CDN and P2P

Everyone is the network and content rules

Outcomes

- Understand the purpose of CDNs
- Discuss the basic operational concepts of a CDN
- Look into the world of peer-to-peer overlays and associated challenges
- Understand basic tradeoffs of indirection in P2P architectures

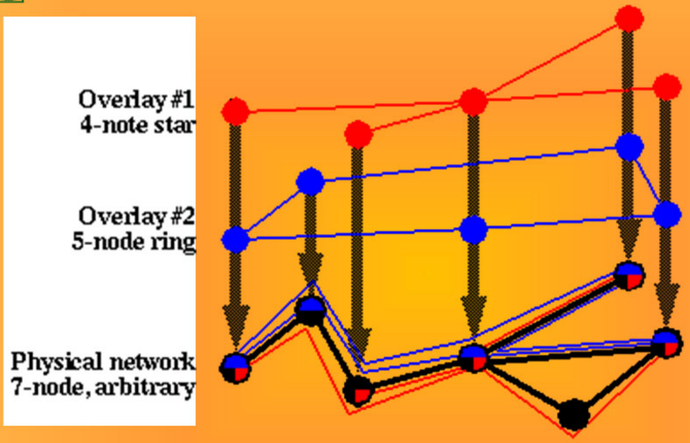
2

3

Why discuss CDNs and P2P together?

- **Both reflect a different set of problems of the traditional client-server model.**
- **Both are focused in *content*, and not in interactive communications.**
- **Both reflect *overlays*.**

What is an Overlay ?



What is the topology of this network?

WHICH network??

www.isi.edu/xbone

Overlay Networks: Overview

- Networks built using an existing network as substrate (*Virtual Networks)*

Internet

  - Initially an overlay on the POTS (Plain Old Telephone System) network
- Overlays are a (quasi) structured virtual topology above the basic transport protocol level that facilitates deterministic search and guarantees convergence
  - Overlays could consist of routing software installed at selected sites, connected by encapsulation tunnels or direct links
- Examples of overlays:
  - MBone, 6Bone
  - P2P (Napster, FreeNet, Gnutella, Bittorrent)
  - Cooperating Caches
  - Server Farms
  - Content Distribution Networks (CDNs)

Content Distribution Networks

Client-Server and distribution models
Caching and load balancing

7

© Rui L. Aguiar (ruilaa@det.ua.pt) - Uni. Aveiro

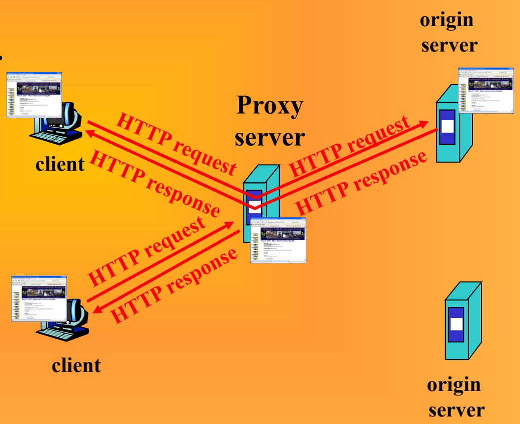
Learning outcomes

- Understand the purpose of content distribution on a network
- Discuss the rationale for CDNs and comment on diferente alternatives
- Describe the architecture of a CDN

(recall FR): Web caches (proxy server)

Goal: satisfy client request without involving origin server

- user sets browser: Web accesses via proxy server
- browser sends all HTTP requests to proxy
  - object in cache: cache returns object
  - else proxy requests object from origin server, then returns object to client



More about Web caching

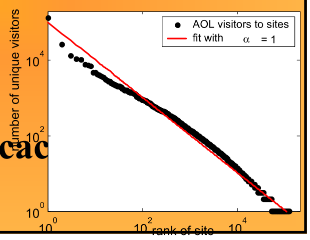
- Proxy server acts as both client and server
- typically proxy server is installed by ISP (university, company, residential ISP)

Why Web caching?

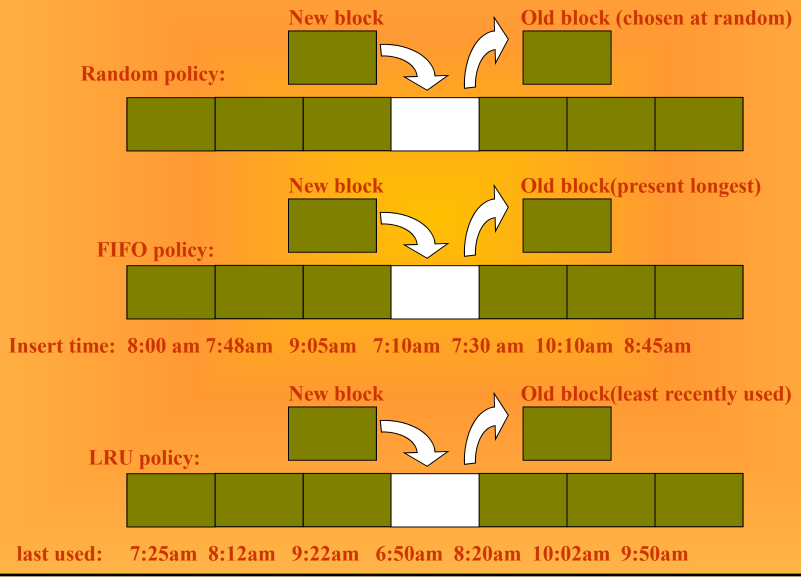
- reduce response time for client request
- reduce traffic on an institution’s access link.

Optimizing performance

- Where to cache content?
  - Popularity of Web objects is Zipf-like
    - a few elements that score *very* high (the left tail in the diagrams)
    - a medium number of elements with middle-of-the-road scores (the middle part of the diagram)
    - a huge number of elements that score very low (the right tail in the diagram)
  - Small number of sites cover large fraction of requests
- Given this observation, how should cac replacement work?



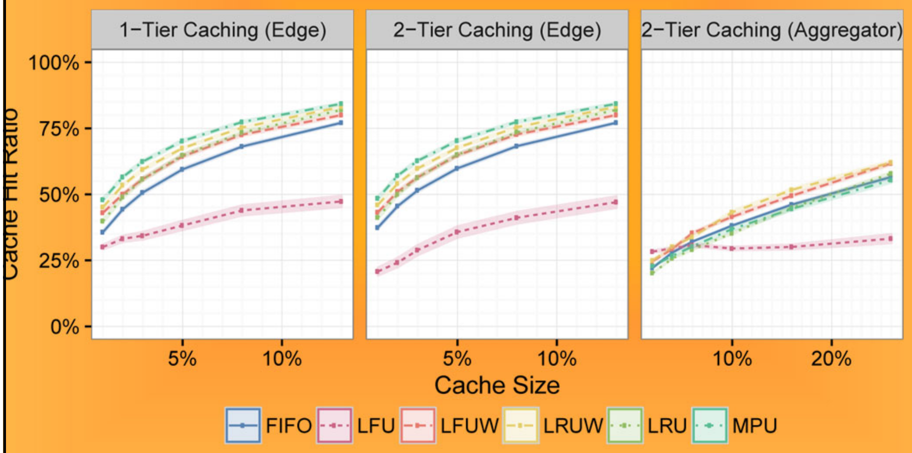
Cache Replacement Policies (I)



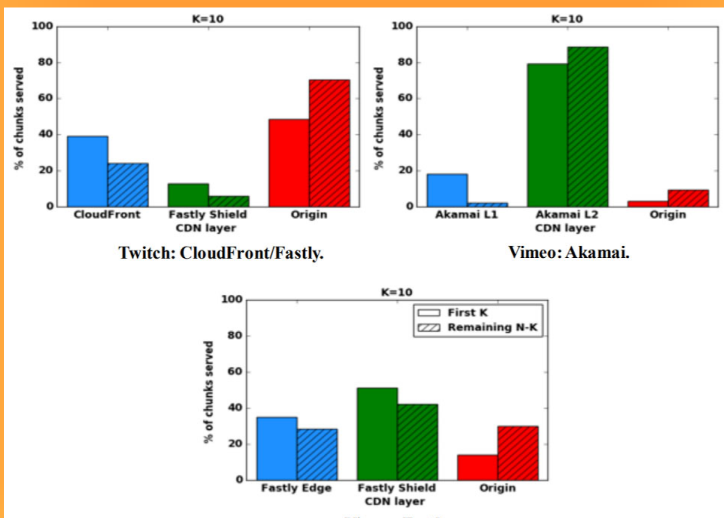
Cache Replacement Policies (II)

- LFU: Least Frequently Used
- MPU: Most Probably Used
- LFU and LRU weighted (give a weight to each page)

Cache Hit Ratio vs Cache Size



Where are objects coming from (even from chunk to chunk)?



Twitch: CloudFront/Fastly.

Vimeo: Akamai.

Vimeo: Fastly.

Caching example

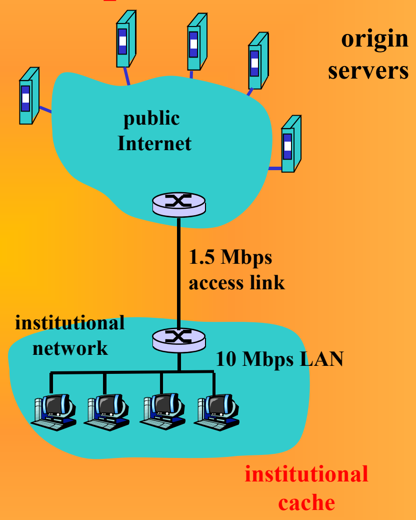
Assumptions

- average object size = 100,000 bits
- avg. request rate from institution's browsers to origin servers = 15/sec
- delay from institutional router to any origin server and back to router = 2 sec

Consequences

- utilization on LAN = 15%
- utilization on access link = 100%
- total delay = Internet delay + access delay + LAN delay

= 2 sec + minutes + milliseconds



Caching example (cont)

possible solution

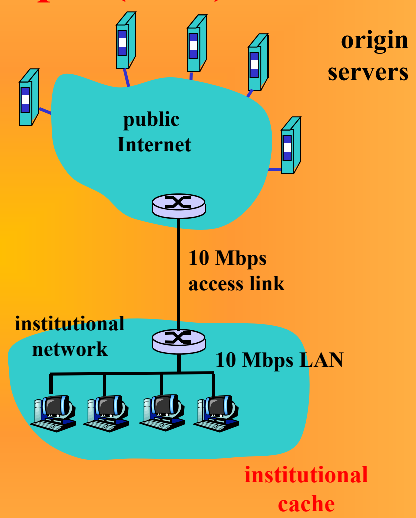
- increase bandwidth of access link to, say, 10 Mbps

consequence

- utilization on LAN = 15%
- utilization on access link = 15%
- Total delay = Internet delay + access delay + LAN delay

= 2 sec + msecs + msecs

- often a costly upgrade



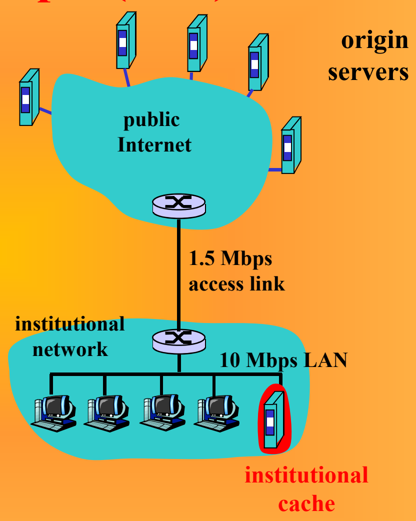
Caching example (cont)

possible solution: install cache

- suppose hit rate is 0.4

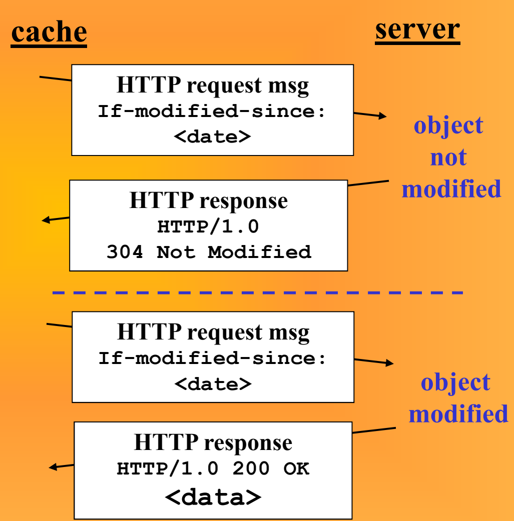
consequence

- 40% requests will be satisfied almost immediately
- 60% requests satisfied by origin server
- utilization of access link reduced to 60%, resulting in negligible delays (say 10 msec)
- total avg delay = Internet delay + access delay + LAN delay = .6*(2.01) secs + .4*milliseconds < 1.4 secs

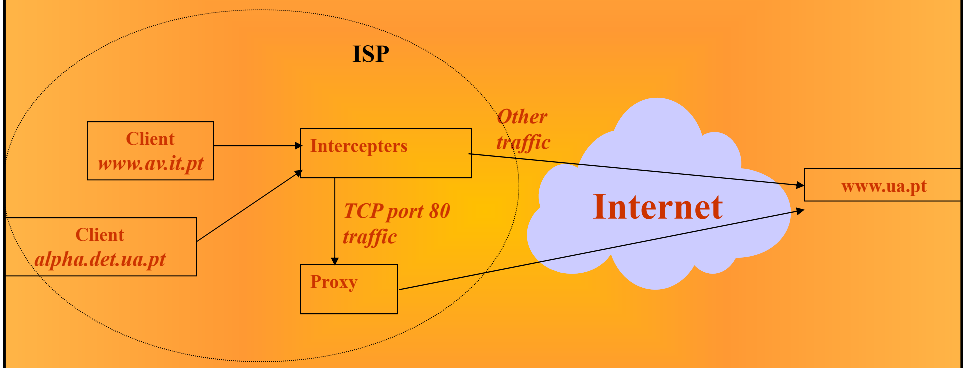


Conditional GET

- Goal: don't send object if cache has up-to-date cached version
- cache: specify date of cached copy in HTTP request
  `If-modified-since: <date>`
- server: response contains no object if cached copy is up-to-date:
  `HTTP/1.0 304 Not Modified`



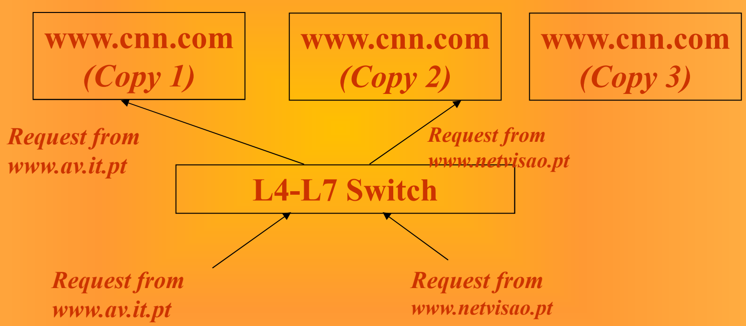
Potential content network structure: Caching Proxies



- Mostly motivated by ISP business interests – reduction in bandwidth consumption of ISP from the Internet
  - Reduced network traffic
  - Reduced user perceived latency

Potential content network structure : Server Farms

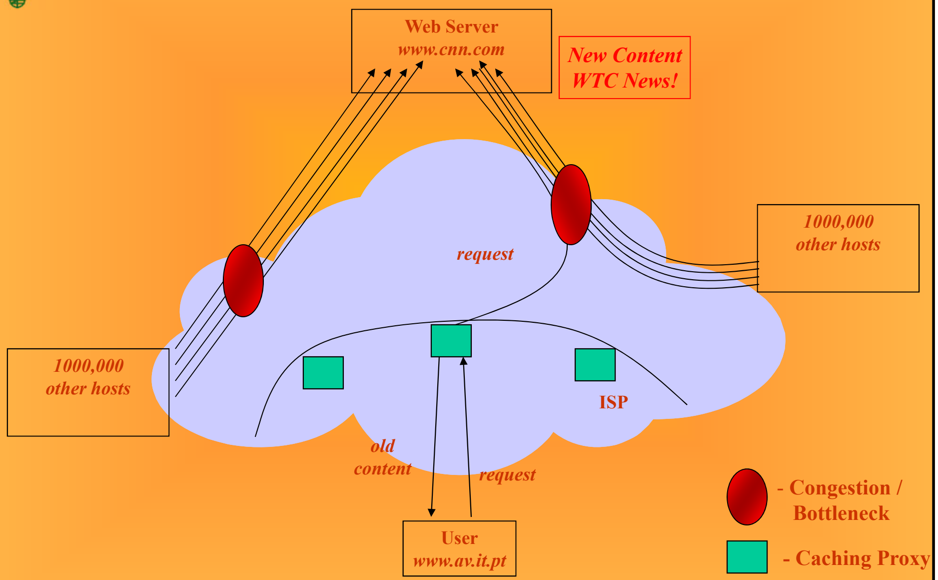
Simple solution to the content distribution problem:
deploy a large group of servers on site



- Arbitrate client requests to servers using an “intelligent” L4-L7 switch
  - Quite used today

Flash Crowds:

Consider, On September 11, 2001



22

Why Not Web-only approaches for content networks?

- Integrating file caching in proxies
  - Optimized for 10KB objects
  - 10GB = 1.000.000 x 10KB
- Memory pressure
  - Disk access is 1000 times slower
  - Working sets do not fit in memory
- Waste of resources
  - More servers needed
  - Provisioning is a must

Problems with *Server farms and Caching proxies*

- **Server farms do nothing about problems due to network congestion, or to improve latency issues due to the network**
- **Caching proxies serve only their clients, not all users on the Internet**
- **Content providers *(say, Web servers)* cannot rely on existence and *correct* implementation of caching proxies**
- **Accounting issues with caching proxies.**
  For instance, *www.cnn.com* needs to know the number of hits to the webpage for advertisements displayed on the webpage

24

Early bird easter egg

- **Como podem as chamadas de voz beneficiar de uma CDN?**

How can voice calls benefit from CDNs?

25

© Rui L. Aguiar (ruilaa@det.ua.pt) - Uni. Aveiro

CDNs

26

© Rui L. Aguiar (ruilaa@det.ua.pt) - Uni. Aveiro

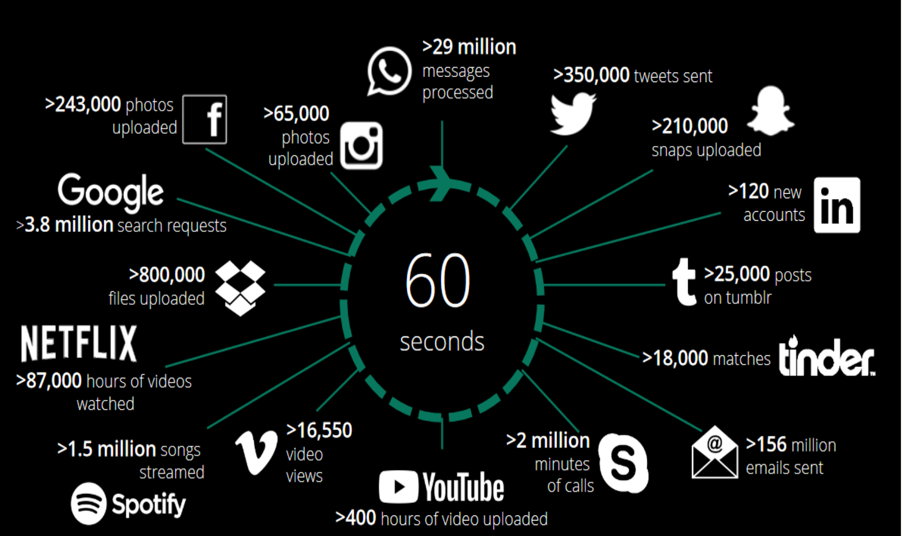
Outline

- **Overall context**
- Challenges
- Potential alternatives?
- Architecture

So much happened in our digitalized world in 2017 – and we have the numbers behind it

Things that happened online in 2017 within 60 seconds

Lots of multimedia content!



28

CDNs Target environment?

Most Web files are small (1KB ~ 100KB)
(initially....)

29

Motivation

- IP based networks
- Web based applications have become the norm for corporate internal networks and many business-to-business interactions
- Large acceptance and explosive growth
  - Serious performance problems
  - Degraded user experience

  For a large set of applications, including VIDEO access
- Improving the performance of networked applications
  - Use many sites at different points within the network
    - Stand alone servers
    - Routers

CDNs basics

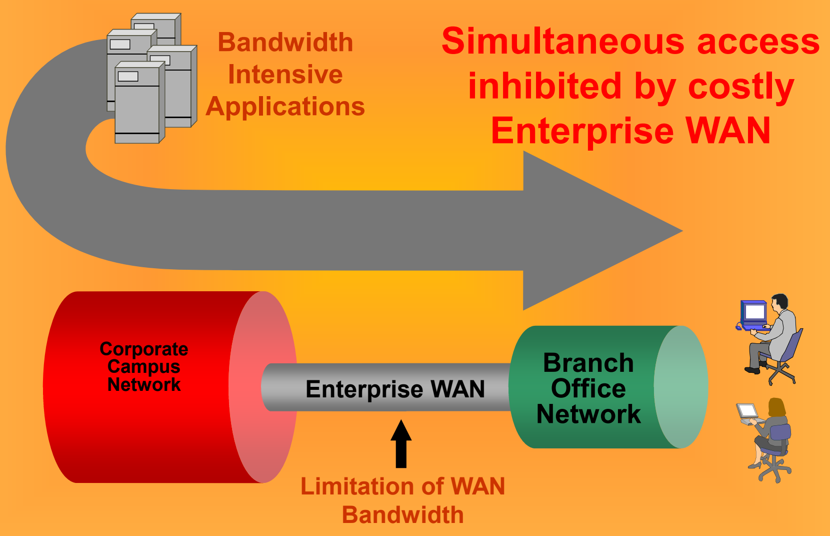
- What is a CDN?
  - A network of servers delivering content on behalf of an origin site
    - A number of CDN companies well established now
      - E.g. Akamai, Digital Island, Speedera, CDN77, Cloudfare, Stackpat
    - Many companies are exploring CDNs
      - Avoid congested portions of the Internet
- Consist of
  - Edge servers deployed at several ISP (Internet Service Provider) access locations and network exchange points
- Large-file service with no custom client, no custom server, no prepositioning
- Improve the response time of an Internet site
  - Offloading the delivery of bandwidth-intensive objects, such as images and video clips
- Intelligent Internet infrastructure that improves the performance and scalability of distributed applications by moving the bulk of their *computation* to servers located at the edge of the network
  - Applications are logically split into two components
    - Executed at an edge server close to the user
    - Executed on a traditional application server

31

CDN Generations

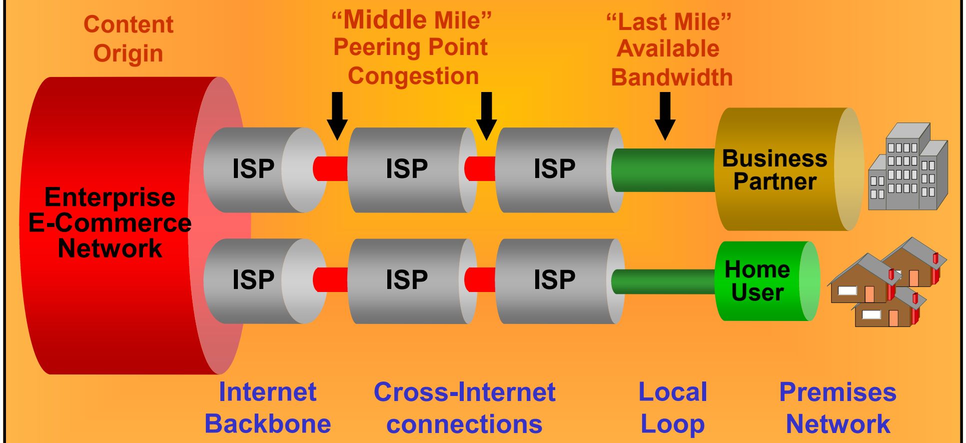
- **First generation (early 90ies)**
  - Accelerate the performance of web sites
  - Support increasing volumes of traffic
    - Key disruption event: 9/11
    - Akamai technologies created
- **Second generation (early 2000ies)**
  - Support high volumes of multimedia traffic
  - Audio/video intensive networks
    - All ISPs developed/used CDNs
- **Third generation (2010+)**
  - Cloud computing
    - Amazon cloud (2008)
  - UGC (user generated content)
  - P2P and interactivity
    - AT&T distributed data centers (2011)
  - Mobile support, and device adapted content

Internal Enterprise Applications

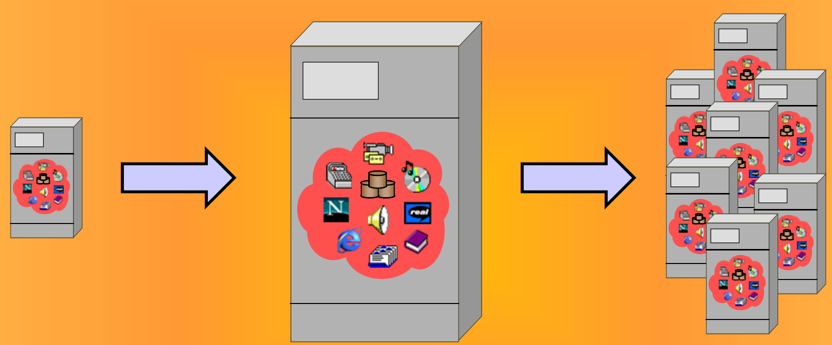


External Enterprise Applications

E-Commerce applications limited by congestion points within the Internet



Content Scaling

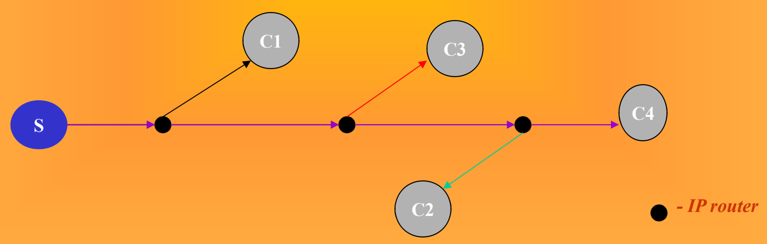


- **Need to scale content to handle numerous clients**
  - **One can only scale 'vertically' to a point**

Multiple servers/locations introduces new issues

Early Motivations for Content Networks (1st generation)

- More hops between client and Web server => more congestion!
- Same data flowing repeatedly over links between clients and Web server
- Origin server is bottleneck as number of users grows
- Flash Crowds *(for instance, Sept. 11)*
  - *The Content Distribution Problem:* Arrange a rendezvous between a content source at the origin server *(www…com)* and a content sink *(users)*

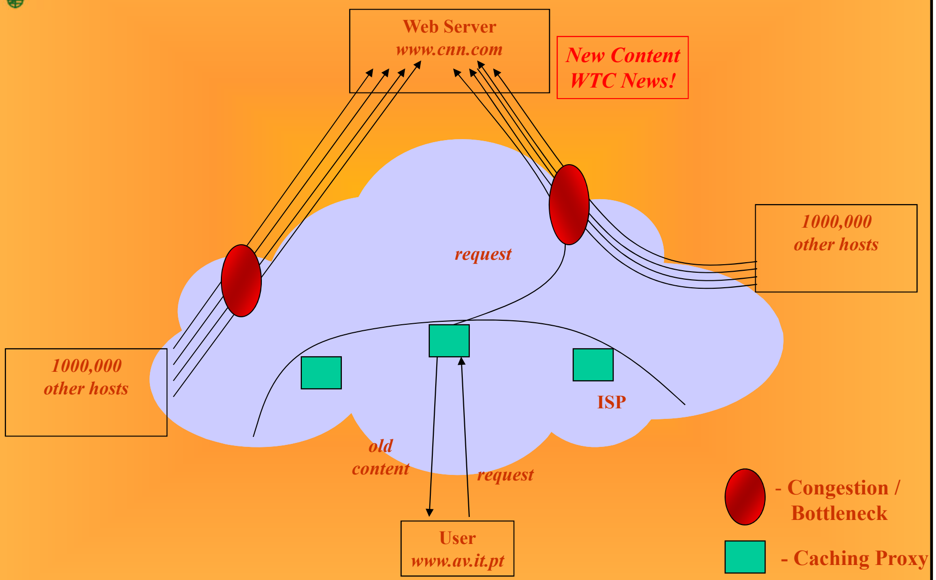


Outline

- Overall context
- **Challenges**
- Potential alternatives?
- Architecture

Flash Crowds:

Consider, On September 11, 2001



38

Flash crowd solution: CDNs..

What is a CDN?

A network of servers delivering content on behalf of an origin site

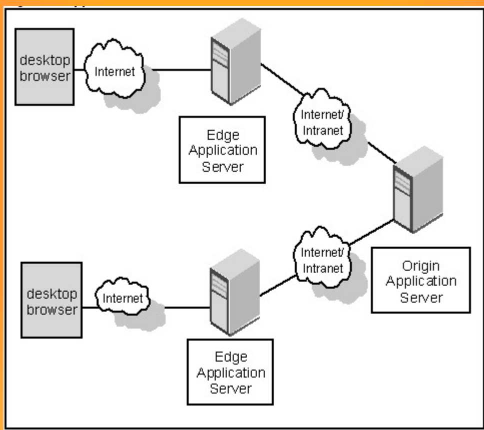
Large-file service with

- No custom client
- No custom server
- No prepositioning
- No rehosting
- No manual provisoning

39

Model

- Application offload (1st generation concern)



40

Content distribution networks

- Client attempts to access the main server site for an application
- It is redirected to one of the other sites
- Each site caches information
  - Avoid going to the main server to get the information/application
- Access a closelly located site
  - Avoid congestion on the path to the main server
- Set of sites used to improve the performance of web-based applications collectivelly
  - Content distribution network

Inside a CDN

- Servers are deployed in clusters for reliability
  - Some may be offline
    - Could be due to failure
    - Also could be “suspended” (e.g., to save power or for upgrade)
- Could be multiple clusters per location (e.g., in multiple racks)
- Server locations
  - Well-connected points of presence (PoPs)
  - Inside of ISPs

43

Advantages

- Better scalability
- Higher availability
- Improved response time from a centrally managed solution
- Nodes constituting the distribution network are designed to be
  - Self-configuring
  - Self-managing
  - Self-diagnosing
  - Self-healing

to ensure easy management and operational convenience

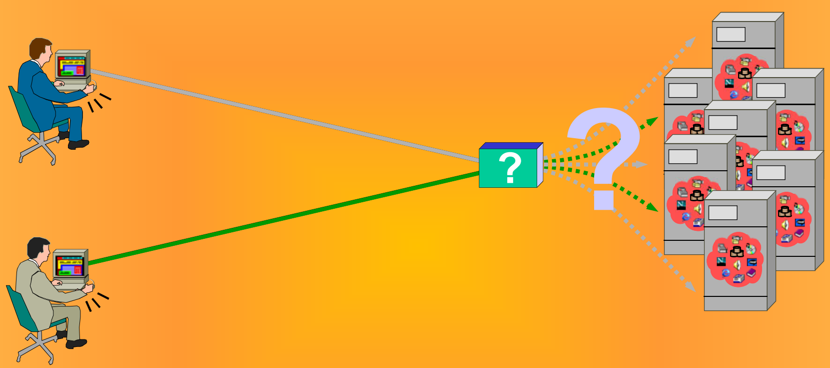
44

Challenges

- Keep consistency among the enterprise data hosted by the offloaded applications
- Share session state among edge and origin application servers
- Distribution, configuration, and management
- Develop programming models consistent with current industry standards such as J2EE
- Application security.

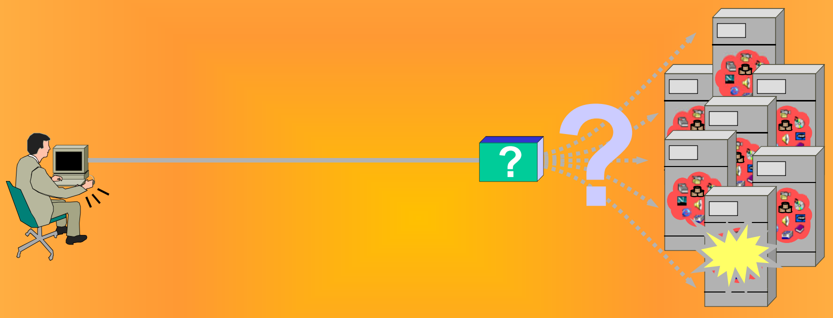
- There is active research into general frameworks to be used to support distributed applications, as well as prototyping the ideas for specific application instances

Load-Sharing Content



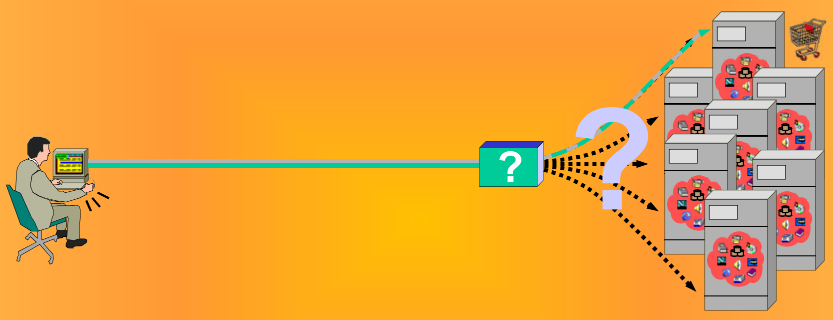
- Handle requests fairly amongst servers/sites
- Easily add servers/sites to content service
- Adjust connections based on server/site load

Content Availability with multiple servers?



- Synchronize content amongst servers/sites
- Avoid faulty servers/sites
- Faulty servers/sites includes invalid/dated content

Persistence with multiple servers?



- Handle applications which use ‘state’
  - Need to learn client ID to satisfy state requirement
  - Need to maintain state for period of time - variable

48

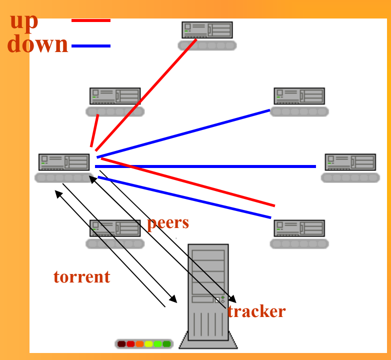
Outline

- Overall context
- Challenges
- **Potential alternatives?**
- Architecture

49

Peer-to-Peer?

- BitTorrent has peaked at ~30% Internet BW



1. Download a "torrent" file
2. Contact the tracker
3. Enter the "swarm" network
4. Chunk exchange policy
   - Rarest chunk first or random
   - Tit-for-tat: incentive to upload
   - Optimistic unchoking
5. Validate the checksums

Benefit: extremely good use of resources!

50

Peer-to-Peer for CDNs?

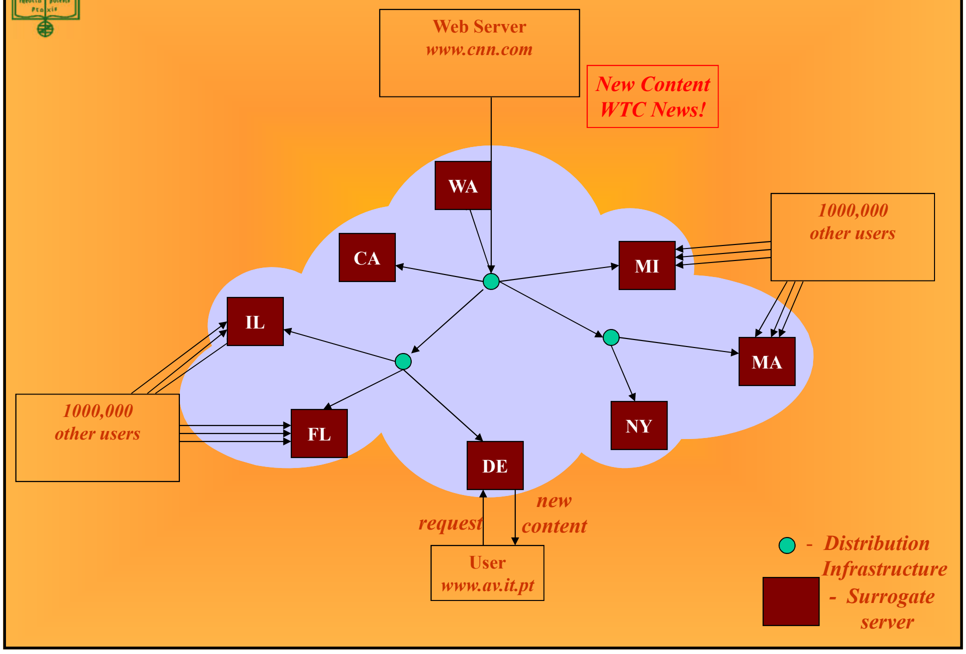
- Custom software
  - Deployment is a must
  - Configurations needed
- Companies may want managed service
  - Handles flash crowds
  - Handles long-lived objects
- Performance problem
  - Hard to guarantee the service quality
  - Others are discussed later

51

Outline

- Overall context
- Challenges
- Potential alternatives?
- Architecture

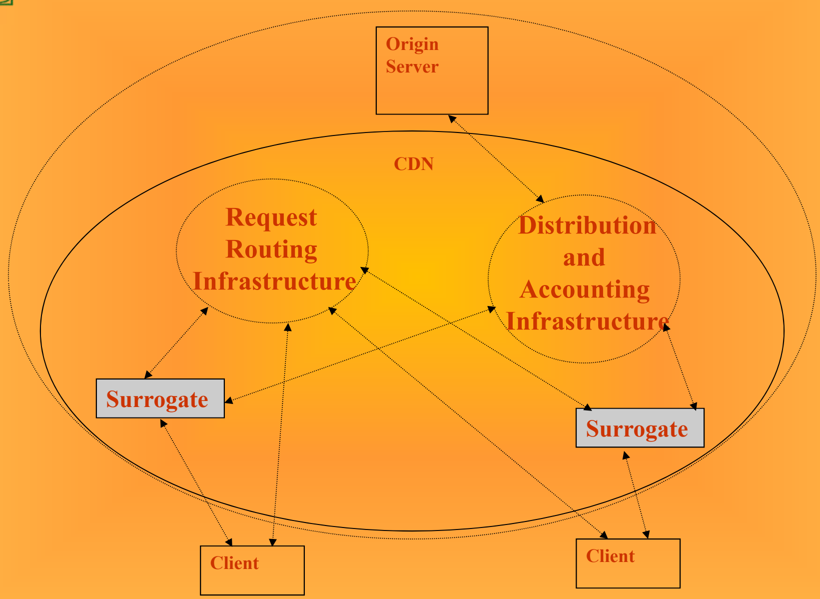
CDN, On September 11, 2001



With CDNs

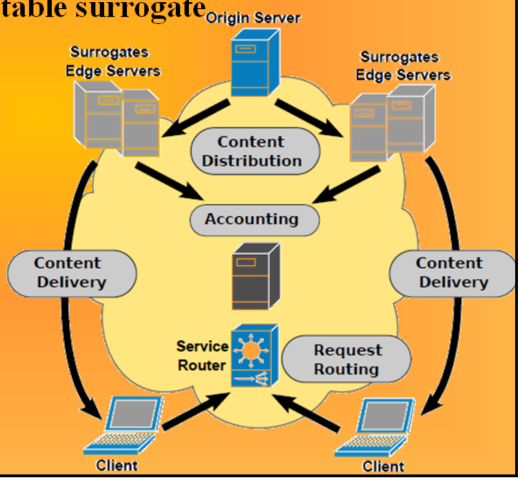
- Overlay network to distribute content from origin servers to users
  - Avoids large amounts of same data repeatedly traversing potentially congested links on the Internet
  - Reduces Web server load
  - Reduces user perceived latency
  - Tries to route around congested networks
- CDN is not a cache!
  - Caches are used by ISPs to reduce bandwidth consumption, CDNs are used by content providers to improve quality of service to end users
  - Caches are reactive, CDNs are proactive
  - Caching proxies cater to their users (web clients) and not to content providers (web servers), CDNs cater to the content providers (web servers) and clients
  - CDNs give control over the content to the content providers, caching proxies do not

CDN Architecture



CDN Components

- *Content Delivery Infrastructure:* Delivering content to clients from surrogates
- *Request Routing Infrastructure:* Steering or directing content request from a client to a suitable surrogate
- *Distribution Infrastructure:* Moving or replicating content from content source (origin server, content provider) to surrogates
- *Accounting Infrastructure:* Logging and reporting of distribution and delivery activities

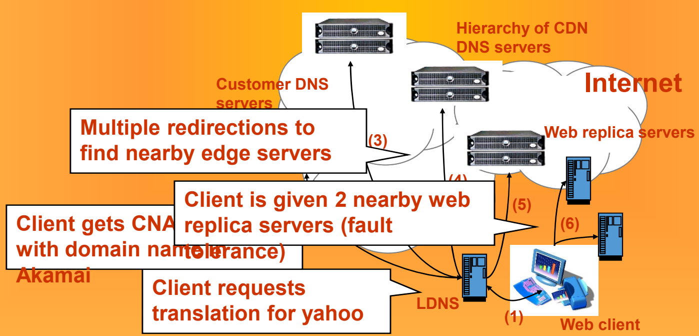


Mapping clients to servers

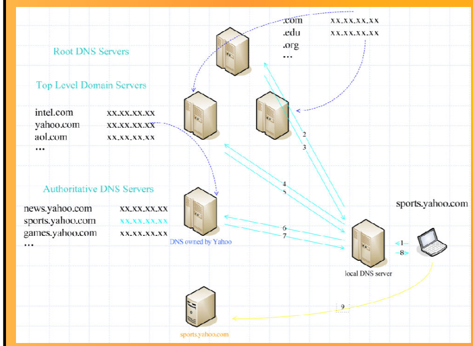
- CDNs need a way to send clients to the "best" server
  - The best server can change over time
  - And this depends on client location, network conditions, server load, …
  - What existing technology can we use for this?
- DNS-based redirection
  - Clients request www.foo.com
  - DNS server directs client to one or more IPs based on request IP
  - Use short TTL to limit the effect of caching

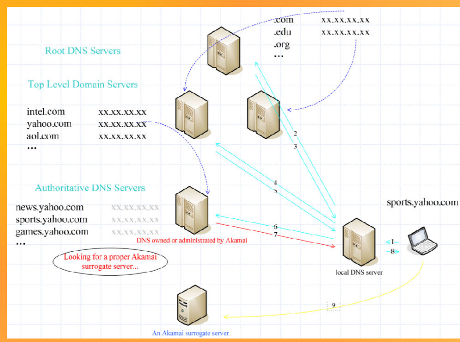
DNS Redirection

- Web client's request redirected to 'close' by server
  - Client gets web site's DNS CNAME entry with domain name in CDN network
  - Hierarchy of CDN's DNS servers direct client to 2 nearby servers



DNS Redirection



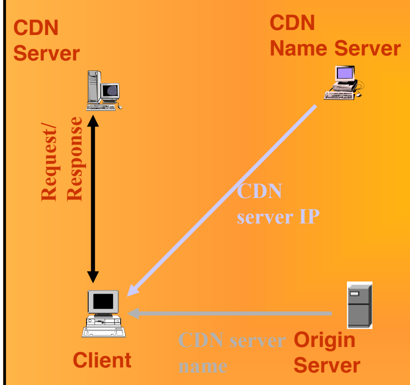


DNS Redirection Considerations

- **Advantages**
  - **Uses existing, scalable DNS infrastructure**
  - **URLs can stay essentially the same**
- **Limitations**
  - **DNS servers see only the DNS server IP**
    - Assumes that client and DNS server are close. Is this accurate?
  - **Content owner must give up control**
  - **Unicast addresses can limit reliability**

60

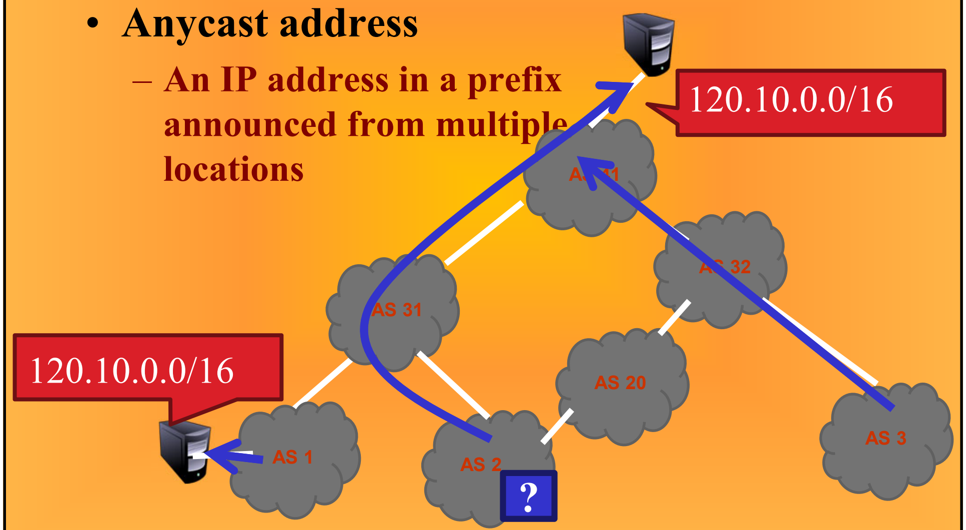
What other CDN techniques are being used?



- DNS redirection (DR)
  - Full-site delivery
  - Partial-site delivery
- URL rewriting
- Hybrid scheme
  - URL rewriting + DNS redirection
- Manual hyperlink selection
- HTTP redirection
- Layer 4 switching
- Layer 7 switching
- Anycast

CDN Using Anycast

- Anycast address
  - An IP address in a prefix announced from multiple locations



62

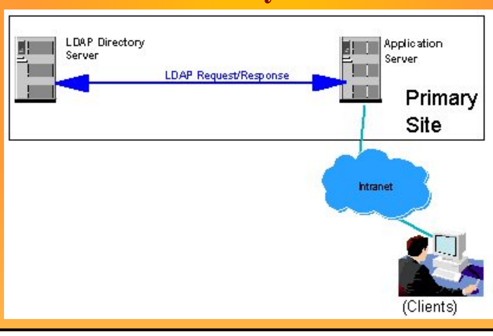
Offloading a portal

- Portal servers allow users to access content and applications from a single access point
  - Users can create persistent, customized views of applications and content chosen from the set of applications and content by the portal administrators
- Portal server pages are personalized
- Often include dynamic content
- Significant amount of computation required for page assembly
  - Application offload

63

Offloading an Enterprise directory

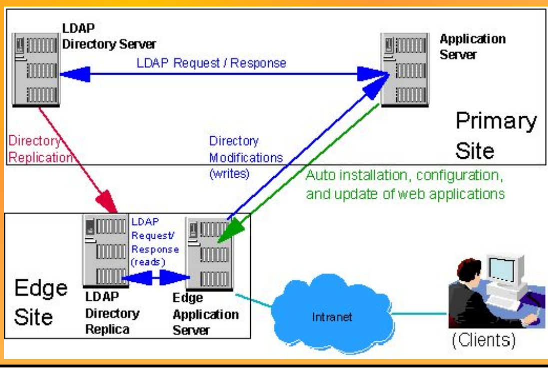
- E.g. a common e-Workplace tool
- The employee data is often stored in a central LDAP directory
  - Separate web-based application providing the interface to the directory



64

Offloading an Enterprise directory

- **Centralized directory**
  - **Convenient to manage**
  - **Performance for clients accessing the directory from remote sites can be poor**
    - E.g. transcontinental network connections suffer from a long delay
- **Offloaded version of the application**



A CDN Architecture