Packets, Protocols and PHP
Networking fundamentals for developers

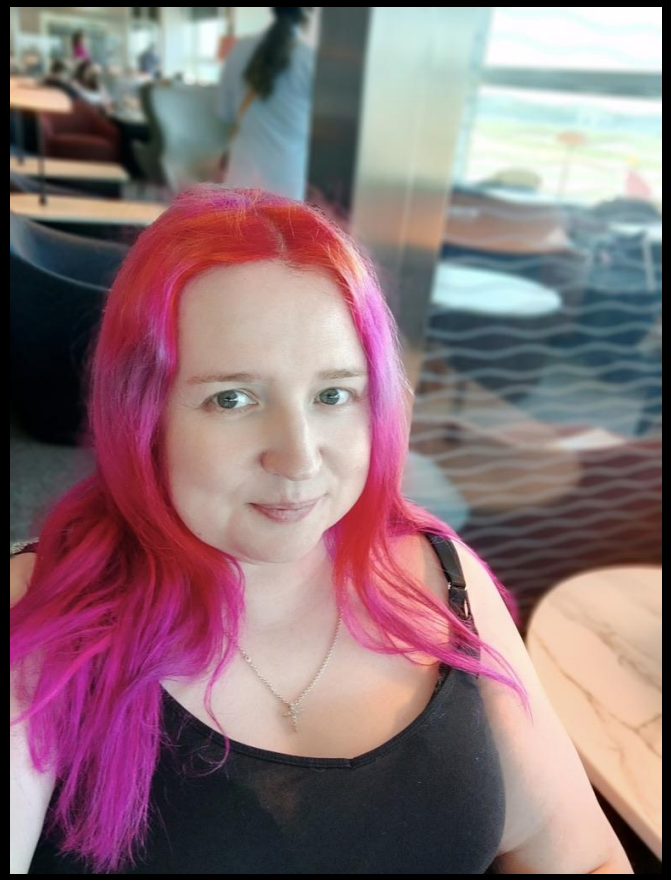
Jessica Smith, Feb 2026

@mintopia@phpc.social

```php
<?php
echo file_get_contents('https://ip.mintopia.net');
```

About Me

- PHP user since 1998
- Originally from Gloucester, now in Reading
- Senior Principal Engineer for DigiCert
- LAN Party Gamer and Event Organiser




OVERCLOCKERS UK
Ducky
EPIC.LAN
intel
EPIC.LAN
intel
EPIC.LAN
EPIC.LAN
intel
EPIC.LAN
intel

```php
<?php
echo file_get_contents('https://ip.mintopia.net');
```

Abstraction

Removing unnecessary details to allow us to focus on the problem we're trying to solve.

Examples

Eloquent ORM

Guzzle

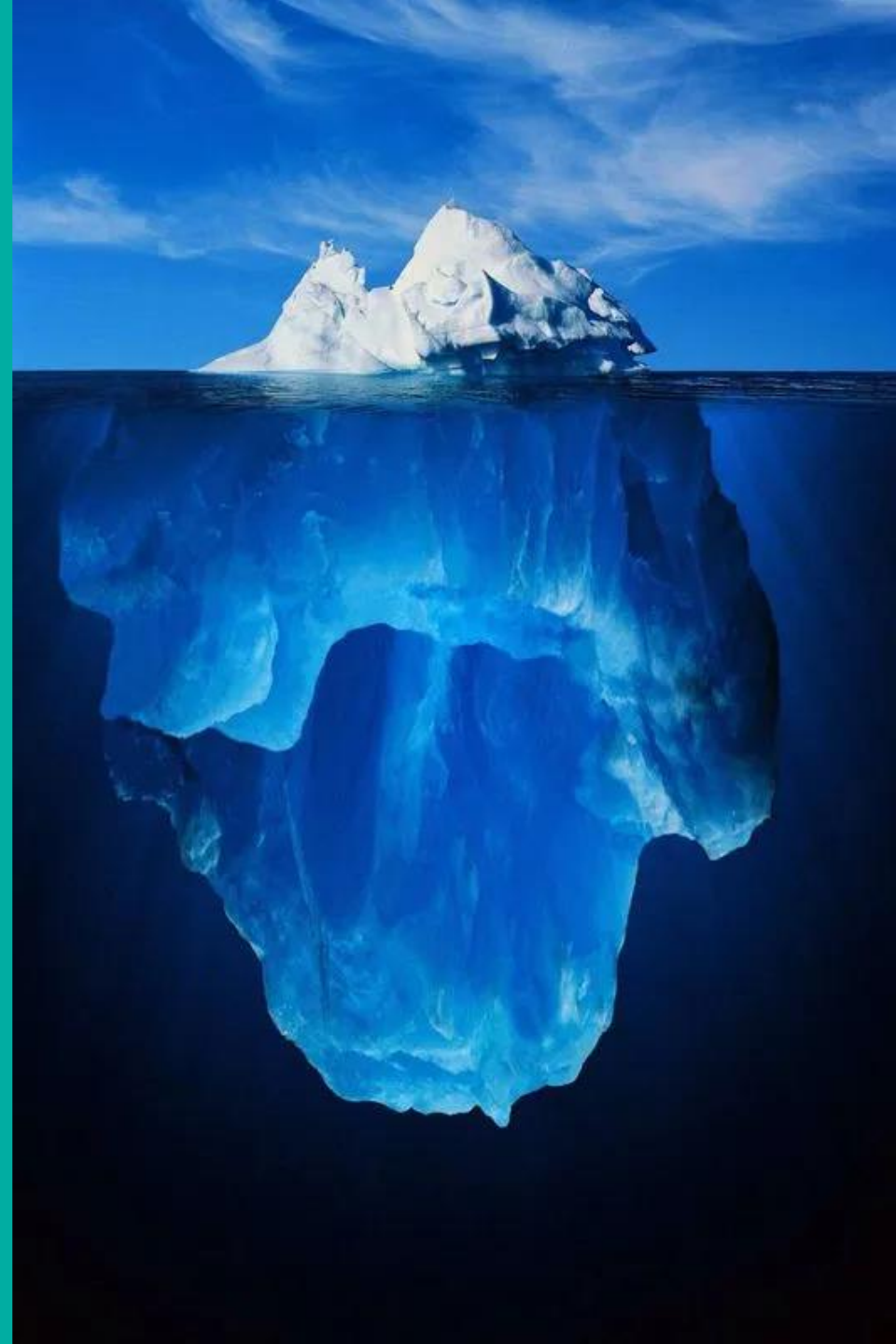
The Law of Leaky Abstractions

"All non-trivial abstractions, to some degree, are leaky."

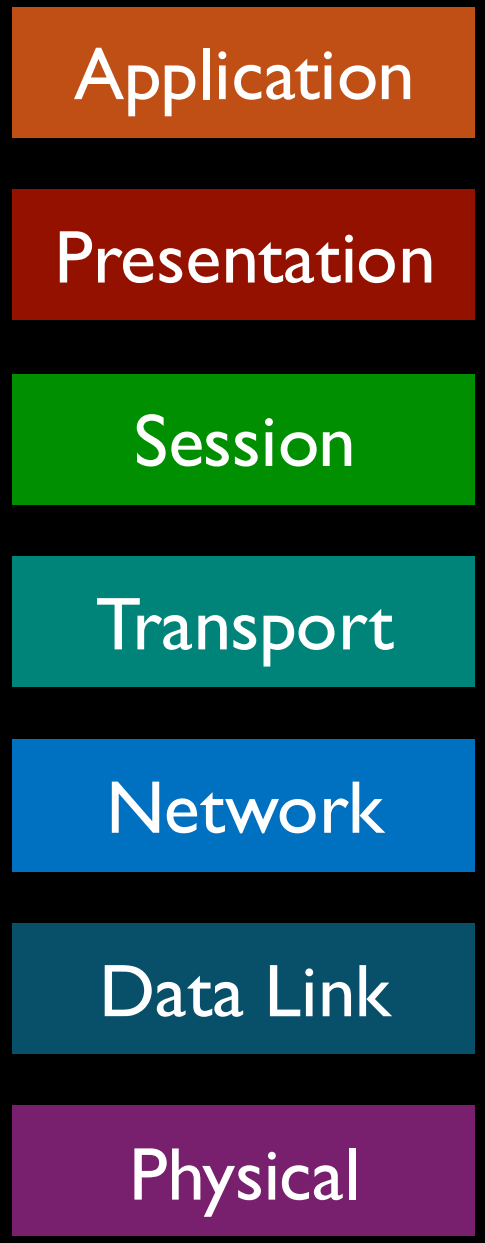
Joel Spolsky

One Function, Many Layers

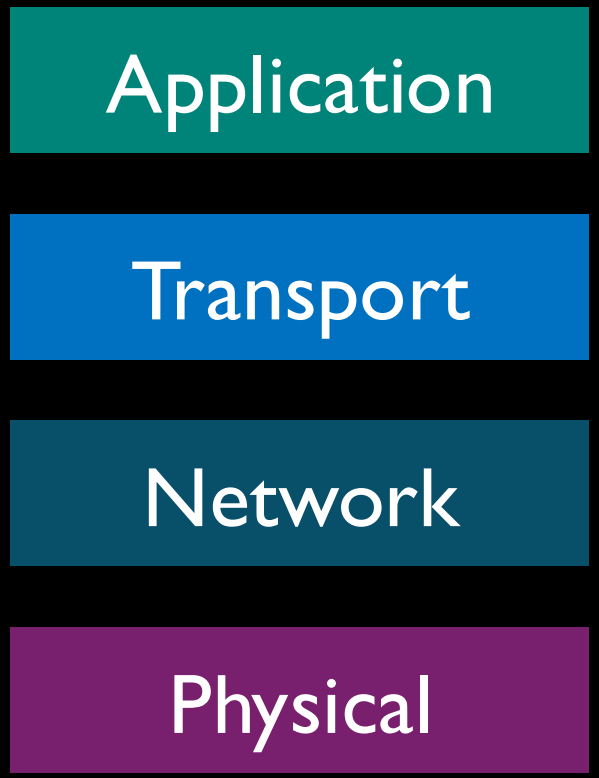
```php
<?php
echo file_get_contents('https://ip.mintopia.net');
```



OSI Model

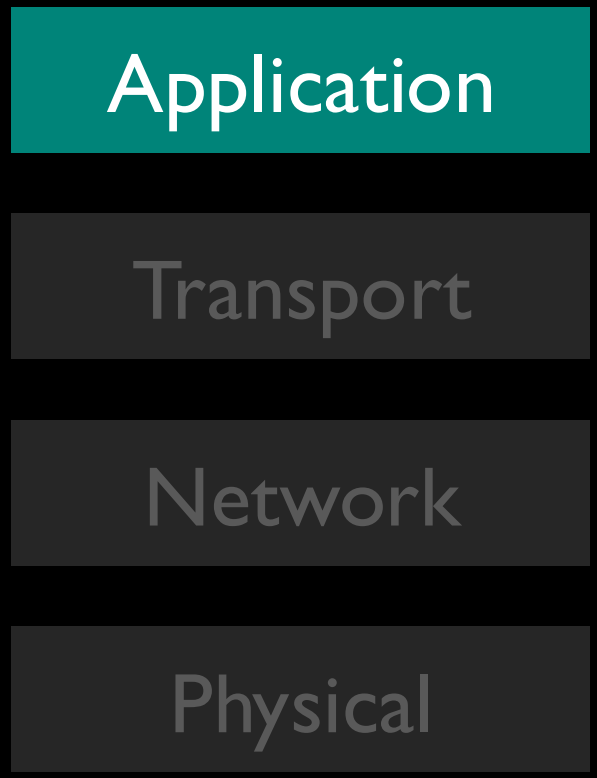

Application
Presentation
Session
Transport
Network
Data Link
Physical

TCP/IP Model



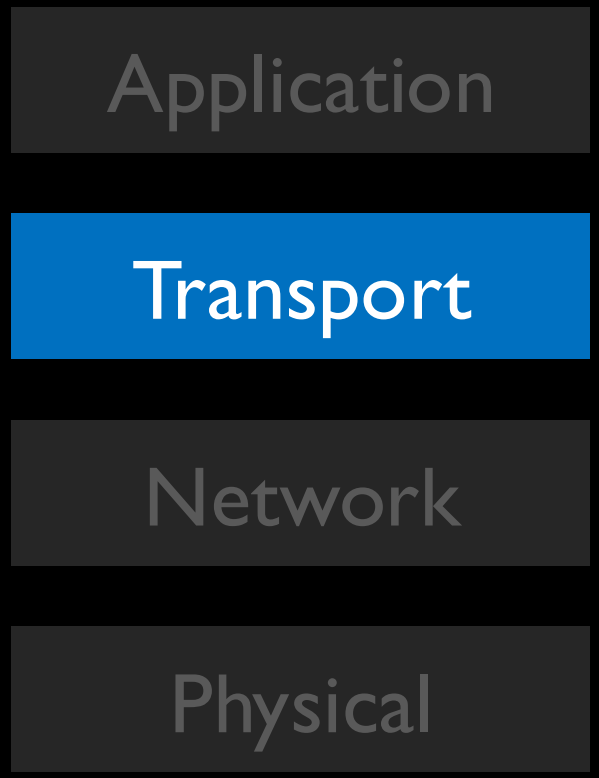
Application Layer

- Applications – PHP, Browser, Email, Games
- Protocols – HTTP, SMTP, FTP
- Compression – gzip
- Encoding – JSON, HTML
- Encryption – TLS (sort of)



Transport Layer

- Protocols – TCP, UDP
- Ports – 21, 22, 25, 80, 110, 443, 34197
- Error Correction
- Flow Control
- Packets – Size limited by MTU



TCP – Transmission Control Protocol

- Connection based
  - Established with three-way hand-shake
- Reliable
  - Packets will arrive in the correct order
  - Missing packets will be resent
- 20 byte header

HTTP

SSH

SMTP

MariaDB

UDP – User Datagram Protocol

- No Connections
- Packets may not arrive in order
- Missing packets are not resent
- 8 byte header
- No Handshake
- Quicker to establish than TCP

DNS

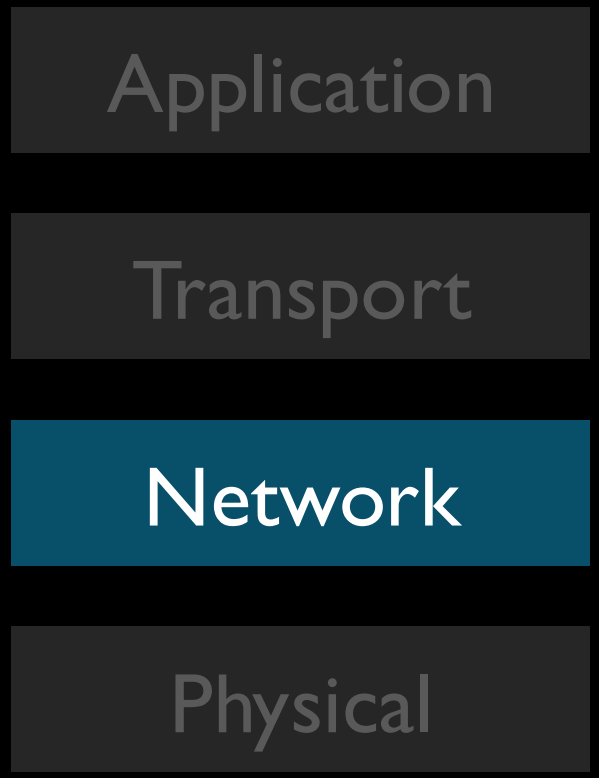
Gaming

VOIP

QUIC

Network Layer

- IP Addresses
  - IPv4
  - IPv6
- Routing
  - Static Routing
  - BGP



IPv4 and IPv6

44.30.69.1

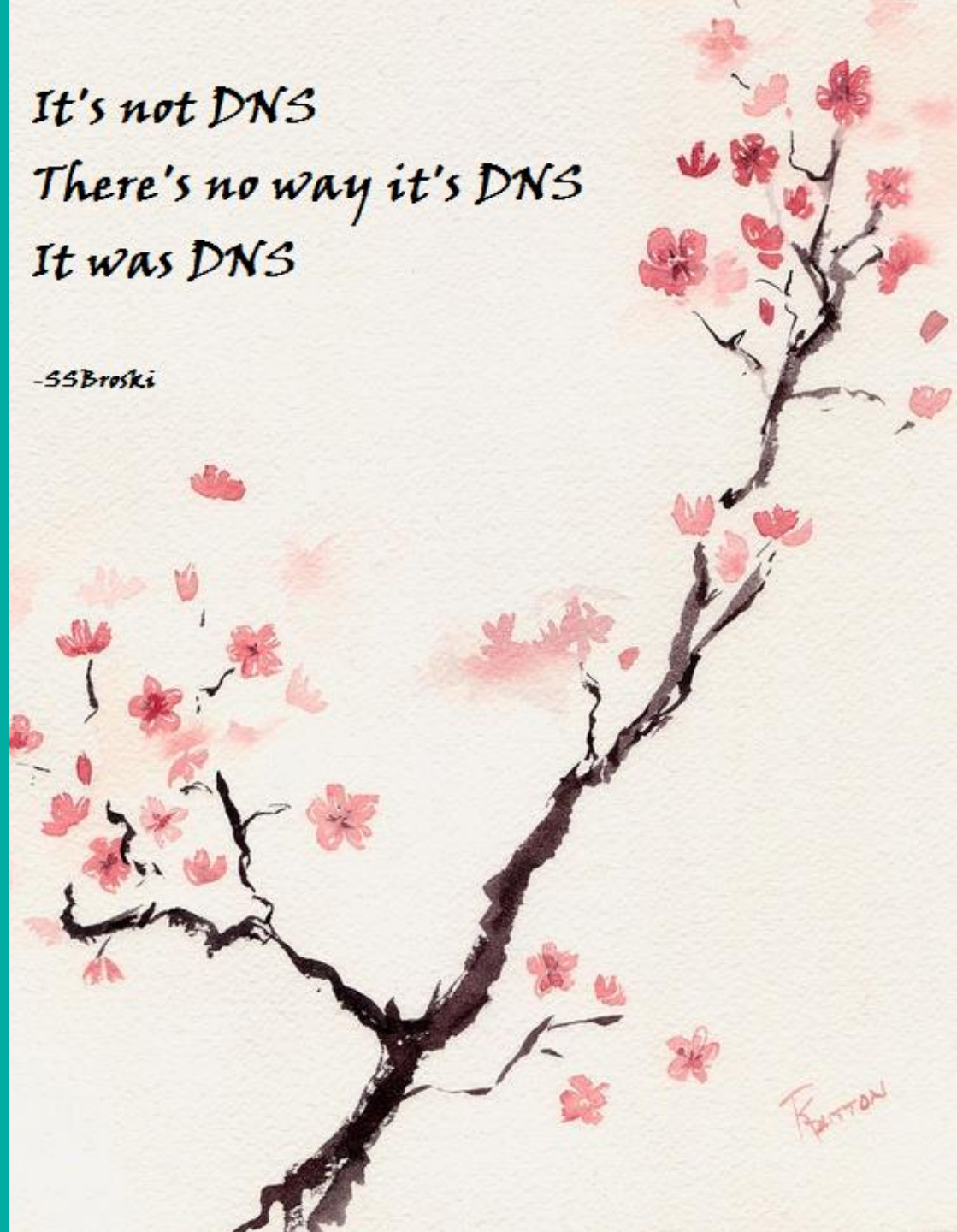
- 32 bit
- 4.3 billion
- We have none left
- NAT and CGNAT used

2a0f:85c1:d91::1

- 128 bit
- 340 undecillion (10^{36})
  340,000,000,000,000,000,000,000,000,000,000,000
- 45% of Internet Traffic is IPv6
- 30% of websites in 2024

DNS

- Maps names to IP addresses
- Many other uses:
  - Defining email servers
  - Specifying certificate authorities
  - Mapping IP addresses to names
- Hierarchy of domains and records.



Routing

- How do we get from here to there?
- Static Routing
- Dynamic Routing
  - OSPF
  - BGP
- IP addresses are grouped together

  44.30.69.0 to 44.30.69.255 is 44.30.69.0/24

  44.30.69.64 to 44.30.69.127 is 44.30.69.64/26

Subnets/Networks and Prefixes

A way of representing groups of IP addresses.

IP Address:	44.30.69.42
Subnet Mask:	255.255.255.128
	11111111.11111111.11111111.10000000
	25
Starting IP:	44.30.69.0
Ending IP:	44.30.69.127

44.30.69.42/25

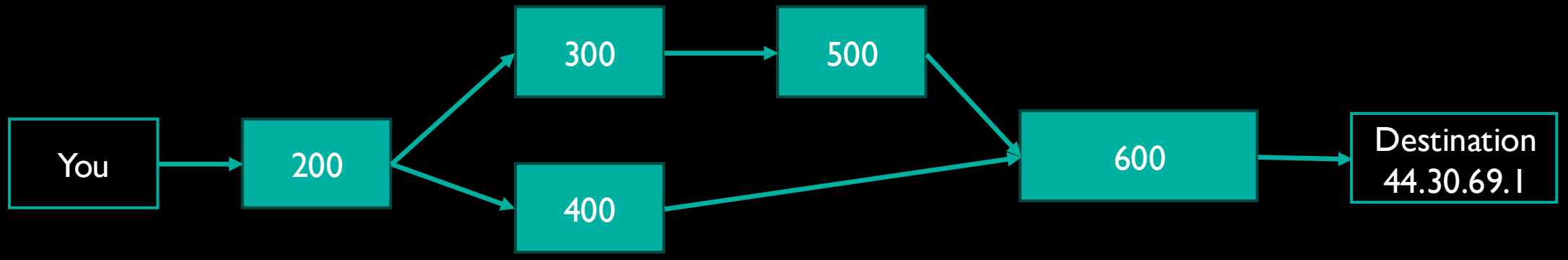
Static Routing

- Literally a list of networks and the IP or network interface you can use to get there
- Usually a default route of 0.0.0.0/0 for anything not defined
- A more specific route has higher priority

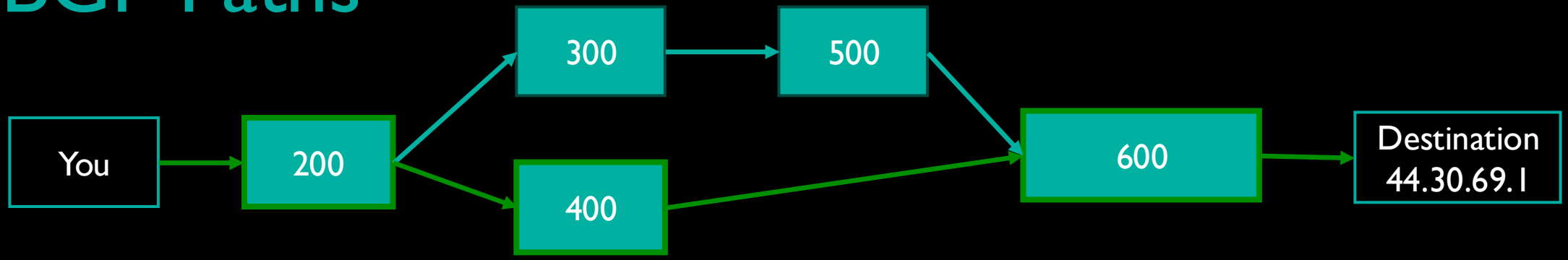
Network	Gateway
192.168.0.10/24	eth0
44.30.69.0/25	tailscale0
192.168.100.0/24	192.168.0.2
0.0.0.0/0	192.168.0.1

Border Gateway Protocol

- The Internet is a collection of connected Autonomous Systems (AS)
- Each AS is represented by a number, eg. 204776
- Where they are connected they announce the networks they can provide a path for



BGP Paths



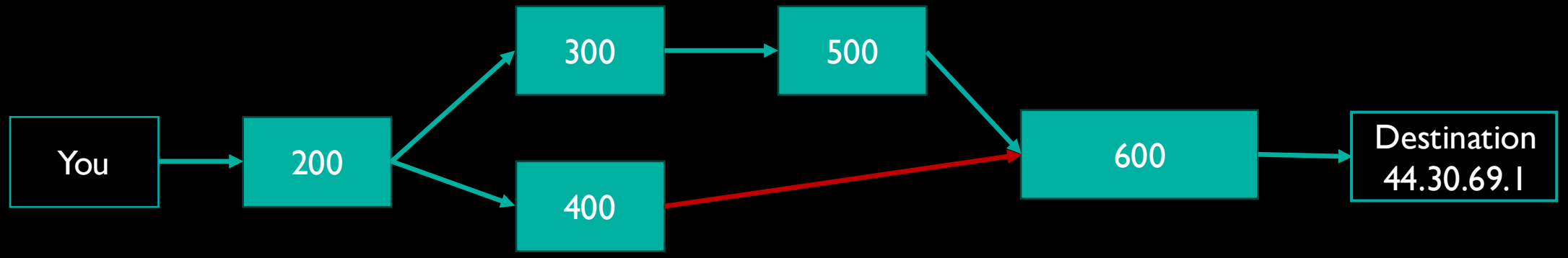
AS 200's Paths

44.30.69.0/24	300 → 500 → 600
44.30.69.0/24	400 → 600

Your Paths

44.30.69.0/24	200 → 400 → 600

BGP and Outages



AS 200's Paths

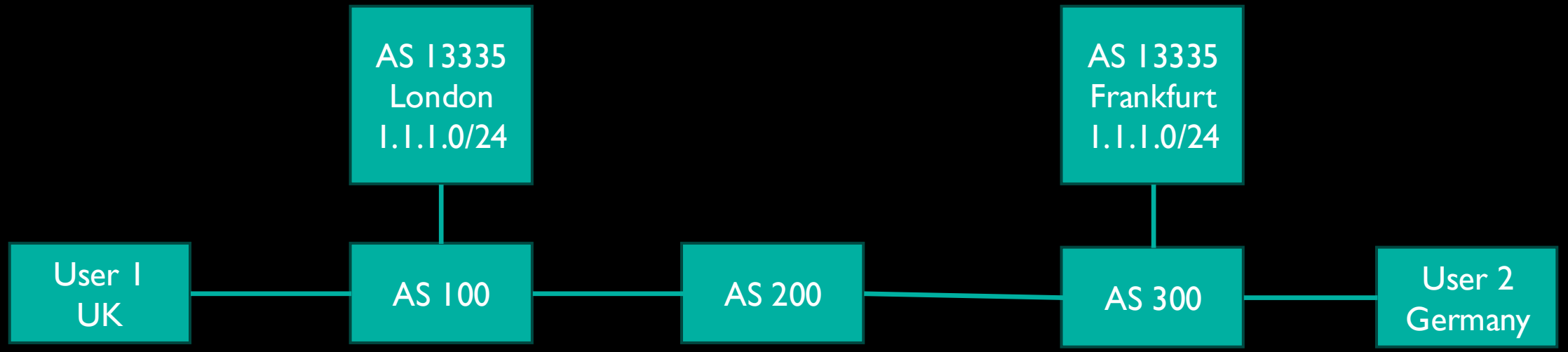
44.30.69.0/24 300 → 500 → 600

~~44.30.69.0/24~~ ~~400 → 600~~

Anycast

How does this IP work everywhere?

Anycast

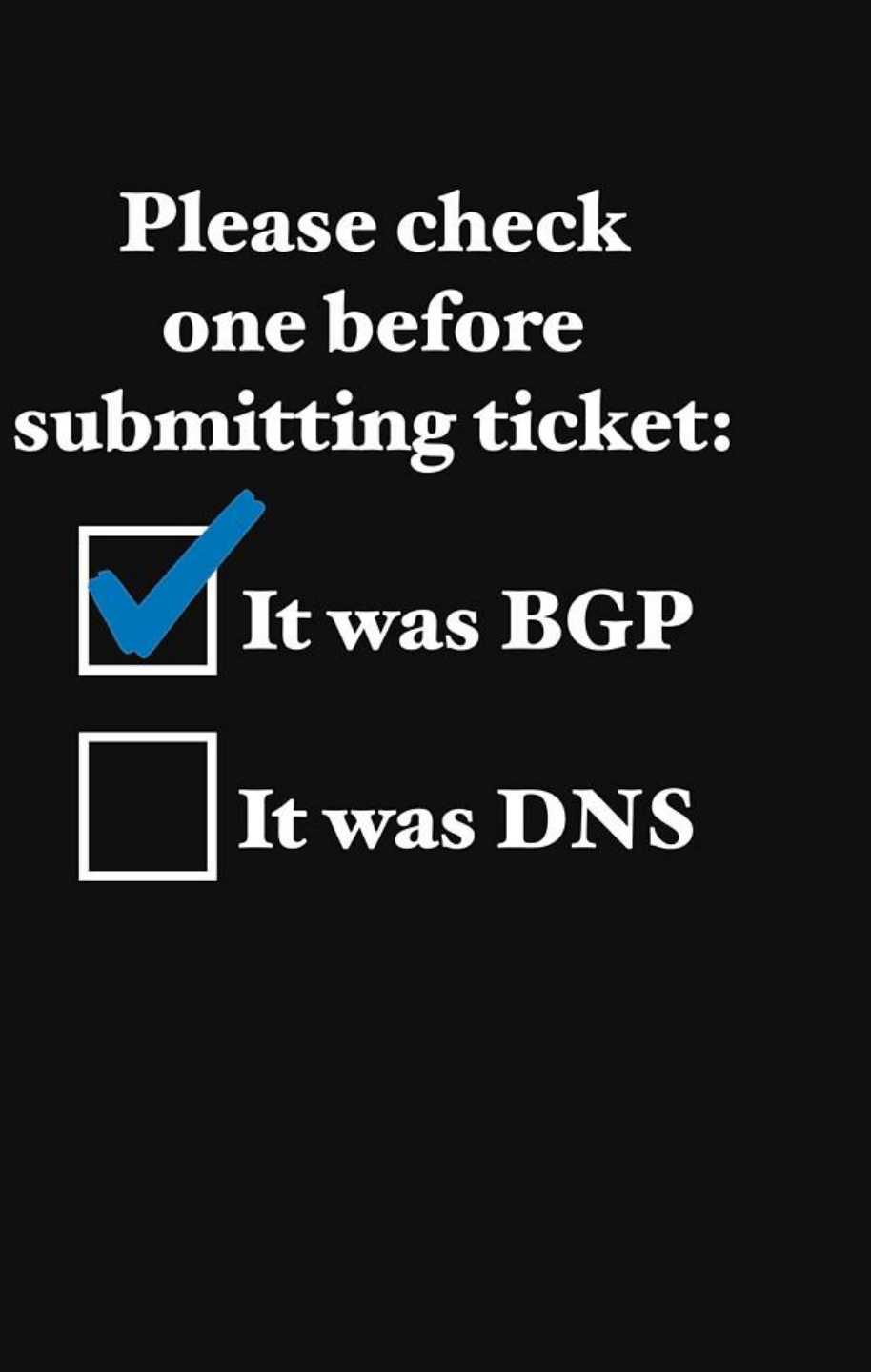


UK:
1.1.1.0/24 100 → 13335
1.1.1.0/24 100 → 200 → 300 → 13335

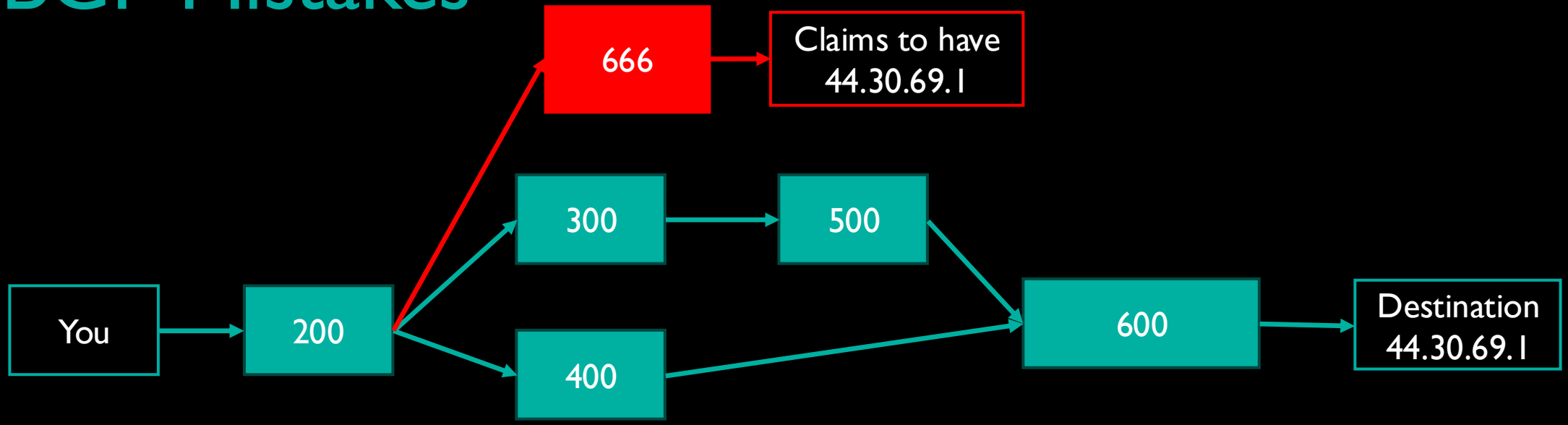
Germany:
1.1.1.0/24 300 → 13335
1.1.1.0/24 300 → 200 → 100 → 13335

BGP Mistakes

- Is it always DNS?
- No. Sometimes it's BGP.
- Sometimes it's DNS *because* of BGP



BGP Mistakes



Valid Paths

44.30.69.0/24 200 → 300 → 500 → 600
44.30.69.0/24 200 → 400 → 600

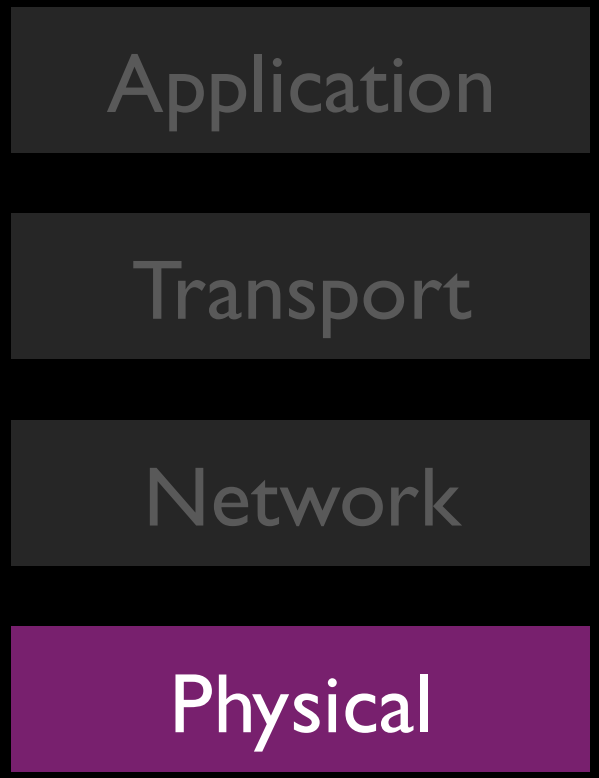
Invalid Path

44.30.69.0/24 200 → 666

Physical Layer

- Network Cables
- Fibre
- Satellite
- 5G
- WiFi

- Latency
- Quality



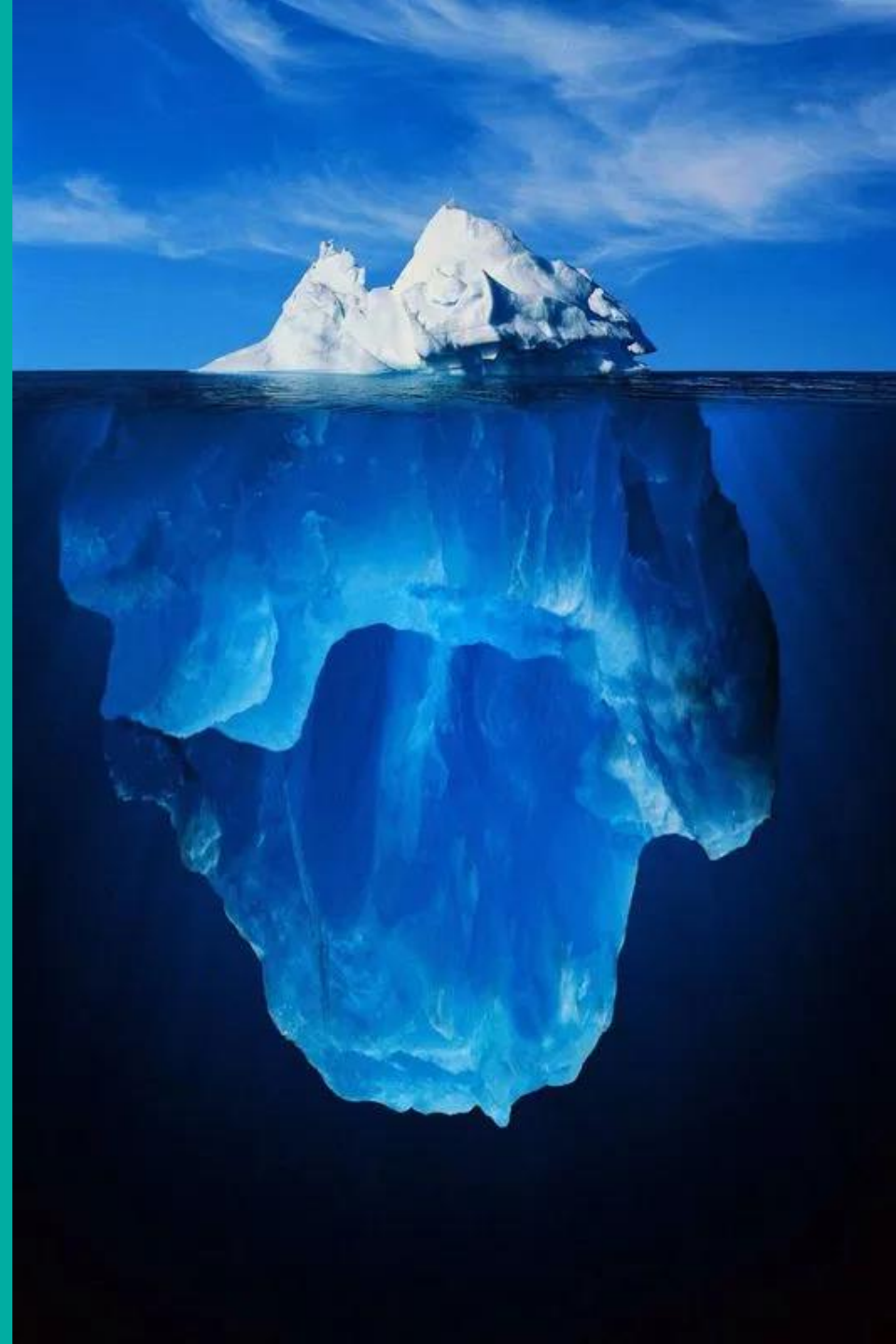
RFC 1149 – IP over Avian Carriers

```
vegard@gyversalen:~$ ping -i 900 10.0.3.1
PING 10.0.3.1 (10.0.3.1): 56 data bytes
64 bytes from 10.0.3.1: icmp_seq=0 ttl=255 time=6165731.1 ms
64 bytes from 10.0.3.1: icmp_seq=4 ttl=255 time=3211900.8 ms
64 bytes from 10.0.3.1: icmp_seq=2 ttl=255 time=5124922.8 ms
64 bytes from 10.0.3.1: icmp_seq=1 ttl=255 time=6388671.9 ms

--- 10.0.3.1 ping statistics ---
9 packets transmitted, 4 packets received, 55% packet loss
round-trip min/avg/max = 3211900.8/5222806.6/6388671.9 ms
```

https://blug.linux.no/project/rfc1149/

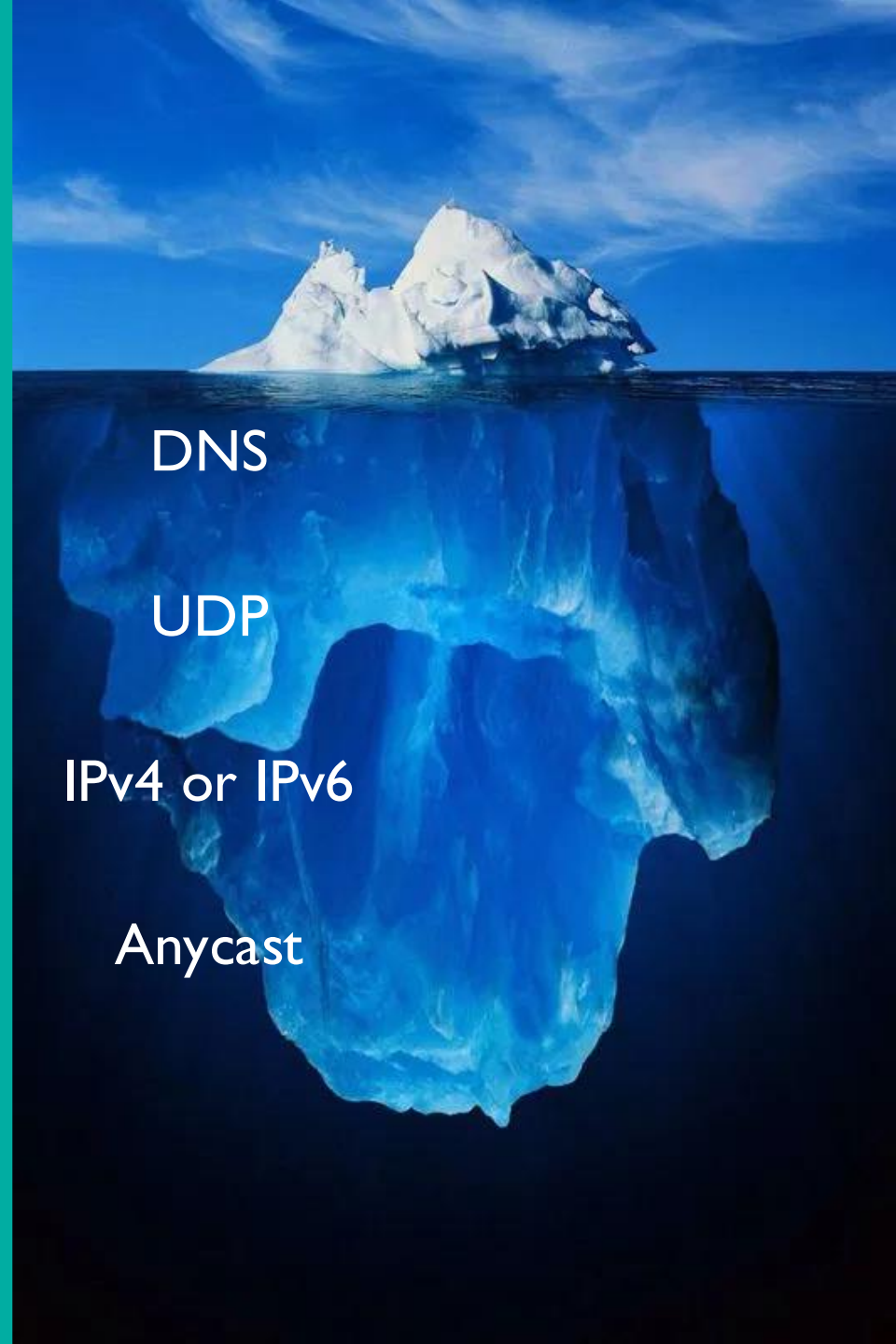
```php
<?php
echo file_get_contents('https://ip.mintopia.net');
```



```php
<?php
echo file_get_contents('https://ip.mintopia.net');
```

DNS

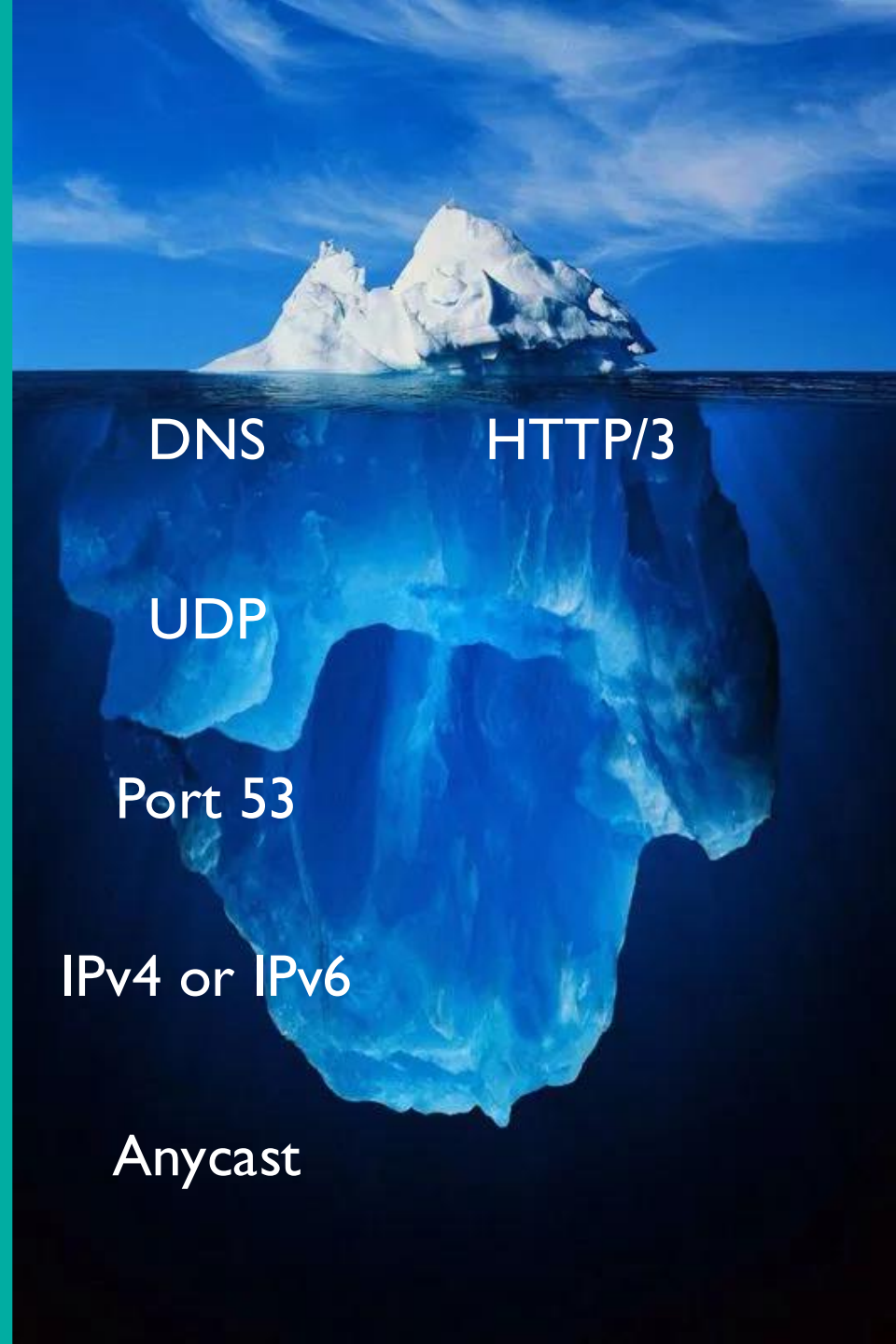
- UDP
  - But can be TCP
  - or served over TLS
- Quick
- Located close to us



```php
<?php
echo file_get_contents('https://ip.mintopia.net');
```

Application

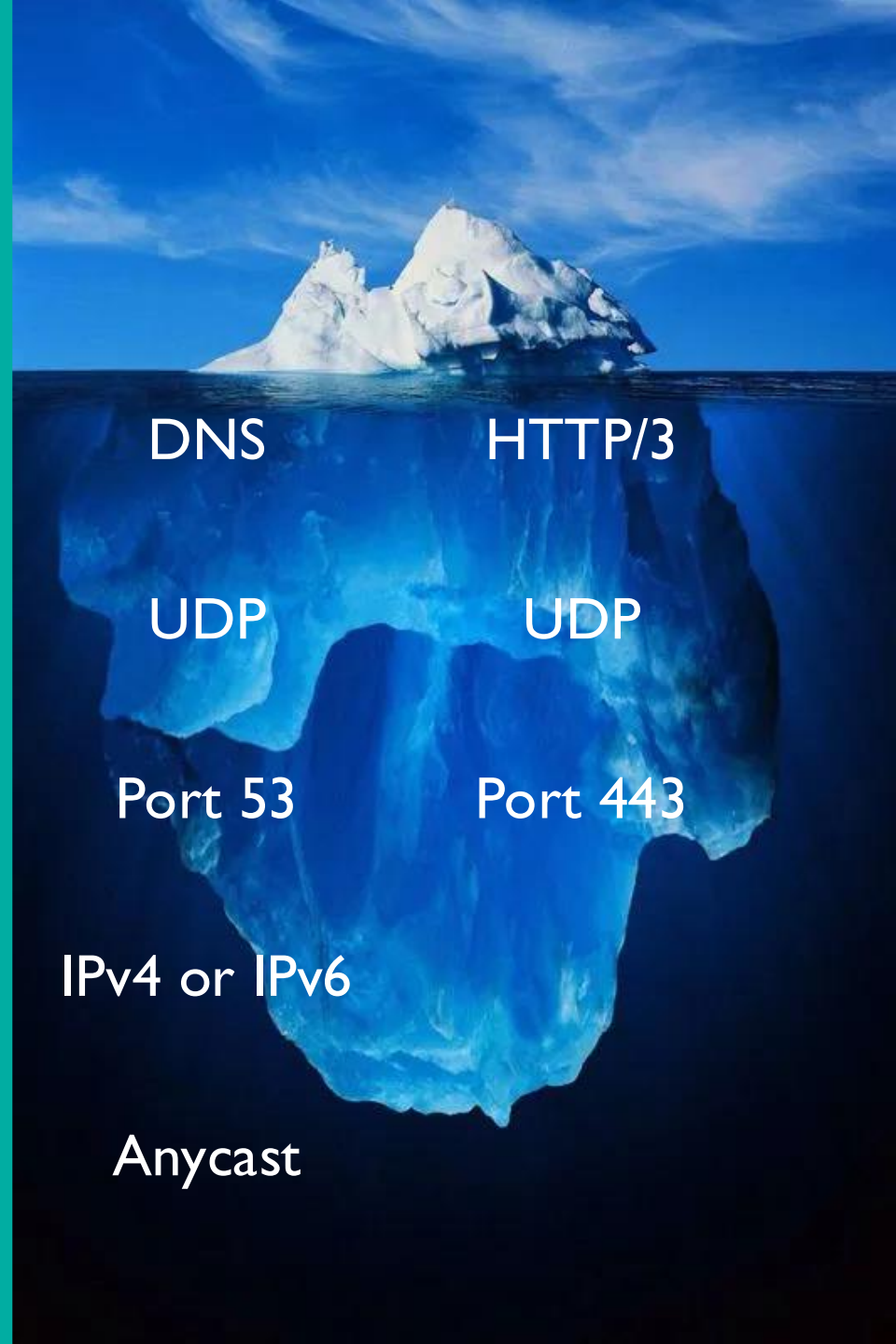
- What version of HTTP?
- HTTP3 is binary, not plaintext
- gzip
- TLS



```php
<?php
echo file_get_contents('https://ip.mintopia.net');
```

Transport

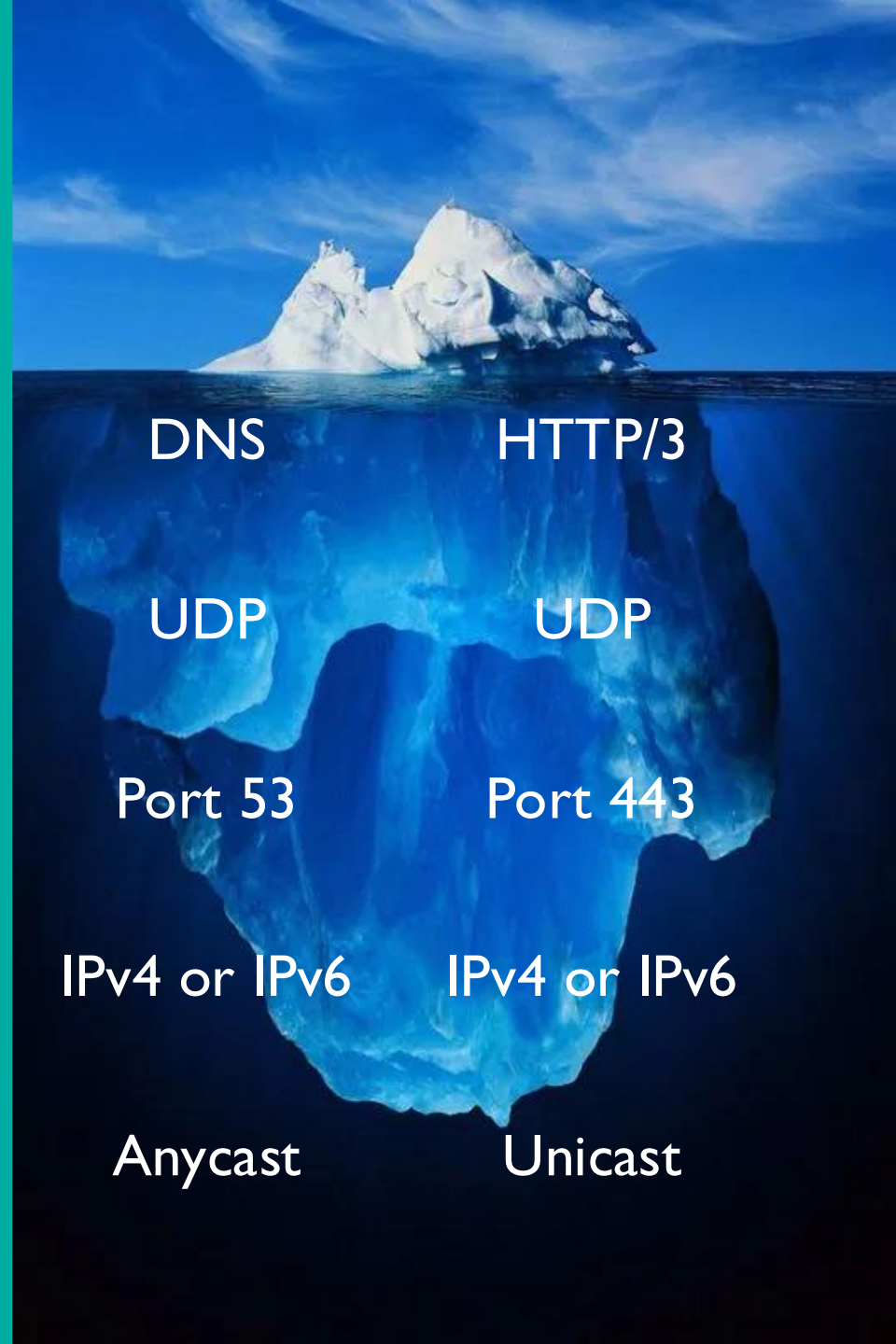
- HTTP/3 is QUIC which is UDP
- Multiplexing, multiple files at once
- Missing packets don't stall the connection
- Port 443



```php
<?php
echo file_get_contents('https://ip.mintopia.net');
```

Network

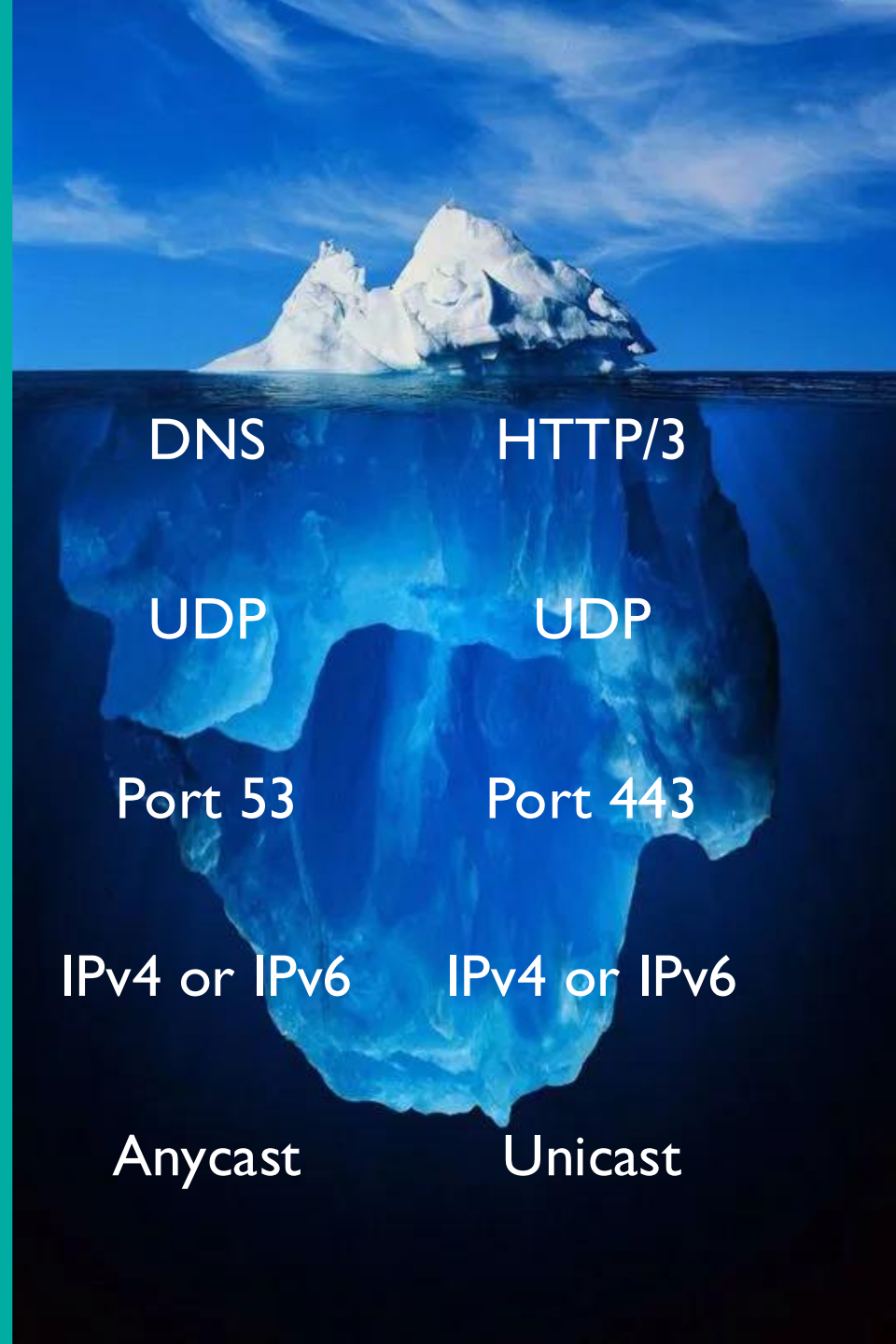
- Either IPv4 or IPv6, we don't care
- If you're storing IP addresses – support IPv6
- Expect multiple users per IP due to NAT
- Using unicast BGP routing to a single host
- But maybe using Anycast to a CDN



```php
<?php
echo file_get_contents('https://ip.mintopia.net');
```

Physical

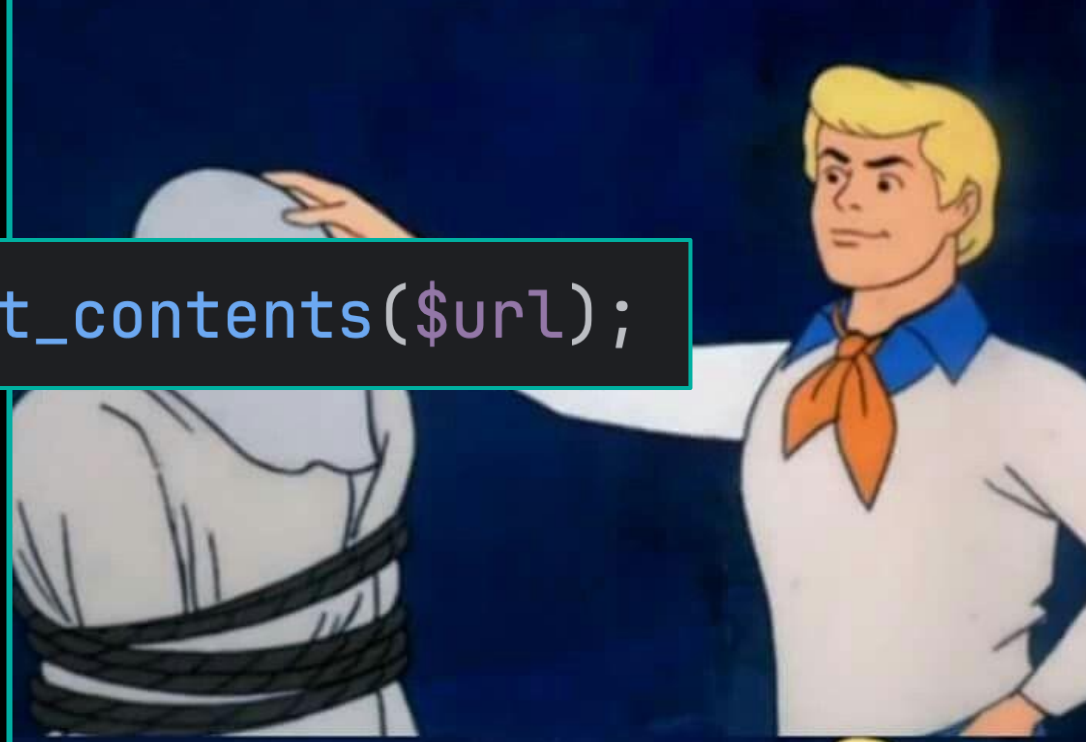
- Hosted on a fast connection
- Physically close to us



But wait…

```
file_get_contents($url);
```

```
file_get_contents($url);
```



```
smart_str_appends(&req_buf, src: " HTTP/1.1\r\n");
```

It's old man HTTP/1.1 !

Abstractions are great…

…until they're not!

- file_get_contents() is 1.1
- Use curl functions
- Use Guzzle (it uses curl if available)

or

- Contribute to PHP and add HTTP 2 and 3 support!

Summary

- Abstractions are great, until they're not
- Use HTTP/3 where possible
- Use a DNS provider supporting anycast
- Locate applications as close to other services as possible

Summary

- Abstractions are leaky
- Use HTTP/3 where possible
- Use a DNS provider supporting anycast
- Locate applications as close to services and users

It's always DNS

Unless it's BGP

Sometimes it's both

Thank you

https://phpc.social/@mintopia

https://github.com/mintopia/networking-fundamentals-2026

Slides