Web and Intranet Performance Issues

Learning Objectives

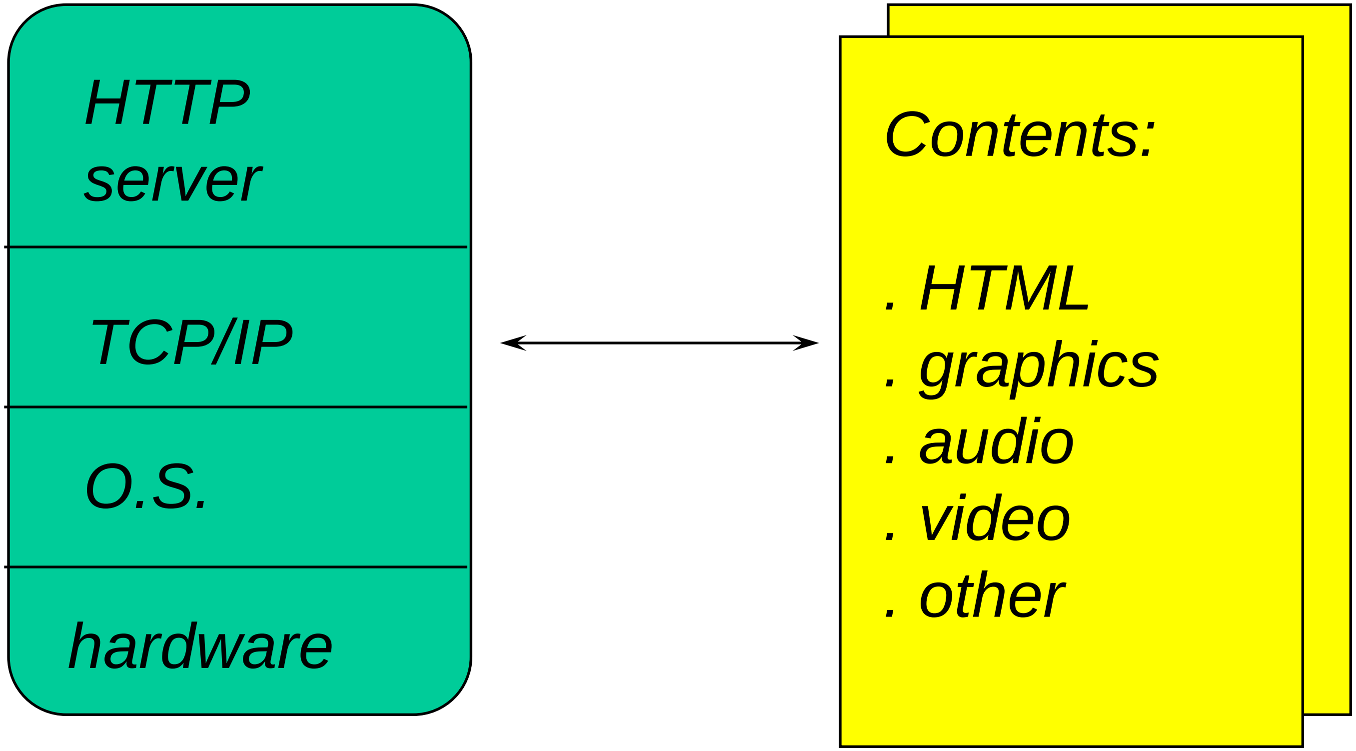
- Server architectures and performance issues
- Web infrastructure components
- Web server workloads
- Bandwidth, latency, and traffic in Web applications
- Capacity planning issues

Web Server Performance Issues

Unpredictable nature of information retrieval and service request over the World-Wide web

- load spikes: 8 to 10 greater than avg.
- high variability of document sizes: from 10^3 to 10^7 bytes

Web Server Components



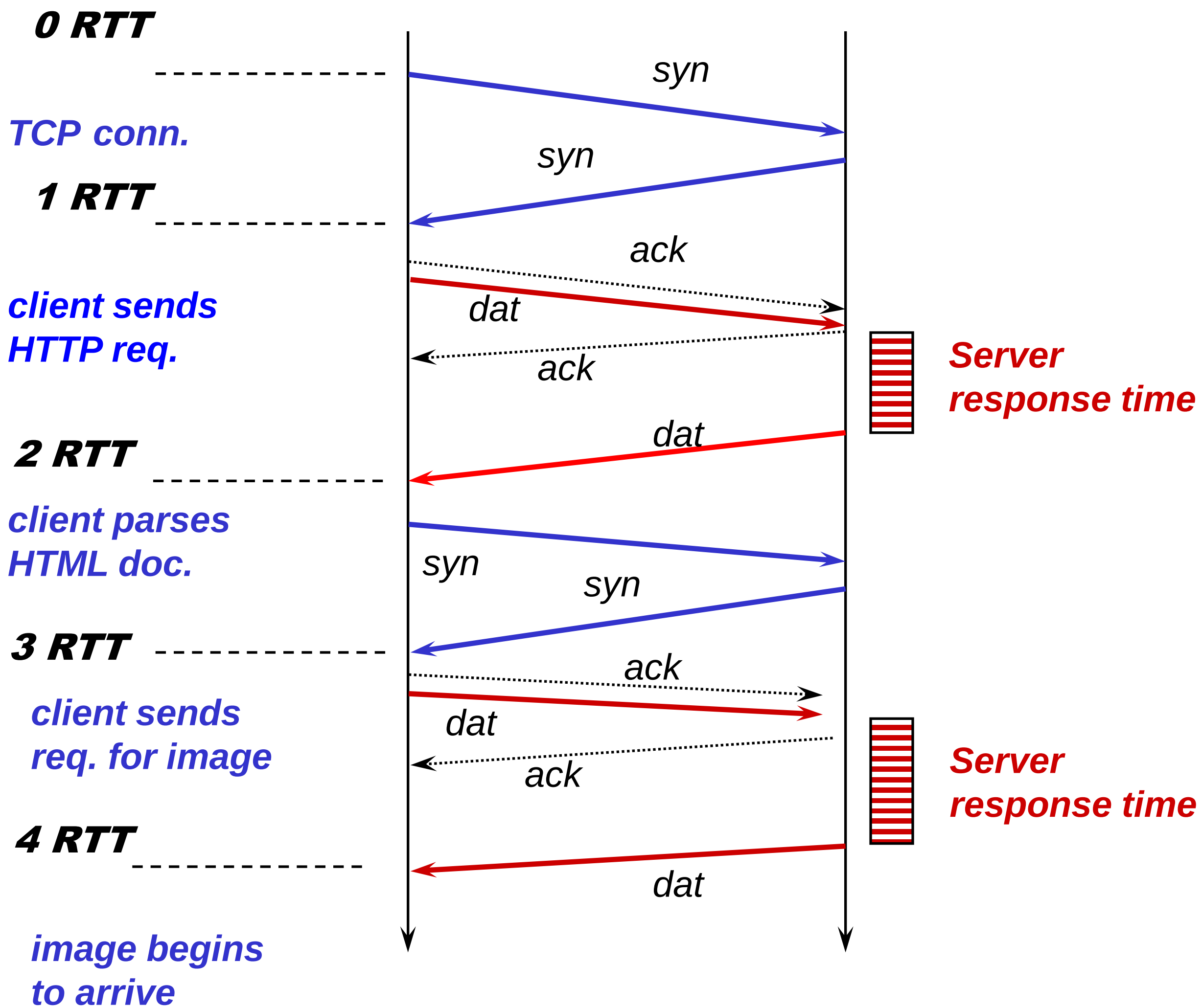
Combination of HTTP and TCP/IP

- HTTP defines a request-response interaction;
- HTTP is a ``stateless’’ protocol;
- One connection per object;
- TCP connection setup overhead;
- Delays due to protocols;
- Small Web objects and TCP ``”slow start’’ algorithm

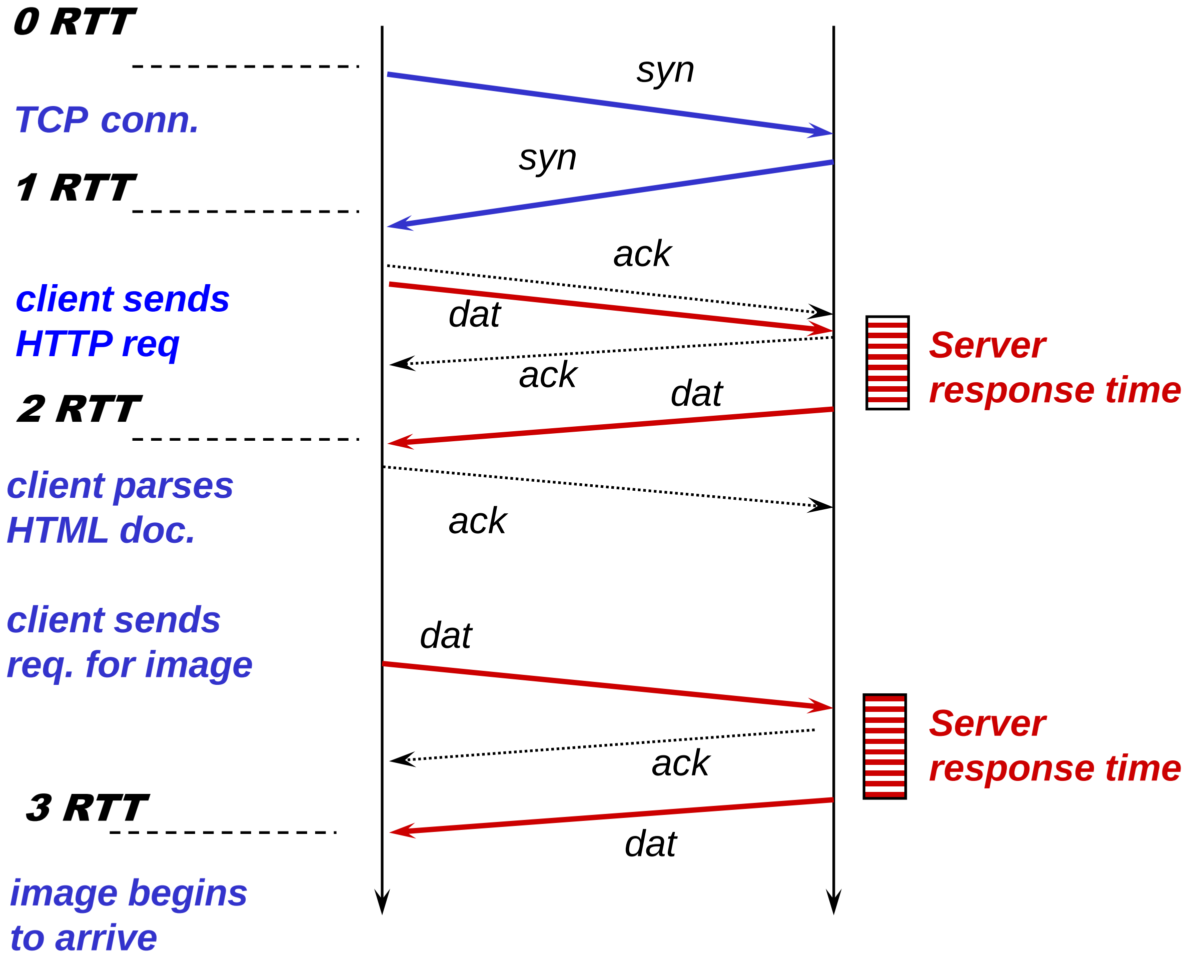
HTTP request-response steps

- map the server to an IP address;
- establish a TCP/IP connection with the server;
- transmit the request (URL, method, etc);
- receive the response (HTML text or other information);
- close the TCP/IP connection.

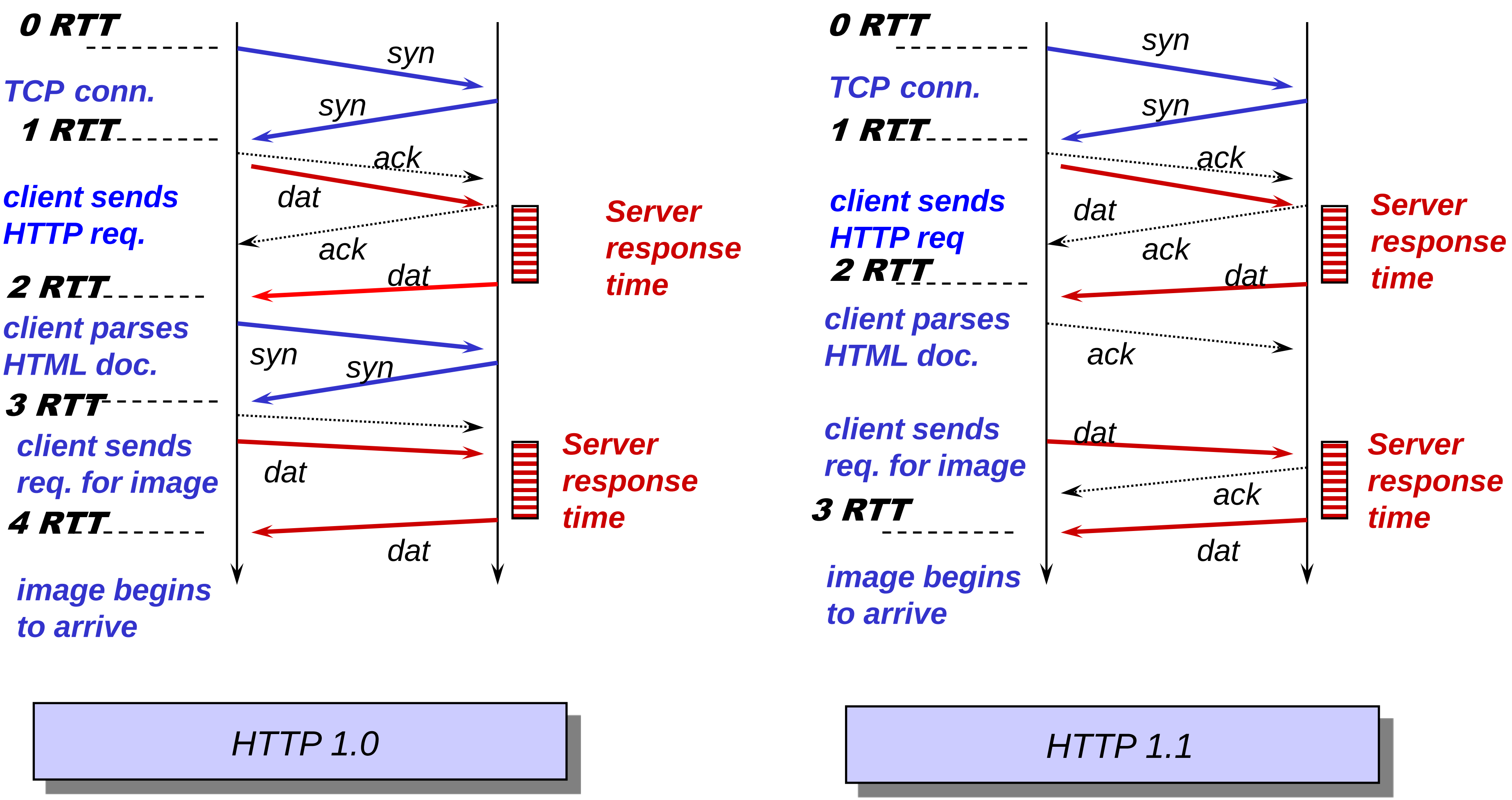
HTTP 1.0 interaction



HTTP 1.1 interaction



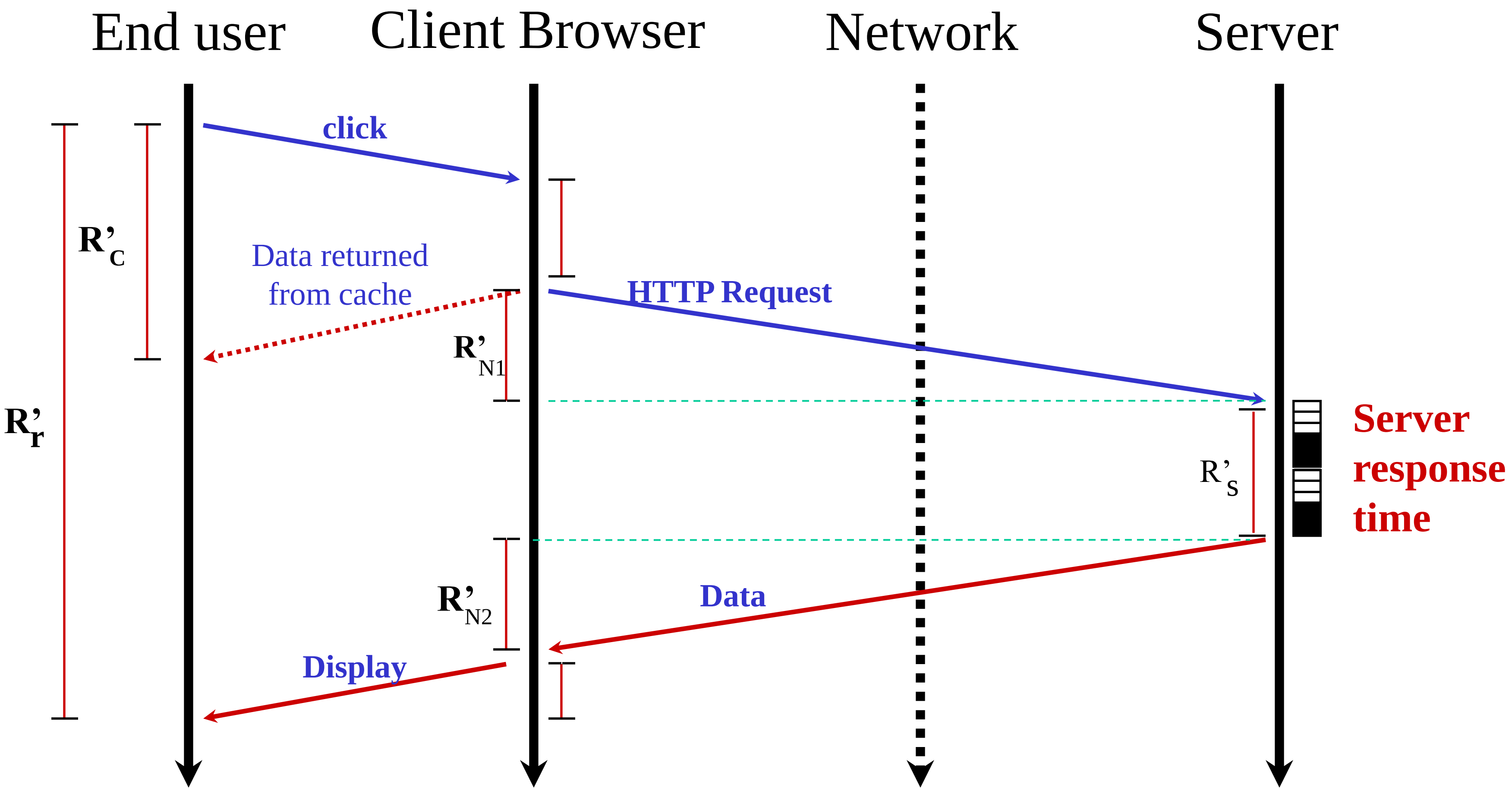
HTTP 1.0 and 1.1 interaction



Where are the delays?

- Browser
  - $R_{browser}$
- Network
  - $R_{network}$
- Server
  - R_{server}
- User response time: R_r
  - $R_r = R_{browser} + R_{network} + R_{server}$ or
  - $R_r = R_{cache}$ if the document is in the user-cache

Anatomy of an HTTP transaction



Adapted from Menascé & Almeida

Average Response Time

- Usually $R_{cache} << R_{network} + R_{server}$

- p_c denotes the fraction of time the data are found in the local cache
- R_{cache}: response time when the data are found in a local cache

$$\boxed{R = p_c \times R_{cache} + (1-p_c) \times R_r}$$

Impact of the Browser's Cache

(example 4.3)

- 20% of the requests are serviced by the local cache
- local cache response time = 400 msec
- average response time for remote Web sites = 3 seconds

Impact of the Browser's Cache

(example 4.3)

- 20% of the requests are serviced by the local cache
- local cache response time = 400 msec
- average response time for remote Web sites = 3 seconds

$$R = p_c \times R_{cache} + (1-p_c) \times R_r$$

$$R = 0.20 \times 0.4 + (1-0.20) \times 3.0$$

$$R = 2.48 \text{ sec}$$

Impact of the Browser's Cache

(example 4.3)

- What if we increase the size of the local cache?
- Previous experiments show that tripling the cache size would raise the hit ratio to 45%. Thus,

Impact of the Browser's Cache

(example 4.3)

- What if we increase the size of the local cache?

- Previous experiments show that tripling the cache size would raise the hit ratio to 45%. Thus,

$$\mathbf{R = p_c \times R_{cache} + (1-p_c) \times R_r}$$

$$\mathbf{R = 0.45x\ 0.4 + (1-0.45) \times 3.0}$$

$$\mathbf{R = 1.83\ sec}$$

Bottlenecks

- As the number of clients and servers grow, overall performance is constrained by the performance of some components along the path from the client to the server.
- The components that limit system performance are called **bottlenecks**

Example of a Bottleneck

- A home user is unhappy with access times to Internet services. To cut response time down, the user is considering replacing the processor of his/her desktop with one twice as fast.

What will be the response time improvement if I upgrade the speed of my desktop computer?

Example of a Bottleneck

(example 4.4)

for an average page:

- avg. network residence time:
  - 7,500 msec
- avg. server residence time:
  - 3,600 msec
- avg browser time:
  - 300 msec
- $R_r = R_{browser} + R_{network} + R_{server} = 300+7{,}500+3{,}600$
- **R_r = 11,400 msec = 11.4 sec**

Example of a Bottleneck

(cont. example 4.4)

- Percentage of time:

$$\%x = R_x \; / \; (R_{browser} + R_{network} + R_{server})$$

- browser = 300/11,400 = 2.14 %
- network = 7,500/11,400 = 65.79 %
- server = 3,600/11,400 = 31.57 %

Example of a Bottleneck

(cont., example 4.4)

- The CPU upgrade affects mainly the browser time:
  - $R^N_{browser} \sim 1/2 \times R_{browser} = 1/2 \times 300 = 150$ msec

- $R^N_r = R^N_{browser} + R_{network} + R_{server}$
- $R^N_r = 150 + 7{,}500 + 3{,}600 = 11.25$ sec.

- Therefore if the speed of the PC were doubled, the response time would decrease only by

$R_r/R^N_r = 11.40/11.25 = 1.3\%$

Perception of Performance

- WWW user:
  - fast response time
  - no connection refused

- Web administrators:
  - high throughput
  - high availability

Need for quantitative measurements

WWW Performance Metrics (I)

- connections/second
- Mbits/second
- response time
  - user side
  - server side
- errors/second

WWW Performance Metrics (II)

Web site activity indicators

- Visit: a series of consecutive Web page requests from a visitor within a given period of time.
- Hit: any connection to a Web site, including in-line requests, and errors.
- Metrics
    - hits/day
    - visits/day
    - unique visitors/day
    - pages views/day

WWW Performance Metrics (III)

Web Advertising Measurements

- Exposure metrics (visits/day, pages/day)
  - site exposure
  - page exposure
  - banner exposure
- Interactivity metrics
  - visit duration time
  - inter-visit duration
  - visit depth (total # of pages a visitor is exposed during a single visit to a Web site)

Example of Performance Metrics

The Web site of a travel agency was monitored for 30 minutes and 9,000 HTTP requests were counted. **We want to assess the server throughput.**

- 3 types of Web objects
  - HTML pages: 30% and avg. size of 11,200 bytes
  - images: 65% and avg. size of 17,200 bytes
  - video clips: 5% and avg. size of 439,000 bytes

Example of Performance Metrics

Throughput

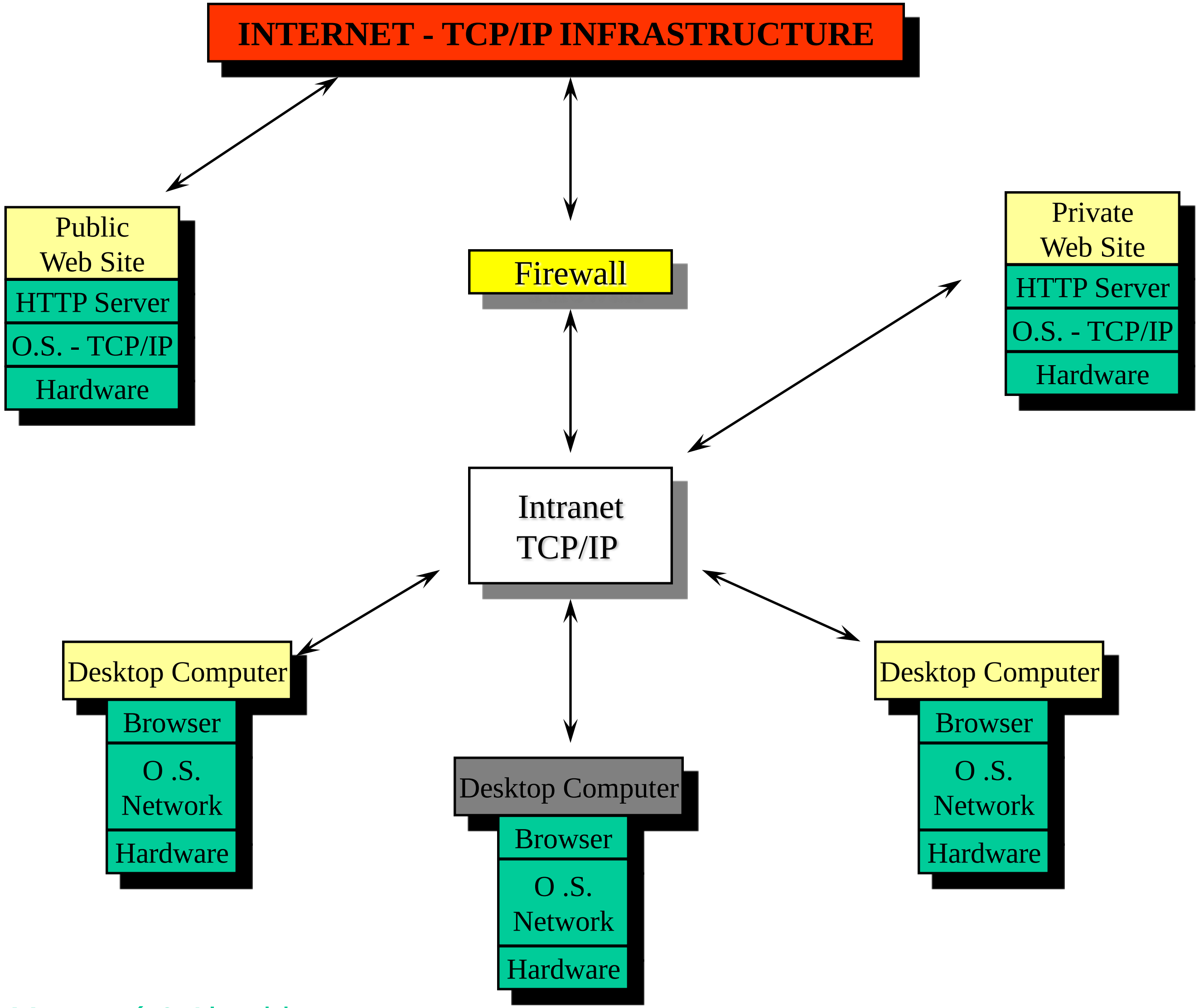
- in terms of requests:
  - (No. of requests)/(period of time) =
  - 9,000/(30 x 60) = 5 requests/sec
- In terms of bits/sec per class
  - (total requests x class % x avg. size) / (period of time)

Example of Performance Metrics

- HTML throughput (Kbps)
  - 9,000 x 0.30 x (11,200 x 8) / 1,800 = 131.25
- Image throughput (Kbps)
  - 9,000 x 0.65 x (17,200 x 8) / 1,800 = 436.72
- Video throughput (Kbps)
  - 9,000 x 0.05 x (439,000 x 8) / 1,800 = 857.42

- Total throughput
  - 131.25 + 436.72 + 857.42 = 1,425.39 Kbps

Web infrastructure



Adapted from Menascé & Almeida

Quality of Service

- As Web sites become a fundamental component of businesses, quality of service will be one of the top management concerns.

- The quality of the services provided by a Web environment is indicated by its service levels, namely:
  - response time
  - availability
  - predictability
  - cost

Quality of Service

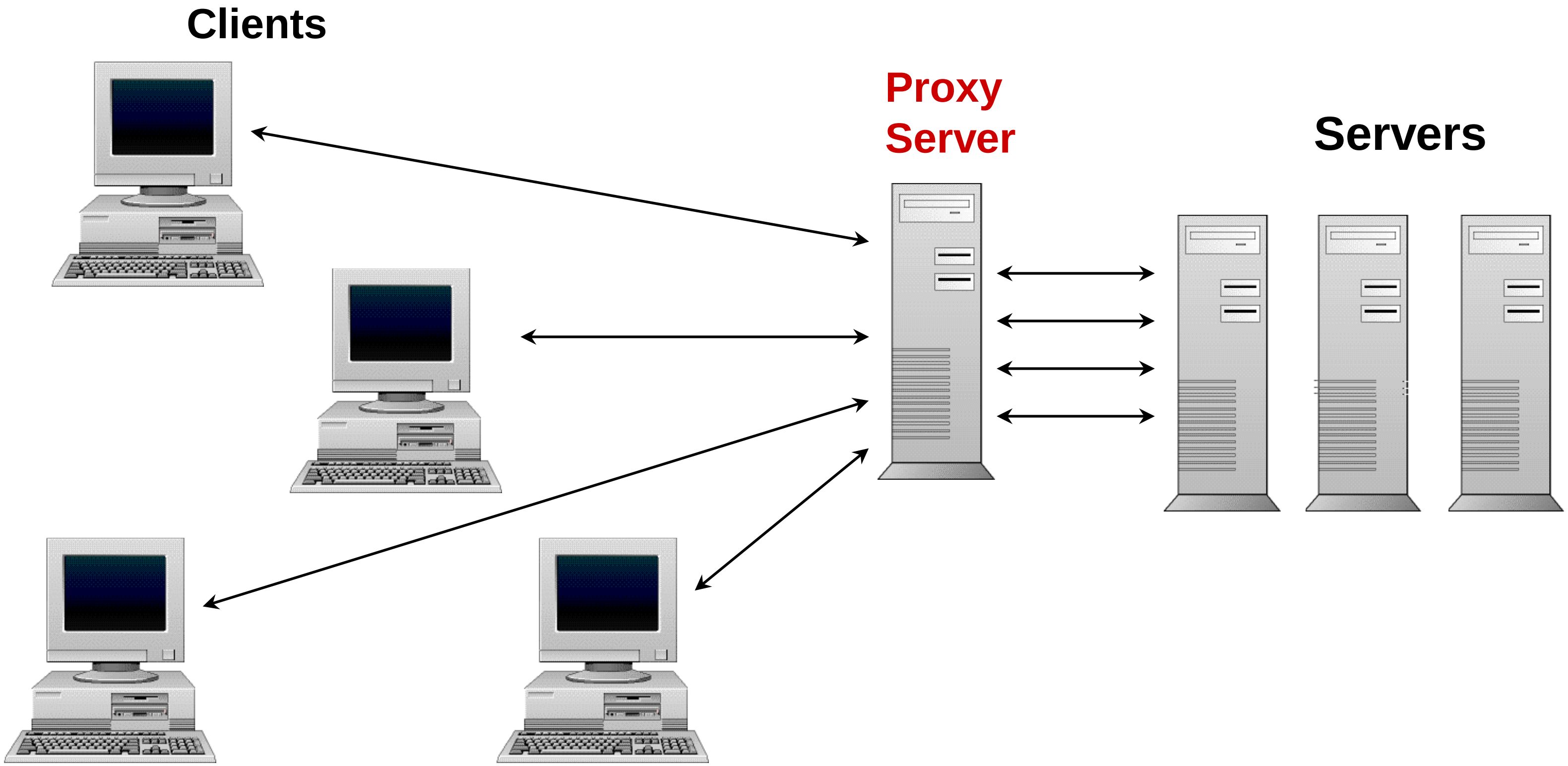
- The problem of quality of service on the Web is exacerbated by the unpredictable nature of interaction of users with Web services. It is usual to see the load of a Web site being multiplied by 8 on the occurrence of a special event.

- How does management establish the service levels of a Web site?

Quality of Service

- Typical questions to help to establish the service level of a Web service:
  - Is the objective of the Web site to provide information to external customers?
  - Do your mission-critical business operations depend on the World Wide Web?
  - Do you have high-end business needs for which 24 hours-a-day, 7 days-a-week uptime and high performance are critical, or can you live with the possibility of Web downtime?

Web Proxy Architecture



Web Proxy Architecture

- A proxy acts as an agent, representing the server to the client and the client to the server.
- A proxy accepts request from clients and forwards them to Web servers.
- Once a proxy receives responses from remote servers, it passes them to clients.
- Proxies can be configured to cache relayed responses, becoming then a caching proxy.

Web Caching Proxy: an example

(example 4.6)

- A large company decided to install a caching proxy server on the corporate intranet. After 6 months of use, management wanted to assess the caching effectiveness. So, we need performance metrics to provide quantitative answer for management.

- <u>Cache A</u>: we have a cache that only holds small documents, with average size equal to 4,800 bytes. The observed hit ratio was 60%.
- <u>Cache B</u>: the cache management algorithm was specified to hold medium documents, with average size of 32,500 bytes. The hit ratio was 20%

Web Caching Proxy: an example

(example 4.6)

- The proxy was monitored during 1 hour and 28,800 requests were handled in that interval.
- Let us compare the efficiency of the two cache strategies by the amount of saved bandwidth

- SavedBandwidth = (No-of-Req.$\times$Hit-Ratio $\times$Size)/Int.

- SavedBandwidth-A = (28,800 $\times$0.6$\times$4,800 $\times$8)/3,600
  = 180 Kbps
- SavedBandwidth-B = (28,800 $\times$0.2$\times$32,500 $\times$8)/3,600
  = 406.3 Kbps

Workload: dynamic Web pages
(example 4.8)

- The Web site of a virtual bookstore receives an average of 20 visitors per second. One out of 10 visitors places an order for books. Each order transaction generates a CGI script, which is executed on the Web server. The Webmaster wants to know what is the CPU load generated by the CGI script.

- Consider that the average CPU service demand of a CGI script is: D_{cpu} = 120 msec.
- Using the Service Demand Law:
- $U_{cpu} = X_{cgi} \times D_{cpu}$

Workload: dynamic Web pages

(example 4.8)

- $X_{cgi} = \lambda_{cgi}$ = VisitRate $\times$ PercentageOfOrders

  = 20 $\times$ (1/10) = 2 CGI/sec
- U_{cpu} = 2 $\times$ 0.12 = 0.24 = 24%

- What would be the impact of replacing the CGI applications by servlets? Let us assume that Java servlet transactions are 30% less resource-intensive than CGI applications.

$$D_{cpu}^{s} = D_{cpu}^{cgi} \times 0.7 = 120 \times 0.7 = 84 \text{ msec.}$$

- The CPU utilization due to servlets would be

  U_{cpu} = 2 $\times$ 0.084 = 0.168 = 16.8%

Novel Features in the WWW

- The Web exhibits extreme variability in workload characteristics:
  - Web document sizes vary in the range of 10^2 to 10^6 bytes
  - The distribution of file sizes in the Web exhibits heavy tails. In practical terms, heavy-tailed distributions indicate that very large values are possible with non-negligible probability.
- Web traffic exhibits a bursty behavior
  - Traffic is bursty in several time scales.
  - It is difficulty to size server capacity and bandwidth to support demand created by load spikes.

Novel Features in the WWW

- The manager of the Web site of a large publishing company is planning the capacity of the network connection.
- 1 million HTTP operations per day
- average document requested was 10 KB
- The required bandwidth (Kbps) is:

  HTTP op/sec $\times$ average size of documents (KB)

  11.6 HTTP ops/sec $\times$ 10 KB/HTTP op = 928 Kbps

- Assume that protocol overhead is 20%

Novel Features in the WWW

- The actual throughput required is

$$928 \times 1.20 = 1.114 \text{ Mbps}$$

  which can be provided by a T1 connection.

- Assume that management decided to plan for peak load. The hourly peak traffic ratio observed was 5 for some big news event. Then the required bandwidth is:

$$1.114 \times 5 = 5.57 \text{ Mbps}$$

  which requires four T1 connections.

Capacity Planning of Web Servers

- It can be used to avoid some of the obvious and most common pitfalls: site congestion and lack of bandwidth. Typical capacity planning questions:
  - ☐Is the corporate network able to sustain the intranet traffic?
  - ☐Will Web server performance continue to be acceptable when twice as many people visit the site?
  - ☐Are servers and network capacity adequate to handle load spikes?

Summary

- Web server problems
- Combination of HTTP and TCP/IP
- Simple examples using operational analysis
- Bottlenecks
- Perception of performance and metrics
- Quality of Service
- Web caching proxy