

L5: Internet Application Topology



Sebastian Magierowski
York University

Outline

- Client/Server
- Server Farms
- Content Distribution Networks (CDNs)
- P2P

Information Exchange & Content Delivery

- Hosts on the internet have a number of different ways of exchanging information
- We focus on systems capable of supporting content delivery (rather than communication per se)
 - Internet more about content than communication
 - Majority of bandwidth used to deliver stored videos
 - YouTube accounts for ~10% of Internet traffic
 - <1994: FTP & email
 - 1994-2003: Web traffic
 - 2003-2008: P2P traffic
 - >2008: Video streaming
 - By 2014 90% of Internet traffic predicted to be video

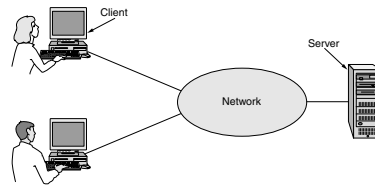
CSE 3213, W14

L5: Application Topology

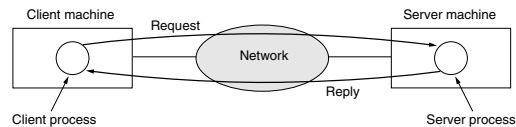
3

Client/Server

- A connection between two hosts
 - Server a powerful machine storing database info
 - Clients ask servers to access remote data
 - Web browsing



- Two processes participate
 - Client side makes request
 - Server side gives reply



[©Pearson, Tanenbaum]

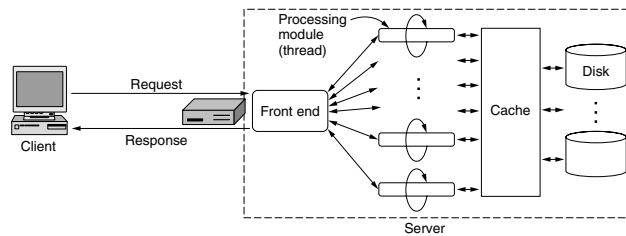
CSE 3213, W14

L5: Application Topology

4

Servers

- Multithreaded machines capable of distributing multiple requests



- To minimize slow disk reads web pages are cached and accessed by each thread

[©Pearson, Tanenbaum]

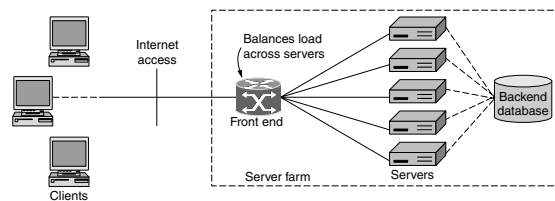
CSE 3213, W14

L5: Application Topology

5

Server Farms

- For popular requests need more than one server
 - Link them all to a common database backend to make them all serve the same files



- How do you distribute queries among the servers?

[©Pearson, Tanenbaum]

CSE 3213, W14

L5: Application Topology

6

Request Distribution

- Have a **front end** router distribute requests to servers
 - All packets of request must be directed to the one correct server
- A Simple Approach:
 - Front end **broadcasts** packets and servers respond to only a fraction of requests by prior agreement
 - E.g.: 16 servers respond to request based on last 4-bits of source IP
 - Broadcast is wasteful of bandwidth (but not horrible)
- Load Balancing (violating layer isolation):
 - Front end (middlebox) snoops higher layer headers (TCP/HTTP) and maps to a server of its choice
 - TCP helps identify individual users
 - HTTP contains cookies which can be used to extract cached data from a previously used server
 - Dangerous to look at higher layers

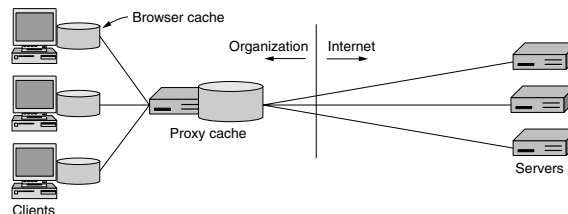
CSE 3213, W14

L5: Application Topology

7

Caching & Web Proxies

- Individual user browsers can cache
 - Relatively small storage
- To reduce traffic large organizations and ISPs can cache as well: **web proxy** shares a cache among users
 - Typically set up for an organization
 - Company
 - ISP
- Share cache among multiple users
 - One request stores in cache, ensuing requests just query cache
 - Not effective for dynamic pages



[©Pearson, Tanenbaum]

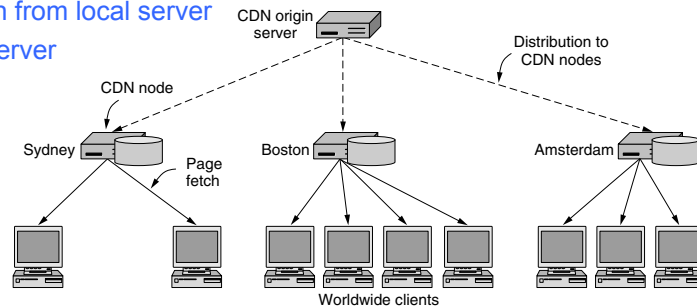
CSE 3213, W14

L5: Application Topology

8

Content Delivery Networks (CDNs)

- For giant websites serving on global scale, outsource content to a CDN
 - Akamai the pioneer in this since 1998
- Provider places copy of page in a set of nodes and directs client to nearest node
 - Clients fetch from local server
 - Not origin server



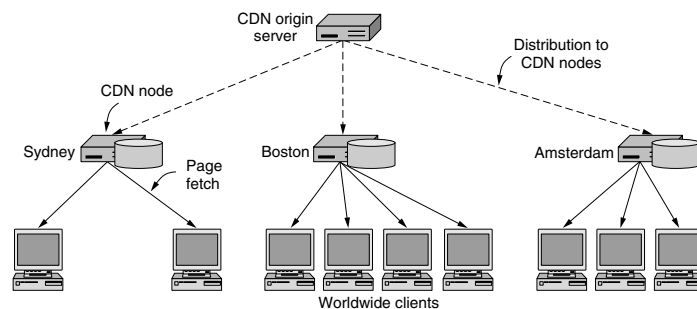
CSE 3213, W14

L5: Application Topology

9

Benefits of CDN Tree

- Origin is not overloaded by multitude of client requests
- Each client gets good performance by fetching from local server
- Total load decreased
 - Despite multiple queries information is brought to a leaf node only once



CSE 3213, W14

L5: Application Topology

10

CDN Implementation

- How do you implement this?
- Can the leafs be caching web proxies?
 - Caching has poor performance when serving more than ~100 clients
 - Statistical nature of web searches has long tail (Zipf distribution)
 - Proxies are configured by clients...
 - ...their activity controls caching
- Mirrors?
 - Replicate content over nodes in different regions
 - Structurally ok, but a static approach that depends on users to do the distribution as they are different websites
- The third approach...

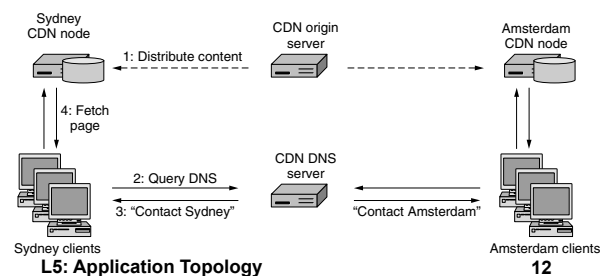
CSE 3213, W14

L5: Application Topology

11

...DNS Redirection

- CDN runs the name servers
 - When a client's web page is looking for an IP address...
 - ...they are directed by the CDN name server based on their own IP address
- For example...
 - Sydney request gets prompted to Sydney CDN node by CDN DNS
 - Future requests return to Sydney CDN by virtue of DNS caching



CSE 3213, W14

L5: Application Topology

12

CDN Redirection

- The CDN DNS redirection is based on sophisticated algorithms
 - mapping client IP to CDN node
 - distance, capacity, CDN load balance, etc.
- Each node consists of many servers
 - DNS redirection not only takes clients to a certain location
 - Another level of the algorithm also distributes them to the appropriate servers

CDN Implementation

- Companies pay CDN to deliver responsive websites
- CDN nodes placed inside locations with good connectivity
 - ISPs
 - ISP benefits because CDN cuts down on their upstream bandwidth (e.g. through Tier-1) and hence saves \$\$\$
 - Improves ISP responsiveness (makes ISP look good)
- Owner gives CDN website content and re-writes code to link through CDN
 - Owner retains control, CDN moves bulk of data
 - `Trailer
`
 - `Trailer
`
- Akamai: 137,000 servers, 87 countries, in 1150 networks

Peer-to-Peer (P2P) Networks

- Napster, 1999
- BitTorrent, 2001
- Many computers pool resources to form content distribution system
- Peers
 - Computers can alternately act as clients and servers to their peers
- No dedicated infrastructure like CDN
- Often no central point of control

P2P Throughput

- N distributed users with M rate duplex lines can form $N \cdot M$ rate duplex throughput through network
- Consider a basic organization
 - N users in a binary connected tree
 - A file is fed from the top node and uploaded to all remaining nodes in a pipelined fashion
 - Total upload bandwidth is?
 - Swap role of leaf and non-leaf nodes
- Self-scaling
 - Usable throughput grows with download demand (i.e. number of users)
- Other structure throughputs are fixed
 - Consider 100, 10-Gbps nodes

BitTorrent Protocol

- 3 Main problems to solve:
 - How to find other peers with desired content
 - Not all peers will have all content initially
 - How to replicate content to provide high-speed download for all
 - How to best diffuse content through the network
 - How to encourage both upload for others while downloading for self
 - Maintain cooperation

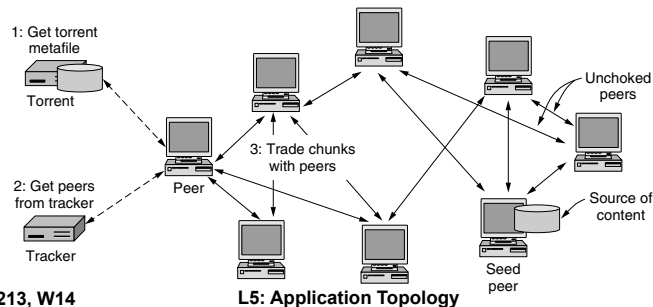
CSE 3213, W14

L5: Application Topology

17

Finding Peers

- Torrent
 - Get torrent from a web page advertising the content
 - ~1000X smaller than content
- Specifies 2 kinds of information
 - Tracker: Server maintaining list of all peers involved in network (swarm)
 - Chunk name list: equal sized pieces making up the content
- Swarm regularly updates tracker on their presence



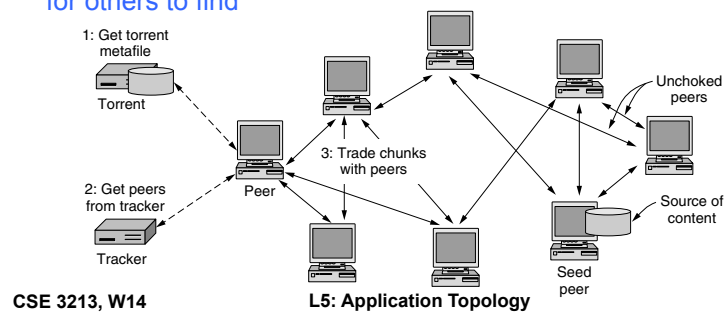
CSE 3213, W14

L5: Application Topology

18

Sharing Content for Rapid Download

- Seeders have the whole file
 - If everyone got seeded chunks in same order bottlenecks would occur
- Peers exchange lists of owned chunks with other peers
 - Select rare chunks that are hard to find for download
 - Downloading a rare chunk creates a copy of it and makes it easier for others to find



Upload/Download Balance

- Reward peers with good upload behaviour
 - Peer randomly chooses other peers for upload/download
 - I.e. if your random choice is bad, stop trading chunks with it
 - But continue randomly searching for better peers
 - If you are not uploading you will tend to be cutoff
 - Download tends to match upload

