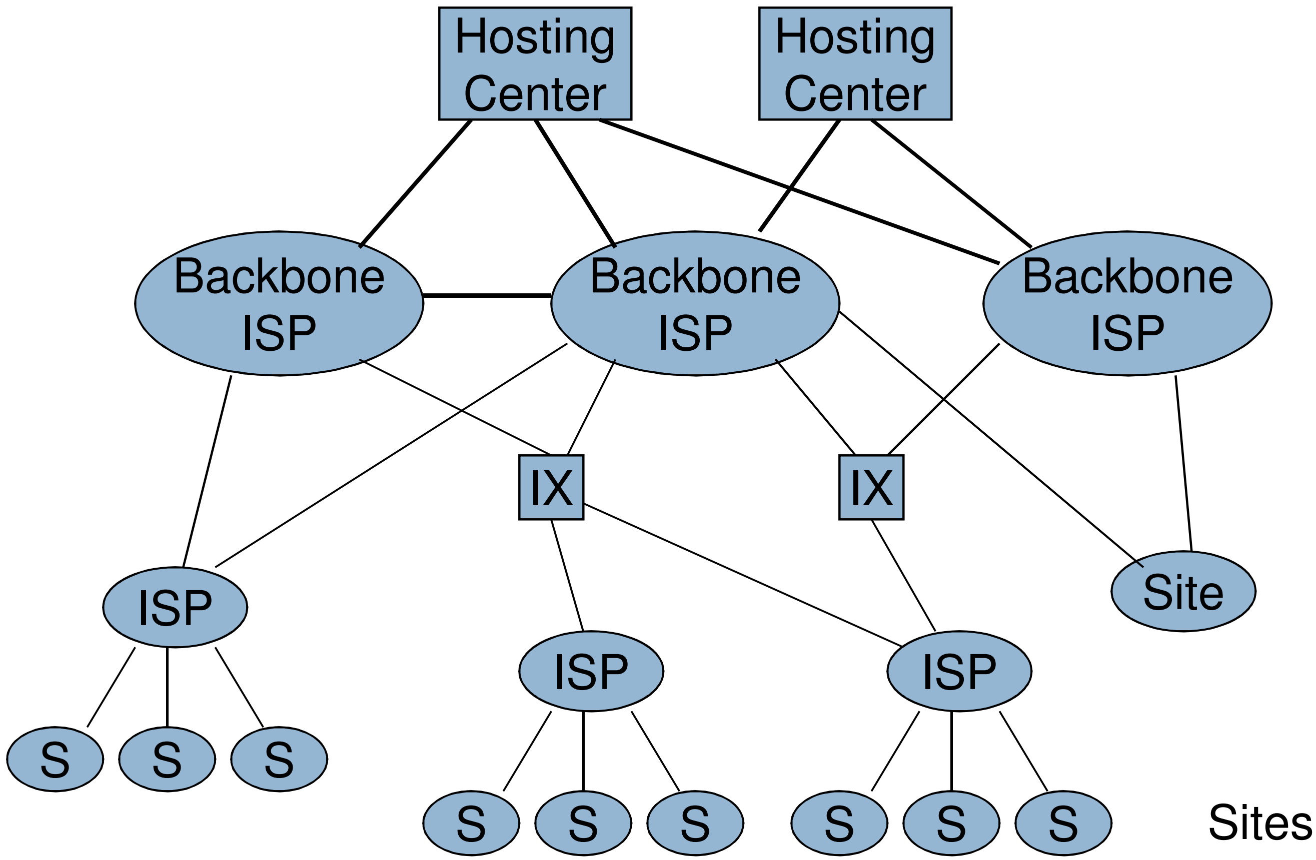
RELIABLE DISTRIBUTED SYSTEMS

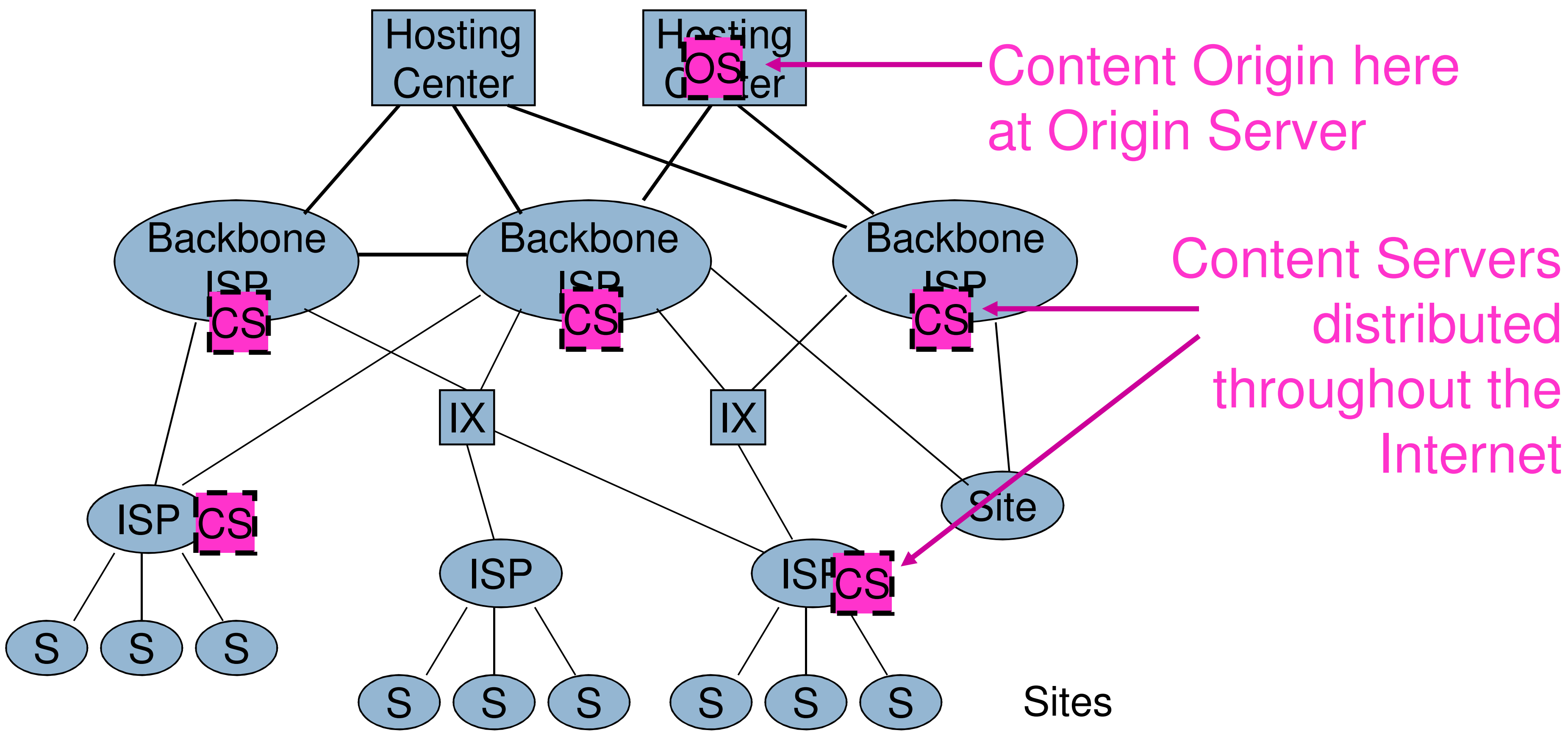
A glimpse into the world of Content Distribution Networks

Based on a slide set developed by Prof. Paul Francis

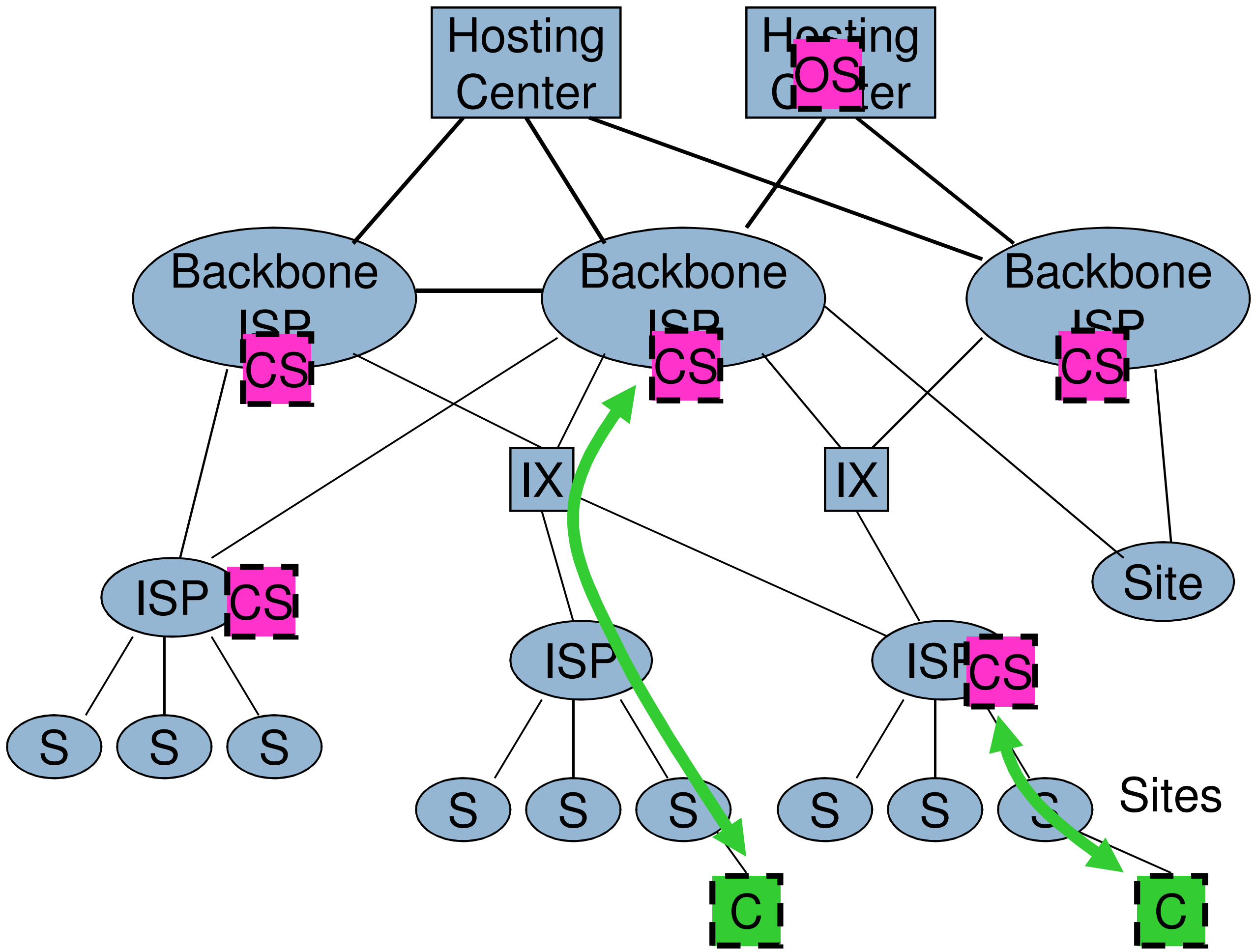
Content Routing Principle
(a.k.a. Content Distribution Network)



Content Routing Principle
(a.k.a. Content Distribution Network)

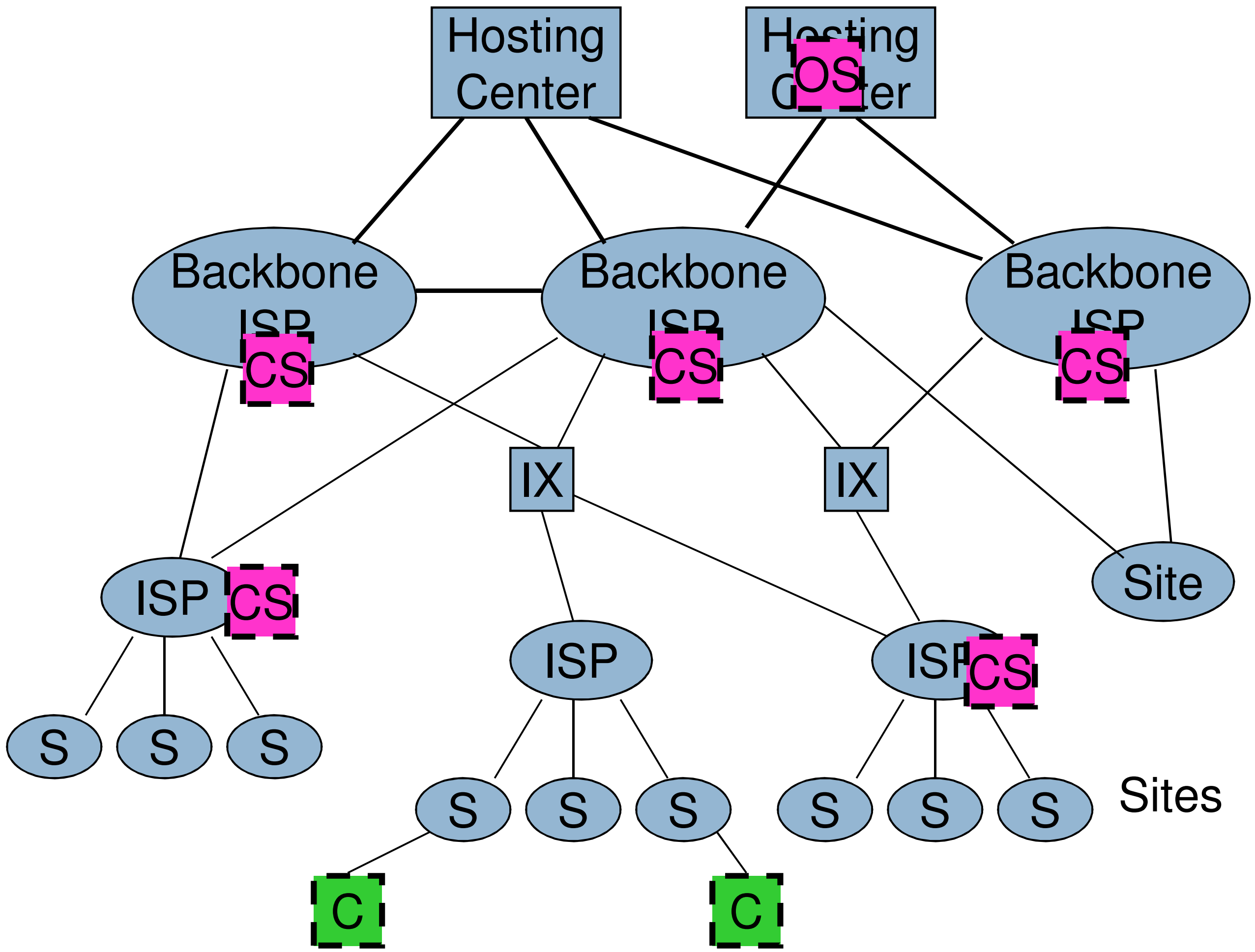


Content Routing Principle
(a.k.a. Content Distribution Network)

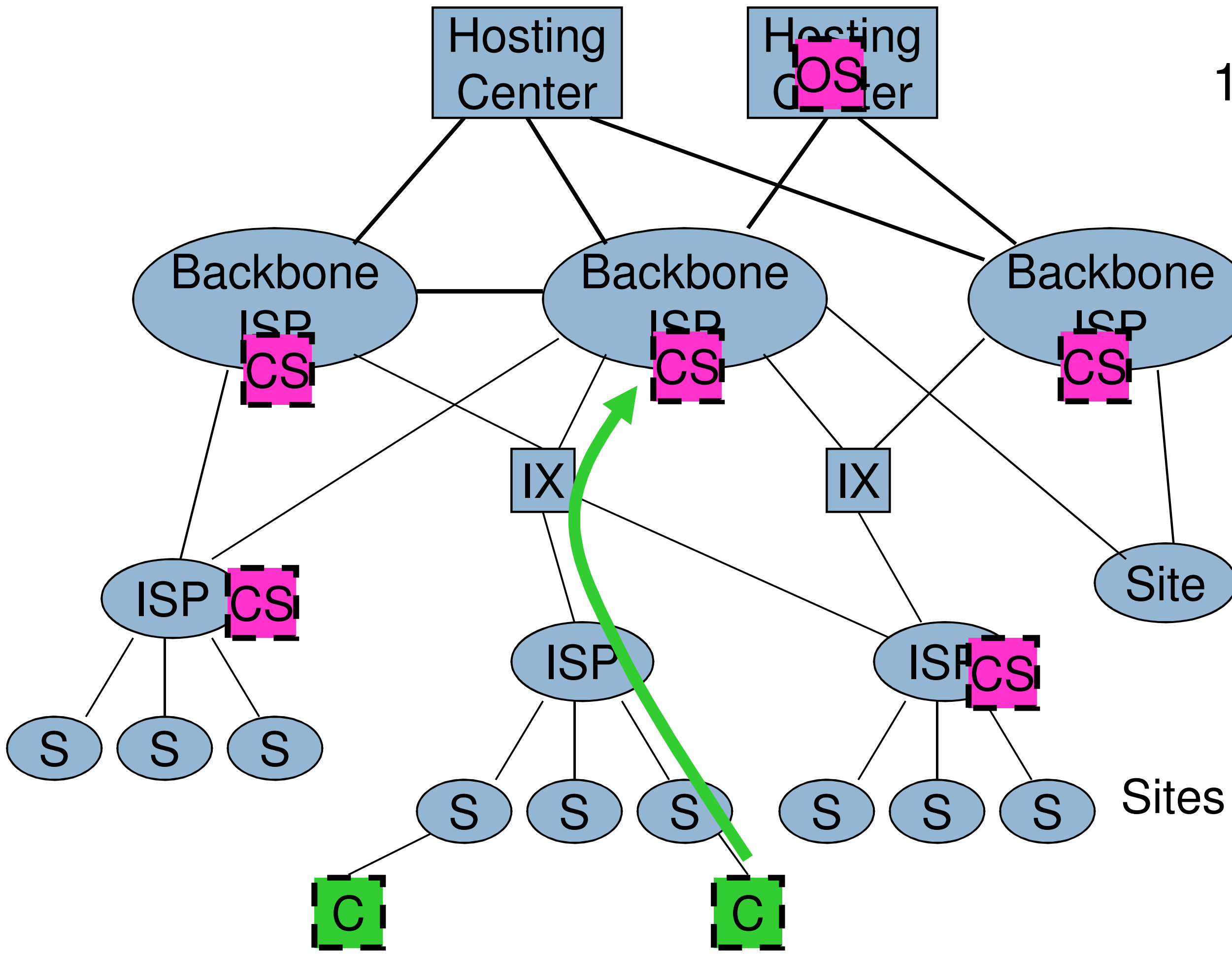


Content is served from content servers nearer to the client

Two basic types of CDN: cached and pushed

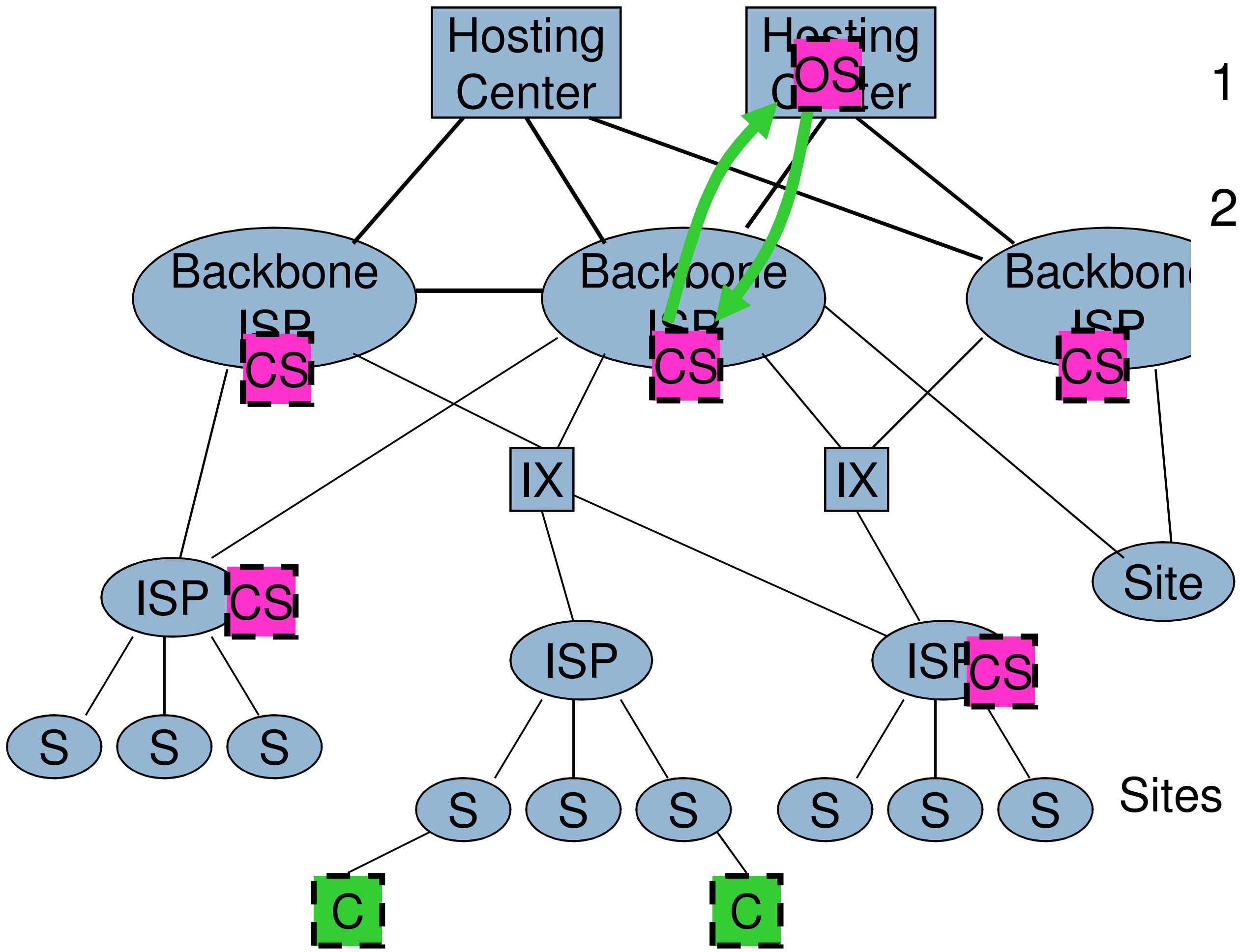


Cached CDN



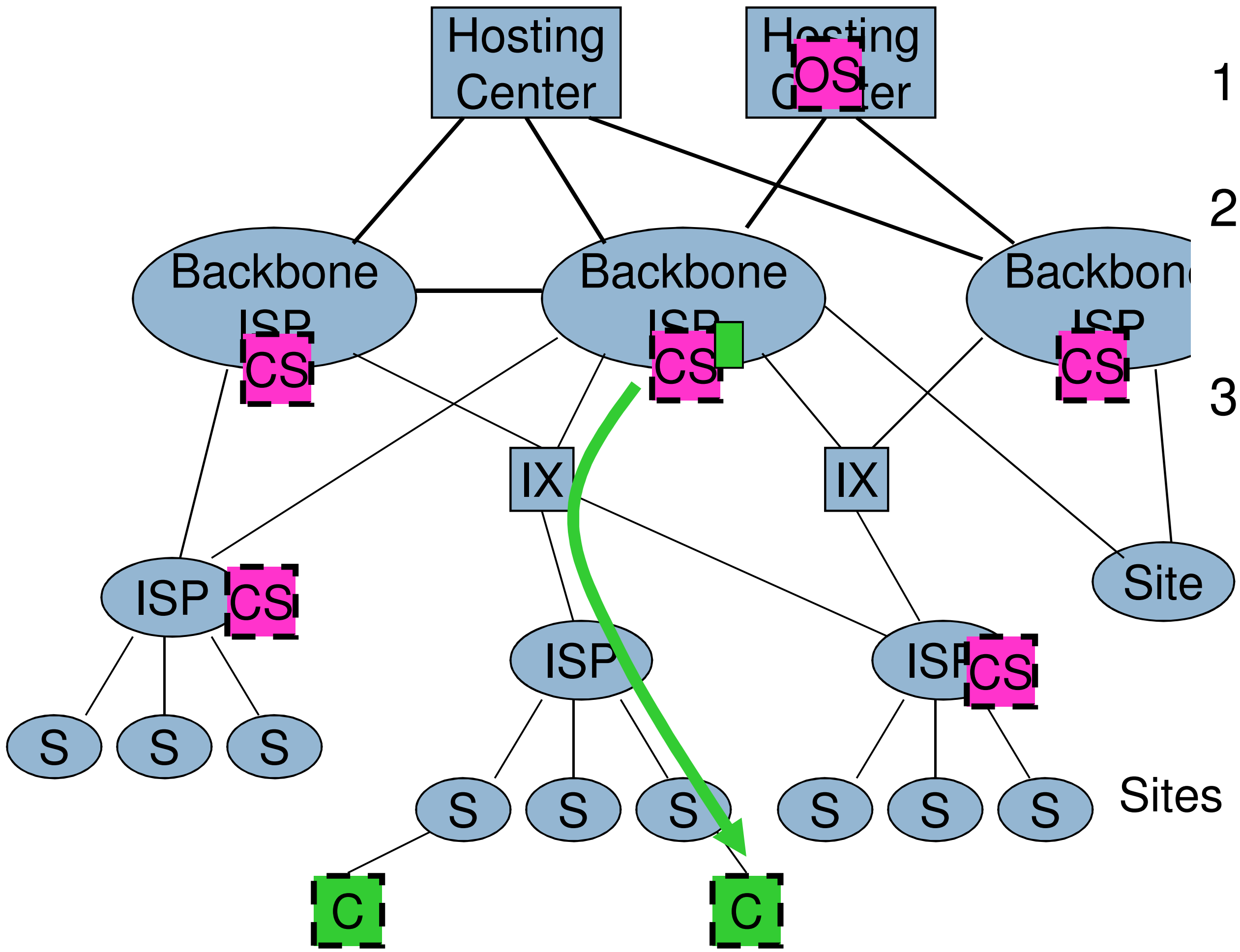
1. Client requests content.

Cached CDN



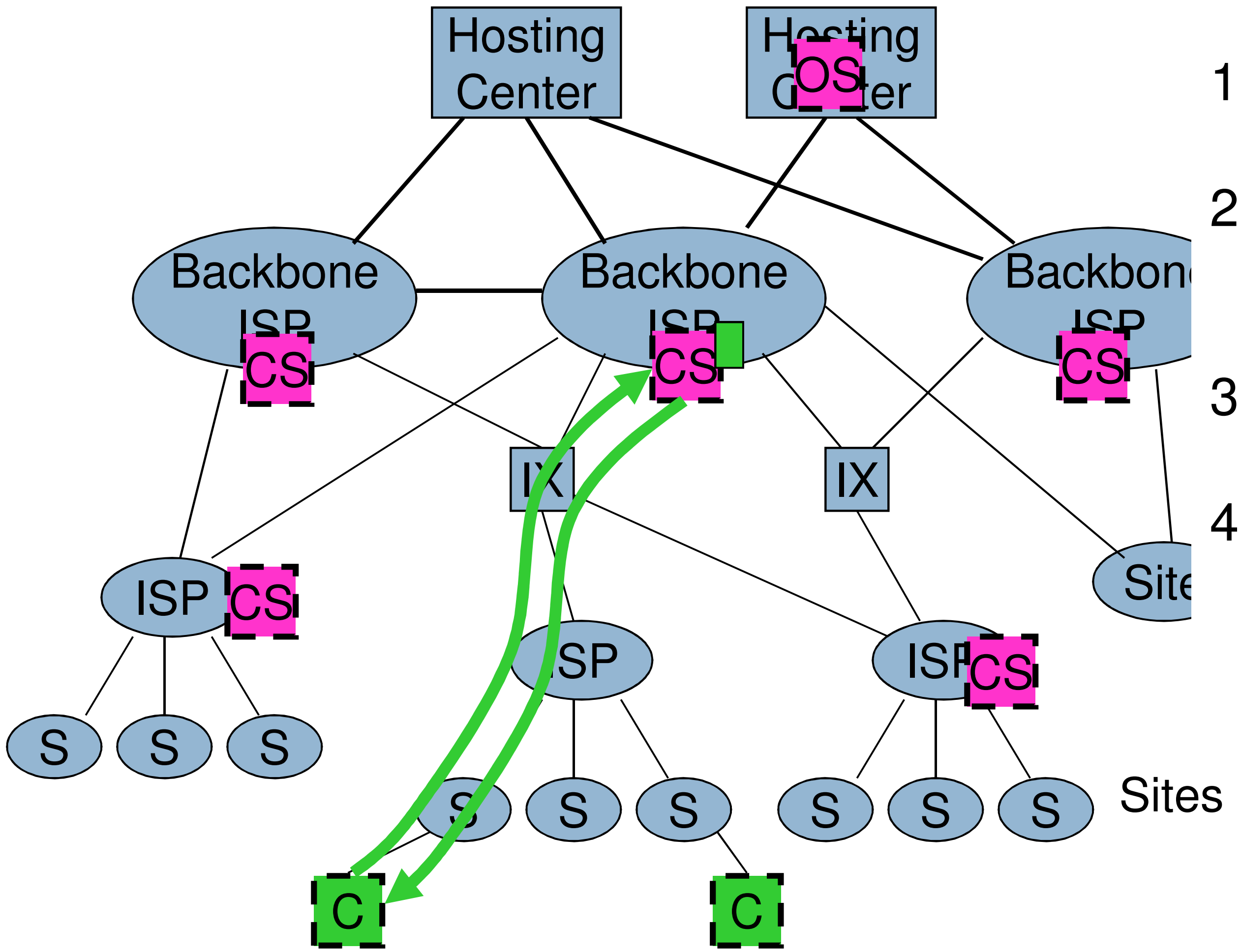
1. Client requests content.
2. CS checks cache, if miss gets content from origin server.

Cached CDN



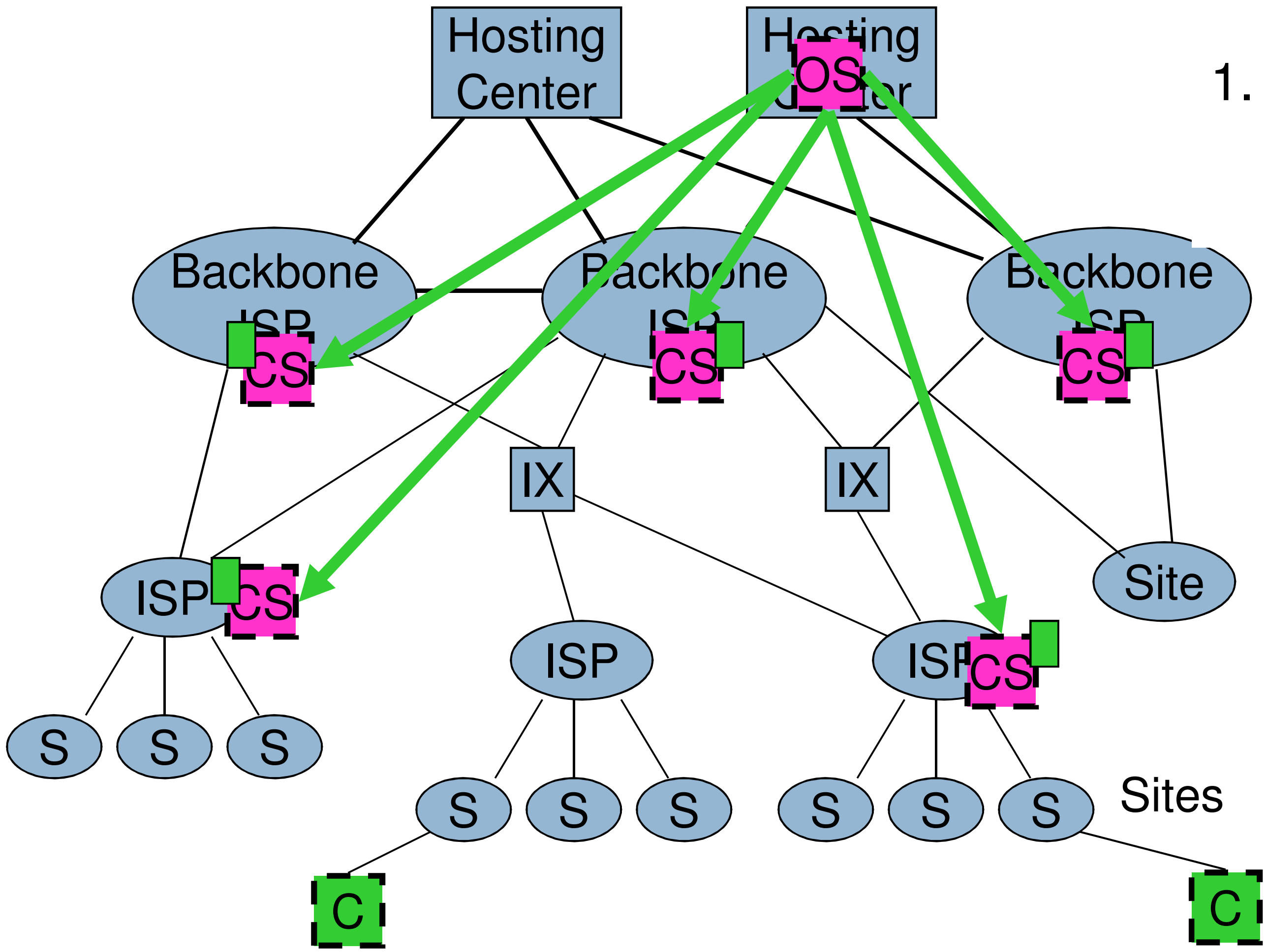
1. Client requests content.
2. CS checks cache, if miss gets content from origin server.
3. CS caches content, delivers to client.

Cached CDN



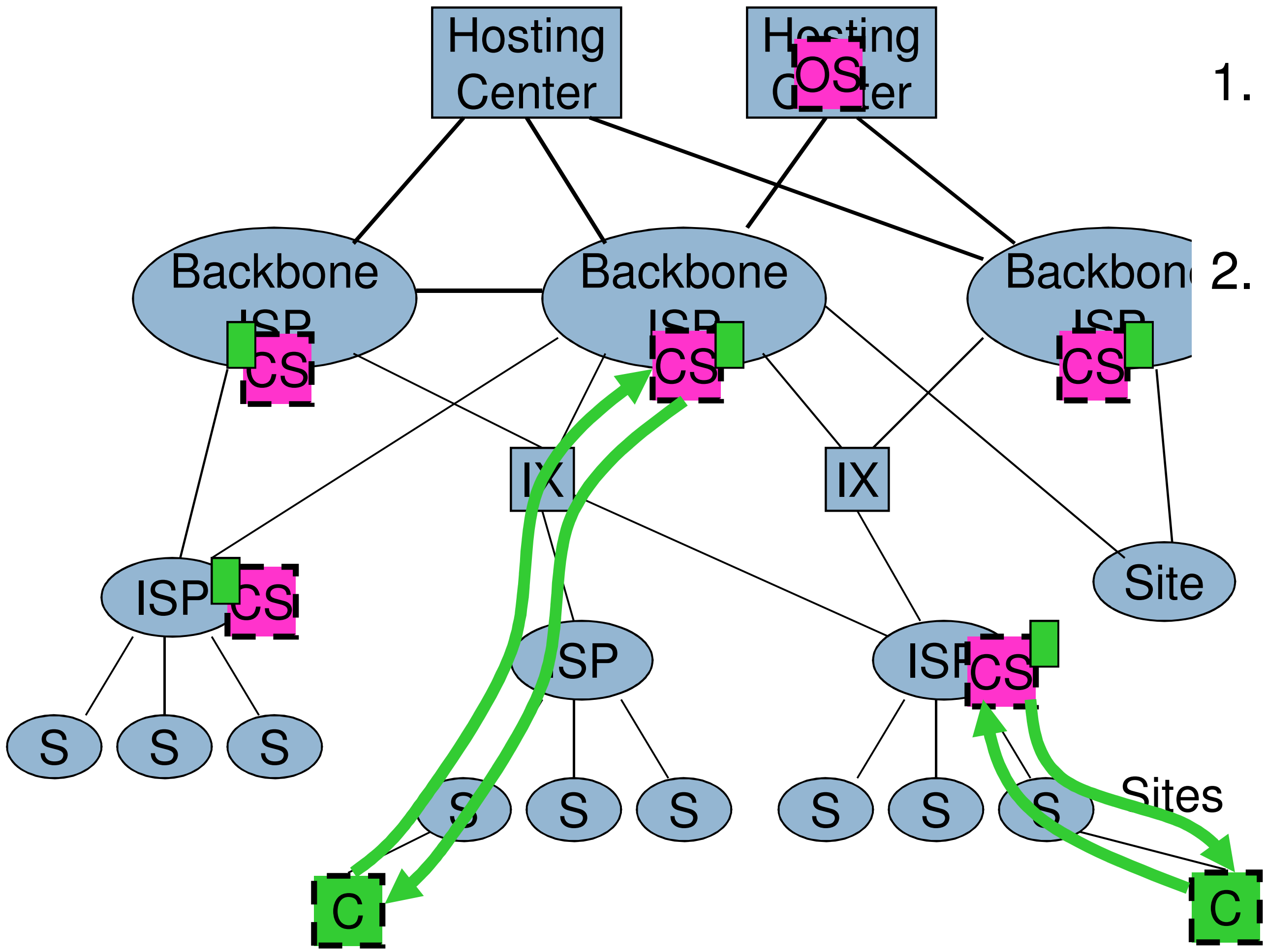
1. Client requests content.
2. CS checks cache, if miss gets content from origin server.
3. CS caches content, delivers to client.
4. Delivers content out of cache on subsequent requests.

Pushed CDN



1. Origin Server pushes content out to all CSs.

Pushed CDN



1. Origin Server pushes content out to all CSs.
2. Request served from CSs.

CDN benefits

- Content served closer to client
  - Less latency, better performance
- Load spread over multiple distributed CSs
  - More robust (to ISP failure as well as other failures)
  - Handle flashes better (load spread over ISPs)
  - *But well-connected, replicated Hosting Centers can do this too*

CDN costs and limitations

- Cached CDNs can't deal with dynamic/personalized content
  - More and more content is dynamic
  - "Classic" CDNs limited to images
- Managing content distribution is non-trivial
  - Tension between content lifetimes and cache performance
  - Dynamic cache invalidation
  - Keeping pushed content synchronized and current

CDN example: Akamai

- Won huge market share of CDN business late 90's
- Cached approach
- Offers full web hosting services in addition to caching services
  - Called edgesuite

Akamai caching services
ARL: Akamai Resource Locator

http://a620.g.akamai.net/7/620/16/259fdbf4ed29de/www.cnn.com/i/22.gif

Host Part — Akamai Control Part — Content URL

a620.g.akamai.net/

/7/620/16/259fdbf4ed29de/

/www.cnn.com/i/22.gif

Thanks to Ratul Mahajan, Univ of Wash, “How Akamai Works”

ARL: Akamai Resource Locator

http://a620.g.akamai.net/7/620/16/259fdbf4ed29de/www.cnn.com/i/22.gif

Content Provider (CP) selects which content will be hosted by Akamai.
Akamai provides a tool
that transforms *this CP URL* into *this ARL*

a620.g.akamai.net/

/www.cnn.com/i/22.gif

ARL: Akamai Resource Locator

http://a620.g.akamai.net/7/620/16/259fdbf4ed29de/www.cnn.com/i/22.gif

This in turn causes the client to access *Akamai's content server* instead of the *origin server*.

a620.g.akamai.net/

/www.cnn.com/i/22.gif

ARL: Akamai Resource Locator

http://a620.g.akamai.net/7/620/16/259fdbf4ed29de/www.cnn.com/i/22.gif

If Akamai's content server doesn't have the content in its cache, it retrieves it using *this URL*.

a620.g.akamai.net/

/www.cnn.com/i/22.gif

ARL Control Part

Type Code
(different types will
have different
contents)

Customer Number
(I.e. CNN, Yahoo…)

???

Content Checksum (May
be used for identifying
changed content. May
also validate content)

/7/620/16/259fdbf4ed29de/

a620.g.akamai.net/

/www.cnn.com/i/22.gif

http://a620.g.akamai.net/7/620/16/259fdbf4ed29de/www.cnn.com/i/22.gif

ARL Host Part

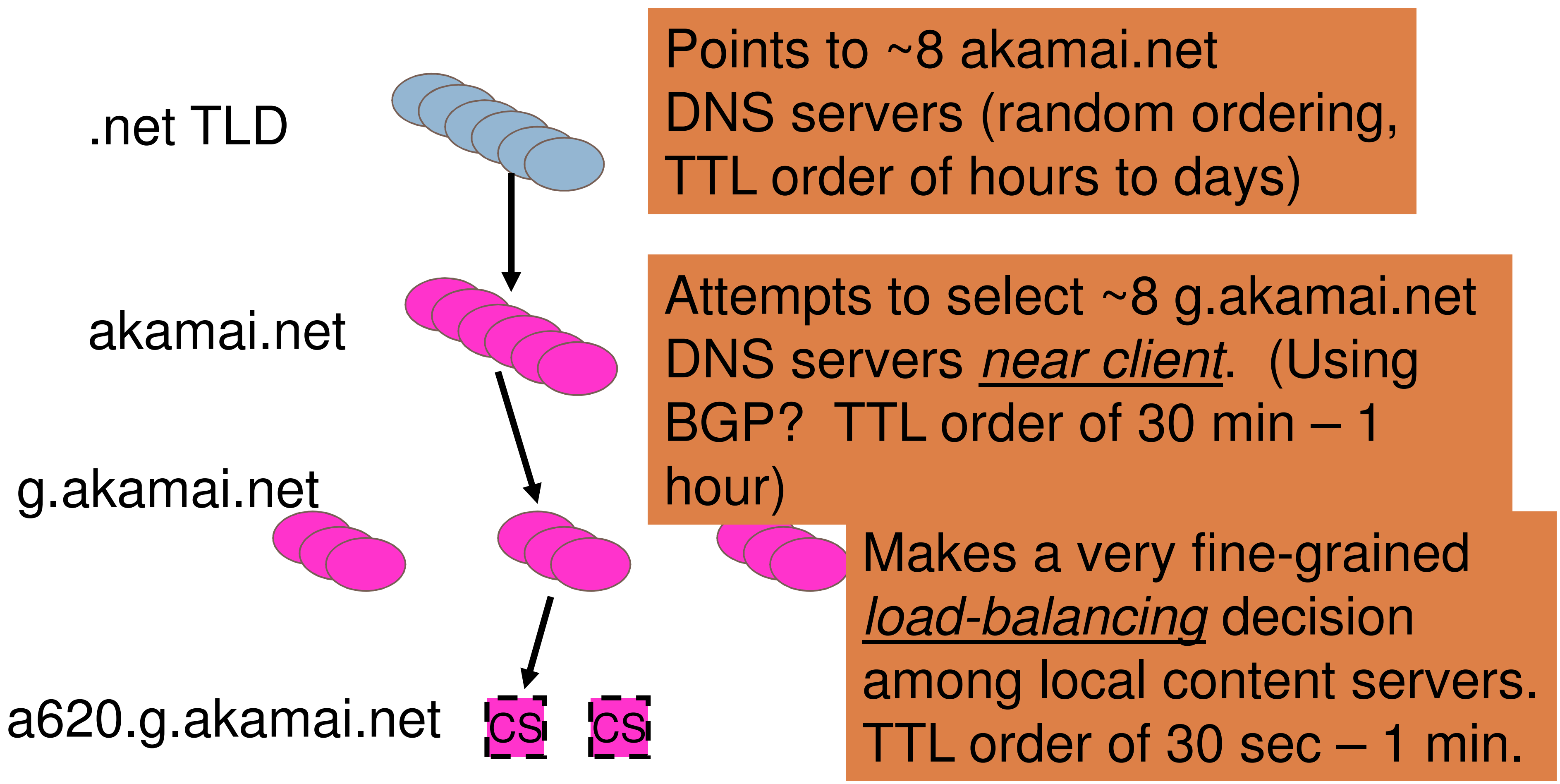
But why such a complex domain name????

/7/620/16/259fdbf4ed29de/

a620.g.akamai.net/ /www.cnn.com/i/22.gif

http://a620.g.akamai.net/7/620/16/259fdbf4ed29de/www.cnn.com/i/22.gif

ARL Host Part



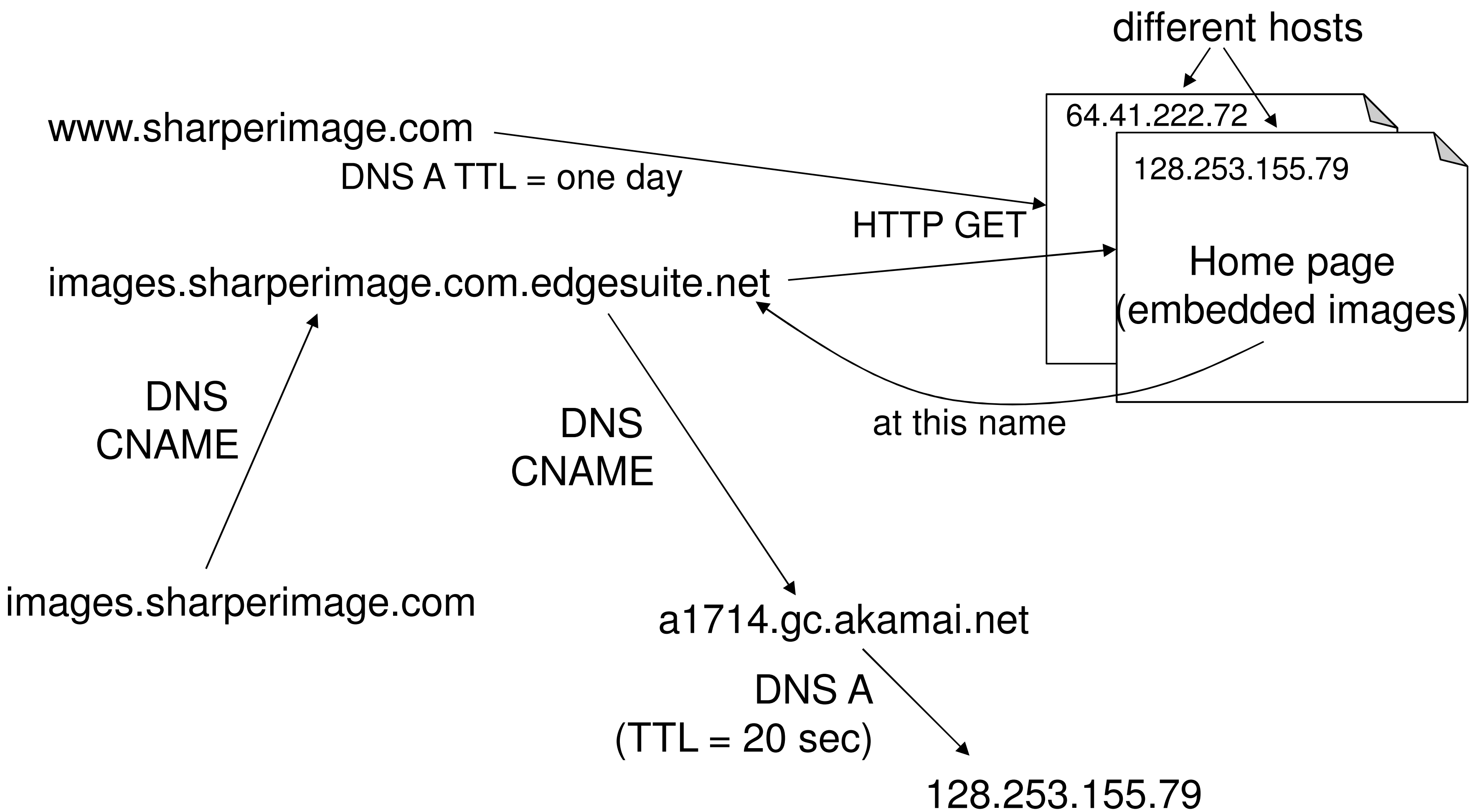
Decision factors

- Load on content servers
  - Amazon claims name servers monitor load on content servers once per second
- Proximity (number of hops) of client
  - BGP peering with ISPs that host Akamai cluster
- Internet congestion

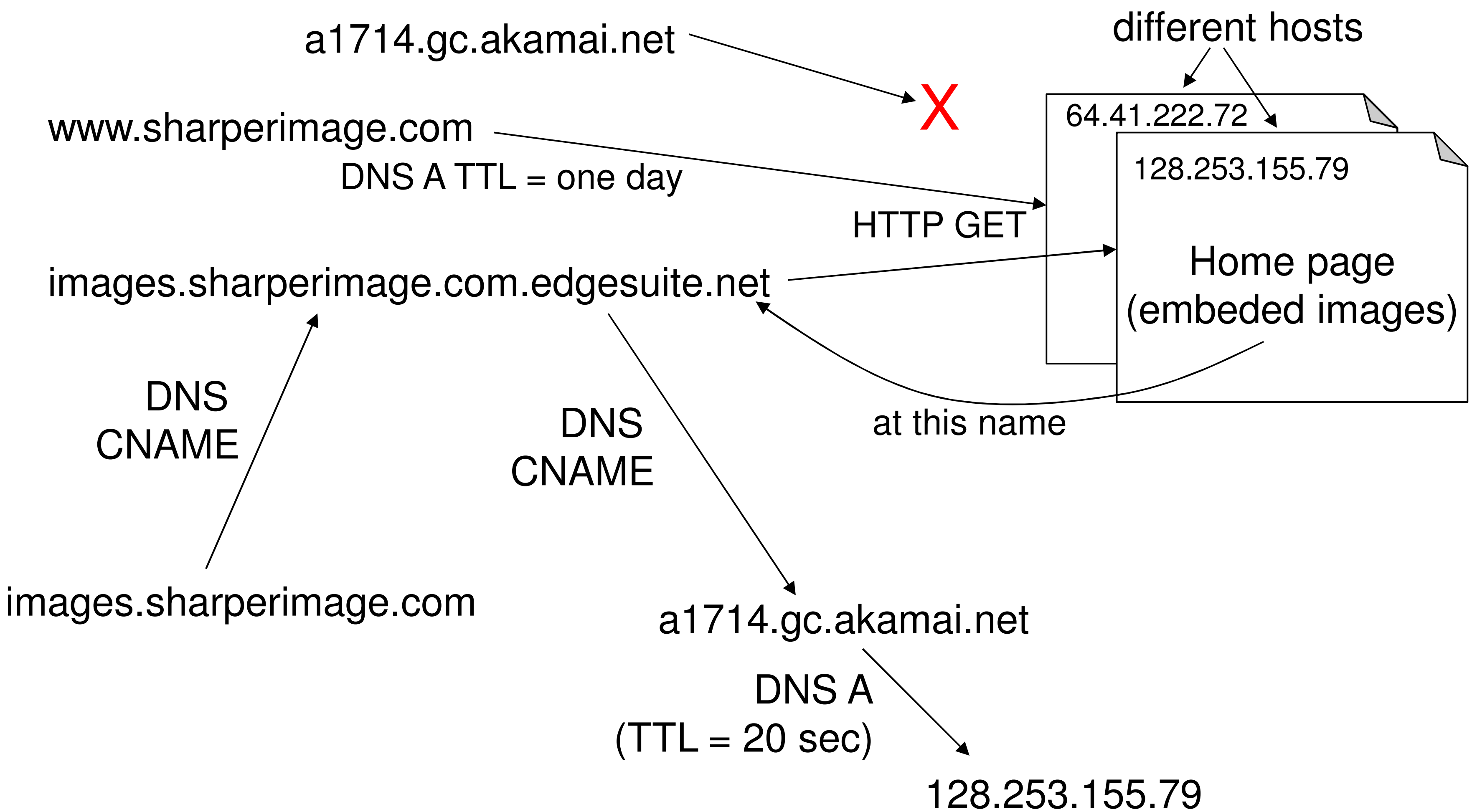
Akamai Edgesuite

- Appears that both DNS and client's web service handled by Akamai
- Also may be that content may be pushed out to edge servers---no caching!

Sharper Image and Edgesuite



Sharper Image and Edgesuite



What may be happening…

- images.sharperimage.com.edgesuite.net returns same pages as www.sharperimage.com
  - But the shopping basket doesn't work!!
- Perhaps akamai cache blindly maps foo.bar.com.edgesuite.net into bar.com to retrieve web page
  - No more sophisticated akamaization
  - Easier to maintain origin web server??
  - Simpler akamai web caches??

Other content routing mechanisms

- Dynamic HTML URL re-writing
  - URLs in HTML pages re-written to point at nearby and non-overloaded content server
  - In theory, finer-grained proximity decision
    - Because know true client, not client's DNS resolver
    - In practice very hard to be fine-grained
  - Clearway and Fasttide did this
  - Could in theory put IP address in re-written URL, save a DNS lookup
    - But problem if user bookmarks page

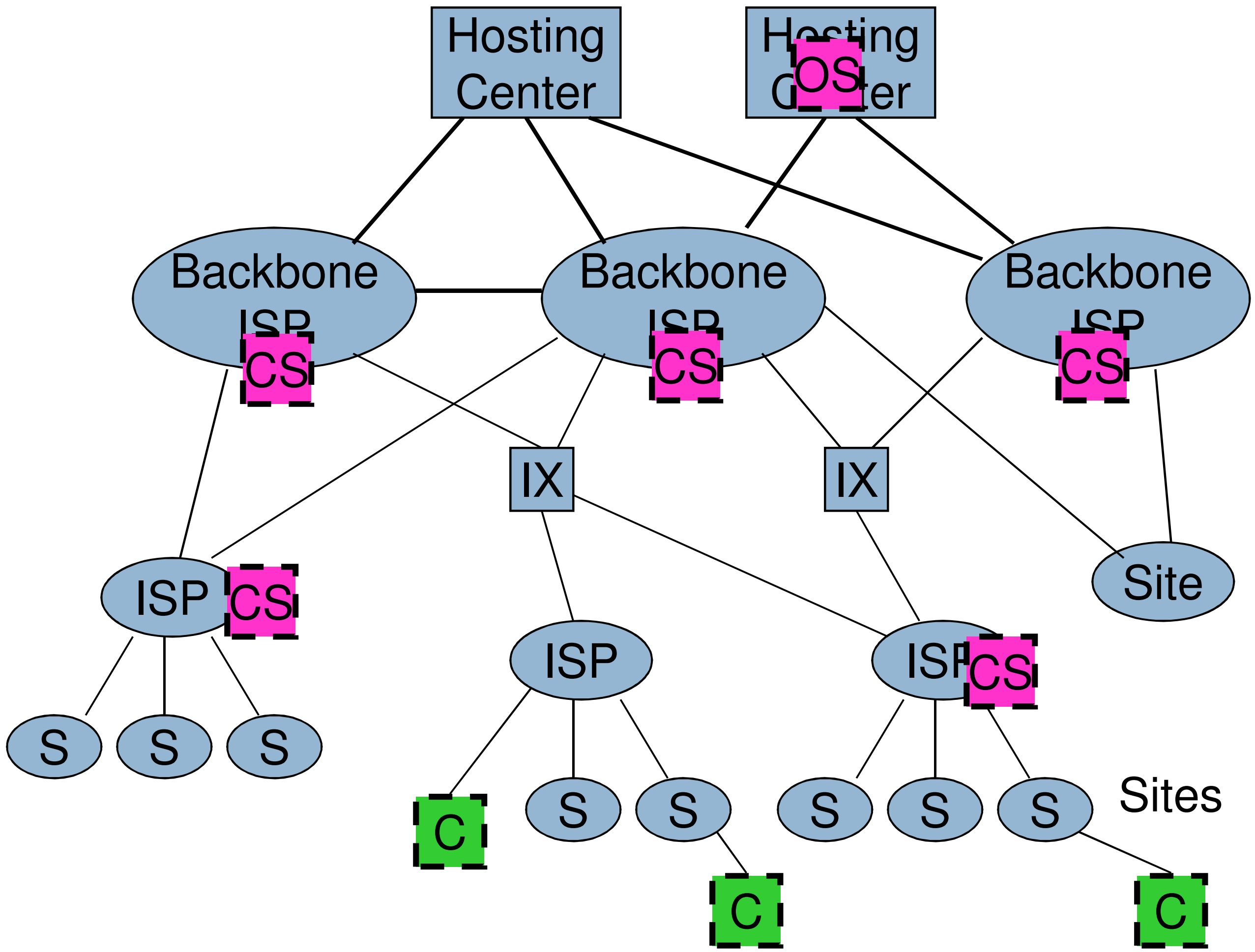
Other content routing mechanisms

- Dynamic .smil file modification
  - .smil used for multi-media applications (Synchronized Multimedia Integration Language)
    - Contains URLs pointing to media
  - Different tradeoffs from HTML URL re-writing
    - Proximity not as important
    - DNS lookup amortized over larger downloads
  - Also works for Real (.rm), Apple QuickTime (.qt), and Windows Media (.asf) descriptor files

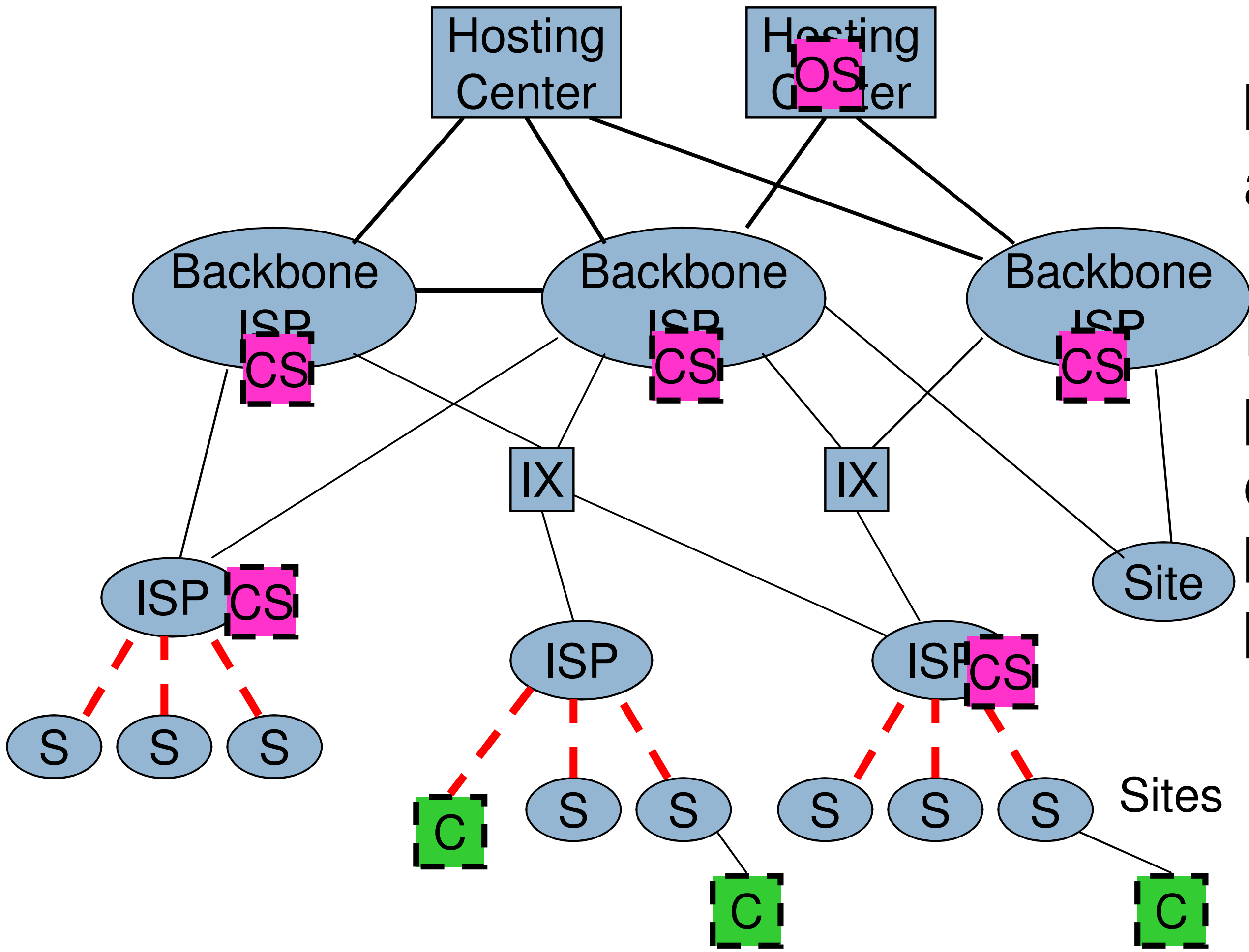
Other content routing mechanisms

- HTTP 302 Redirect
  - Directs client to another (closer, load balanced) server
  - For instance, redirect image requests to distributed server, but handle dynamic home page from origin server
- See *draft-cain-known-request-routing-00.txt* for good description of these issues
  - But expired, so use Google to find archived copy

How well do CDNs work?



How well do CDNs work?



Recall that the bottleneck links are at the edges.

Even if CSs are pushed towards the edge, they are still behind the bottleneck link!

Elements of a web page Download

- DNS round trip
- TCP handshake (2 round trips)
- Slow-start
  - ~8 round trips to fill DSL pipe
  - total 128K bytes
    - Compare to 56 Kbytes for cnn.com home page
    - Download finished before slow-start completes
- Total 11 round trips
- Example: US coast-to-coast propagation delay is ~15 ms
- 30 ms reduction in RTT means 330 ms total reduction
  - Certainly noticeable
  - So if we can reduce latency we can improve tcp (and thus http) performance

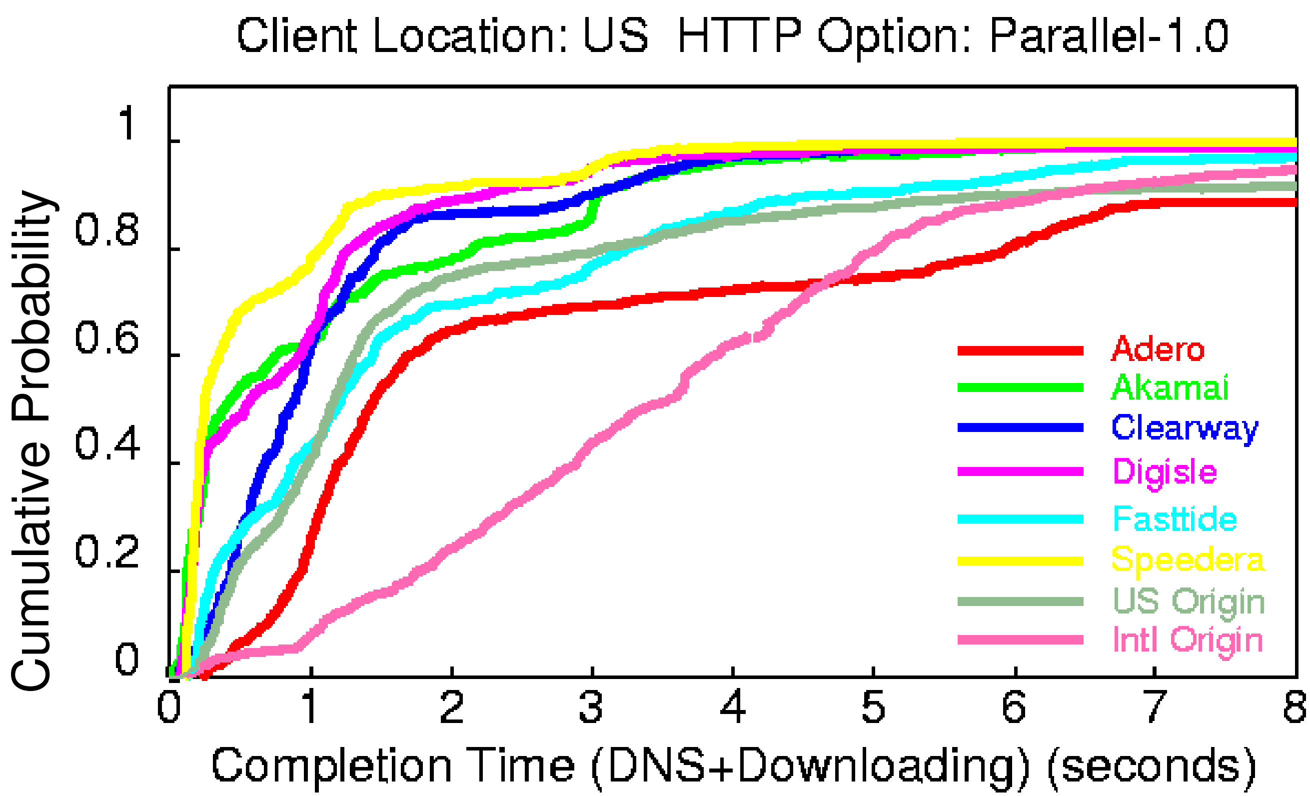
AT&T Study

- Zhang, Krishnamurthy and Wills
  - AT&T Labs
- Traces taken in Sept. 2000 and Jan. 2001
- Compared CDNs with each other
- Compared CDNs against non-CDN sites

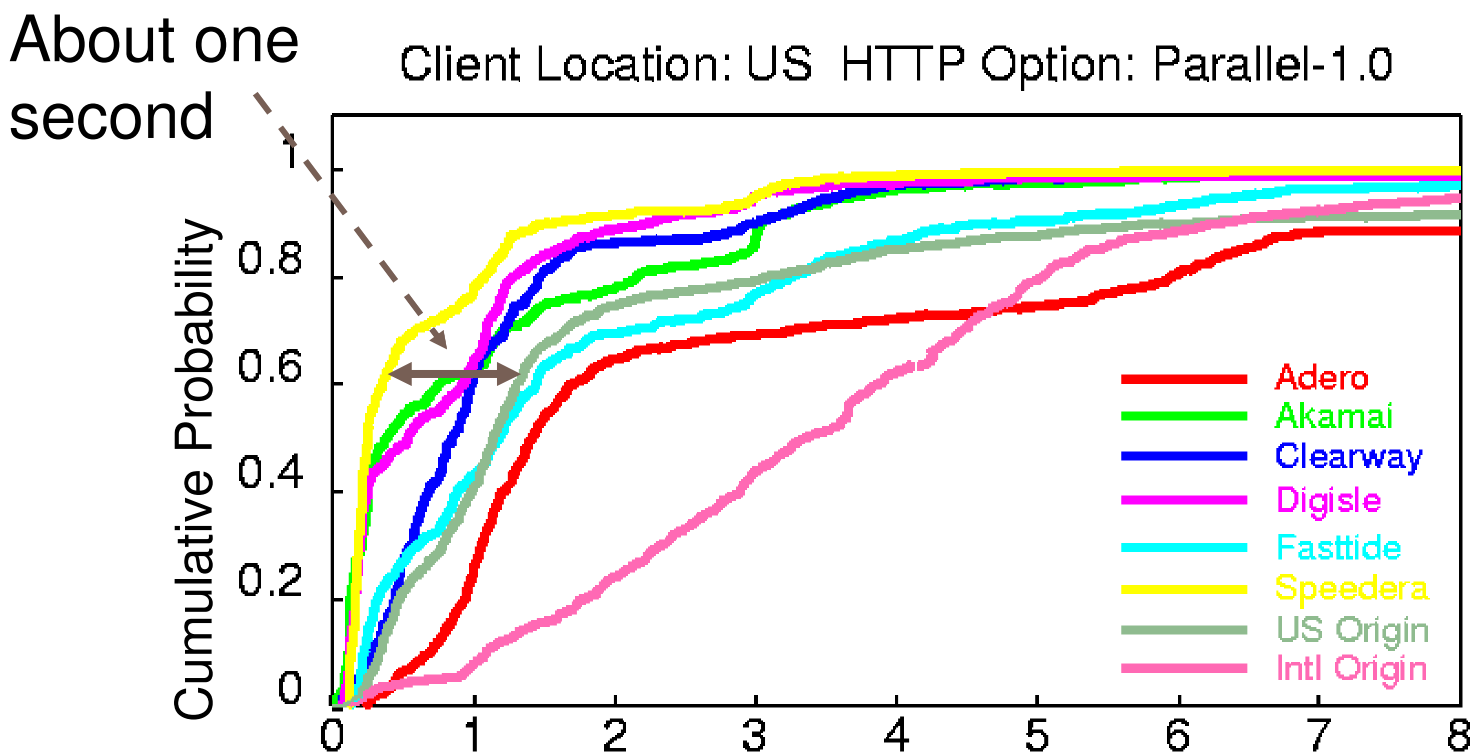
Methodology

- Selected a bunch of CDNs
  - Akamai, Speedera, Digital Island
    - Note, most of these gone now!
- Selected a number of non-CDN sites for which good performance could be expected
  - U.S. and international origin
  - U.S.: Amazon, Bloomberg, CNN, ESPN, MTV, NASA, Playboy, Sony, Yahoo
- Selected a set of images of comparable size for each CDN and non-CDN site
  - Compare apples to apples
- Downloaded images from 24 NIMI machines
  - Widely deployed measurement "platforms"

Response Time Results Including DNS Lookup Time



Response Time Results
Including DNS Lookup Time

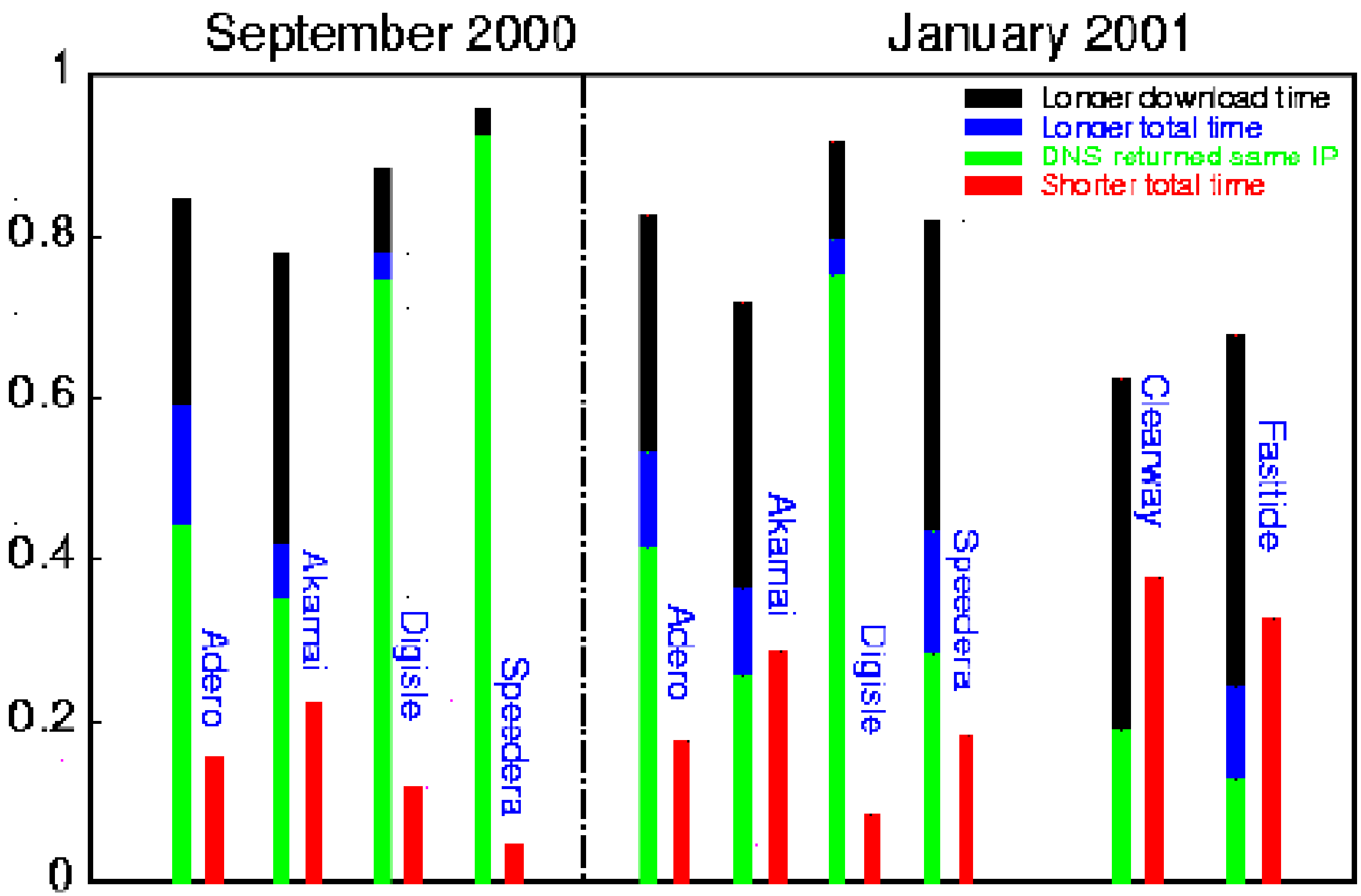


Author conclusion: CDNs generally provide much shorter download time.

CDNs out-performed non-CDNs

- Why is this?
- Consider the ability to pick good content servers using DNS redirection
  - Recall: short DNS TTLs give CDNs more control over which servers clients can use
- They compared time to download with a fixed IP address versus the IP address dynamically selected by the CDN for each download

Effectiveness of DNS load balancing



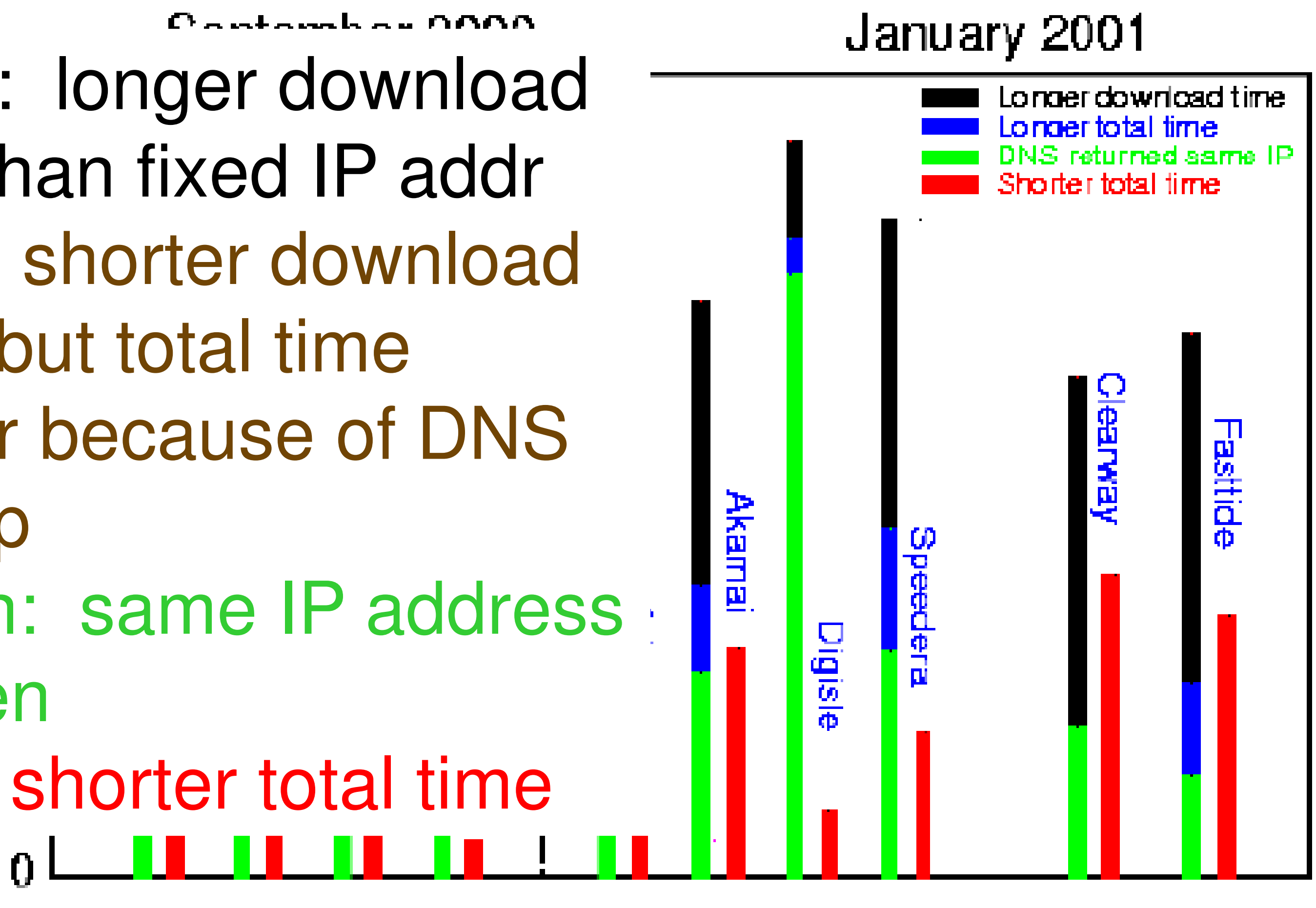
Effectiveness of DNS load balancing

Black: longer download time than fixed IP addr

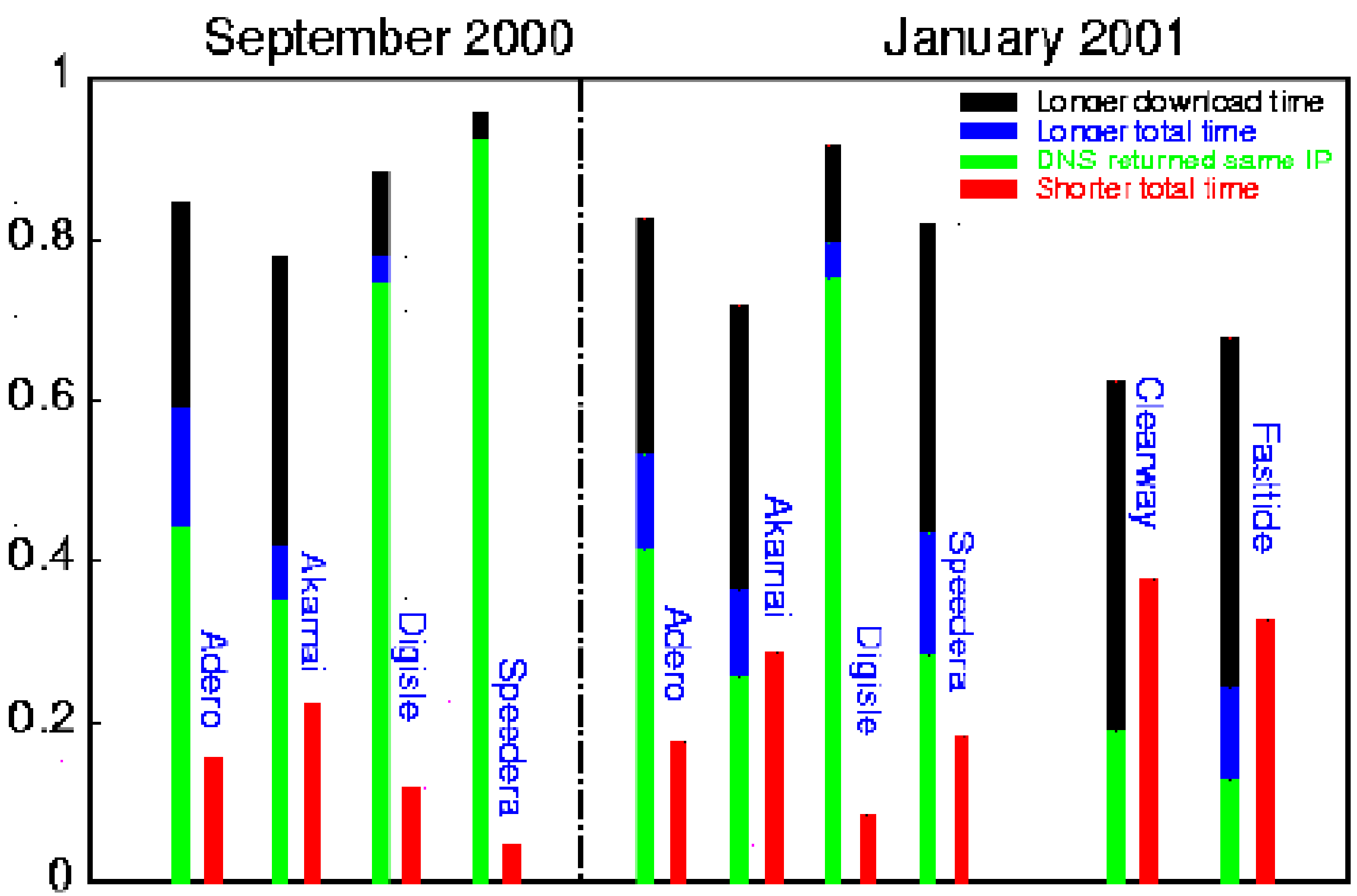
Blue: shorter download time, but total time longer because of DNS lookup

Green: same IP address chosen

Red: shorter total time



DNS load balancing not very effective



Other findings of study

- Each CDN performed best for at least one (NIMI) client
  - Why? Because of proximity?
- The best origin sites were better than the worst CDNs
- CDNs with more servers don't necessarily perform better
  - Note that they don't know load on servers…
- HTTP 1.1 improvements (parallel download, pipelined download) help a lot
  - Even more so for origin (non-CDN) cases
  - Note not all origin sites implement pipelining

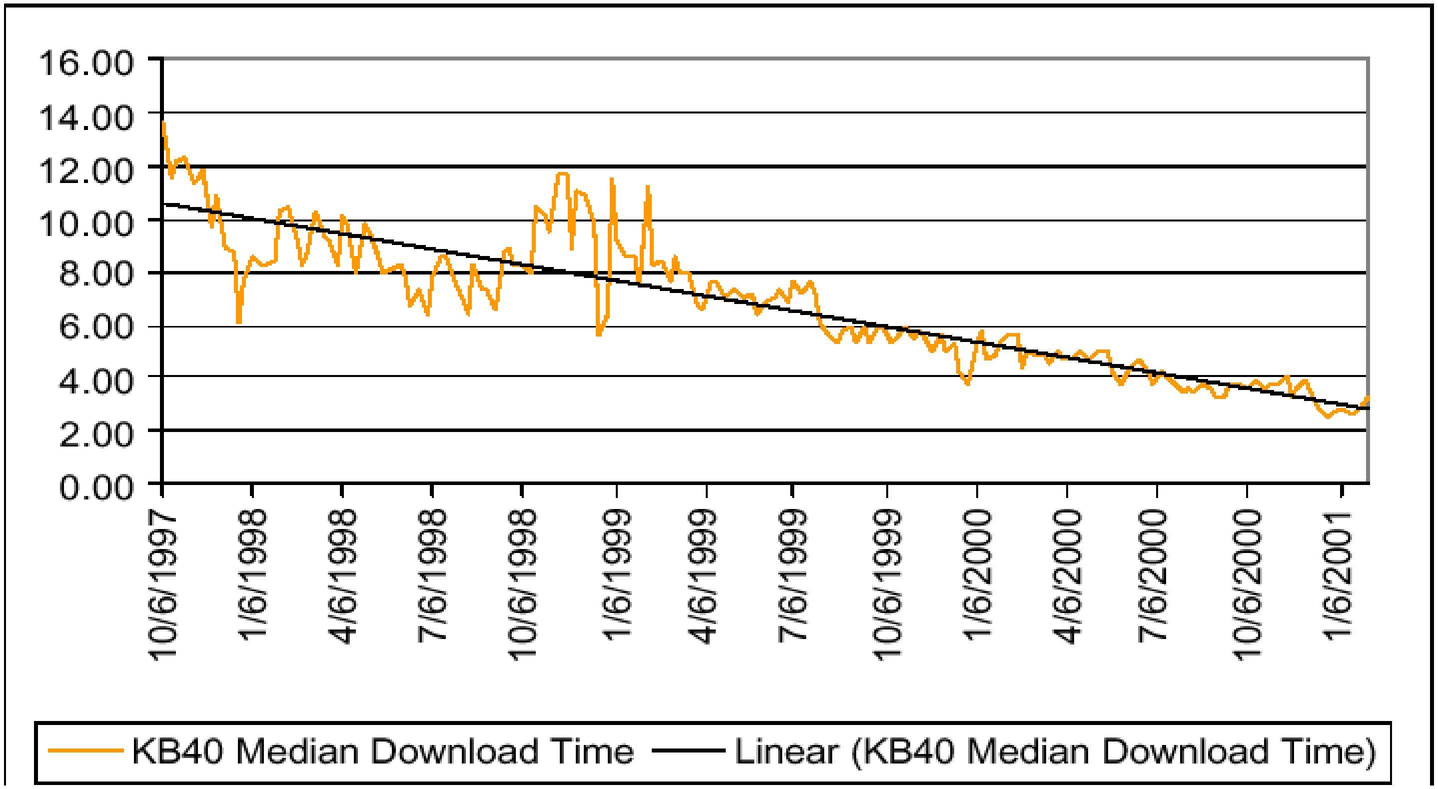
Ultimately a frustrating study

- Never actually says *why* CDNs perform better, only that they do
- For all we know, maybe it is because CDNs threw more money at the problem
  - More server capacity and bandwidth relative to load

Another study

- Keynote Systems
  - “A Performance Analysis of 40 e-Business Web Sites”
- Doing measurements since 1997
  - (All from one location, near as one can tell)
- Latest (publically available) measurement January 2001

Historical trend: Clear improvement



Performance breakdown

Basically says that smaller content leads to shorter download times (duh!)

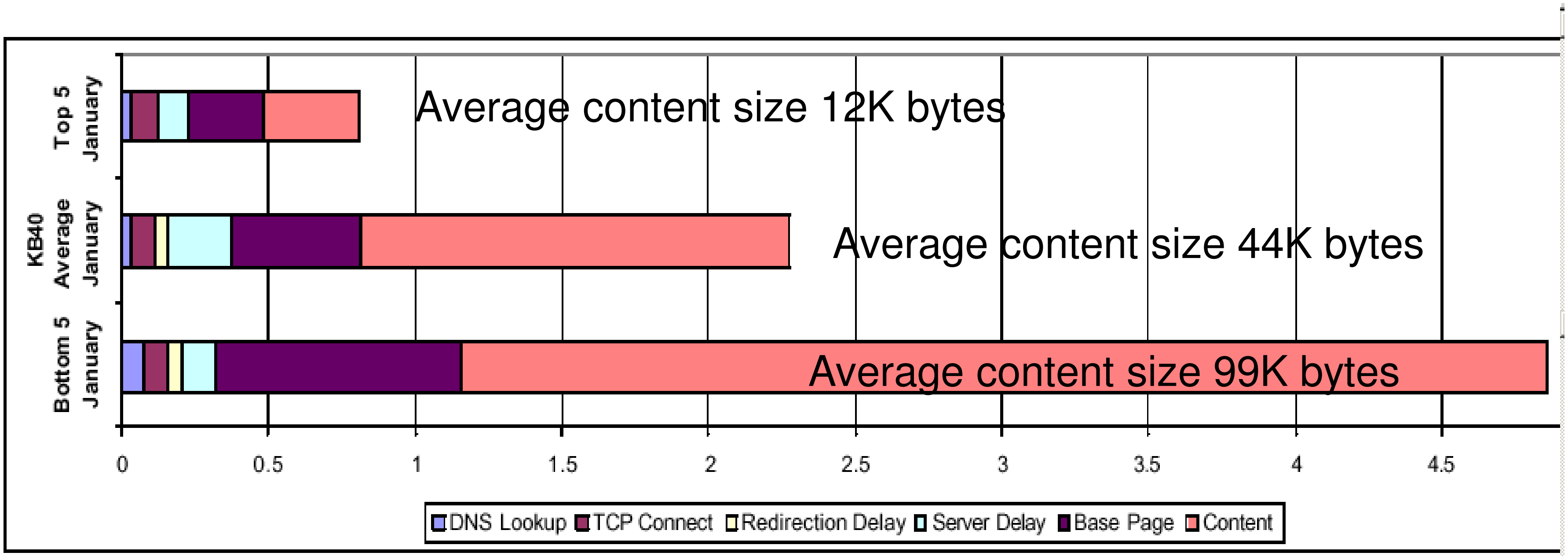
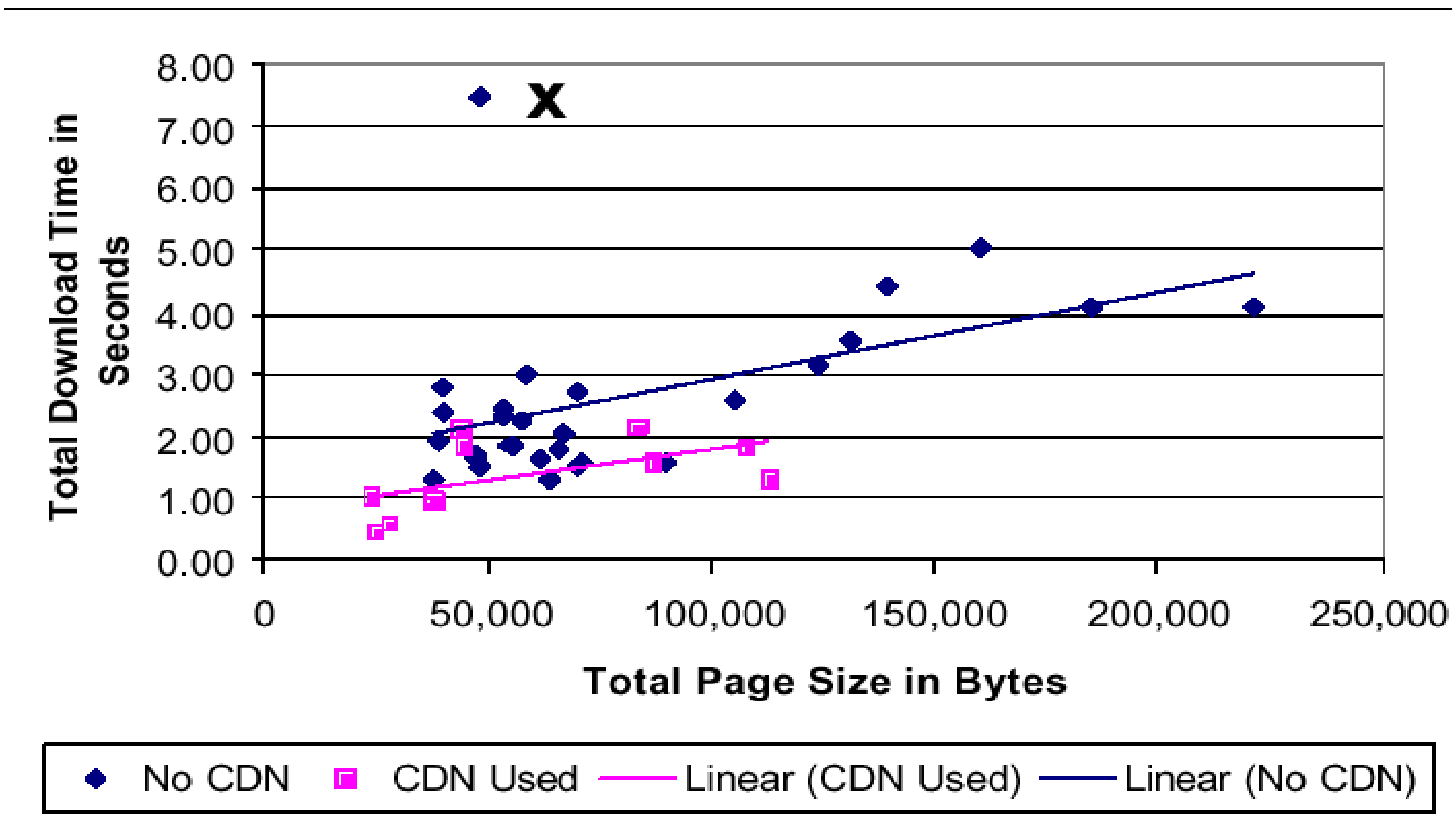
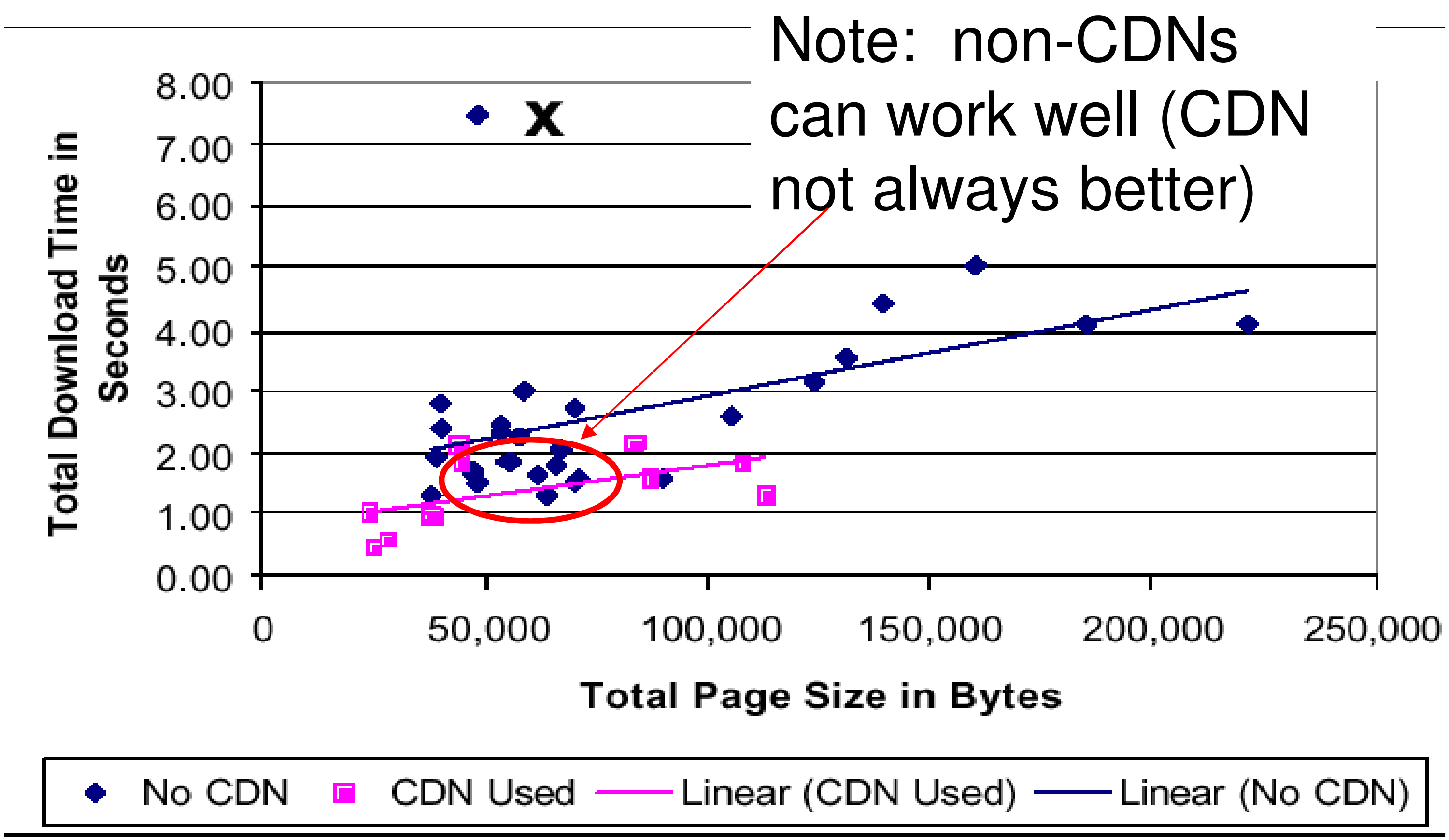


Figure 3. Download Time Components for Top 5, Average Site, and Bottom 5 in January 2001

Effect of CDN: Positive
(but again, we don't know why)



Most web sites not using CDN (4-1)



Summary

- CDNs still used and still performing well
  - On a par or better than best non-CDN web sites
- CDNs not very different from each other
- We don't know why CDNs perform well
  - But could very well simply be server capacity
- Knowledge of client location valuable more for customized advertising than for latency
  - Advertisements in right language

Layered Naming

- Recent proposal for discovery: naming requires four distinct layers:
  1. User-level descriptor (ULD) lookup (e.g. search string, email address, etc)
  2. Service-ID descriptor (SID): a sort of index naming the service and valid over the duration of this interaction
  3. SID to Endpoint-ID (EID) mapping: client-side protocol (e.g. HTTP) maps from SID to EID
  4. EID to IP address "routing": server side control over the decision of which "delegate" will handle the request
- Today we tend to blur the middle two layers and lack standards for this process, forcing developers to innovate
- See: "A Layered Naming Architecture for the Internet", Balikrishnan *et. al.*, ACM SIGCOMM Aug. 2004, Portland.

Research challenges

- Naming and discovery are examples of research challenges we're now facing in the Web Services arena
- Still unchartered territory

Homework (not to hand in)

- Continue to read Parts I and II of the book
- Visit the semantic web repository at www.w3.org
- What does that community consider to be a potential "home run" for the semantic web?

WS & RPC Connectivity Issues: Network Address Translation

- IP Address – 32 bits only.
  - Address Space Shortage.
- NATs invented to overcome this problem.
- Have a NAT box in between a private network and the internet.
- Can use locally allocated addresses within private network.
- The NAT router maps the internal IP address:port to the external IP address:port and vice-versa.