Networking Training

Academy Batch 2025

PRESIDIO

Agenda

Application-oriented

- CDNs
- SSL/TLS
- DNS Systems and Configuring your DNS
- Overlay Networks and Tunneling
- Anycast and Global Traffic Routing
- Networking Solutions on Public Clouds

Content Delivery Networks (CDNs)

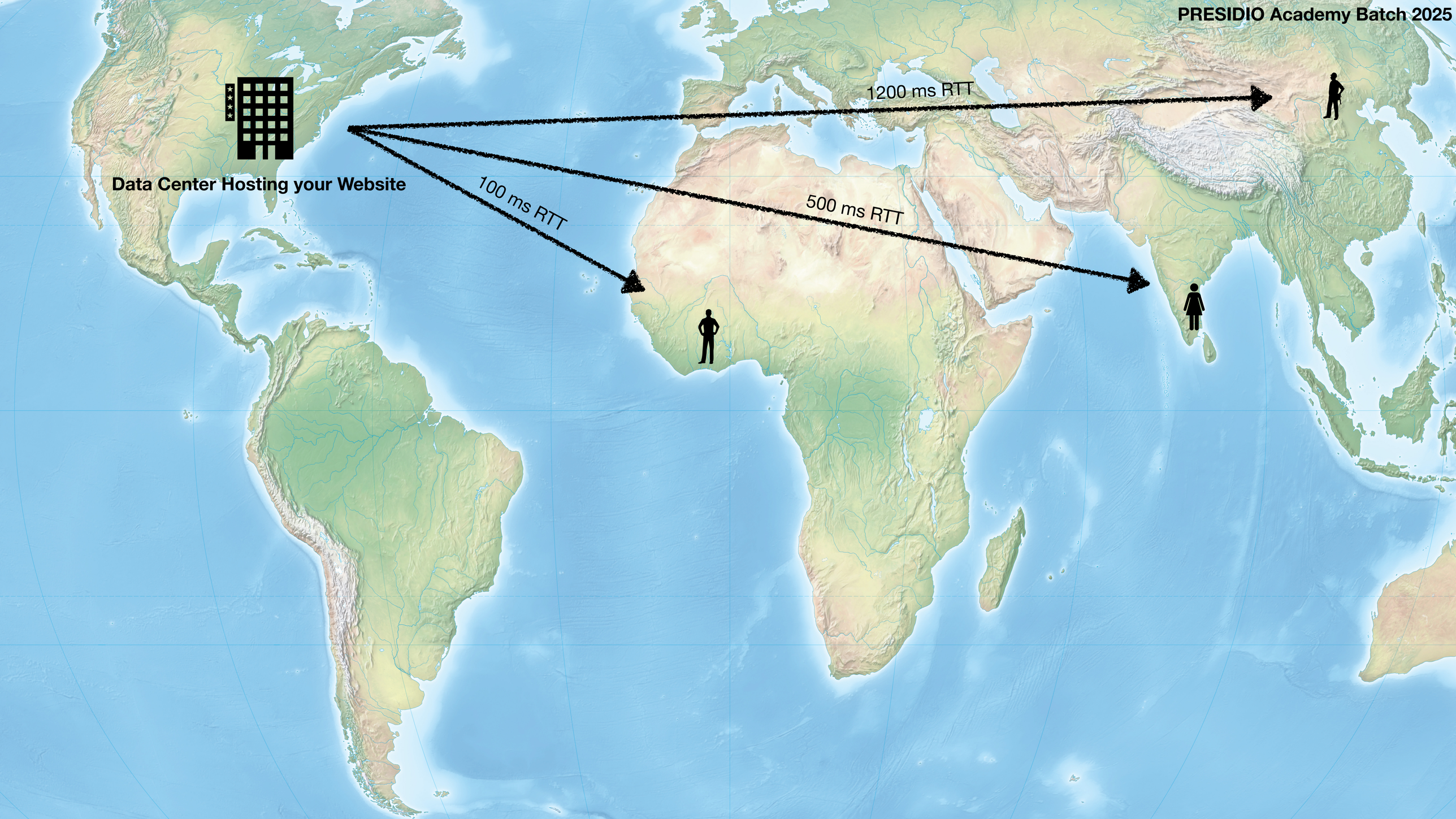
CDNs

What are they?

- A globally distributed network of edge servers (POPs)
- Stores cached copies of web content
- Serves users from nearest location
- Shields origin from heavy load

PRESIDIO Academy Batch 2025

Data Center Hosting your Website


PRESIDIO Academy Batch 2025
Data Center Hosting your Website
1200 ms RTT
500 ms RTT
100 ms RTT


PRESIDIO Academy Batch 2025
Data Center Hosting your Website
PoP
PoP
PoP
< 5 ms Consistent RTT Achieved
Backbone Network
PoP to Client

CDNs – PoPs

Technologies Powering CDNs

- Local data centers hosting CDN edge servers
- Strategically placed across the globe
- Cache and serve content nearest to user
- Reduce latency & improve reliability
- Connects to source via dedicated backbone networks

AWS

Azure

CDNs – Anycast

Technologies Powering CDNs

- Same IP address advertised by multiple PoPs
- Routing protocols (BGP) send user to *nearest* PoP
- Provides load distribution and failover
- Core technology enabling global scale CDNs

CDNs – Control Parameters

HTTP Headers

- Caching decisions live at layer-7 (application layer)
- Because only at layer-7 do we understand HTTP semantics (headers, cookies, query params)
- Layer-7 HTTP headers lets us decide:
  - *What* to cache (static assets vs dynamic responses)
  - *How long* to cache (TTL)
  - When to *invalidate* (based on freshness rules)
  - Who gets what (cache varies by user-agent, geo, cookies)

CDNs – Control Parameters

Client Side: Cache Control

Sent by CDN

`Cache-Control:` `max-age=<seconds>`

On the **client side**, this header is used to tell browsers how they should store and reuse a response, which directly affects performance and freshness of data.

CDNs – Control Parameters

Client Side: Cache Control

`Cache-Control: max-age=<seconds>` { Tells the browser how long (in seconds) it can reuse the cached response before it considers it stale.

On the **client side**, this header is used to tell browsers how they should store and reuse a response, which directly affects performance and freshness of data.

CDNs – Control Parameters

Client Side: Cache Control

`Cache-Control:` `must-revalidate`

{ Ensures that once the cached response becomes stale (after `max-age`), the browser must re-check with the server before using it again. Prevents the browser from serving outdated content even if it’s still available in cache.

On the **client side**, this header is used to tell browsers how they should store and reuse a response, which directly affects performance and freshness of data.

CDNs – Control Parameters

Client Side: Cache Control

`Cache-Control:` `immutable`

{ Tells the browser that the response will never change during its freshness lifetime, so it doesn’t need to revalidate it even if the user refreshes. Useful for versioned static files like JavaScript or CSS bundles.

On the **client side**, this header is used to tell browsers how they should store and reuse a response, which directly affects performance and freshness of data.

CDNs – Control Parameters

Client Side: Cache Control

`Cache-Control:` `private`

On the **client side**, this header is used to tell browsers how they should store and reuse a response, which directly affects performance and freshness of data.

{ The response can be stored in the browser’s cache but not in shared caches (like a CDN or a proxy). This is important for user-specific resources such as personal dashboards or account pages that shouldn’t be cached across users.

CDNs – Control Parameters

Client Side: Cache Control

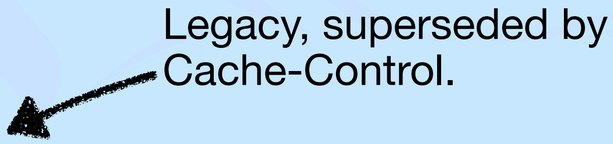
`Cache-Control:` `no-store`

{ Prevents browsers from caching the response at all, nothing saved in memory or disk. Each request always goes to the server.

On the **client side**, this header is used to tell browsers how they should store and reuse a response, which directly affects performance and freshness of data.

CDNs – Control Parameters

Client Side: Cache Control



Expires

This header is an older HTTP/1.0 way of telling browsers and caches until when a resource is considered fresh. It specifies an absolute date and time in GMT.

```
Cache-Control: 3600

Expires: 2020-12-17 18:00:00.000 GMT+0
```

CDNs – Control Parameters

Client Side: Cache Validation

CDN Sends:

`E-Tag: <hash>`

A unique identifier, often a hash or fingerprint, for a resource. The client saves it along with the resource.

Client Sends Back:

`If-None-Match: <hash>`

The client sends a request when the maximum age of the resource expires and wants to verify if the resource has been modified before retrieving a new resource.

CDNs – Control Parameters

Client Side: Cache Validation

CDN Sends:

`Last-Modified: <date>`

The timestamp indicates the last time the resource was modified.

Client Sends Back:

`If-Modified-Since: <date>`

Sent by client to re-evaluate whether the content has changed after max-age has expired.

CDNs – Parameters

Summary of Client-side Parameters

Cache Control

`Cache-Control`

- `max-age=<seconds>`
- `no-store`
- `private`
- `immutable`
- `must-revalidate`

`Expires`

Legacy, superseded by Cache-Control.

Cache Validation

`E-Tag` and **`If-None-Match`**

Hash-based verification.

`Last-Modified` and **`If-Modified-Since`**

Time-based verification.

Overlay Networks and Tunneling

Overlay Networking and Tunneling

Overlay Networks

- Physical networks are rigid (bound by topology)
- Business needs: multi-tenant, mobility, cloud
- Overlay = abstraction above underlay
- Solve problems without touching physical infra

Overlay Networking and Tunneling

Tunneling

- Process of wrapping one packet inside another
- Provides illusion of direct connectivity
- Can add security, abstraction, or protocol extension

Overlay Networking and Tunneling

Real-world Use Cases

- Docker’s Overlay Network
- SSH Tunneling
- Virtual Private Networks

Demo: SSH Tunneling

Overlay Networking and Tunneling

Virtual Private Networks: Need

- Public networks are untrusted
- Data can be intercepted or tampered
- Organizations need secure remote access

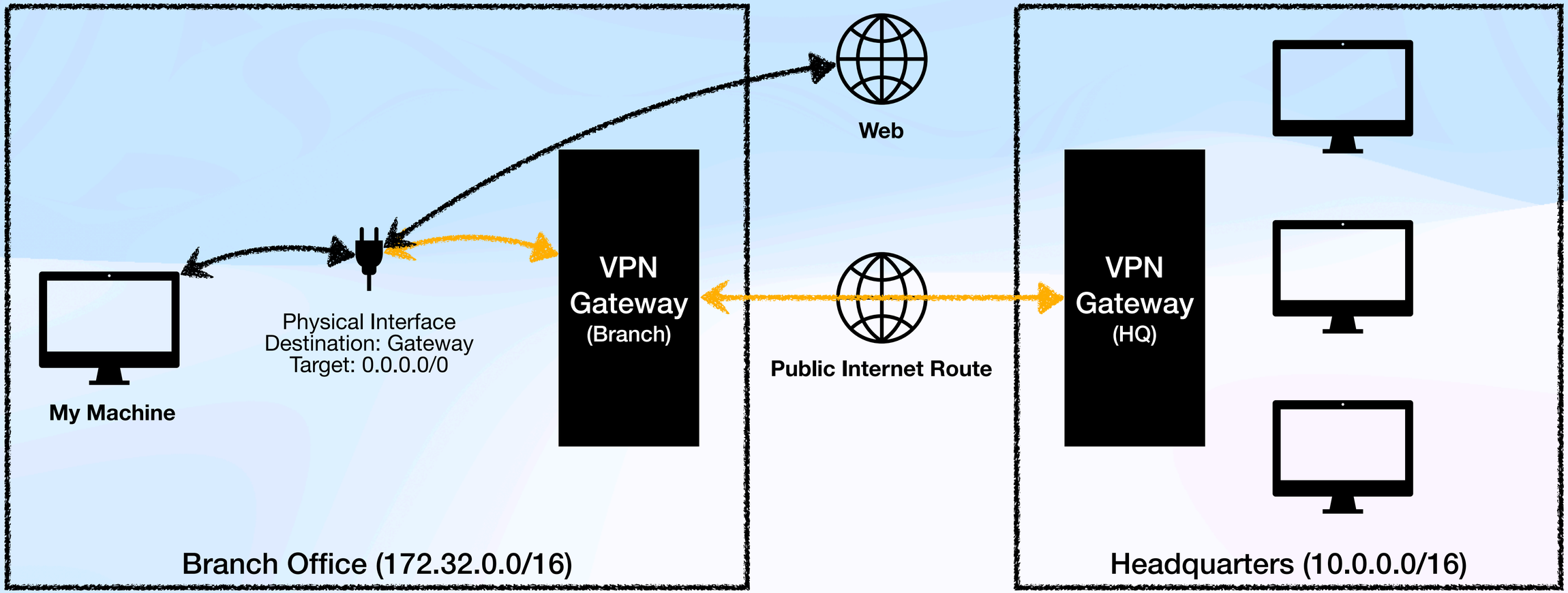
Overlay Networking and Tunneling

Virtual Private Networks

- A secure tunnel between two points
- Uses encryption to protect data
- Provides authentication of endpoints
- Can be:
  - Site-to-Site
  - Remote Access

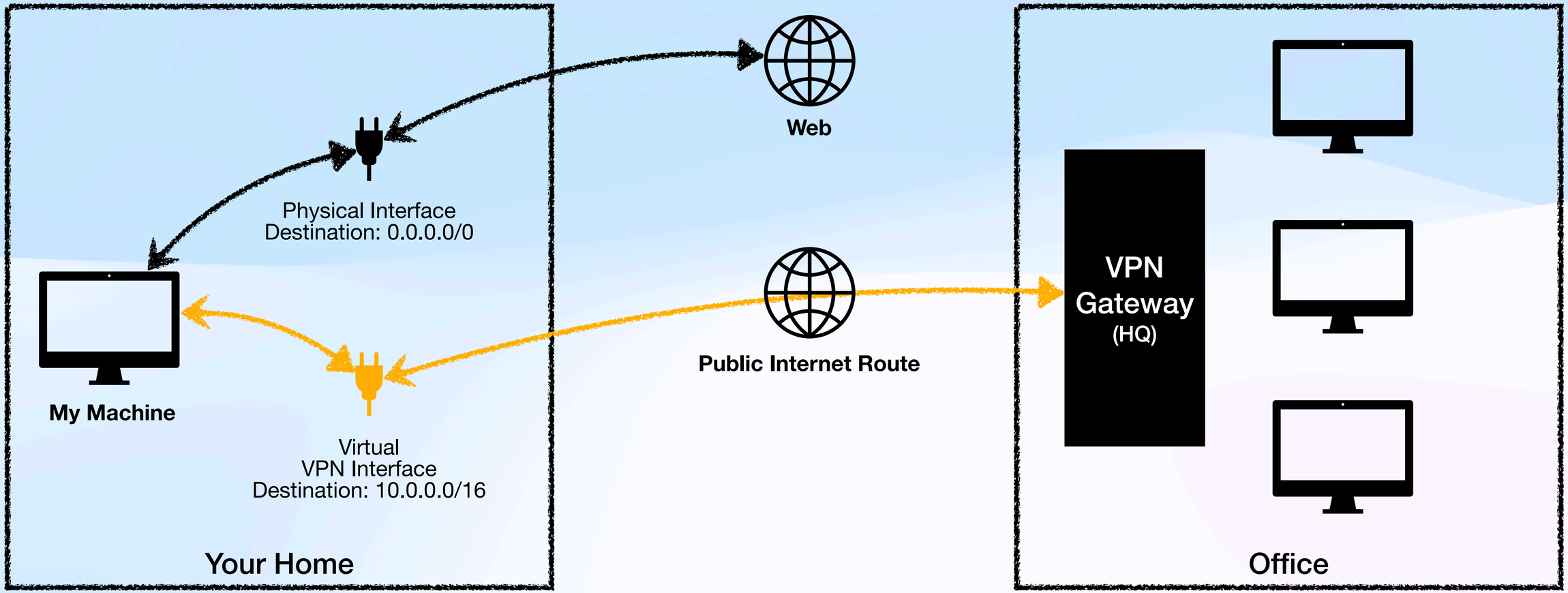
Overlay Networking and Tunneling

Virtual Private Networks: Site-to-Site



Overlay Networking and Tunneling

Virtual Private Networks: Remote-to-Site



Overlay Networking and Tunneling

Differences between Site-to-Site to Point-to-Site VPN

Site-to-Site

- Network to network connectivity
- Centrally managed via gateways on both sides
- Primarily used for exposing remote network
- IP address of device remains same

Point-to-Site

- Device to network connectivity
- Require device setup on “point” end
- Exposes internal services and masks location
- IP address assigned from remote network (overlay)

Summary

- CDN
- Controlling CDN: Control and Validation Parameters
- Anycast Networks
- Overlay Networks and Tunneling

Networking Services on Public Cloud

SSL/TLS

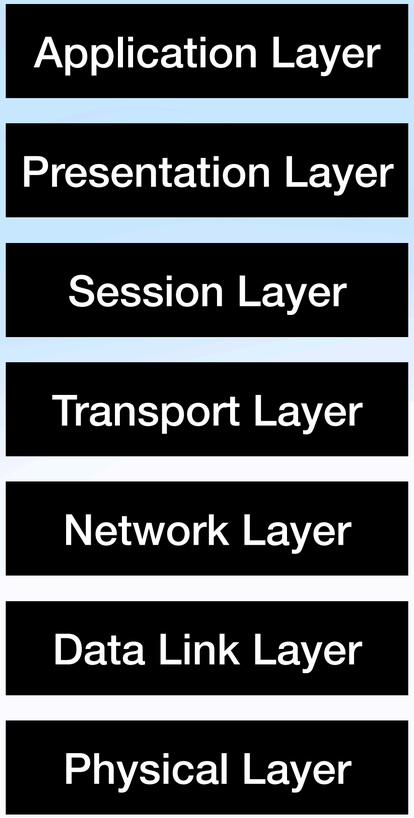
SSL/TLS

History

- SSL: Original protocol to secure the web, but had many flaws
- SSL 3.0: More robust, became the base for TLS
- TLS: Standardized by IETF, improved on SSL
- Modern TLS: Stronger encryption, faster handshakes, mandatory forward secrecy
- Today: SSL is obsolete, TLS is the only standard in use

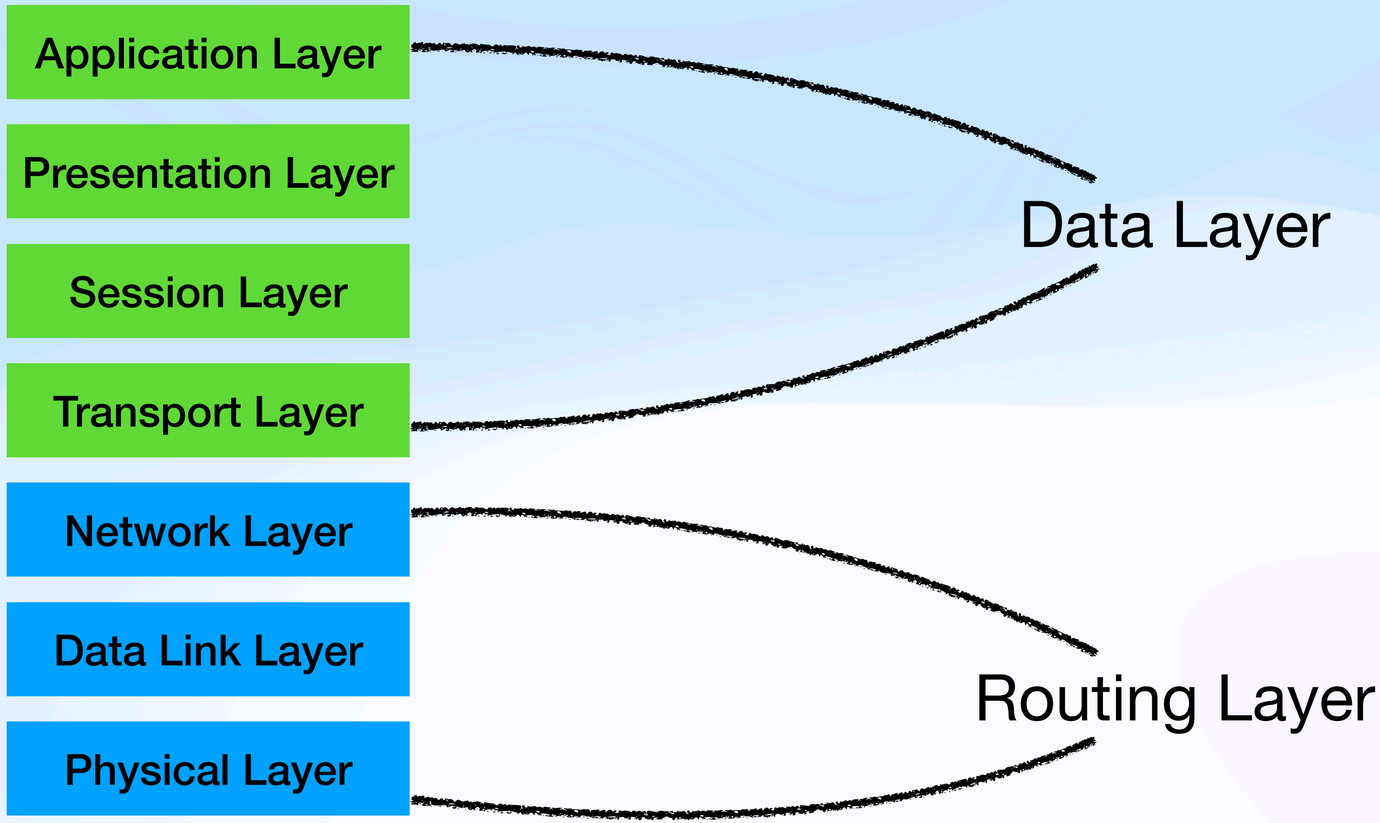
SSL/TLS

OSI Model



SSL/TLS

OSI Model



SSL/TLS

OSI Model