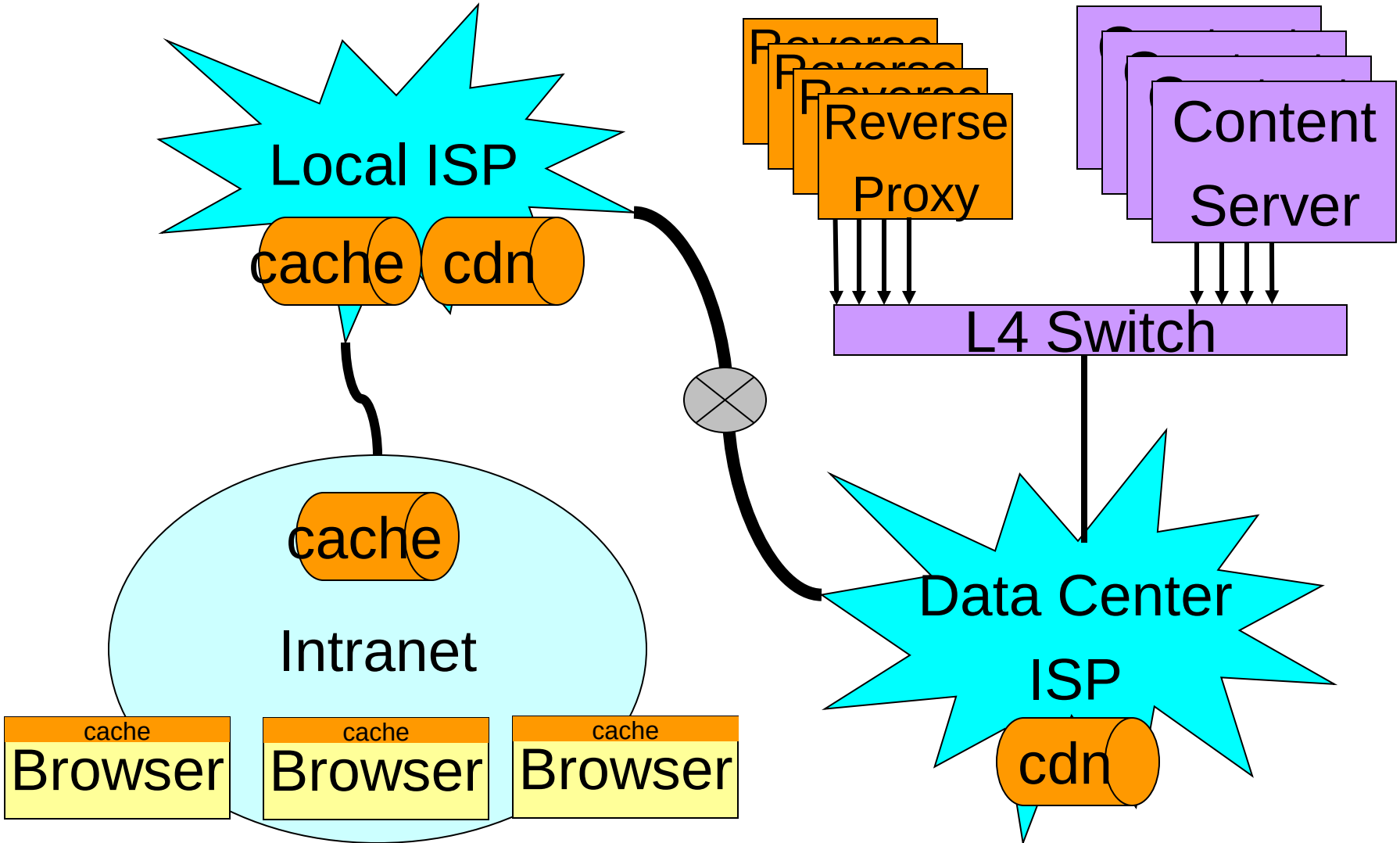
Content Delivery Networks (CDN)

Dr. Yingwu Zhu

Web Cache Architecure



History

- 1998 – 1st CDNs appear. Save $ by putting more web sites on a CDN, reliability and scalability without expensive hardware and management
- 1999 – several companies (Akamai, Mirror Image) became the specialists in providing fast and reliable delivery of Web content, earning large profits
- 2000 – U.S. only, CDNs are a huge market generating $905 millions, reaching $12 billion by 2007
- 2001 – the flash crowd event (numerous users access a web site simultaneously), e.g., Sept. 11 2001 when users flooded popular news sites, making the sites unavailable. Flash events transfer more $ to CDN sale income
- 2002 – Large-scale ISPs (AT&T) tend to build their own CDN functionality, providing customized services
- 2004 – More than 3000 companies using CDNs, spending more than $20 million monthly. CDN providers doubled their revenue from streaming media operations in 2004 compared to 2003.
- 2005 – CDN revenue for both streaming video and Internet radio is estimated to grow at 40%, spending more than $450 million for delivery of news, film, sports, music and entertainment.

Content Delivery - a bit of History

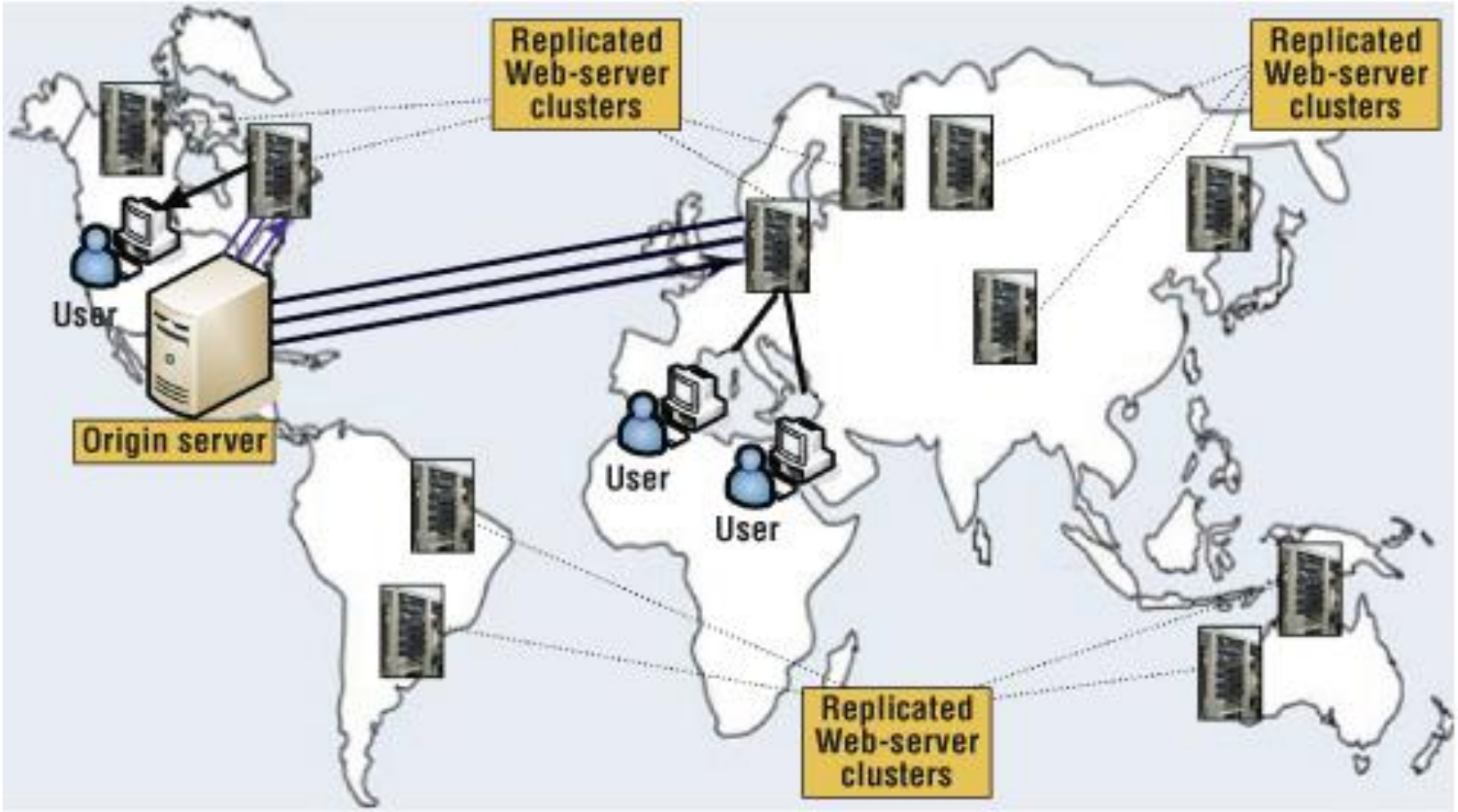
- Individual Web servers
  - Increase in Web content
- Web Server Farms
  - Issue of Flash Crowds
- Replication of same Web content around the globe in a net of Web servers
  - Not financially viable for individual content providers (say, bbc.com) to set up their own server networks
  - Expensive hardware, maintenance, energy cost?

Content Delivery Networks (CDN)

- *What*: Geographically distributed network of Web servers around the globe (by an individual provider, E.g. Akamai).
  - Many ISP points of presence (POP)
- *Why:* Improve the performance and scalability of content retrieval.
- *How:* Allow content providers to replicate their content in a network of servers.

Conventional CDN Architecture

Classical Example: Akamai



- Figure Ref:http://arxiv.org/pdf/cs/0609027

Conventional CDN Architectures

- *Commercial CDN*
  - Centralized Client-Server Architecture
  - Owned by corporate companies
  - E.g: Akamai
- *Academic CDN*
  - Peer-to-peer Architecture
  - Designed to reduce the cost
  - E.g: Globule

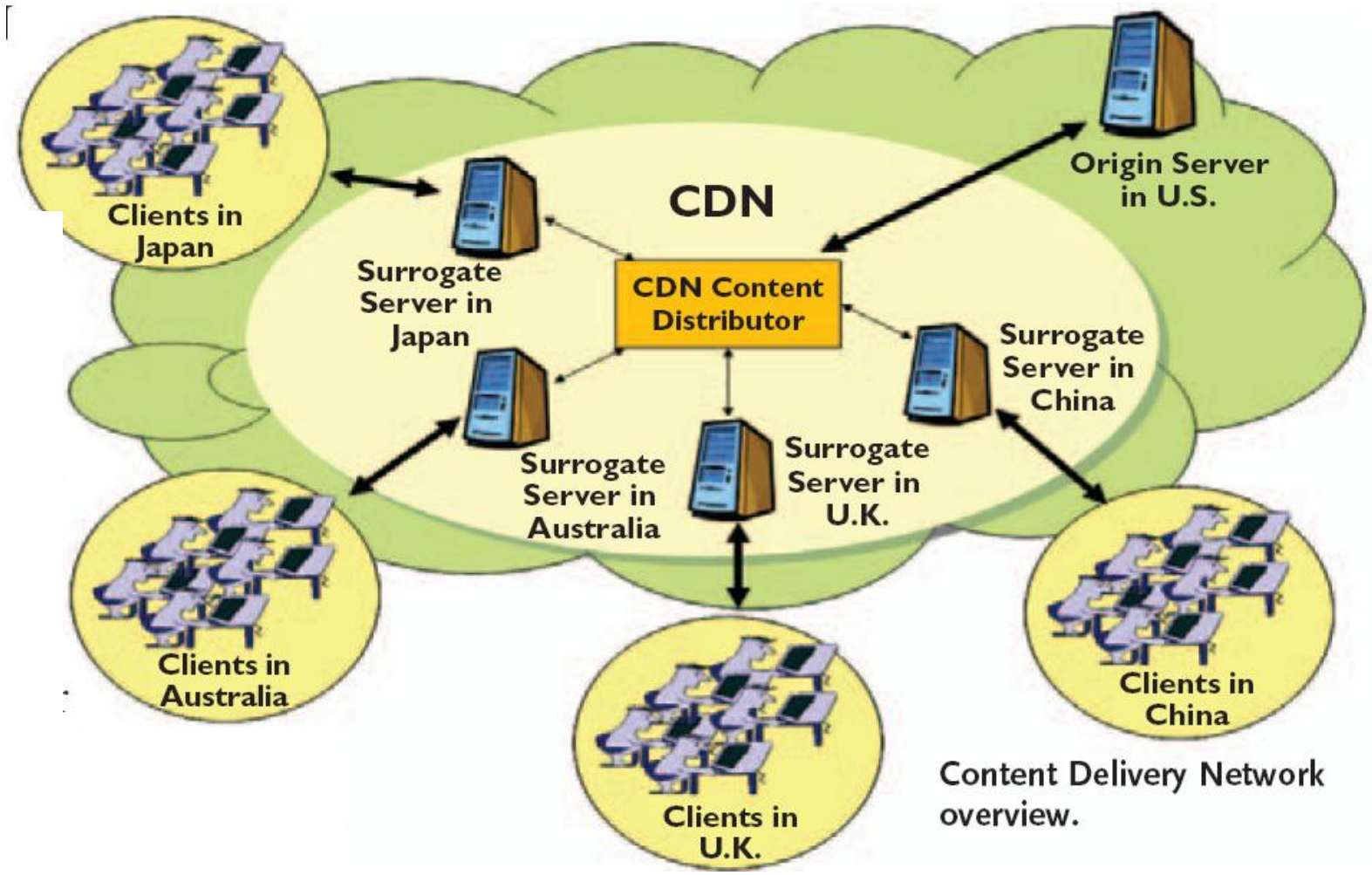
What is CDN ?

- The CDNs are means to offload some or all of the (mainly static content) content delivery burden from the *origin server*. A replica server, which delivers content on behalf of the origin server is called a *CDN server*.
- Aimed to address …
  - Client perceived latency (e.g. web browsers).
  - Capacity management of the server.
  - Caching as a side-effect.

What is CDN ?

- CDN is an architecture for efficient delivery of (web) content to a large number of clients
- CDNs are operated by companies which charge content providers for the delivery services
- CDNs are mostly transparent to the end-user
  - Meaning: You can see CDNs being used only if you look at actual DNS requests or read HTML-source of a page
- Commercial CDNs for actual content delivery:
  - Akamai, Panther Express, SAVVIS, VitalStream
- Academic CDNs for research on content delivery:
  - CoDeeN, CoralCDN, Globule

A Big Picture



Content Delivery Network overview.

Advantages of using CDN

- Reduce customers' needs in investing web site infrastructures and decrease operational cost of managing such infrastructures
- Bypass traffic jams on the web
  - Requested data is close to the clients
  - Avoid traversing bottleneck links
- Improve content delivery quality, speed, and reliability
- Reduce load on the original server
- Load balancing?

CDN – why?

- One of the main goals of CDNs is to put content provider in control over how her content is cached
- Content provider signs a contract with CDN
  - Contract specifies how content can be cached
- Contract also means CDN will follow what content provider wants
- CDNs typically charge per-byte of traffic served
- CDNs can be used for any kind of content
  - Typically main use is for web content
  - Streaming media has also been delivered over CDNs

CDN--How?

- Original servers
- A set of surrogate servers or CDN servers
  - Geographically distributed worldwide
  - Cache original servers' content
- Routers
  - deliver the client's requests to a best fitted CDN server (latency, load balancing, etc)
- Network elements
  - Distribute content from the original servers to surrogate/CDN servers
- Accounting mechanism
  - Provide logs and accounting info. to the original servers

How does CDN work?

- Users send requests to origin server
- Requests somehow intercepted by redirection service
- Redirection service forwards user's request to the "best" CDN content server
- Content served from the CDN content server

CDN- Design Issues

- CDN operates CDN *content servers*
- Content servers are placed close to users
  - In terms of network distance
- Some or all of the content from the content provider (original server) is replicated on the content servers
  - Different content servers might have different content
- Users access content from the "nearest" content server
- Challenges:
  - How to redirect clients (request redirection)?
  - How to replicate content?
    - Usually happens over a private network
    - Can optimize according to many criteria

Request Redirection

- Key to CDNs
- Select the most appropriate CDN content server for user requests
  - DNS redirection
    - Complete/full
    - Partial
  - URL rewrite

Request Redirection

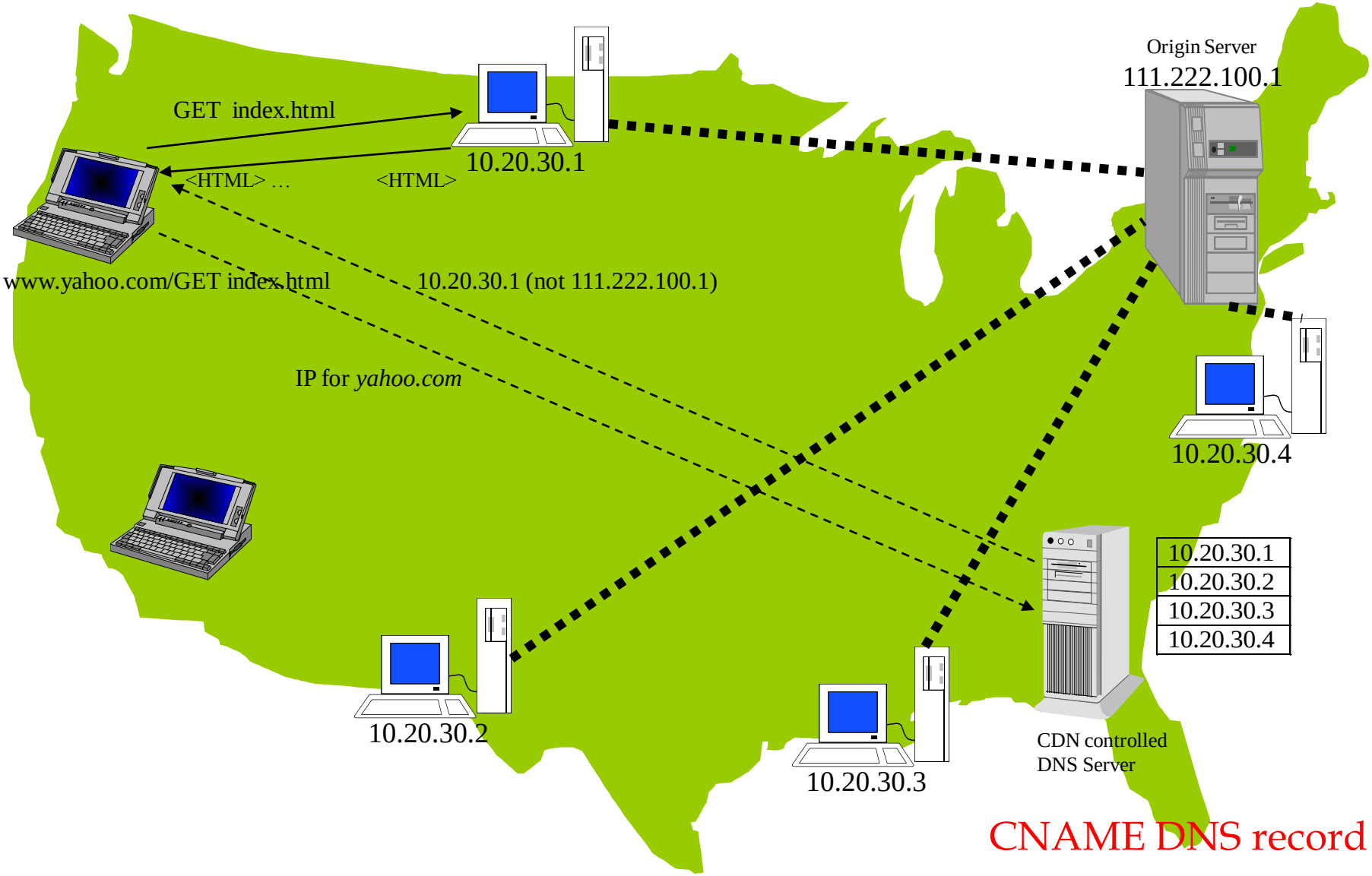
- DNS redirection

  Authoritative DNS server is controlled by the CDN infrastructure. Distributes the load to the various CDN servers depending whatever policy (e.g. round-robin, least loaded CDN server, geographical distance etc.) using DNS trick.

- URL rewriting

  Main page still comes from the origin server, but URL for the embedded objects, e.g. images, clips are rewritten, which points to a any of the CDN server. Some vendors rewrite using hostname and some uses IP address directly.

Full Site DNS redirection example



CNAME DNS record

Vendors: Adero(Full), Akami and Digital Island (Partial)

DNS Redirection

- Client's DNS request comes to CDN's nameserver
  - Somehow, see below for two possibilities
- Typically the request has to go through some steps through the CDN's DNS hierarchy
- Each step redirects the client to a nearby nameserver
- Finally, last nameserver returns the address of a nearby content server
- For the infrastructure, CDN needs to measure the state of the network
  - Needed to determine which servers are the closest
  - Network measurements to determine current state

Two DNS Redirection Types

- Full redirection
  - Any request for origin server is redirected to CDN
  - Basically, CDN takes control of content provider's DNS zone
  - Benefit: All requests are automatically redirected
  - Disadvantage: May send lots of traffic to CDN, hence expensive for the content provider, $ per byte
- Partial redirection
  - Content provider marks which objects are to be served from CDN
  - Typically, larger objects like images are selected
  - Refer to images as: <img src=http://cdn.com/foo/bar/img.gif>
  - When client wants to retrieve image, DNS request for cdn.com gets resolved by CDN and image is fetched from the selected content server
  - Pro: Fine-grained control over what gets delivered
  - Con: Have to (manually) mark content for CDN

Two DNS Redirection Types

- Full redirection
  - All requests redirected to content servers
- Partial redirection
  - Get HTML page from origin server, images from content server
  - Need to open new TCP connection for images

DNS Redirection: other issues

- DNS redirection has one (big) problem
  - Because redirection is based on DNS queries, the content server is chosen based on who sent that query
- DNS queries do not come from clients, but from the DNS servers used by the clients
- *Why is this a problem?*
- In many cases it's not a problem
  - For example, clients in a university use university's nameserver
- In many cases, it's a big problem
  - Larger ISPs might run only a few nameservers
  - Especially in US for dial-up users, DNS lookups are concentrated
  - This means the content server is optimized for the nameserver, not the actual client
  - The difference can sometimes be very large

URL rewrite

- Modify pages at the origin server on the fly
- Change embedded URL's based on up-to-date knowledge of the network and CDN server loads
- Does not require additional DNS lookups
- Fasttide, Clearway

Partial DNS redirect/URL rewriting example

index.html

```
<HTML>
<BODY>
<A HREF="/about_us.html"> About Us </A>
<IMG SRC="www.clearway1.net/www.yahoo.com/img1.gif">
<IMG SRC="www.clearway2.net/www.yahoo.com/img2.gif">
<IMG SRC="10.20.30.2/www.yahoo.com/img3.gif">
</BODY>
</HTML>
```

Vendors: Clearway (URL RW)

CDN: other issues

- Content server placement
- Content selection
- Content outsourcing

Content Server Placement

- Minimize user-perceived latency
  - Put content servers close to the users
- Minimize cost
  - Content outsourcing cost
- Algorithms to achieve both

Content selection

- How much content should be replicated to content server?
- Full site replication
  - Simple, but high storage cost, outsourcing cost
- Partial replication
  - Content grouping based on correlation or access frequency
  - Replicate content groups

Content Outsourcing

- Cooperating push-based
  - Content is prefetched to content servers from the original server
  - Content servers cooperate in order to reduce the replication and update cost
  - CDNs maintain the mapping between content and content servers

Some Facts ...

- CDN mainly used for image files (static contents).
- Content server by the CDN is a static in the nature. Only 0.3% content changed for existing URLs and at the most 13% new URLs were introduced.
- Large increase in deployment in the CDN between Nov 99 (only 1-2% of top 670 sites) and Dec 2000 (25% of the popular sites).
- Akamai seems to be most popular CDN vendor.
- Images are 96-98% of the CDN served contents. But only 40-46% of the CDN-served bytes. Rest is dynamic content ?
- CDN images cache-hit rate is 30-80%.
- CDNs can not be used for something that involves authentication etc.