

CCN: Network Applications Protocols

Dr. E.SURESH BABU

Assistant Professor

Computer Science and Engineering Department

National Institute of Technology, Warangal.

Warangal, TS, India.





Outline

- ❖ Application Layer Protocols
- ❖ Web Protocols
 - ✓ Overview of HTTP



Application Layer Protocols

Why is Application Layer Protocols are Required

- ❖ Usually, Network Processes communicate with each other by sending **MESSAGES** into sockets.
 - ✓ **How these messages are structured?**
 - ✓ **What are the meanings of the various fields in the messages?**
 - ✓ **When do the processes send the messages?**
- ❖ **Application-layer Protocol** will define the answer to all the above questions

Purpose of Application Layer Protocols

- ❖ Specifically, **Application-layer Protocol** specifies the details, such as:
 - ✓ The **syntax and semantics of messages** that can be exchanged
 - ✓ Whether the **client or server initiates interaction**
 - ✓ **Actions to be taken** if an error arises
 - ✓ How the **two sides know when to terminate communication**

Application Layer Protocols

- ❖ An **Application-layer Protocol** is only one piece of a **network application**
- ❖ **Application-layer Protocols** specify two aspects of interaction
 - 1. Representation**
 - 2. Transfer.**

Two Key Aspects of an Application-layer Protocol.

Aspect	Description
Data Representation	Syntax of data items that are exchanged, specific form used during transfer, translation of integers, characters, and files between computers
Data Transfer	Interaction between client and server, message syntax and semantics, valid and invalid exchange error handling, termination of interaction

❖ **Web** uses separate protocols to describe **web page syntax and web page transfer**.

Application Protocol Examples

- ❖ **Web Browsing**
- ❖ **Email**
- ❖ **File Transfer**
- ❖ **Remote Login And Remote Desktop**
- ❖ **Domain Name System (Name Lookup)**

**Application Layer Protocols :
Web Protocols**

Introduction to Web

- ❖ In Early 1990s, a major new application **World Wide Web** had arrived as the first Internet application.
- ❖ The **World Wide Web** is one of the most widely used **services** in the Internet.
- ❖ Web servers as a platform for many killer applications emerging after 2003, including **YouTube, Gmail, and Facebook**.

Advantages of Web

- ❖ Using Web,
 - ✓ It is relatively easy for any individual to make information available over the Web
 - ✓ Everyone can become a publisher at extremely low cost.
 - ✓ Hyperlinks and search engines help us navigate through an ocean of Web sites.
 - ✓ Graphics stimulate our senses.
 - ✓ Forms, JavaScript, Java applets, and many other devices enable us to interact with pages and sites.

Three Key Standards- World Wide Web Service Uses'

- ❖ Web is complex, many protocol standards have been devised to specify various aspects and details

Standard	Purpose
HyperText Markup Language (HTML)	A representation standard used to specify the contents and layout of a web page
Uniform Resource Locator (URL)	A representation standard that specifies the format and meaning of web page identifiers
HyperText Transfer Protocol (HTTP)	A transfer protocol that specifies how a browser interacts with a web server to transfer data

Hyper Text Markup Language (HTML)

- ❖ Representation standard for **multimedia documents**
- ❖ Specifies document is entirely in **printable text**
- ❖ Document contains **markup guidelines** rather than precise, detailed formatting or typesetting instructions

Uniform Resource Locator (URL)

protocol:// computer_name : port / document_name ? parameters

<http://nitw.ac.in/cse/faculty.html>

<http://10.45.10.1:80/cse/faculty.html>

World Wide Web- Review

- ❖ A **Web page** (also called a document) consists of **objects**.
 - ✓ An object can be **HTML file, a JPEG image, a Java applet, or a video clip**
 - ✓ All the **objects** are addressable by a **single URL**.
 - ✓ Most Web pages consist of a **base HTML file** and **several referenced objects**.

World Wide Web- Review

`www.someschool.edu/someDept/pic.gif`
Host Name Path Name

Overview of HTTP

Hyper Text Transfer Protocol (HTTP)

- ❖ The **Hyper Text Transfer Protocol (HTTP)** is a Web's application-layer protocol.
- ❖ HTTP is the **heart of the Web**.
- ❖ The HTTP is the **primary transfer protocol** that a **browser uses to interact with a web server**.

Hyper Text Transfer Protocol (HTTP)

- ❖ HTTP usually implemented in two programs: a client program and a server program, which are two different end systems that talk to each other by exchanging HTTP messages.
- ❖ In the client-server model, a browser is a client that extracts a server name from a URL and contacts the server.

Hyper Text Transfer Protocol (HTTP)

- ❖ HTTP can be characterized as follows:
 - ✓ Specifies format and meaning of messages
 - ✓ Each message represented as text
 - ✓ Transfers arbitrary binary data
 - ✓ Can download or upload data

How HTTP Works ?

- ❖ HTTP defines
 - ✓ How Web clients request Web pages from Web servers and
 - ✓ How servers transfer Web pages to clients.

HTTP used Client/Server Model



HTTP used Client/Server Model

- ❖ **Client:** Web Browser requests, receives, (using HTTP protocol) and “displays” Web objects
- ❖ **Server:** Web server sends (using HTTP protocol) objects in response to requests

Underlying Protocol

- ❖ HTTP uses **TCP** as its underlying **transport protocol**
- ❖ The **HTTP client** first **initiates a TCP connection** with the **server**.
- ❖ Once the **connection is established**,
 - ✓ The **browser and the server processes** access **TCP through their socket interfaces**.

HTTP uses TCP as a underlying protocol

- ❖ Client initiates **TCP connection** (creates socket) to server, port 80
- ❖ Server accepts **TCP connection from client**
- ❖ **HTTP messages** (application-layer protocol messages) **exchanged** between **browser (HTTP client)** and **Web server (HTTP server)**
- ❖ **TCP connection closed**

HTTP uses TCP as a underlying protocol

- ❖ TCP provides a **reliable data transfer service** to HTTP.
 - ✓ **HTTP need not worry** about **lost data or reordering of data** within the network.
- ❖ It is job of **TCP and the other protocols** in the lower layers of the protocol stack.

Observation

- ❖ HTTP is "**Stateless**" Protocol
 - ✓ **Server maintains no information about past client requests**
- ❖ If the HTTP protocols that **maintain past history (state)**
 - ✓ **It is complex !**
 - ✓ It is required, when **server/client crashes**, their views of "**state**" **may be inconsistent**, must be **reconciled**

Non-Persistent and Persistent Connections

❖ When this **client-server interaction** is taking place over **TCP**,

1. Each **request/response pair** be sent over a **separate TCP connection** (**Non-persistent Connections**).
2. All the **request/response pair** should be sent over the **same TCP connection**. (**Persistent Connections**)

Observation

❖ **HTTP** uses both **non-persistent connections and persistent connections**.

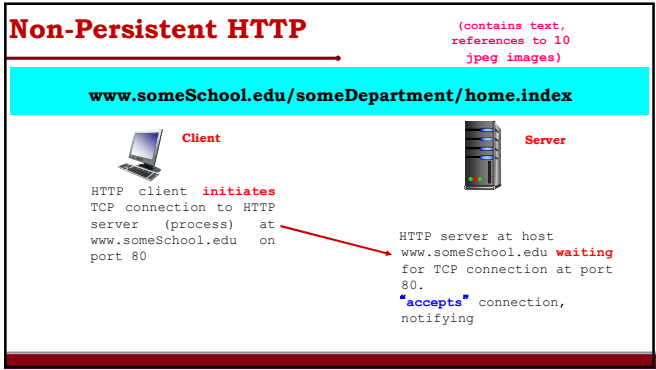
- ✓ By **default**, **HTTP** uses **persistent connections**
- ✓ **HTTP** clients and servers must be **configured** to use **non-persistent connections**

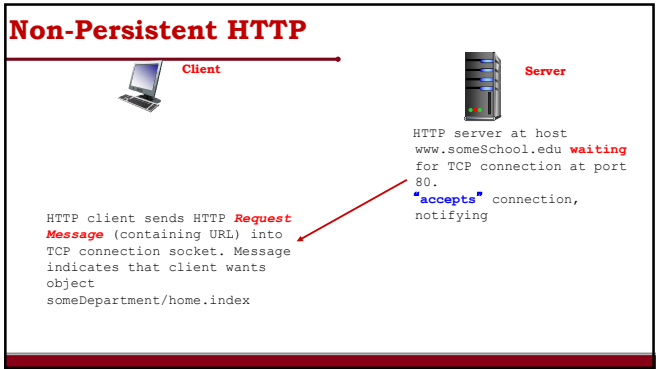
Non-Persistent Connections

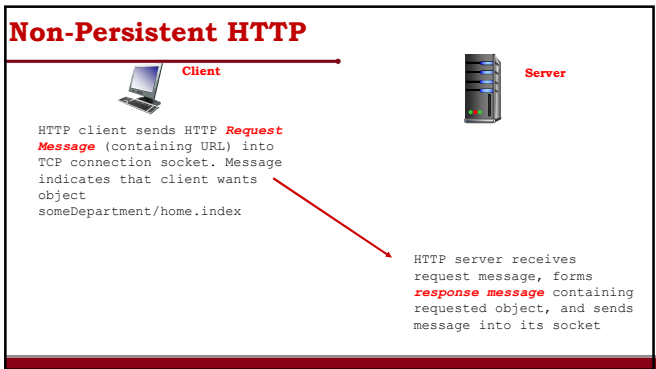
❖ **At most one object** sent over **TCP** connection

- ✓ **connection then closed**

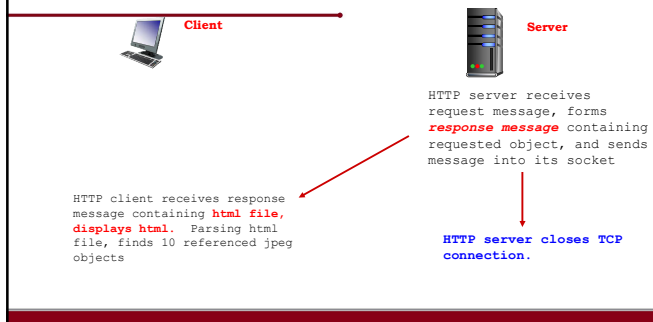
❖ Downloading **multiple objects** required **multiple connections**



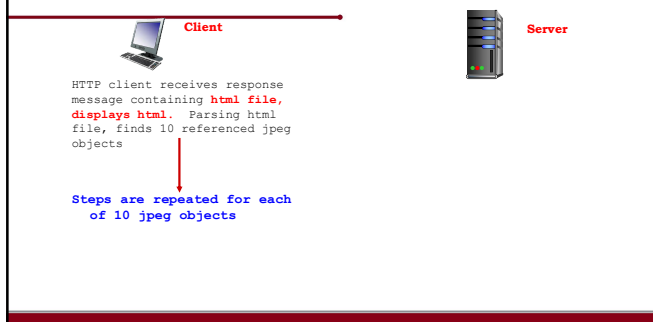




Non-Persistent HTTP



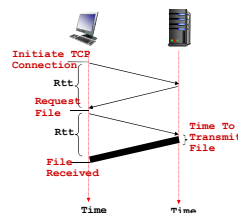
Non-Persistent HTTP



Non-persistent HTTP: Response Time

❖ **RTT (definition):** Time for a small packet to travel from client to server and back

❖ Non-persistent HTTP Response Time = **2RTT + file transmission time**



Non-Persistent HTTP : Limitations

- ❖ Requires **2 RTTs per object**
- ❖ **OS overhead** for each TCP connection
- ❖ Browsers often **open parallel TCP connections** to fetch referenced objects

Persistent HTTP

- ❖ **Multiple Objects** can be sent over **single TCP connection** between client, server.
 - ✓ Server leaves **connection open** after sending response
 - ✓ Subsequent **HTTP messages** between same client/server sent over **open connection**
 - ✓ Client **sends requests** as soon as it encounters a referenced object

Persistent HTTP

- ❖ Requires only **one RTT** for all the referenced objects

HTTP Message Format

- ❖ Two types of **HTTP messages: Request, Response**
- ❖ HTTP request message: **ASCII (human-readable format)**

The Four Major HTTP Request Types

- ❖ Once it establishes a connection, a browser sends an **HTTP request to the server**

Request	Description
GET	Requests a document; server responds by sending status information followed by a copy of the document
HEAD	Requests status information; server responds by sending status information, but does not send a copy of the document
POST	Sends data to a server; the server appends the data to a specified item (e.g., a message is appended to a list)
PUT	Sends data to a server; the server uses the data to completely replace the specified item (i.e., overwrites the previous data)

HTTP Message Format

request line
(GET, POST,
HEAD commands)

Header
Lines

carriage return,
line feed at start
of line indicates
end of header lines

GET /index.html HTTP/1.1\r\nHost: www-net.cs.umass.edu\r\nUser-Agent: Firefox/3.6.10\r\nAccept: text/html,application/xhtml+xml\r\nAccept-Language: en-us,en;q=0.5\r\nAccept-Encoding: gzip,deflate\r\nAccept-Charset: ISO-8859-1,utf-8;q=0.7\r\nKeep-Alive: 115\r\nConnection: keep-alive\r\n\r\n

Carriage Return Character
Line-feed Character

HTTP Response Message

Status Line
(Protocol
Status Code
Status Phrase)

Header
Lines


Data, E.G.,
Requested
HTML File

HTTP/1.1 200 OK\r\nDate: Sun, 26 Sep 2010 20:09:20 GMT\r\nServer: Apache/2.0.52 (CentOS)\r\nLast-Modified: Tue, 30 Oct 2007 17:00:02 GMT\r\nETag: "17dc6-a5c-bf716880"\r\nAccept-Ranges: bytes\r\nContent-Length: 2652\r\nKeep-Alive: timeout=10, max=100\r\nConnection: Keep-Alive\r\nContent-Type: text/html; charset=ISO-8859-1\r\n\r\ndata data data data data ...

HTTP Response Status Codes

❖ Status Code Appears In 1st Line In Server-to-client Response Message.

Status Code	Corresponding Status String
200	OK
400	Bad Request
404	Not Found




Outline

❖ Application Layer Protocols

❖ Web Protocols

- ✓ Overview of HTTP



Thank You

[illegible]