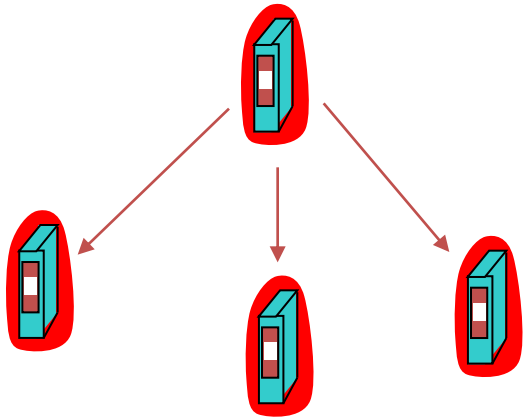


Content Distribution Networks

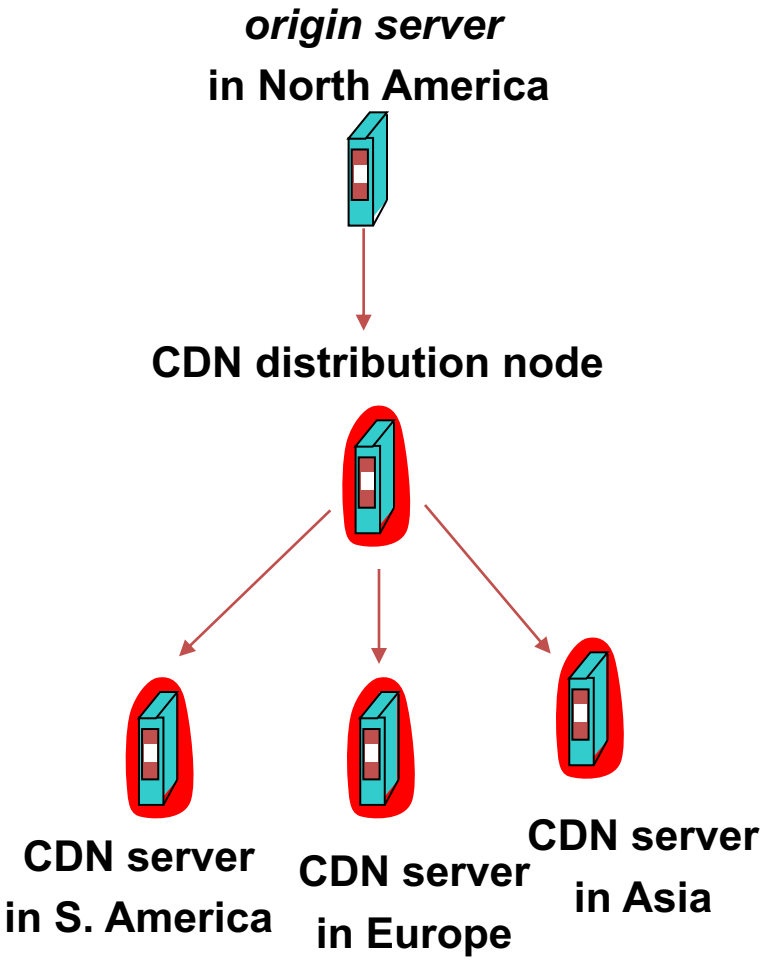
Lecture 16

COS 461: Computer Networks

Kyle Jamieson

Content Distribution Network (CDN)

- Proactive content replication
  - Content provider (e.g., CNN) contracts with a CDN
- CDN replicates the content
  - On many servers spread throughout the Internet
- Updating the replicas
  - Reactive by TTL or updates pushed to replicas when the content changes



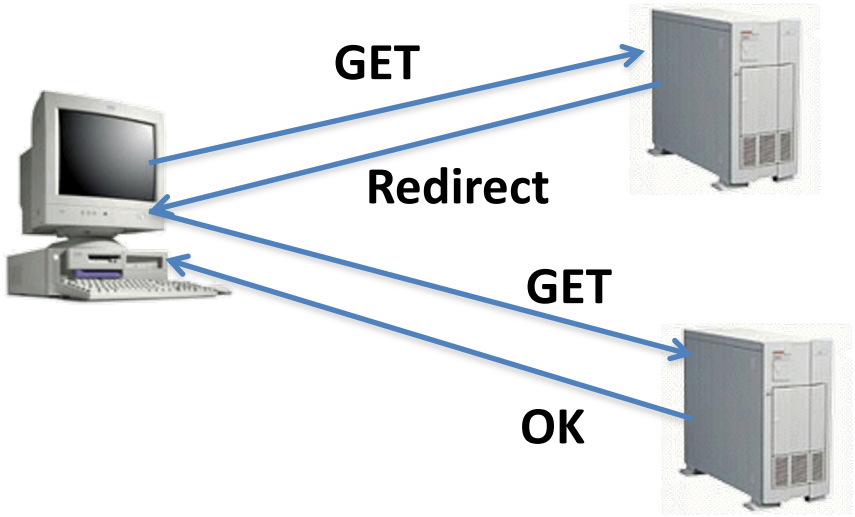
Server Selection Policy

- Live server
  - For availability
- Lowest load
  - To balance load across the servers
- Closest
  - Nearest geographically, or in round-trip time
- Best performance
  - Throughput, latency, …
- Cheapest bandwidth, electricity, …

Requires continuous monitoring of liveness, load, and performance

Server Selection Mechanism

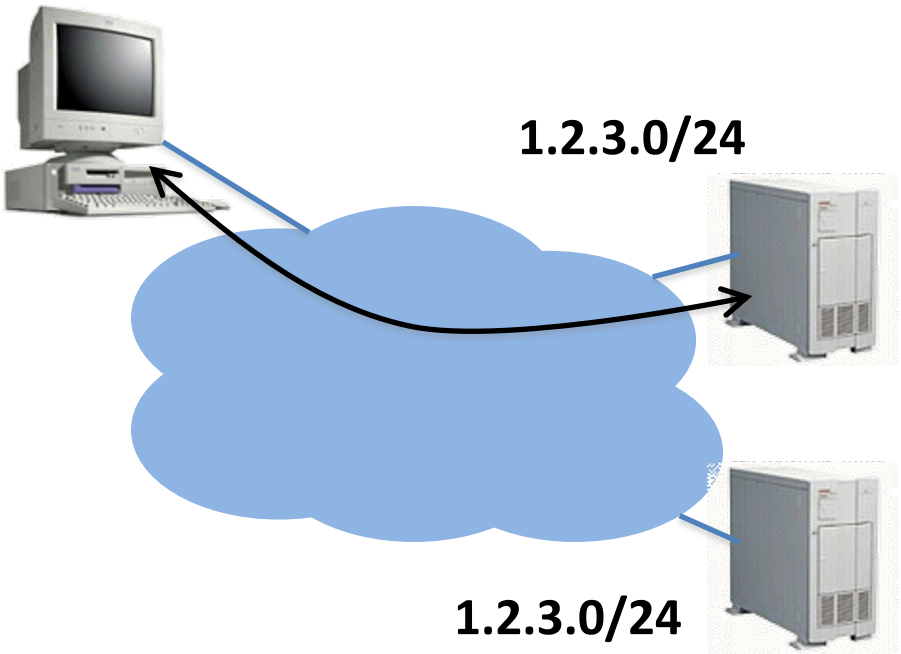
- Application
  - HTTP redirection



- Advantages
  - Fine-grain control
  - Selection based on client IP address
- Disadvantages
  - Extra round-trips for TCP connection to server
  - Overhead on the server

Server Selection Mechanism

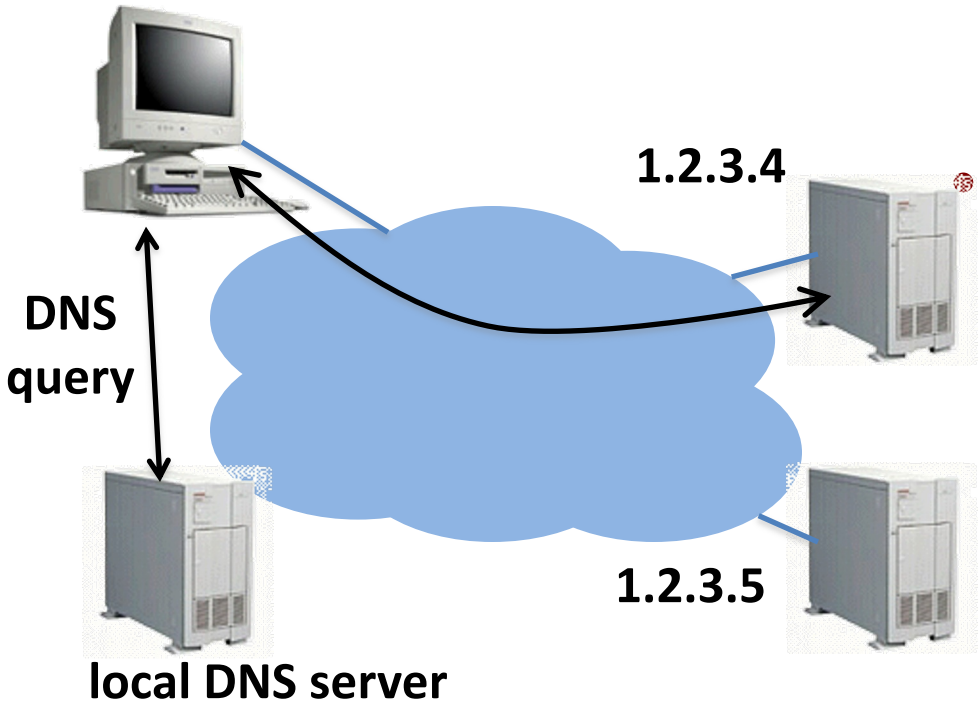
- Routing
  - Anycast routing



- Advantages
  - No extra round trips
  - Route to nearby server
- Disadvantages
  - Does not consider network or server load
  - Different packets may go to different servers
  - Used only for simple request-response apps

Server Selection Mechanism

- Naming
  - DNS-based server selection



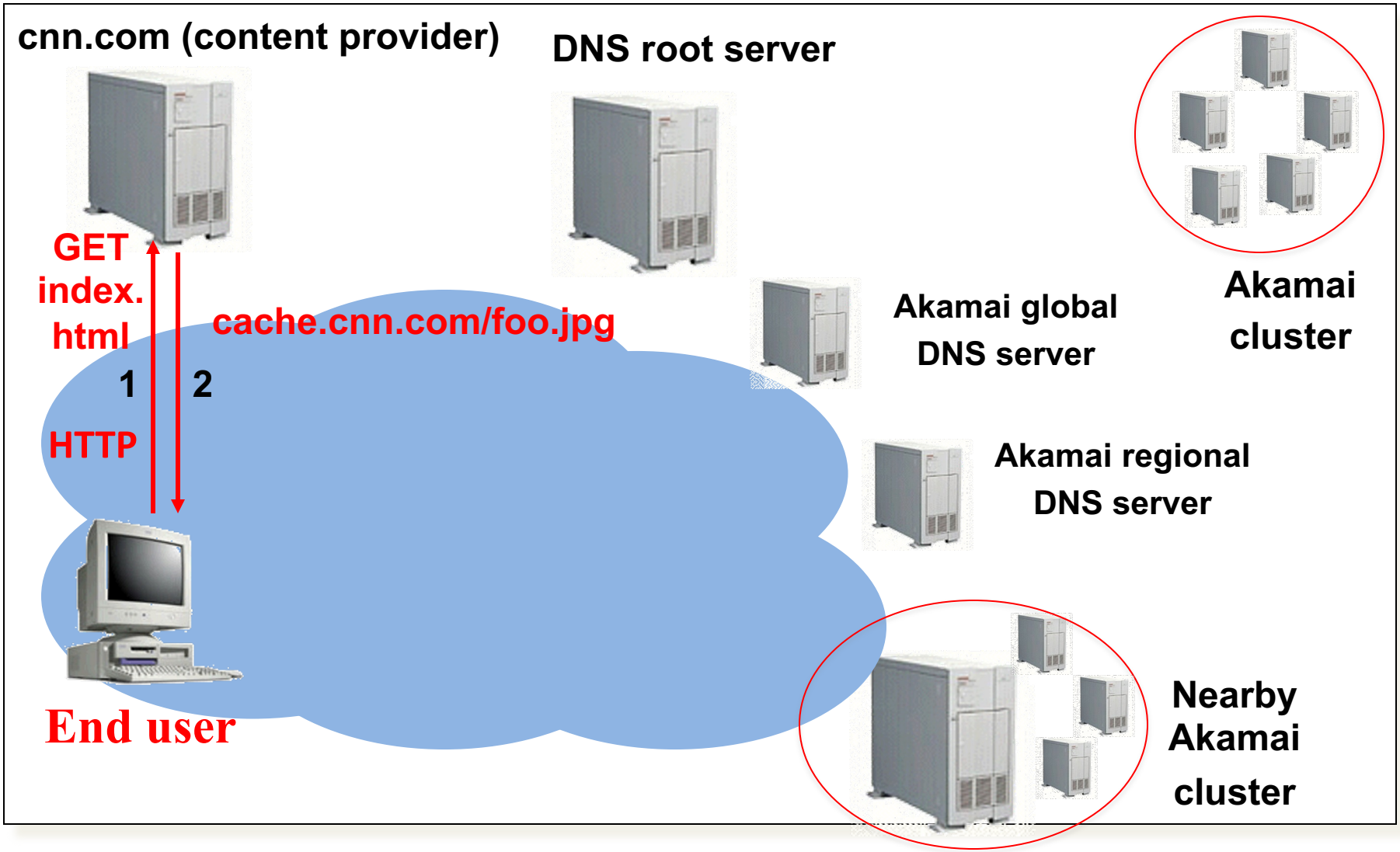
- Advantages
  - Avoid TCP set-up delay
  - DNS caching reduces overhead
  - Relatively fine control
- Disadvantage
  - Based on IP address of local DNS server
  - "Hidden load" effect
  - DNS TTL limits adaptation

How Akamai Works

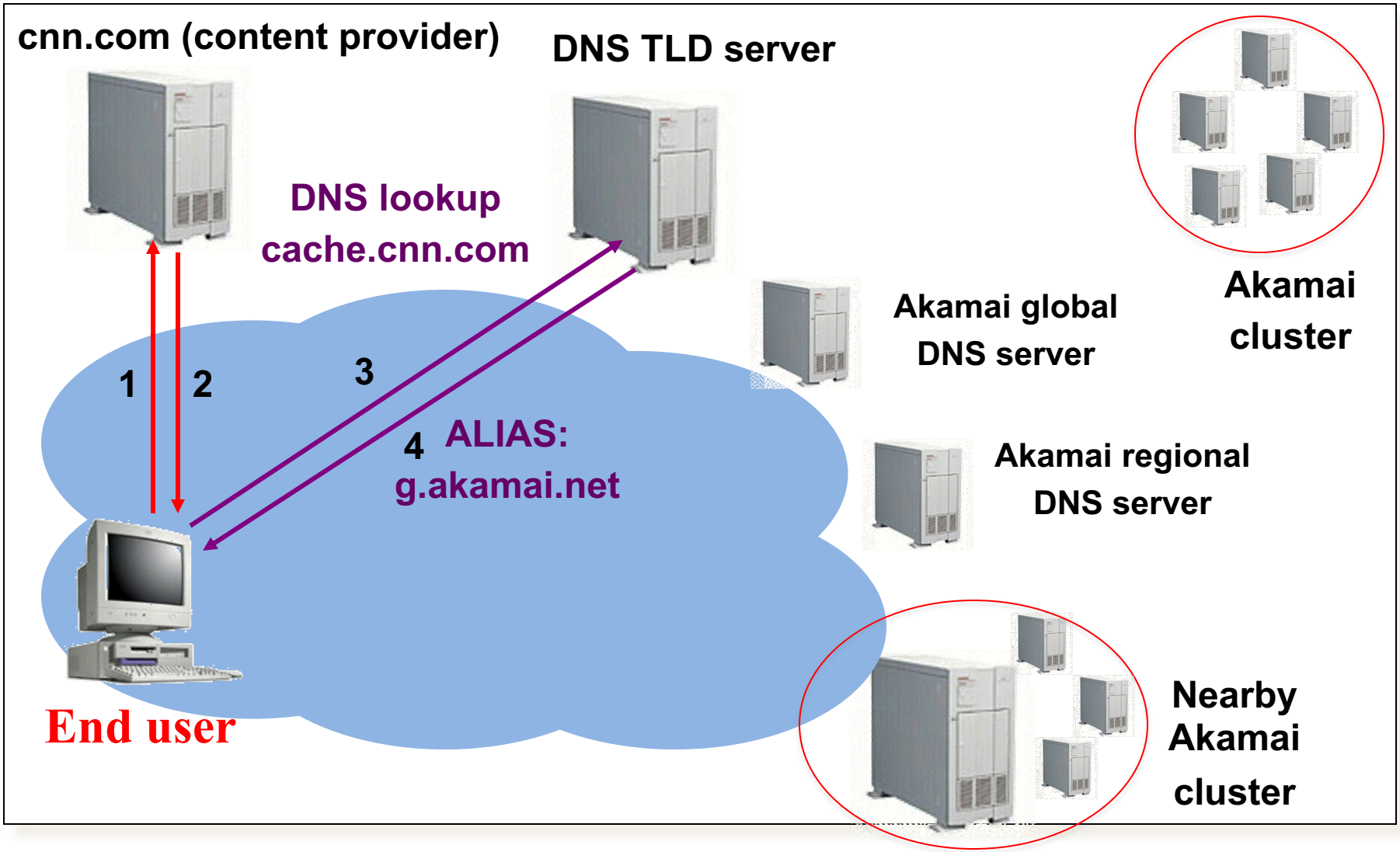
Akamai Statistics

- Distributed servers
  - Servers: ~275,000
  - Networks: 1,500
  - Countries: 136
  - Arranged into *clusters* of co-located servers
- Many customers
  - 50% of Fortune Global 500 Corp.
- Network
  - Up to 50 Tbps daily
  - 2019 Cricket World Cup: 25.3M concurrent viewers
  - 85% of Internet is one network hop from Akamai servers

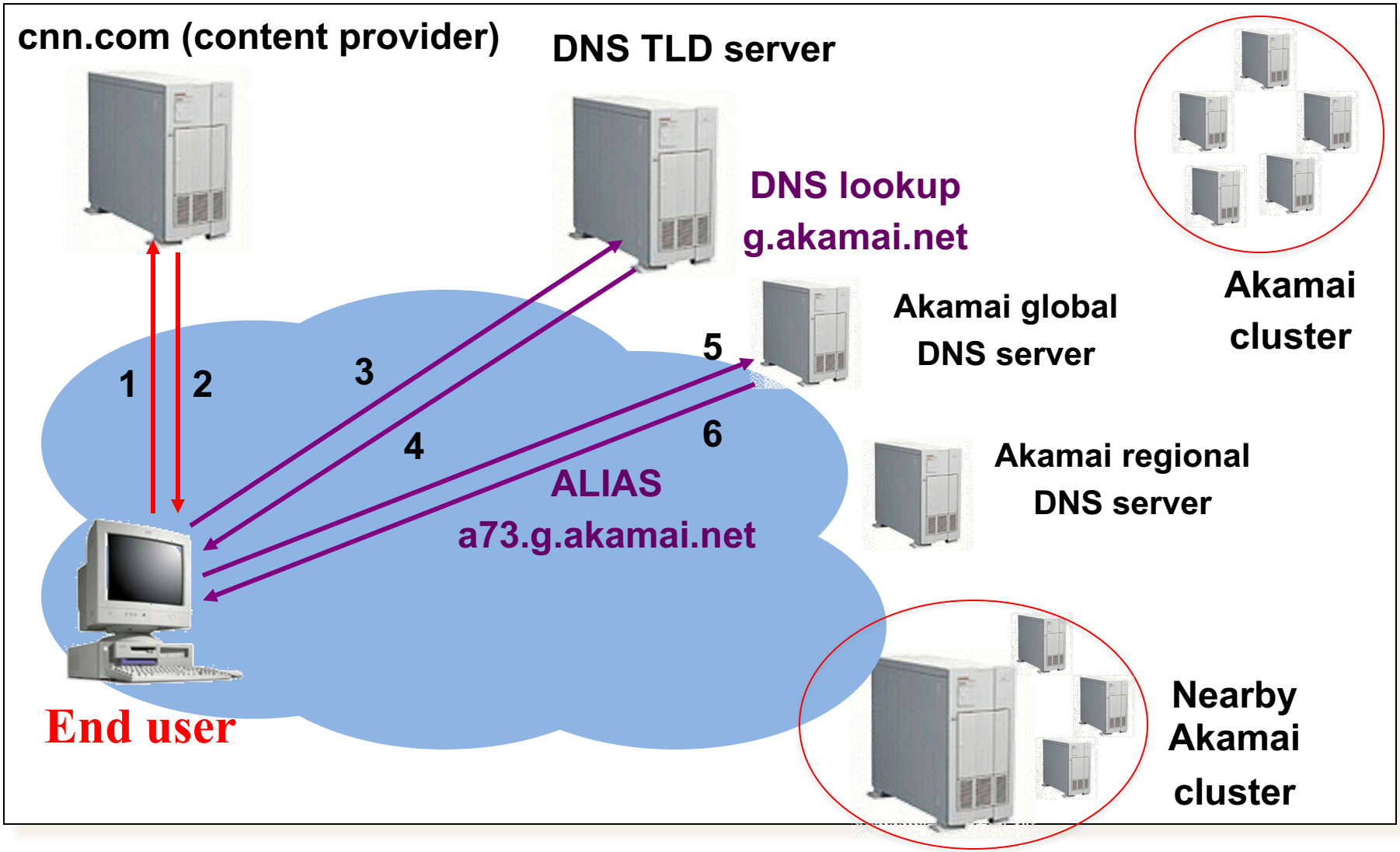
How Akamai Uses DNS



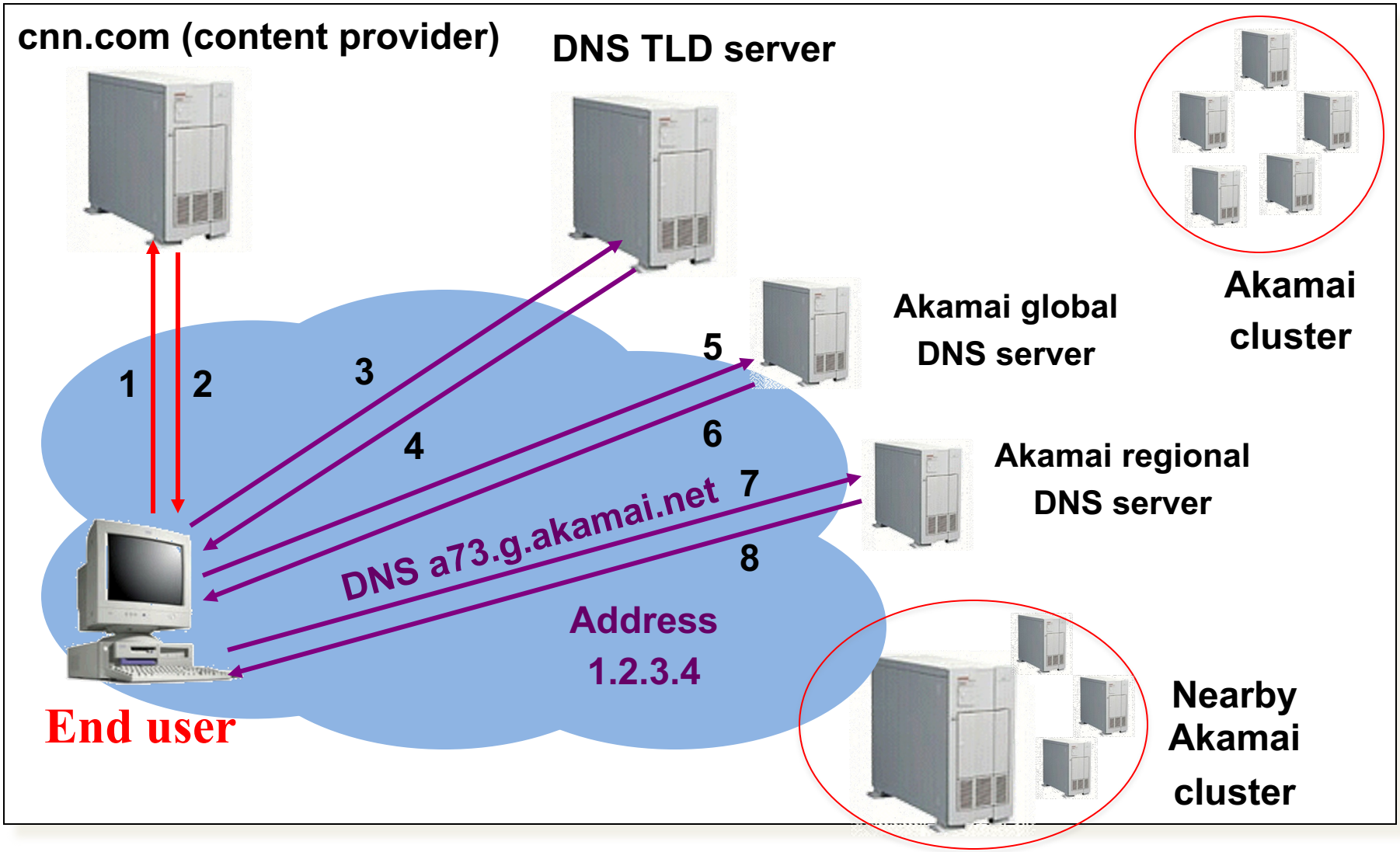
How Akamai Uses DNS



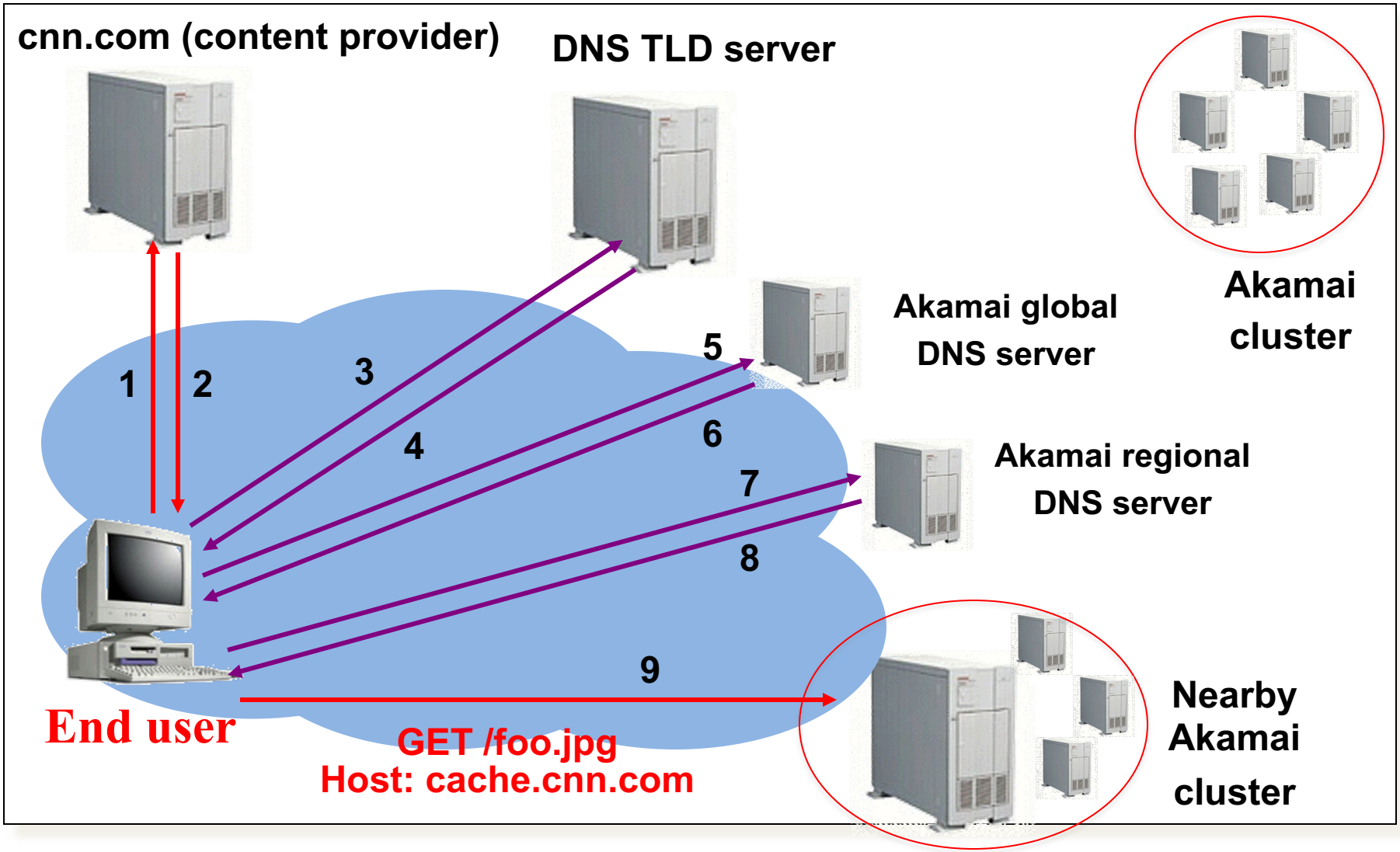
How Akamai Uses DNS



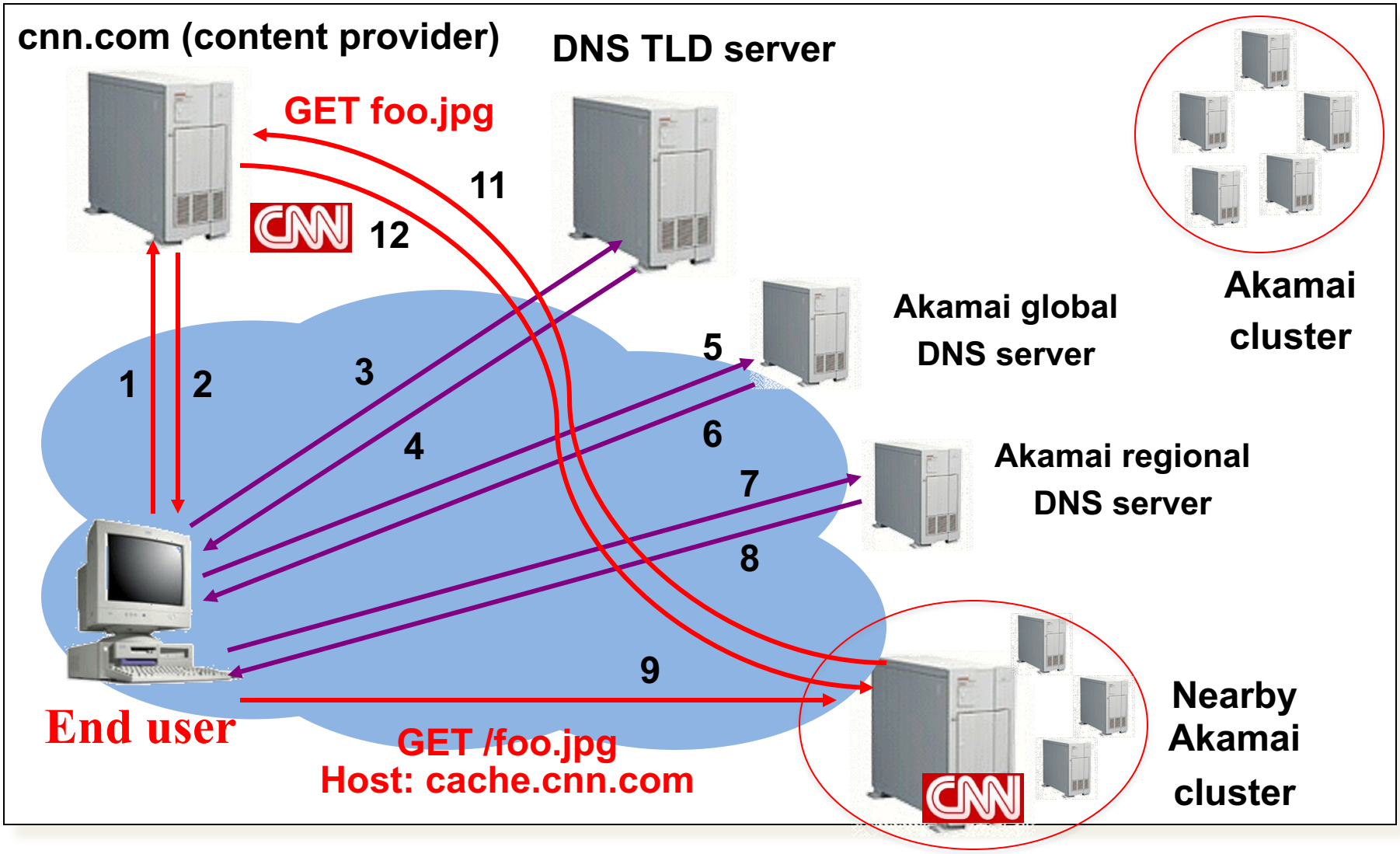
How Akamai Uses DNS



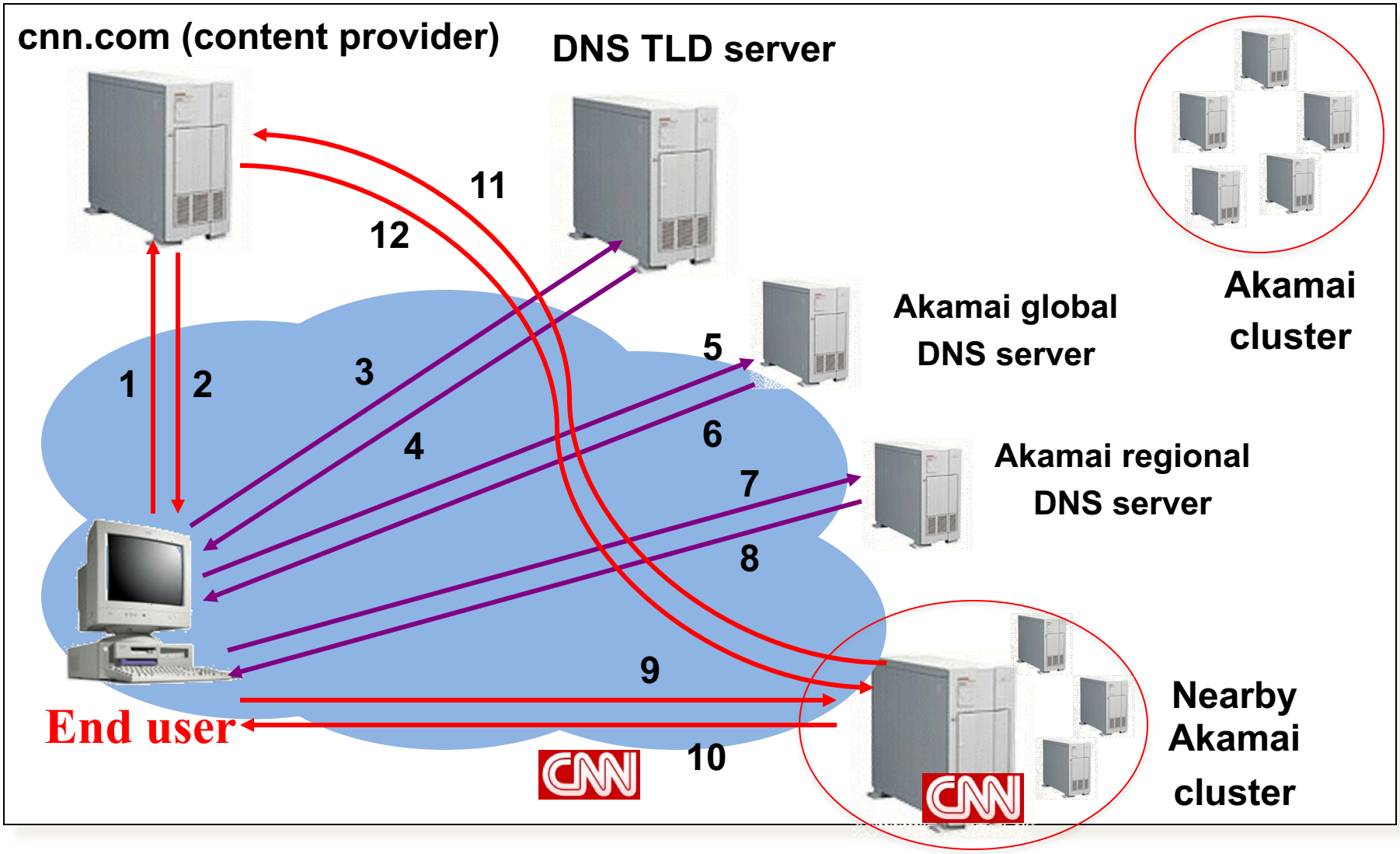
How Akamai Uses DNS



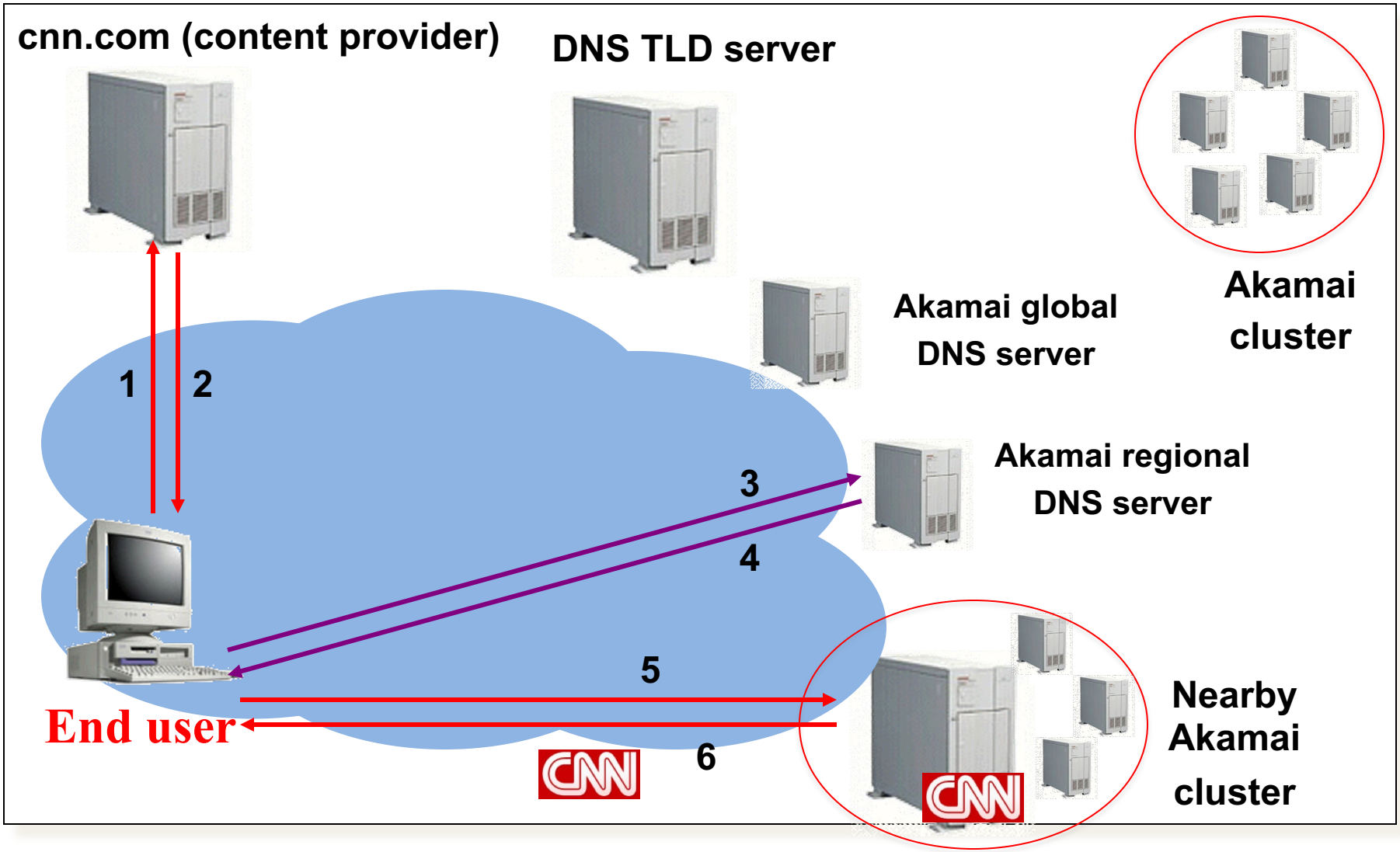
How Akamai Uses DNS



How Akamai Uses DNS



How Akamai Works: Cache Hit



Mapping System

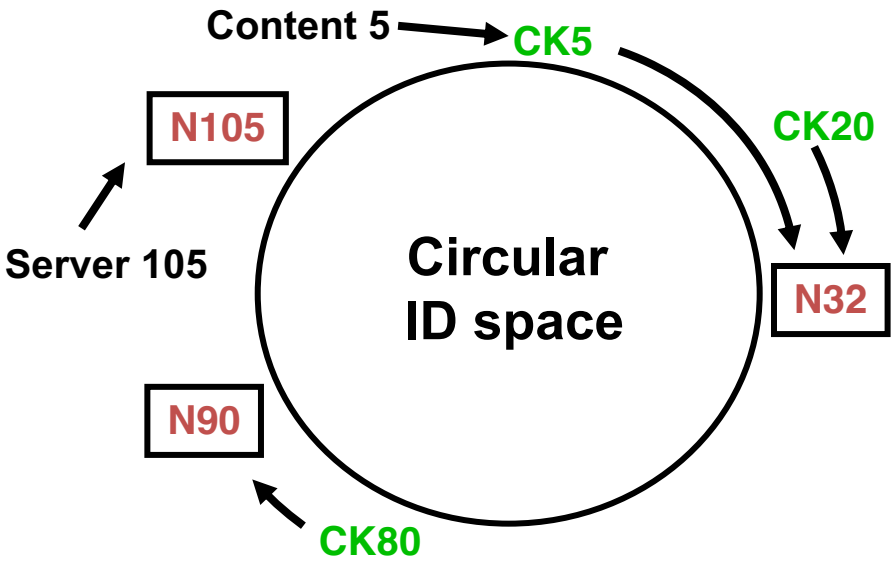
- Equivalence classes of IP addresses
  - IP addresses experiencing similar performance
  - Quantify how well they connect to each other
- Collect and combine measurements
  - Ping, traceroute, BGP routes, server logs
    - *e.g.*, over 100 TB of logs per day
  - Network latency, loss, and connectivity

Routing Client Requests within 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

Selecting server inside cluster

- "Consistent hashing" algorithm
  - content_key = hash(URL) mod N
  - node_key = hash(server ID) mod N
  - **Rule:** Place content on server whose node_key is the successor server of URL's content_key



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
- Failed cluster or network path
  - High-level map updated quickly
- Failed path to customer's origin server
  - Route packets through an intermediate node

Akamai Transport Optimizations

- Bad Internet routes
  - Overlay routing through an intermediate server
- Packet loss
  - Sending redundant data over multiple paths
- TCP connection set-up/teardown
  - Pools of persistent connections
- TCP congestion window and round-trip time
  - Estimates based on network latency measurements

Akamai Application Optimizations

- Slow download of embedded objects
  - Prefetch when HTML page is requested
- Large objects
  - Content compression
- Slow applications
  - Moving applications to edge servers
  - E.g., content aggregation and transformation
  - E.g., static databases (e.g., product catalogs)

Conclusion

- Content distribution is hard
  - Many, diverse, changing objects
  - Clients distributed all over the world
- Moving content towards client is key
  - Reduces latency, improves throughput, reliability
- Contribution distribution solutions evolved
  - Reactive caching, load balancing, to
  - Proactive content distribution networks