Distributed Systems

23. Content Delivery Networks (CDN)

Paul Krzyzanowski

Rutgers University

Fall 2016

Motivation

- Serving web content from one location presents problems
  - Scalability
  - Reliability
  - Performance

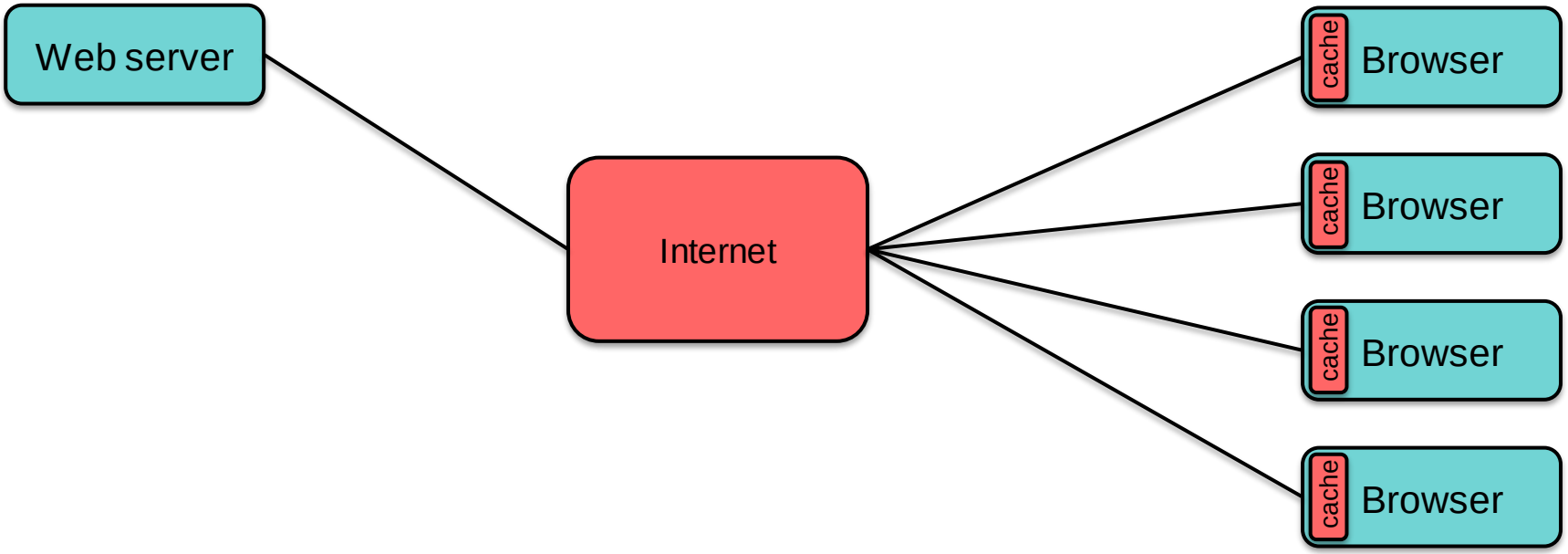
- "Flash crowd" problem
  - What if everyone comes to your site at once?

- 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

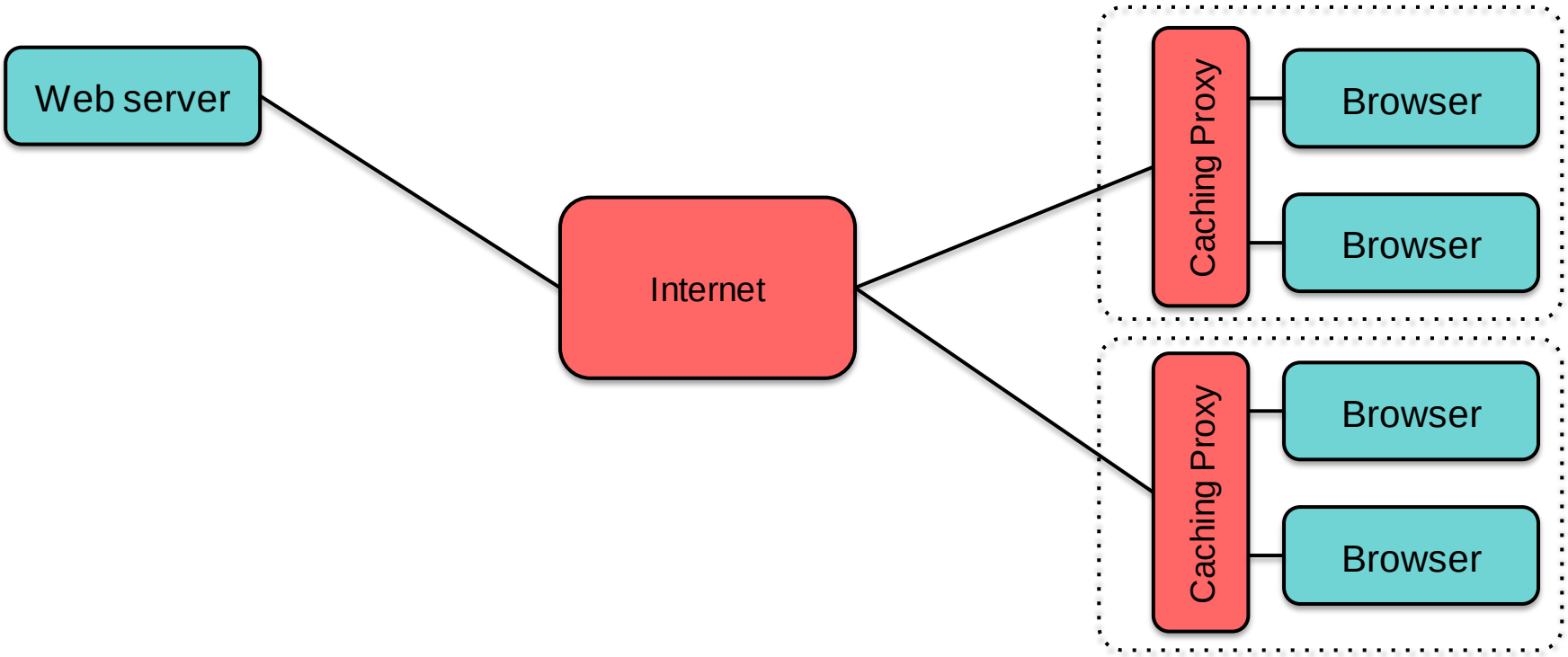


Every request goes to the server.

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

– but that cached data is local to the browser

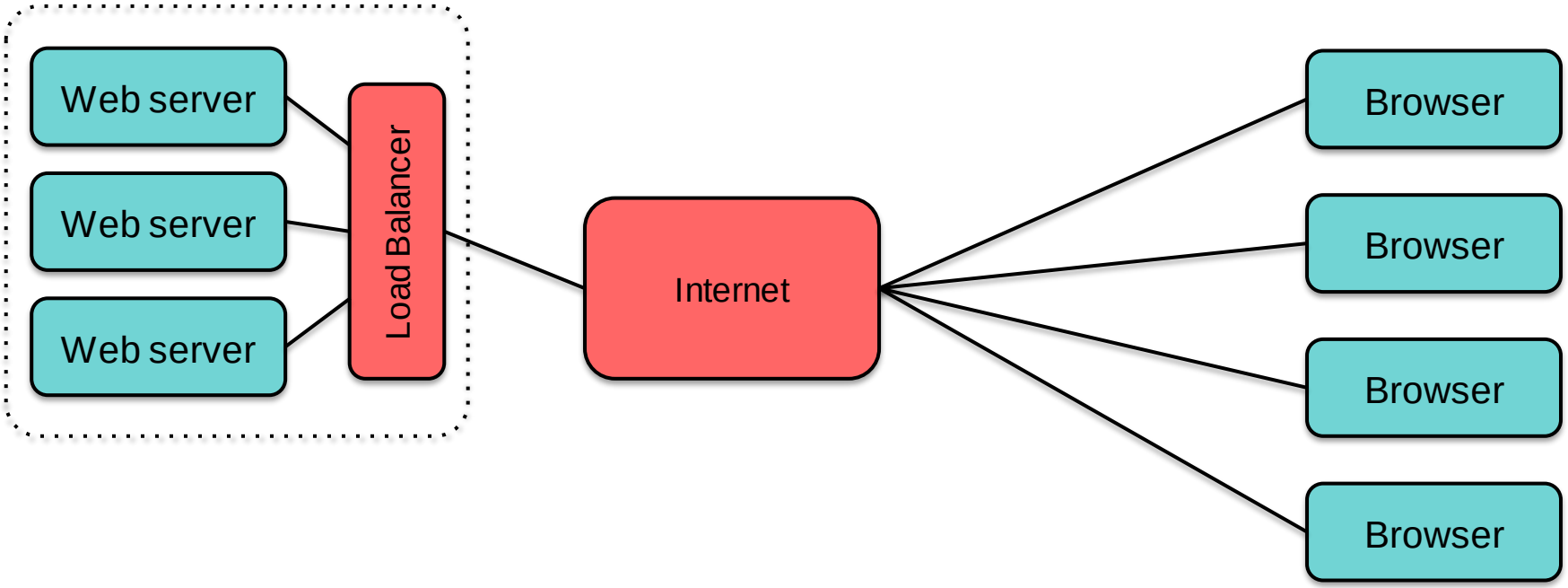
Caching Proxies



Caching proxy in an organization.

Take advantage of what others before you have recently accessed.

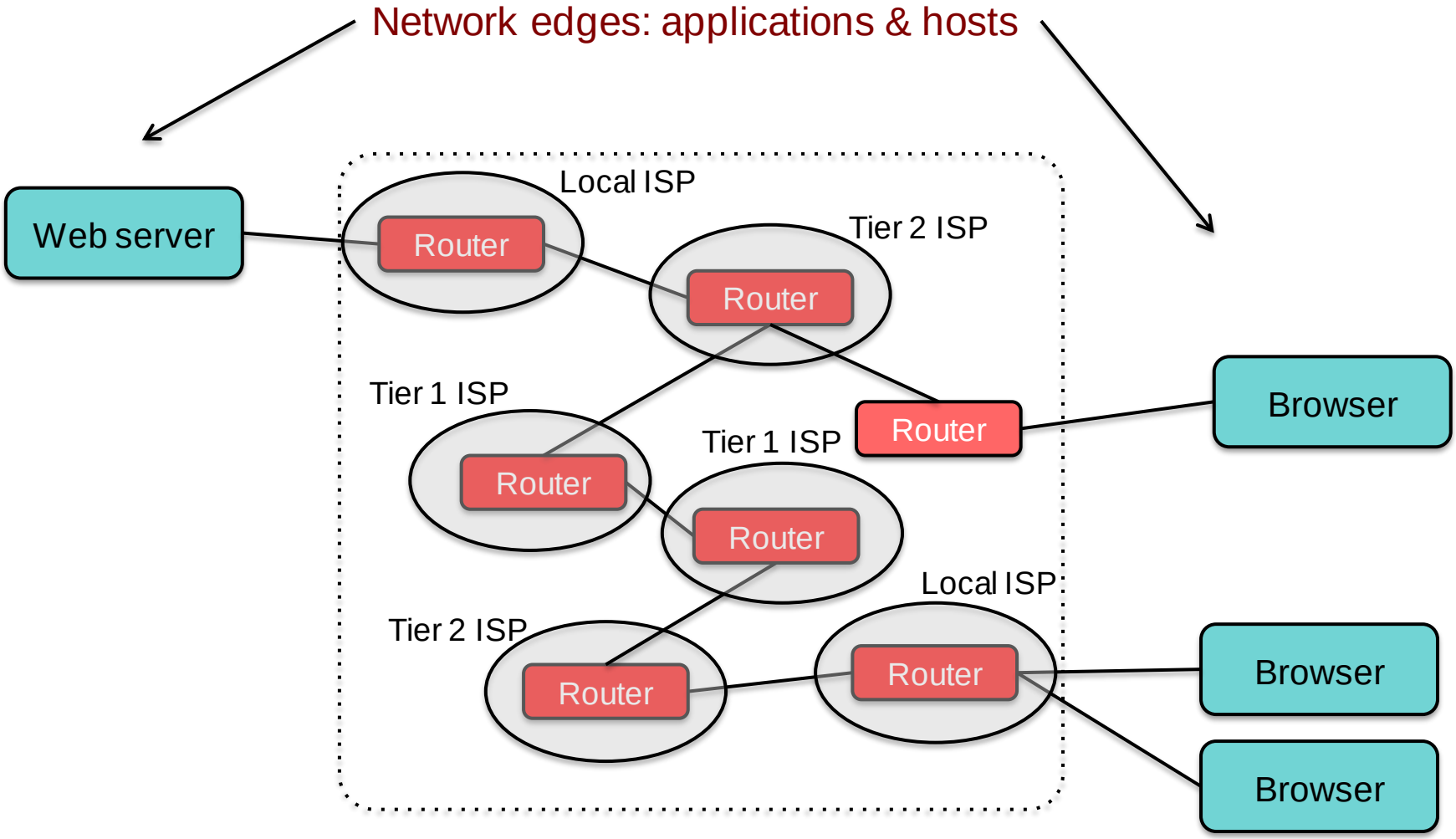
Load Balancing



Increase capacity at the server.

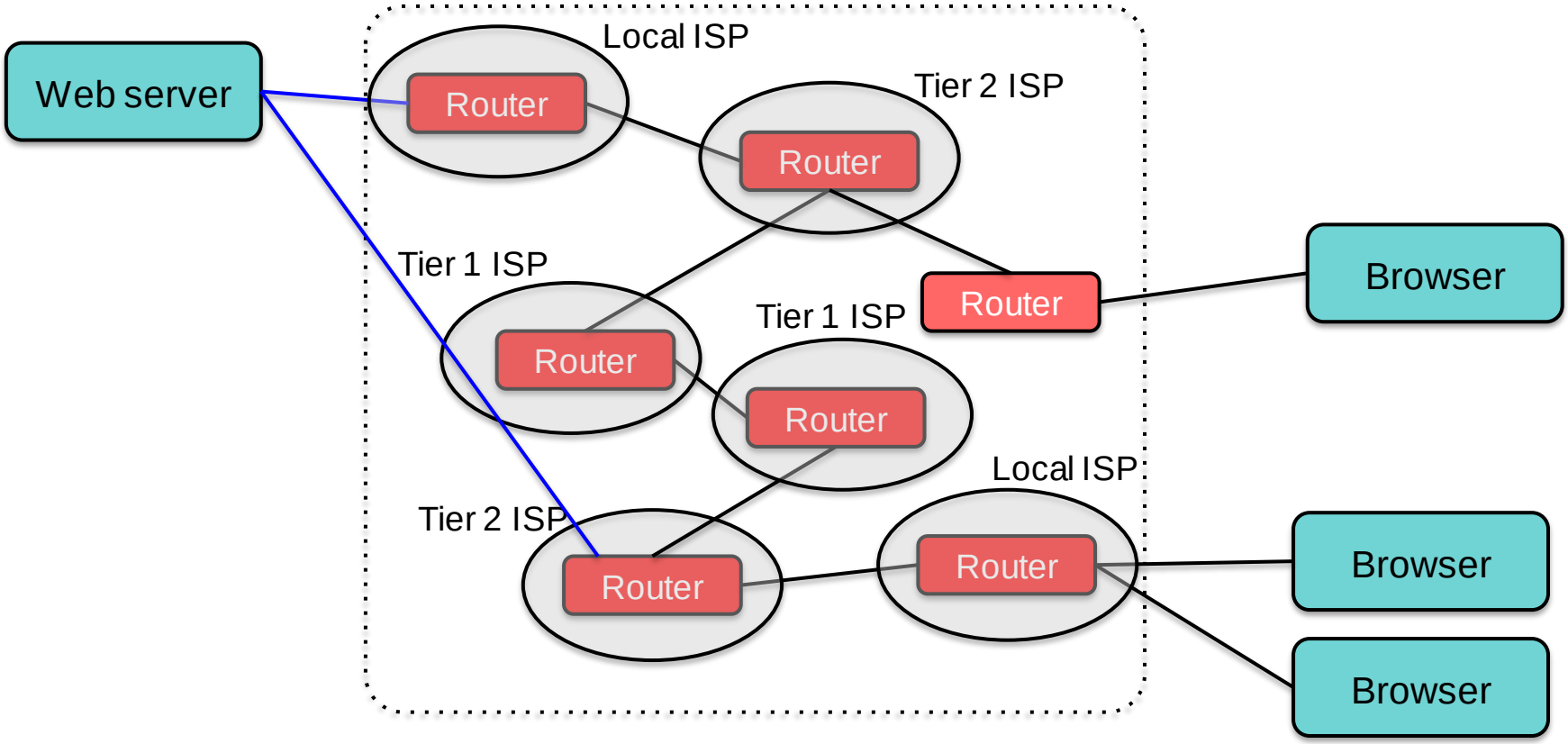
Internet connectivity can be a bottleneck … + latency from client to server.

Internet End-to-End Packet Delivery



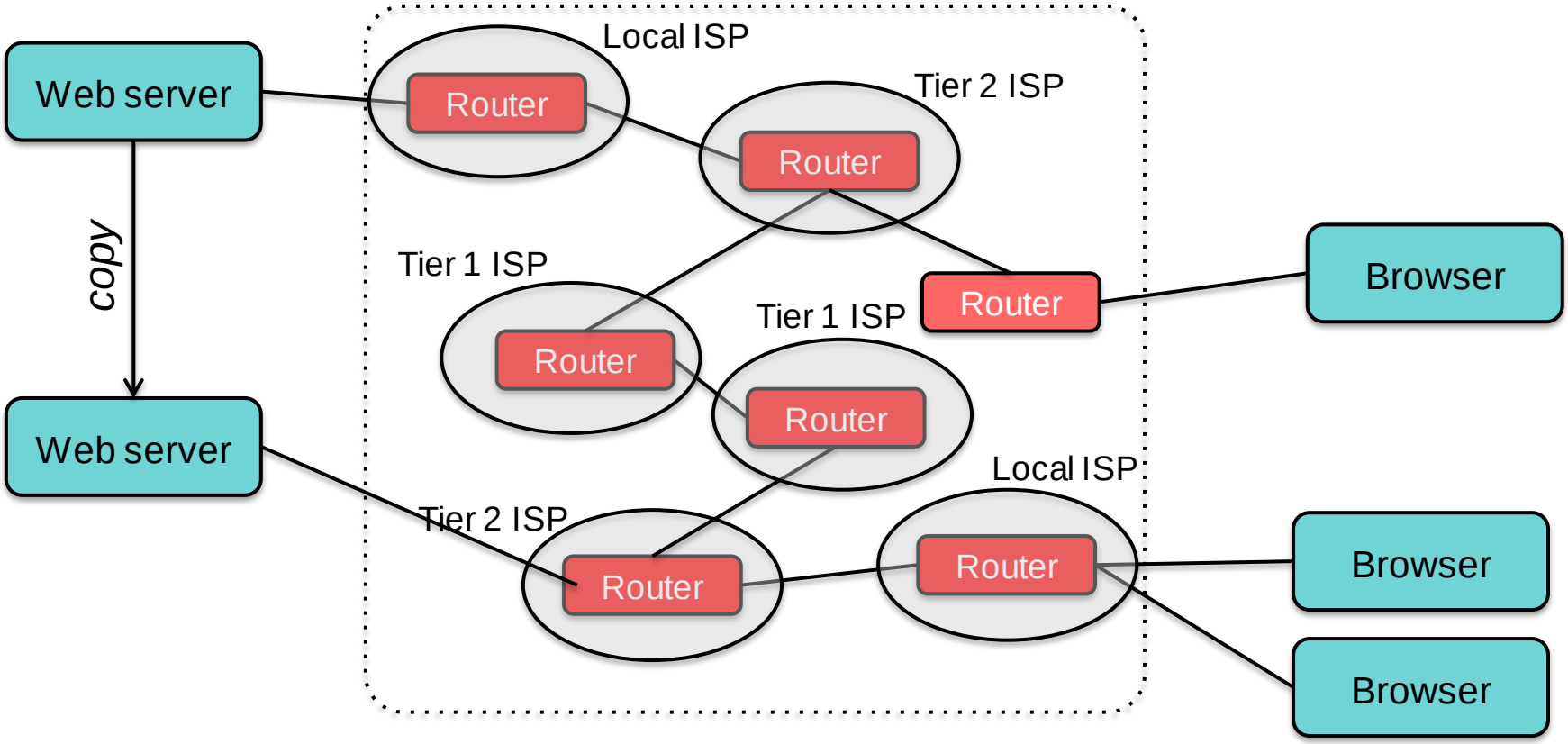
Multihoming

- 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**
  - Load balance among multiple servers
  - Multiple ISPs if network congestion is a concern
- **Availability**
  - Replicate servers
  - Multiple data centers & ISPs
- **Performance**
  - Mirror (replicate) servers for load balancing
  - 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 216,000 servers in 1,500 networks across 120 countries
  - Delivers 15-30% of all web traffic

    … reaching over 30 Terabits per second

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

Akamai's goal

Try to serve clients from servers likely to have the content

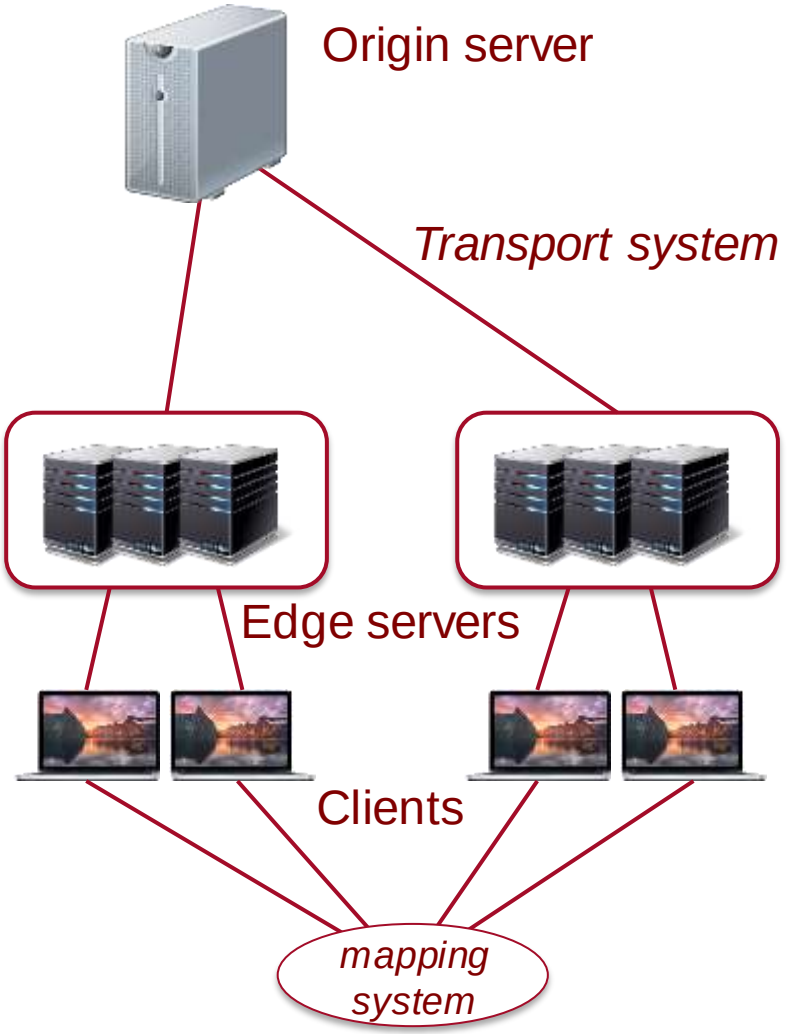
- **Nearest**: lowest round-trip time
- **Available**: server that is not too loaded
- **Likely**: server that is likely to have the data

Overlay Network

- The Internet is a collection of many autonomous networks
  - Connectivity 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
- The Overlay network
  - Collection of caching servers at many, many ISPs
  - All know about each other

Overlay Network

1. Domain name lookup
   - Translated by mapping system to an edge server that can serve the content
   - Use **custom DNS servers**
     - Take requestor's address into into account to find the *nearest* edge
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



Mapping: Domain Name Lookup

- Akamai uses **Dynamic DNS servers**
- 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

Akamai collects network performance data

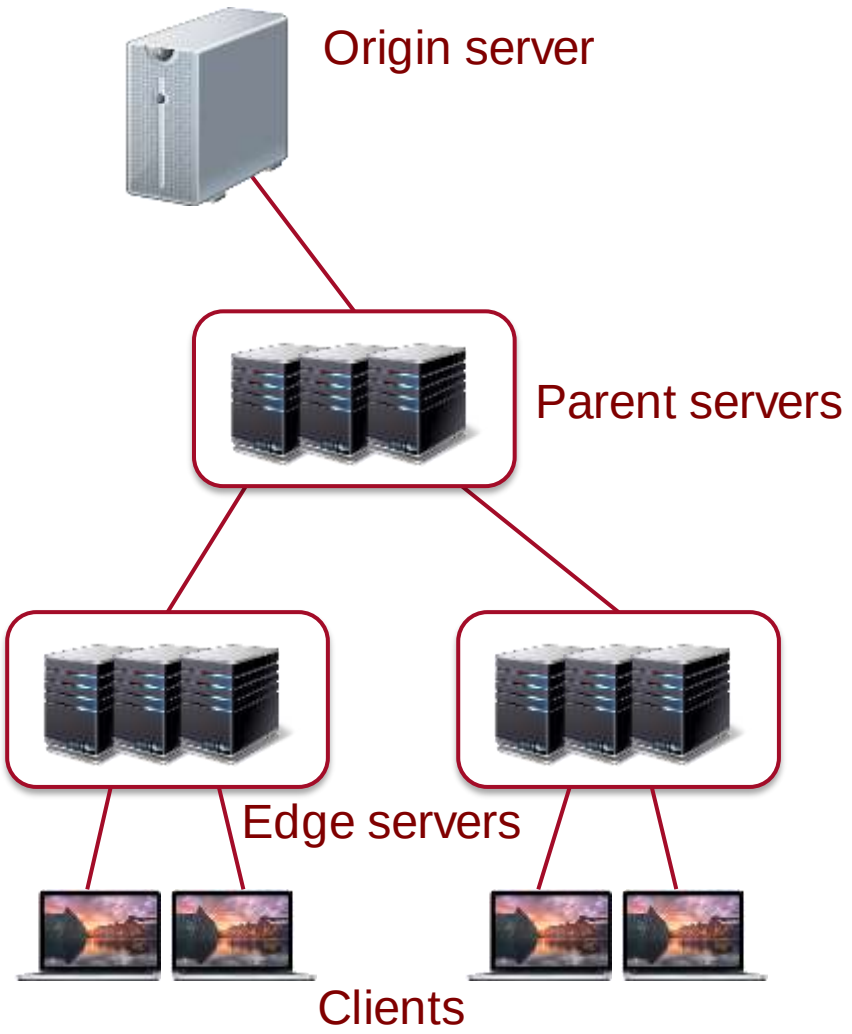
- Map network topology
  - Based on BGP and *traceroute* information
  - Estimate hops and transit time
- Content servers report their load to a monitoring application
- Monitoring app publishes load reports to a local (Akamai) DNS server
- Akamai DNS server determines which IP addresses to return when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

Benefits of an overlay network CDN

1. Caching
2. Routing
3. Security

Caching

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

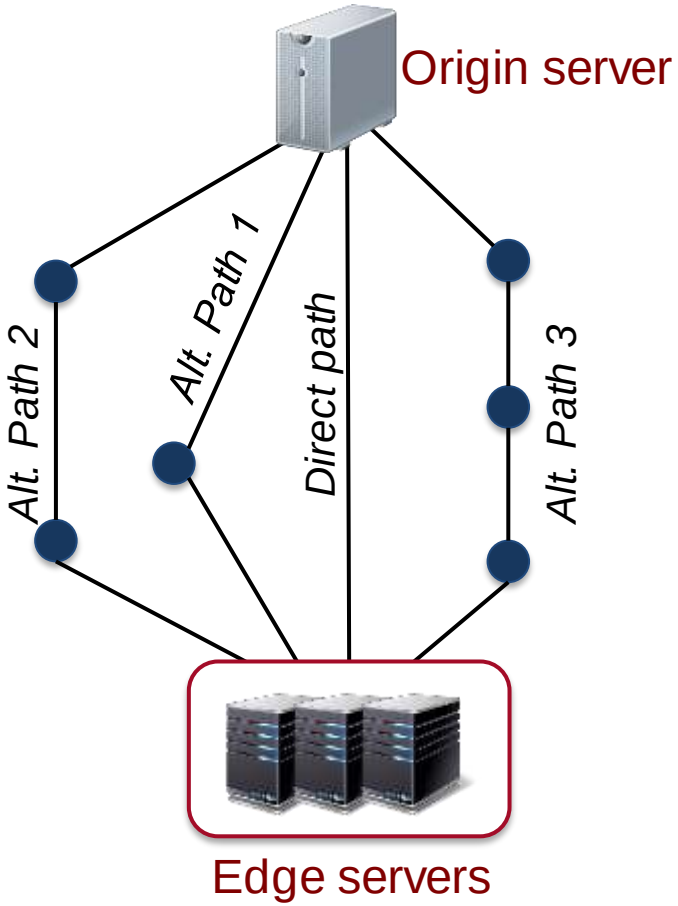


Types of content

- 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 Akamai network
  - Stream is delivered from the entry-point server to multiple edge servers
  - Edge servers serve content to end users.

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



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

Other Things CDNs Do

Signed URLs in Amazon CloudFront

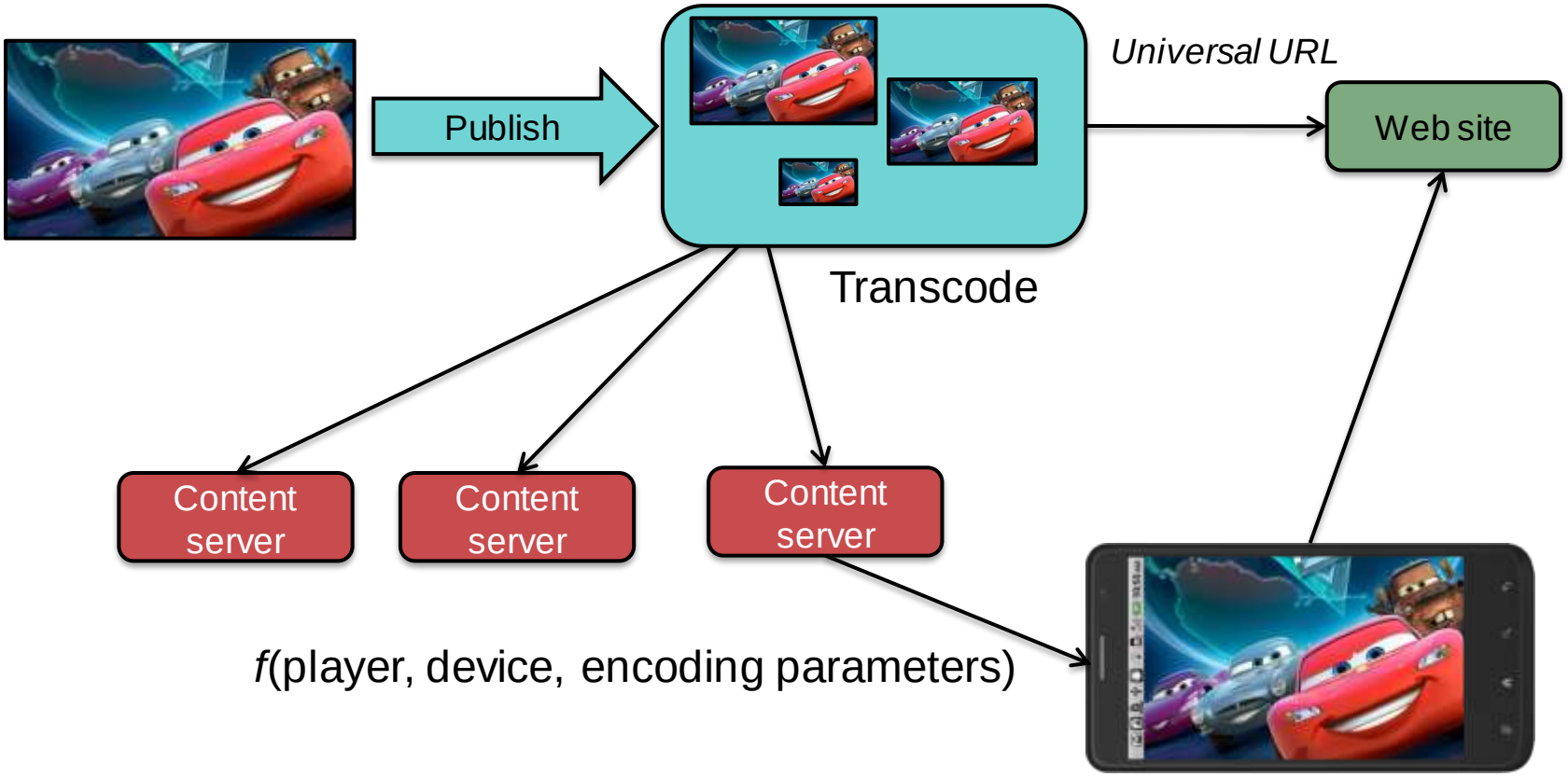
- Example: Amazon CloudFront CDN
  - Similar in concept to Akamai
  - Requests for content are routed to the nearest edge location
    - Cached content with original located at *origin servers*
  - Integrates with back-end Amazon services

- Private content: provide special URLs for restricted content
  - Control access to content via a signed URL
  - URL contains:
    - policy or a reference to a policy
    - Signature = encrypted hash
      - URL cannot be modified
  - Policies include:
    - Validity: start time & expiration time
    - Range of IP addresses that are allowed to access the object

Limelight Orchestrate™

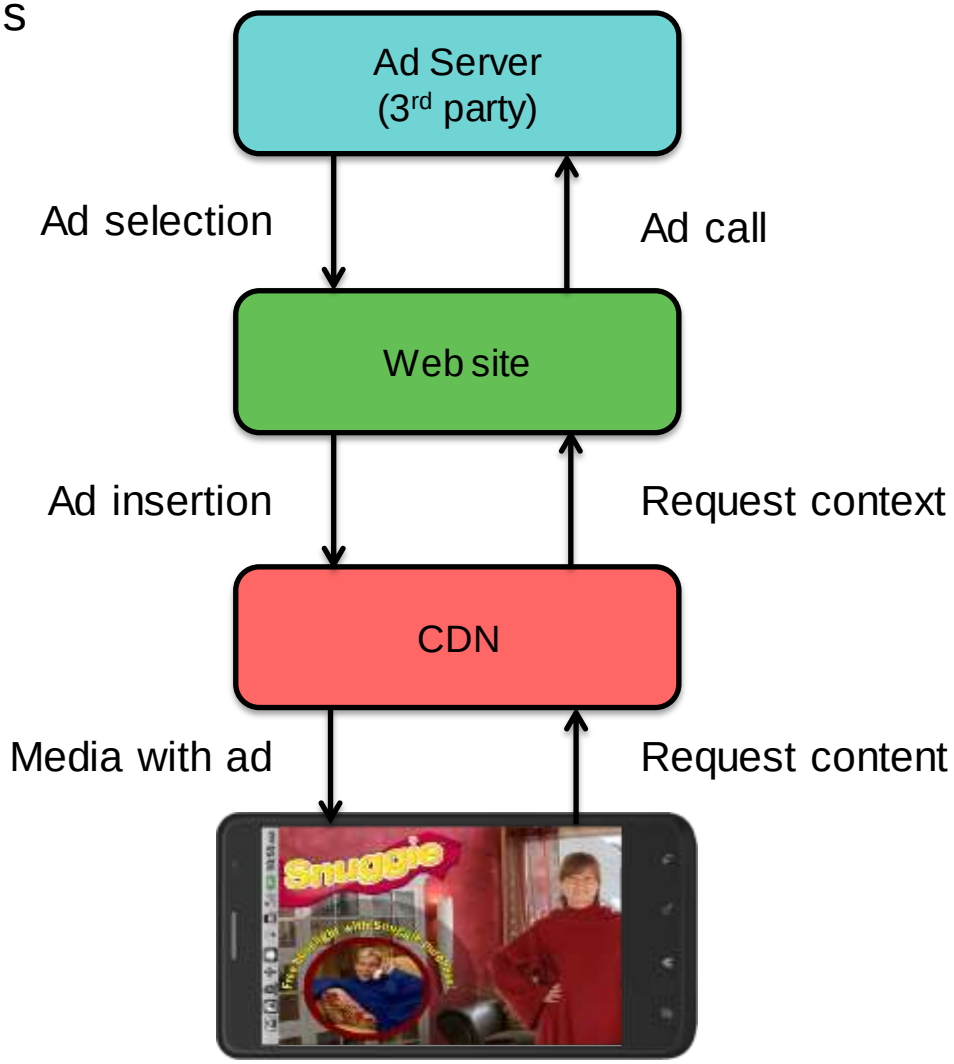
- Focus on video distribution and content management
- Video transcoding
  - Encode video to a variety of formats
  - Support playback on various devices: different formats & bitrates
- Ad insertion
  - Integrate with ad servers (DoubleClick, LiveRail, Tremor, YuMe)
  - Pre-roll, post-roll, mid-roll, overlay, etc.

LimeLight Orchestrate™ Transcoding



Server-side Video Ad Insertion

Example: Limelight Reach Ads



The End