How do we deliver content to users?

- Many types of content: web pages, images, videos, tweets, social network posts, etc.
- Content originates from
  - Organizations (e.g., NYTime, Netflix) -- called content providers
  - Users (e.g., Twitter)
- **Where can content be stored?**
  - Private data center
  - Public cloud
  - User’s device
  - In-network cache
- **What metrics matter to users?**
  - Wait-time -- e.g., load time for Facebook, buffer time for video
  - Quality -- e.g., video resolution
- **What network metrics influence this?**
  - Latency
  - Bandwidth
  - Jitter
- **Where can bottlenecks occur?**
  - First mile -- connection from client to ISP; DSL, cable, 3G
  - Last mile -- connection from server to ISP; 1G or 10G link, possibly shared (in a cloud), may not be full-bisection bandwidth
  - Servers -- finite amount of resources to serve requests; can add more servers and use a load balancer to divide requests among multiple servers
  - Connections between ISPs (or ASes) -- high-bandwidth within an ISPs network, but limited bandwidth between ISPs
- Bottlenecks can become even worse when content is popular
  - Studies show content follows a zipf or power-law distribution -- most popular content has several magnitudes more accesses than less popular content
  - Flash crowd -- phenomenon where many users try to access the same content at the same time
    - E.g., picture Ellen Tweeted from the Oscars
    - System can quickly become overwhelmed -- even if a lot of effort has gone into making the system scale quickly
**What are the (dis)advantages of storing/serving content from specific locations?**

**How do we avoid the disadvantages?**

- Distribute content so that:
  - They don’t all share the same last mile link
  - Any client is only a few AS hops away from a server
  - Short geographic distance is also beneficial to reduce propagation delay

**Proxy Caches**

- **Forward proxy** -- cache close to the client
  - Controlled by client-side AS
  - Explicit proxy -- must configure browser
  - Implicit (or transparent) proxy -- proxy is deployed on the path taken by traffic and requests are intercepted

- **Reverse proxy** -- cache close to the server
  - Used by Google

**Limitations of caching**

- Lots of content is not cacheable
  - Dynamic data -- e.g., stock prices, scores, webcams
  - Dynamic pages -- e.g., results depend on parameters
  - Encrypted content -- cannot decrypt requests or replies

- Limited analytics -- content owner wants to measure accesses
- Overhead of refreshing cached data
Content Distribution Networks

- Network of servers designed specifically for distributing content
  - Many groups of servers distributed throughout the world (or throughout the portion of the Internet where clients are located)
  - Popular content **proactively** replicated on one or more servers in each group — proxy caches use **reactive** caching
  - Peering arrangements with many ISPs (or ASes)
  - Mechanism for clients to find the “best” server to contact

- Difficult for every company that wants to provide content to build their own content distribution network
  - If you are big enough, e.g., Google, you can build a CDN
  - Otherwise, you contract with a company that has built a CDN, e.g., Akami or EdgeCast, and they distribute content for you
  - To upload content, clients still need to connect to a server in a more “centralized” data center

- How do you pick the “best” server (i.e., node) to contact?

**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, ...**

**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

- **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

- **Naming**
  - DNS-based server selection
- **Advantages**
  - Avoid TCP set-up delay
  - DNS caching reduces overhead
  - Relatively fine control
- **Disadvantages**
  - Based on IP address of local DNS server
  - “Hidden load” effect
  - DNS TTL limits adaptation

- Who uses CDNs?
  - Netflix
    - Login page & movie selection pages come from a VM in EC2 — easter egg! bottom of Netflix page contains an ID for the VM in EC2 that provided page
    - Movie is delivered from CDN nodes
  - Pinterest — study this in assignment #4