HTTP & Cellular Networks

CS640, 2014-04-09

Outline

- HTTP
- Cellular network structure
- Radio resource control

Web

- Inspired by hypertext -- one document links to another
    - HyperText Markup Language (HTML) -- used to define basic content and layout of a web page
    - Supplemented by CSS, JavaScript, images, documents, Flash/Silverlight, and other files
- Uniform resource locator (URL) specifies location of an object

HyperText Transfer Protocol (HTTP)

- Plain text messages in a request/response sequence -- lines terminated by \r\n
- Request
    - Start line
        - Method to execute
            - GET -- retrieve document
            - HEAD -- retrieve metadata about document
            - POST -- send data to server
        - URL -- may exclude DN and put this in an option
        - HTTP/1.0 or HTTP/1.1
    - Options/parameters
        - User-Agent -- browser name/version, OS name/version
        - Host -- DN portion of URL
        - Cookie
    - Blank, then data (only for methods like POST)
- Reply
    - Start line
        - HTTP/1.0 or HTTP/1.1
        - Status
            - 200 OK
            - 403 Forbidden
            - 404 Not found
            - 301 Moved permanently
            - 418 I’m a teapot
    - Options/parameters
        - Content-Length
        - Content-Type
        - Server -- server name/version
        - Cache-Control -- how long object can be cached
        - Last-Modified
    - Blank, then data

Example Fetching a Web Page

- DNS lookup
- Establish TCP connection
- Send HTTP request
- Receive HTTP reply
- Close TCP connection
- Parse HTML
- Establish TCP connection
- Send HTTP request for image
- Receive HTTP reply for image
- Close TCP connection
- …request other objects in page
- …perform more DNS lookups if objects (e.g. ads) are in different domains (e.g., CDN)
- …render page while other objects are being fetched

Inefficiencies in HTTP

- Problem: Using a separate TCP connection for each object in a web page has a lot of overhead for connection setup and teardown

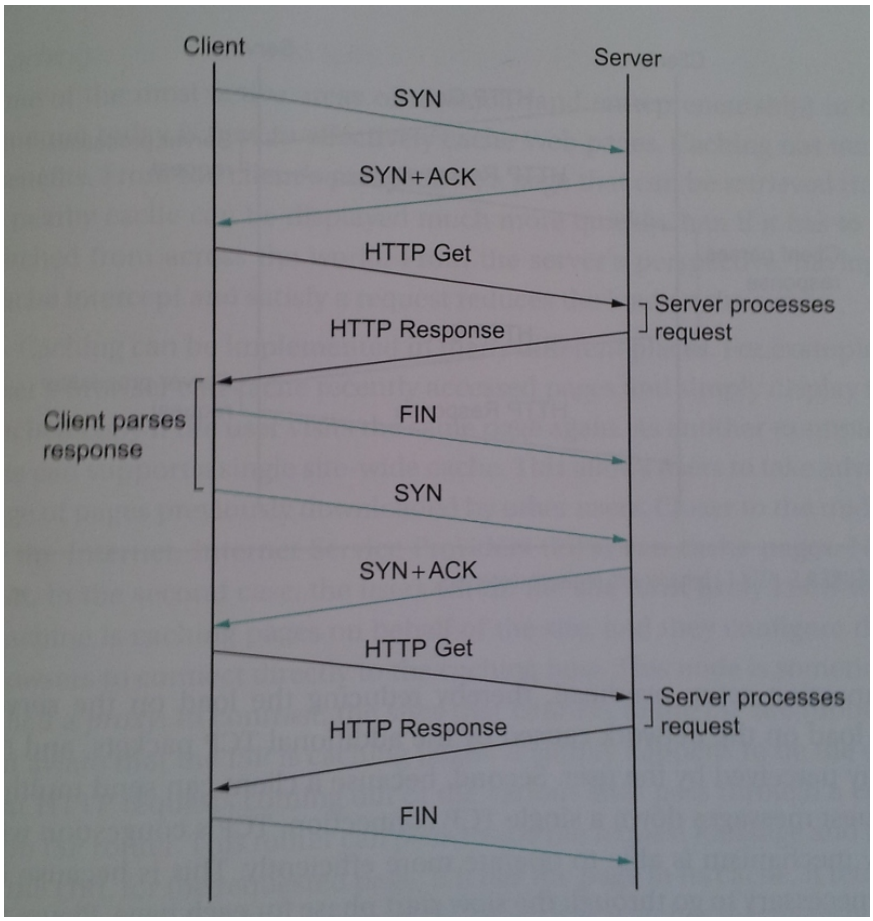
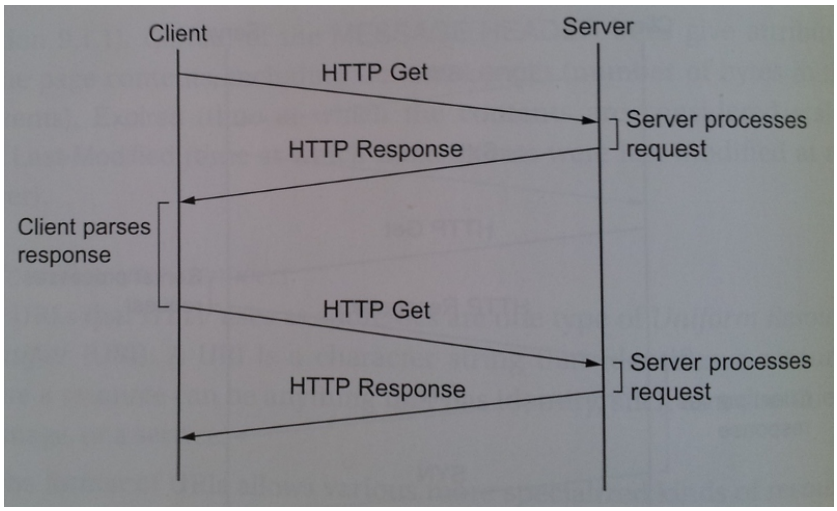


Fig 9.4 pg. 715

  - For each object: 2 RTTs for connection setup plus at least 1 RTT for fetching data

- Solution: HTTP 1.1 introduced persistent connections
    - Exchange multiple request/response messages over the same TCP connection



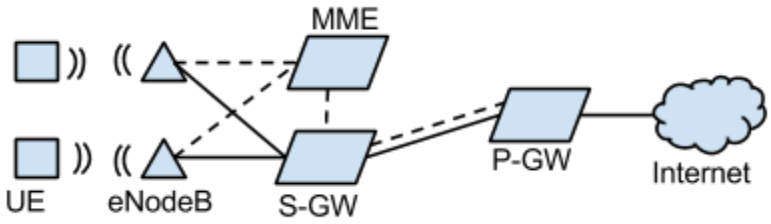
    - Only need to establish one connection to each server providing content for a page
        - If content is coming from multiple servers (e.g., main page and ads come from 2 different domains), you still need > 1 connection
    - Also benefits throughput
        - For each connection, congestion window starts at 1 and is increasingly exponentially using slow start
        - Takes lots of RTTs to reach maximum throughput -- 8 RTTs for 1 Mbps link; 15 RTTs for 100Mbps link
        - Using one connection means initial slow start only occurs once -- still invoke slow start later if timeout occurs due to loss, but ideally losses are handled through fast retransmit/fast recovery where slow start is not invoked
    - Challenge: how long should a connection stay open?
        - Overhead at server to maintain connections for 1000s of clients
        - Throughput benefits far outweigh this overhead
- SPDY
    - Web content transfer protocol designed by Google
    - Further improves upon persistent connection ideas of HTTP 1.1 by allowing multiple requests to be issued at once -- rather than issuing one request, waiting for a response, then issuing the next request, etc.

Web Servers

- Past: mostly in private data centers and public co-location centers
- Today: hosted in public clouds
- Multiple tiers -- front-end, business logic, data store

Cellular Network Structure

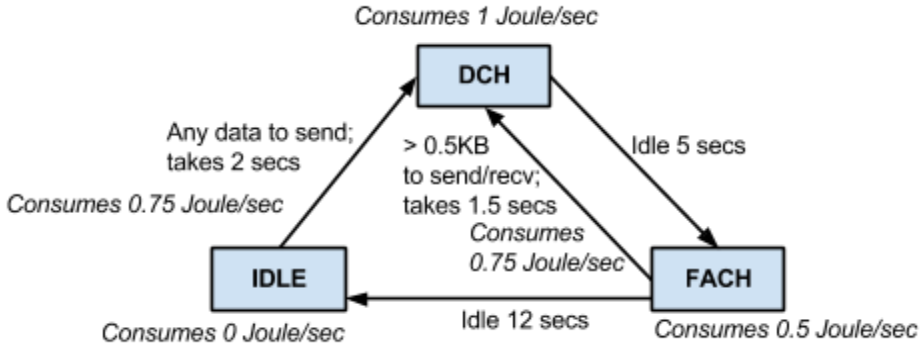
- Standards -- lots of acronyms and versions of standards, and lots of marketing nonsense from cellular service providers
    - 1G
    - 2G -- GPRS was 2.5G, EDGE was 2.75G
    - 3G
    - 3G with HSPA (High Speed Packet Access) -- marketed as 4G, but really 3G with enhancements
    - 4G LTE
- Components



    - UE = user equipment
    - eNodeB = base station
    - MME = Mobility Management Entity -- signaling messages for UE, nodeB, and GWs
    - S-GW = Serving Gateway -- router that forwards between eNodeB and P-GW
    - P-GW = Packet data network Gateway -- communicates with the outside world

Radio Resource Control (RRC)

- Determines how much power UE uses to transmit
- Three states: IDLE, FACH (forward access channel), DCH (dedicated access channel)
    - Power consumption: none, low, high
    - Data rate: none, low, high
- Transition between states based on whether you have data to send and when you last sent data



    - IDLE to DCH if you have any data to send -- takes 2 seconds to change radio to that state
    - DCH to FACH if you are idle for some time (5 seconds)
    - FACH to DCH if you have more than some amount of data queued to send
    - FACH to DCH if you are still idle for some time (12 seconds)
- Trade-off between power consumption and delay
    - Go to FACH or IDLE faster => save more power, but need to add delay to transition back more frequently
- Behavior of applications can really impact power/delay trade-off
    - If you send data every 8 seconds, you'll be in high power mode for 5 seconds, low power mode for 3 second and delay sending of next packet by 1.5 seconds
    - If you send bursts of data every 1 minute, you'll stay in high power mode longer initially, but you'll only transition to low power mode and then idle once => only incur delay at beginning of burst and spend less time in lower power mode
    - Research study examined how real application's decisions impact power consumption and delay in practice
        - Pandora internet radio use to fetch full song when playback of song started, but would fetch advertising image every 60 seconds
        - Incurred cost of going to high power state every 60 seconds
        - If you fetch enough images for duration of song when song is downloaded, then you don't need to go to high power state again and avoiding wasting power sitting in DCH and FACH state
- *Example:*
    - *Assume a device wants to transmit a total of 60KB of data, and the available bandwidth is 80Kbps (i.e., it takes 1s to transmit 10KB of data).*
    - *How many Joules of energy are consumed in 90s if a device wants to transmit 20KB of data at time 0s, 30s, and 60s?*
    - *How many Joules of energy are consumed in 90s if a device wants to transmit 60KB of data at time 0s?*
- Takeaway: layers make things convenient, but they abstract away details that can sometimes be useful