# Computer Networking

Jihong Gan

## Contents

**0 Front Matter** **1**
- 0.1 Schedule . . . 1
- 0.2 Motivating Questions . . . 2
- 0.3 References . . . 2

**1 Computer Networks and the internet** **3**
- 1.1 What is the internet? . . . 3
  - 1.1.1 Nuts n bolts definition . . . 3
  - 1.1.2 Service definition . . . 4
- 1.2 The network edge . . . 4
- 1.3 The Network Core . . . 4
  - 1.3.1 Packet Switching . . . 4
  - 1.3.2 A network of networks . . . 5
- 1.4 Delay, Loss, and Throughput . . . 5
  - 1.4.1 Types of delay . . . 5
  - 1.4.2 Queueing delay . . . 5
  - 1.4.3 Throughput . . . 6
- 1.5 Protocol layers and their service models . . . 6
  - 1.5.1 Internet protocol stack . . . 6
  - 1.5.2 Encapsulation . . . 7

**2 Application Layer** **7**
- 2.1 Principles of Network Applications . . . 7
- 2.2 The Web and HTTP . . . 8
  - 2.2.1 Overview . . . 8
  - 2.2.2 Non-Persistent vs. Persistent Connections . . . 8
  - 2.2.3 HTTP message format . . . 9
  - 2.2.4 Cookies . . . 10
  - 2.2.5 Web Caching . . . 10
  - 2.2.6 HTTP/2 . . . 10
- 2.3 Domain Name System DNS . . . 11

## 0 Front Matter

Notes on computer networking.

### 0.1 Schedule

| Week of | Monday | Wednesday | Readings | Assignments |
|---|---|---|---|---|
| 01/02/23 | **No Class** | [Introduction and Overview] | 1.1, 1.3, 1.4 | A1 Out |
| 01/09/23 | [Protocol Layering] | [HTTP and the Web] | 1.5 | |
| 01/16/23 | **No Class** | [DNS and CDN] | 2.2, 2.4 | |
| 01/23/23 | [Video Streaming and Cloud Systems] | [Transport Layer] | 2.6, 3.1, 3.2, 3.3, 3.4 | A1 Due, A2 Out |
| 01/30/23 | [TCP Basics] | [Flow and Congestion Control] | 3.5, 3.6 | |
| 02/06/23 | [More Congestion Control] | [Network Layer and IP] | 3.7, 4.1, 4.3.1, 4.3.2, 4.3.5 | |
| 02/13/23 | [IP Routers] | [*Midterm Review*] | 4.2 | |
| 02/20/23 | **No Class** | **MIDTERM** | | A2 Due |
| 02/27/23 | **No Class** | **No Class** | | A3 Out |
| 03/06/23 | [Routing Fundamentals] | [Intra-AS Routing] | 5.1, 5.2, 5.3 | |
| 03/13/23 | [IP Addressing and Inter-AS Routing] | [BGP] | 4.3.3, 5.4 | |
| 03/20/23 | [Software-Defined Networking] | [Link Layer] | 4.4, 5.5, 6.1, 6.3 | A3 Due, A4 Out |
| 03/27/23 | [Switched LAN] | **No Class** | 6.4 | |
| 04/03/23 | [Wireless Networking] | [Datacenter Networking] | 6.6, 7.1, 7.2, 7.3 | |
| 04/10/23 | [*Final Review*] | **No Class** | | |
| 04/17/23 | **No Class** | | | A4 Due |
| 04/21/23 | **FINAL** | | | |

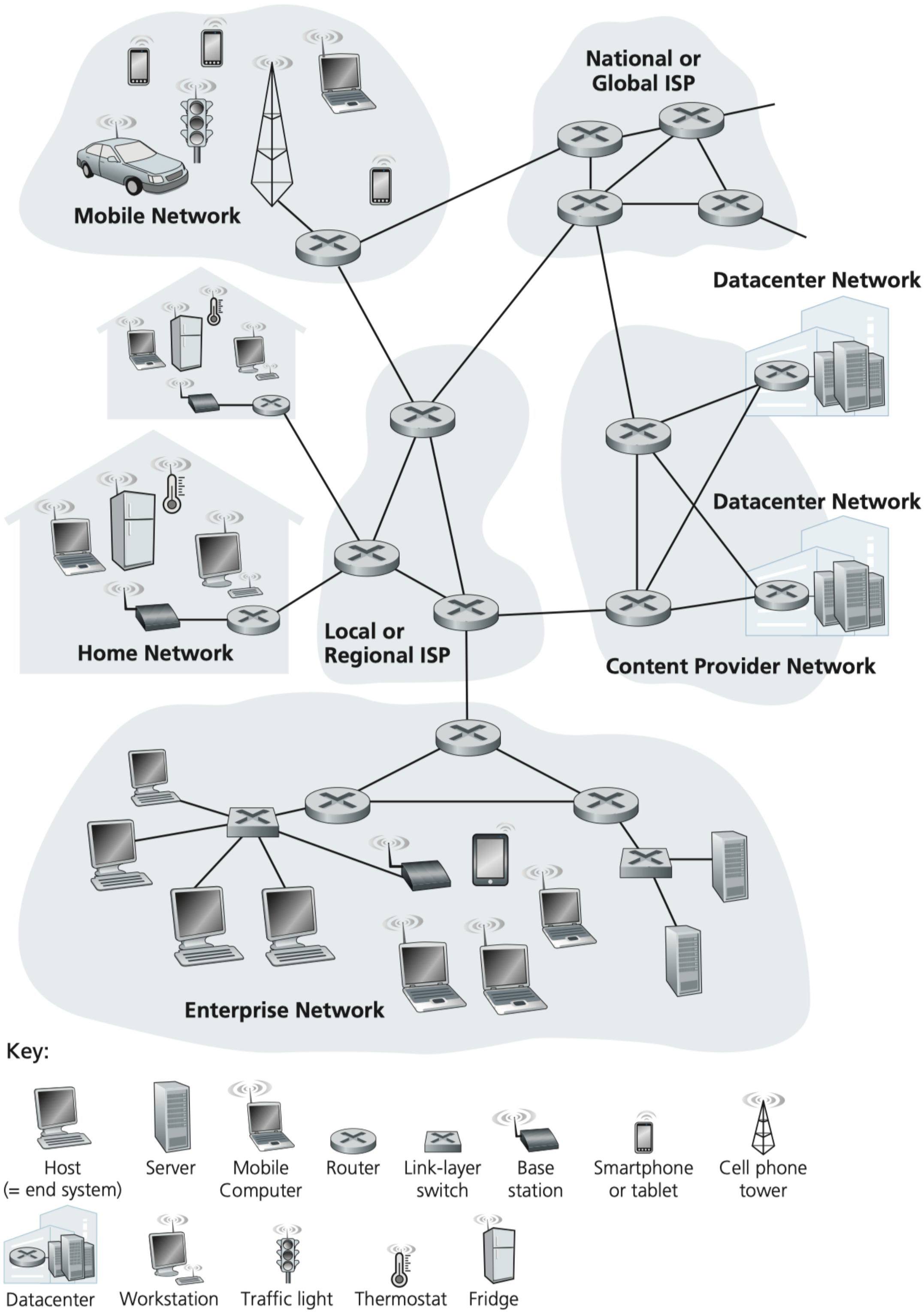
## 0.2 Motivating Questions

- If I get to design the internet, what would I do?
- What happens when I type an URL into a browser?
- What happens when I search on Google?
- How does p2p downloading work, and how do people get caught pirating with p2p?
- How do I mess with the website of my high school?
- How does TCP/IP work?
- To what extent can my privacy be at risk on the internet? How?
- How does VPN work?
- How does the GFW work, and how to bypass it?
- What does it take to build a social networking protocol?

## 0.3 References


- [KR20] *Computer Networking: A Top-Down Approach, 8th Edition*, Kurose and Ross, 2020.
- [PD21] *Computer Networks: A Systems Approach, 6th Edition*, Peterson and Davie, 2021.

# 1 Computer Networks and the internet




## 1.1 What is the internet?

### 1.1.1 Nuts n bolts definition

- a computer network that interconnects billions of computing devices (**Hosts/end systems**) throughout the world.

- End systems are connected together by a network of **communication links** and **packet switches**.
  - Communication links examples: copper wire, optical fiber, radio spectrum.
  - A packet switch takes packets from one of its communication links to another. Eg. routers (in the network core), link-layer switches(like wifi router and modem).
- **Packet**: segment of data with header sent and reassembled.
- End systems access the Internet through **Internet Service Providers (ISPs)**. Each ISP is in itself a network of packet switches and communication links.
  - Eg. local cable company, university ISPs
  - Eg. lower-tier ISPs are interconnected by upper-tier ISPs such as AT&T
- End systems, packet switches, and other pieces of the Internet run **protocols** that decides how they talk to each other
  - Eg. **TCP/IP**

#### 1.1.2 Service definition

- an infrastructure that provides services to applications running on end systems
  - eg. supports sending email, browsing web, streaming videos

### 1.2 The network edge

- Access networks: networks that connect end systems to the first router
  - mobile network: 5G phone -> cell tower -> ISP router
  - Enterprise network
  - Home network: TV -> modem -> ISP router
  - Datacenter network
  - Empowered by physical media
    * eg. cables, radio channels, fiber optics
- The internet puts most of its complexity in its periphery

### 1.3 The Network Core

- Def. The mesh of packet switches and communication links that interconnects the access networks
- Two ways of moving data through networks: packet switching and circuit switching
  - packet switching is simpler and has more transmission capacity.

#### 1.3.1 Packet Switching

- Packet switches uses **store-and-forward transmission**
  - packet switches must receive the entire packet and process it before it forwards its first bit to the outbound link.
- The packets are pipelined through the path because of store-and-forward
  - The delay of sending $P$ packets of $L$ bits each over a path of $N$ links each of rate $R$ is $d = (N + P - 1)L/R$
- Since it takes time to put packets onto the link, routers have **output buffers** that stores packets before they are sent out.
  - When the buffer is full, the **packet loss** occurs
- Routers figures out where a packet should go via a **forwarding table**, which maps (portions of) the destination IP address to the router's outgoing link
  - The forwarding table is set automatically by **routing protocols**.

### 1.3.2 A network of networks

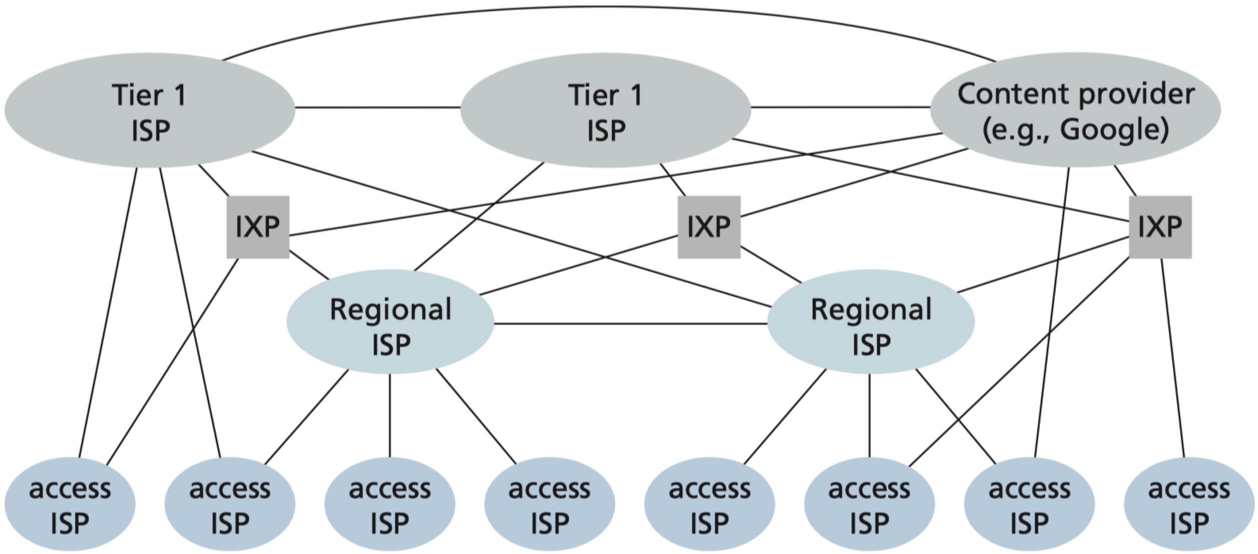



**Figure 1.15** ♦ Interconnection of ISPs

## 1.4 Delay, Loss, and Throughput

### 1.4.1 Types of delay

- Processing delay: time to process the packet at the router
  - microseconds or less
- Queuing delay
  - microseconds to milliseconds
- Propagation delay: time for the packet to travel on the link
  - milliseconds in wide area networks
- Transmission delay
  - `# bits / transmission rate`
- **Total nodal delay** is the sum of the four above.
  - Its exact composition can vary significantly

### 1.4.2 Queueing delay

- Depends on the rate at which packets arrive at the queue, the transmission rate of the link, and the nature of the traffic.
- Estimated by **traffic intensity**.
  - Suppose that the packets arrive at the queue at $a$ bits/sec, all packets consist of $L$ bits, and the transmission rate of the link is $R$ bits/sec. The traffic intensity is $aL/R$.
  - If $aL/R > 1$, the queue will grow without bound
- Typically, the arrivals to a queue do not follow any pattern. So, queueing delay grows exponentially with the traffic intensity.

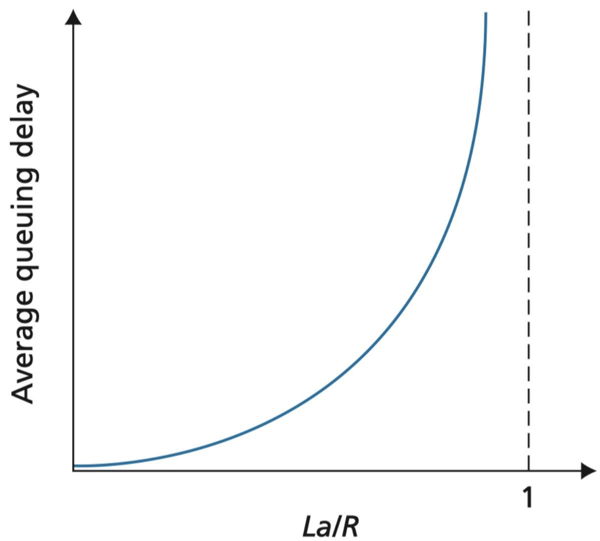



**Figure 1.18** ♦ Dependence of average queuing delay on traffic intensity

### 1.4.3 Throughput

- when there is no other intervening traffic, the throughput can simply be approximated as the minimum transmission rate along the path between source and destination.
- more generally the throughput depends not only on the transmission rates of the links along the path, but also on the intervening traffic.

## 1.5 Protocol layers and their service models

- Network protocols are organized in layers.
- Each layer provides a service (**service model**) to the layer above it and uses the service of the layer below it.
- A protocol layer can be implemented in software or hardware or both.
- A layer n protocol distributed among the end systems, packet switches, and other components that make up the network.

### 1.5.1 Internet protocol stack

| ˆ |
|---|
| Application |
| Transport |
| Network |
| Link |
| Physical |

- **Application layer**: where network applications and their protocols reside
  - Eg. HTTP, FTP, DNS, SMTP
  - **Message**: application layer packet
- **Transport layer** transports messages between applications
  - Eg. TCP, UDP

– **Segment**: transport layer packet
- **Network layer** routes segments between hosts
  – Eg. IP, routing protocols
  – **Datagram**: network layer packet
- **Link layer** delivers datagrams from one node (host or router) to another
  – Eg. Ethernet, WiFi
  – **Frame**: link layer packet
- **Physical layer** moves bits within the frame from one node to another
  – Eg. copper wire, optical fiber, radio spectrum

### 1.5.2 Encapsulation

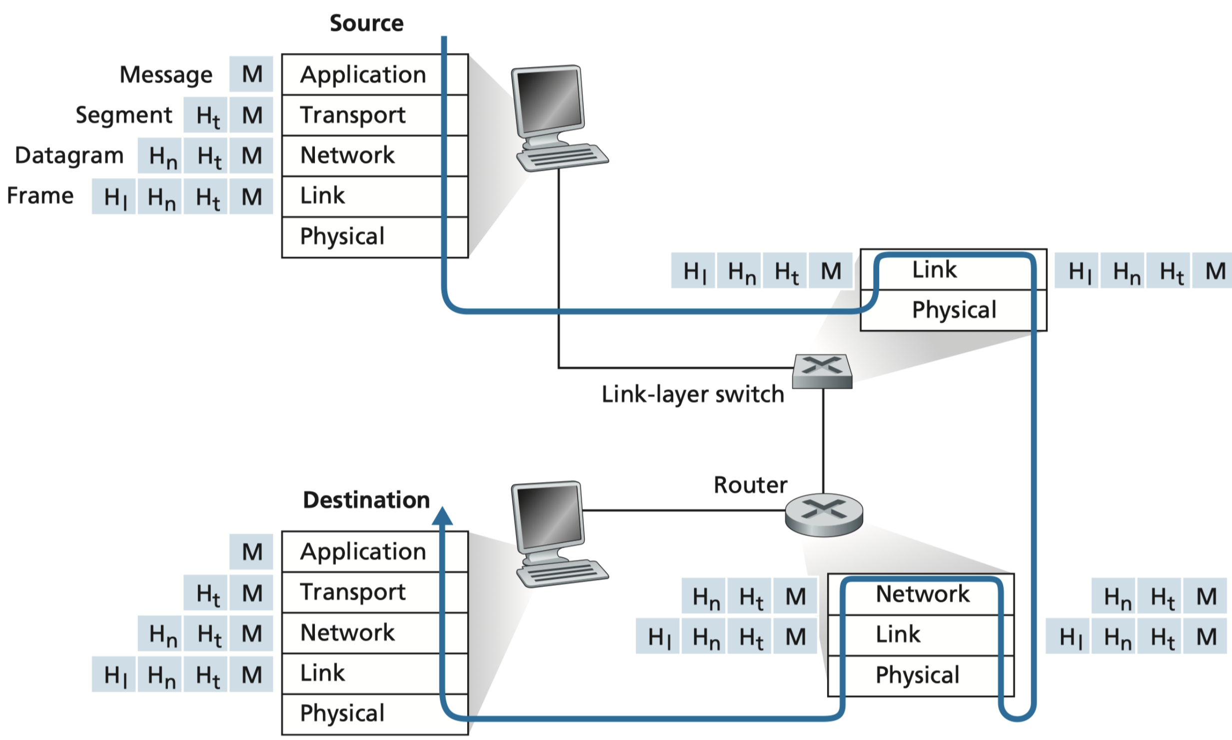



**Figure 1.24** ♦ Hosts, routers, and link-layer switches; each contains a different set of layers, reflecting their differences in functionality

- Link-layer switches and routers
  – both only implement the bottom layers
  – Routers implement the network layer but link-layer switches do not
- At each layer, a packet has two types of fields: header fields and a **payload field**. The payload is typically a packet from the layer above.

# 2 Application Layer

## 2.1 Principles of Network Applications

- Architectures
  – **Client-server**
    * Eg. the Web, email
    * Hosts do not communicate with each other

- **Peer-to-peer**
  - Eg. BitTorrent, Skype
  - Hosts communicate to each other with no servers involved
- In either architecture, an application consists of *pairs of processes* that send *messages* to each other over a network.
  - By convention, the process that initiates the communication is called the **client** and the other process is called the **server**.
    - Eg. In the Web, Web browser (client) initiates the communication with the Web server.
    - Eg. In p2p, each process is both client and server.
- Processes interacts with the network via **sockets**, the interface between application layer and transport layer.
  - A socket is identified by a pair of IP address and port number.

## 2.2 The Web and HTTP

### 2.2.1 Overview

- The Web is a client-server app that uses HTTP as its application-layer protocol. Client and server processes talk to each other via HTTP messages.
- A **web page** consists of objects, files like HTML files and images. A web page usually consists of a **base HTML file** which references other objects via URLs. **Web browsers** implement the client side of HTTP.
- HTTP is built on top of TCP
- HTTP is a **stateless** protocol. HTTP servers store no information about clients.

### 2.2.2 Non-Persistent vs. Persistent Connections

- Internet app developers must decide should each request/response pair be a separate TCP connection or should all messages be sent over a single TCP connection. The former is called **non-persistent connection** and the latter is called **persistent connection**. HTTP can be used in both ways.
- Non-persistent HTTP sets up a TCP connection for exactly one request message and response pair. So retrieving a web page with 10 objects requires 10 TCP connections, which is a big overhead.
- Each TCP connection takes 2RTT due to three-way handshake, one RTT to establish the connection and one RTT for messaging, another big overhead.
- With HTTP/1.1 persistent connection, server leaves the connection open after responding. Multiple web pages can be retrieved over the same connection. The default mode HTTP uses persistent connection with pipelining.

### 2.2.3 HTTP message format

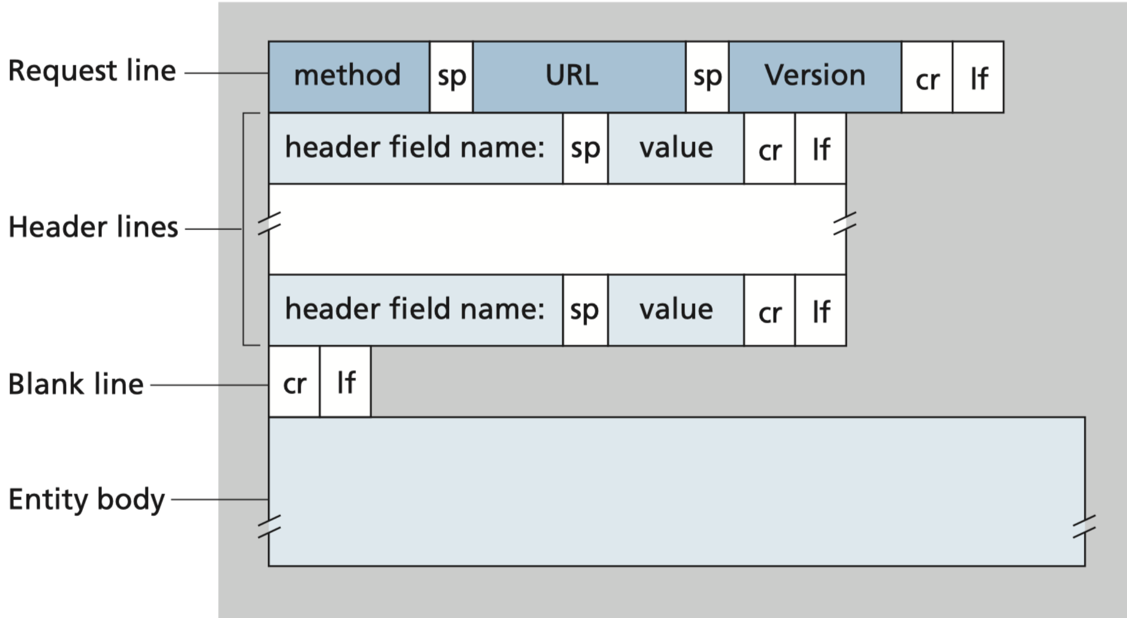



**Figure 2.8** ♦ General format of an HTTP request message

- Request

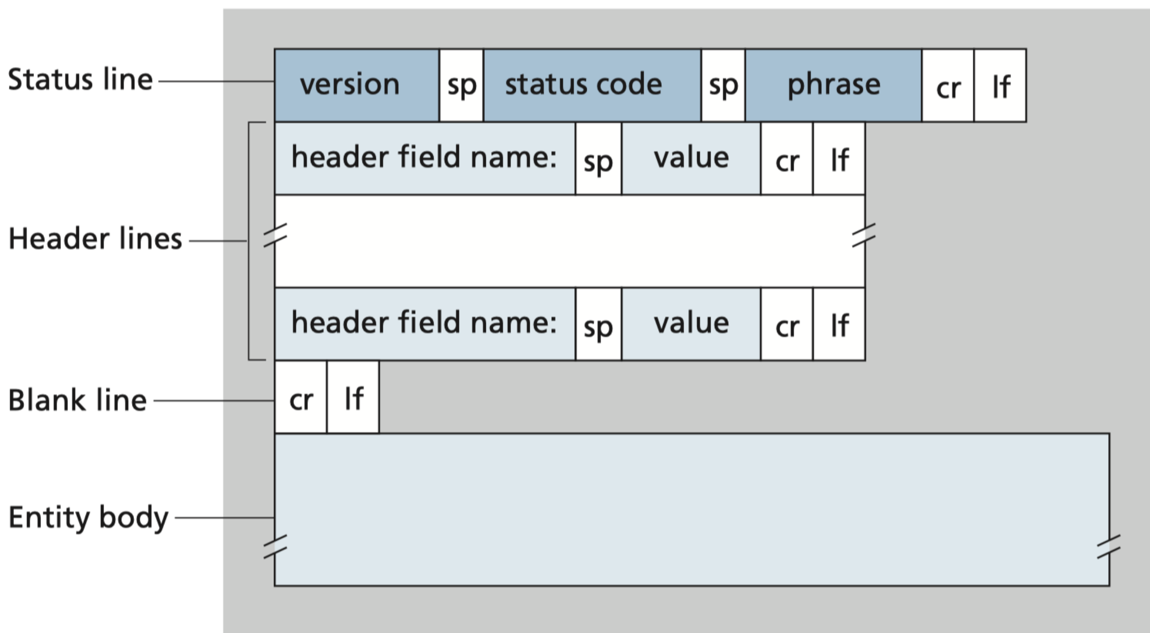



**Figure 2.9** ♦ General format of an HTTP response message

- Response

### 2.2.4 Cookies

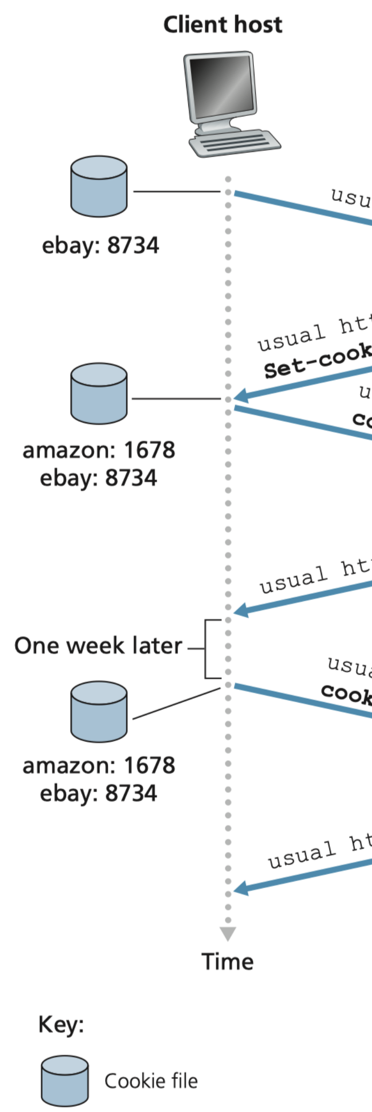



**Figure 2**

- Allows web sites to identify users. Creates a user session on top of stateless HTTP.

### 2.2.5 Web Caching

- A Web cache—also called a proxy server—is a network entity that satisfies HTTP requests on the behalf of an origin Web server. The Web cache has its own disk storage and keeps copies of recently requested objects in this storage.
- Benefits
  1. Shorten response time
  2. Reducing traffic on an institution's access link to the Internet. Reducing web traffic in the Internet as a whole. Localizing traffic
- Eg. CDNs Content distribution networks
- Handle potentially stale objects with conditional GETs

### 2.2.6 HTTP/2

- Problem with HTTP/1.1

  - Sending all objects through a persistent TCP connection causes **Head of Line (HOL) blocking**: larger objects in an HTML page blocks the rest.
  - Clients motivated to open parallel TCP connections for the same web page to gain an unfair share of the bandwidth.
- HTTP/2 solutions
  - Banning parallel TCP connections for a single webpage
  - Breaking the HTTP message into smaller chunks called **frames**. Multiplexing (time-sharing) the connection among frames.

### 2.3 Domain Name System DNS

- What
  - A distributed database implemented in a hierarchy of **DNS servers**.
  - An application-layer protocol that allows hosts to query the database.
- Services
  - Translate domain names to IP addresses
  - Host aliasing
  - Load distribution
- How DNS works
  - All DNS messages are sent over UDP port 53
  - A hierarchy of servers
    * Root DNS servers provide the IP address of the top-level domain (TLD) servers
    * TLD servers (eg. .com, .edu, .org) provide the IP address of the authoritative DNS servers
    * Authoritative DNS servers provide the IP address of the publicly accessible hosts
  - Local DNS servers
    * Forwards DNS queries to the DNS servers hierarchy
    * Cache the results of previous queries
  - DNS queries can be iterative or recursive