CN-Advanced
L33

HTTP Persistent Connections

Dr. Ram P Rustagi
rprustagi@ksit.edu.in
http://www.rprustagi.com
https://www.youtube.com/rprustagi

HTTP Headers

- General Headers
  - Used by both clients and servers
    - Date, connection, MIME-version …
- Request headers: specific to request message
  - Client-IP, Host, Accept, Referer, User-agent …
- Response headers: specific to server response
  - Server, Age, set-cookie, …
- Entity headers
  - Deals with entity body
    - Content-Type, Content-Length
    - Caching headers
      - Etag, Expires, Last-Modified

HTTP Header Fields

- Reference source for header field definitions
  - http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html

- Accept
- Accept-Charset
- Accept-Encoding
- Accept-Language
- Accept-Ranges
- Age
- Allow
- Authorization
- Cache-Control
- Connection
- Content-Encoding
- Content-Language
- Content-Length
- Content-Location
- Content-MD5
- Content-Range
- Content-Type
- Date
- ETag
- Expect
- Expires

HTTP Header Fields

- Host
- If-Match
- If-Modified-Since
- If-None-Match
- If-Range
- If-Unmodified-Since
- Last-Modified
- Location
- Max-Forwards
- Pragma
- Proxy-Authenticate
- Proxy-Authorization
- Range
- Referer
- Retry-After
- Server
- Transfer-Encoding
- Upgrade
- User-Agent
- Vary
- Via
- Warning
- WWW-Authenticate

HTTP connections

Non-persistent HTTP

- At most one object sent over TCP connection
  - connection then closed
- Downloading multiple objects required multiple connections

Persistent HTTP

- Multiple objects can be sent over single TCP connection between client, server

Question?

- Explain in non-technical context
  - E.g. Using radio-taxi

Non-persistent HTTP

suppose user enters URL:

`www.example.net/someexample`

(contains text, references to 10 jpeg images)

1a. HTTP client initiates TCP connection to HTTP server at www.example.net on port 80

1b. HTTP server at host www.someSchool.edu waiting for TCP connection at port 80. "accepts" connection, notifying client

2. HTTP client sends HTTP *request message* (containing URL) into TCP connection socket. Message indicates that client wants object someDepartment/home.index

3. HTTP server receives request message, forms *response message* containing requested object, and sends message into its socket

time

Non-persistent HTTP (cont.)

4. HTTP server closes TCP connection.

5. HTTP client receives response message containing html file, displays html. Parsing html file, finds 10 referenced jpeg objects

time

6. Steps 1-5 repeated for each of 10 jpeg objects

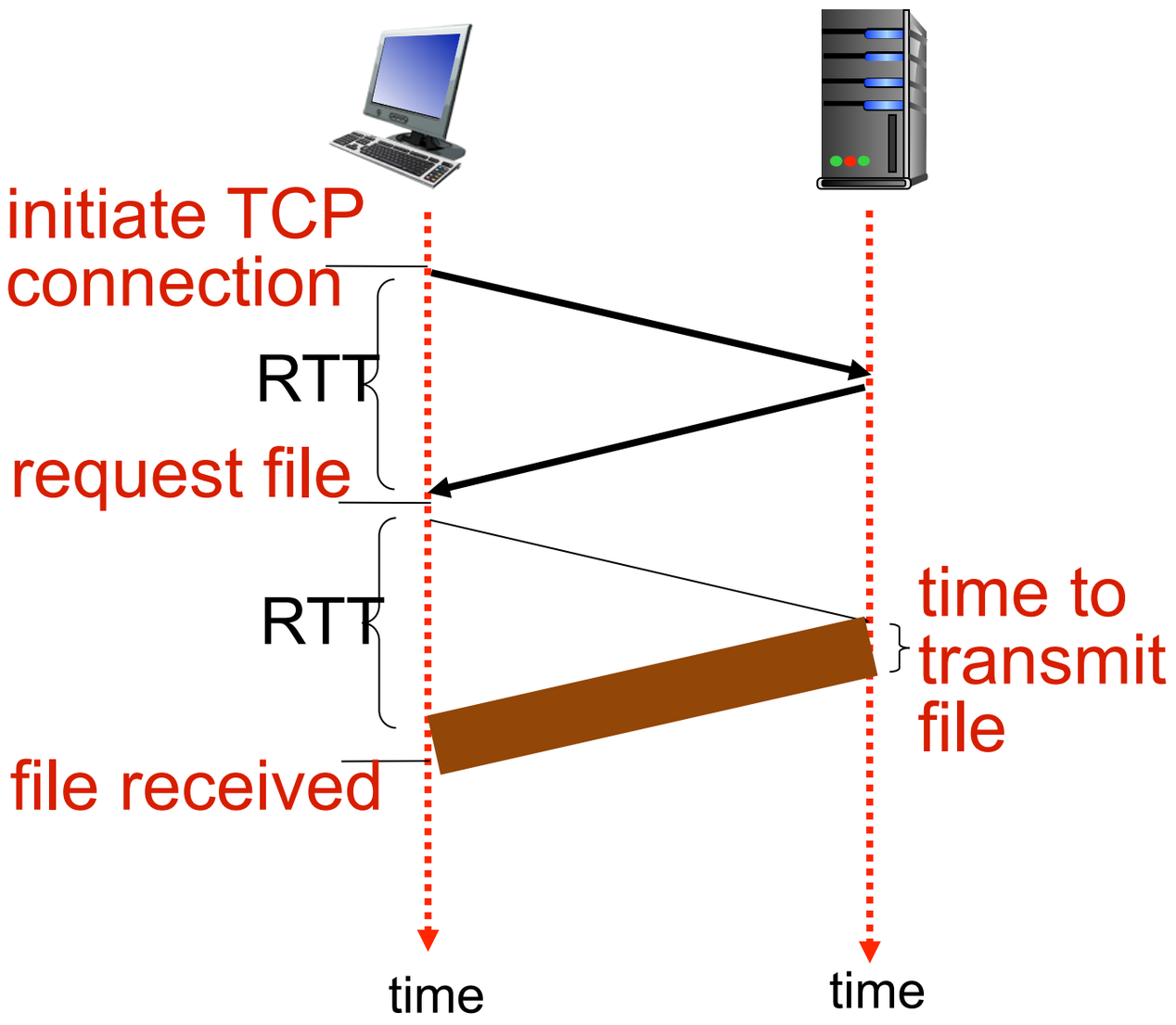
Non-persistent HTTP: response time

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

HTTP response time:

- One RTT to initiate TCP connection
- One RTT for HTTP request and first few bytes of HTTP response to return
- File transmission time
- Non-persistent HTTP response time =

  2RTT+ file transmission time



Persistent HTTP

Non-persistent HTTP issues:

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

Persistent HTTP:

- 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
- As little as one RTT for all the referenced objects

Q: Persistent vs non-Persistent HTTP

- Consider a web page consists of 7 embedded objects.
- Consider that a browser uses 3 parallel connections.
- Consider that RTT time between browser and server
  - 1 second
- Assume that transmission time is zero and display time by the browser after receiving contents is also zero.
- Find out the time taken to display this web page, when
- Browser uses non-persistent HTTP connections?
- Browser uses persistent HTTP Connections?

Persistent Connections

- Apache Config

```
KeepAlive On
MaxKeepAliveRequests 100
KeepAliveTimeout 50
```

- Browser (firefox) config
  - URL "about:config"
  - change the value of (default 6)
    - `network.http.max-persistent-connections-per-server`
- In the browser (firefox) use the URL
  - having some embedded links
  - monitor in wireshark

Trying out HTTP (client side) for yourself

1. **nc** to your favorite Web server:

```
nc rprustagi.com 80
```

opens TCP connection to port 80 (default HTTP server port) at rprustagi.com
anything typed in sent to port 80 at rprustagi.com

2. type in a GET HTTP request:

```
GET /workshops/web/ HTTP/1.1
Host: rprustagi.com
```

by typing this in (hit carriage return twice), you send this minimal (but complete) GET request to HTTP server

3. look at response message sent by HTTP server!
(or use Wireshark to look at captured HTTP request/response)

Summary

- HTTP Headers
  - Connection:
  - Keepalive:
- Non-persistent connections
  - Each request initiates new TCP connection
- Persistent connections
  - one TCP connection serves many HTTP requests
- …