Ch2 Application Layer

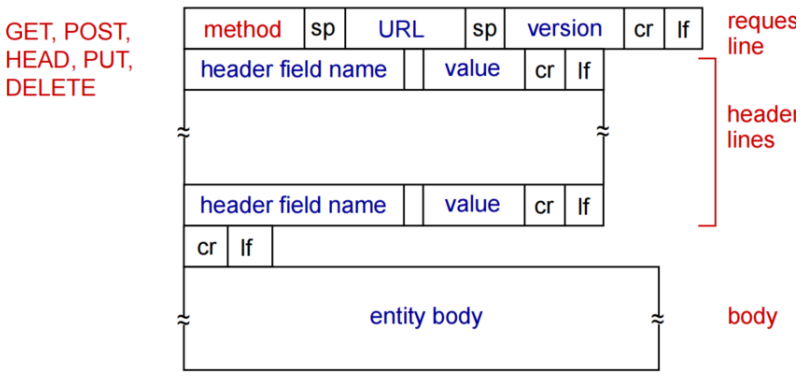
- def:
  - run on end system
  - no need to write software for network core
  - network architecture vs. application architecture
    - network: fixed, 5-layer Internet architecture
    - application: designed by developer, dictates how the app is structured over the various end system.
      - cs & p2p
- network applications
  - architecture: CS or P2P
    - CS: Web or mail
      - Server:
        - always-on
        - permanent IP addr
        - data center
      - Client:
        - intermittently connected
        - dynamic IP addr
    - P2P: P2P file sharing
      - no always-on server
      - arbitrary end system directly communicate
      - self scalability
      - complex management
  - choose transport layer protocol: TCP or UDP
    - How do apps exchange msgs?
      - **Processes**: snd/recv msgs from network
        - same host: inter-process communication
        - different hosts: network
        - client process: initiate communication
        - server process: wait to be contacted
        - controlled by developer
      - **Socket**: processes snd/recv msgs to/from

        - Def: Interface between Process and Computer Networks
        - receive msgs: Socket must be identified by
            - host IP: 32 bits = 4 Bytes
            - server port
    - How to choose **transport services**?
        - 4 broad classes
            - **reliable data transfer**
                - delivered correctly， completely， in proper order
                - file transfer, web transactions
            - **Throughput**
                - 带宽 sensitive: multimedia
                - mail
            - **Timing**
                - real time: internet phone, interactive games
            - **Security**
        - TCP
            - connection-oriented
            - reliable: without error
            - congestion control: 控制发送的速率
            - no timming
            - no minimum throughput guarantee
            - no security
                - TCP+SSL (app layer)
        - UDP
            - connectionless
            - unreliable: error, loss
            - no congestion control
            - no timing
            - no throughput guarantee
            - no security
- application protocol: HTTP for Web, SMTP for email

- **Web & HTTP (Hypertext Transfer Protocol)**
    - Web
        - consists of objects
        - base HTML + referenced objects addressed by URL(host name + path name)

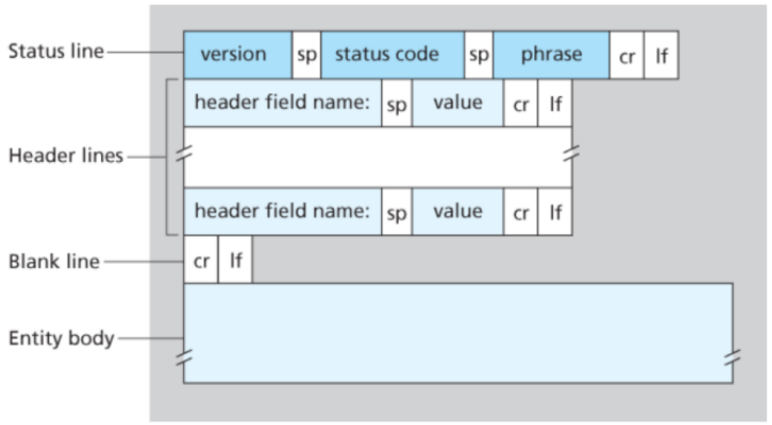
- **HTML: Hypertext markup language 超文本标记语言**
- use HTTP as application layer protocol

- HTTP
  - overview
    - TCP
      - C: initiates TCP con(socket), dst port=80
      - S: accept TCP con
    - stateless
      - 不记忆过去request信息
    - persistent & non-persistent
      - non
        - 每一次发送请求都要确认一遍连接 2RTT + T_trans
      - persistent
        - 一次连接发多个object
  - request & response
    - request



For example, the entity body is used with the POST method (e.g., search words to a search engine).

      - \r carriage return 回车：光标回到当前行起始位置
      - \n line-feed 换行：光标到下一行
      - get: download file, visit Web
      - post: register, preview
      - head: get head info, check modify date, debugging
      - put: upload file
      - delete
    - response



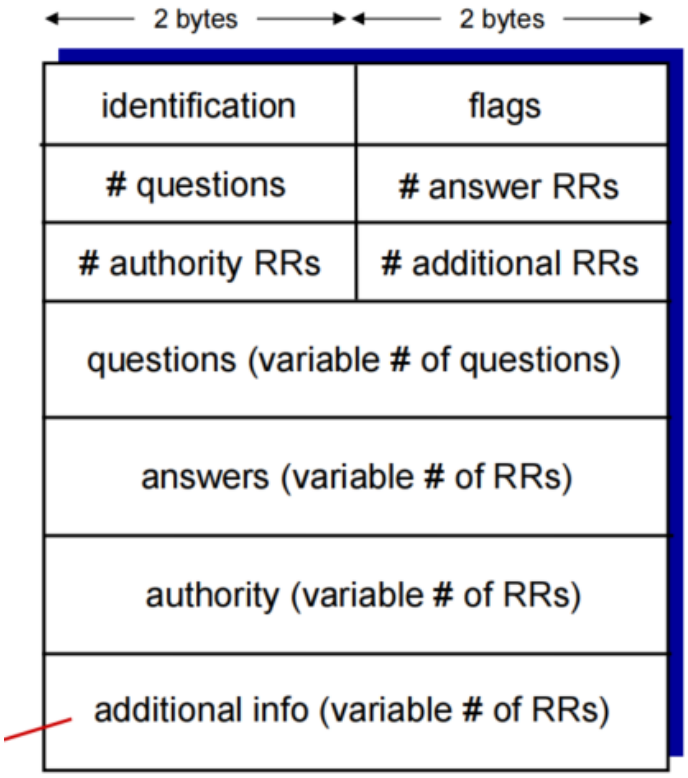
    - 200 ok
    - 301 moved permanently
    - 400 bad request: not understood by server
    - 404 not found on server
    - 505: http version
- cookies
    - a server must handle thousands of TCP connection simultaneously
    - 在4个地方
        - response header line
        - next request header line
        - store on user's host
        - back-end database at Web server
    - state
    - use for
        - authorization
        - shopping carts
        - recommendations
        - user session on top of stateless HTTP
- web caching: proxy server
    - 不用每次都请求origin server
    - 流程
        - browser 与Web cache建立TCP连接， send HTTP request
        - Web cache checks object 是否存在
            - 在：return
            - 不在： open TCP connection to origin server, request and get the object and send back to the browser within an HTTP response
            - copy

        - ISP: Company, university
        - both server and client
        - benefit:
            - reduce response time
            - reduce traffic in access link and whole internet
            - reduce costs for upgrade bandwidth
        - Conditional get: If-Modified-Since(request)+ date
            - 每次browser request
        - 计算题
- mail
    - overview
        - 3 major components
            - user agents
                - for user to read mail
                - Outlook
            - mail servers
                - user mailbox
                - msg queue
            - SMTP(use TCP transmission control protocol)
                - between mail servers to send email msgs
                - both client and server sides of SMTP run on mail server
                - ??
    - **SMTP: Simple Mail Transfer Protocol**
        - using TCP, port = 25: 1.5 RTT to connection
        - HELLO, MAIL FROM, RCPT TO, DATA, QUIT
        - alice--> sender's server & sender's server --> receiver's server
        - 3 phases for transfer
            - handshaking: 1.5 RTT
            - transfer of msgs
            - closure
        - msgs
            - commands: text
            - response: status code & phrase
        - SMTP vs. HTTP
            - HTTP: pull, ASCII in header

        - SMTP: push, ASCII in header and body, multiple objects sent in one msg
    - !
        - 要有server：用户关机就无法传输，丢失
        - 有两个server：压力不会太大，重复发送直到成功
- Mail Message Format
    - header
        - to
        - from
        - subject
    - blank line
    - body
- Mail Access Protocol
    - the receiver obtain his msgs
    - **POP3: Post Office Protocol 3**
        - TCP, post=110, download and delete mode, stateless
        - authorization: user+pass, get +OK/-ERR
        - transaction: list, retr, dele, Quit
        - update: 自动delete
    - **IMAP: Internet Mail Access Protocol**
        - folder, state
    - HTTP
        - browser
- **DNS: Domain name system**
    - services
        - host name --> IP
        - alias host name --> canonical
        - mail server aliasing
        - load distribution
            - replicated Web servers
        - 流程
            - application invokes DNS ans specifying the host name
            - DNS sends a query into the network and get reply to port 53
            - pass the mapping to the application
            - UDP: faster and smaller data pkt
    - structure

- hierarchical & distributed
  - why not centralize DNS
    - single point failure
    - traffic
    - distance
    - maintenance: update freq
  - Root DNS: find IP addr of .com/.org/.edu TLD DNS servers
  - **TLD: Top-level DNS**: to get google.com authoritative DNS server
  - authoritative: to get IP address
  - Local DNS server:
    - does not belong to hierarchy
    - When a host connects to an ISP, the ISP provides the IP addresses of one or more of local DNS servers
- iterated and recursive
  - iterated query: 返回其他可以询问的服务器的name
  - recursive: 递归查询，直到找到结果
- cache, timeout after some time

- protocol
  - DNS record: distributed database storing **resource records (RR)**
    - **A:** name(host name), value(IP addr)
      - 一个host的权威服务器一定包含它的A record
      - 不是权威服务器可能会包含A record在cache中
    - **NS**: name(domain), value(host name of auth. server for this domain)
      - 非权威服务器会包含
    - **CNAME**: name(alias), value(canonical)
    - **MX**: name(alias), value(canonical of mailserver)
  - query and reply



- insert record into DNS

```
(networkutopia.com, dns1.networkutopia.com, NS)

(dns1.networkutopia.com, 212.212.212.1, A)
```

- P2P
  - Def. no always on server, directly connect
    - file distribution(BitTorrent), streaming, VoIP
  - P2P vs. CS: File distribution time 计算题
  - BitTorrent
    - protocol for file distribution
    - torrent：参与分发的peers构成的集合，每个peer上传或者下载chunks 256kb，并且4可以随时加入或者离开
    - each torrent都有一个tracker：管理peer是否存在，Obtain list of peers
    - 当一个peer加入torrent时会向tracker register，然后tracker send a subset of peers IP addr to the peer. TCP con and become neighboring peers
    - peer 会定期30s向它的 nbr 询问有什么chunks（TCP）
    - which chunk should she request first
      - ask for list, periodically
      - request missing chunks, **rarest first**
    - to which should she send request chunks?
      - **tit-for-tat**: 选择 sending her chunks at highest rate
      - every 30 s randomly select 1 additional peer and send chunks

- video streaming and context distribution network
  - different users have different capabilities
  - video streaming
    - video basics
      - sequence of images displayed at constant rate (FPS)
      - array of pixels: bits
      - Coding (Compression): use *redundancy within and between* images to decrease # bits used to encode image: spatial & temporal
      - **CBR(constant bit rate)**
      - **VBR(variable bit rate)**
    - http streaming
      - 所有client都会收到相同的encoding，用DASH解决
      - **DASH: Dynamic Adaptive Streaming over HTTP**
      - Server:
        - divide video into chunks
        - each chunk store, encode at diff rate
        - manifest file: list of URL for diff chunks encoded at diff rate
      - Client:
        - 定期测量SC位宽
        - consult manifest file, request 1 chunk
    - adaptive streaming over http
  - Content distribution networks (CDN)
    - 这么多视频同时发送这么多用户
    - 建立多个视频副本在多个地方
    - cluster selection strategy
      - geographically closest
      - Periodic **real-time measurements** of delay and loss performance
- UDP & TCP