

# Web Content Delivery

## Reading: Section 9.1.2 and 9.4.3

*Acknowledgments: Lecture slides are from Computer networks course thought by Jennifer Rexford at Princeton University. When slides are obtained from other sources, a reference will be noted on the bottom of that slide. A full list of references is provided on the last slide.*

# Outline: Web Content Distribution



- Main ingredients of the Web
  - URL, HTML, and HTTP
  - HTTP: the protocol and its stateless property
- Web Systems Components
  - Clients
  - Servers
  - DNS (Domain Name System)
- Interaction with underlying network protocol: TCP
- Scalability and performance enhancement
  - Server farms
  - Web Proxy
  - Content Distribution Network (CDN)

# Web History



- Before the 1970s-1980s
  - Internet used mainly by researchers and academics
  - Log in remote machines, transfer files, exchange e-mail
- Internet growth and commercialization
  - 1988: ARPANET gradually replaced by the NSFNET
  - Early 1990s: NSFNET begins to allow commercial traffic
- Initial proposal for the Web by Berners-Lee in 1989
- Enablers for the success of the Web
  - 1980s: Home computers with graphical user interfaces
  - 1990s: Power of PCs increases, and cost decreases



# Main ingredients of the Web

- URL

- Denotes the global unique location of the web resource
- Formatted string

e.g., `http://www.sharif.edu/index.html`

Protocol for communicating with server (e.g.,  
`http`)

Name of the server (e.g., `www.sharif.edu`)

Name of the resource (e.g., `index.html`)

- HTML

- Actual content of web resource, represented in ASCII

# Main ingredients of the Web: HTML



- **HyperText Markup Language (HTML)**
  - Format text, reference images, embed hyperlinks
  - Representation of hypertext documents in ASCII format
  - Interpreted by Web browsers when rendering a page
- **Web page**
  - Base HTML file
  - referenced objects (e.g., images), Each object has its own URL
- **Straight-forward and easy to learn**
  - Simplest HTML document is a plain text file
  - Automatically generated by authoring programs



# Main ingredients of the Web

- URL

- Denotes the global unique location of the web resource
- Formatted string

e.g., <http://www.sharif.edu/index.html>

Protocol for communicating with server (e.g.,  
http)

Name of the server (e.g., [www.sharif.edu](http://www.sharif.edu))

Name of the resource (e.g., index.html)

- HTML

- Actual content of web resource, represented in ASCII

- HTTP

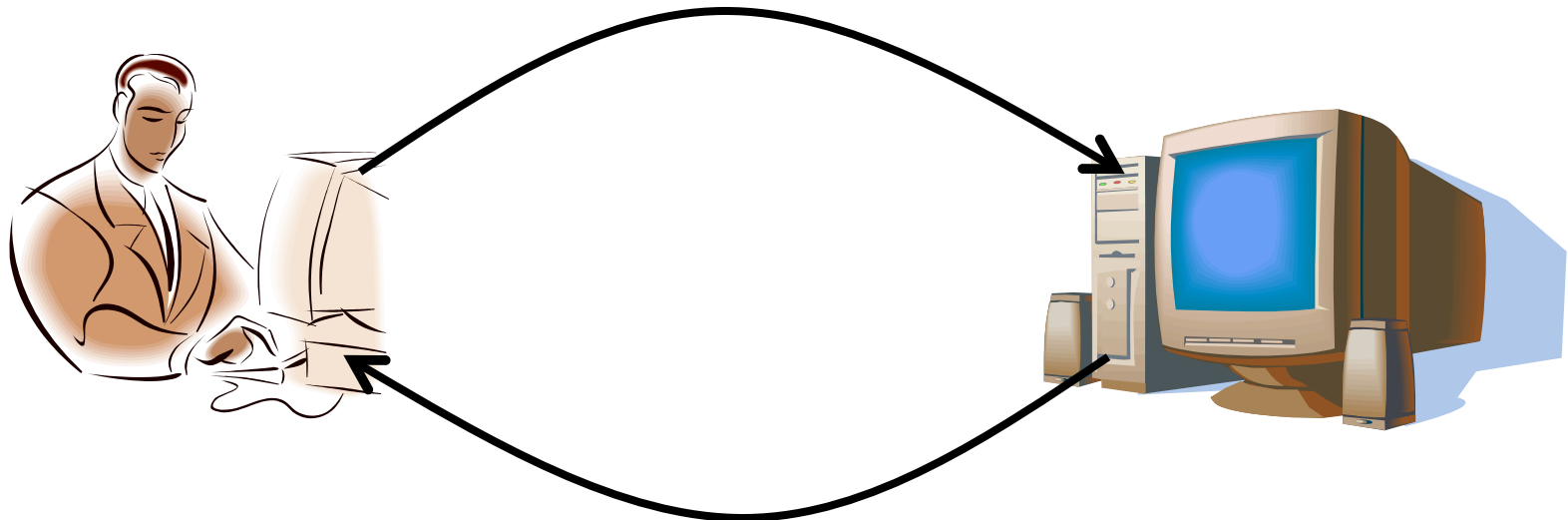
- Protocol for client/server communication

# Main ingredients of the Web: HTTP



- Client program
  - E.g., Web browser
  - Running on end host
  - Requests service
- Server program
  - E.g., Web server
  - Provides service

`GET /index.html`



`"Site under construction"`

# Outline: Web Content Distribution



- Main ingredients of the Web
  - URL, HTML, and HTTP
  - HTTP: the protocol and its stateless property
- Web Systems Components
  - Clients
  - Servers
  - DNS (Domain Name System)
- Interaction with underlying network protocol: TCP
- Scalability and performance enhancement
  - Server farms
  - Web Proxy
  - Content Distribution Network (CDN)



# HTTP Example: Request and Response Message

```
GET /courses/archive/spring08/ce443/ HTTP/1.1  
Host: www.ce.sharif.edu  
User-Agent: Mozilla/4.03  
<CRLF>
```

Request

```
HTTP/1.1 200 OK  
Date: Mon, 6 Feb 2006 13:09:03 GMT  
Server: Netscape-Enterprise/3.5.1  
Last-Modified: Mon, 6 Feb 2006 11:12:23 GMT  
Content-Length: 21  
<CRLF>  
Site under construction
```

Response



# HTTP Request Message

- Request message sent by a client
  - Request line: method, resource, and protocol version
  - Request headers: provide information or request
  - Body: optional data (e.g., to “POST” data to the server)

request line  
(GET, POST,  
HEAD commands)

header  
lines

```
GET /somedir/page.html HTTP/1.1
Host: www.someschool.edu
User-agent: Mozilla/4.0
Connection: close
Accept-language: fr
```

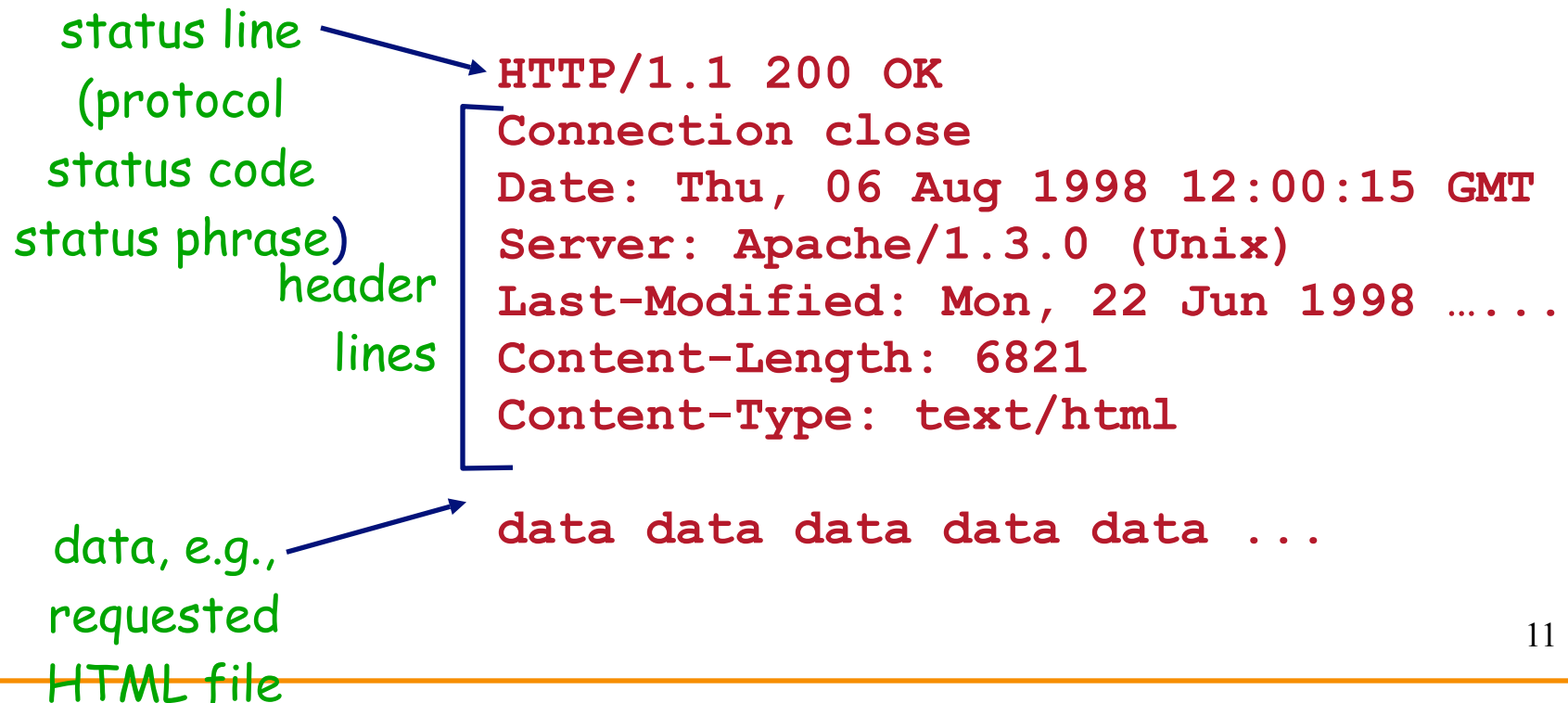
Carriage return,  
line feed  
indicates end  
of message

(extra carriage return, line feed)



# HTTP Response Message

- Response message sent by a server
  - Status line: protocol version, status code, status phrase
  - Response headers: provide information
  - Body: optional data





# HTTP:

## Request Methods and Response Codes

- Request methods include
  - GET: return current value of resource, ...
  - HEAD: return the meta-data associated with a resource
  - POST: update a resource, provide input to a program, ...
  - Etc.
- Response code classes
  - 1xx: informational (e.g., “100 Continue”)
  - 2xx: success (e.g., “200 OK”)
  - 3xx: redirection (e.g., “304 Not Modified”)
  - 4xx: client error (e.g., “404 Not Found”)
  - 5xx: server error (e.g., “503 Service Unavailable”)

# HTTP is a Stateless Protocol



- Stateless
  - Each request-response exchange treated independently
  - Clients and servers not required to retain state
- Statelessness to improve scalability
  - Avoids need for the server to retain info across requests
  - Enables the server to handle a higher rate of requests

# Outline: Web Content Distribution



- Main ingredients of the Web
  - URL, HTML, and HTTP
  - HTTP: the protocol and its stateless property
- Web Systems Components
  - Clients
  - Servers
  - DNS (Domain Name System)
- Interaction with underlying network protocol: TCP
- Scalability and performance enhancement
  - Server farms
  - Web Proxy
  - Content Distribution Network (CDN)

# Web Systems Components



- **Clients**
  - Send requests and receive responses
  - Browsers, spiders, and agents
- **Servers**
  - Receive requests and send responses
  - Store or generate the responses
- **DNS (Domain Name System)**
  - Distributed network infrastructure
  - Transforms site name -> IP address
  - Direct clients to servers

# Web Browser



- **Generating HTTP requests**
  - User types URL, clicks a hyperlink, or selects bookmark
  - User clicks “reload”, or “submit” on a Web page
  - Automatic downloading of embedded images
- **Layout of response**
  - Parsing HTML and rendering the Web page
  - Invoking helper applications (e.g., Acrobat, PowerPoint)
- **Maintaining a cache**
  - Storing recently-viewed objects
  - Checking that cached objects are fresh

# Typical Web Transaction



- User clicks on a hyperlink
  - `http://www.cnn.com/index.html`
- Browser learns the IP address of the server
  - Invokes `gethostbyname(www.cnn.com)`
  - And gets a return value of `64.236.16.20`
- Browser establishes a TCP connection
  - Selects an ephemeral port for its end of the connection
  - Contacts `64.236.16.20` on port 80
- Browser sends the HTTP request
  - “`GET /index.html HTTP/1.1`  
`Host: www.cnn.com`”

# Typical Web Transaction (Continued)



- Browser parses the HTTP response message
  - Extract the URL for each embedded image
  - Create new TCP connections and send new requests
  - Render the Web page, including the images
- Opportunities for caching in the browser
  - HTML file
  - Each embedded image
  - IP address of the Web site

# Web Systems Components



- **Clients**
  - Send requests and receive responses
  - Browsers, spiders, and agents
- **Servers**
  - Receive requests and send responses
  - Store or generate the responses
- **DNS (Domain Name System)**
  - Distributed network infrastructure
  - Transforms site name -> IP address
  - Direct clients to servers

# Web Server



- **Web site vs. Web server**
  - Web site: collections of Web pages associated with a particular host name
  - Web server: program that satisfies client requests for Web resources
- **Handling a client request**
  - Accept the TCP connection
  - Read and parse the HTTP request message
  - Translate the URL to a filename
  - Determine whether the request is authorized
  - Generate and transmit the response

# Web Server: Generating a Response



- Returning a file
  - URL corresponds to a file (e.g., /www/index.html)
  - ... and the server returns the file as the response
  - ... along with the HTTP response header
- Returning meta-data with no body
  - Example: client requests object “if-modified-since”
  - Server checks if the object has been modified
  - ... and simply returns a “HTTP/1.1 304 Not Modified”
- Dynamically-generated responses
  - URL corresponds to a program the server needs to run
  - Server runs the program and sends the output to client

# Hosting: Multiple Sites Per Machine



- Multiple Web sites on a single machine
  - Hosting company runs the Web server on behalf of multiple sites (e.g., [www.foo.com](http://www.foo.com) and [www.bar.com](http://www.bar.com))
- Problem: returning the correct content
  - [www.foo.com/index.html](http://www.foo.com/index.html) vs. [www.bar.com/index.html](http://www.bar.com/index.html)
  - How to differentiate when both are on same machine?
- Solution: multiple servers on the same machine
  - Run multiple Web servers on the machine
  - Have a separate IP address for each server

# Hosting: Multiple Machines Per Site



- Replicating a popular Web site
  - Running on multiple machines to handle the load
  - ... and to place content closer to the clients
- Problem: directing client to a particular replica
  - To balance load across the server replicas
  - To pair clients with nearby servers
- Solution:
  - Takes advantage of Domain Name System (DNS)

# Web Systems Components



- **Clients**
  - Send requests and receive responses
  - Browsers, spiders, and agents
- **Servers**
  - Receive requests and send responses
  - Store or generate the responses
- **DNS (Domain Name System) and the Web**
  - Distributed network infrastructure
  - Transforms site name -> IP address
  - Direct clients to servers



# DNS Query in Web Download

- User types or clicks on a URL
  - E.g., <http://www.cnn.com/2006/leadstory.html>
- Browser extracts the site name
  - E.g., [www.cnn.com](http://www.cnn.com)
- Browser calls `gethostbyname()` to learn IP address
  - Triggers resolver code to query the local DNS server
- Eventually, the resolver gets a reply
  - Resolver returns the IP address to the browser
- Then, the browser contacts the Web server
  - Creates and connects socket, and sends HTTP request



# Multiple DNS Queries

- Often a Web page has embedded objects
  - E.g., HTML file with embedded images
- Each embedded object has its own URL
  - ... and potentially lives on a different Web server
  - E.g., <http://www.myimages.com/image1.jpg>
- Browser downloads embedded objects
  - Usually done automatically, unless configured otherwise
  - Requires learning the address for [www.myimages.com](http://www.myimages.com)

# When are DNS Queries Unnecessary?



- Browser is configured to use a proxy
  - E.g., browser sends all HTTP requests through a proxy
  - Then, the proxy takes care of issuing the DNS request
- Requested Web resource is locally cached
  - E.g., cache has <http://www.cnn.com/2006/leadstory.html>
  - No need to fetch the resource, so no need to query
- Resulting IP address is locally cached
  - Browser recently visited <http://www.cnn.com>
  - So, the browser already called *gethostbyname()*
  - ... and may be locally caching the resulting IP address

# Directing Web Clients to Replicas



- Simple approach: different names
  - www1.cnn.com, www2.cnn.com, www3.cnn.com
  - But, this requires users to select specific replicas
- More elegant approach: different IP addresses
  - Single name (e.g., www.cnn.com), multiple addresses
  - E.g., 64.236.16.20, 64.236.16.52, 64.236.16.84, ...
- Authoritative DNS server returns many addresses
  - And the local DNS server selects one address
  - Authoritative server may vary the order of addresses

# Clever Load Balancing Schemes



- Selecting the “best” IP address to return
  - Based on server performance
  - Based on geographic proximity
  - Based on network load
  - ...
- Example policies
  - Round-robin scheduling to balance server load
  - U.S. queries get one address, Europe another
  - Tracking the current load on each of the replicas

# Outline: Web Content Distribution



- Main ingredients of the Web
  - URL, HTML, and HTTP
  - HTTP: the protocol and its stateless property
- Web Systems Components
  - Clients
  - Servers
  - DNS (Domain Name System)
- Interaction with underlying network protocol: TCP
- Scalability and performance enhancement
  - Server farms
  - Web Proxy
  - Content Distribution Network (CDN)

# TCP Interaction: Multiple Transfers

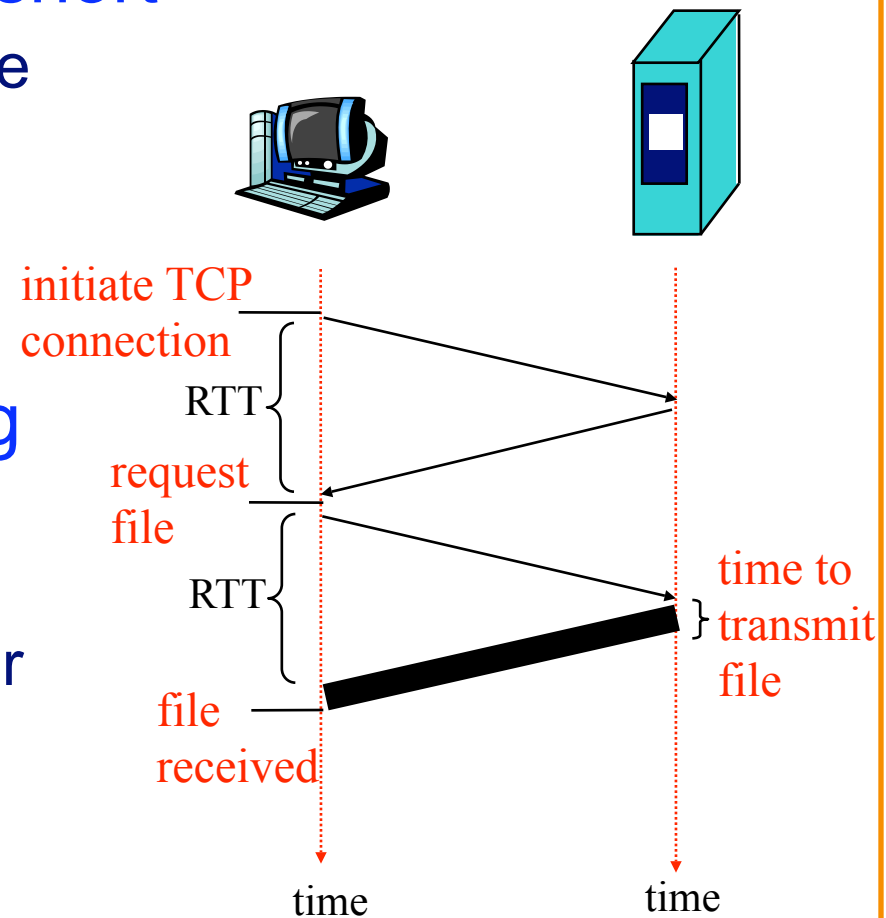


- Most Web pages have multiple objects
  - E.g., HTML file and multiple embedded images
- Serializing the transfers is not efficient
  - Sending the images one at a time introduces delay
  - Cannot start retrieving second images until first arrives
- Parallel connections
  - Browser opens multiple TCP connections (e.g., 4)
  - ... and retrieves a single image on each connection
- Performance trade-offs
  - Multiple downloads sharing the same network links
  - Unfairness to other traffic traversing the links

# TCP Interaction: Short Transfers



- Most HTTP transfers are short
  - Very small request message (e.g., a few hundred bytes)
  - Small response message (e.g., a few kilobytes)
- TCP overhead may be big
  - Three-way handshake to establish connection
  - Four-way handshake to tear down the connection



# TCP Interaction: Short Transfers



- Round-trip time estimation
  - Maybe large at the start of a connection (e.g., 3 seconds)
  - Leads to latency in detecting lost packets
- Congestion window
  - Small value at beginning of connection (e.g., 1 MSS)
  - May not reach a high value before transfer is done
- Detecting packet loss
  - Timeout: slow ☹️
  - duplicate ACK
    - requires many packets in flight
    - which doesn't happen for very short transfers ☹️

# TCP Interaction: Persistent Connections

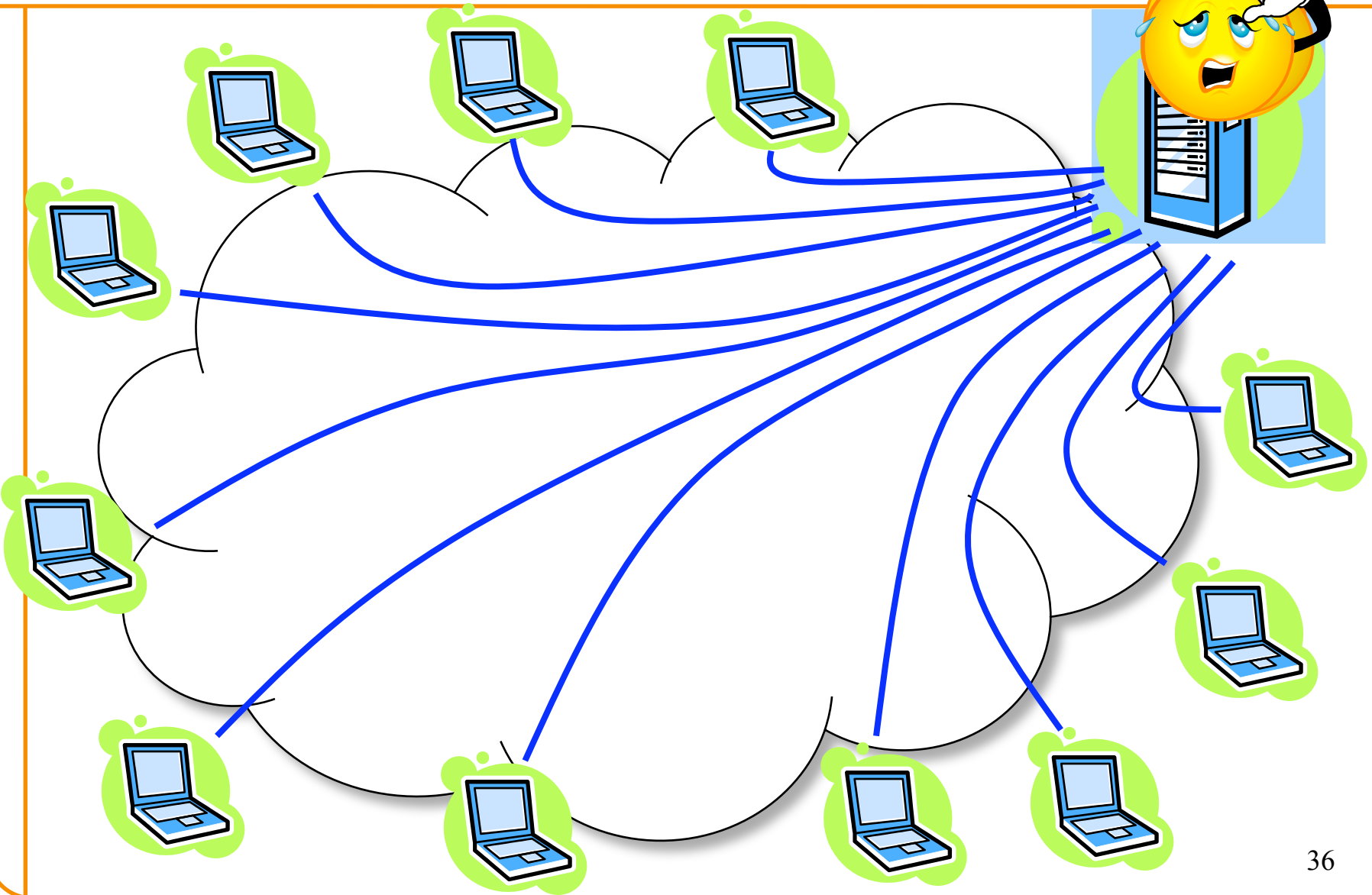


- Handle multiple transfers per connection
  - Maintain the TCP connection across multiple requests
  - Either the client or server can tear down the connection
  - Added to HTTP after the Web became very popular
- Performance advantages
  - Avoid overhead of connection set-up and tear-down
  - Allow TCP to learn a more accurate RTT estimate
  - Allow the TCP congestion window to increase

# Web Content Delivery



# Scalability Limitation

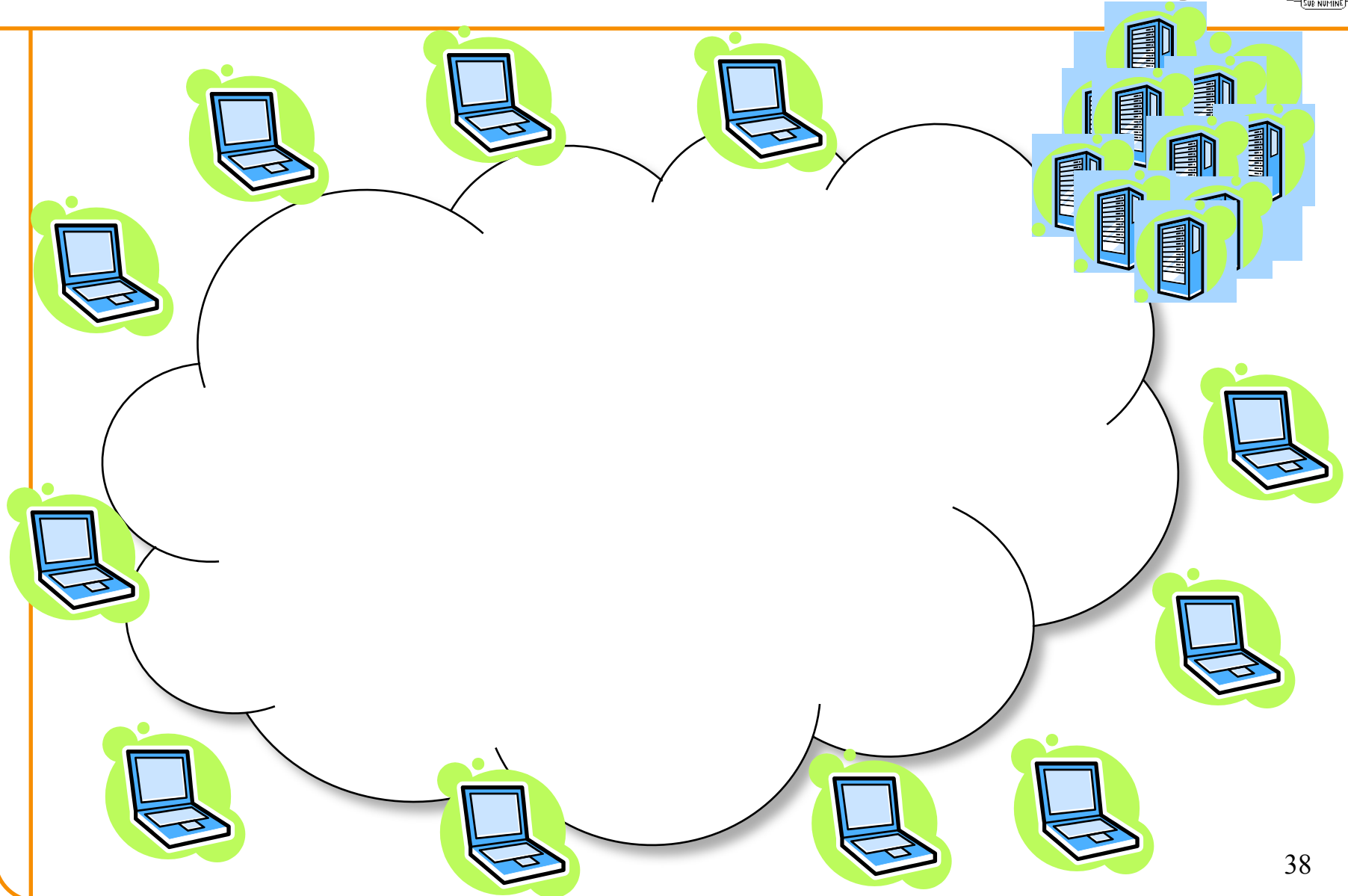


# Outline: Web Content Distribution



- Main ingredients of the Web
  - URL, HTML, and HTTP
  - HTTP: the protocol and its stateless property
- Web Systems Components
  - Clients
  - Servers
  - DNS (Domain Name System)
- Interaction with underlying network protocol: TCP
- Scalability and performance enhancement
  - Server farms
  - Proxy
  - Content Distribution Network (CDN)

# Server Farms (motivated for scalability)



# Server Farms



- **Definition**
  - a collection of computer servers to accomplish server needs far beyond the capacity of one machine.
  - Often have both a primary and backup server allocated to a single task (for fault tolerance)
- **Web Farms**
  - Common use of server farms is for web hosting

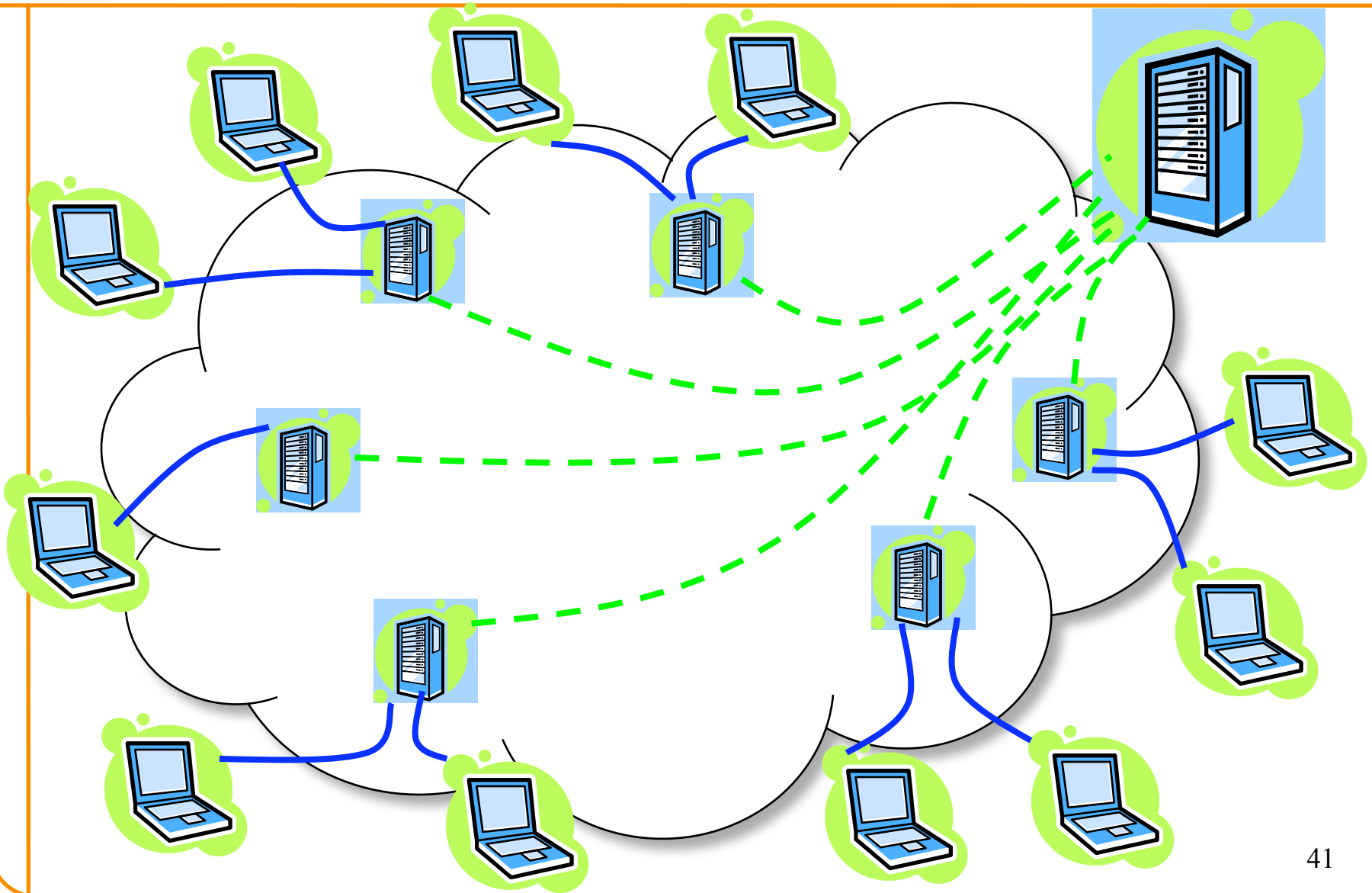


# Outline: Web Content Distribution



- Main ingredients of the Web
  - URL, HTML, and HTTP
  - HTTP: the protocol and its stateless property
- Web Systems Components
  - Clients
  - Servers
  - DNS (Domain Name System)
- Interaction with underlying network protocol: TCP
- Scalability and performance enhancement
  - Server farms
  - **Web Proxy**
  - Content Distribution Network (CDN)

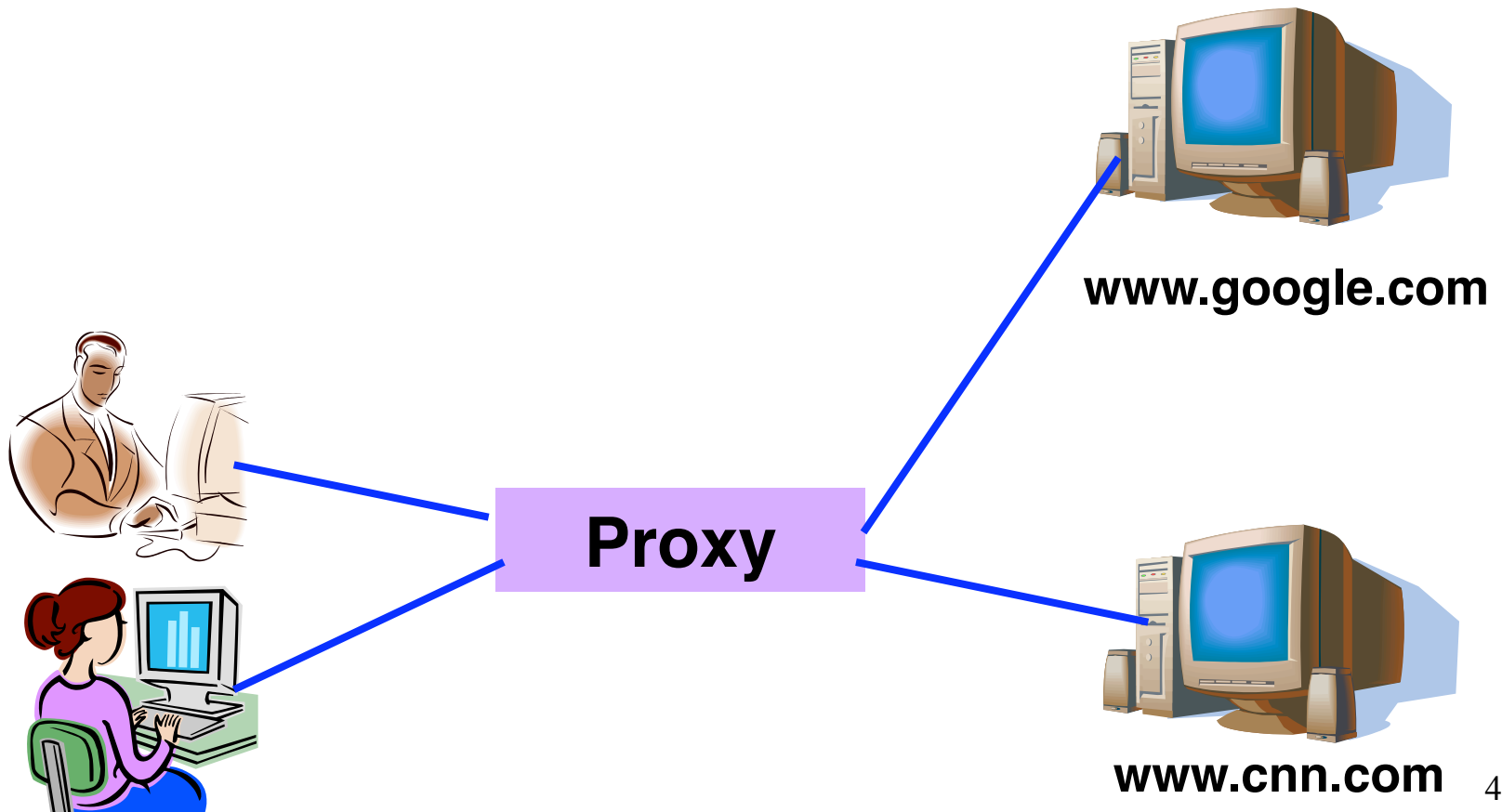
# Web Proxies



# Web Proxies are Intermediaries



- Proxies play both roles
  - A server to the client
  - A client to the server





# Proxy Caching

- Client #1 requests <http://www.foo.com/fun.jpg>
  - Client sends “GET fun.jpg” to the proxy
  - Proxy sends “GET fun.jpg” to the server
  - Server sends response to the proxy
  - Proxy stores the response, and forwards to client
- Client #2 requests <http://www.foo.com/fun.jpg>
  - Client sends “GET fun.jpg” to the proxy
  - Proxy sends response to the client from the cache
- Benefits
  - Faster response time to the clients
  - Lower load on the Web server
  - Reduced bandwidth consumption inside the network

# Getting Requests to the Proxy



- **Explicit configuration**
  - Browser configured to use a proxy
  - Directs all requests through the proxy
  - Problem: requires user action
- **Transparent proxy (or “interception proxy”)**
  - Proxy lies in path from the client to the servers
  - Proxy intercepts packets en route to the server
  - ... and interposes itself in the data transfer
  - Benefit: does not require user action

# Other Functions of Web Proxies



- Anonymization
  - Server sees requests coming from the proxy address
  - ... rather than the individual user IP addresses
- Transcoding
  - Converting data from one form to another
  - E.g., reducing the size of images for cell-phone browsers
- Prefetching
  - Requesting content before the user asks for it
- Filtering
  - Blocking access to sites, based on URL or content

# Outline: Web Content Distribution



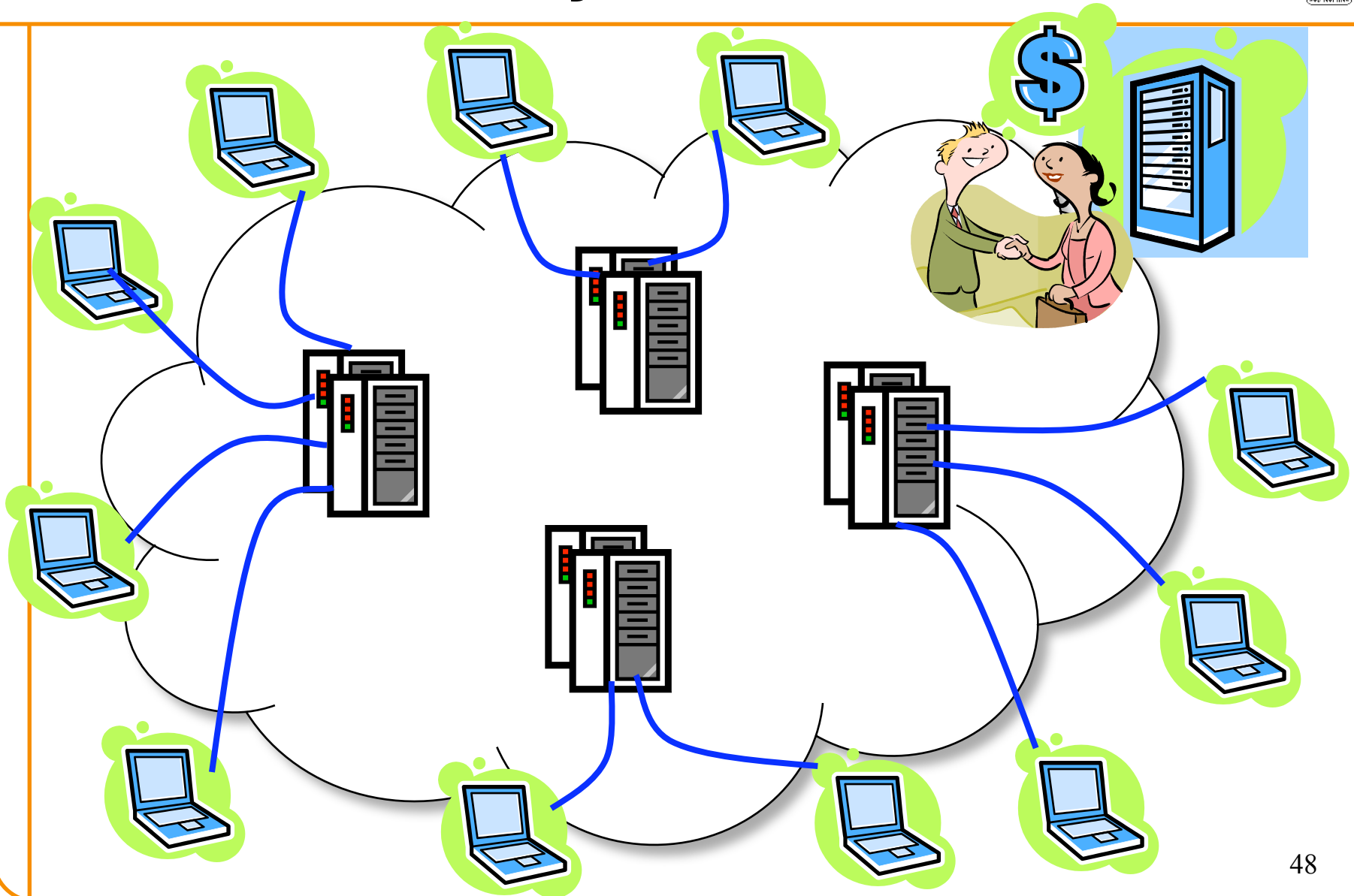
- Main ingredients of the Web
  - URL, HTML, and HTTP
  - HTTP: the protocol and its stateless property
- Web Systems Components
  - Clients
  - Servers
  - DNS (Domain Name System)
- Interaction with underlying network protocol: TCP
- Scalability and performance enhancement
  - Server farms
  - Web Proxy
  - Content Distribution Network (CDN)



# Motivation for CDN

- Providers want to offer content to consumers
  - Efficiently
  - Reliably
  - Securely
  - Inexpensively
- The server and its link can be overloaded
- Peering points between ISPs can be congested
- Alternative solution: Content Distribution Networks
  - Geographically diverse servers serving content from many sources

# Content Delivery Networks

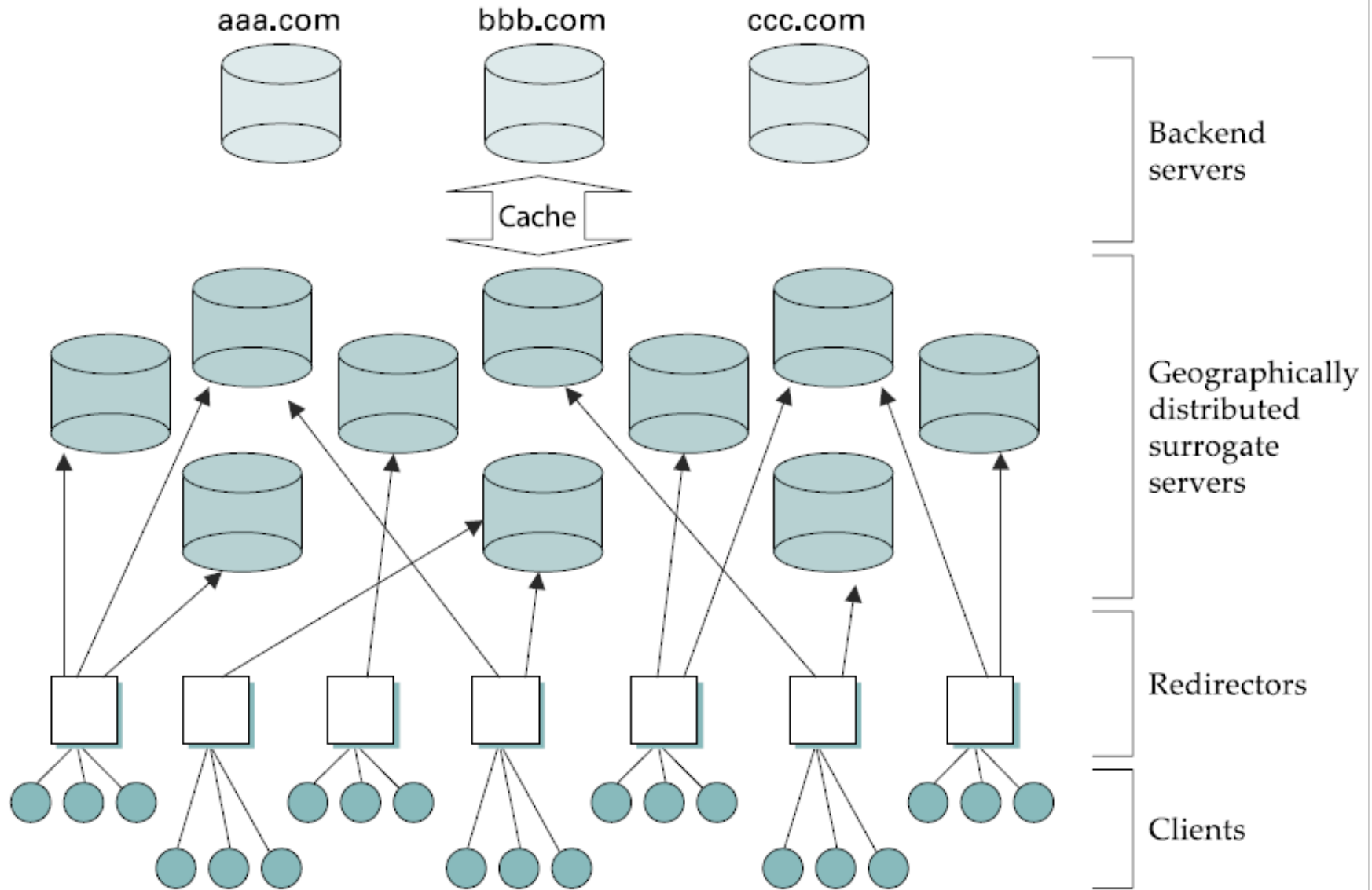




# CDN Architecture

- Proactively replicate data by caching static pages
- Architecture
  - Backend servers
  - Geographically distributed surrogate servers
  - Redirectors (according to network proximity, balancing)
  - Clients
- Redirector Mechanisms
  - Augment DNS to return different server addresses
  - Server-based redirection: based on HTTP redirect feature

# CDN Architecture



# Summary: Web Content Distribution



- Protocols and Standards
  - URL, HTML, and HTTP
  - HTTP Interaction with underlying network protocol: TCP
- Systems Components: Client/Server
- Web interaction with DNS infrastructure
- Scalability and performance enhancement
  - Server farms: replication
  - Web Proxy: indirection
  - Content Distribution Network (CDN): indirection and replication
- Next Lecture on Translating Addresses
  - DNS, DHCP, and ARP