CSCE 463/612
Networks and Distributed Processing
Spring 2024

Application Layer VI

Dmitri Loguinov
Texas A&M University

February 21, 2024

Chapter 2: Roadmap

2.1 Principles of network applications

2.2 Web and HTTP

2.3 FTP

2.4 Electronic Mail

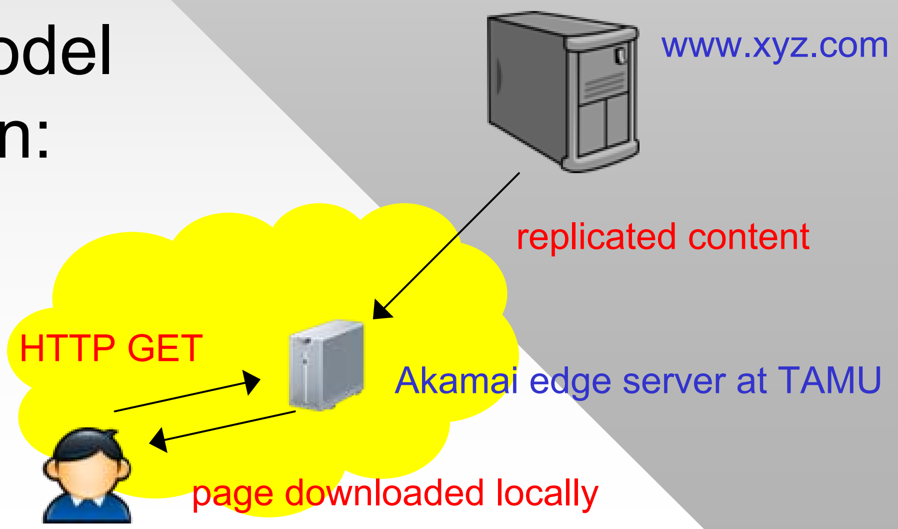
- SMTP, POP3, IMAP

2.5 DNS (extras)

2.6 P2P file sharing

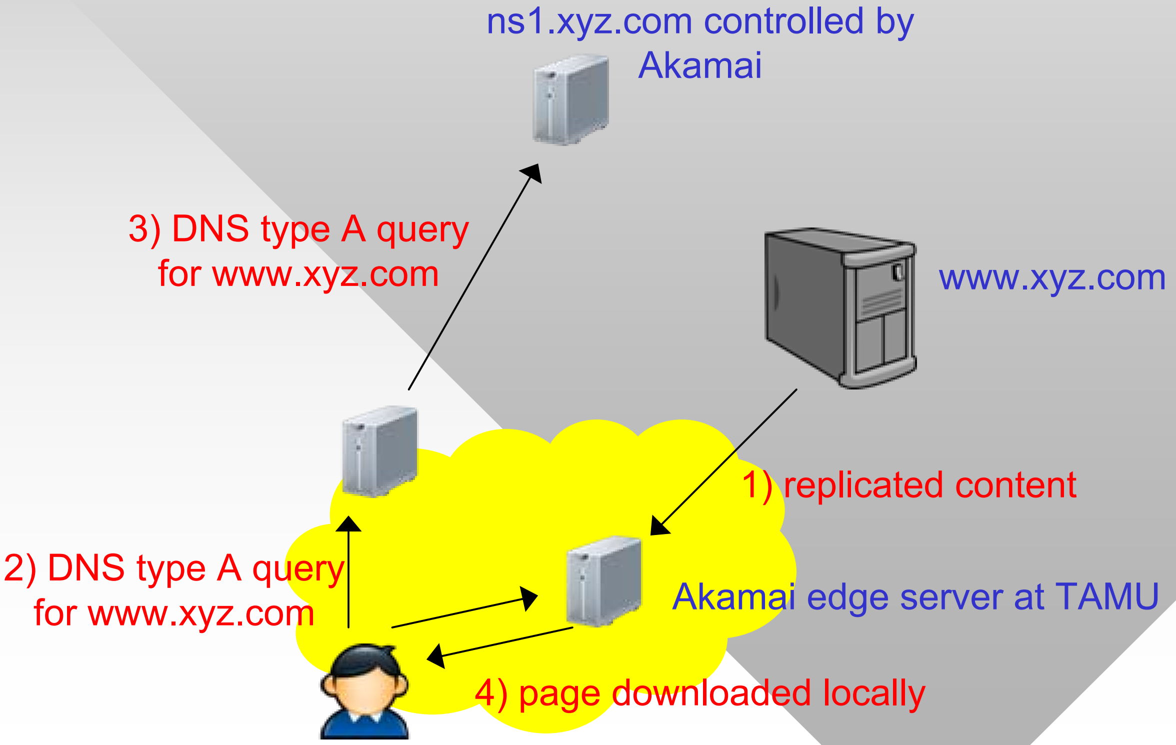
CDNs

- Content Distribution Networks (CDNs)
  - Push replicated content (files, video, images) towards edges
  - Distributed system of application-layer servers
- One of the pioneering CDNs is Akamai
  - J. Dilley, B. Maggs, J. Parikh, H. Prokop, R. Sitaraman, and B. Weihl, “Globally Distributed Content Delivery,” IEEE Internet Computing, Sep/Oct 2002.
- Desired model of operation:



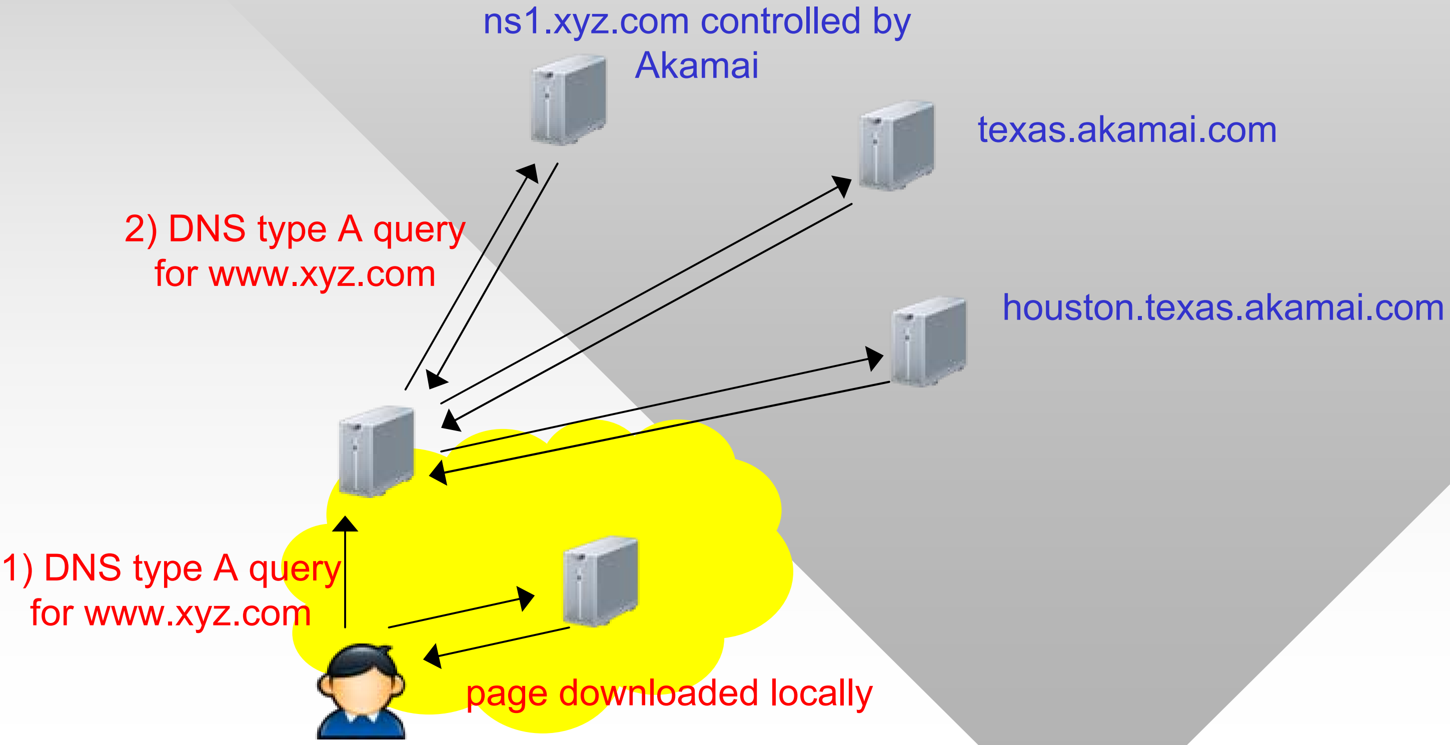
CDNs 2

- How to direct user to closest replica?
  - Akamai relies on DNS to bounce the user to the best server
  - Based on location of local resolver finds the best server (e.g., using distance, load, latency, available bandwidth)


ns1.xyz.com controlled by Akamai
3) DNS type A query for www.xyz.com
www.xyz.com
1) replicated content
2) DNS type A query for www.xyz.com
Akamai edge server at TAMU
4) page downloaded locally

CDNs 3

- How many servers are there?
  - Around 365K in 135 countries and 1350 networks
- Often Akamai produces long redirect chains
  - Usually through CNAMEs based on the IP of local resolver


ns1.xyz.com controlled by Akamai
texas.akamai.com
2) DNS type A query for www.xyz.com
houston.texas.akamai.com
1) DNS type A query for www.xyz.com
page downloaded locally

CDNs 4

- One research problem in CDNs is how to determine best edge server for the user
  - If multiple options are present, which one is better?
  - What if closest server is overloaded?
  - Not all servers have every possible version of content
  - Need to account for ISP agreements on bandwidth
- Example:
  - Lookup from Germany gives out an IP in Frankfurt

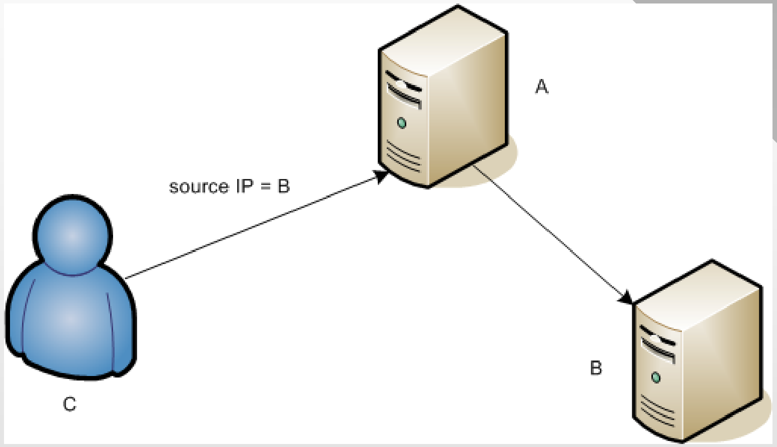
```
www.dhs.gov CNAME www.dhs.gov.edgekey.net
www.dhs.gov.edgekey.net CNAME e4340.dscg.akamaiedge.net
e4340.dscg.akamaiedge.net A 23.45.237.161 (TTL 20 seconds)
```

  - Same lookup from TAMU produces an IP in Dallas

CDNs 5

- One pitfall of CDNs is that distance from user to their local resolver is generally unknown
  - May lead to inaccuracies for large ISPs
- Another drawback is long resolution chains
  - 15 CNAMEs back-to-back is not just huge latency, but also prone to incorrect configuration, dead-ends, loops
  - Caching helps with latency, but Akamai uses extremely small TTLs (e.g., 20 sec), so might still be an issue
- Useful online tools
  - dnswatch.info shows a full trace of lookups from the root
  - ip2location.com, ipgeolocation.io map IPs to country/city
  - Registrars (e.g., ARIN, RIPE) allocate subnets; their whois database can be used to map IPs to owner networks

DNS Vulnerabilities



- Terminology: IP spoofing
  - Packets with fake source IP
- For spoofing to work, ISP network of attacker must allow such packets to depart
  - Robert Beverly, Arthur Berger, Young Hyun, and K Claffy, “Understanding the Efficacy of Deployed Internet Source Address Validation Filtering,” ACM IMC, 2009
  - Of 12K IPs tested, 31% were able to spoof (18% across the US, 5% for edu and home networks)
- TCP spoofing is hard
  - Almost impossible to complete the handshake without knowing parameters of the response packet (only B sees them)
- However, UDP spoofing is easy

DNS Vulnerabilities 2

- Terminology: amplification attacks
  - Hacker transmits small packets to intermediate hosts, which then generate more traffic towards the victim
  - Relies on spoofing the IP of the victim
  - Difficult to trace as the attacker remains hidden
- DNS amplification (1999)
  - Short questions produce large replies, combined with spoofing
  - Large reply = many answers or additional records
- How much amplification can be achieved?
  - IP+UDP+DNS headers = 40 bytes, question $\approx$ 15 bytes
  - Maximum reply is 512 bytes over UDP, ratio 9.3:1
  - 1 Mbps upstream bandwidth per attacker host → 9.3 Mbps

Chapter 2: Roadmap

2.1 Principles of network applications

2.2 Web and HTTP

2.3 FTP

2.4 Electronic Mail

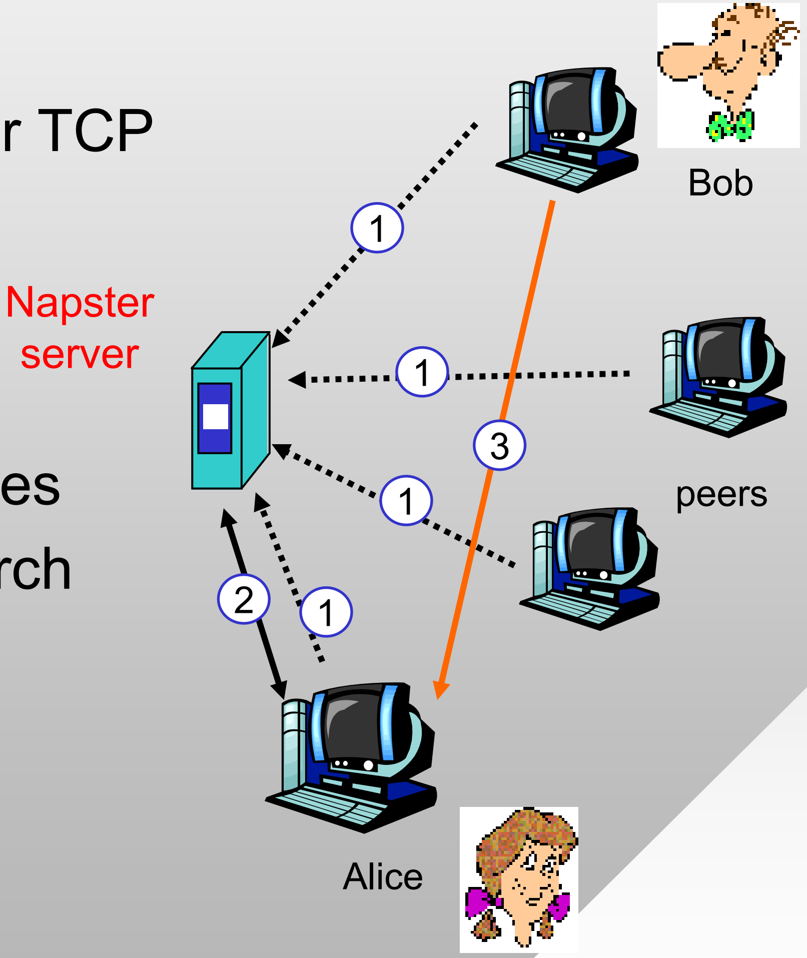
- SMTP, POP3, IMAP

2.5 DNS

2.6 P2P file sharing

Hybrid P2P

- Napster (1999)
  - Application-layer protocol over TCP
  - Centralized directory server
- Sequence of steps
  - Connect to server, login
  - Upload your IP/port + list of files
  - Give server keywords for search
  - Select “best” answer (ping)
  - Download from peer
- Single point of failure
- Performance bottleneck
- Target for litigation due to copyright infringement

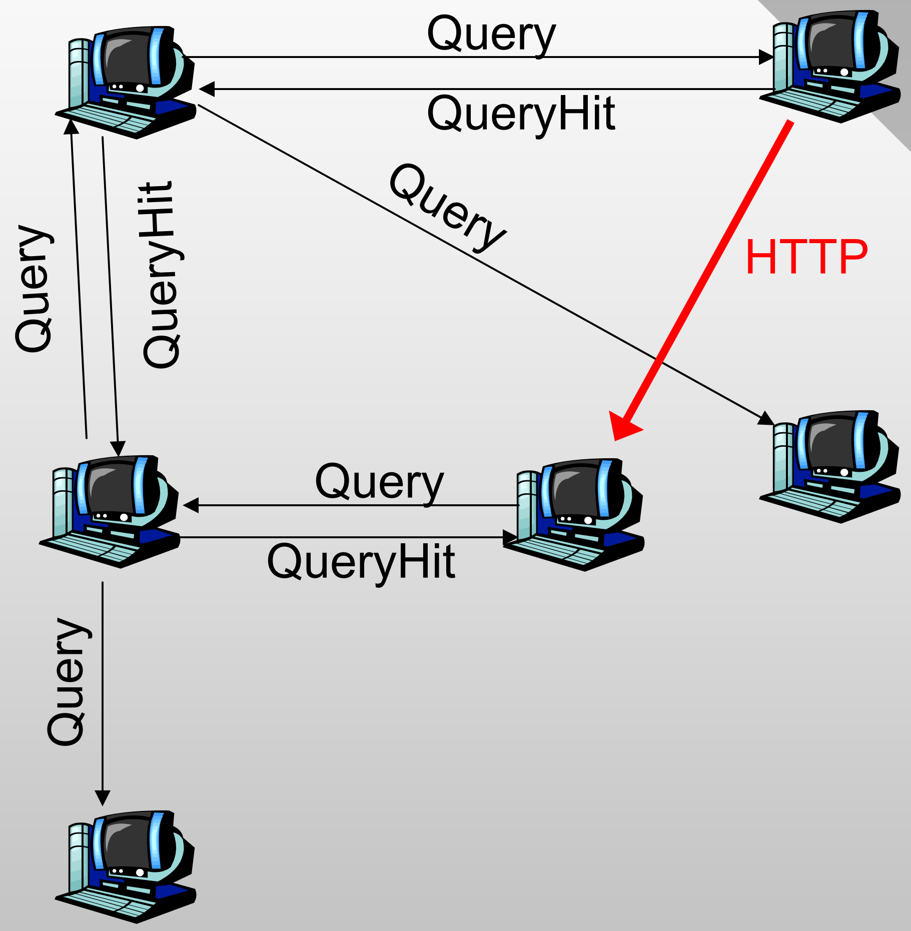


Decentralized P2P

- Napster folded in 2002
  - Other P2P systems took over (Gnutella, KaZaA, BitTorrent, eDonkey)
- Gnutella/0.4 (2001)
  - Public-domain protocol
  - Fully distributed design
- Many Gnutella clients implementing protocol
  - Limewire, Morpheus, BearShare

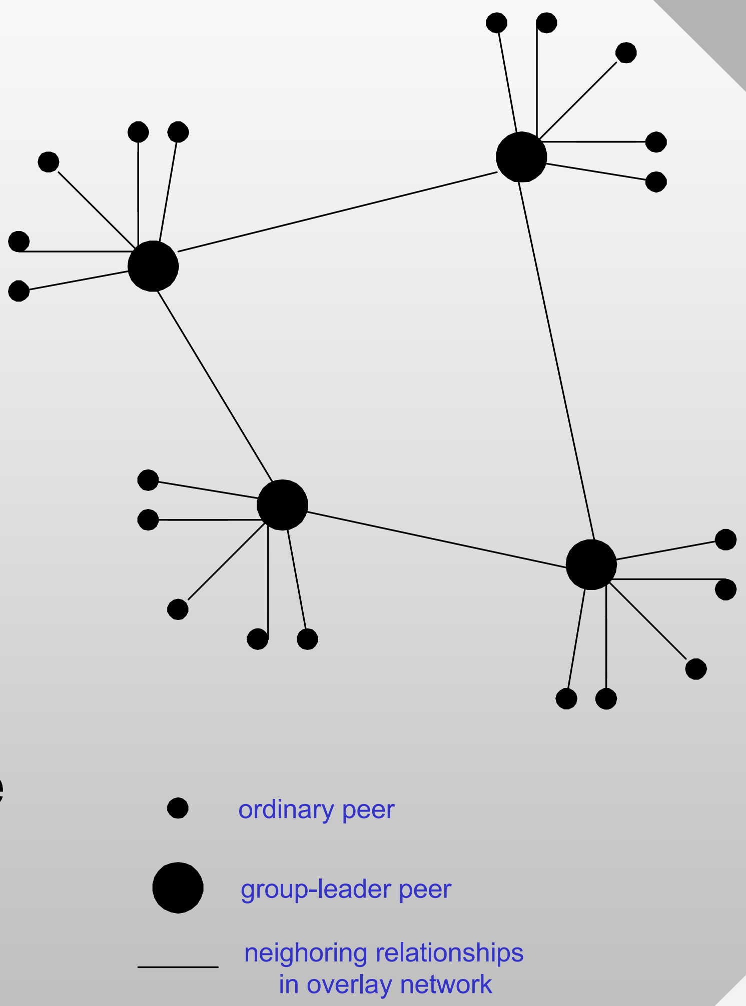
- How to find content?
- Idea: construct a graph
  - Edge between peer X and Y if there's a TCP connection between them
- All active peers and edges are called an overlay network
  - Peer typically connected to < 30 neighbors
- Search proceeds by flooding up to some depth
  - Limited-scope flooding

Decentralized P2P



- Queries are P2P
  - Inefficient due to huge volumes of traffic
  - Average degree k, depth of flood d, overhead $(k-1)^d$
- Downloads are P2P from a single user
  - Unreliable (peer departure or failure kills transfer)
  - Inefficient (asymmetry of upstream/downstream bandwidth)
- Join protocol (bootstrapping)
  - Find an entry peer X, flood its neighbors to obtain more candidates, establish connections to those who accept

Hierarchical P2P



- Gnutella/0.4 scaled to about 25K users and then choked
- Alternative construction proposed by KaZaA (2002)
  - Peer is either a group leader (supernode) or assigned to one
- Group leader tracks the content of all its children, acting like a mini-Napster
  - Peers query their group leaders, which flood the supernode graph until some number of matches found
  - Query-hits not routed, but sent directly to original supernode

Hierarchical P2P

- With 150 neighbors, this architecture is 150x more efficient than Gnutella/0.4 in message overhead
  - With 389M downloads as of 2008, KaZaA was more popular than Napster ever was, accounting for 50% of ISP bandwidth in some regions and running 3M concurrent users
- Gnutella/0.6 soon adopted the same structure
  - Scaled to 6.5M online users, 60M unique visitors per week
- Additional features
  - Hashed file contents to identify exact version of files
  - Upload and request queuing at each user, rate-limiting
  - Parallel downloads from multiple peers
  - Support for crawl requests that reveal neighbors

Other P2P

- Terminology: user holding a complete file is a seed
  - Traditional systems download only from seeds
  - Seed departs, transfer fails
- Idea: let non-seeds grab chunks from each other
  - Peers organize into a group (torrent) based on the file they're downloading
- Traditional systems download files sequentially
  - Starvation for final blocks
- Idea: maximize availability
  - Participants forced to serve chunks they have to others
  - Rarest chunk in torrent is always replicated first
- Known as BitTorrent (2001)
  - Protocol with many implementations
  - Requires trackers to keep torrent membership
  - Had more concurrent users that YouTube and Facebook combined
- Built-in incentives to share
  - Rate-limiting (choking) based on upload activity

Other P2P

- Tor (Onion Router)
  - Anonymity network of peers
- Each packet sent through a random chain of P2P nodes
  - Final user relays packet towards destination
  - Return packets processed similarly along reverse path
- Tor can be run thru an API
  - Extremely slow
  - Many exit points are known and blocked by Google
- Roughly 36M users

- Freenet
  - Anonymous information exchange, hiding identities of communicating parties
- Original Skype chat
  - Video streaming services either directly between users or relayed through non-firewalled peers
- Distributed Hash Tables
  - General class of P2P systems that map information into high-dimensional search space with guaranteed log(N) bounds on delay to find content