What is a Content Delivery Network?

The thing that accelerates the internet

ASHISH PRATAP SINGH

MAR 04, 2025

♡ 125 💬 5 🔁 10 Share

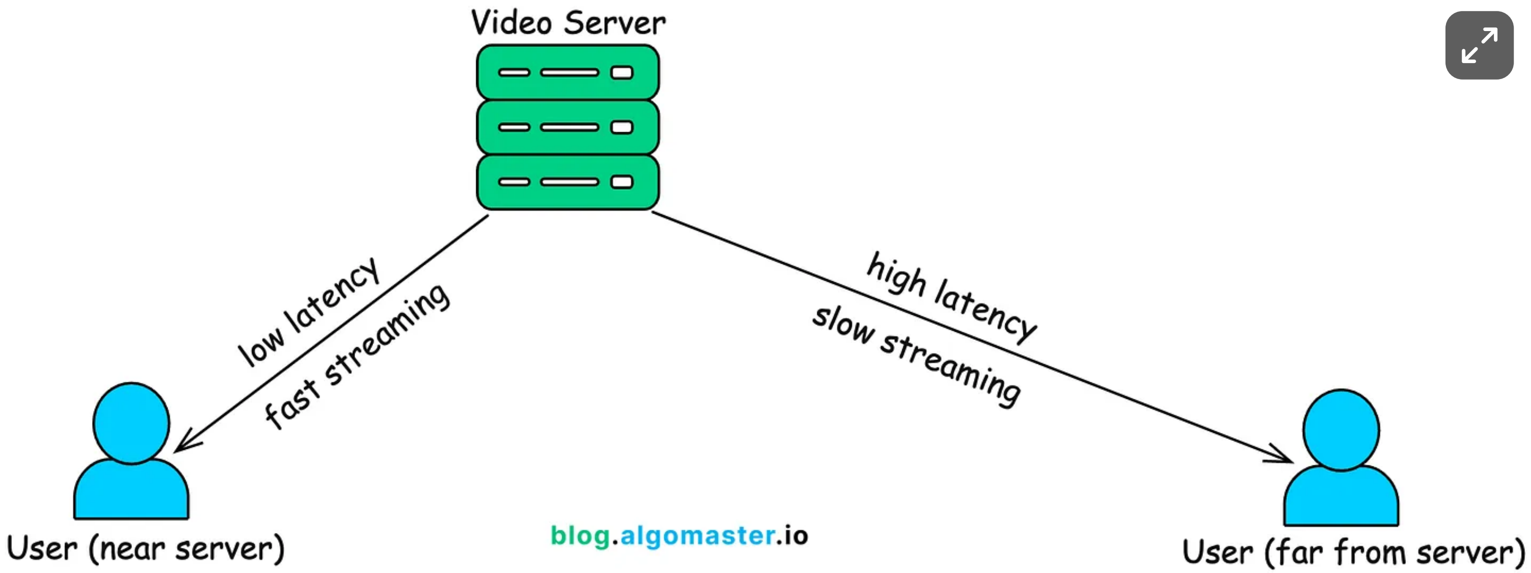
Imagine you've built an app that serves **video content** to millions of users worldwide.

To keep things simple, you host all your videos in **one geographical location**.

At first, everything seems to work fine—users located **near the server** enjoy smooth playback with minimal buffering.

But as your audience grows globally, you start noticing a problem.

Users in distant regions experience **significant latency**, **slow load times**, **and frustrating buffering issues.** The farther they are from your server, the longer it takes for data to travel across the network, degrading their experience.



To fix this, you need a way to **bring your content physically closer to your users,** reducing the distance data must travel.

This is exactly what a **Content Delivery Network** (CDN) does.

In this article, we will explore what a CDN is, how it works, its benefits, different use-cases, and popular CDN providers.

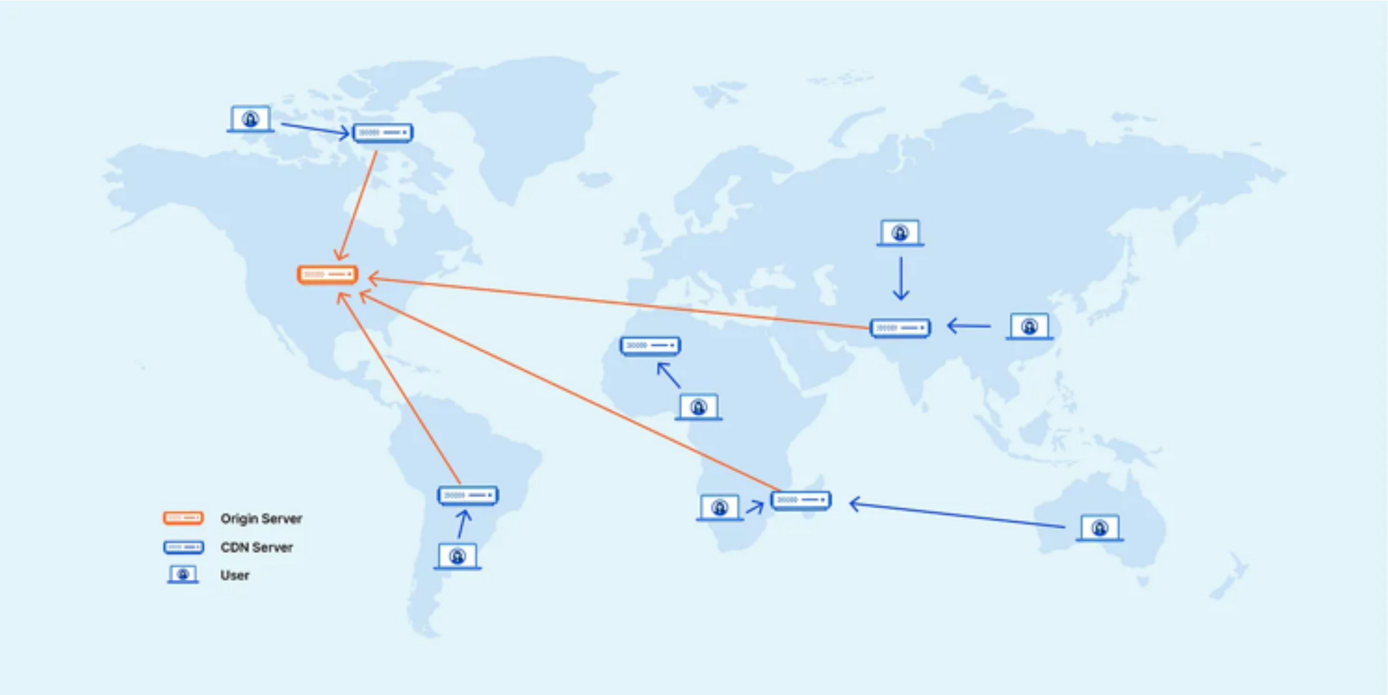
If you're enjoying this newsletter and want to get even more value, consider becoming a paid subscriber.

As a paid subscriber, you'll unlock all **premium articles** and gain full access to all premium courses on algomaster.io.

---

1. What is a CDN?

A CDN is a geographically distributed network of servers that work together to deliver **web content** (like HTML pages, JavaScript files, stylesheets, images, and videos) to users based on their **geographic location**.



source: https://www.cloudflare.com/learning/cdn/what-is-a-cdn/

The primary purpose of a CDN is to deliver content to end-users with **high availability** and **performance** by reducing the physical distance between the server and the user.

When a user requests content from a website, the CDN redirects the request to the nearest server in its network, **reducing latency** and **improving load times.**

2. How Does a CDN Work?

A **CDN** operates using three key components:

- **Edge Servers** – Located at Points of Presence (PoP) locations, these servers cache and deliver content closer to users.
- **Origin Servers** – The primary servers where the original content is stored.
- **DNS** (**Domain Name System**) – Directs user requests to the nearest edge server instead of the origin server.

By leveraging edge servers distributed across multiple geographical regions, CDNs minimize latency and accelerate content delivery.

Here’s a step-by-step breakdown of how a CDN works:

1. **User Request** – A user visits a website and requests content, such as an image, a webpage, or a video. This request must be **resolved to a server** that can serve the content.
2. **DNS Resolution** – The browser **sends a DNS query** to resolve the web content

address (e.g., `https://cdn.example.com/images/logo.png`) into an IP address. The DNS return the **nearest CDN edge server's IP address** rather than the origin server.

3. **Cache Check**
    - If the content is **already cached** at the edge server, it is served immediately to the user.
    - If not, the edge server forwards the request to the origin server. The origin server processes the request and **sends the content back** to the edge server. The edge server caches the content it retrieved from the origin server.
4. **Subsequent Requests** – Once cached, future requests for the same content are served **directly from the edge server**, reducing load on the origin and improving speed.

> CDNs use a **Time-to-Live** (**TTL**) **mechanism** to determine how long content remains cached before expiring. To ensure users always receive the latest version, CDNs **periodically refresh and update cached content** from the origin server.

---

3. Benefits of Using a CDN

- **Faster Load Times** – By serving content from the nearest edge server, CDNs reduce latency and improve page load speed.
- **Reduced Server Load** – CDNs offload traffic from the origin server by caching static assets, reducing resource consumption.
- **Improved Availability and Reliability** – With multiple servers in different locations, CDNs prevent single points of failure.
- **Scalability**: CDNs can handle traffic spikes more efficiently than traditional hosting, making them ideal for websites with fluctuating traffic patterns.
- **Global Reach**: CDNs make it easier to deliver content to users worldwide, regardless of their location.
- **Enhanced Security** – Many CDNs offer DDoS protection, Web Application Firewalls (WAFs), and bot mitigation to secure applications.

While CDNs offer many benefits, it's important to note that they also introduce some challenges like:

- **Increased Complexity:** Integrating a CDN requires proper DNS configuration, cache rules, and content invalidation policies.
- **Increased Cost:** Many CDN providers charge based on bandwidth usage and request volume. For high-traffic websites, CDN costs **can be substantial**,

especially for video streaming, gaming, and software distribution.

4. Use Cases of CDNs

- **Accelerating Website Performance**
  - Websites with **global traffic** use CDNs to ensure **fast page loads** for users regardless of location.
  - **CDNs cache static assets** (images, CSS, JavaScript) at **edge servers**, reducing the time required to fetch them from the origin.
- **Video Streaming & OTT Platforms**
  - CDNs **optimize video content delivery** by caching video files closer to users, minimizing buffering and latency.
  - Supports **adaptive bitrate streaming** (ABR) to serve video based on the user's internet speed.
  - **Example:** Netflix, YouTube, and Spotify use CDNs to serve videos and music

in real-time to user.

- **Online Gaming**
  - Multiplayer **online games** require **low-latency** content delivery to ensure a smooth gaming experience.
  - CDNs help distribute game updates, patches, and downloadable content (DLCs) faster.
- **Content & Media Distribution**
  - News websites and content platforms **deliver images, articles, and videos** through a CDN to handle large traffic spikes.
- **Software Distribution & Updates**
  - Operating system and software vendors use CDNs to **distribute large files, updates, and patches** quickly.
  - Accelerates the distribution of software updates and applications to users worldwide.
  - **Example:** Microsoft, Apple, and Google use CDNs for distributing Windows updates, macOS updates, and Android app downloads.

5. Popular CDN Providers

Here are some of the most widely used CDN providers:

- Akamai: One of the oldest and largest CDN providers, known for its extensive global network and robust security features.
- Cloudflare: Offers a comprehensive suite of performance and security services, including a free tier for smaller websites.
- Fastly: Known for its real-time content delivery and edge computing capabilities.
- Amazon CloudFront: Integrated with AWS, provides seamless scalability and extensive integration with other AWS services.
- Google Cloud CDN: Leverages Google's global network infrastructure to ensure high performance and low-latency content delivery.
- Microsoft Azure CDN – Designed for applications hosted on Microsoft Azure, providing seamless integration with other Azure services.

Choosing the Right CDN

Selecting the best CDN depends on your **use case**, **budget**, and **platform integration requirements**.

Cloudflare and Fastly are great for performance and security, while CloudFront, Google Cloud CDN, and Azure CDN offer seamless cloud integration.

Akamai is a preferred choice for high-scale enterprise applications requiring a robust global network.

Conclusion

A Content Delivery Network is an essential tool for any online service aiming to deliver content quickly and reliably to a global audience.

By understanding how CDNs work, the benefits they offer, and how to choose and implement the right one, you can significantly enhance the performance, security, and scalability of your web applications.

Thank you for reading!

If you found it valuable, hit a like ❤️ and consider subscribing for more such content every week.

This post is public so feel free to share it.

P.S. If you’re enjoying this newsletter and want to get even more value, consider becoming a paid subscriber.

As a paid subscriber, you'll unlock all **premium articles** and gain full access to all premium courses on algomaster.io.

There are group discounts, gift options, and referral bonuses available.

Checkout my Youtube channel for more in-depth content.

Follow me on LinkedIn and X to stay updated.

Checkout my GitHub repositories for free interview preparation resources.

I hope you have a lovely day!

See you soon,

Ashish

125 Likes · 10 Restacks

← Previous

Next →

Discussion about this post

Comments Restacks

Write a comment...

Udit Sharma 16 Mar

@Ashish :I think the cache miss arrow is incorrect ? In a “pure” CDN flow, the end user never goes directly to the origin when there’s a cache miss. Instead, the user’s request still goes to the CDN edge server, and the CDN edge server then forwards that request (a “miss”) upstream to the origin. So strictly speaking, the arrow labeled “Cache Miss” should come from the CDN edge server to the origin server, not from the user to the origin server.

LIKE (8) REPLY SHARE

Yuvaraj A 4 Mar

Liked by Ashish Pratap Singh

Thank you for sharing. Worth reading😄

LIKE (2) REPLY SHARE

3 more comments...