The Essential CDN Guide
An Introduction to Content Delivery Networks
CONTENTS

Introduction .................................................................................................................. 3
What is a CDN? ............................................................................................................... 4
Content Caching ........................................................................................................... 7
Route Optimization ....................................................................................................... 10
Front End Optimization ................................................................................................. 13
CDN Architecture ........................................................................................................ 15
CDN and SSL/TLS ........................................................................................................ 18
INTRODUCTION

Content delivery networks (CDNs) are at the core of every essential website function—including accelerating web traffic, improving visitors’ experience and enabling every website to go global.

As internet users, we interact with CDNs every day. Yet most don’t know anything about the technologies behind them or their functional processes.

Intended for all user levels, this e-book bridges that gap. It’s a comprehensive resource about what CDNs are, their functions and their benefits. For those in the process of selecting a CDN, you’ll find information you need to make an educated decision.

Each chapter is self-contained, providing comprehensive knowledge about the topic it covers.

CDN benefits

- Improve page load speed
- Manage and distribute high traffic loads
- Block spammers, scrapers and other bad bots
- Localize content accessibility without excessive cost
- Reduce bandwidth consumption
- Load balance between multiple servers
- Protect websites from DDoS attacks
What is a CDN?

A CDN is a network of web servers strategically positioned in data centers all over the world.

Because of the physical distance between a website visitor and the origin (source) server hosting the website, content often has to travel a great distance resulting in a delay before it is displayed in a browser. The delay or latency is the duration between the moment a site visitor sends an initial HTTP request to load a page to when it is rendered in their browser. Among many other functions, CDNs cache web server content and solve latency problems by greatly reducing the distance between the browser and the web server.

Every Second Counts

Studies show that a second-long delay causes a 7 percent drop in conversions, an 11 percent drop in page views and a 16 percent drop in customer satisfaction.

Figure 1 - CDN caching servers are located at many PoPs
HOW A CDN WORKS
Shortening the distance between visitors and the origin server, a CDN stores a duplicate, or cached, website version in multiple geographical locations.

Each distributed location is called a point of presence (PoP). Each PoP holds any number of caching servers. They deliver cached website content to site visitors making page requests from within that region, as shown in Figure 1. By reducing the distance each request needs to travel, pages load quicker within visitors’ browsers, providing an overall better user experience.

Every CDN has components that perform specialized functions. In addition to PoPs, these include DNS servers, caching servers and storage servers.

DNS Server - A domain name server (DNS) directs browser requests to the appropriate web site.

When a CDN is used, a browser request is first directed to a DNS server that’s part of the CDN. The DNS server then directs the browser request to a PoP that’s close to the browser rather than sending the request all the way to the origin server. This saves time, and the browser loads the web page much more quickly.

Figure 2 - A CDN brings your content closer to a visitor by routing a request to the PoP closest to your visitor.
**Reverse Proxy** - A reverse proxy is a web server that retrieves content on behalf of the web browsers. Deployed in front of your website’s backend server(s), modern CDNs use reverse proxy technology to achieve two important functions. These are:

**Website Security**
Deployed at the edge of your network, CDNs help filter traffic and block distributed denial of service (DDoS) attacks to your website. Security threats such as malicious bots are prevented from having access to your website and web applications.

**Load Balancing**
By being the single point through which all requests are made to your site, a CDN can effectively distribute requests to an array of servers. This results in improved load distribution and site performance.

To activate a CDN, you’ll need to modify your DNS and subdomain configurations. Your root domain will have a record pointing to one of the CDN’s IP ranges. For subdomains, their CNAME will be modified to indicate a CDN-provided subdomain address (e.g., ns1.cdn.com). If this sounds confusing, don’t worry—the CDN support team will help step you through the (average) five-minute onboarding process.
CONTENT CACHING

WEB CACHE

Content caching is a core CDN service that enables delivery to geographically dispersed visitors (referred to as clients in Figure 3). The process is similar to how a browser caches previously-accessed files to a local hard drive for faster subsequent access.

By storing copies of website content on numerous cache proxy servers located around the world, faster regional distribution to local clients is achieved.

Static files—those not expected to change very often or change based on individual visitor identity—comprise the bulk of cached content on CDN servers.

Figure 3 - Content caching improves page load speed and reduces bandwidth consumption.
CDN CACHING BENEFITS

- Using a CDN is a cost-effective way to reduce bandwidth because fewer requests are sent to origin servers. Bandwidth costs can be reduced by 40 - 80 percent for some sites.
- When cached content is delivered from a nearby PoP, increased page load speed greatly enhances the user experience.
- CDNs are built to be highly resilient and can withstand unexpected traffic peaks and DDoS attacks.

Cache server—Each cache server is a repository for website content copied from the origin server.

Caching algorithms—operational code— are based on average memory reference time. Efficient caching relies on a high hit ratio, indicating that the requested resources (e.g., static text and image files) are present in the server’s cache.

Cache headers—HTTP cache headers mark cacheable content and determine how long it stays cached. By optimizing caching policies, the headers determine caching strategy—including the freshness of your site content. Having to manually write rules for every cached file spread across a CDN drains human resources and is prone to error. To eliminate this challenge, modern CDNs can override cache header directives when they’re found to be suboptimal.

Figure 4 - Cache servers are distributed to maximize coverage and speed up content delivery.
Such technology also assists in caching dynamic content (where freshness isn’t an issue), even if such content has been marked uncacheable by default.

**Cache control**—Cache control adjusts the amount of content to be cached, as well as the efficiency with which caching is performed.

Benefits include:
- Caching adjustment for regionally popular content
- Automated cache rules for frequently accessed assets
- Time-sensitive archive and expiry policies
- Predictive replication for high-demand content

Cache control also permits the identification of caching opportunities for dynamically generated files. By observing patterns, cache control can determine if the same HTML page is being repeatedly served to multiple visitors. In this way, it facilitates the handling of popular dynamic content as if it were static.

**Purge Cache**—Optimal cache management is attained through options such as purge cache control. This offers you the ability to refresh cached files at will. Purge effectiveness is gauged by how long it takes a directive to propagate through the entire CDN.

Check with your CDN provider about its caching rules. Depending on your CDN provider, you may only be allowed to refresh entire cache storage, rather than a set of specific files. Other CDNs may also limit the number of purge cache requests you can make at one time.
ROUTE OPTIMIZATION

Many CDNs also address a number of additional technical aspects, ranging from increased website availability to enhanced site security. But their primary focus is to improve content delivery and therefore the user experience.

CDNs further improve network performance through route optimization. This is achieved largely by using anycast to optimize content delivery.

ANYCAST
Anycast routing lets CDNs bring content closer to visitors by accessing the shortest possible delivery route. To help understand anycast benefits, let’s first look at its counterpart, unicast. In the case of unicast, each network node has a single IP address and all requests to a node are forwarded only to it.

With anycast, however, multiple nodes are able to advertise the same address. This makes any responding node a valid recipient of any client request. With Anycast routing, the request is sent to the nearest node.

This is possible because the devices and technologies governing the internet traffic flow identify and prioritize the shortest path. The shorter the route, the better the connection time for your website visitors. Anycast also offers the advantage of having multiple backup choices for any endpoint. In other words, if one data center is offline, the next nearest handles the request.

NETWORK HOPS
En route between the visitor and the node that ultimately handles their request, the data packets may pass through network bridges, routers and gateways. Each is called a hop and adds to overall processing time—much like the difference between a direct airline flight and one requiring one or more
plane changes. As depicted in Figure 5, the bottom route provides the least amount of latency.

Tracert (Windows) and Traceroute (OSX) are tools you can use to trace the route from your computer to any given host. In Figure 6, it took 14 hops in this instance to reach Imperva.com.

Within a CDN, keeping track of hop counts permits routing devices to identify the nearest node. In tandem with other factors and network optimization methods, this helps determine the best connection paths for visitor requests within a network.

Figure 5 - More hops increase processing time and causes latency

Figure 6 - Tracert can reveal the number of hops between two points
**TIER 1 PROVIDERS**

Internet service providers (ISPs) are classified into three groups, as determined by their connection and payment agreements.

Tier 1 refers to an established group of ISPs (e.g., AT&T, Deutsche Telekom and others) comprising a major core of the internet. They operate their own networks and do not pay for bandwidth. Their global presence permits end-to-end routing of international traffic.

Using a CDN that has agreements with tier 1 providers lets you leverage their benefits. This ensures that visitors reach your site with minimal hops and very low risk of data packet loss.

*Figure 7 - Tier 1 ISPs provide direct routes*
Front end optimization (FEO) is also known as content optimization. It reduces file size and the number of requests needed for a page to load, thereby making your website load more quickly in a site visitor’s browser. Front end delays account for up to 80 percent of all website response times. CDNs help you achieve FEO through file compression, code minification and image optimization.

**FILE COMPRESSION**
Each of your website pages is rendered from a collection of HTML, JavaScript, CSS or other code. Complex pages require larger code files that result in slower load times. Helping to avoid such delays and optimizing pages for fast responsiveness, file compression reduces files to a fraction of their original size. One of the best compression tools preferred for websites is gzip. Consolidating all webpage files in a single compressed tar file, it provides high compression rates while also offering fast encoding and decoding times. CDNs automatically do all of this for you.

**CODE MINIFICATION**
So as to optimize blocks of code, this FEO process takes advantage of how web developers write code and how machines read it. Developers write code for easy comprehension (including breaks, spaces, comments and lines) by humans who need to periodically maintain it. But such characters being non-essential for machines, they can read code much faster without all of the human flourishes. Code minification strips such characters to improve performance. Only the essentials remain, often reducing code by almost half its original length. This file compression reduces it by an extra 5 - 10 percent.

```
// I'm Just a Code Comment on Minification Example -

function() {
  fill(0, 0, 0);
  text("Minification makes code smaller without changing its behavior", 100, 100);
}

minifyExample();
```

Figure 7 - Images show before and after code minification, which strips out the non-essential characters in code, allowing machines to read code faster and improve performance.
CDNs have the ability to completely automate code minification loss.

**IMAGE OPTIMIZATION**

Because most image formats are already compressed (e.g., .JPG, .PNG), image optimization follows a different route. Further compression requires modifying image data by reducing some of the header information or reducing original image quality. Called lossy compression, it’s an option when some reduction in image quality is determined to not be perceptible by a site visitor. It can be used to remove color gradations and reduce pixel complexity.

**PROGRESSIVE RENDERING**

This is an advanced method of on-the-fly image optimization offered by CDNs. Here, the CDN initially loads a highly pixelated image, causing the visitor to realize it’s being rendered in their browser. As orchestrated by the CDN and achieved over the course of a few milliseconds, their browser progressively loads a series of better image versions until the highest resolution image is ultimately rendered.

![Figure 8 - Progressive image rendering improves perceived load time without sacrificing image quality.](image-url)

This fully-automated method offers a good balance between page responsiveness and image quality. Progressive rendering diminishes perceived load time without sacrificing image quality. It also works well with pictorial galleries.
CDN ARCHITECTURE

CDN architecture determines how efficiently it functions.

ROUND-TRIP TIME (RTT)
Measured in milliseconds (ms), RTT is the duration it takes for a visitor’s browser to send a request and receive a response from a server. Four factors affect RTT—file size and internet speed not being among them. Instead, they are:

- Physical distance
- Number of nodes
- Amount of traffic
- Transmission mediums

A high performance website greatly relies on RTT, since content loading within a browser begins only after its initial request is returned by the destination server.

REGIONAL POINTS OF PRESENCE (POPS)
A CDN comprised of widely distributed regional PoPs reduces round-trip request time. By minimizing latency, such a configuration improves speed for all of your website visitors regardless of their geographical location. It also reduces bandwidth consumption and provides scalability in the event of abnormal traffic or an outage.

NETWORK DESIGN
Four major factors dictate CDN design.

Performance – The main purpose of a CDN being to minimize latency, they’re constructed for optimal connectivity. For example, PoPs are located in major data centers where backbone providers peer with one another. Additionally, your CDN service has likely established peering agreements with major tier 1 carriers and other CDNs to provide its customers with the fastest routes for all of their traffic.
Reliability - A CDN can commit to service level agreements (SLAs) that range from 99.9 – 99.999 percent uptime depending on its capabilities. To offer such an SLA, CDNs cannot have any single point of failure. This is achieved through hardware and software redundancy as well as phased maintenance cycles that negate any form of service disruption. They also build in disaster management and mitigation processes in addition to disaster recovery systems.

Scalability - CDN architecture must provide sufficient network and processing resources at multiple levels to accommodate irregular high traffic volumes without suffering any performance degradation. It should also include built-in security against DDoS attacks (e.g., dedicated scrubbing servers).

Responsiveness – The larger the CDN network, the longer it can take to propagate a network-wide configuration change. Therefore a CDN should offer optimum responsiveness in relation to such deployments and upgrades.
CDN TECHNOLOGY

Two deployment topologies are used in CDN architecture.

Scattered CDN - This deployment uses a high number of medium and low-capacity PoPs to densely cover select geographical locations, the focus being on physical proximity. Early CDNs used this during the transition from twisted-pair copper wiring to fiber-optics.

<table>
<thead>
<tr>
<th>Pros</th>
<th>Cons</th>
</tr>
</thead>
<tbody>
<tr>
<td>Physical proximity minimizes latency</td>
<td>Higher maintenance costs</td>
</tr>
<tr>
<td>Effective in low-connectivity regions</td>
<td>RTT prolonged by multiple connection points</td>
</tr>
<tr>
<td>Smaller PoPs are easier to deploy</td>
<td>Cumbersome to deploy new configurations</td>
</tr>
</tbody>
</table>

Consolidated CDN - This topology uses a small number of high-capacity PoPs located at major data centers to increase coverage. Its centralized approach is a more modern approach to CDN design.

<table>
<thead>
<tr>
<th>Pros</th>
<th>Cons</th>
</tr>
</thead>
<tbody>
<tr>
<td>High-capacity servers are better for DDoS mitigation</td>
<td>Less effective in low-connectivity regions</td>
</tr>
<tr>
<td>Permits agile configuration deployment</td>
<td>High-capacity PoPs more difficult to deploy</td>
</tr>
<tr>
<td>Lower maintenance costs</td>
<td></td>
</tr>
</tbody>
</table>
Secure sockets layer (SSL) is a cryptographic protocol used to establish a secure connection between a web browser and server. All data sent using SSL is encrypted and can only be viewed by the intended recipient. This stops a third-party from eavesdropping. Private information such as credit card details, email content and login details are kept secure.

Providing superior encryption, the transport layer security (TLS) protocol has replaced SSL (after version 3.0 was phased out in 2015). But note that TLS may still often be referred to as SSL.

**SSL/TLS Handshake** - The SSL/TLS handshake refers to the sending and receiving of requests between browser and server before a connection is established. A basic TCP handshake involves a three-way back-and-forth exchange to establish a connection. A connection is established in the duration of a single round trip (Figure 12, dark blue).

A SSL/TLS handshake requires more exchanges to establish a secure connection. Two or more round trips result in slower-loading pages.

To achieve security without slow page loads, a CDN compensates for latency introduced by the multiple round trips SSL/TLS requires. A CDN’s ability to shorten RTT ensures you can use SSL/TLS connections and operate a high-performance website.

![Figure 11 - Establishing a basic TCP and SSL/TLS connection](image-url)
SSL/TLS and Keep-Alive - In between visits to your website, keep-alive (also known as a persistent connection) lets a CDN maintain an open connection with your origin server for a duration set by a HTTP timeout value. As long as your site is visited by anyone within the timeout period, subsequent visitors won’t endure additional SSL/TLS negotiations between the CDN and origin server.

In relation to keep-alive, Information Security Stack Exchange—"a question and answer site for information security professionals"—provides a very good technical explanation about the difference between HTTP connections and sessions. It helps you understand how to avoid a SSL/TLS handshake with every new initial request.

SSL/TLS Certificates - SSL/TLS relies on two types of digital certificates. So-called trusted certificates are purchased from a certificate authority (CA) such as Symantec and reside on your webserver. Free certificates are self-signed and can be generated using OpenSSL.

A CA-issued certificate is recommended because a self-signed certificate can’t prevent security alarm messages from popping up on your website and interrupting visitors. TLS implementation also determines the quality of your site security and whether it might be vulnerable. This is determined by three items:

- Protocol support
- Key exchange support
- Cipher support

The SSL/TLS certificate grade is public knowledge and reflects user opinion of your site. Qualys SSL Labs offers a free tool that lets you learn the grade of any site.

Along with a number of other documents and tools, the non-commercial entity also offers Bulletproof SSL and TLS, “a complete guide to deploying secure servers and web applications.”
SSL via Proxy - Using a CDN ensures the first leg of site visitors’ SSL/TLS connections is always established using the service’s own certificate. This overcomes challenges (and those pop-up security alarm messages) associated with having a free, self-signed SSL certificate or a low-graded one. A CDN gives you access to top SSL/TLS via proxy.

![Diagram of SSL via Proxy](image)

Figure 12 - The CDN always has a highly trusted SSL/TLS certificate

CDNs AND DDOS PROTECTION

At Incapsula, our content delivery network has evolved into a platform that offers more than just improved performance. Incapsula CDN is a vehicle for delivery of security and high availability solutions.

Our CDN leverages its integrated DDoS protection and web application security services, as well as its real-time traffic monitoring, failover and load management features.

DDoS attacks fall into three main categories:

**Volumetric Attacks** - These brute force attacks are the most common, including ICMP floods, UDP floods and spoof packet floods. They flood bandwidth and block access to online resources.

**Protocol Attacks** - These attacks target online server resources, rather than bandwidth. They mostly affect communication equipment such as firewalls and load balancers.

**Application Layer Attacks** - These are the most sophisticated types of DDoS attacks because they mimic human user behavior. They’re hard to detect, but they’re capable of overwhelming the application server. And when the server crashes, it brings down the entire site.
CONCLUSION

CDNs are an important part of a website’s operations. If you are looking to improve user experience and keep your site safe, consider using a CDN service to augment your efforts to build a high performing website.

CDNs are also better suited to responding to vulnerabilities. Additional chapters covering more advanced topics—such as enhanced security—will appear in an updated edition of this e-book. These will include:

- CDN and Security
- Multi-CDN Strategies
- CDN and SEO
- Choosing a CDN

Incapsula uses a high capacity network to prevent DDoS attacks:
- 3 Tbps of on-demand scrubbing capacity that can process 30 billion attack packets per second
- Real time visibility into application layer attacks
- Adjustable security rules
Find out how you can optimize website performance with a free 14-day trial

• It’s easy.
  • No software to download or equipment to install
  • On-boarding requires only a DNS change
  • Includes load balancing and web application acceleration

Get Started Today

Questions? Contact us