Hey! I’m Jedda.


I’m a CISSP and technology & information security consultant from Melbourne Australia, where I specialise in Apple platform security & PKI.


Outside of work I’m a girl dad and I love college football - *go Utes!*
It’s somewhere around 3 or 4am local time for me, so ☕ 🆙

📡 Network Relay - just a HTTP proxy?

- Spoiler alert; a Network Relay is a HTTP proxy
- So much more than that, and is the result of decades of iteration and convergence
- Rooted in transformative protocols and technologies such as QUIC and MASQUE

In this session:

- Talk a little about the history of HTTP, proxying and how this makes Network Relay possible
- Touch on and show how a Network Relay works, both technically and to a user in macOS
- Discuss the differences between traditional VPN and remote access and Network Relays

🐇 Network Relay - a classic rabbit hole topic

- End user experience & OS
- MDM deployment & lifecycle
- Network infrastructure, routing & protocols (TCP, UDP, QUIC)
- Server infrastructure (Relays are “web” servers and can be scaled as such)
- PKI & Cryptography (TLS, key exchange, trust & identity, ACME & managed device attestation)
- User privacy concepts (oblivious HTTP & DNS)
- Moving target - lots of ongoing work at the IETF (MASQUE working group)
- This session will touch on some pieces of these and zip right by others at speed
  - if this topic interests you, there is so much more to learn and discuss

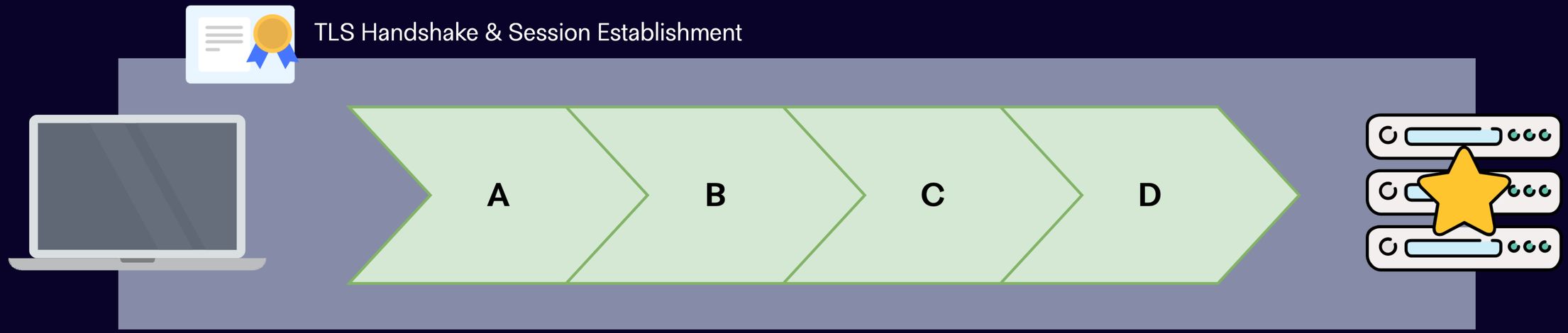
A *(very abridged)* history lesson on HTTP

Jedda

HTTP/1.x

- HTTP/1.0 originally introduced as a HyperText Transfer Protocol - primarily used to fetch static documents; text, images - one resource at a time
- HTTP/1.1 introduded connection persistence and the concept of pipelining
- Pipelining allowed queuing of resources within a single TCP connection to the an origin server
- HTTP/1.1 also first introduced the CONNECT verb which enabled early proxying scenarios

HTTP/1.x

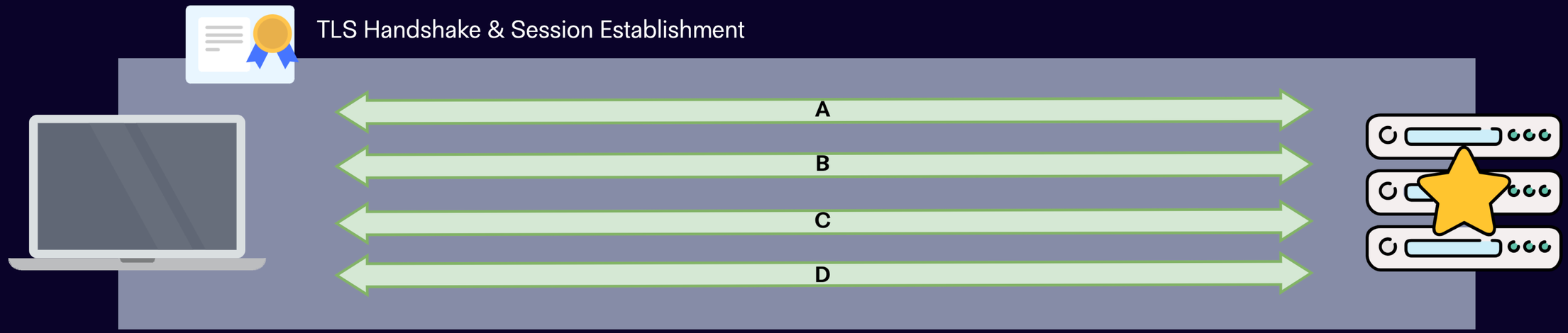


Single TCP connection with pipelined (queued) resources

HTTP/2

- Moved from an unstructured textual to a structured binary framed format
- Introduced multiplexing via individual streams on a single connection
- Multiplexing improves performance and solves many of the HoL blocking issues at the application layer
- HTTP/2 remains based on TCP and we still see HoL at the transport layer - this is exacerbated on lossy links

HTTP/2



Single TCP connection with multiplexed streams for resources

HTTP/3 & QUIC

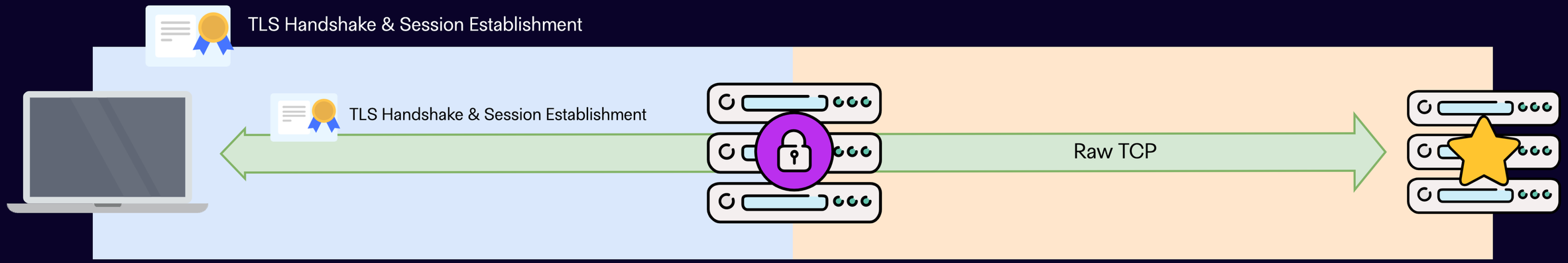
- HTTP/3 moves transport to QUIC:
  - Based on UDP which eliminates transport layer HoL blocking
  - Integrates and mandates TLS1.3 encryption
  - More advanced congestion control, true independence of streams & connection migration
  - Faster connection establishment through a single handshake (& session resumption)

- HTTP/3 more resilient than HTTP/2 but not always more performant due to user-space vs kernel implementations and CPU cost for additional cryptographic operations
- This will likely improve further over time as implementations are further optimised

🔐 What is a Network Relay?

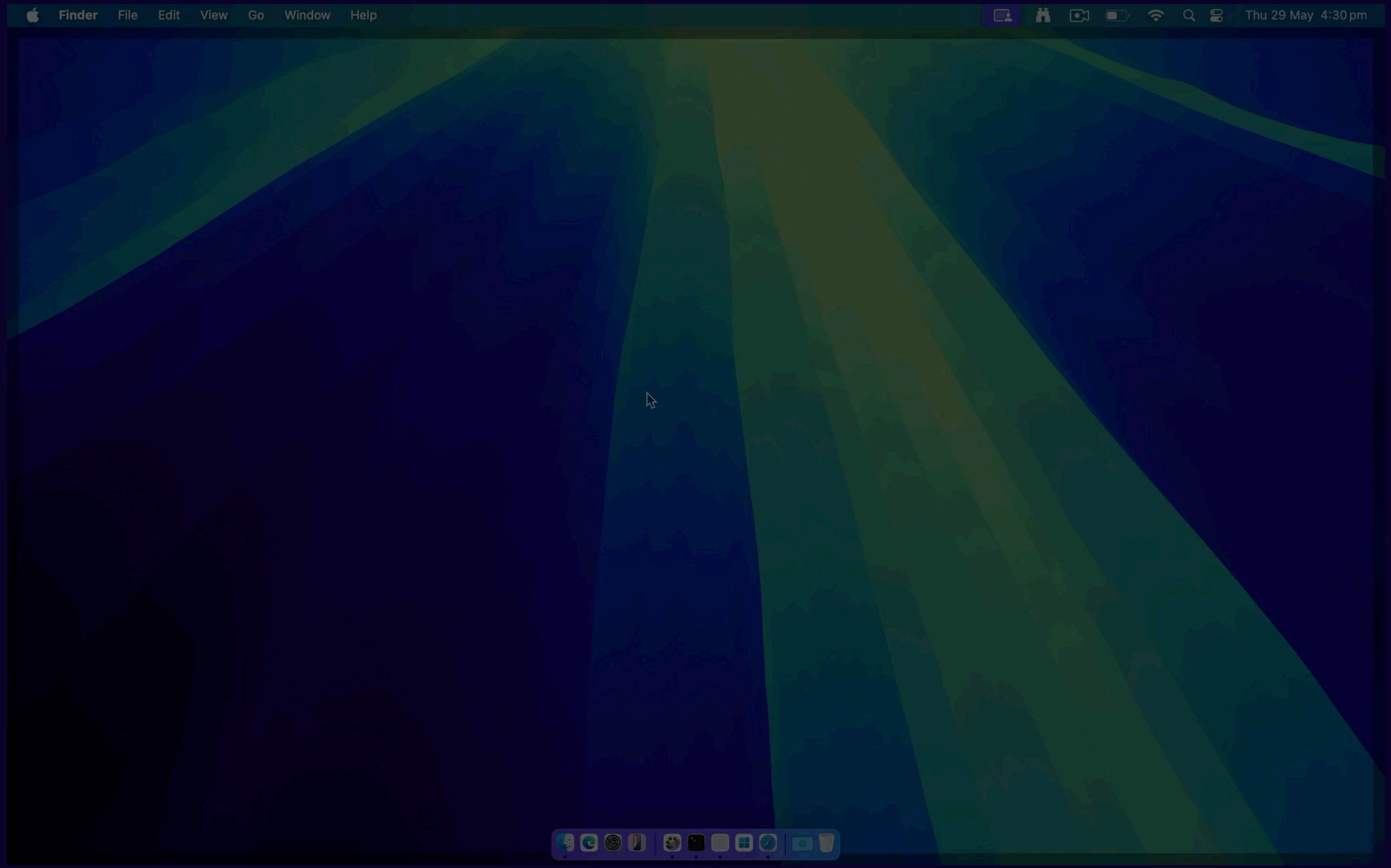
- A Network Relay is a HTTP proxy that supports both CONNECT (Extended) and MASQUE:
  - CONNECT used to tunnel TCP traffic (including HTTP/1.1 & HTTP/2)
  - MASQUE (CONNECT-UDP) used to tunnel UDP traffic (including QUIC & HTTP/3)
- Expected to run across HTTP/3 and QUIC with fallback to HTTP/2
  - By being transported using QUIC, gets many of the performance benefits for free
- OS matches traffic based on subdomains and FQDN - similar to VPN on Demand
- Can (& should) use certificate authentication for Client mutual TLS with the Relay
  - this includes ACME & Managed Device Attestation identities

What is a Network Relay?



"Outer" HTTP CONNECT Request (TCP or QUIC "Connection")

```
GET /.well-known/masque/udp/api-safari-aapse2c.smoot.apple.com/443/ HTTP/2
```


Finder File Edit View Go Window Help
Thu 29 May 4:30 pm

🔐 Traditional VPNs on Apple platforms

- Lots of examples of these, from OS integrated IPSec and IKEv2 to 3rd party agent based
- Often require interactive authentication from end users - some even require “self service” login to VPN portals to generate individual configuration files. Can be hard to deploy via MDM.
- For the most part, VPNs necessitate:
  - virtual network interface and tunnel establishment
  - session key agreement and re-keying
  - IP address allocation & changes to the system’s routing table
  - changes to the system’s DNS resolution - sometimes split
  - connection lifecycle; what comes up must come back down

💖 Why lean in and love Network Relay

- No IP addressing or routing table changes and management
- Built in security and encryption and agility via client-server negotiation
- Support for mTLS and attested device hardware identity for authentication
- Great performance and durability with QUIC - particularly on lossy networks
- Can work alongside other relays or even other VPNs
- Flexible traffic matching based on domain names
- Built in support for “per-App” relay across mixed deployment models
- Where the puck is going - Apple investing in this tech across their ecosystem and at a standards level. Lots of contribution at the IETF and ongoing work

os26 - Post-quantum cryptography

- iOS/iPadOS/macOS & visionOS 26 support the `X25519MLKEM768` algorithm for hybrid key exchange in TLS1.3 (via supported_groups in ClientHello)
- Requires server support (now supported in OpenSSL and BoringSSL)
- Used by default for all URLSession, and Network.framework connections (which make up the vast majority of network calls)
- Supported by iCloud Private Relay, Network Relay, DNS over HTTPS & other system services
- Supported by Envoy (needs explicit config via TLS `ecdh_curves`)

Apple Support - Prepare your network for quantum-secure encryption in TLS
https://support.apple.com/en-us/122756

Jedda

os26 - Post-quantum cryptography

```
% nscurl --tls-diagnostics https://relay.credibly.cc --http3 --http3-prior-knowledge
Starting TLS Diagnostic

================================================================================

Negotiated TLS version (codepoint): 0x0304
Negotiated TLS key exchange group (name): X25519MLKEM768
Negotiated TLS ciphersuite (codepoint): 0x1302

================================================================================
```

Apple Support - Prepare your network for quantum-secure encryption in TLS
https://support.apple.com/en-us/122756

Jedda

os26 - Post-quantum cryptography

```
## Envoy Stats (':9901/stats' admin listener)

listener.0.0.0.0_443.ssl.ciphers.TLS_AES_128_GCM_SHA256: 2
listener.0.0.0.0_443.ssl.ciphers.TLS_AES_256_GCM_SHA384: 15
listener.0.0.0.0_443.ssl.curves.X25519MLKEM768: 17
listener.0.0.0.0_443.ssl.handshake: 17
listener.0.0.0.0_443.ssl.versions.TLSv1.3: 17
```

Apple Support - Prepare your network for quantum-secure encryption in TLS
https://support.apple.com/en-us/122756

Jedda

🪄 Interested in learning more?

- My recent article includes: more of the history and protocol fundamentals,
- Swift's NERelay and NERelayManager vs `com.apple.dnsSettings.managed`,
- Intricacies around DNS flows and traffic matching,
- Practical examples of an open source Network Relay using Envoy (with more demos), and
- Some additional commentary on performance, privacy, obsfuctation and deployment models.
- `#network-relay` channel on the MacAdmins Slack - great place to discuss this emerging topic!

Jedda Wignall

Beneath the MASQUE - a dive into Network Relay technology on Apple platforms

https://jedda.me/beneath-the-masque-network-relay-on-apple-platforms/

Jedda

Thank you!

Questions?

Jedda Wignall

Beneath the MASQUE - a dive into Network Relay technology on Apple platforms

https://jedda.me/beneath-the-masque-network-relay-on-apple-platforms/