



ISP DNS Server

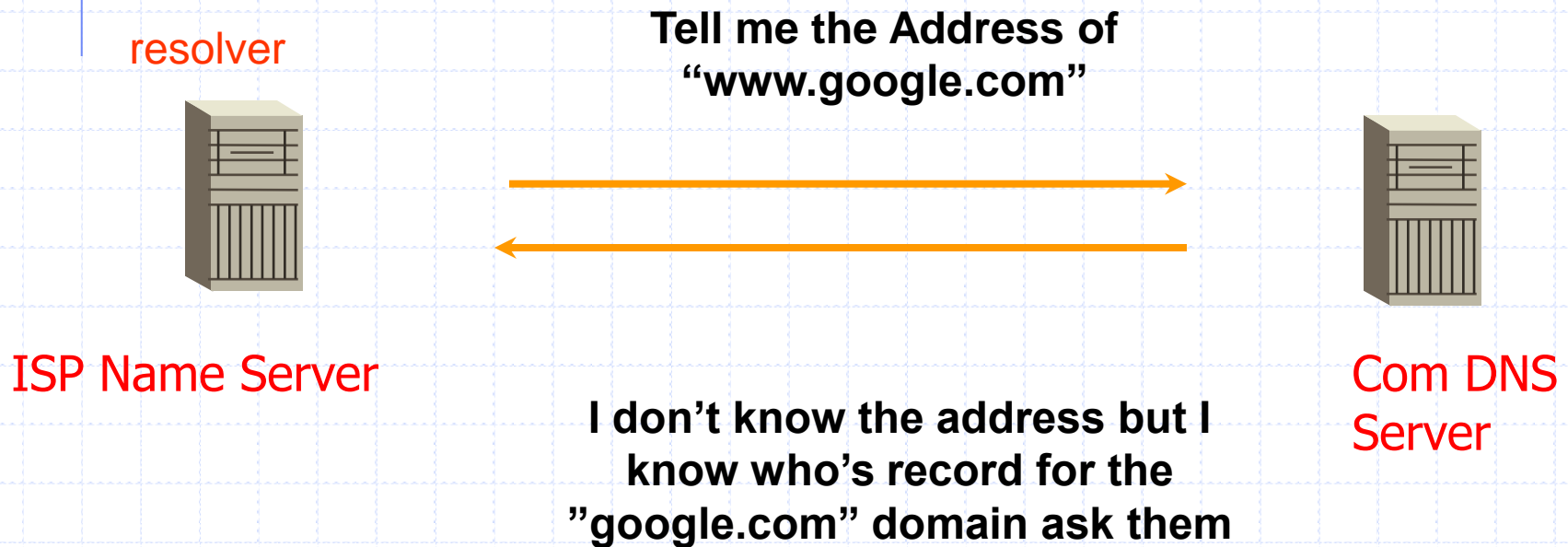
Name Resolution

Step 2: Resolver of your ISP's name server starts by asking one of the root servers .



Name Resolution

Step 3: Resolver of your ISP's name server then asks one of the "com" name servers as directed.



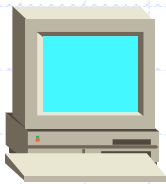
Name Resolution

Step 4: Resolver of your ISPs recursive name server then asks one of the “google.com” name servers as directed.



Name Resolution

Step 5: ISP DNS server then send the answer back to your PC. The DNS server will “remember” the answer for a period of time



Client PC



**The Address of www.google.com
is 216.239.53.99**



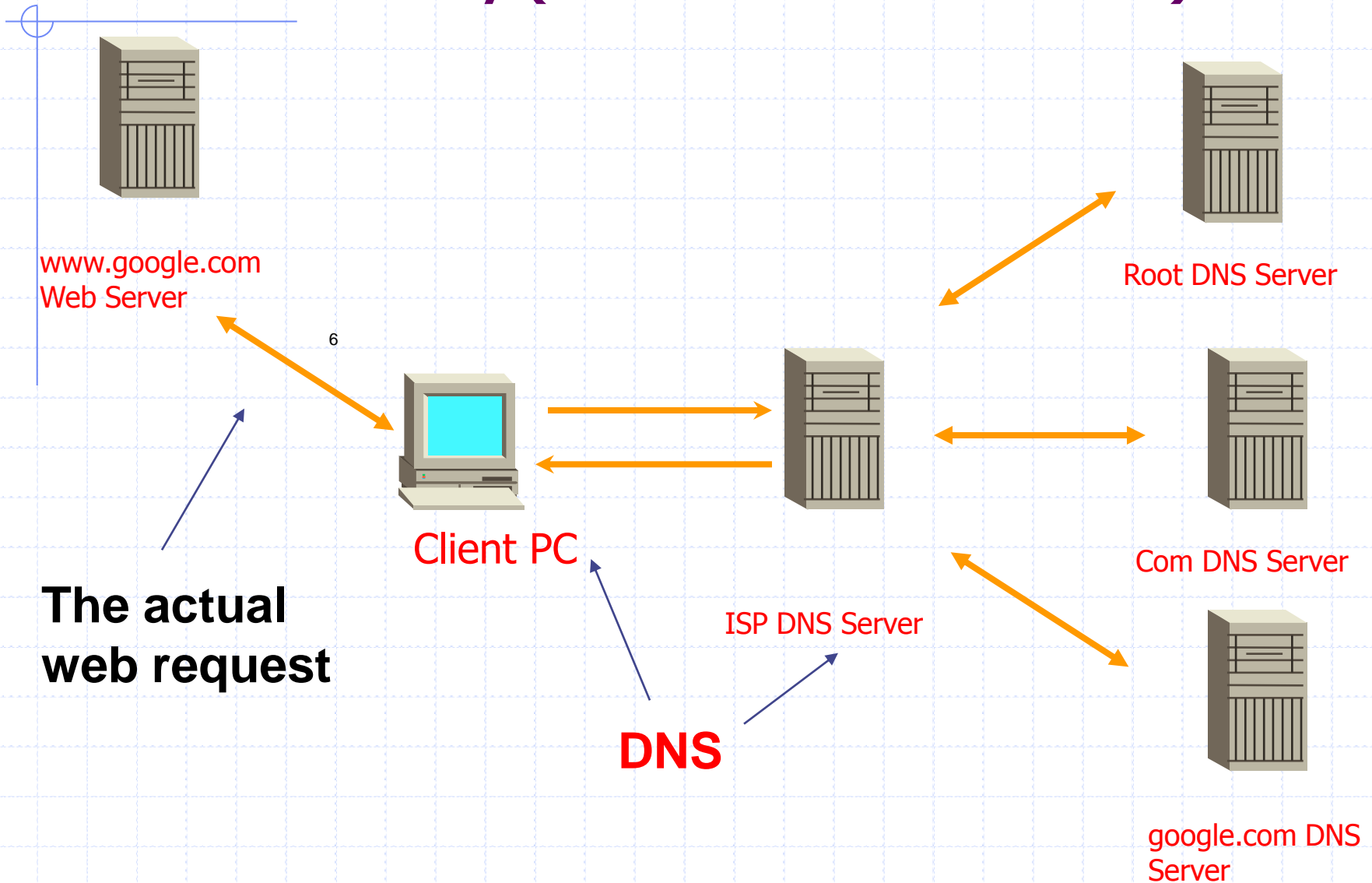
ISP DNS
Server

Name Resolution

Step 6: Your PC can then make the actual HTTP request to the web server.



Summary(Name Resolution)



Kind of Resource Record

◆ What is a Resource Record?

- A domain contains several different resource record
- Resource record analogous to files
(file have various record(exe,txt· ·))
- Resource Record classified into several types

Import Type :A, PTR,MX,NS SOA record

Kind of Resource Record

- The “A”(address) record
 - Enables a host name(computer name) to IPv4 address translation
- The PTR(Pointer) record
 - Enables IPv4 address to a host name translation
(this is used for the mail user verification)
- The MX(Mail Exchange) record
 - Specifies the name and relative preference of mail servers.
- The NS(Name Server) record
 - NS records define the name servers
- The SOA(Name Server) record
 - The SOA record contains information about configuration of name server.

Summary(DNS)

- ◆ Structure of domain name
- ◆ Name Resolution
- ◆ Kind of resource record

Example DNS Setting (BIND)

(Red Hat Linux)

General configuration file: `/etc/named.conf`

```
options {
    directory "/var/named";
    // query-source address * port 53;
};

controls {
    inet 127.0.0.1 allow { localhost; } keys { rndckey; };
};

zone "." IN {
    type hint;
    file "named.ca";
};

zone "localhost" IN {
    type master;
    file "localhost.zone";
    allow-update { none; };
};

zone "0.0.127.in-addr.arpa" IN {
    type master;
    file "named.local";
    allow-update { none; };
};
```

```
zone "ipap.blade-server.jp" {
    type master;
    file "ipap.blade-server.jp";
};

zone "64.230.210.in-addr.arpa" {
    type master;
    file "64.230.210.in-addr.arpa";
};

zone "jtec.ipap.blade-server.jp" {
    type slave;
    file "jtec.ipap.blade-server.jp";
    masters { 210.230.64.244; };
};

include "/etc/rndc.key";
```

Example DNS Setting (BIND)

(Red Hat Linux)

Forward master zone file: /var/named/ipap.blade-server.jp

```
$TTL      86400

@          IN      SOA      dns.ipap.blade-server.jp. admin.dns.ipap.blade-server.jp. (
                                2005031902 ; Serial
                                10800      ; Refresh
                                3600       ; Retry
                                604800     ; Expire
                                86400      ; Minimum
                                )

          IN      NS       dns.ipap.blade-server.jp.

          IN      MX 10    mail.ipap.blade-server.jp.

dns       IN      A        210.230.64.2
mail      IN      A        210.230.64.2
www       IN      A        210.230.64.2
ftp       IN      A        210.230.64.2
web       IN      CNAME    www

jtec      IN      NS       ns.jtec.ipap.blade-server.jp.
```

Example DNS Setting (BIND)

(Red Hat Linux)

Reverse master zone file: /var/named/64.230.210.in-addr.arpa

```
$TTL      86400
@          IN      SOA      dns.ipap.blade-server.jp. admin.dns.ipap.blade-server.jp. (
                                2005031902 ; Serial
                                10800      ; Refresh
                                3600       ; Retry
                                604800     ; Expire
                                86400      ; Minimum
                                )
          IN      NS       dns.ipap.blade-server.jp.
2         IN      PTR      dns.ipap.blade-server.jp.
```

Start / Stop

```
# service named start
```

```
# service named stop
```

Protocol

- Requests are made to the server in a specific format – a protocol
- This is called an application-layer protocol.

WWW System

◆ What is a WWW system?

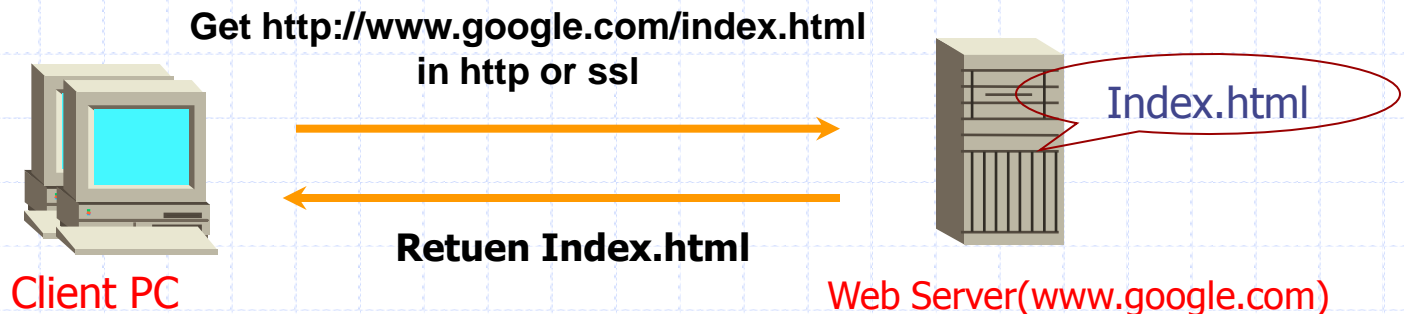
A system of Internet servers that support specially formatted documents. The documents are formatted in a markup language called HTML (*HyperText Markup Language*) that supports links to other documents, as well as graphics, audio, and video files.

◆ WWW components

- **Server** - respond with requested Web pages
- **Clients** - request and renders Web pages returned by server
- **URL** - the global address of documents and other resources on the World Wide Web
- **HTTP Protocol** - the underlying protocol used by the World Wide Web
- **SSL Protocol** - SSL is a Web protocol for establishing authenticated and encrypted sessions between Web servers and Web clients.
- **HTML** - the authoring language used to create Web page on the WWW
- **CGI** - Program is executed on the Web server in real –time

WWW structure

1. **Clients** use browser application to send **URLs** via **HTTP** or **SSL** to **servers** requesting a Web page
2. Web pages constructed using **HTML** (or other markup language) and consist of text, graphics, sounds plus embedded files
3. Servers (or caches) respond with requested Web page
 - ♦ Or with error message(ex. File not found 404 error)
4. Client's browser renders Web page returned by server
 - ♦ Page is written using Hyper Text Markup Language (HTML)
 - ♦ Displaying text, graphics and sound in browser
 - ♦ Writing data as well
5. The entire system runs over standard networking protocols (TCP/IP, DNS,...)

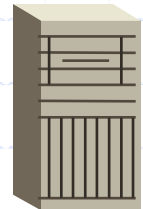


Requesting Web page with http

Clients request <http://www.dion.ne.jp/> in http



Client PC



Web Server

GET / HTTP/1.0
User-Agent: compatible; MSIE 5.01; Windows NT 5.0
Host: www.dion.ne.jp:80
Accept: */*

HTTP/1.1 200 OK
Server: Netscape-Enterprise/4.1
Date: Thu, 31 May 2001 13:06:22 GMT
Content-type: text/html
Last-modified: Thu, 31 May 2001 08:59:56 GMT
Content-length: 23804
Accept-ranges: bytes

GET /image/logo.gif HTTP/1.0
User-Agent: compatible; MSIE 5.01; Windows NT 5.0
Host: www.dion.ne.jp:80
Accept: */*

<HTML>
<HEAD>
<TITLE>Welcome to DION Home page</TITLE>

Length: 1,016 [image/gif]

HTTP Request Format

GET / HTTP/1.0

User-Agent: compatible; MSIE 5.01;

Windows NT 5.0

Host: www.dion.ne.jp:80

Accept: *.*

request-line (request request-URL HTTP-version)

headers (0 or more)

<blank line>

- ◆ First type of HTTP message: *requests*
 - Client browsers construct and send message

- ◆ Typical HTTP request:
 - GET http://www.google.com/index.html HTTP/1.0

HTTP Response Format

HTTP/1.1 200 OK

Server: Netscape-Enterprise/4.1

Date: Thu, 31 May 2001 13:06:22 GMT

Content-type: text/html

Last-modified: Thu, 31 May 2001 08:59:56 GMT

Content-length: 23804

Accept-ranges: bytes

<HTML>

<HEAD>

<TITLE>Welcome to DION Home pageII</TITLE>

status-line (HTTP-version response-code response-phrase)

headers (0 or more)

<blank line>

body

◆ Second type of HTTP message: *response*

- Web servers construct and send response messages

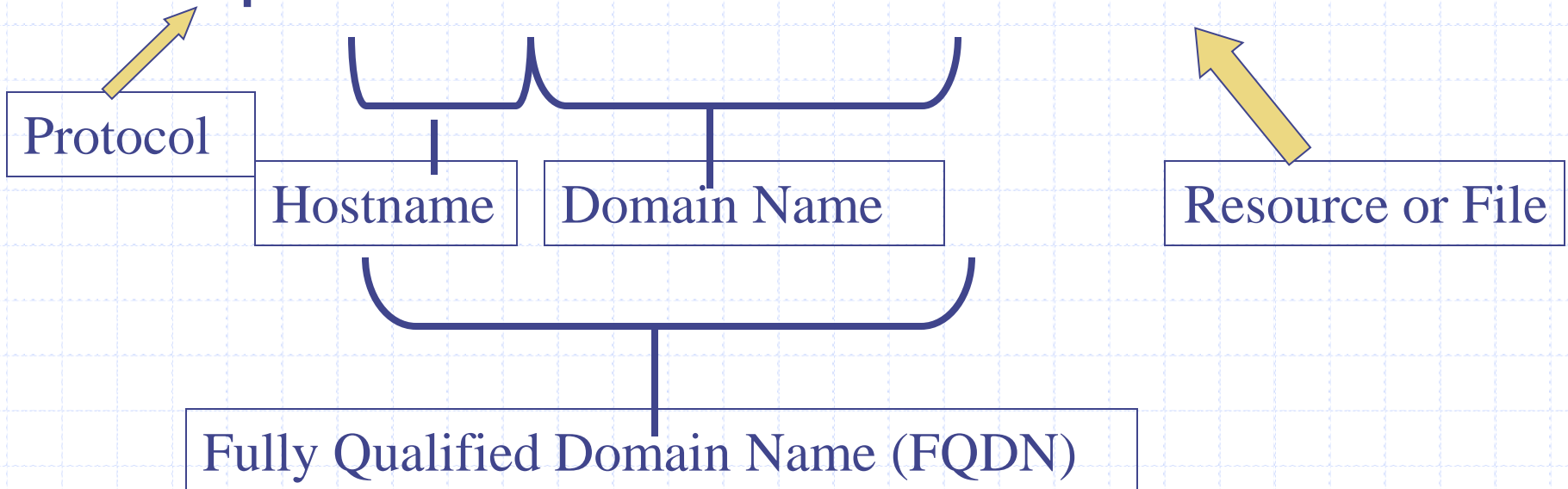
◆ Typical HTTP response:

- HTTP/1.0 200 OK

URL

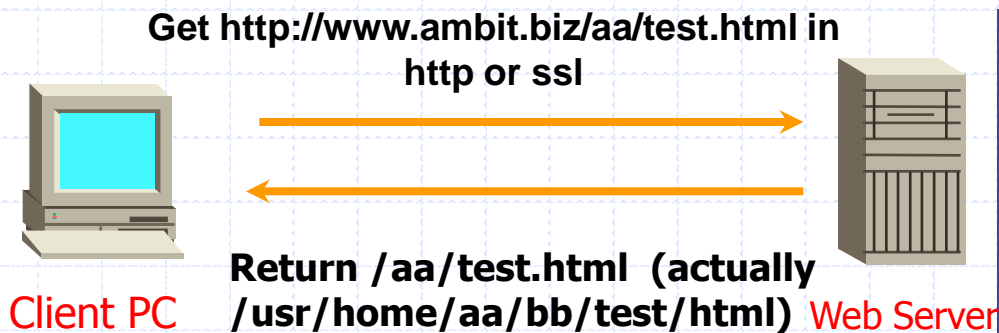
A domain name is contained in a URL. See the examples below.

`http://www.internic.net/index.html`

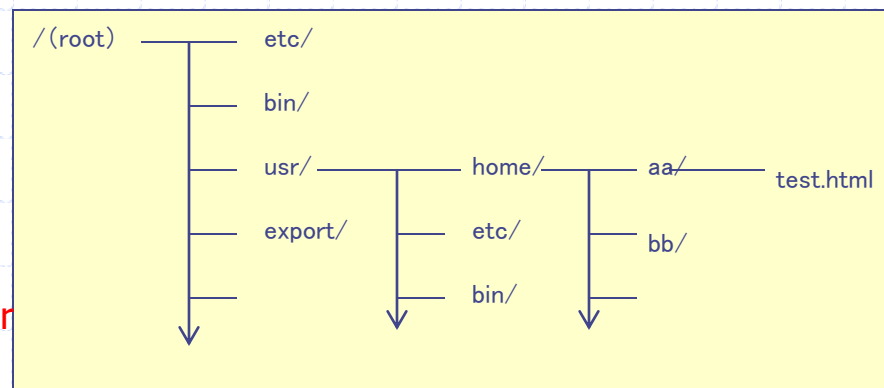


URL to File Mapping

- ◆ File name is specified in Request Message
- ◆ Server maps that name to a real file
 - Mapping can be done in whichever way server wants
 - For example, /aa/test.html is actually /usr/home/aa/bb/test.html
 - In your web server, you can choose your own



Directory structure of web server



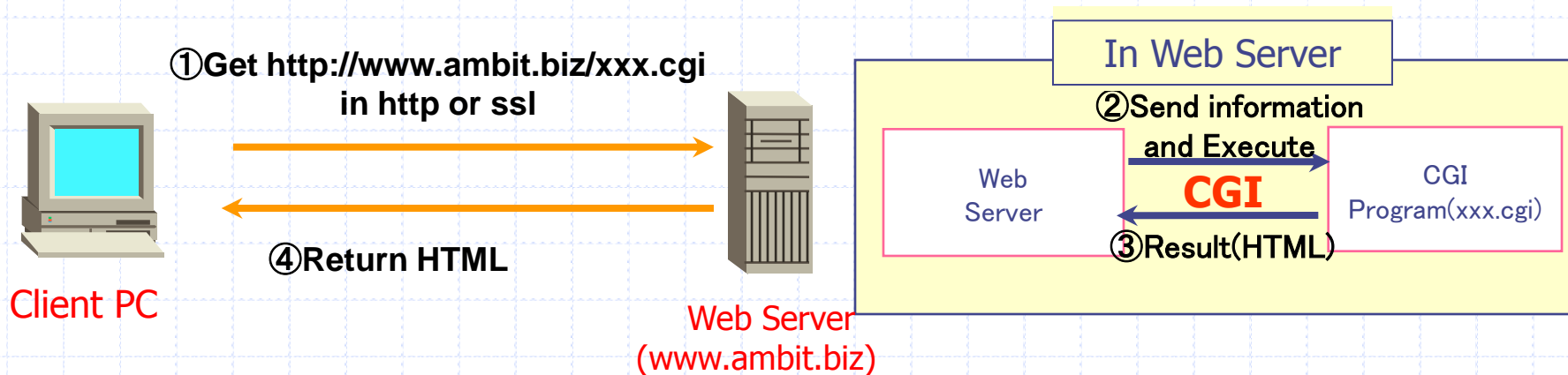
CGI(common gateway interface)

◆What is a CGI ?

- A standard interface between Web server software and other programs running on the same machine

◆Major programming language of CGI

- Perl,PHP,Java,C/C++ ...



SSL(secure socket layer)

- ◆ a protocol for transmitting private documents via the Internet.
- ◆ , URLs that require an SSL connection start with https: instead of http:.
- ◆ SSL works by encrypt data that's transferred over the SSL connection

Summary(WWW)

- ◆ Components of WWW system
- ◆ Requesting Web page with http
- ◆ SSL
- ◆ CGI

Example WWW Setting (Apache)

(Red Hat Linux)

General configuration file: /etc/httpd/conf/httpd.conf

```
...  
Listen 210.230.64.244:80  
...  
ServerAdmin webmaster@jtec.ipap.blade-server.jp  
...  
DocumentRoot "/var/www/html"  
  
<Directory "/var/www/html">  
    Options FollowSymLinks  
    AllowOverride None  
    Order allow,deny  
    Allow from all  
</Directory>
```

Apache server can be started with default configuration.

Document root: /var/www/html

Access log file: /var/log/httpd/access_log

Error log file: /var/log/httpd/error_log

Start / Stop

```
# service httpd start
```

```
# service httpd stop
```

Mail System

◆ Mail System Components

➤ **MTA(Mail Transfer Agent)**

Mail transfer agents are used to transfer messages between machines. User agents give the message to the transfer agent, who may pass it onto another transfer agent, or possibly many other transfer agents.

➤ **MUA(Mail User Agent)**

A mail user agent is an application run directly by a user. User agents are used to compose and send out-going messages as well as to display, file and print messages which have arrived in a user's mail-box.(ex. Outlook Express)

➤ **SMTP(Simple Mail Transfer Protocol)**

a protocol for sending e-mail messages between machines.

➤ **POP(post office protocol)**

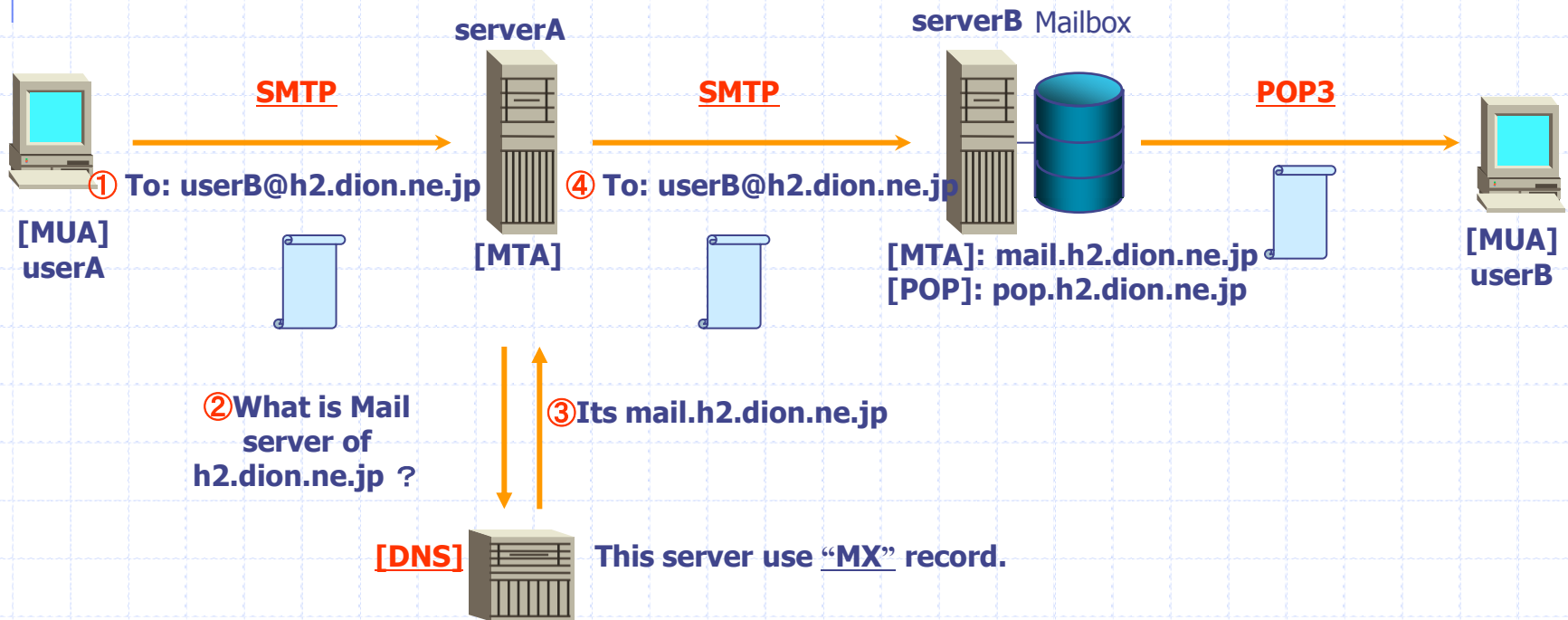
IMAP(internet Message access Protocol)

a protocol for retrieve e-mail from a mail server.

Sending Mail With SMTP

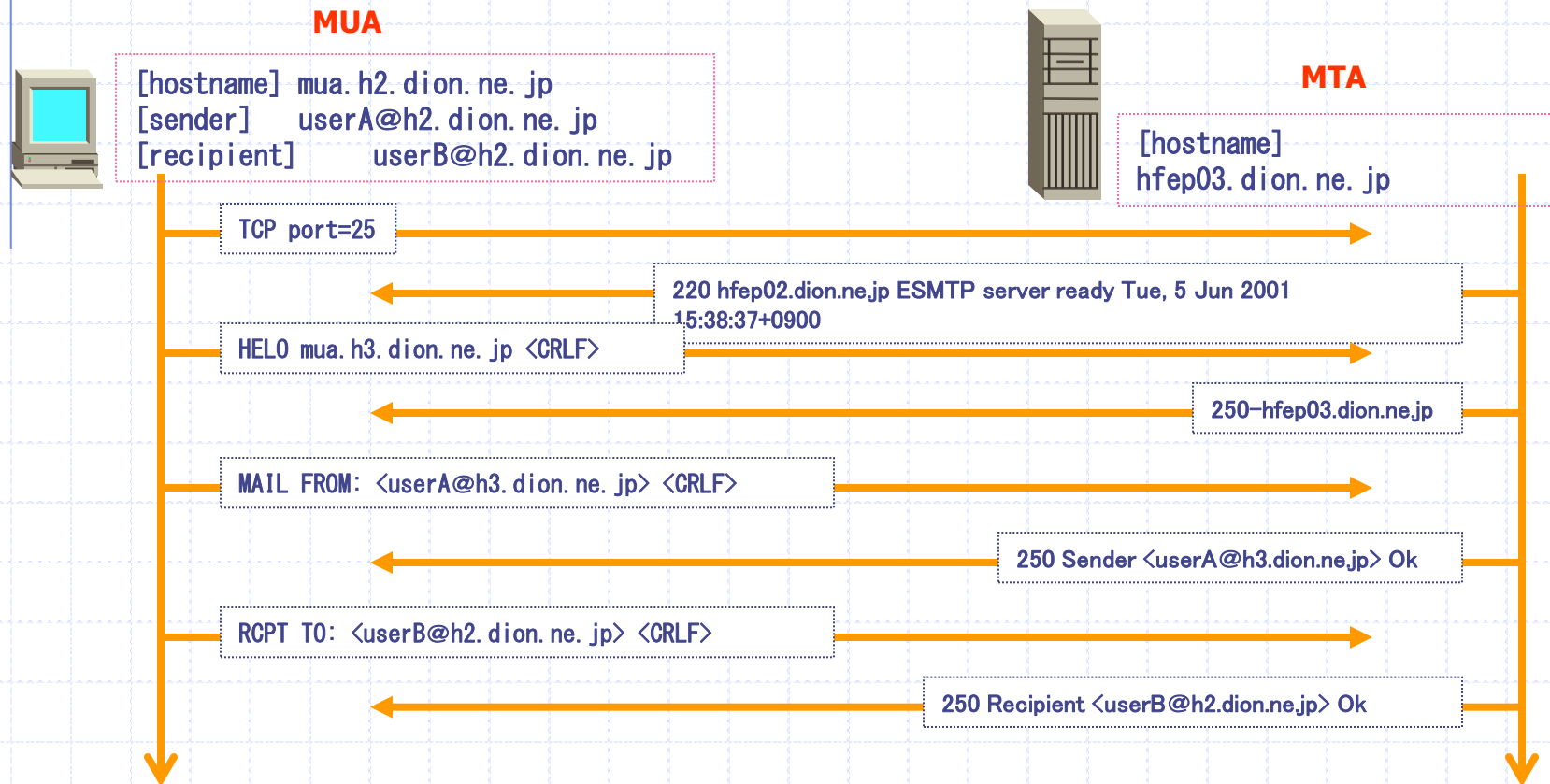
◆ userA send messages to userB@h2.dion.ne.jp

- ① MUA(of userA) transmits message to default SMTP server(serverA)
- ②③ ServerA Uses DNS to get recipient-SMTP IP address
- ④ Sender-SMTP server(serverA) transmits to recipient-SMTP server(serverB)



Sending Mail With SMTP

◆ Communication between MUA(userA) and MTA(1/2)

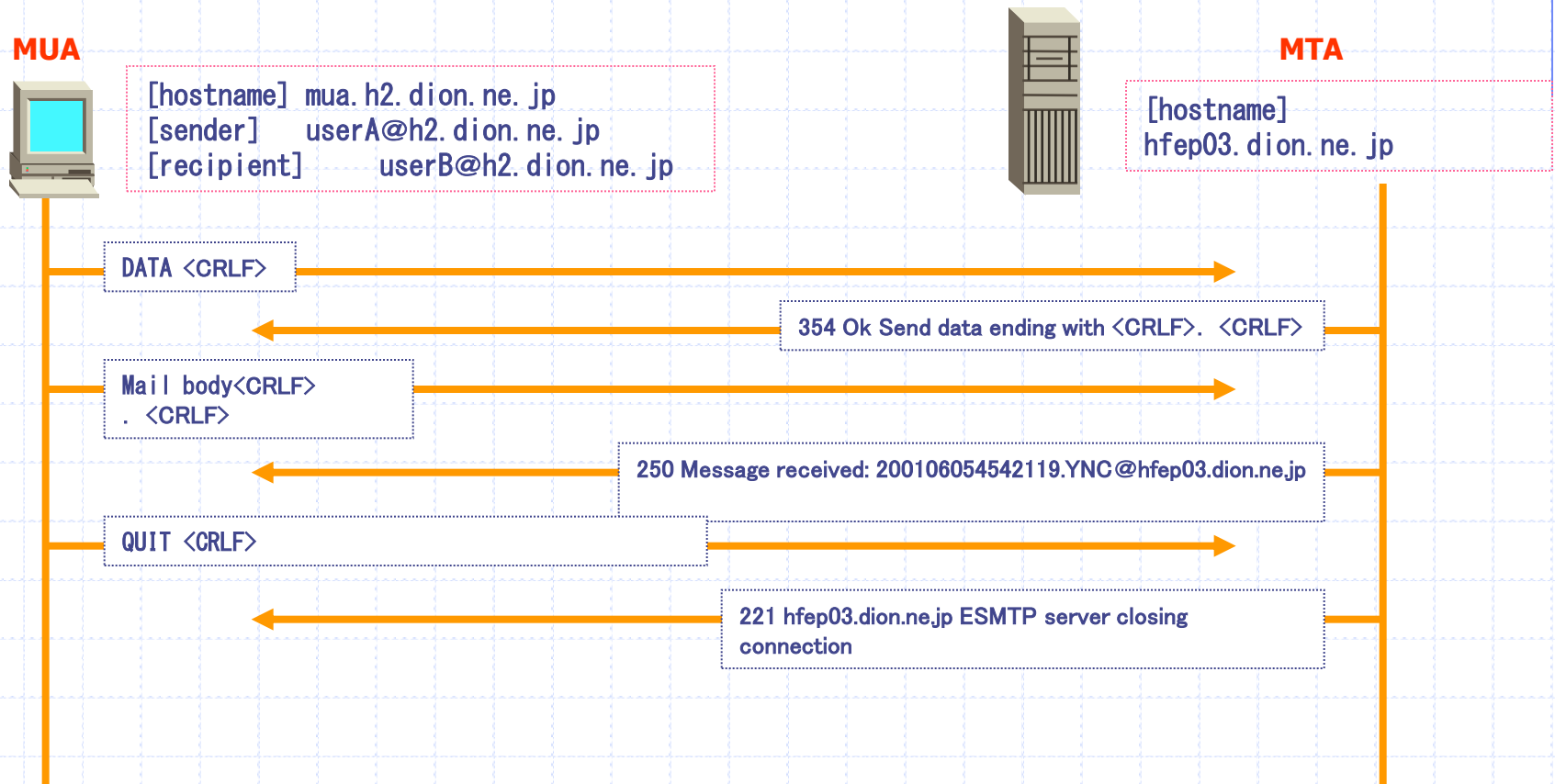


reference

<http://www.rfc-editor.org/rfcsearch.html>

Sending Mail With SMTP

◆ Communication between MUA(userA) and MTA(2/2)



Sending Mail With SMTP

- ◆ Important parts of that session are the Mail From: and the Rcpt To:
- ◆ Everything after the DATA is treated as the message.
- ◆ Server adds some information about date/processing time of message (ex. Received)
- ◆ Mail clients(ex.Outlook) adds some information about detail information of message.
(ex. Date, x-mailer)

Components of e-mail

Envelop

- SMTP server use this data to send mail

MAIL FROM: userA@h2.dion.ne.jp
RCPT TO: userB@kddi.com

DATA

From: "userA" <userA@h2.dion.ne.jp>
Subject: TEST MAIL
To: userB@kddi.com

Header

- Mail client read

test
userA@kddi.com

body

Mail Header

Return-Path: <aaa@kddi.com>

Received: from usjk1004.kddi.com (usjk1004.kddi.com [211.4.169.26])
by vodka.app.kddi.ne.jp (Postfix) with ESMTP id ss
for <aaa@app.kddi.ne.jp>; Tue, 31 Aug 2004 19:57:25 +0900 (JST)

Received: from kddiaaa (unknown [202.239.183.242])
by vodka.app.kddi.ne.jp (Postfix) with SMTP id ss
for <aaa@kddi.com>; Tue, 31 Aug 2004 19:57:23 +0900 (JST)

Message-ID: <024b01c48f49\$4b9f4450\$0a991aac@kddiaaa>

From: "aaa" <aaa@kddi.com>

To: <aaa@kddi.com>

Subject: test

Date: Tue, 31 Aug 2004 19:57:23 +0900

Content-Type: text/plain;
charset="iso-2022-jp"

Content-Transfer-Encoding: 7bit

X-Mailer: Microsoft Outlook Express 6.00.2800.1437

test

**SMTP server record path
information at
“Received” and “Mail
From:” of envelop at
“Return-Path”**

Mail-Header(example)

Field-name: Field-body (standard)

From: Sender's address

Sender: Clarify a sender when sender's address is not obvious

To: Mailing address

Cc: Carbon copy

Reply-To: Return address

Message-ID: ID to be inherent

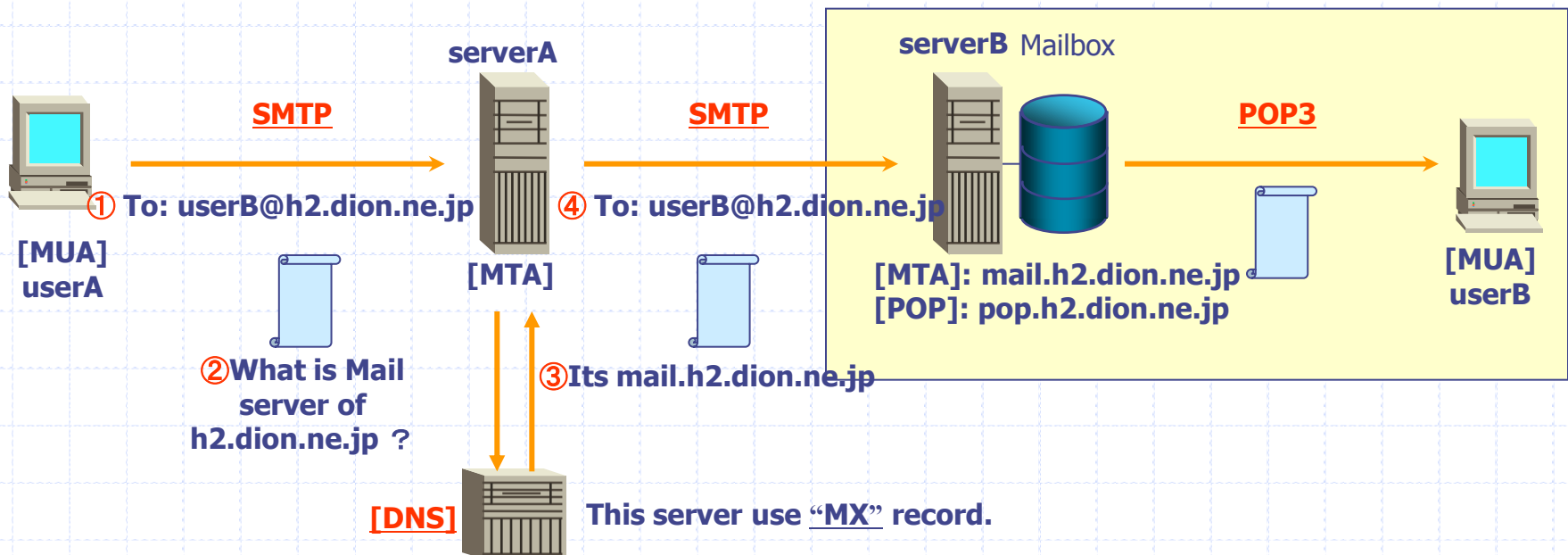
Subject: Title

Date: Time when the mail is sent

Return-Path: Return address for an error mail

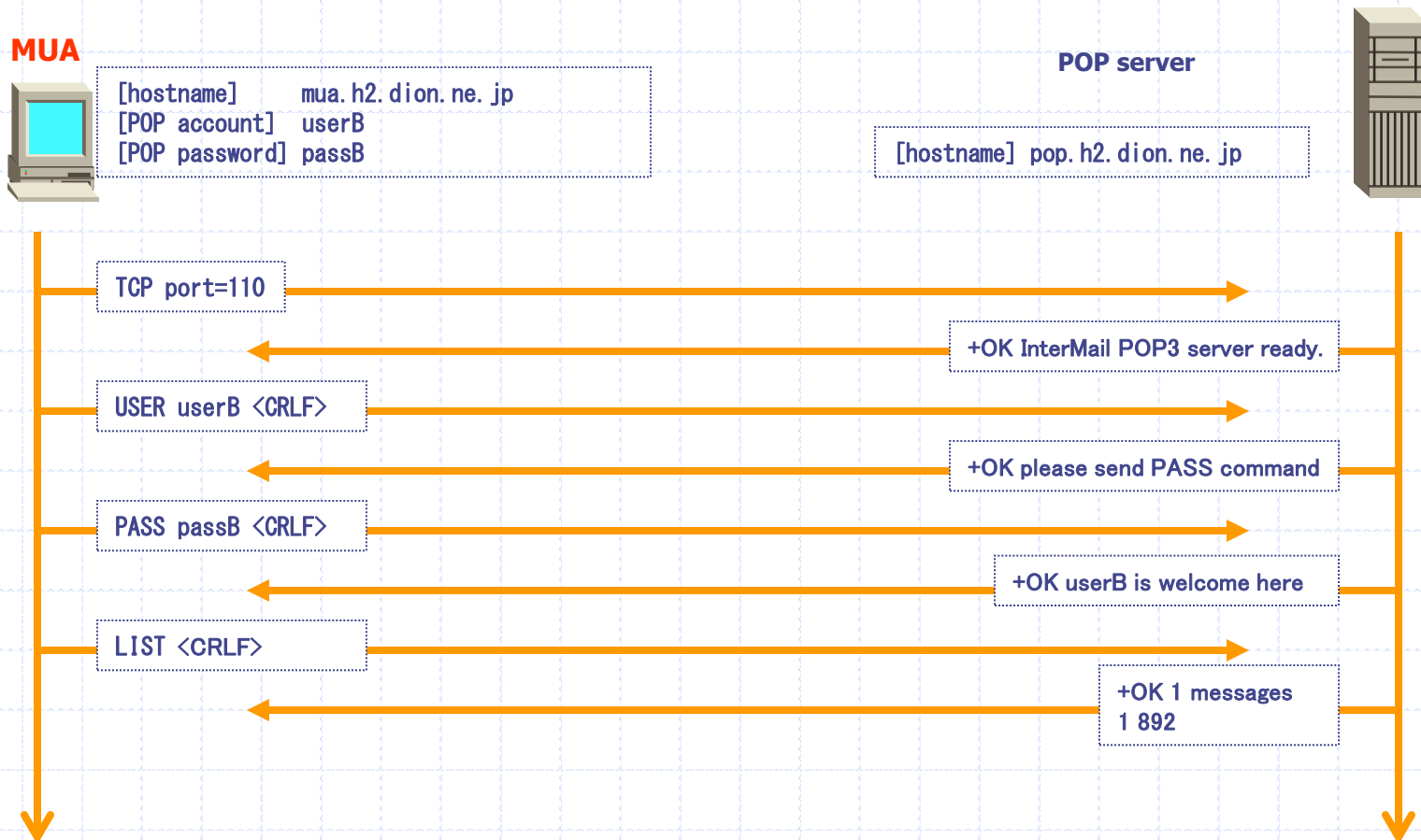
Retrieving Mail With POP

- ◆ userB retrieving Mail from POP server(pop.h2.dion.ne.jp)
 - ◆ POP3 Client occasionally connects to POP server(pop.h2.dion.ne.jp)
 - Downloads copies of messages
 - Marks as deleted or leaves them there
 - ◆ Leaving them allows retrieval from a second client
 - ◆ Client keeps track of messages already received



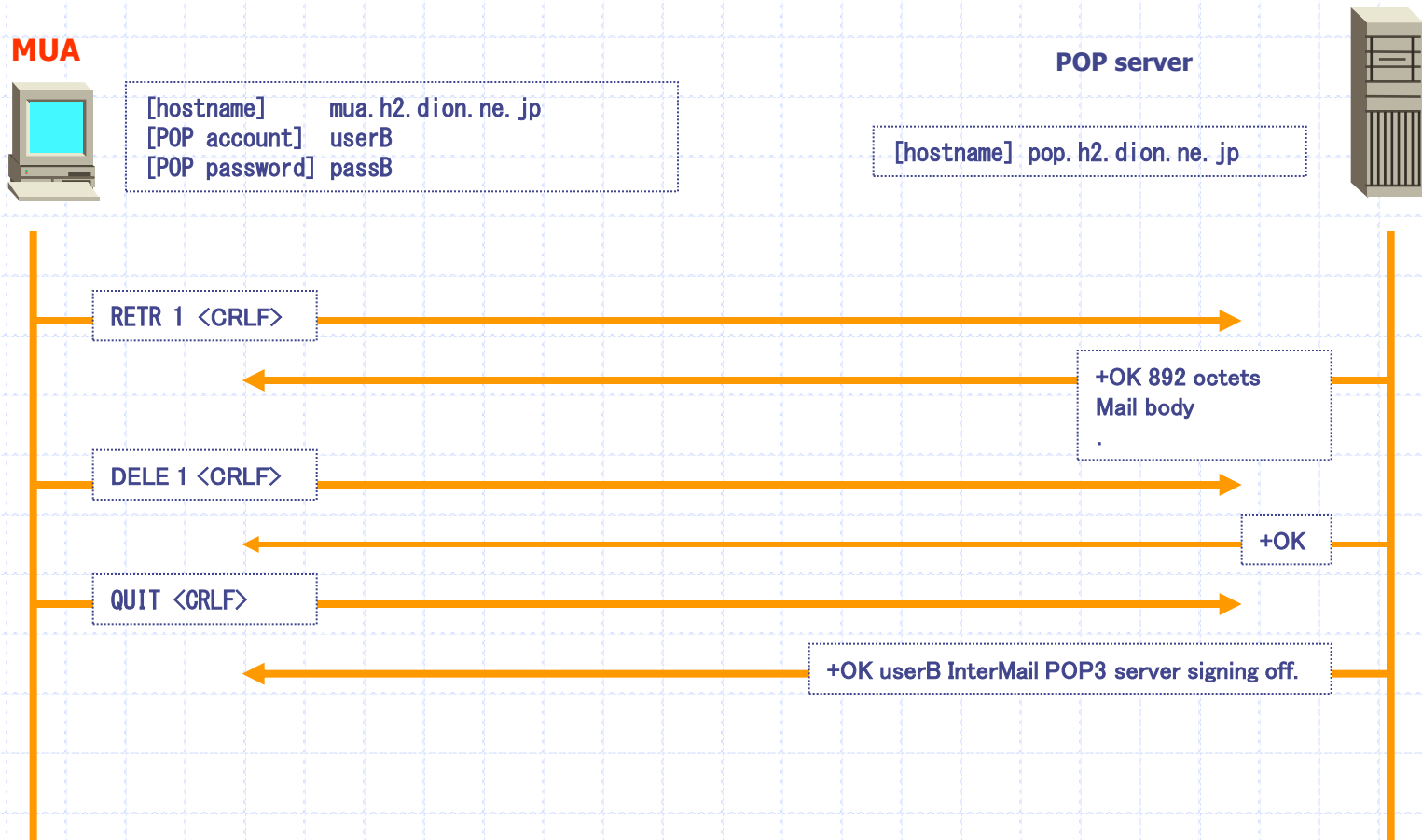
Retrieving Mail With POP

◆ Communication between MUA(userA) and POP server(1/2)



Retrieving Mail With POP

◆ Communication between MUA(userA) and POP server(1/2)



SPAM

◆ Why SMTP server can't block SPAM mail ?

- SMTP protocol can't check sender
- Only a effective check is to make sure domain exists

→ Because There are no user authentication, Faked user send SPAM mail.

◆ Seems to be about 30% to 60% of mail traffic, and increasing

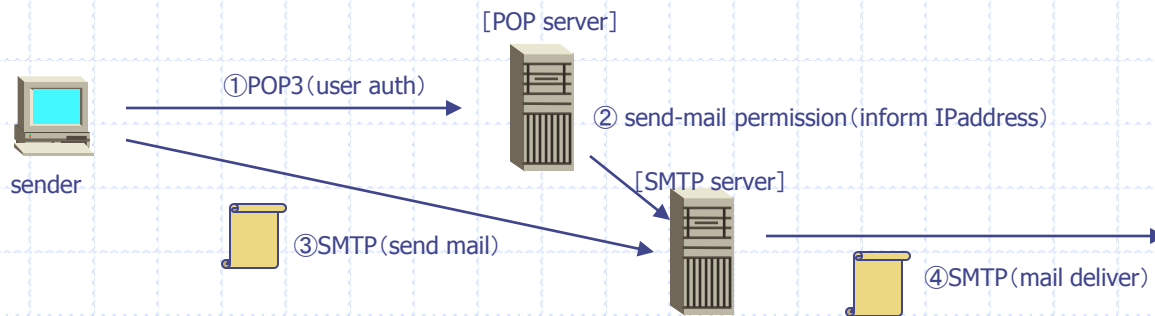
Reference site <http://www.brightmail.com/spamstats.html>

◆ Mail deliver is delayed because of SPAM

Anti-SPAM Measures

◆POP before SMTP

When POP access is done, server approves SMTP connection



◆Open Relay Server Data Base

In particular it has some defenses against spamming as the mail bombs and unsolicited junk mail in the form of options for refusing messages from particular hosts, networks, or senders.

Relay Server Data Base <http://www.ordb.com> <http://spamcop.net/>

<http://dmoz.org/Computers/Internet/Abuse/Spam/Blacklists/>

Summary

- ◆ MUA,MTA and their role
- ◆ Sending Mail with SMTP
- ◆ Retrieving Mail with POP
- ◆ SPAM and Anti-spam measures

Example Mail Setting (Sendmail)

(Red Hat Linux)

General configuration file: /etc/mail/sendmail.mc

```
...  
dnl # DAEMON_OPTIONS(`Port=smtp,Addr=127.0.0.1, Name=MTA')dnl  
...
```

Sendmail server can be started with default configuration.

Type next command and “sendmail.cf” file changes.

```
# make sendmail.cf
```

Access configuration file: /etc/mail/access

```
localhost.localdomain    RELAY  
localhost                RELAY  
127.0.0.1                RELAY  
xxx.xxx.xxx.xxx          RELAY
```

Sendmail server allows relaying from the hosts which are written in this file.

Example Mail Setting (Sendmail)

(Red Hat Linux)

Automatically generated configuration file: /etc/mail/sendmail.cf

Log file: /var/log/maillog

Mail queue: /var/spool/mail/<username>

Start / Stop

```
# service sendmail start
```

```
# service sendmail stop
```