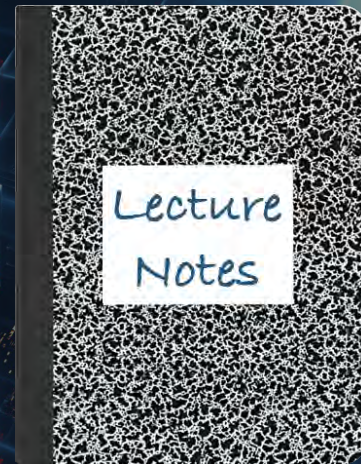


CS 417 – DISTRIBUTED SYSTEMS

Week 11: Content Delivery

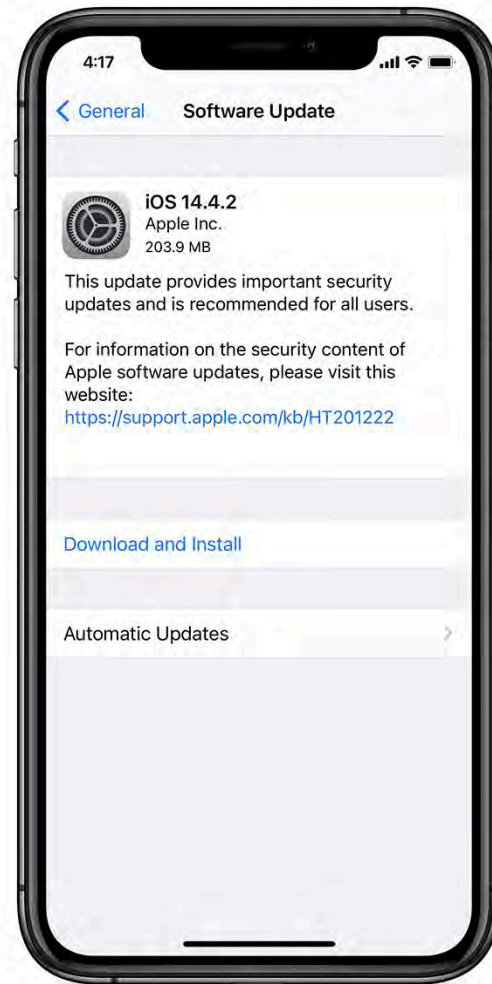
Part 1: Content Delivery Networks



Paul Krzyzanowski

© 2021 Paul Krzyzanowski. No part of this content, may be reproduced or reposted in whole or in part in any manner without the permission of the copyright owner.

How do you update
~900M phones?



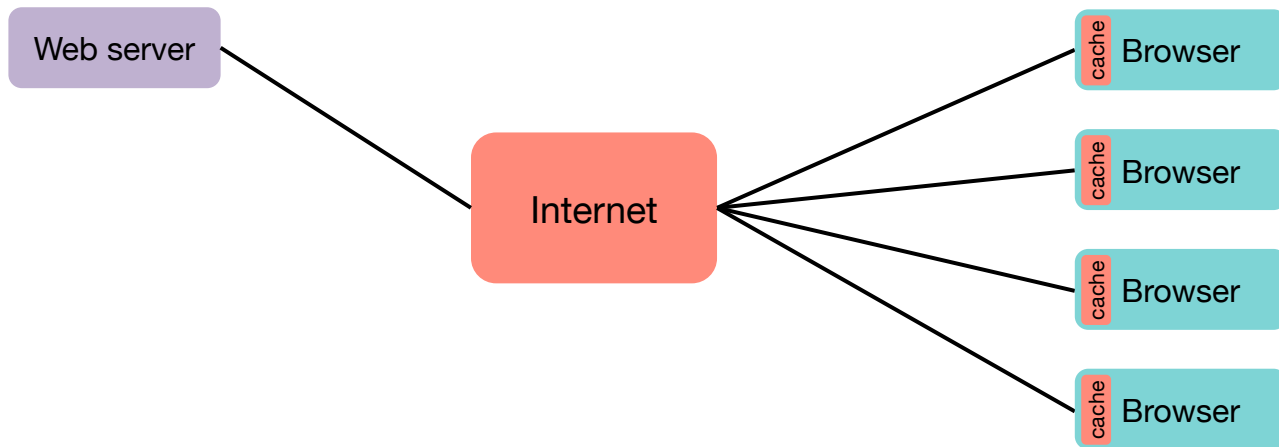
Motivation

- Serving content from one location presents problems
 - Scalability
 - Reliability
 - Performance
- “Flash crowd” problem
 - What if everyone comes to your site at once?
- What do we do?
 - Cache content and serve requests from multiple servers at the network edge (close to the user)
 - Reduce demand on site’s infrastructure
 - Provide faster service to users
 - Content comes from nearby servers

Focus on Content

- Computing is still done by the site host's server(s)
- Offload the static parts – they often make up the bulk of the bytes:
 - Images
 - Video
 - CSS files
 - Static pages

Serving & Consuming Content

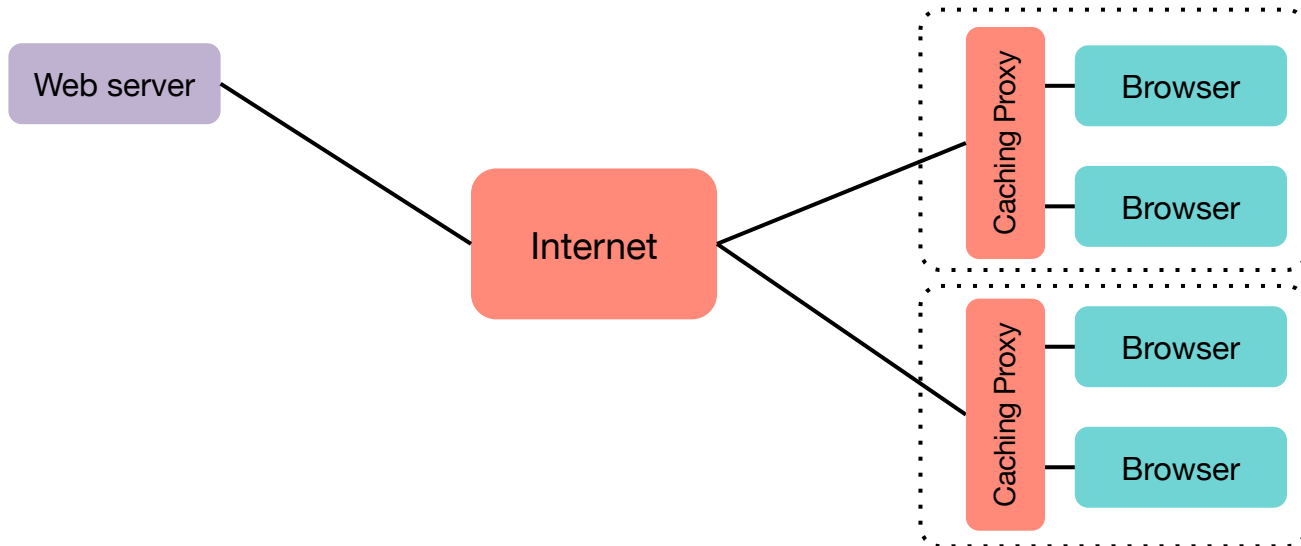


Every request goes to the server.

Repeated requests from one client may be optimized by **browser-based caching**

... but that cached data is local to the browser

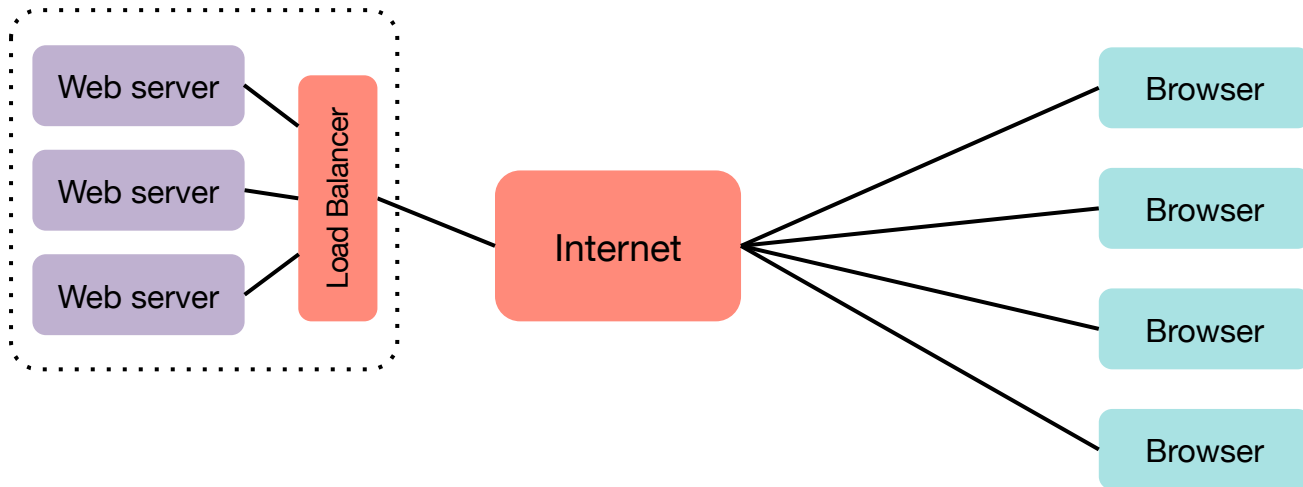
Caching Proxies



Caching proxy in an organization.

Take advantage of what others before you have recently accessed.

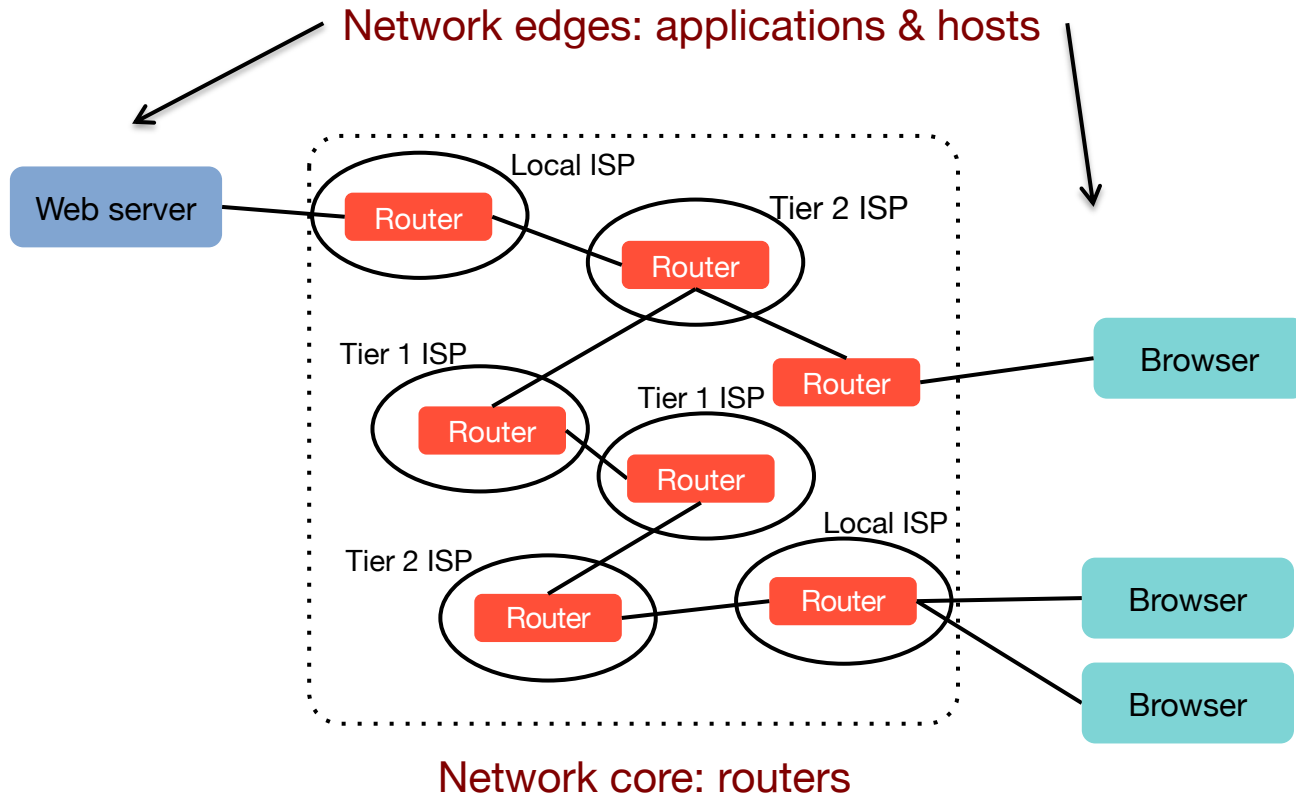
Load Balancing



Increase capacity at the server.

Internet connectivity can be a bottleneck ... + latency from client to server.

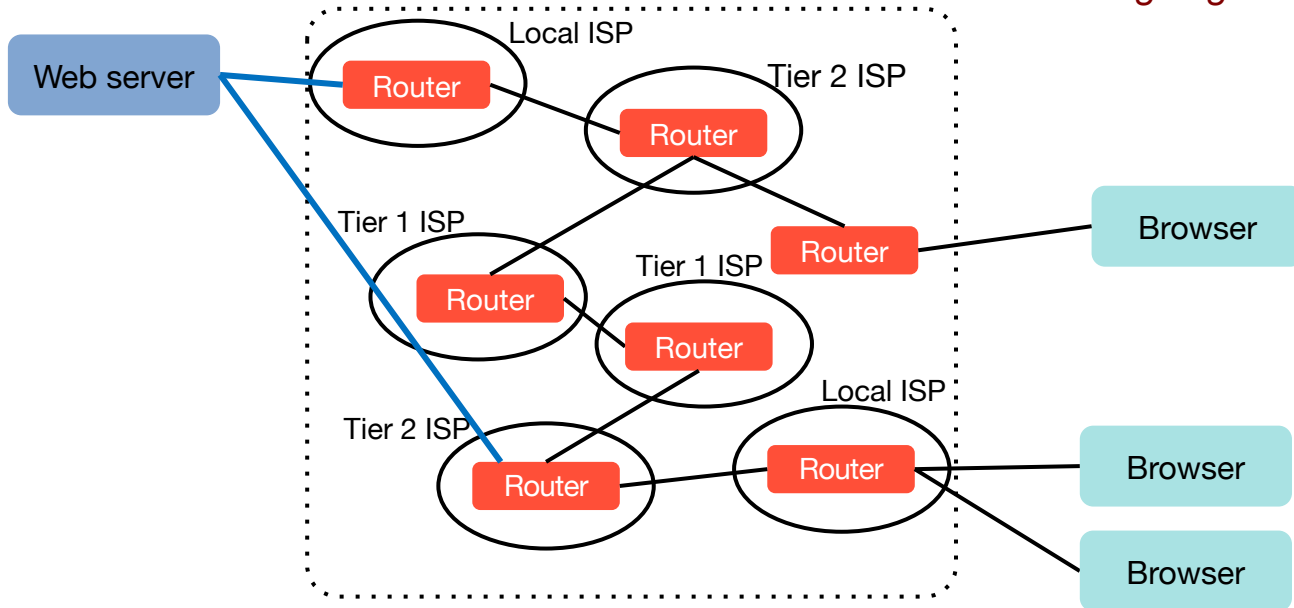
Internet End-to-End Packet Delivery



Multihoming

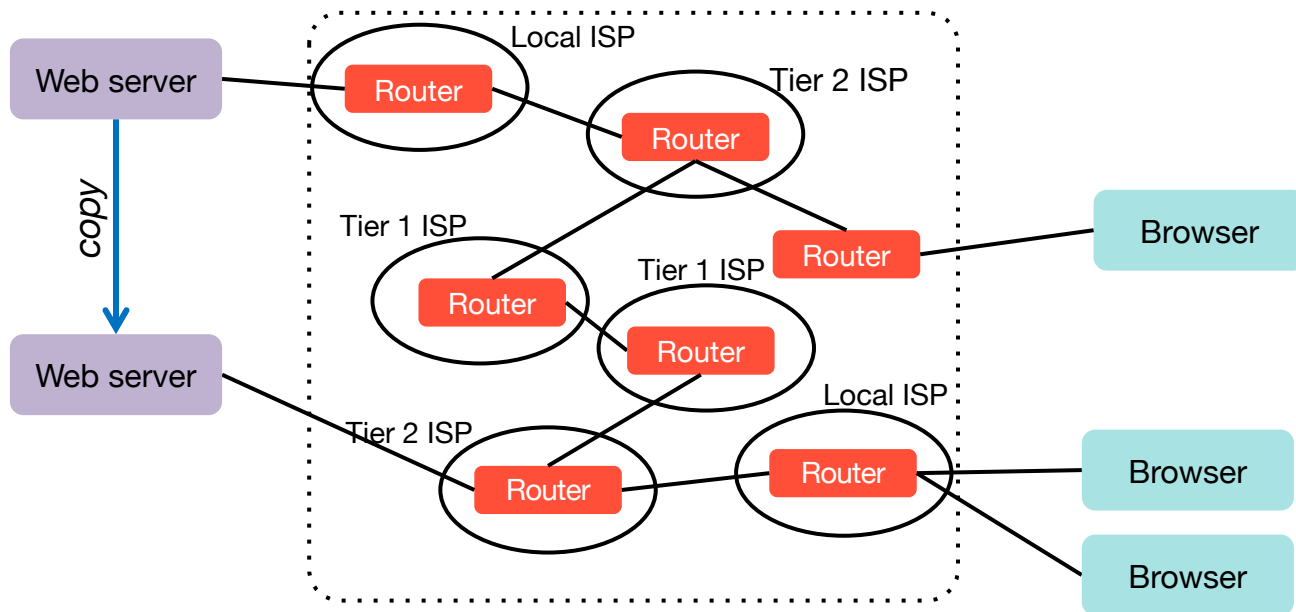
- Get network links from multiple ISPs
- Server has one IP address but multiple links
- Announce address to upstream routers via BGP:

*Provides clients with a **choice of routes** and **fault tolerance** for a server's ISP going down*



Mirroring (Replication)

- Synchronize multiple servers
- Use multiple ISPs: location-based load balancing, ISP & server fault tolerance



Improving scalability, availability, & performance

- **Scalability**

- Mirror (replicate) servers for load balancing among multiple servers
- Multiple ISPs if network congestion is a concern

- **Availability**

- Replicate servers
- Multiple data centers & ISPs

- **Performance**

- Cache content and serve requests from multiple servers at the network edge (close to the user)
 - Reduce demand on site's infrastructure
 - Provide faster service to users
 - Content comes from nearby servers

But these approaches have problems!

- **Local balancing**
 - Data center or ISP can fail
- **Multihoming**
 - IP protocols (BGP) are often not quick to find new routes
- **Mirroring at multiple sites**
 - Synchronization can be difficult
- **Proxy servers**
 - Typically, a client-side solution
 - Low cache hit rates

All require extra capacity and extra capital costs

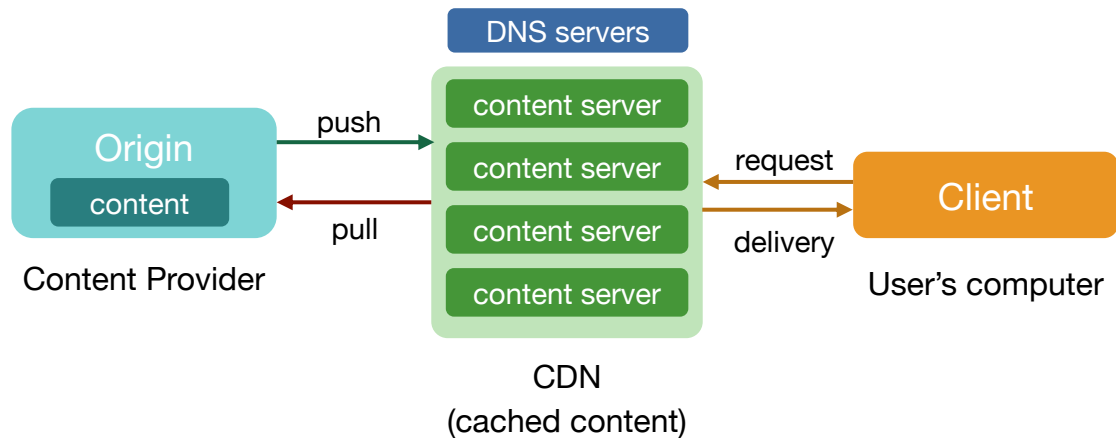
Akamai Distributed Caching

- Company evolved from MIT research
- "Invent a better way to deliver Internet content"
- Tackle the "flash crowd" problem
- Akamai runs on ~325,000 servers in ~1,435 networks across ~130 countries
 - Delivers 15-30% of all web traffic ... reaching over 100 Terabits per second
 - Used by
 - More than 950 enterprise hardware and software companies
 - More than 850 retailers
 - More than 300 of the world's banks and 325 financial services firms
 - More than 275 telcos, carriers, and ISPs
 - More than 226 broadcast and pay TV networks
 - More than 225 game publishers
 - ...



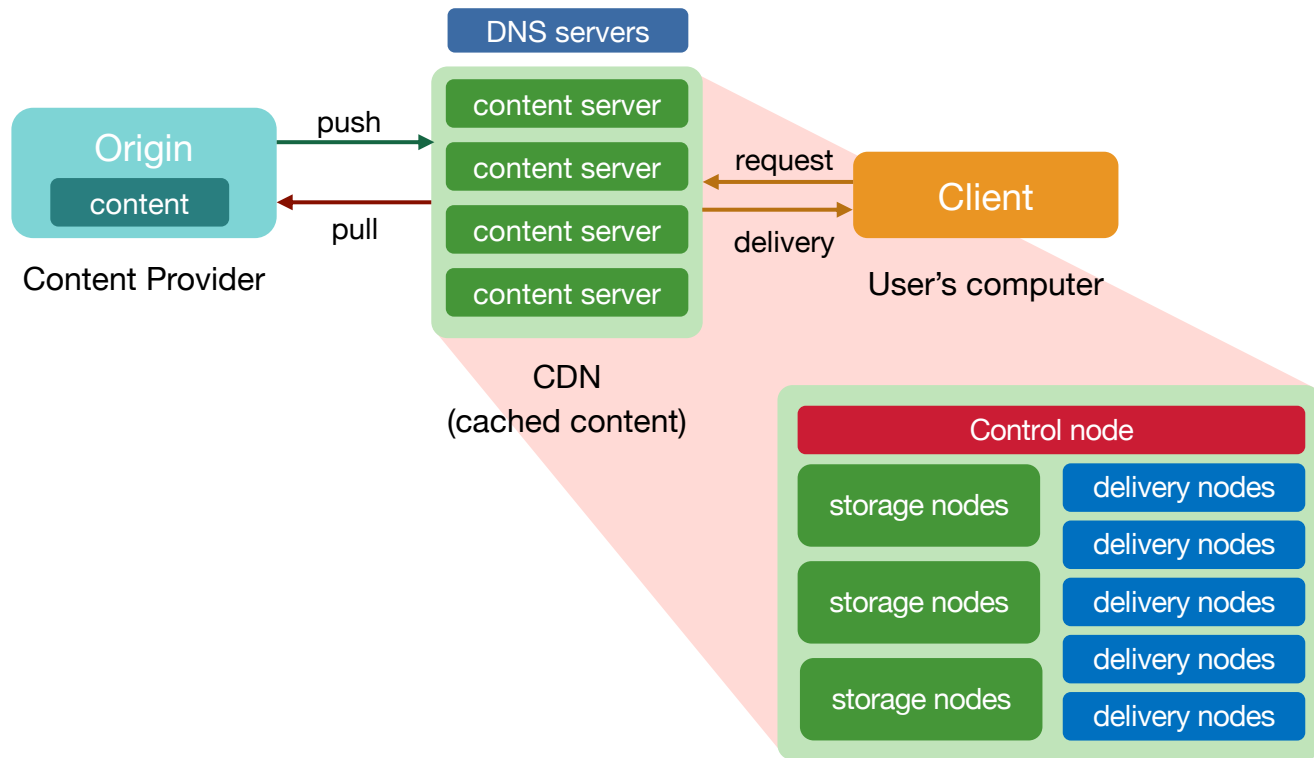
http://www.akamai.com/html/about/facts_figures.html

CDN Structure: Pushing & Pulling



- **Push CDNs**
 - Origin must store content manually onto delivery nodes
- **Pull CDNs**
 - Delivery nodes request content from the origin

CDN Structure: Storage, Delivery & Control



Akamai's goal

Try to serve clients from caching servers that are:

- **Nearest**: lowest round-trip time
- **Available**: server that is not too loaded
- **Likely**: server that is likely to have the data

Overlay Network

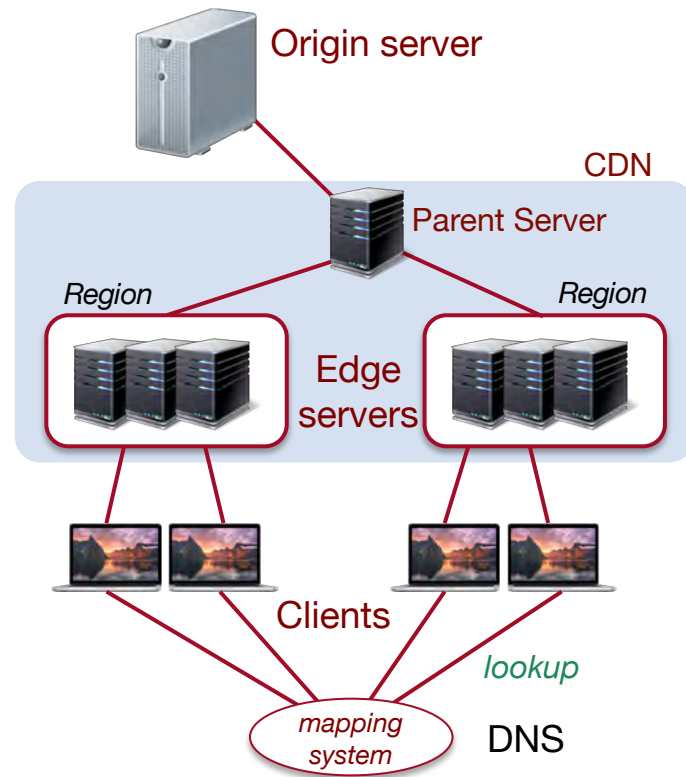
The Internet is a collection of many autonomous networks

- Routing is based on **business decisions**
 - Peering agreements, **not performance**
- An ISP's top performance incentives are:
 - Last-mile connectivity to end users
 - Connectivity to servers on the ISP
- **Akamai's Overlay network**
 - Collection of caching servers at many, many ISPs
 - All know about each other
 - High-performance routing & tiered distribution

Overlay Network: (1) DNS Lookup

1. Domain name lookup

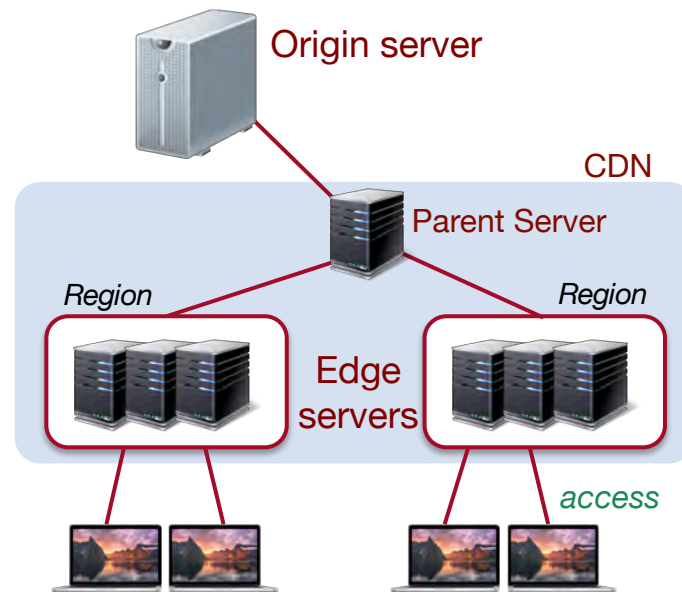
- Translated by **mapping system** to an edge server that can serve the content
- Use custom **dynamic DNS servers**
 - Origin sets up a DNS CNAME (alias) record to point to an Akamai domain (e.g., www.example.com.edgesuite.net)
 - Take requestor's address into account to find the *nearest edge*
- Resolve a host name based on:
 - User location (minimize network distance)
 - Server health
 - Server load
 - Network status
 - Load balancing
- Try to find an edge server at the customer's ISP



Overlay Network: (2) Multi-Tier Content Lookup

2. Browser sends request to the given edge server

- Edge server may be able to serve content from its cache
- May need to contact the **origin server** via the **transport system**

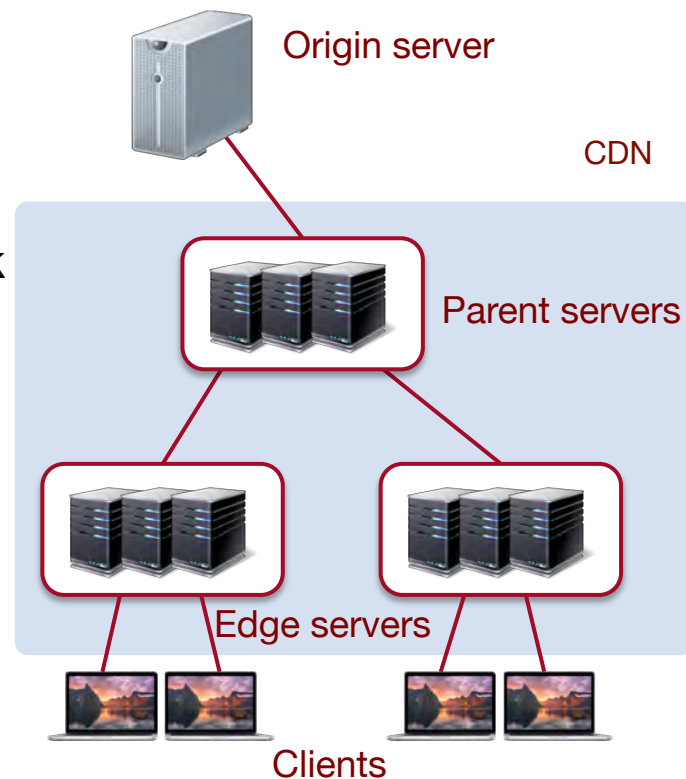


Benefits of a CDN

1. Caching
2. Routing
3. Security
4. Analytics
5. Cost

1. Caching

- Goal: Increase hit rate on edge servers
 - Reduce hits on origin servers
- Two-level caching
 - If edge servers don't have the data, check with **parent servers**
- Static content can be served from caches
 - Dynamic content still goes back to the origin



1. Caching: types of content

- **Static content**

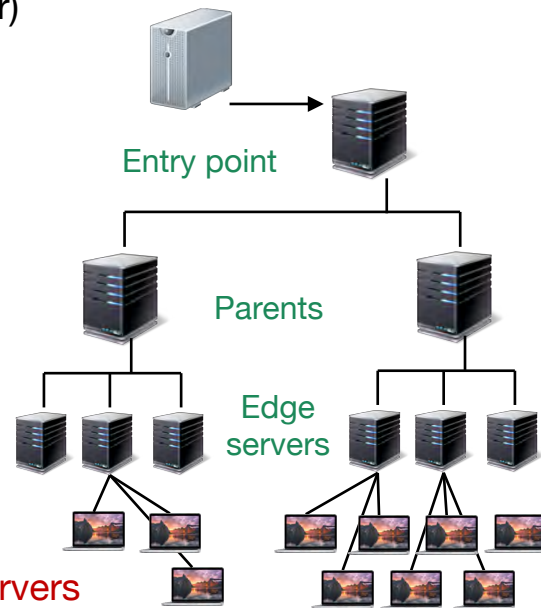
- Cached depending on original site's requirements (never to forever)

- **Dynamic content**

- Caching proxies cannot do this
- Akamai uses *Edge Side Includes* technology (www.esi.org)
 - Assembles dynamic content on edge servers
 - Similar to server-side includes
 - Page is broken into fragments with independent caching properties
 - Assembled on demand

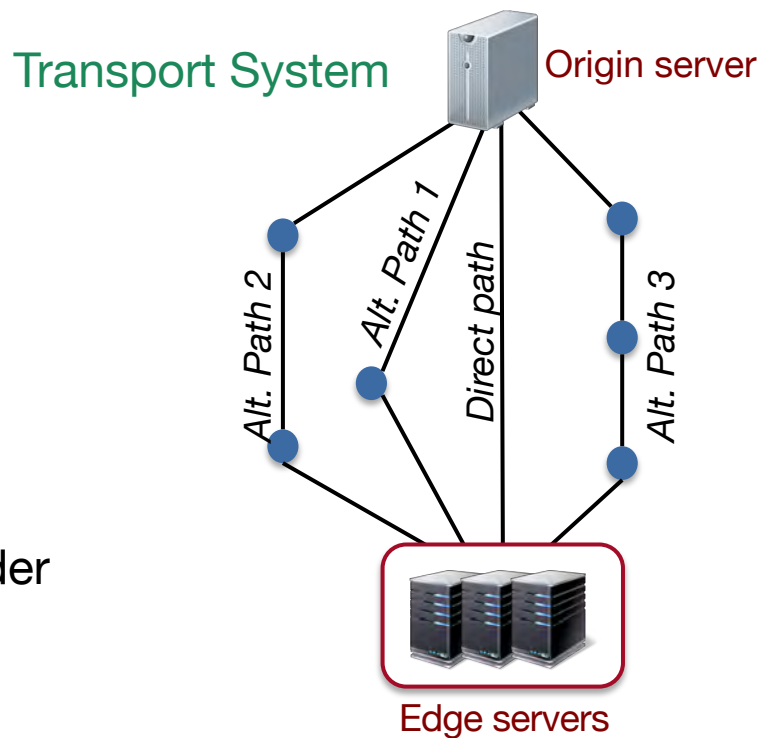
- **Streaming media**

- Live stream is sent to an entry-point server in the CDN network
- Stream is delivered from the **entry point server** to **multiple edge servers**
- Edge servers serve content to end users



2. Routing

- Route to parent servers or origin via the overlay network
- Routing decision factors:
 - measured latency
 - packet loss
 - available bandwidth
- Results in **ranked list of alternate paths** from edge to origin
- Each intermediate node acts as a forwarder
 - Keep TCP connections active for efficiency



3. Security

- High capacity
 - Overwhelm DDoS attacks
- Expertise
 - Maintain systems and software
- Extra security software
 - Hardened network stack
 - Detect & defend attacks
- Shield the origin
 - Attacks hit the CDN, not the origin

4. Analytics

- Reports on quality of service, latency, media performance
- Engagement: # views, duration, abandoned plays
- Geography: zip code, continent, region, ISP
- Clients: devices, operating systems
- Most popular content
- Session: bandwidth, referrer URL, session duration

Collect network performance data

- **Map network topology**
 - Based on **BGP** and *traceroute* information
 - Estimate hops and transit time
- **Monitor load**
 - Content servers report their load to a monitoring application
 - Monitoring app publishes load reports to a local (Akamai) DNS server
- **Assign servers**
 - Dynamic DNS server determines which IP addresses to return when resolving names
- **Load shedding:**
 - If servers get too loaded, the DNS server will not respond with those addresses

5. Cost

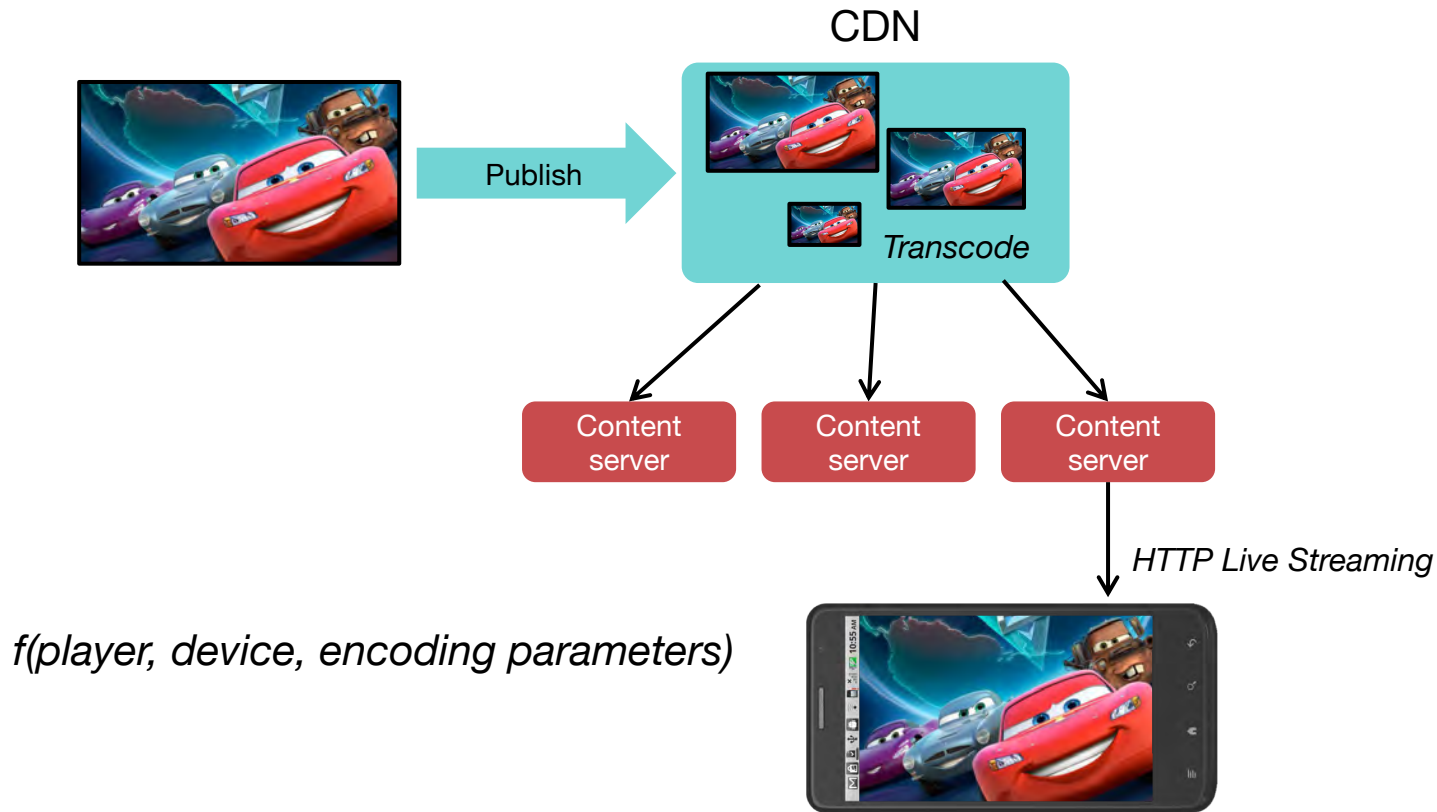
- Infrastructure on demand
 - CDN absorbs majority of content
- Instant worldwide scaling based on demand
- Business advantages

Video Streaming via CDNs

How is live video different?

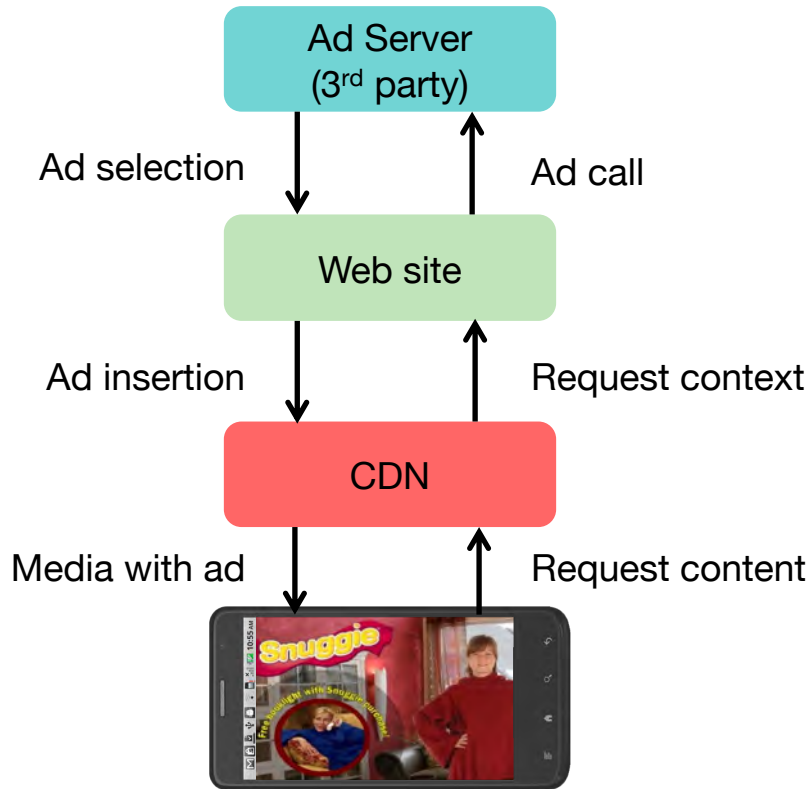
- Live video cannot be cached
 - Progressive downloads – watch video while downloading
 - vs. direct downloads – download first, watch later
- HTTP Live Streaming (HLS): most popular way to access video
 - Use generic HTTP servers
 - Deliver on-demand video just like any other content
- Adaptive bitrate coding (ABR) – added at CDN
 - Break video stream to chunks (between 2-10 seconds)
 - CDN encodes chunks at various bitrates (quality & resolution)
 - Uses feedback from user's playback client to pick optimal next chunk
 - Revise constantly

ABR Transcoding



Server-side Video Ad Insertion

- Pre-roll, post-roll, mid-roll, overlay, etc.
- Clickable ads, skippable ads
- Integrate with ad servers (DoubleClick, LiveRail, Tremor, YuMe, ...)
- Supported by Google Dynamic Ad Insertion, Amazon AWS Server-Side Ad Insertion, Limelight Orchestrate™, Verizon Smartplay, ...



Example: Limelight Reach Ads

The End