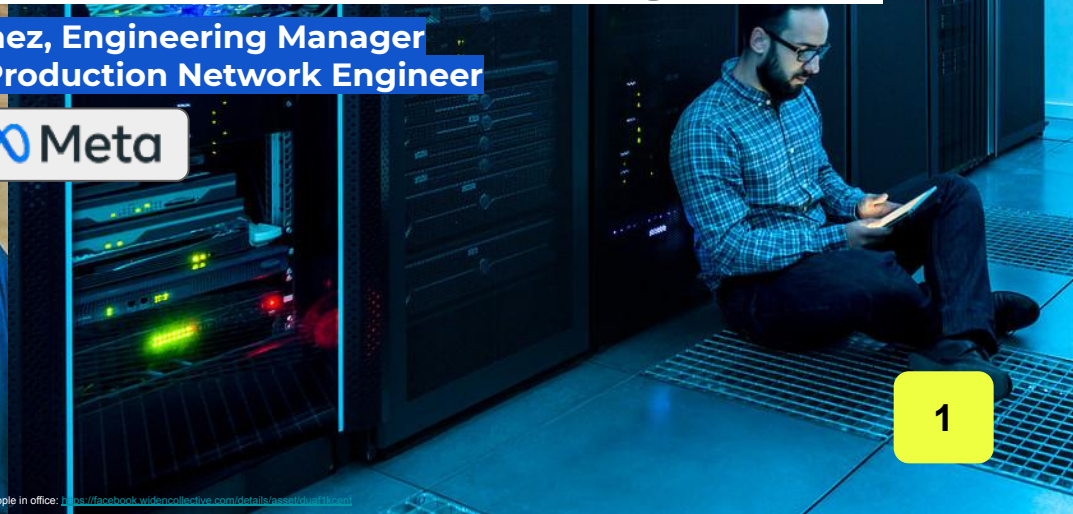




One evolution of the Network Engineer

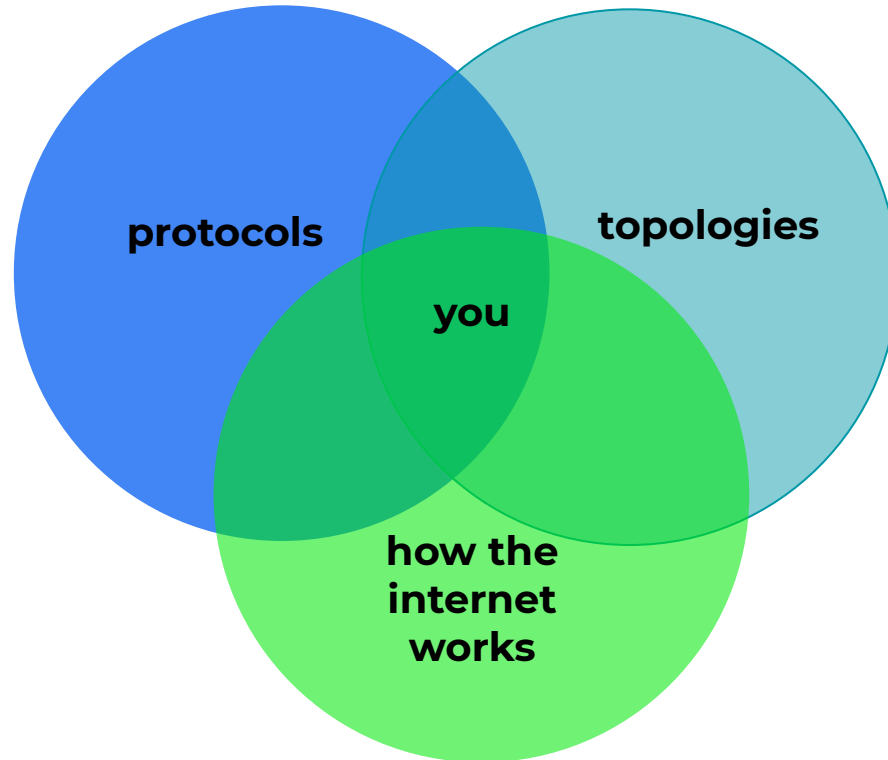
Elena Sanchez, Engineering Manager
& Jose Leitao, Production Network Engineer



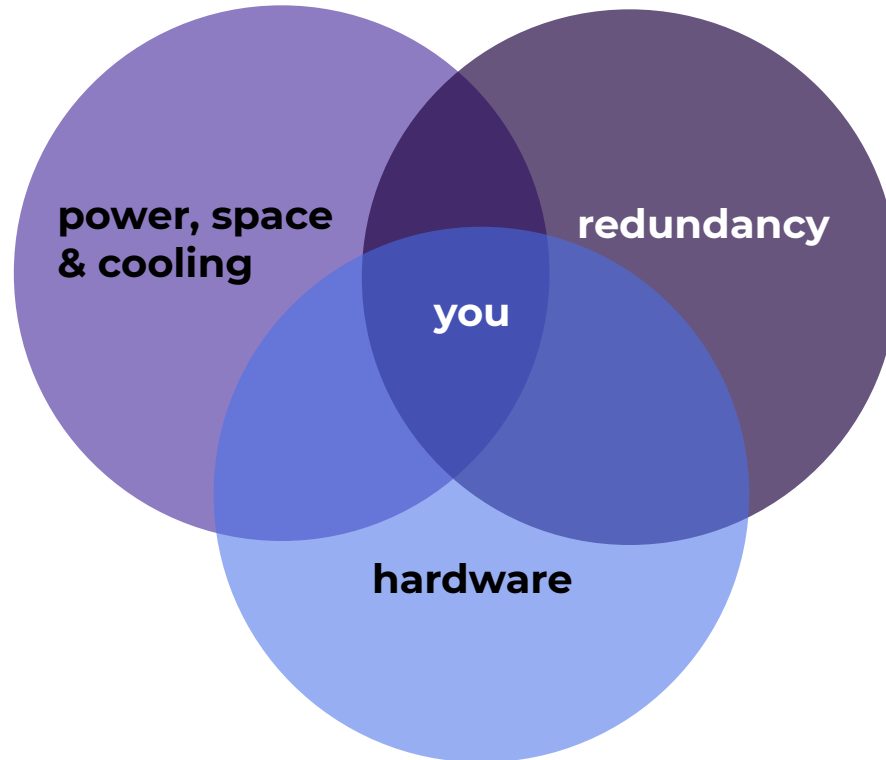
Starting a career as a network engineer



Network knowledge



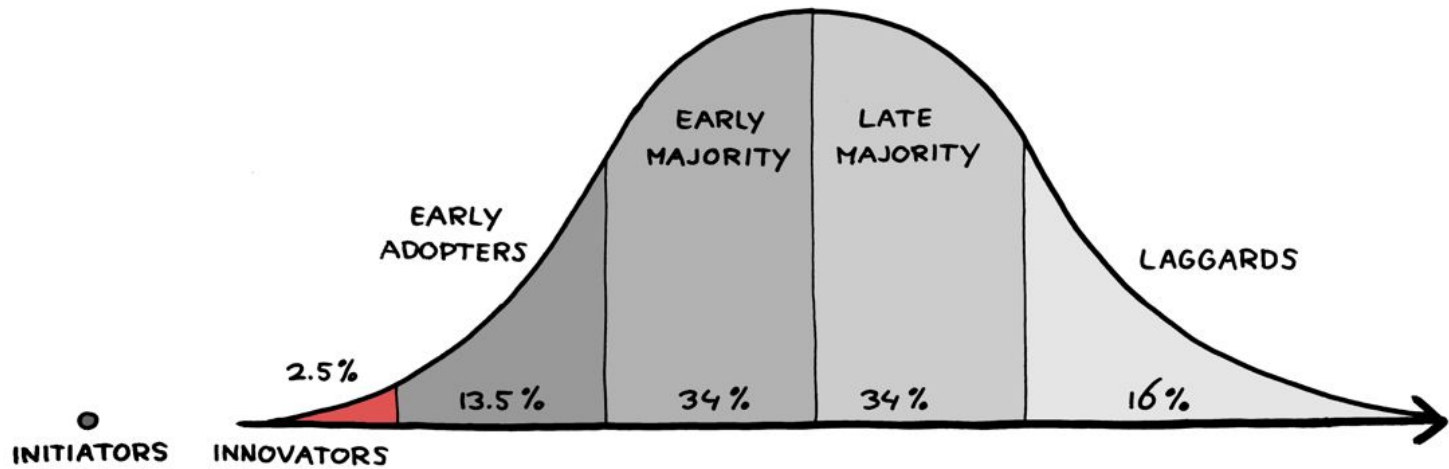
Physical infrastructure understanding



What about working at **scale?**

“Big networks have fixed everything”





Vendor options

Interacting with network gear

- JSON/XML via CLI
- NETCONF / SNMP
- REST APIs
- Thrift / gRPC
- TL1
- Streaming Telemetry

```
let regex = /^(\\d{1,2}-){2}\\d{2}(\\d{2})?$/;  
console.log(regex.test('01-01-1990'));  
// true  
console.log(regex.test('01-01-90'));  
// true  
console.log(regex.test('01-01-190'));  
// false
```



More examples

Upgrading



More examples

Probing systems

What are the **gaps** to bridge?

How these gaps manifest?

**Software development
not being a core
component of people
day-to-day work**

How these gaps manifest?

**Software development
not being a core
component of people
day-to-day work**

**Struggles with writing
reliable code or solutions to
be deployed at scale**

How these gaps manifest?

**Having surface or
incomplete knowledge
on the fundamentals
of ARP, TCP/IP, DNS,
BGP, MPLS, OSPF**

How these gaps manifest?

Having surface or incomplete knowledge on the fundamentals of ARP, TCP/IP, DNS, BGP, MPLS, OSPF

Struggles with applying the theory when proposing or scaling solutions

How these gaps manifest?

**Manual monitoring,
reactive troubleshooting
and manual fixes
(because it's faster!)**

How these gaps manifest?

**Manual monitoring,
reactive troubleshooting
and manual fixes
(because it's faster!)**

**Missed opportunity to
introduce more
standardisation, device
data gathering and
automation (yes, there's an
upfront cost)**

Ideas to **bridge the gaps**

Ideas to bridge the gaps

Making automation a core part of the role (as much as possible)

Getting familiar with unit and integration testing (to deploy reliable code)

Ideas to bridge the gaps

Getting comfortable with languages like Python, C++, Rust, Golang

Understanding the use cases in which a language can better help with automation, data collection, data processing, etc.

Ideas to bridge the gaps

In-depth understanding of routing protocols like BGP (or your favourite one)

Ensuring your knowledge covers common use cases, limits, and (if possible) getting to work with it in the real world

Some (good) books

Clean code
by Robert
Cecil Martin

**How to Measure
Anything** by
Douglas W.
Hubbard

**Think Python: An
Introduction to
Software Design** by
Allen B. Downey

**Network Programmability
and Automation** by Jason
Edelman, Matt Oswalt, and
Scott S. Lowe

**Our approach to solving this gap:
launching career defining programs**



**Developing talent
through the
Rotational Network
Engineering Program**



Boost a
career's initial
progression



Finding raw talent & showing them a glimpse of our complex tech challenges (via internships)

How do we do this?

6 to 9 months

Tasks

Project



We also focus on developing core skills

**Time management,
documenting, communications,
public speaking, interviewing,
giving & receiving technical &
non-technical feedback**



**Modelling success through
career mentors**

**Informing decisions
towards specialisation**

A night cityscape with glowing white arcs in the sky and a central 'Thanks!' text box. The background shows a dense urban skyline with illuminated buildings and streets. Overlaid on the sky are several white, glowing arcs that intersect and curve across the frame. In the center, a white rectangular box contains the word 'Thanks!' in a large, bold, black sans-serif font.

Thanks!