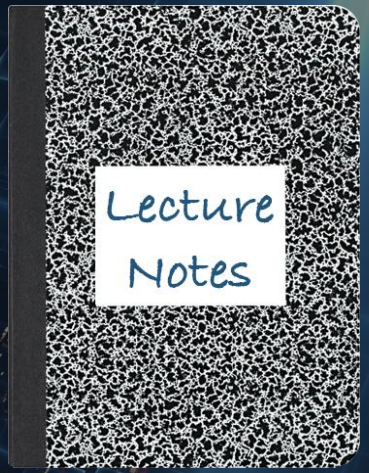


CS 417 – DISTRIBUTED SYSTEMS

Week 11: Content Delivery

Part 2: Content Delivery Networks

Paul Krzyzanowski

How do you update ~1B phones

… or enable millions of people to stream the same video?

Motivation

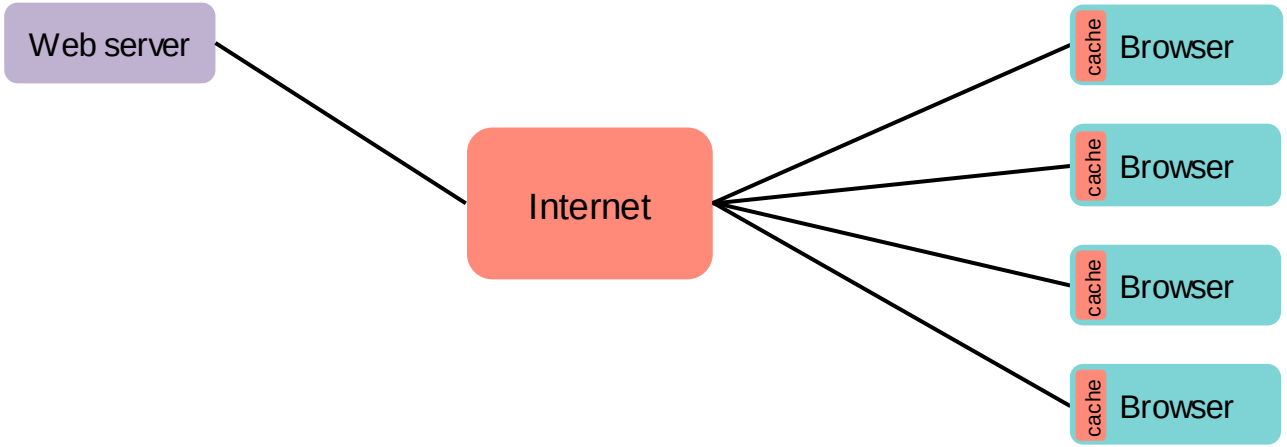
- Serving content from one location presents problems
  - Scalability
  - Reliability
  - Performance
- "Flash crowd" problem
  - What if everyone comes to your site at once?
- What do we do?
  - Cache content and serve requests from multiple servers at the network edge (close to the user)
    - Reduce 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

How can we make content access more efficient?

Serving & Consuming Content

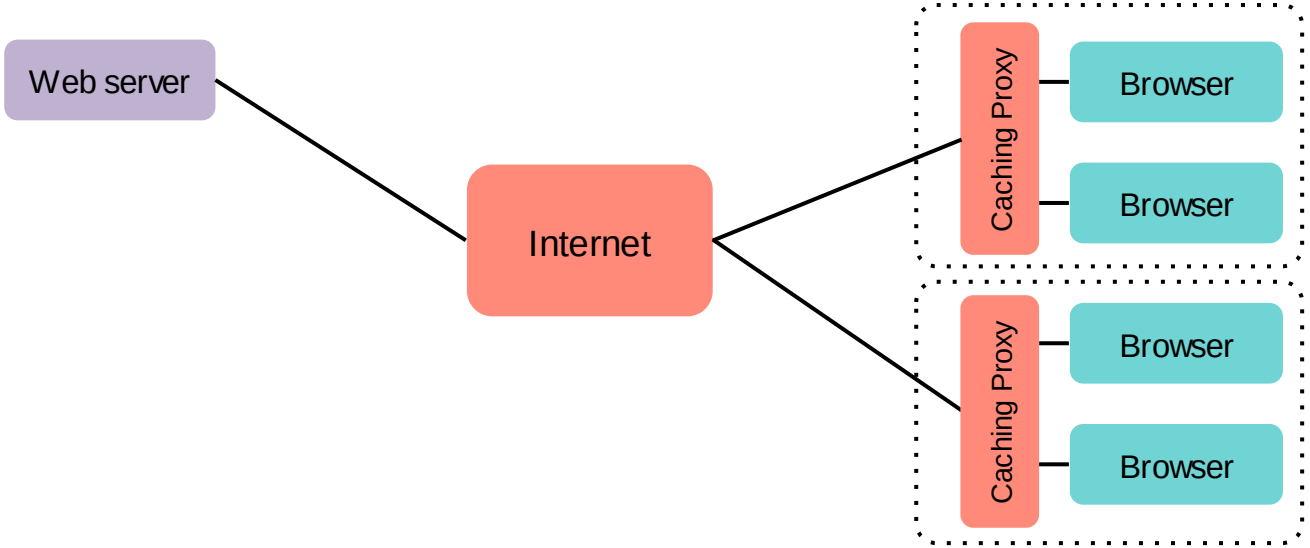


Every request goes to the server.

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

… but that cached data is local to the browser

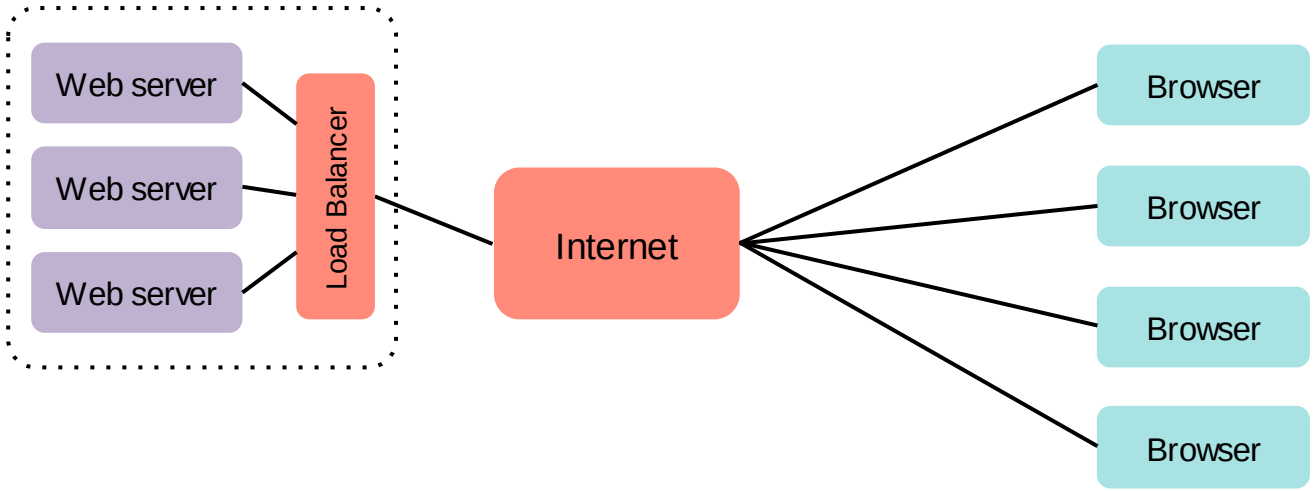
Caching Proxies



Caching proxy in an organization.

Take advantage of what others before you have recently accessed.

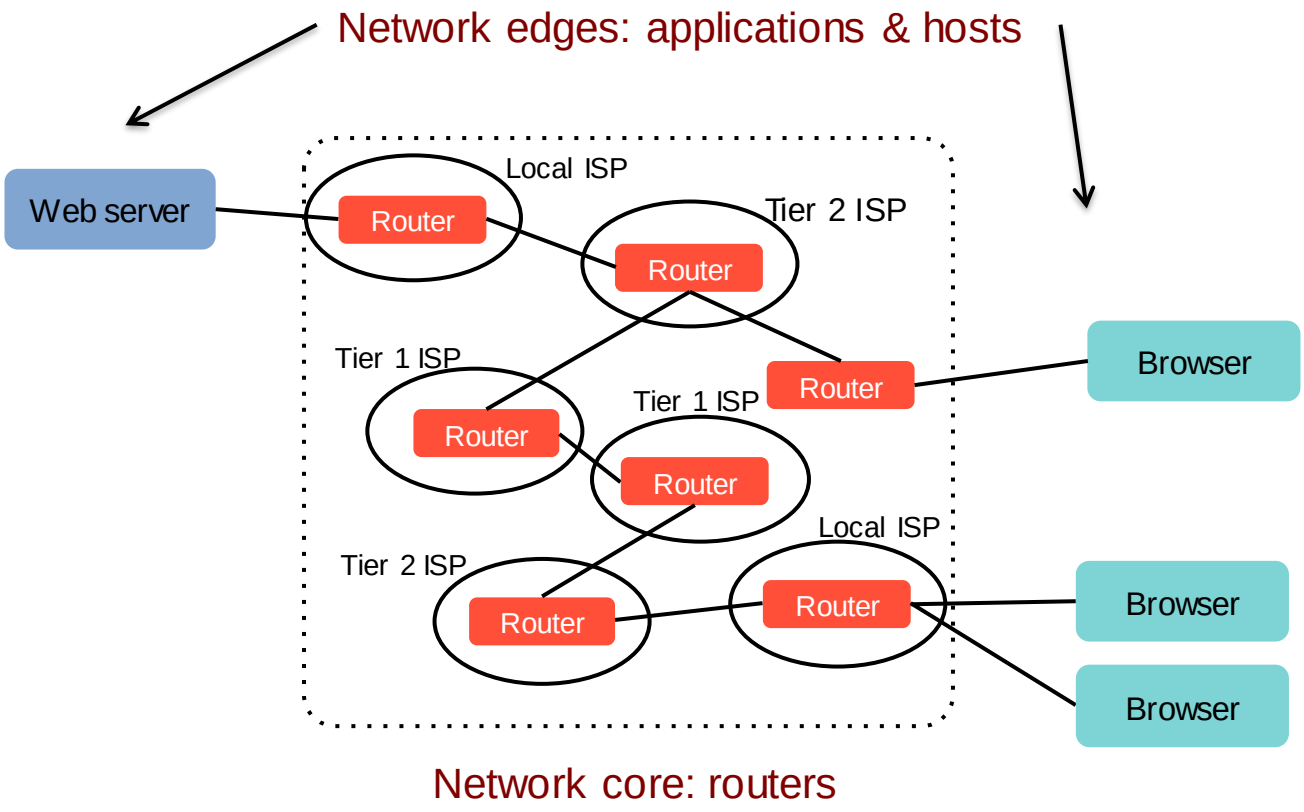
Load Balancing



Increase capacity at the server – multiple servers may serve content

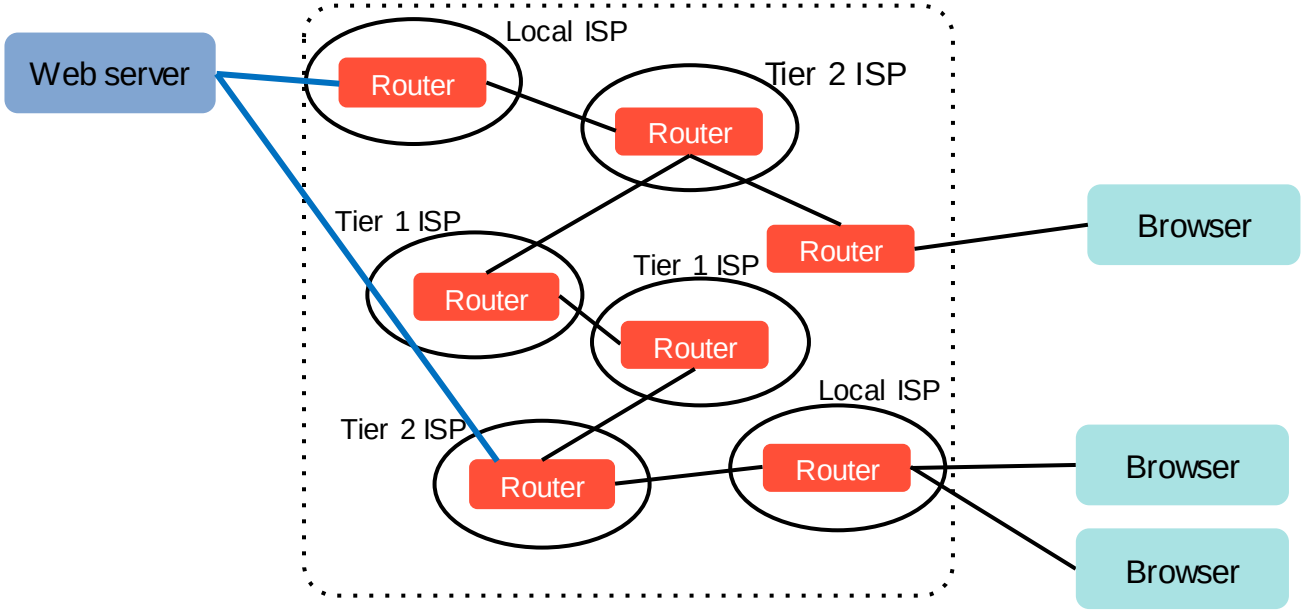
Internet connectivity can be a bottleneck … + latency from client to server.

Internet End-to-End Packet Delivery



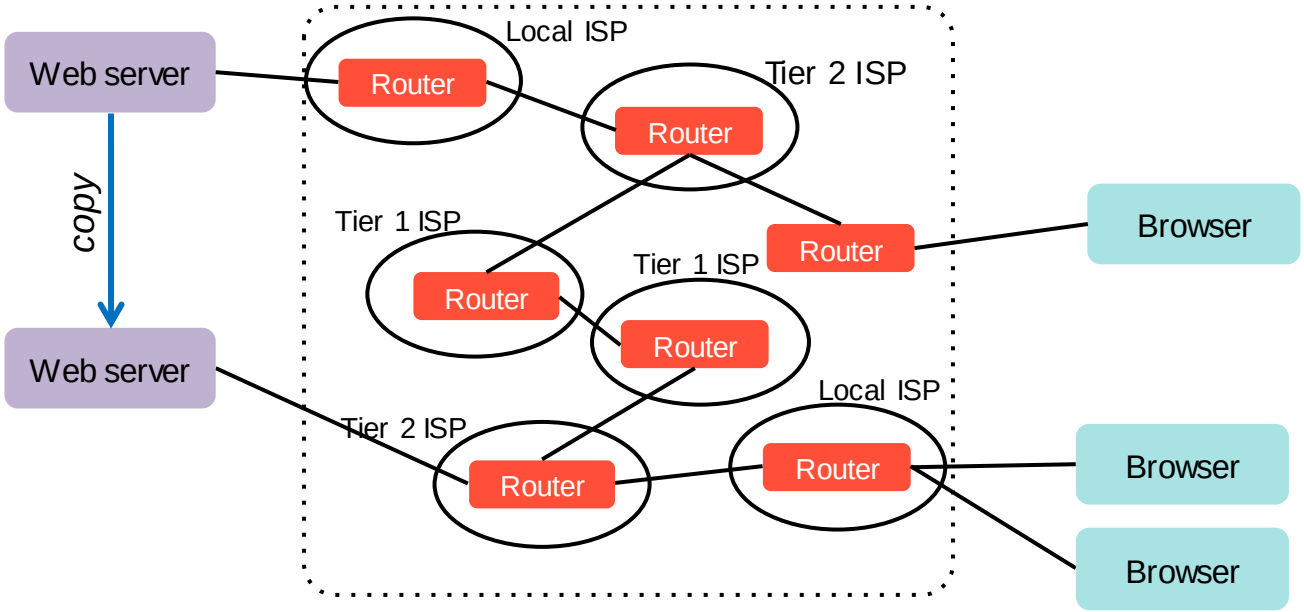
Multihoming

- 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)

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



Improving scalability, availability, & performance

- **Improve Scalability**
  - Mirror (replicate) servers for load balancing among multiple servers
  - Use multiple ISPs if network congestion is a concern
- **Improve Availability**
  - Replicate servers
  - Deploy across multiple data centers & ISPs
- **Improve Performance**
  - Cache & serve content requests from multiple servers throughout the network edge (close to the users)
    - Reduce demand on the site's infrastructure
    - Provide faster service to users
      - Content comes from nearby servers

But these approaches have challenges!

- **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

Content Delivery Networks

Akamai Distributed Caching

- Company evolved from MIT research
- "Invent a better way to deliver Internet content"
- Tackle the "flash crowd" problem
- Akamai runs on ~365,000 servers in ~1,600 networks across ~135 countries
  - Delivers 15-30% of all web traffic … reaching over 200 Terabits per second
  - Used by
    - 45 of the top 50 brokerages
    - 18 of the top 20 telecommunications carriers
    - All top 10 video streaming services
    - All top 10 video game companies
    - All top 10 banks
    - All top 10 software companies
    - All 6 U.S. military branches

    …

https://www.akamai.com/company/facts-figures

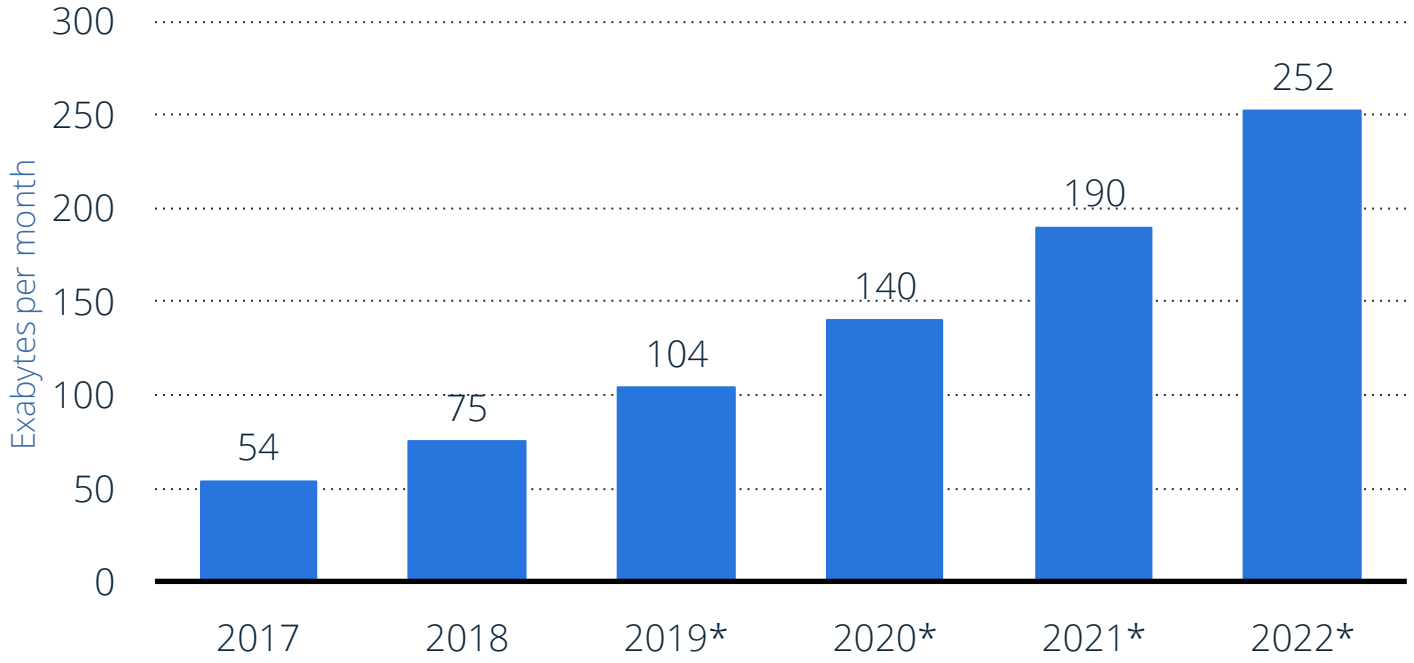
https://www.akamai.com/site/en/documents/akamai/overview-of-akamai-personal-data-processing-activities-and-role.pdf

Content Delivery as a Service

- **Some of the largest Internet companies** (e.g., Google, Microsoft, Amazon, Facebook, Apple) **run their own CDNs**
  - Redundant, globally-distributed data centers connected to many ISPs
- **For most companies, it doesn't make sense**
  - Huge capital expense
  - Huge operating costs
  - Capacity is not always needed, so most networks & servers will be underutilized
- **CDNs are a service**
  - Let someone else figure out scalable content delivery

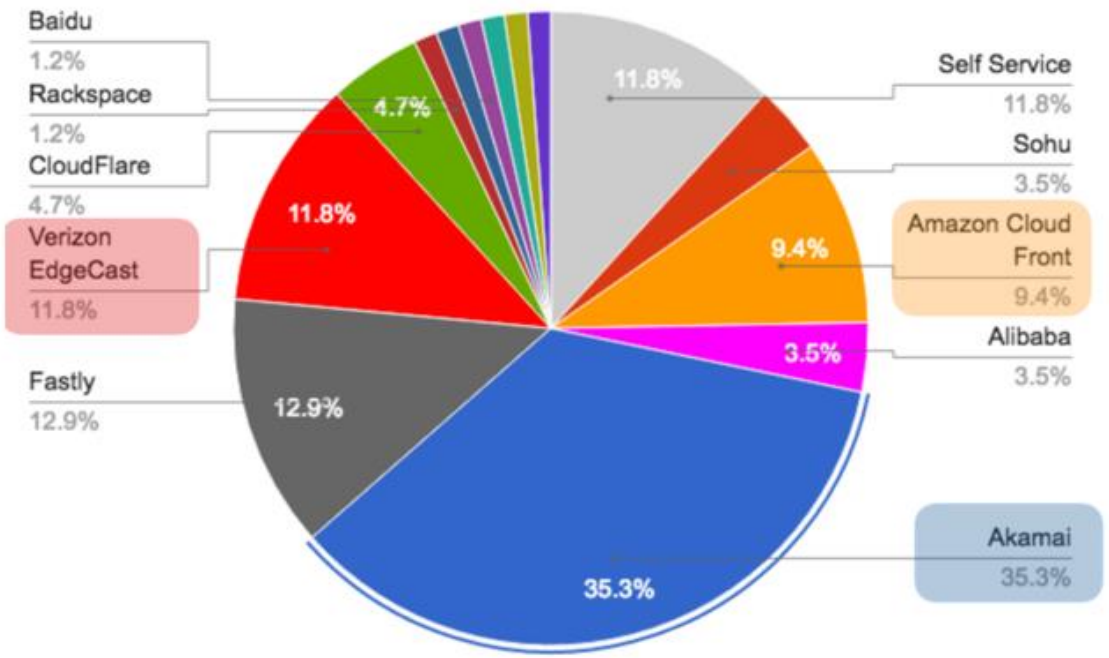
Content Delivery as a Service

Forecast: global content delivery network internet traffic 2017-2022
(in exabytes per month)



I couldn't find newer data with some quick web searching.
Further information regarding this statistic can be found on page 8.
Source(s): Cisco Systems; ID 267184

CDN Providers – Market Share

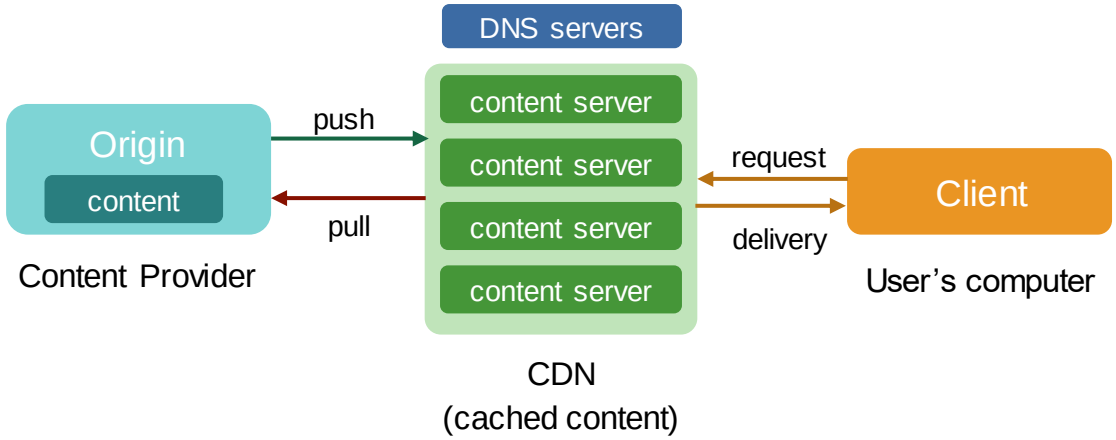


Akamai	35%
Fastly	13%
Verizon EdgeCast	12%
AWS CloudFront	10%
Self Service CDNs	12%
Cloudflare	5%

Source: T4 CDN Market Share – January 23, 2021 (a bit dated but the latest chart I could find)

https://www.t4.ai/industry/cdn-market-share

CDN Structure: Pushing & Pulling



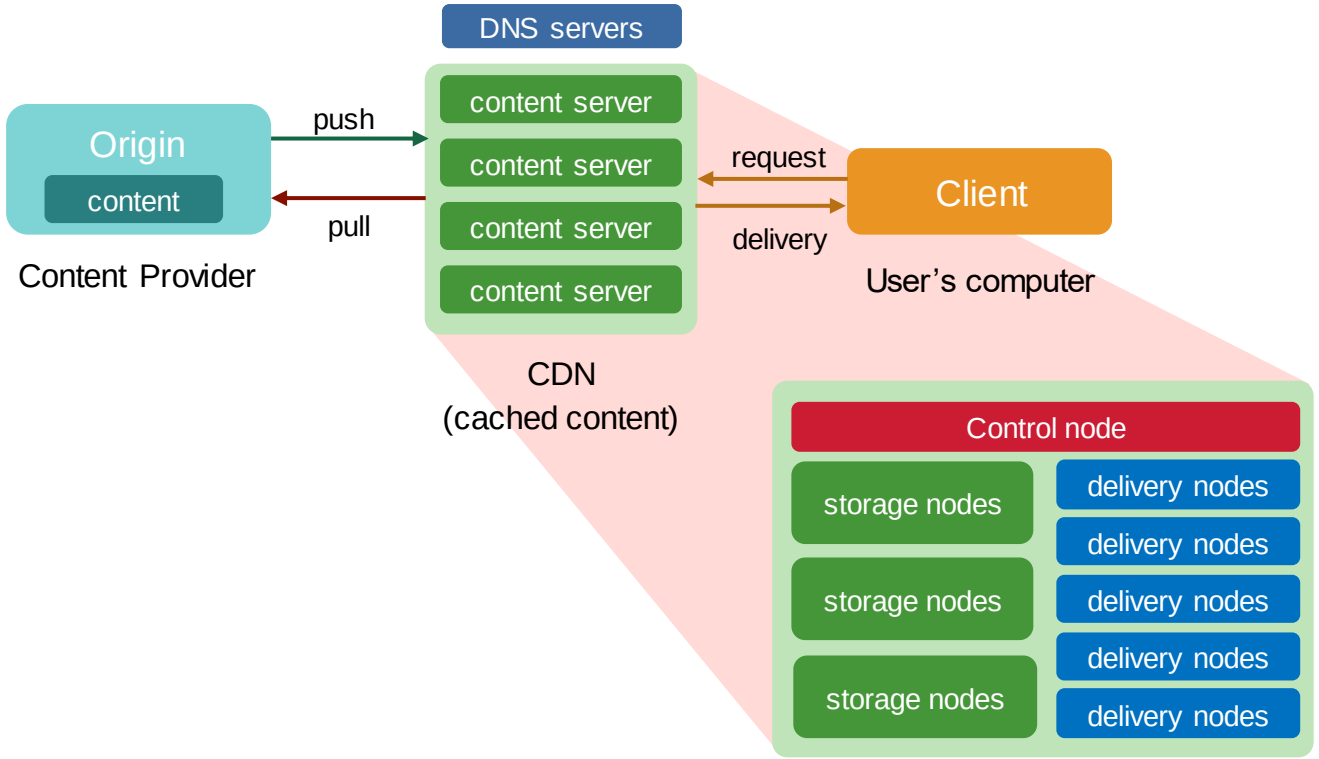
Push CDNs

– Origin must store content manually onto delivery nodes

Pull CDNs

– Delivery nodes request content from the origin

CDN Structure: Storage, Delivery & Control



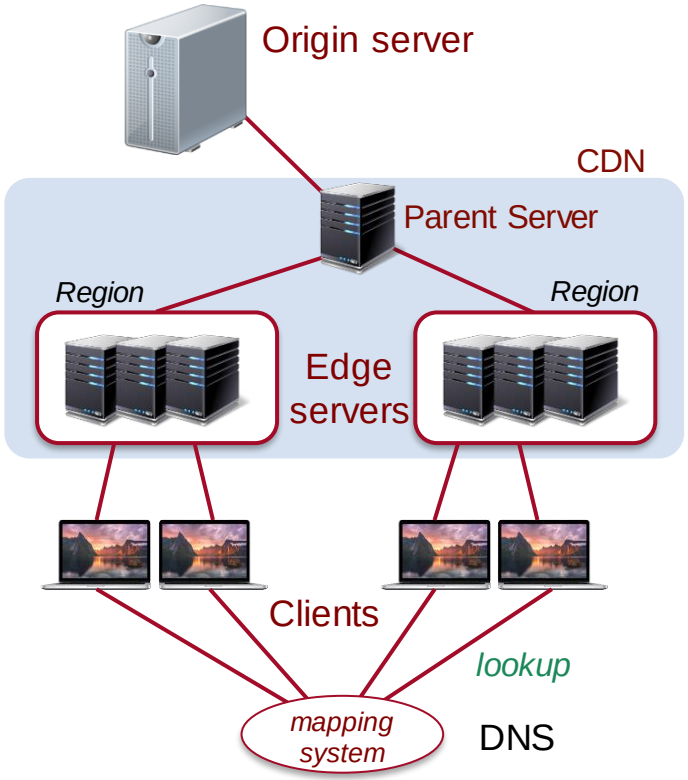
Akamai's goal (and CDNs in general)

Try to serve content to 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

Content lookup: (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 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



DNS Setup

The company's original content is hosted on its server = origin server

Edge servers in the CDN cache content and take place of the actual servers that host the site's content

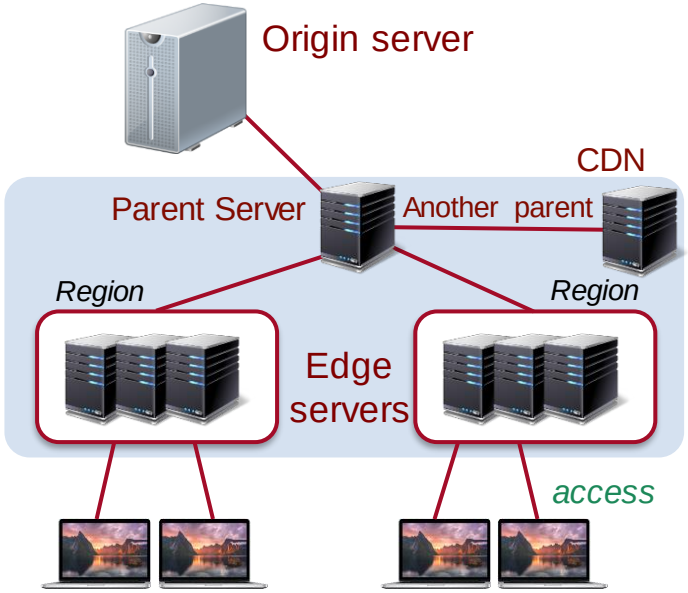
Example

`www.staples.com` is an alias (called a CNAME in DNS) that maps to Akamai's `www.staples.com.edgekey.net`:

```
www.staples.com.  86094     IN   CNAME     www.staples.com.edgekey.net.
www.staples.com.edgekey.net. 21294 IN CNAME     e6155.a.akamaiedge.net.
e6155.a.akamaiedge.net.     8     IN   A     184.87.69.100
```

Content lookup: (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
   - If the content is not found, broadcast the query to other edge servers in the region
   - If the content is not found, ask the parent server – this is tiered distribution and avoids adding more traffic to the origin
   - If the content is not found, the parent asks its peers (other parent servers)
   - Finally, contact the origin server via the transport system

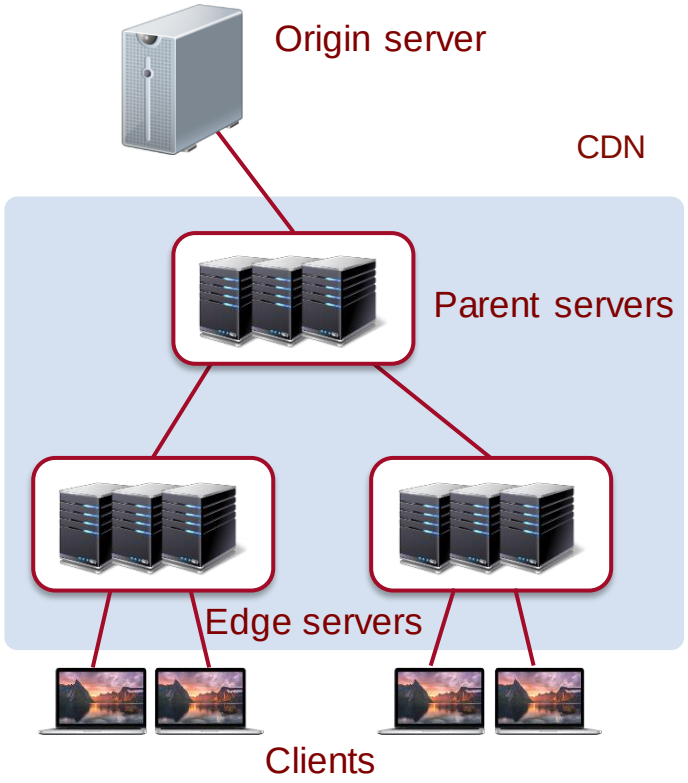


Benefits of a CDN

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: caching controls

The organization can control how specific content is cached in a Property Manager:

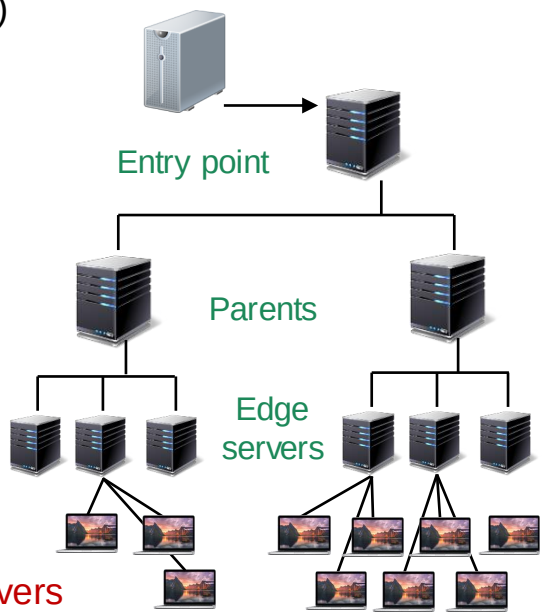
- Use HTTP headers
- Ignore HTTP headers and use a custom time-to-live
- Never cache

An HTTP `Cache-control` header can specify:

- `max-age`: specify how long a file can stay in the cache (seconds)
- `no-store`: don't cache – content that changes and shouldn't be cached
- `no-cache`: revalidate each request (e.g., check with origin via `if-modified-since`)
- `public`: content can be cached publicly for all requests
- `private`: only the user's browser is allowed to cache the content

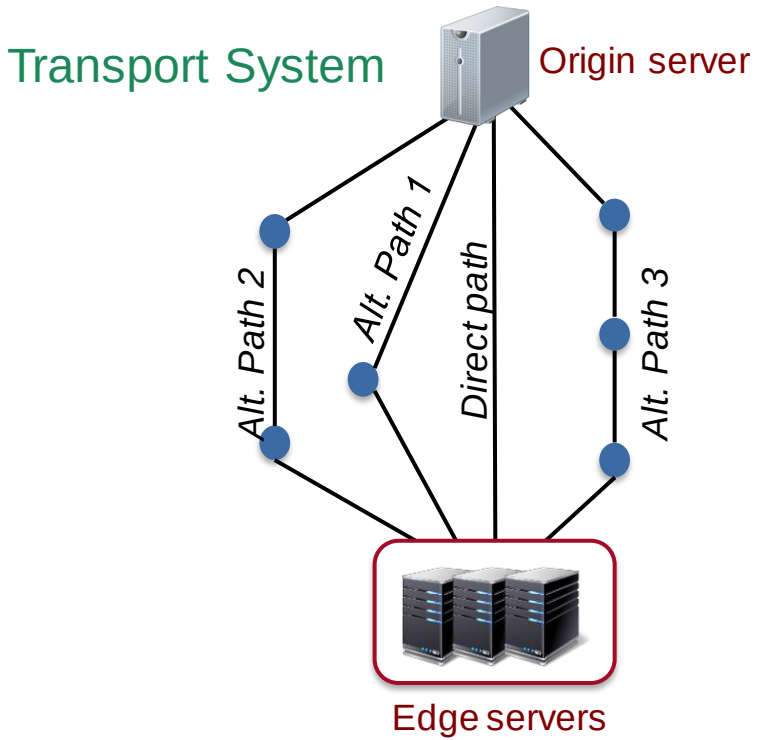
1. Caching: types of content

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



2. Routing: Overlay Network

The Internet is a collection of many autonomous networks

- Routing is based on business decisions
  - Peering agreements, not performance
- An ISP's 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 ISPs across many regions
  - All know about each other
  - Servers periodically test performance to origin servers
  - High-performance routing

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

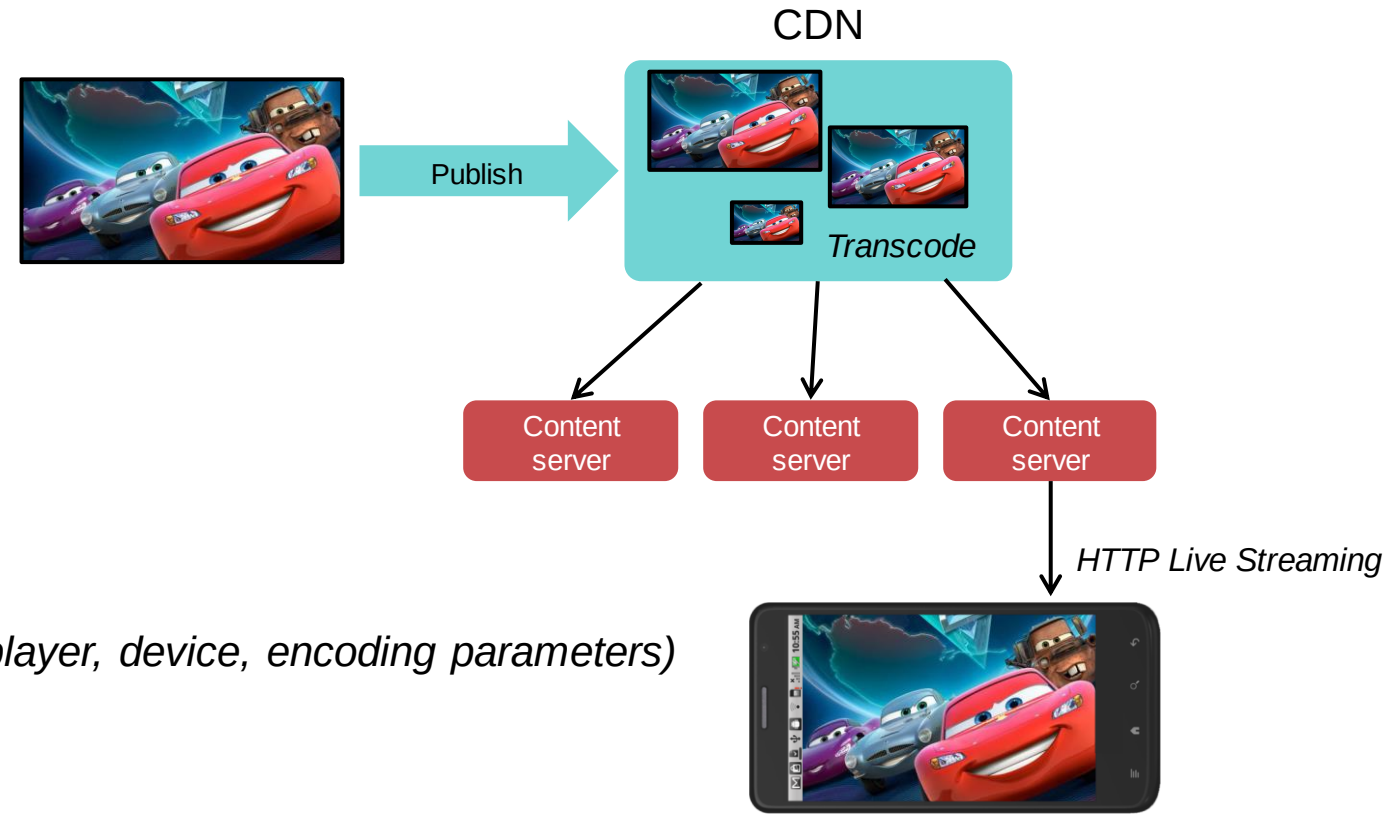
- Infrastructure on demand
  - CDN absorbs majority of content
- Instant worldwide scaling based on demand
- Business advantages

Video Streaming via CDNs

How is live video different?

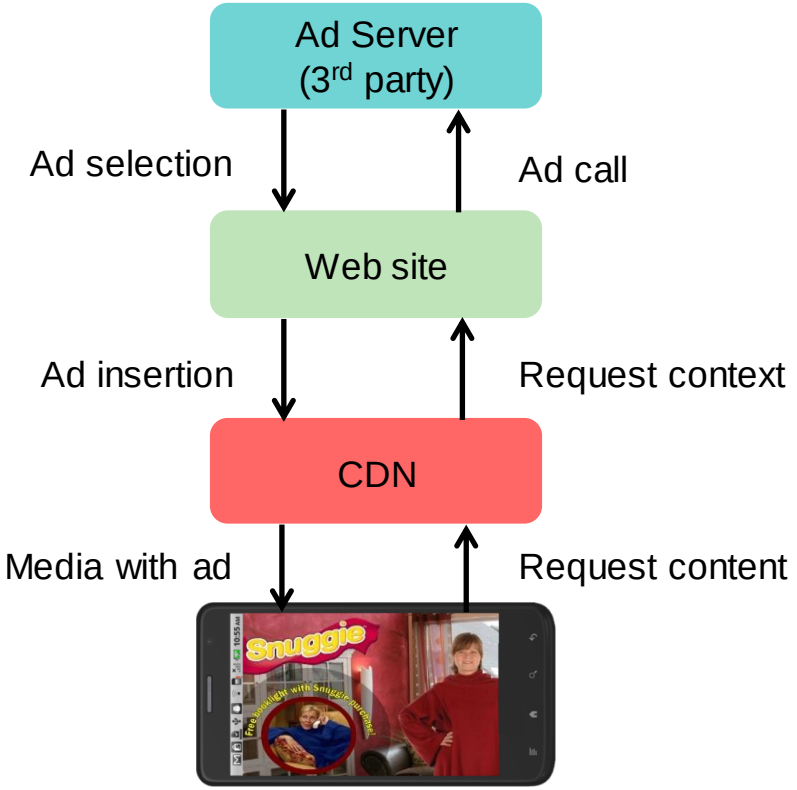
- Live video cannot be cached
  - Progressive downloads – watch video while downloading
  - vs. direct downloads – download first, watch later
- **HTTP Live Streaming** (HLS): most popular way to access video
  - Use generic HTTP servers
  - Deliver on-demand video just like any other content
- **Adaptive 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

Colors

- Text goes here – link – followed link
- Here is some callout text … and in blue
- Here is some green callout text

Link color

Followed Link color

Text Text Text Text Text

Text Text Text Text Text Text Text

Text Text Text Text Text Text Text Text

Background 2

Text 2