Class Meeting: Lectures 15 and 16
HTTP and the Web, Content Distribution Networks

Kyle Jamieson

COS 461: *Computer Networks*

[Selected content adapted from M. Freedman, B. Maggs and S. Shenker]

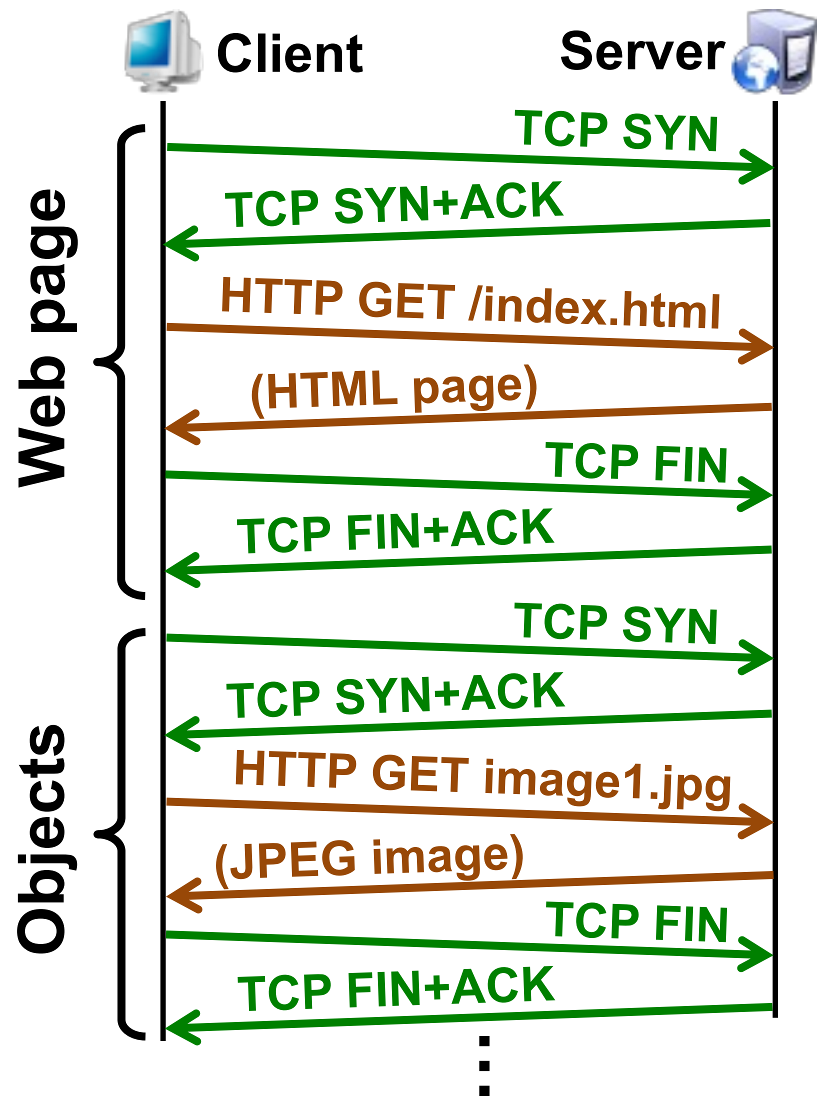
Today

1. The Web: HTTP, hosting, and caching

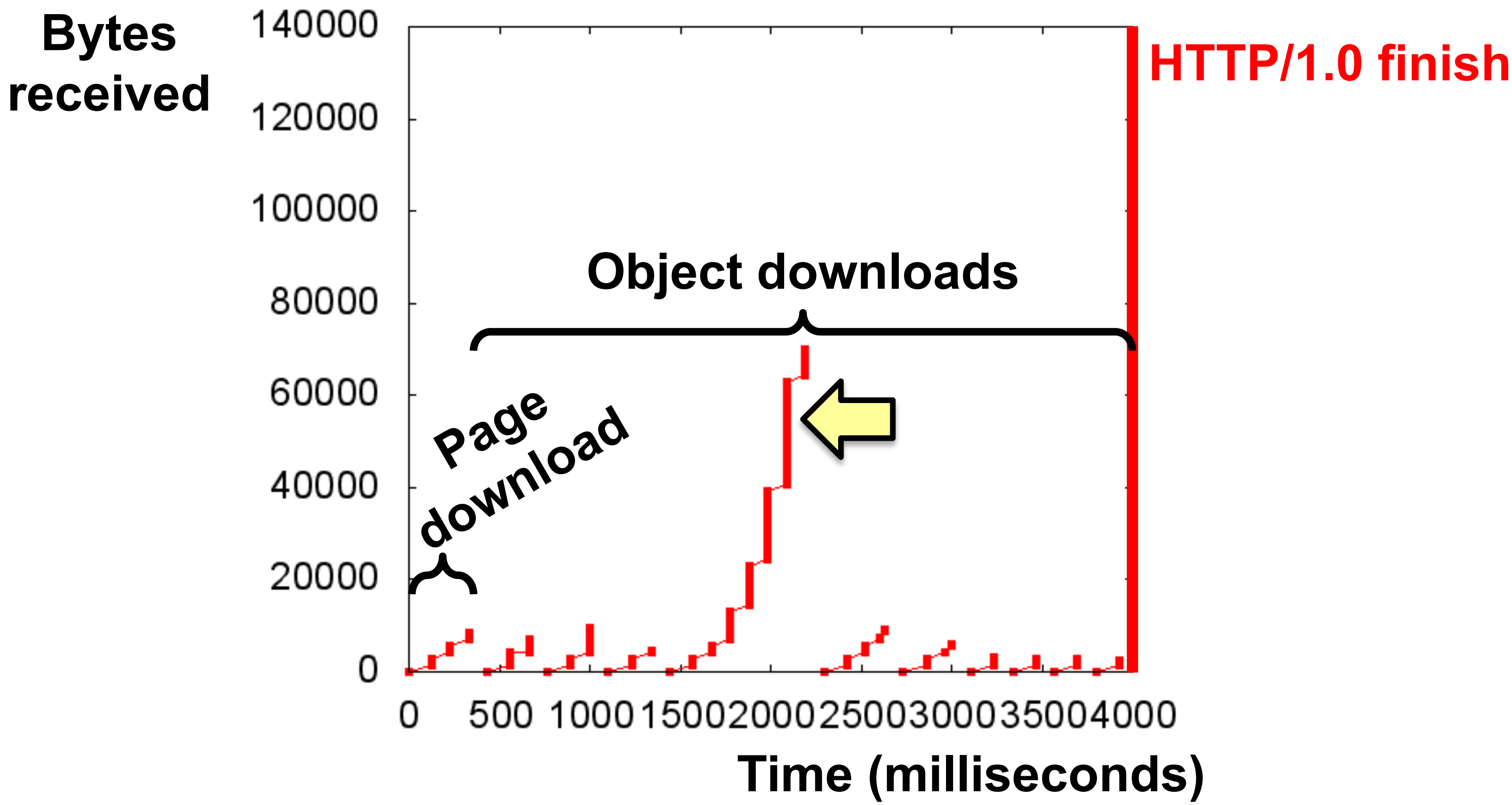
2. Content distribution networks (CDNs)

Anatomy of an HTTP/1.0 web page fetch

- Web page = HTML file + embedded images/objects
- *Stop-and-wait* at the granularity of objects:
  - Close then open new TCP connection for **each object**
    - Incurs a TCP RTT delay each time
  - Each TCP connection may stay in "slow start"



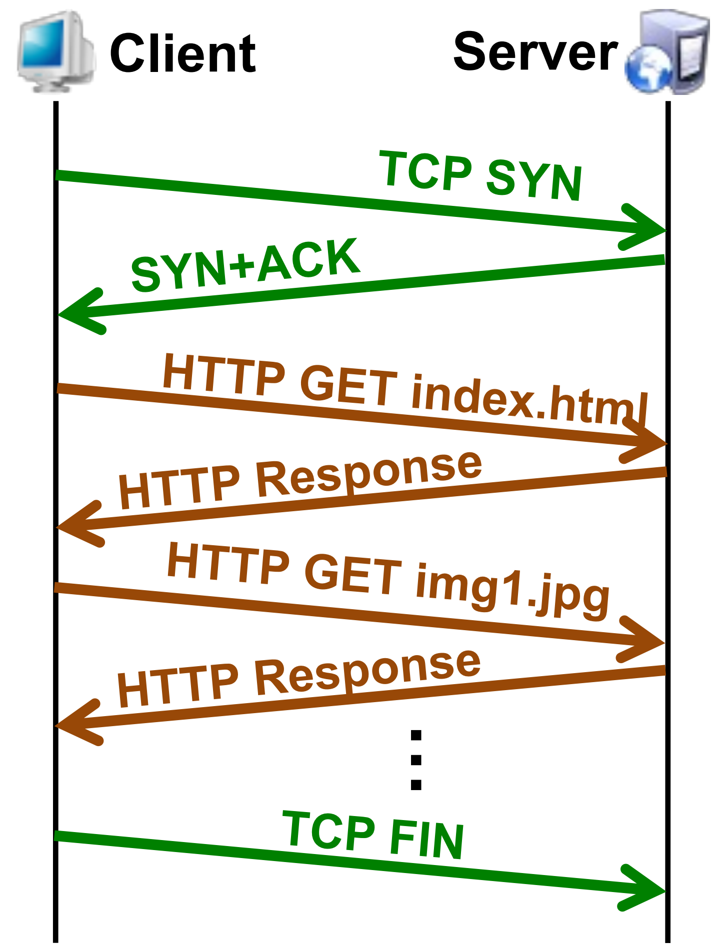
HTTP/1.0 webpage fetch: Timeline



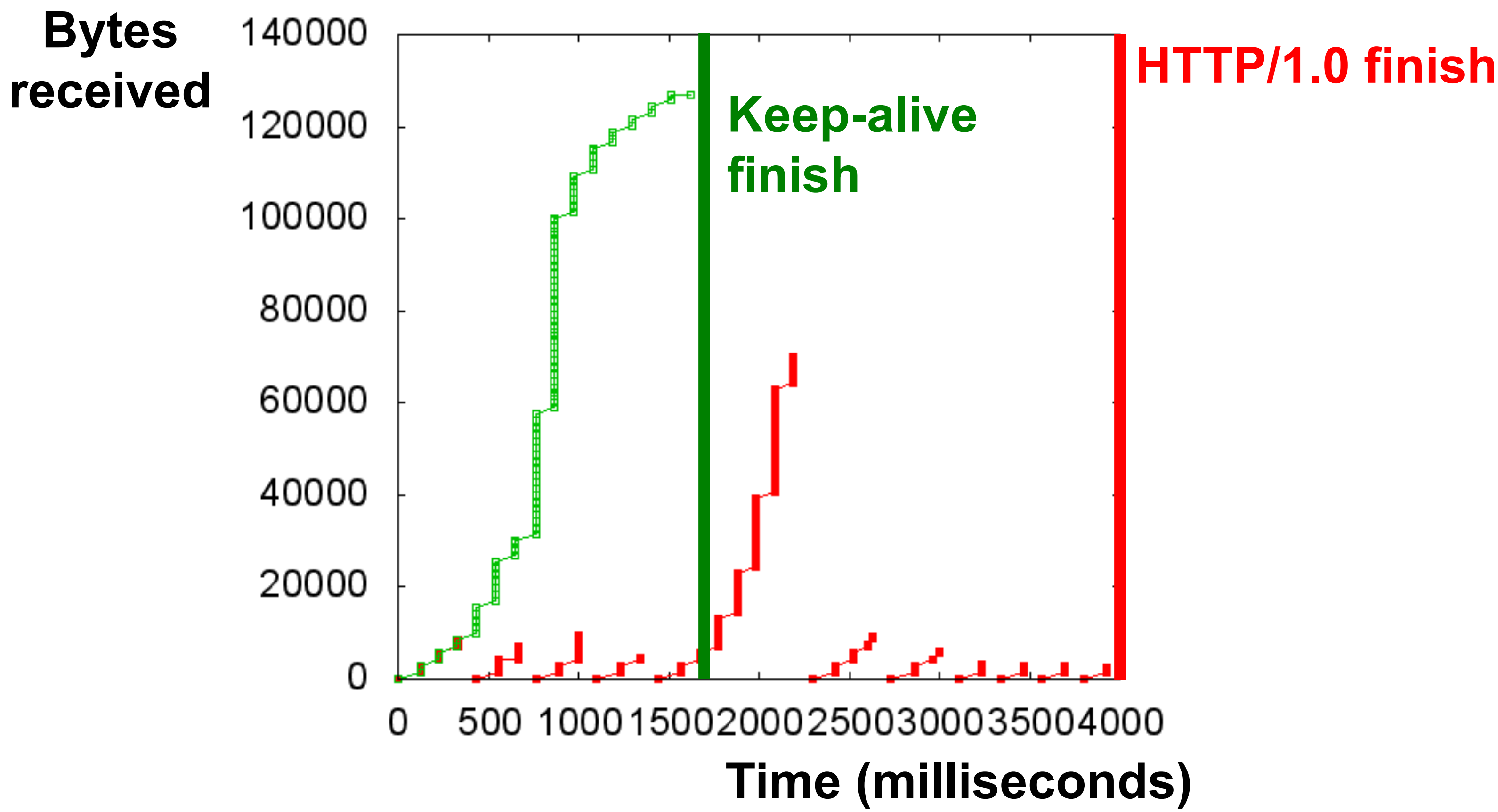
- Fetch 8.5 Kbyte page with 10 objects, most < 10 Kbyte

Letting the TCP connection persist

- Known as *HTTP keepalive*
- **Still stop-and-wait** at the granularity of objects, at the application layer
  - HTTP response fully received before next HTTP GET dispatched
    - ≥ 1 RTT per object



HTTP Keepalive avoids TCP slow starts



Incur one slow start, but stop-and-wait to issue next request

Pipelining within HTTP

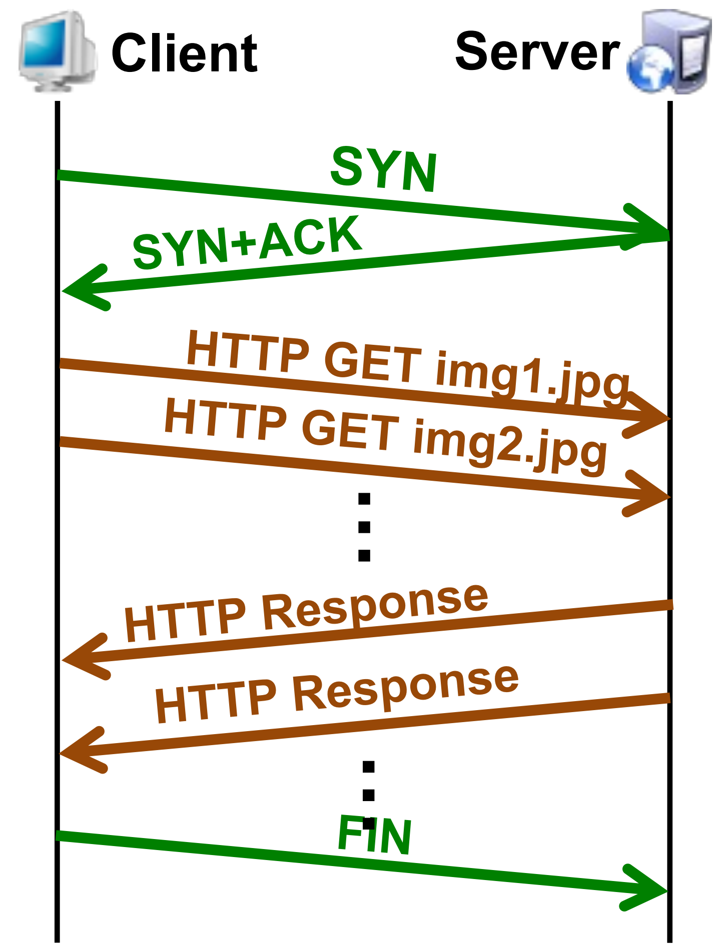
- Idea: Pipeline HTTP GETs and their responses

- Main benefits:

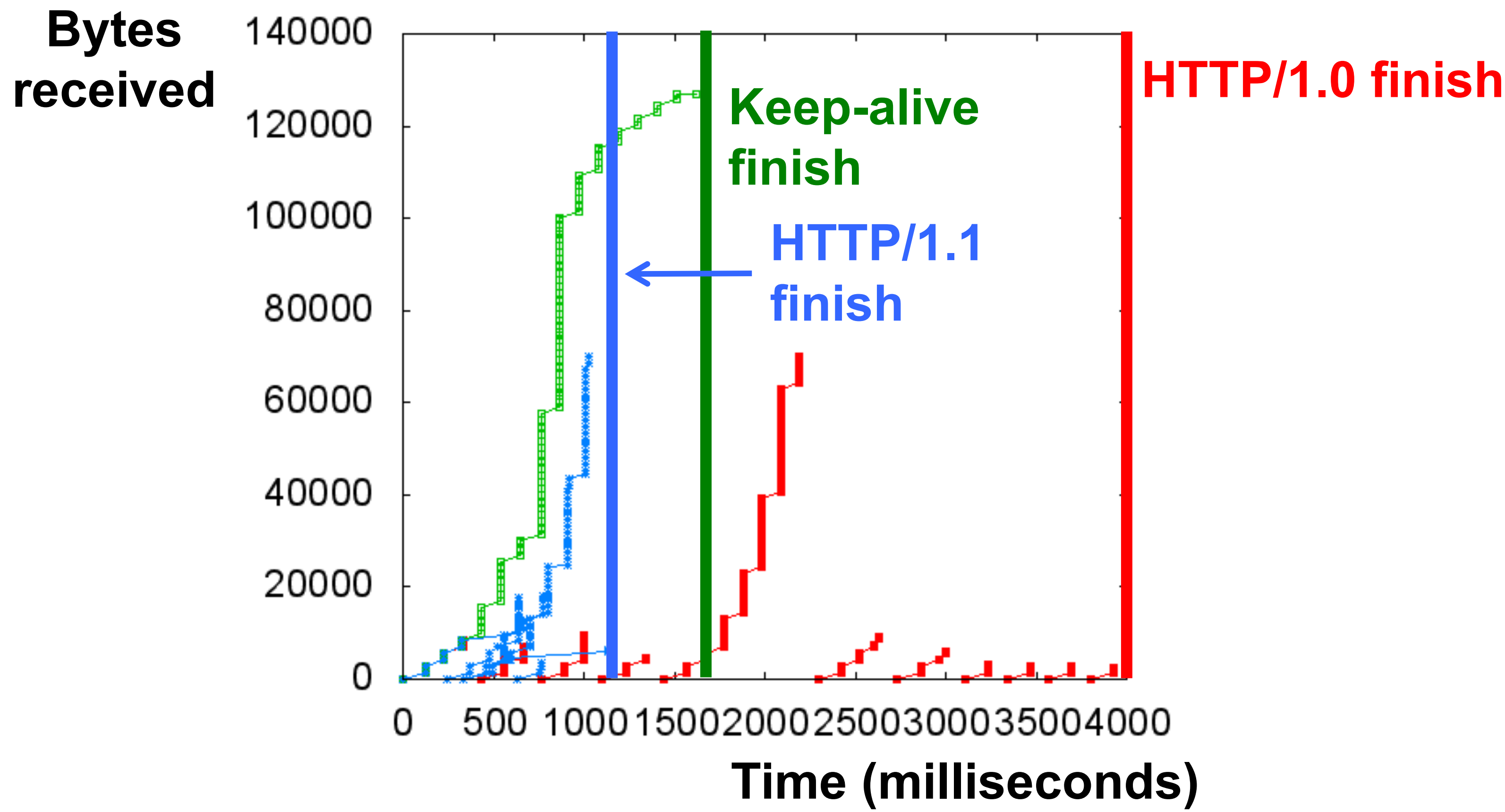
1. Amortizes the RTT across multiple objects retrieved

2. Reduces overhead of HTTP requests, packing multiple requests into one packet

- Implemented in HTTP/1.1



Pipelined HTTP requests overlap RTTs



- Many HTTP requests and TCP connections at once
- Overlaps RTTs of all requests

Today

1. The Web: HTTP, **hosting, and caching**
   - **Handling heavy loads**

2. Content distribution networks (CDNs)

Hosting: Multiple machines per site

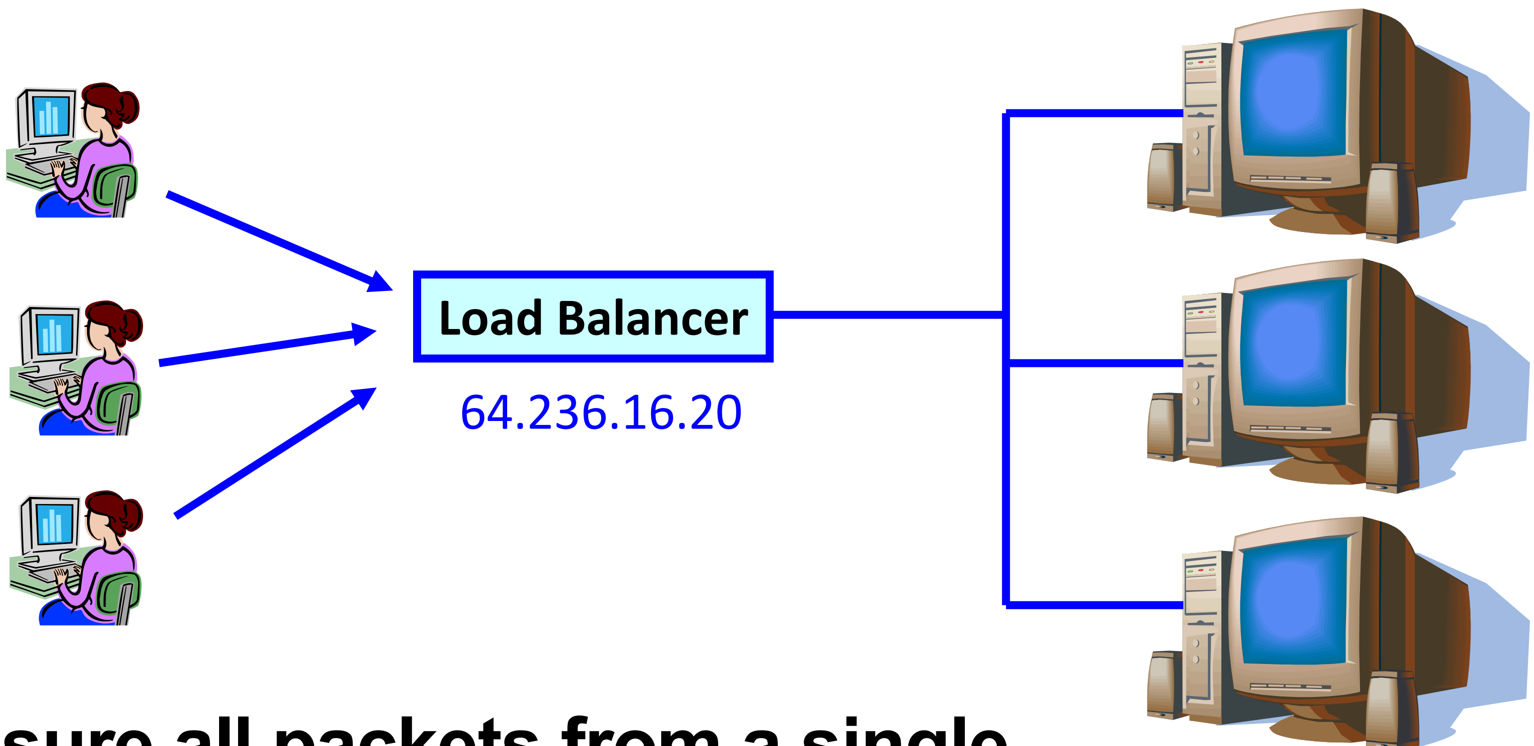
- **Problem:** Overloaded popular web site
  - Replicate the site across multiple machines
    - Helps to handle the load

- Want to direct client to a particular replica. Why?
  - Balance load across server replicas

- **Solution #1:** Manual selection by clients
  - Each replica has its own site name
  - Some Web page lists replicas (*e.g.*, by name, location), asks clients to click link to pick

Hosting: Load-balancer approach

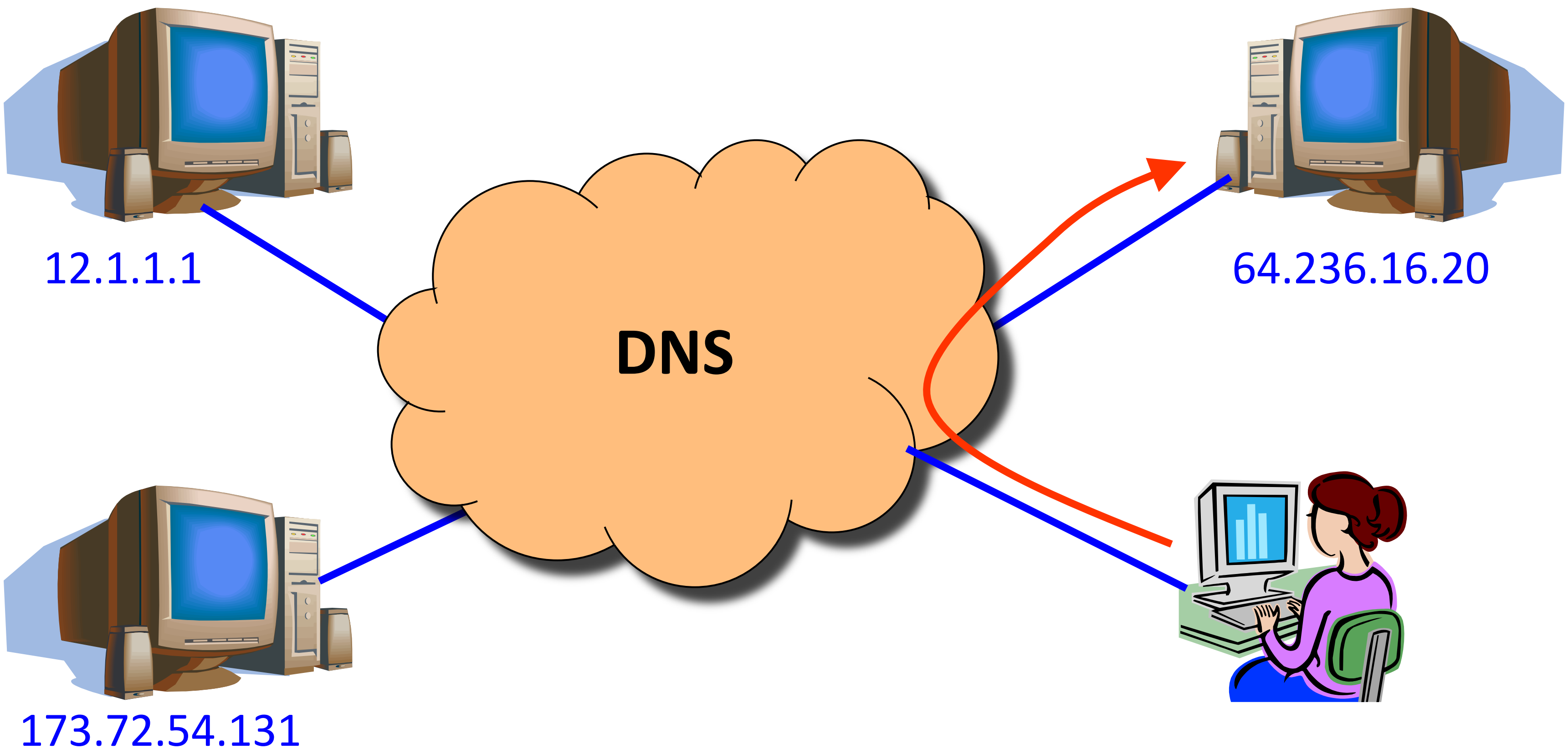
- **Solution #2:** Single IP address, multiple machines
  - Run multiple machines behind a single IP address



  - **Ensure all packets from a single TCP connection go to the same replica**

Hosting: DNS redirection approach

- **Solution #3:** Multiple IP addresses, multiple machines
  - Same DNS name but different IP for each replica
    - DNS server returns IP addresses "round robin"



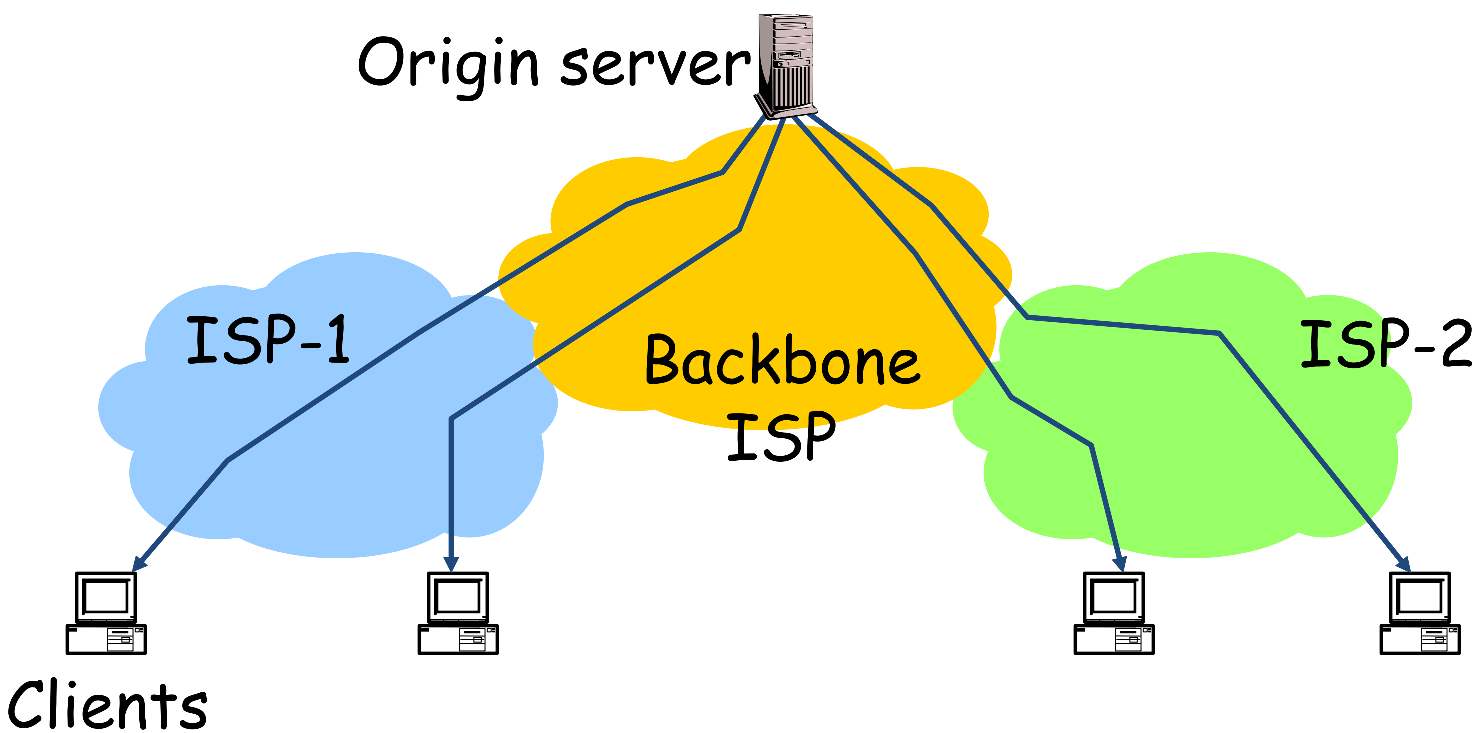
Hosting: Summary

- Load-balancer approach
  - No geographical diversity ✘
  - TCP connection issue ✘
  - Does not reduce network traffic ✘

- DNS redirection
  - No TCP connection issues ✔
  - Simple round-robin server selection
    - May be less responsive ✘
  - Does not reduce network traffic ✘

Web caching

- Many clients transfer the same information
  - Generates **redundant** server and network load
  - Also, clients may experience high **latency**



Why web caching?

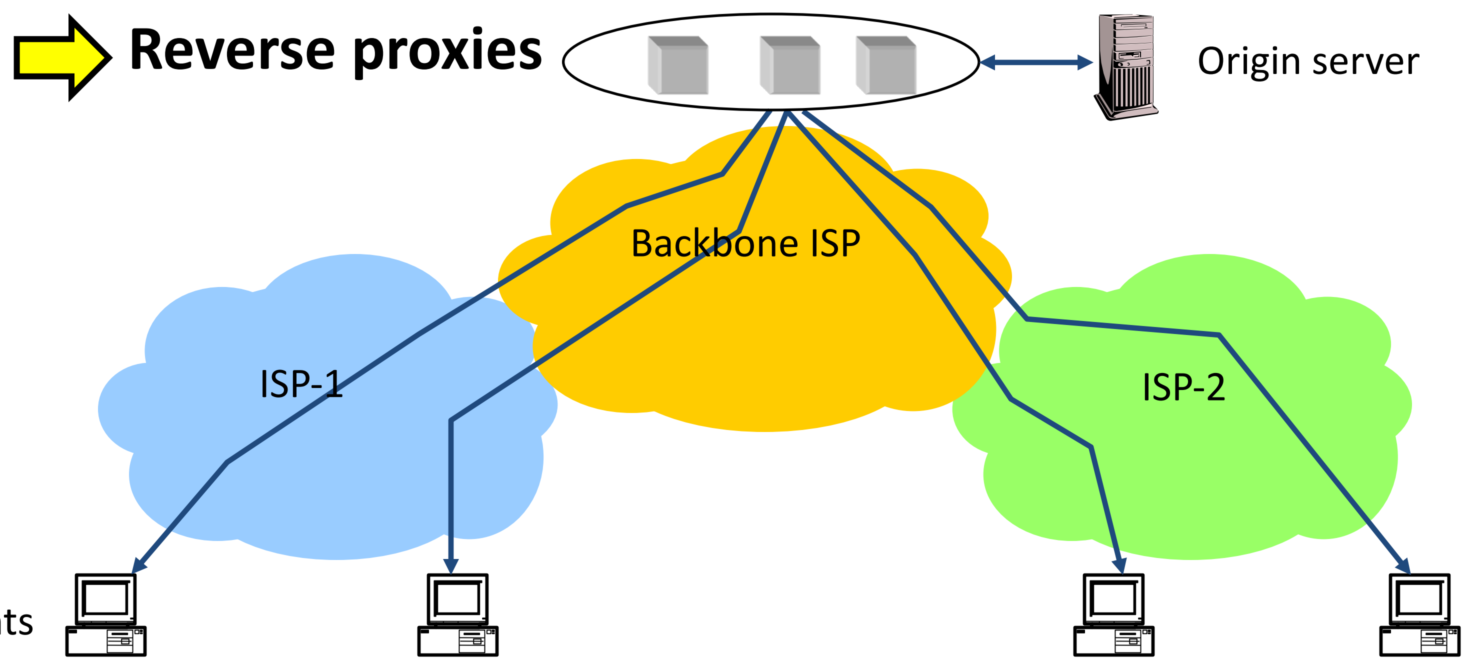
- Motivation for placing content closer to client:
  - User gets better response time
    - Content providers get happier users
  - Network gets reduced load

- Why does caching work? Exploits locality of reference

- How well does caching work?
  - Very well, up to a limit
  - Large overlap in content
  - But many unique requests

Caching with Reverse Proxies

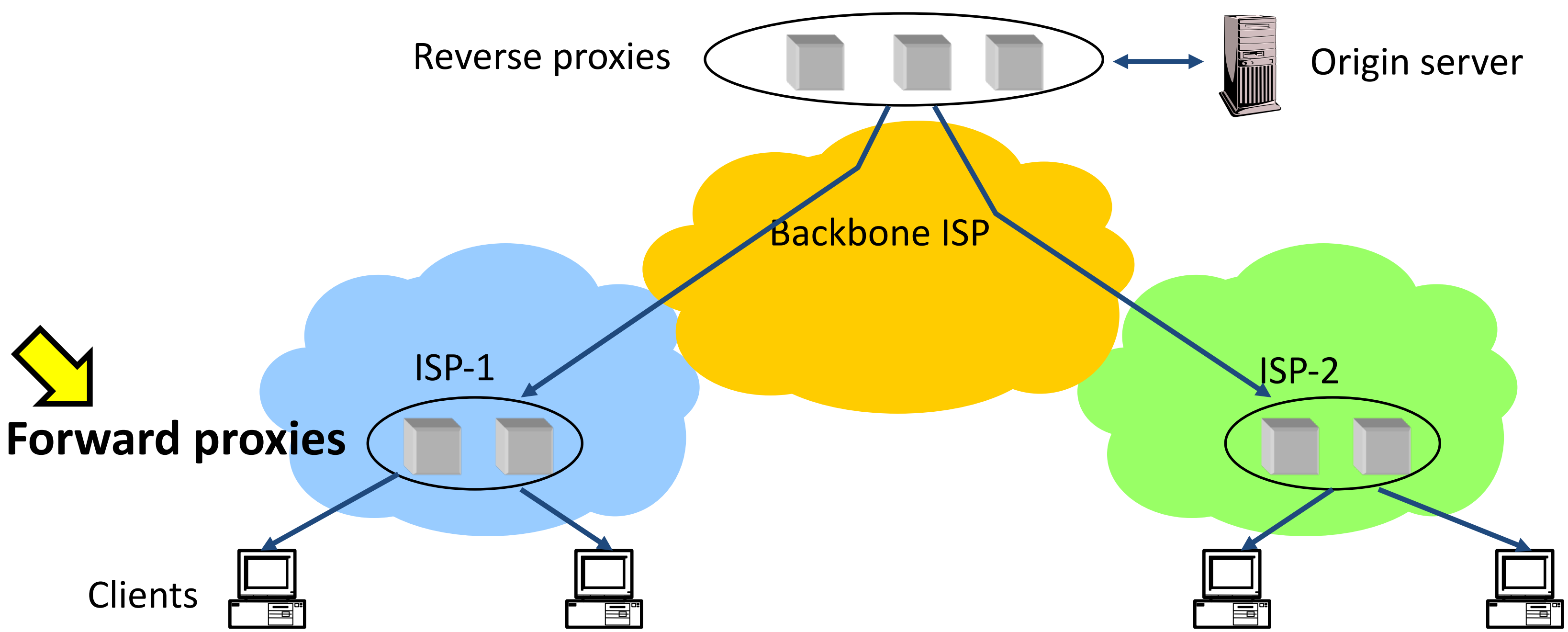
- Cache data close to origin server → decrease server load
  - Typically done by content providers
  - Client thinks it is talking to the origin server (the server with content)
- Does not work for **dynamic content**



Caching with Forward Proxies

- Cache close to clients → less network traffic, less latency
  - Typically done by ISPs or corporate LANs
  - **Client configured** to send HTTP requests to forward proxy

- Reduces traffic on ISP-1's access link, origin server, and backbone ISP



Caching & Load-Balancing: Outstanding problems

- Problem *ca.* 2002: *How to reliably deliver large amounts of content to users worldwide?*
  - Popular event: **"Flash crowds" overwhelm** (replicated) web server, access link, or back-end database infrastructure
  - More rich content: audio, video, photos
- Web caching: Diversity of content requests causes **low cache hit rates (25–40%)**

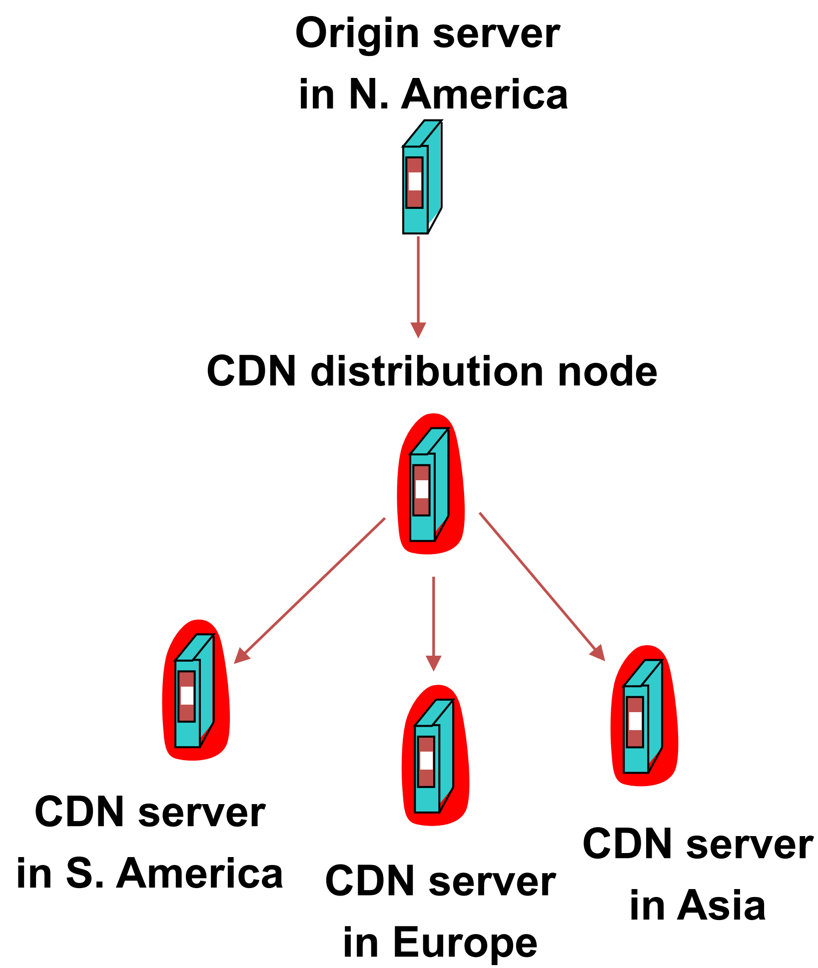
Today

1. The Web: HTTP, hosting, and caching

2. **Content distribution networks (CDNs)**
   – Akamai case study

Content Distribution Networks

- Proactive content replication
  - Content provider (*e.g.* CNN) pushes content out from its own *origin server*
- CDN **replicates** the content
  - On many servers spread throughout the Internet
- Updating the replicas
  - Updates **pushed to replicas** when the content changes



Replica selection: Goals

- Live server
  - For availability

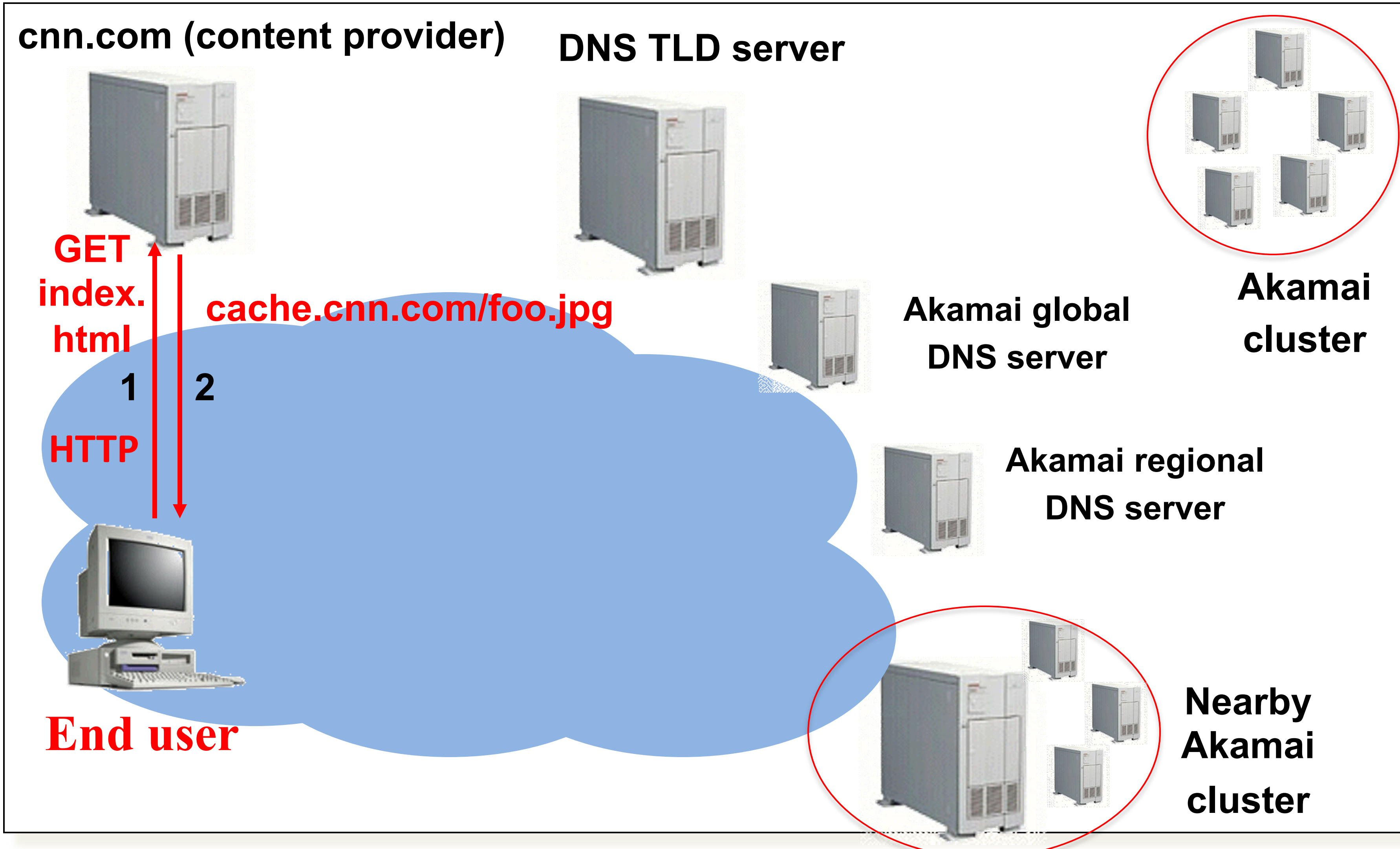
Requires continuous monitoring of liveness, load, and performance

- Lowest load
  - To balance load across the servers
- Closest
  - Nearest geographically, or in round-trip time
- Best performance
  - Throughput, latency, reliability…

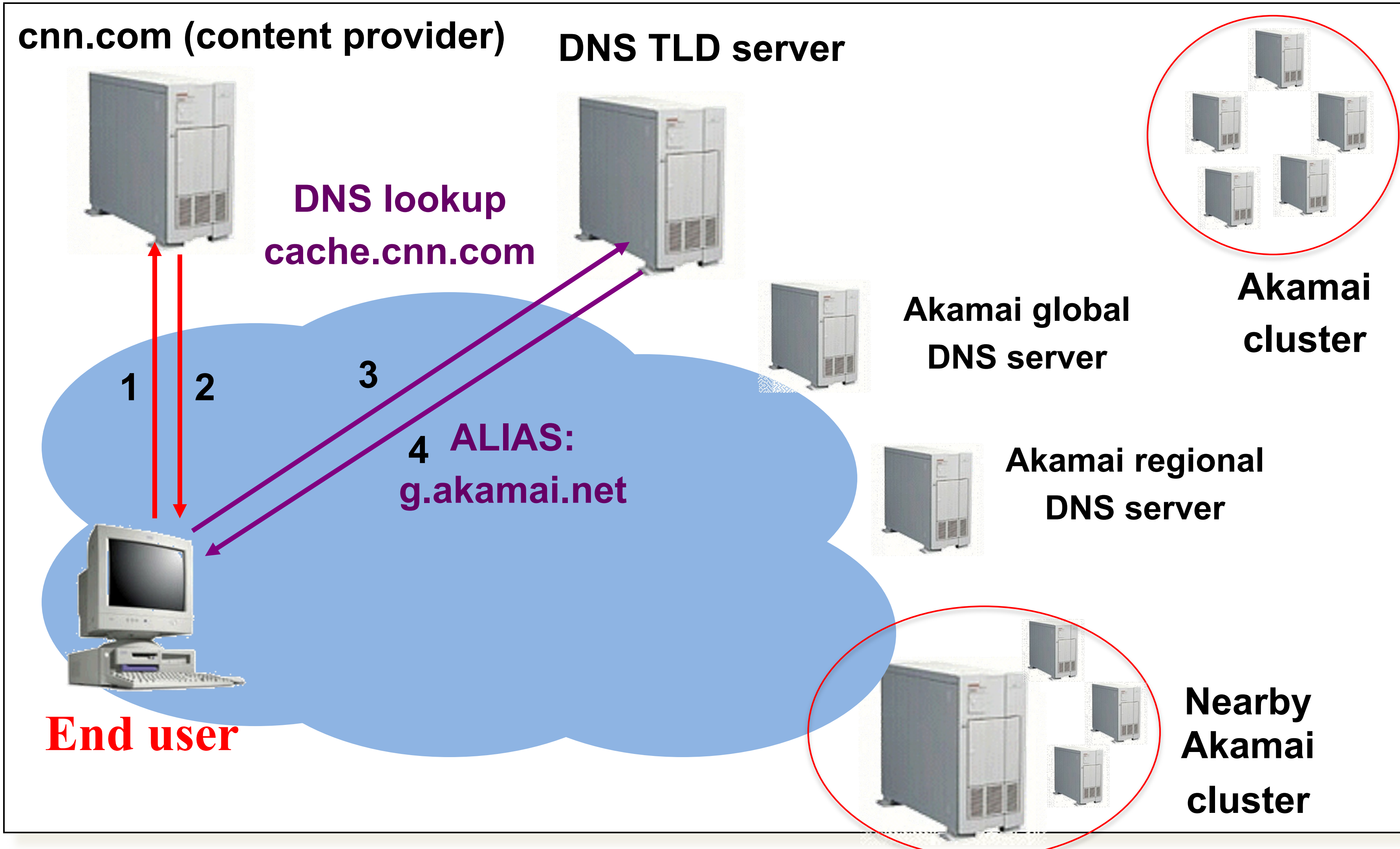
Akamai statistics

- Distributed servers
  - Servers: ~100,000
  - Networks: ~1,000
  - Countries: ~70
- Many customers
  - Apple, BBC, FOX, GM IBM, MTV, NASA, NBC, NFL, NPR, Puma, Red Bull, Rutgers, SAP, …
- Client requests
  - 20+M per second
  - Half in the top 45 networks
  - 20% of all Web traffic worldwide

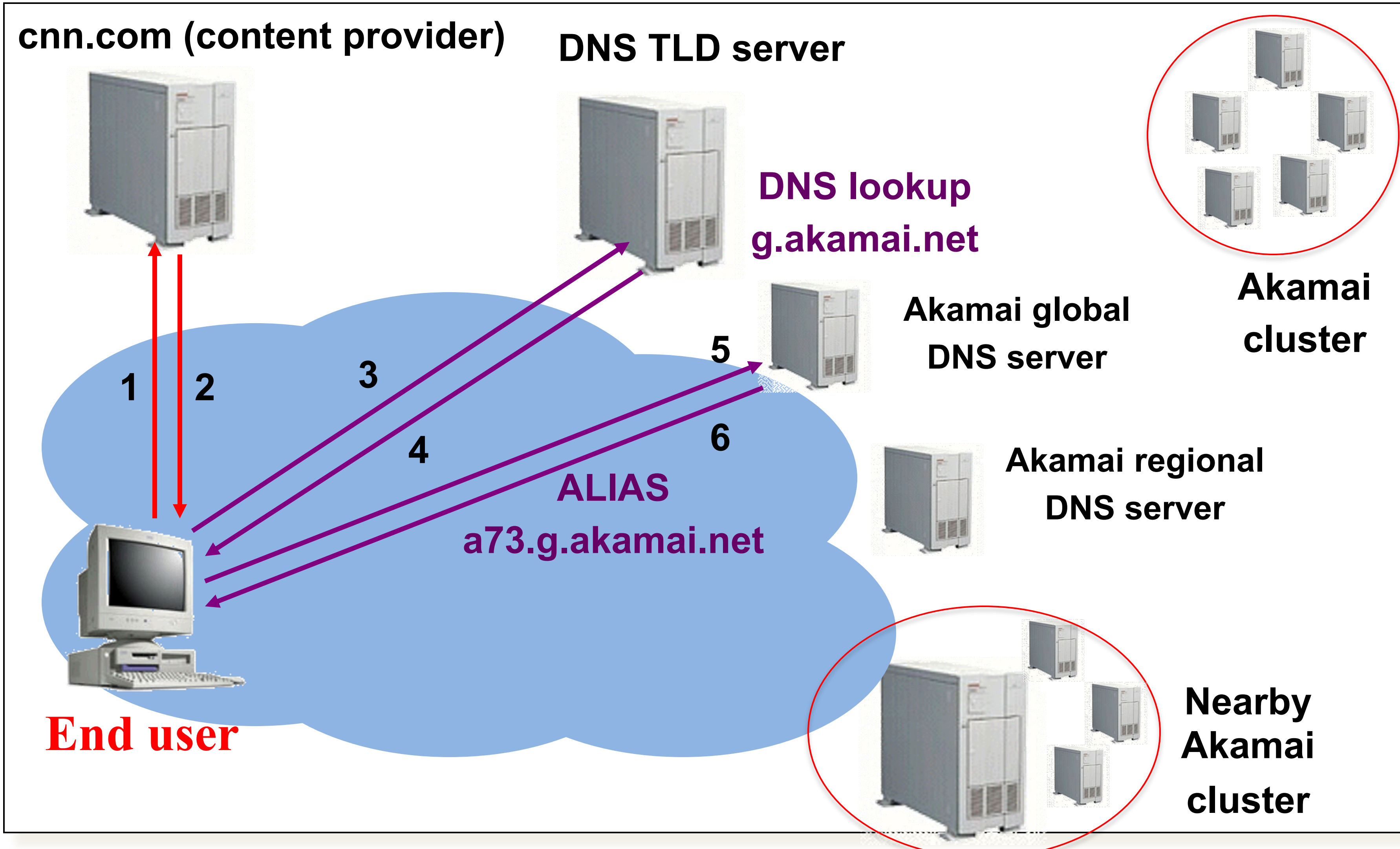
How Akamai Uses DNS



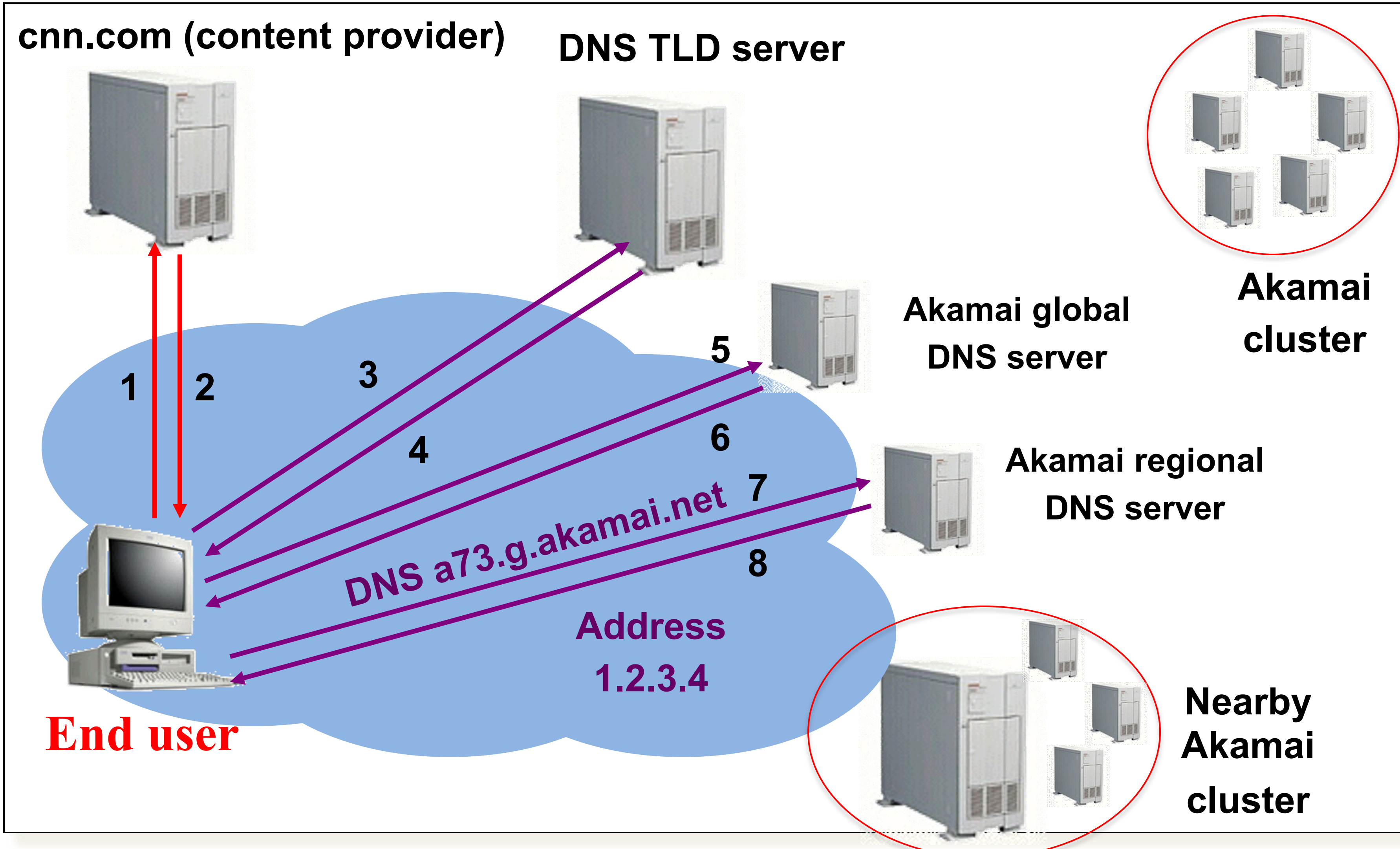
How Akamai Uses DNS



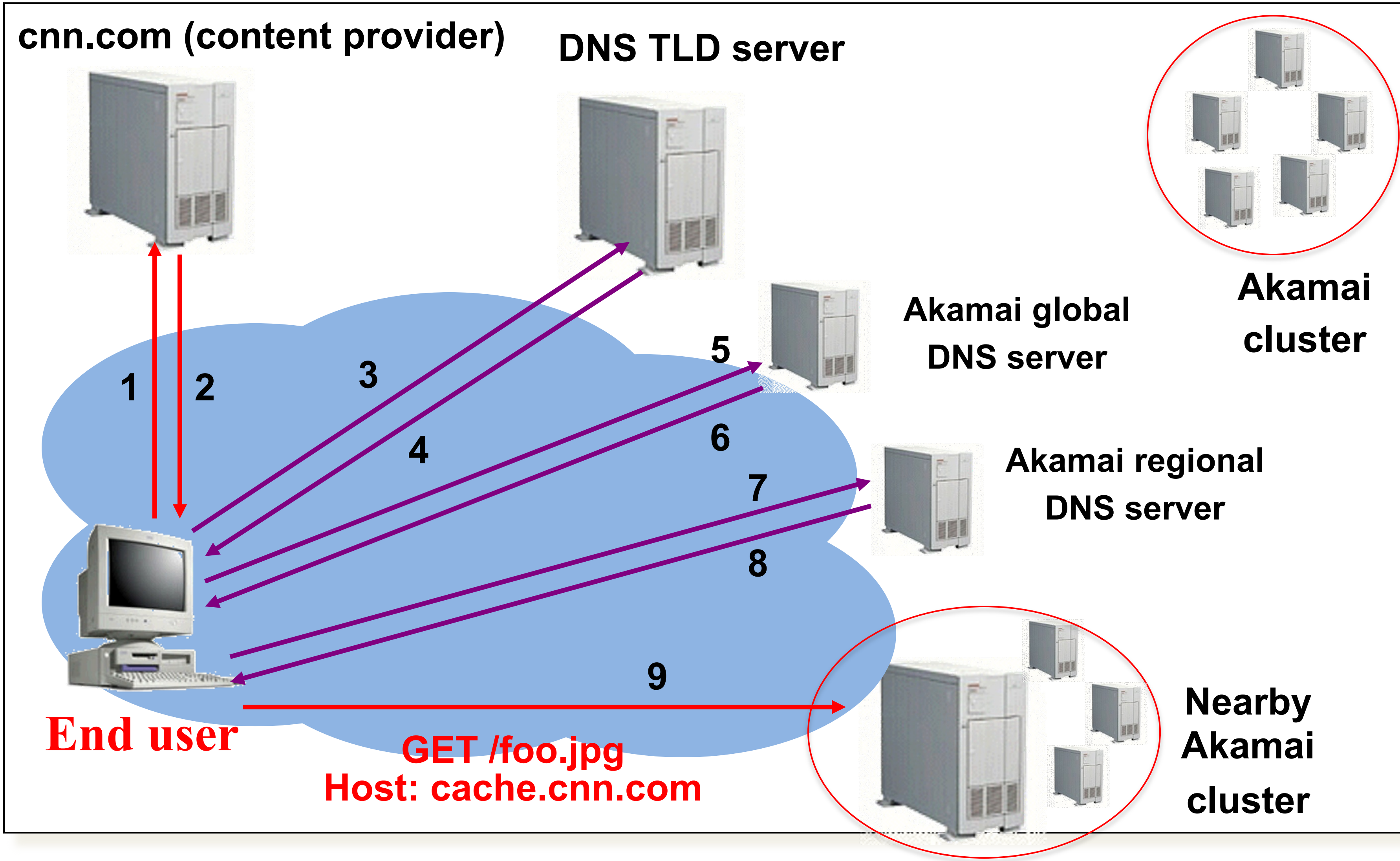
How Akamai Uses DNS



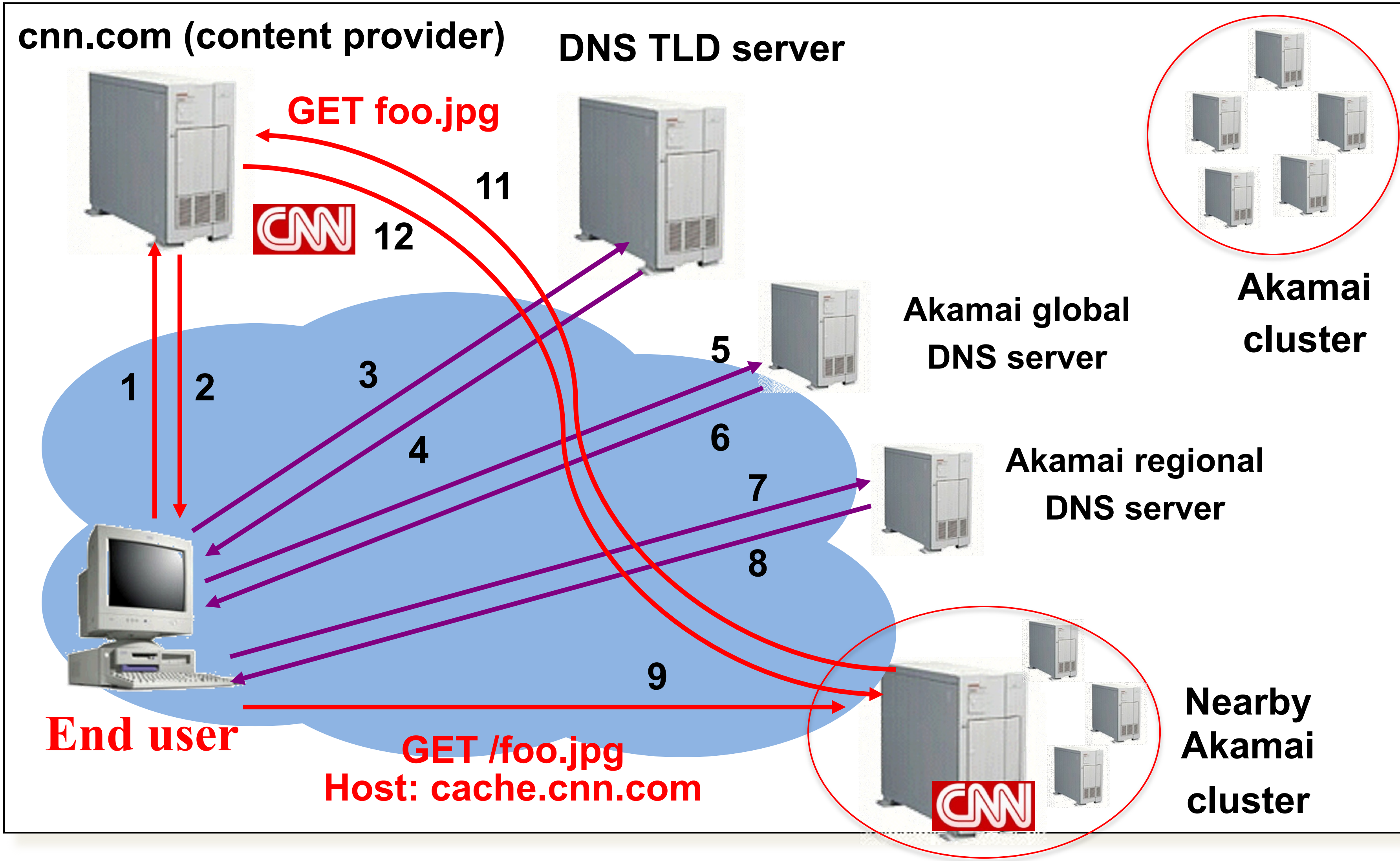
How Akamai Uses DNS



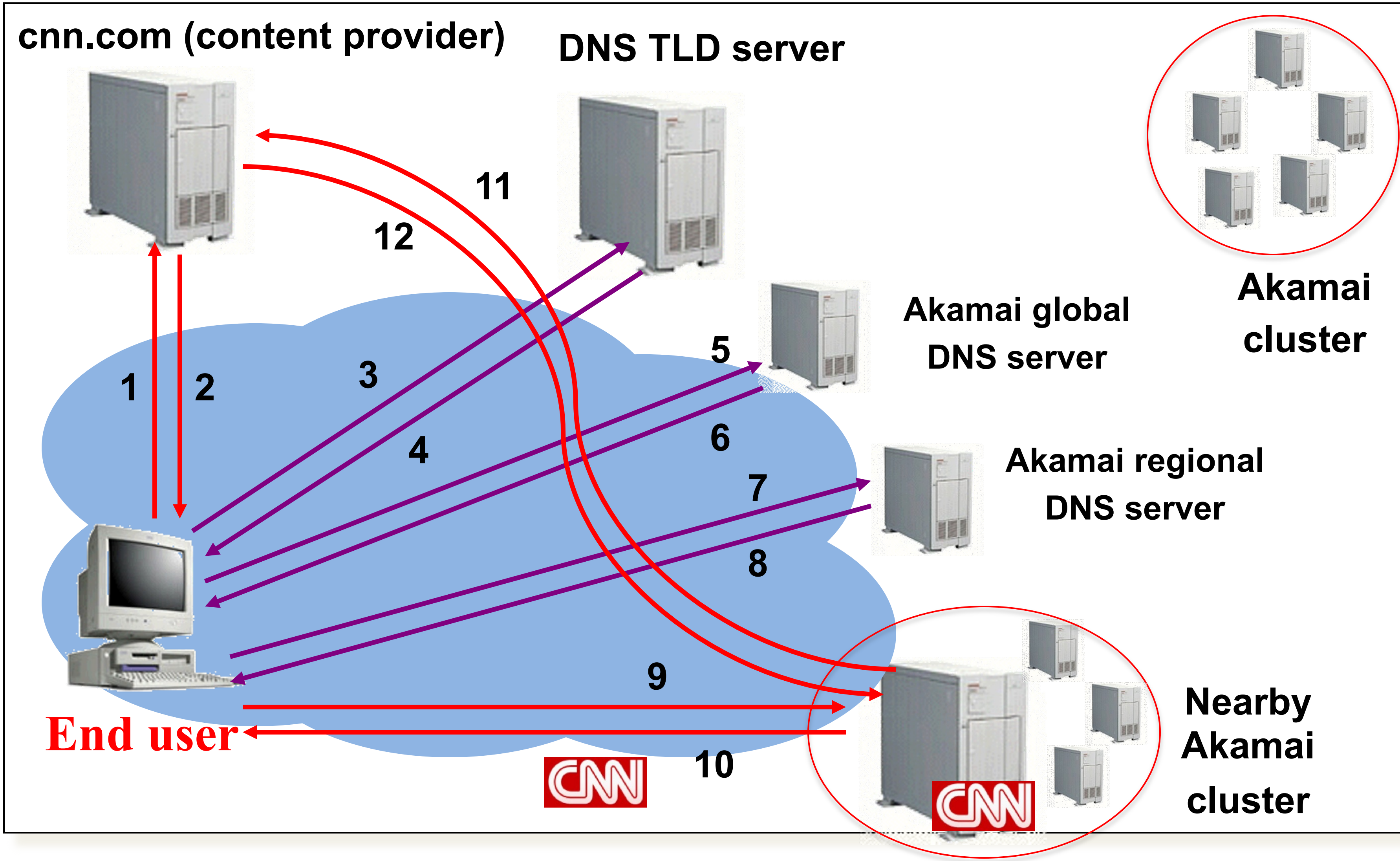
How Akamai Uses DNS



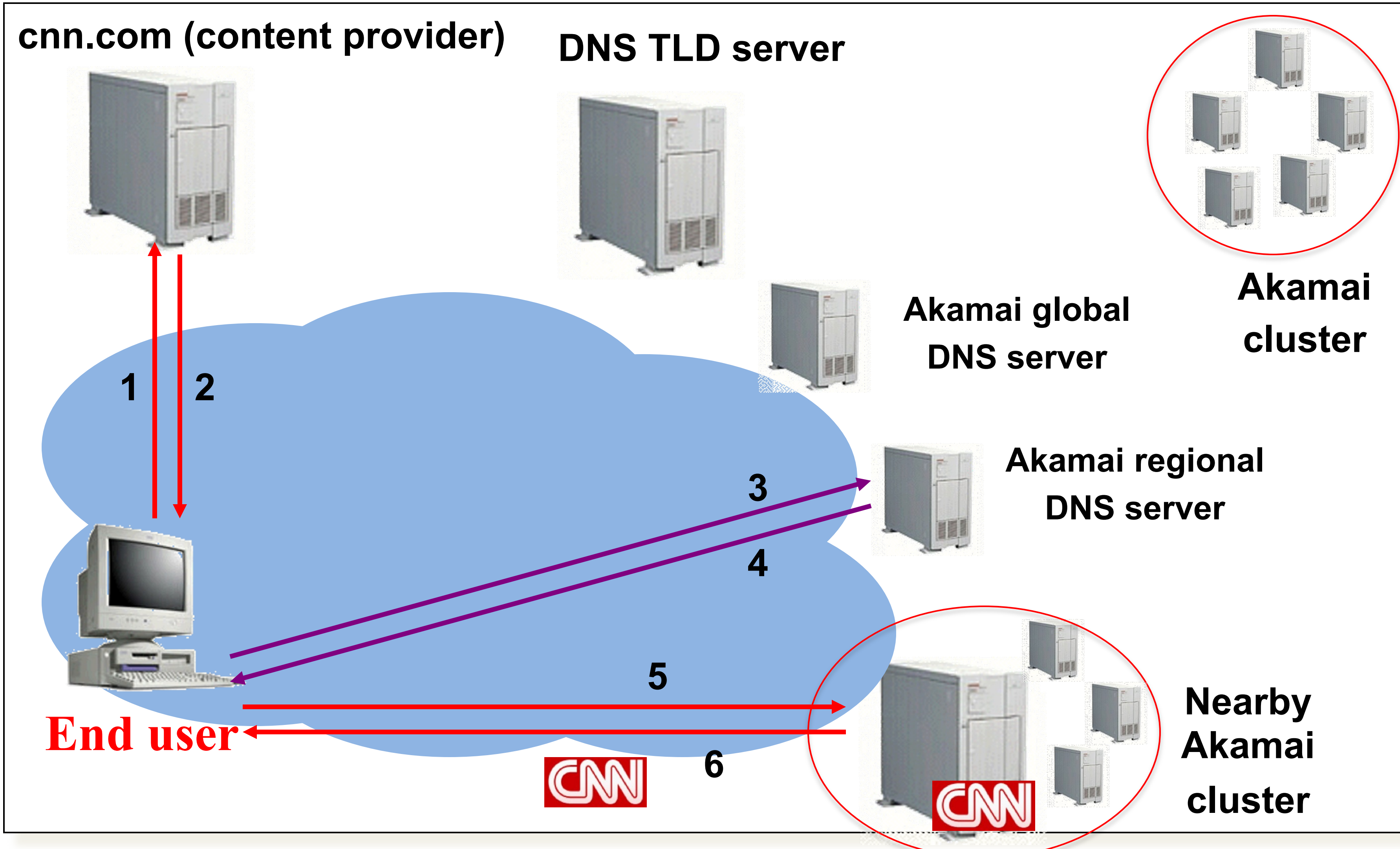
How Akamai Uses DNS



How Akamai Uses DNS



How Akamai Works: Cache Hit



Mapping System

- To make these decisions need a map!

- Equivalence classes of IP addresses
  - IP addresses experiencing similar performance
  - Quantify how well they connect to each other

- **Collect and combine** measurements
  - Ping, traceroute, BGP routes, server logs
    - *e.g.*, over 100 TB of logs per days
  - Network latency, loss, throughput, and connectivity

Routing client requests with the map

- Map each **IP class** to a preferred **server cluster**
  - Based on performance, cluster health, etc.
  - Updated roughly every minute
    - Short, 60-sec DNS TTLs in Akamai regional DNS accomplish this
- Map client request to a server in the cluster
  - **Load balancer** selects a specific server
  - *e.g.*, to **maximize** the cache hit rate

Adapting to failures

- Failing **hard drive** on a server
  - Suspends after finishing "in progress" requests

- Failed **server**
  - Another server takes over for the IP address
  - Low-level map updated **quickly** (load balancer)

- Failed **cluster**, or **network path**
  - High-level map updated **quickly** (ping/traceroute)

Take-away points: CDNs

- Content distribution is hard
  - Many, diverse, changing objects
  - Clients distributed all over the world

- Moving content to the client is key
  - Reduces latency, improves throughput, reliability

- Content distribution solutions **evolved:**
  - Load balancing, reactive caching, to
  - Proactive content distribution networks

Next in 461:

Network Security and Specialized Topics:

- Wireless Networking
- Software-Defined Networking