Chapter 15

Content Delivery Networks (CDNs)

1. What is a CDN?

- **Definition:** A CDN is an *overlay network* of geographically distributed *caching servers* (which are reverse proxies). It's architected around the design limitations of the internet's network protocols.
- **How it Works:**
  - Clients using a CDN hit URLs that resolve to the CDN's caching servers.
  - When a CDN server receives a request, it checks its local cache for the resource.
  - **Cache Miss:** If not found locally, the CDN server *transparently fetches* it from the origin server (e.g., your application server) using the original URL.
  - The response from the origin is then cached locally on the CDN server and returned to the client.
- **Examples:** Well-known CDN services include *AWS CloudFront* and *Akamai*.

2. The Overlay Network: The Core Benefit

- While caching is a benefit, the *underlying network substrate* of the CDN is often considered its main advantage.
- **Limitations of the Public Internet:**
  - Composed of thousands of networks.
  - Its core routing protocol, *BGP (Border Gateway Protocol)*, was not primarily designed for performance.

  - BGP typically uses the *number of hops* to determine path cost, without adequately considering actual latencies or network congestion.
- **CDN as an Optimized Network:**
  - A CDN is an *overlay network* built on top of the public internet.
  - It employs various techniques to *reduce response times* for network requests and *increase the bandwidth* of data transfers.

A. Addressing Latency & Reliability

- Minimizing latency between client and server is critical (as discussed in Chapter 2 regarding TCP).
- Physical limitations like the *speed of light* mean long distances inherently cause high latency (e.g., >100ms across the world).
- Long-distance data transfer over the public internet also faces *increased error rates.*
- **Solution - Proximity:**
  - CDN clusters are placed in *multiple geographical locations* to be physically closer to clients.
  - **Client-to-Cluster Routing (How clients find the nearest cluster):**
    * *Global DNS load balancing*: An extension to DNS that considers the client's location (inferred from its IP address).
    * It returns a list of geographically closest CDN clusters, also taking into account current *network congestion* and the *health of the clusters.*

B. Optimized Routing & Connections

- **Strategic Placement:** CDN servers are often located at *Internet Exchange Points (IXPs)*, where different Internet Service Providers (ISPs) connect their networks.
  - This means a significant portion of the communication (from origin server to clients) flows over links controlled by the CDN.
  - The "first mile" and "last mile" hops (at either end) tend to have low latencies due to short physical distances.
- **Advanced Routing Algorithms:**
  - The CDN's overlay network uses routing algorithms optimized to select paths with *reduced latencies and congestion.*
  - These decisions are based on *continuously updated data* about the network's health.
- **TCP Optimizations:**

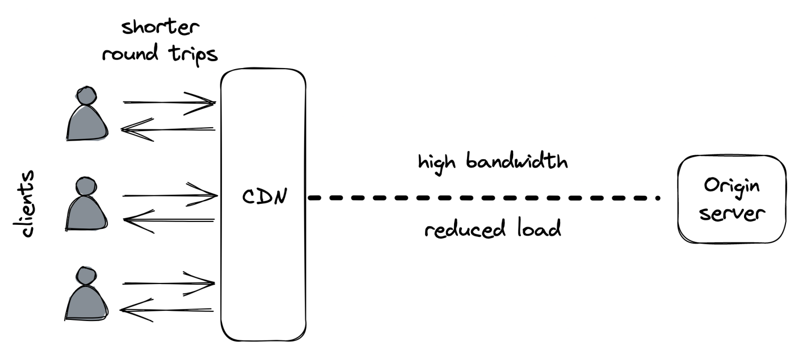

Figure 1: A CDN reduces the round trip time of network calls for clients and the load for the origin server.

  - CDNs exploit TCP optimizations where possible, such as: - Using *pools of persistent connections* between their servers to avoid the overhead of repeatedly setting up new TCP connections. - Employing *optimal TCP* *window sizes* to maximize effective bandwidth.

C. Beyond Static Content Caching

- The overlay network can also be used to *speed up the delivery of dynamic resources* that cannot be cached.
- **Security Role:** When used in this capacity, the CDN effectively becomes the frontend for the application, helping to *shield it against Distributed Denial-of-Service (DDoS) attacks.*

3. Caching Layers within a CDN

- A CDN can employ *multiple layers of content caching.*

A. Edge Clusters

- These form the *top layer* and are deployed at various geographical locations, close to end-users.
- **Handling Infrequently Accessed Content:**
  - If content is not available at an edge server (a cache miss at the edge), the edge server must fetch it from the origin server.
  - This fetching process benefits from the CDN's *efficient and reliable overlay network*, making it better than a direct fetch over the public internet.

B. Trade-off: Number of Edge Clusters vs. Cache Hit Ratio

- **Cache Hit Ratio:** The likelihood of finding a requested object in the cache.
- **The Trade-off:**

  - A *higher number of edge clusters* allows the CDN to serve more geographically dispersed clients effectively (lower latency).
  - However, it can lead to a *lower cache hit ratio* because requests are spread across more caches, each holding a smaller portion of the total content. This, in turn, can increase the load on the origin server.

C. Intermediary Caching Clusters (Regional Caches)

- To mitigate the above trade-off, CDNs can use one or more *intermediary caching clusters* (sometimes called regional caches).
- These are deployed in a *smaller number of geographical locations* compared to edge clusters.
- They are designed to *cache a larger fraction* of the original content than any single edge server.
- Edge servers can fetch content from these regional caches instead of going directly to the origin for certain misses.

D. Intra-Cluster Content Partitioning

- Within any single CDN cluster (edge or regional), the content is *partitioned among multiple servers.*
- This is necessary because *no single server* would typically be able to hold all the data the CDN might cache.
- Each server in the cluster is responsible for serving only a *specific subset* of the cached content.
- *(Data partitioning is a core scalability pattern discussed further in Chapter 16 of the PDF).*