Web

Computer System and Network Administration

Department of Computer Science & Information Engineering
National Cheng Kung University
2016 Fall

Outline

- Web hosting
  - Basics
  - Client-Server architecture
  - HTTP protocol
  - Static vs. dynamic pages
  - Virtual hosts
- Proxy
  - Forward proxy
  - Reverse proxy
  - squid

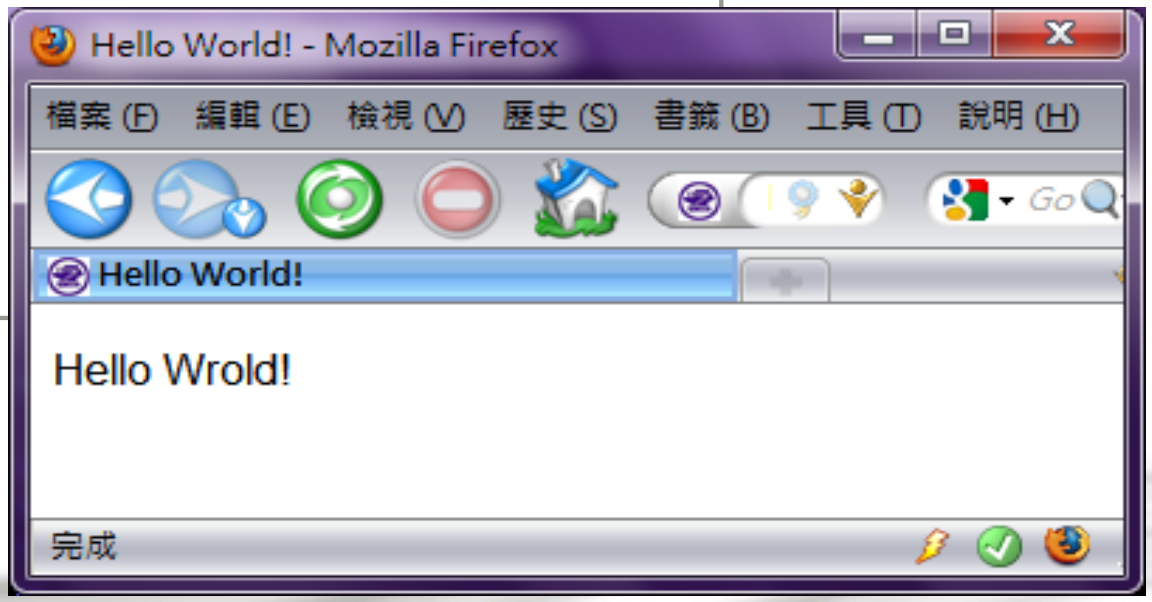
Web Hosting – Basics (1)

- Three major techniques in WWW (World Wide Web) System
  - HTML
  - HTTP
  - URL
- HTML – HyperText Markup Language
  - Providing a means to describe the structure of text-based information in a document.
  - The original HTML is created by Tim Berners-Lee.
  - Published in 1993 by the IETF as a formal "application" of SGML (with an SGML Document Type Definition defining the grammar).
  - The HTML specifications are maintained by the World Wide Web Consortium (W3C).
    - http://www.w3.org/

Web Hosting – Basics (2)

- HTML (2)
  - Mark-up the text and define presentation effect by HTML Tags.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN">
<html>
	<head>
	<title>Hello World!</title>
	</head>
	<body>
		<p>Hello Wrold!</p>
	</body>
</html>
```

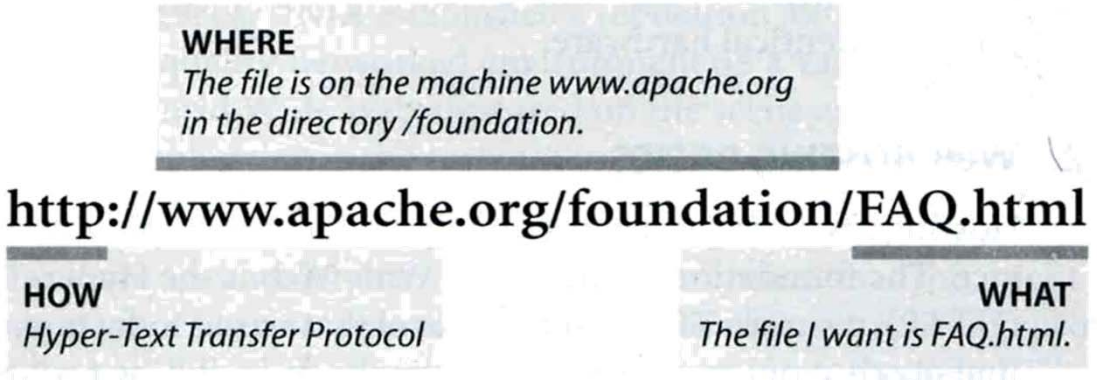


Web Hosting – Basics (3)

- HTTP – Hyper-Text Transfer Protocol
  - A TCP-based protocol
  - Communication method between client and server. All browsers and web servers have to follow this standard.
  - Originally designed to transmit HTML pages.
  - Now it is used to format, transmit, and link documents of various media types
    - Text, picture, sound, animation, video, …
  - HTTPS – secured version.

Web Hosting – Basics (4)

- URL – Uniform Resource Locator
  - Describe how to access an object shared on the Internet (RFC 1738)
  - Format
    - Protocol :// [[username [:password] @] hostname [:port]] [/directory] [/filename]



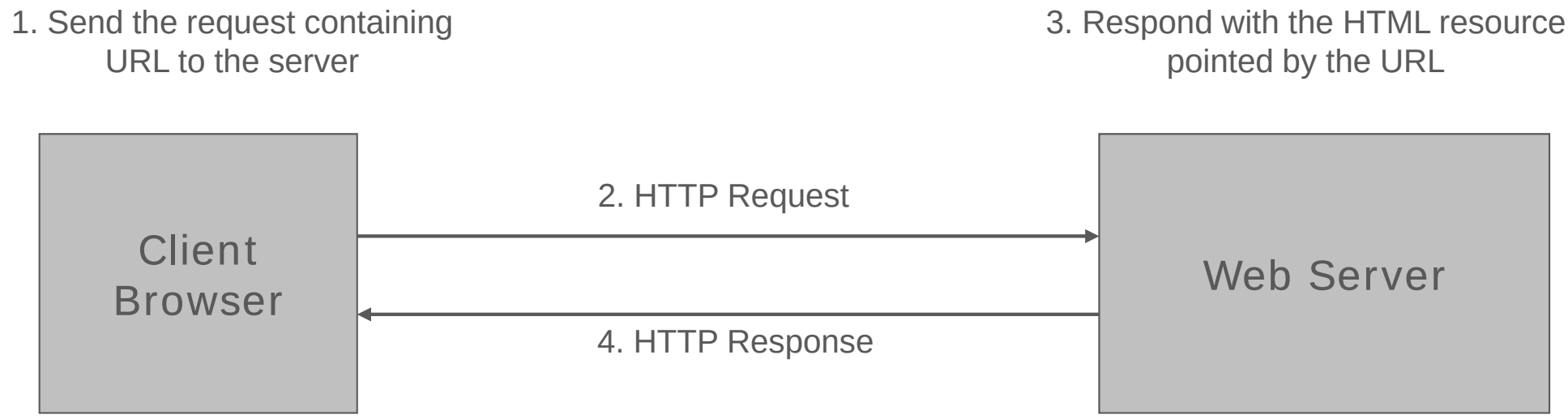
Web Hosting – Basics (5)

- URL Protocols

Proto	What it does	Example
http	Accesses a remote file via HTTP	http://imslab.org/
https	Accesses a remote file via HTTP/SSL	https://imslab.org/
ftp	Accesses a remote file via FTP	ftp://imslab.org/
file	Access a local file	file:///home/tsaimh/.tcshrc
mailto	Sends mail	mailto:tsaimh@csie.ncku.edu.tw
news	Accesses Usenet newsgroups	news:tw.bbs.comp.386bsd

Web Hosting – Client-Server Architecture (1)

- Client-server architecture
  - Web Client: Request certain page using URL
  - Web Server: Answer HTTP request



Web Hosting – Client-Server Architecture (2)

- Using “telnet” to retrieve data from web server

```
 > telnet imslab.org 80
Trying 140.116.82.246...
Connected to IMSLAB.org.
Escape character is '^]'.
GET /~tsaimh/hello.html http/1.0

HTTP/1.1 200 OK
Date: Thu, 20 Dec 2012 02:37:01 GMT
Server: Apache/2.2.21 (FreeBSD) mod_ssl/2.2.21 OpenSSL/0.9.8q DAV/2
PHP/5.3.8 with Suhosin-Patch
Last-Modified: Thu, 20 Dec 2012 02:35:37 GMT
ETag: "34d9c5d-c6-4d13f96069040"
Accept-Ranges: bytes
Content-Length: 198
Connection: close
Content-Type: text/html

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN">
<html>
     <head>
     <title>Hello World!</title>
     </head>
     <body>
          <p>Hello Wrold!</p>
     </body>
</html>
```

Web Hosting – The HTTP Protocol (1)

- HTTP: Hypertext Transfer Protocol
  - RFCs: (HTTP 1.1)

    http://www.faqs.org/rfcs/rfc2068.html

    http://www.faqs.org/rfcs/rfc2616.html (Updated Version)
  - Useful Reference: http://jmarshall.com/easy/http/
  - A network protocol used to deliver virtually all files and other data on the World Wide Web.
    - HTML files, image files, query results, or anything else.
  - Client-Server Architecture
    - A browser is an HTTP client because it sends requests to an HTTP server (Web server), which then sends responses back to the client.

Web Hosting – The HTTP Protocol (2)

- Clients:
  ※ Send Requests to Servers
  - Action "path or URL" Protocol
    - Actions: GET, POST, HEAD
    - Ex. GET /index.php HTTP/1.1
  - Headers
    - Header_Name: value
    - Ex. Host: imslab.org
  - (blank line)
  - Data ...

- Servers:
  ※ Respond to the clinets
  - Status:
    - 200: OK
    - 403: Forbidden
    - 404: Not Found
    - 426: Upgrade Required
    - Ex. HTTP/1.1 200 OK
  - Headers
    - Same as clients
    - Ex. Content-Type: text/html
  - (blank line)
  - Data...

Web Hosting – The HTTP Protocol (3)

- Get v.s. Post (client side)
  - Get:
    - Parameters in URL

      GET /get.php?a=1&b=3 HTTP/1.1
    - *No data content*
    - Corresponding in HTML files
      - Link URL: http://imslab.org/get.php?a=1&b=3
      - Using Form:

      <form method= "GET" action= "get.php" > … </form>
  - Post:
    - Parameters in Data Content

      POST /post.php HTTP/1.1
    - Corresponding in HTML files
      - Using Form:

      <form method= "POST" action= "post.php" > … </form>

Web Hosting – The HTTP Protocol (4)

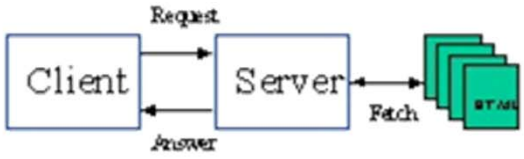
- HTTP Headers:
  - What HTTP Headers can do?

    [Ref] http://www.cs.tut.fi/~jkorpela/http.html
    - Content information (type, date, size, encoding, …)
    - Cache control
    - Authentication
    - URL Redirection
    - Transmitting cookies
    - Knowing where client come from
    - Knowing what software client use
    - …

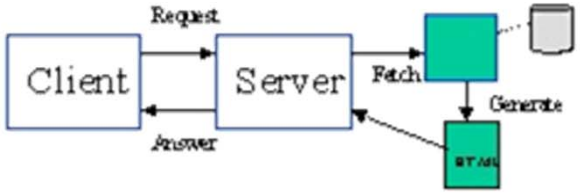
Web Hosting – Static vs. Dynamic Pages (1)

- Static vs. Dynamic Pages

Static vs. Dynamic



An HTML document stored in a file is a static Web page. Unless the file is edited, its content does not change.

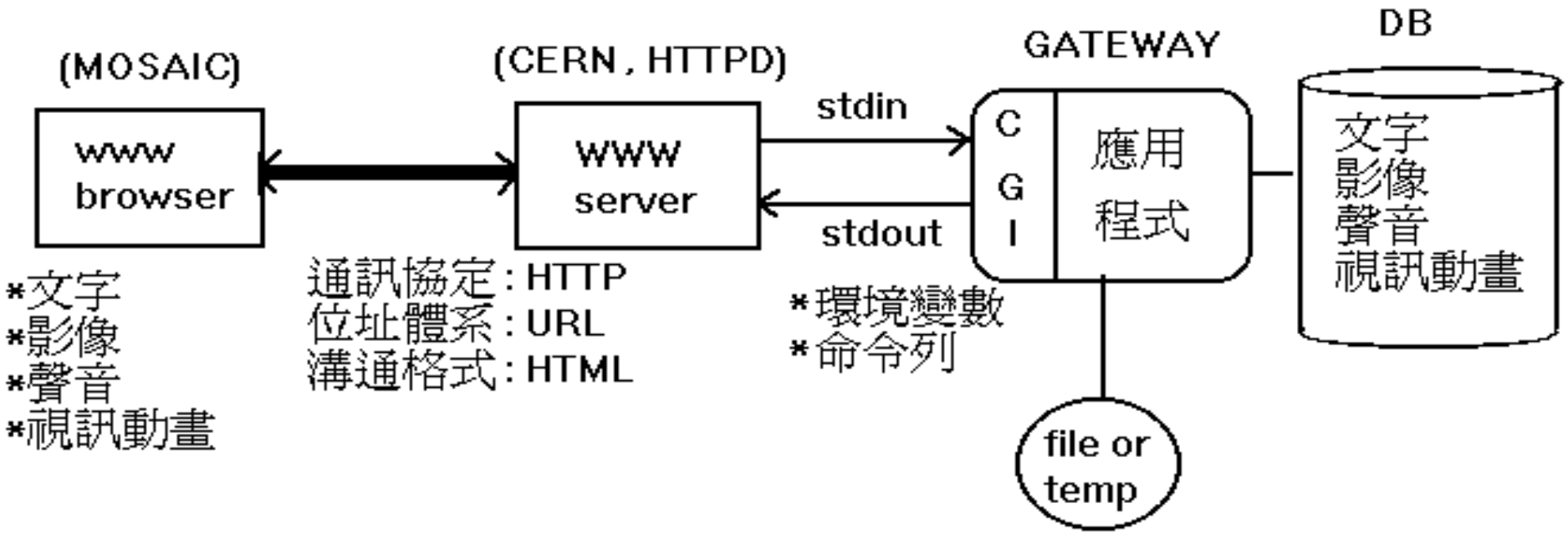


A dynamic Web page is generated or partially generated each time it is accessed.

  - Technologies of Dynamic Web Pages
    - Client Script Language
      - JavaScript, VBScript
    - Client Interactive Technology
      - Java Applet, Flash, AJAX
    - Server Side
      - CGI (Perl, C/C++, …)
      - ASP, JSP, PHP

Web Hosting – Static vs. Dynamic Pages (2)

- CGI (Common Gateway Interface)
  - A specification that allows an HTTP server to exchange information with other programs



(圖 1)WWW主從架構應用示意圖

Web Hosting
– Virtual Hosting (1)

- Providing services for more than one domain-name (or IP) in one web server.
- IP-Based Virtual Hosting v.s. Name-Based Virtual Hosting
  - IP-Based – Several IPs (or ports)
  - Name-Based – Singe IP, several hostnames
- Example (Apache configuration)

```
NameVirtualHost 140.116.82.246

<VirtualHost 140.116.82.246>
ServerName imslab.org
DocumentRoot “/home/www"
</VirtualHost>

<VirtualHost 140.116.82.246>
ServerName tsaimh.imslab.org
DocumentRoot "/home/tsaimh/www"
</VirtualHost>
```

```
<VirtualHost 140.116.82.246:80>
DocumentRoot /home/www
ServerName imslab.org
</VirtualHost>

<VirtualHost 140.116.82.243:80>
DocumentRoot /home/tsaimh/www
ServerName tsaimh.imslab.org
</VirtualHost>
```

Web Hosting – Virtual Hosting (2)

Q: How Name-Based Virtual Hosting works?

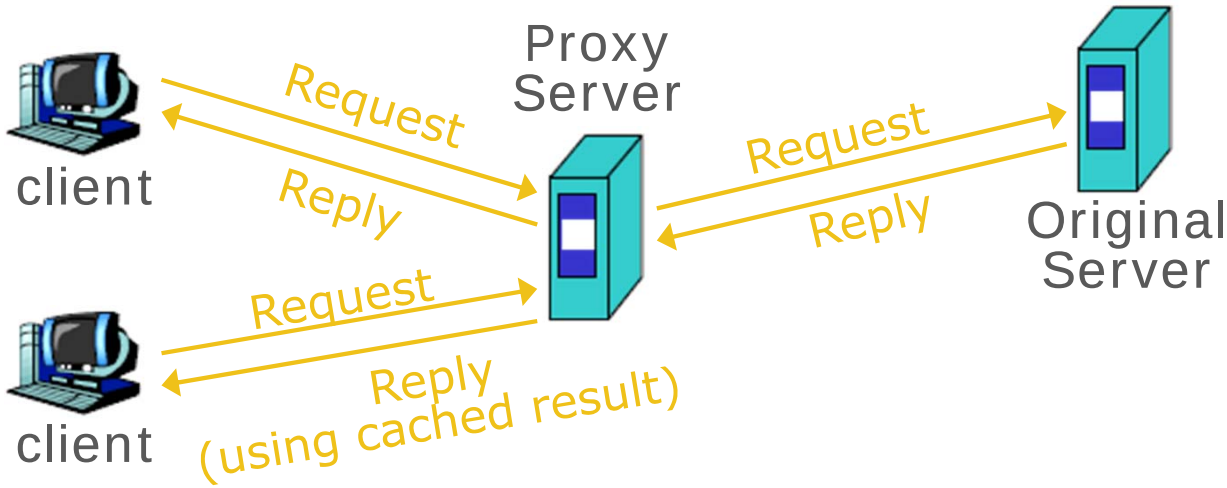
A: It primarily uses HTTP **Host** Header.

```
> telnet imslab.org 80
Trying 140.116.82.246...
Connected to IMSLAB.org.
Escape character is '^]'.
GET / http/1.0
Host: imslab.org

HTTP/1.1 200 OK
Date: Thu, 20 Dec 2012 02:48:39 GMT
Last-Modified: Sat, 03 Dec 2011 11:37:41 GMT
Content-Length: 127
...
Content-Type: text/html

<html>
<head>
<meta http-equiv=refresh
content="0;url=http://imslab.org/cht/">
<title></title>
</head>
<body>
</body>
</html>
```

```
> telnet imslab.org 80
Trying 140.116.82.246...
Connected to IMSLAB.org.
Escape character is '^]'.
GET / http/1.0
Host: tsaimh.imslab.org

HTTP/1.1 200 OK
Date: Thu, 20 Dec 2012 02:52:37 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Content-Length: 1773
Content-Type: text/html

<html>
    <!DOCTYPE html PUBLIC '-//W3C//DTD XHTML
1.1//EN'
'http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd'
>
<html xmlns='http://www.w3.org/1999/xhtml'
xml:lang='en' >

...
```

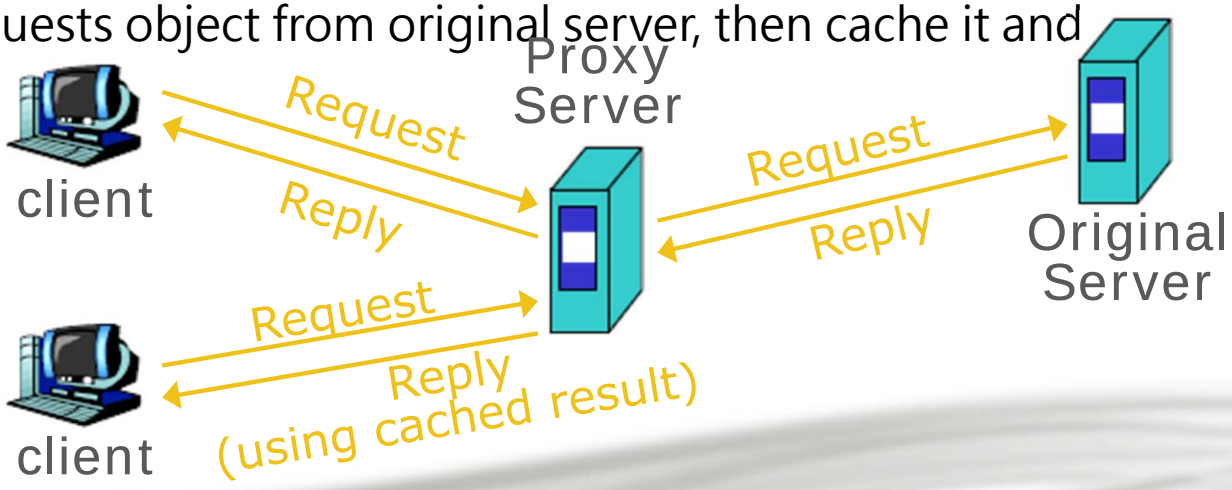
Proxy

- Proxy
  - A proxy server is a server which serves its clients by:
    - Making requests to other servers to request web pages.
    - Caching some results for the same requests in the future.
  - Goals:
    - Performance
    - Stability
    - Central Control
    - …etc.
  - Roles:
    - Forward Proxy
    - Reverse Proxy
  - Targets
    - Web pages/FTP files
    - TCP/IP Connections
    - …etc.



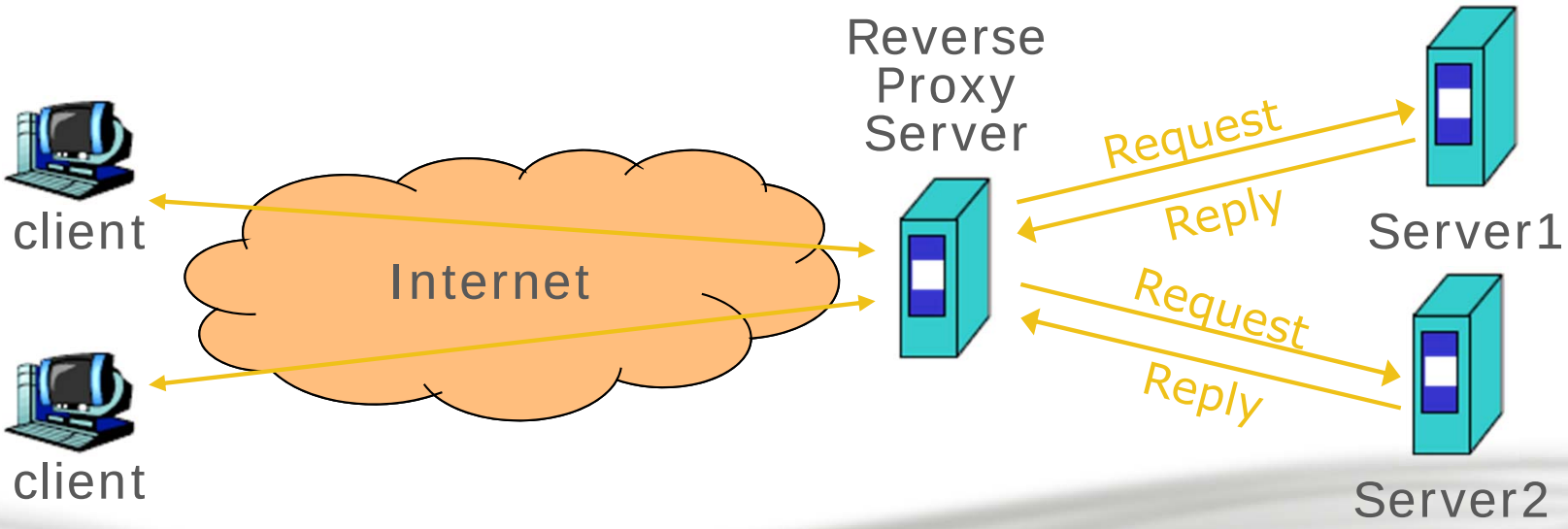
Proxy
– The Forward Proxy

- Forward Proxy
  - Proxy the outgoing requests, for the reason of
    - Bandwidth saving
    - Performance
    - Central control
  - When objects requested are
    - In cache, return the cached objects
    - Otherwise, proxy server requests object from original server, then cache it and return to client



Proxy
– The Reverse Proxy

- Reverse Proxy
  - Proxy the incoming requests, for the reason of
    - Reducing Server Load (by caching)
    - Load Balance
    - Fault Tolerant
  - Reverse proxy acts as the original server: accepting incoming requests, replying corresponding result. SEAMLESS for clients!



Proxy – SQUID

- A web proxy server & cache daemon.
  - Supports HTTP, FTP
  - Limited support for TLS, SSL, Gopher, HTTPS
- Port install: /usr/ports/www/squid{,30,31}
- Startup:
  - /etc/rc.conf
    - squid_enable="YES"
  - /usr/local/etc/rc.d/squid start
- Configuration Sample/Documents:
  - /usr/local/etc/squid/squid.conf.default