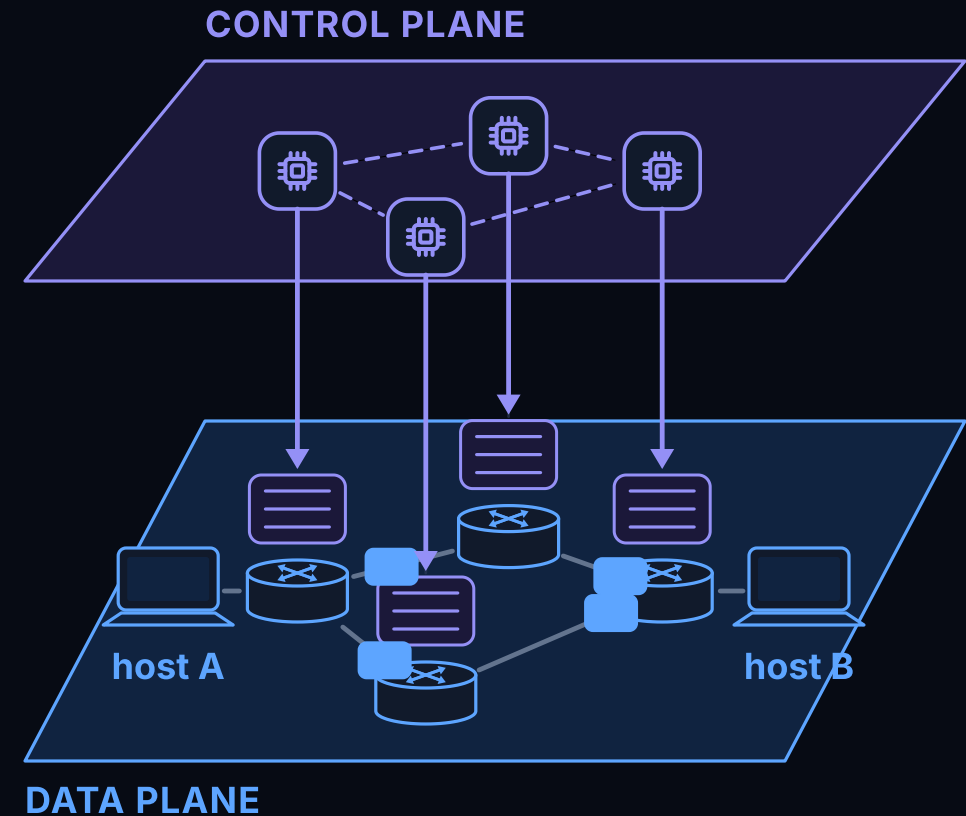


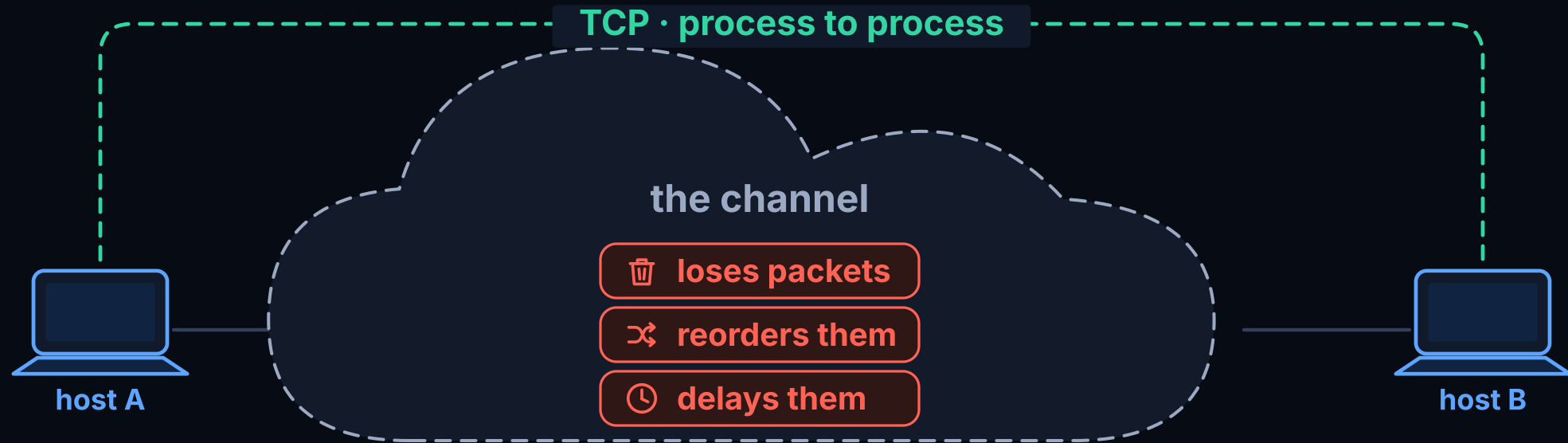
The Network Layer: An Overview

Chapter 3 built TCP on a channel it never looked inside. This lecture opens the channel: the one layer every machine on the path runs, the two jobs it does, and the two planes those jobs live on.

Yaxiong Xie



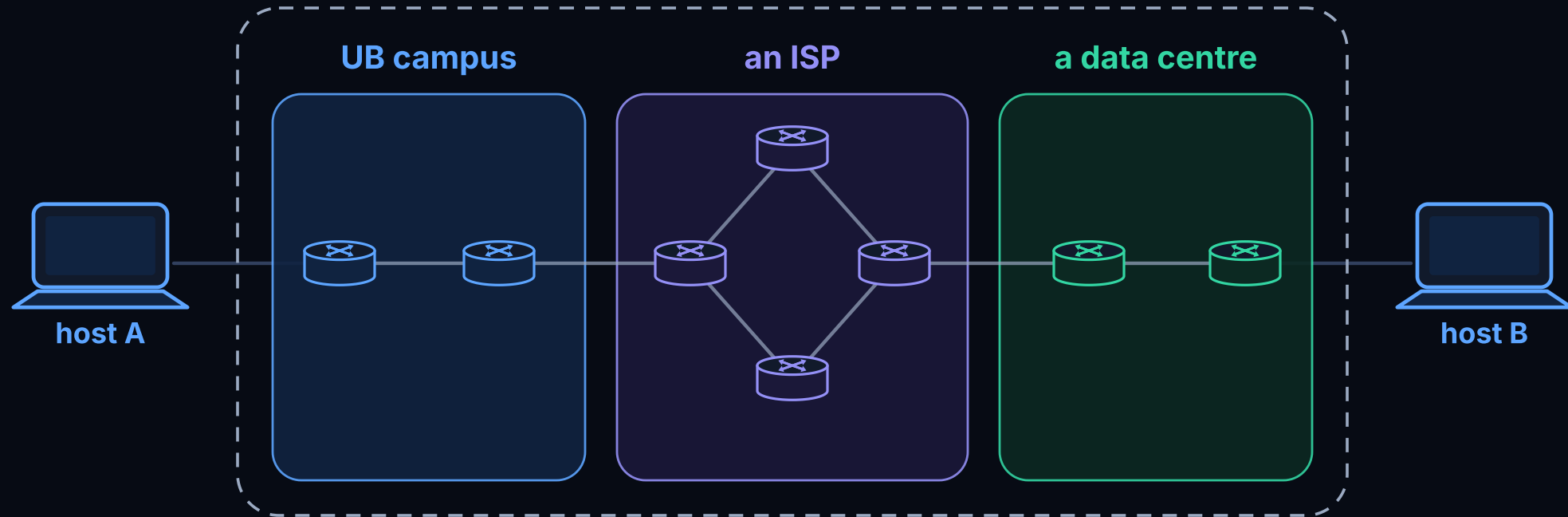
Chapter 3 treated the channel as one box



"The Internet is a series of tubes." A U.S. senator, 2006. From where TCP stands, it is.

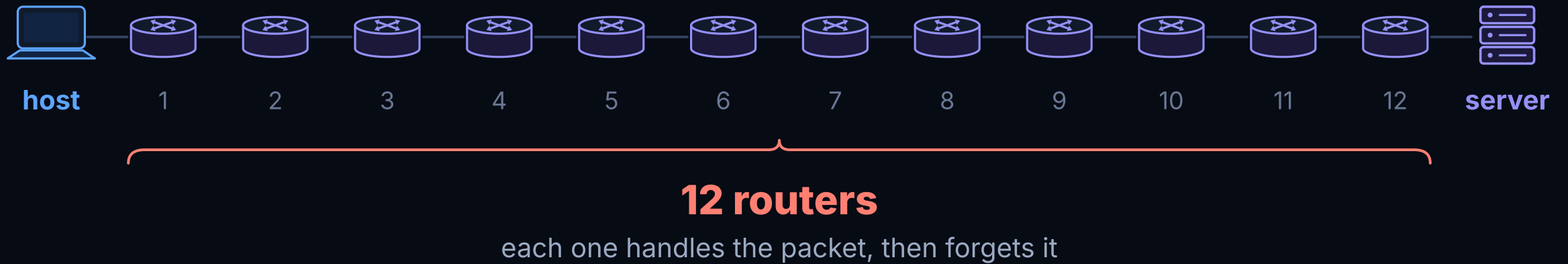
TCP was built on top of a channel nobody looked inside of.

What is inside the box?



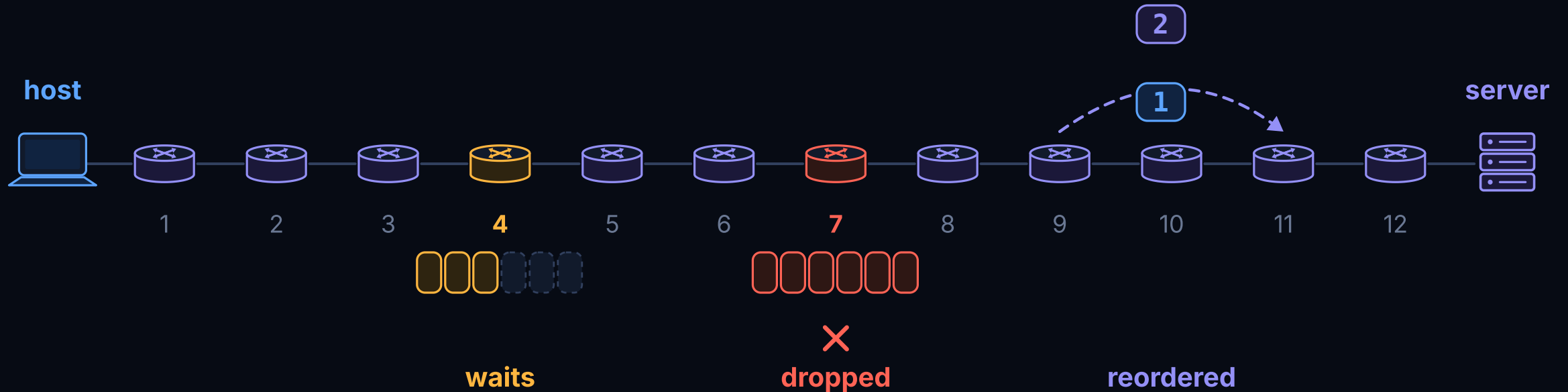
Routers and links, run by several organisations. Every packet crosses them all.

One packet, twelve machines, one hop at a time



TCP talks to the far host. Twelve machines carry the packet, and TCP sees none of them.

Any one of them can drop, delay or reorder the packet



The channel Chapter 3 fought is made of these machines. This chapter opens them.

Three questions, three parts

01



The layer

What does it do, and
on which machines?

02



Two jobs

What does a router do,
and who tells it what?

03



Two planes

How are the two jobs
kept apart, and joined?

This lecture is the map. The lectures after it open one part each.

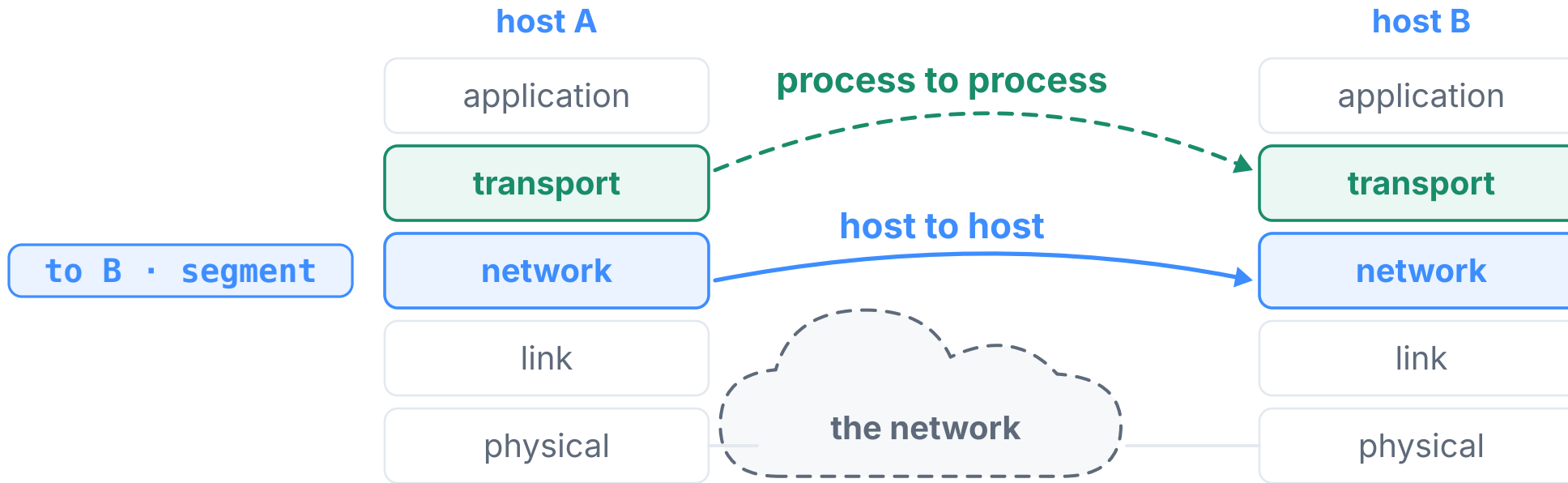
What does the network layer do, and where?

01

The layer

Chapter 3 moved a segment from one process to another. Between those two processes lie a dozen machines that TCP never sees. This part is about the layer that runs on all of them.

Transport goes process to process. Who goes host to host?



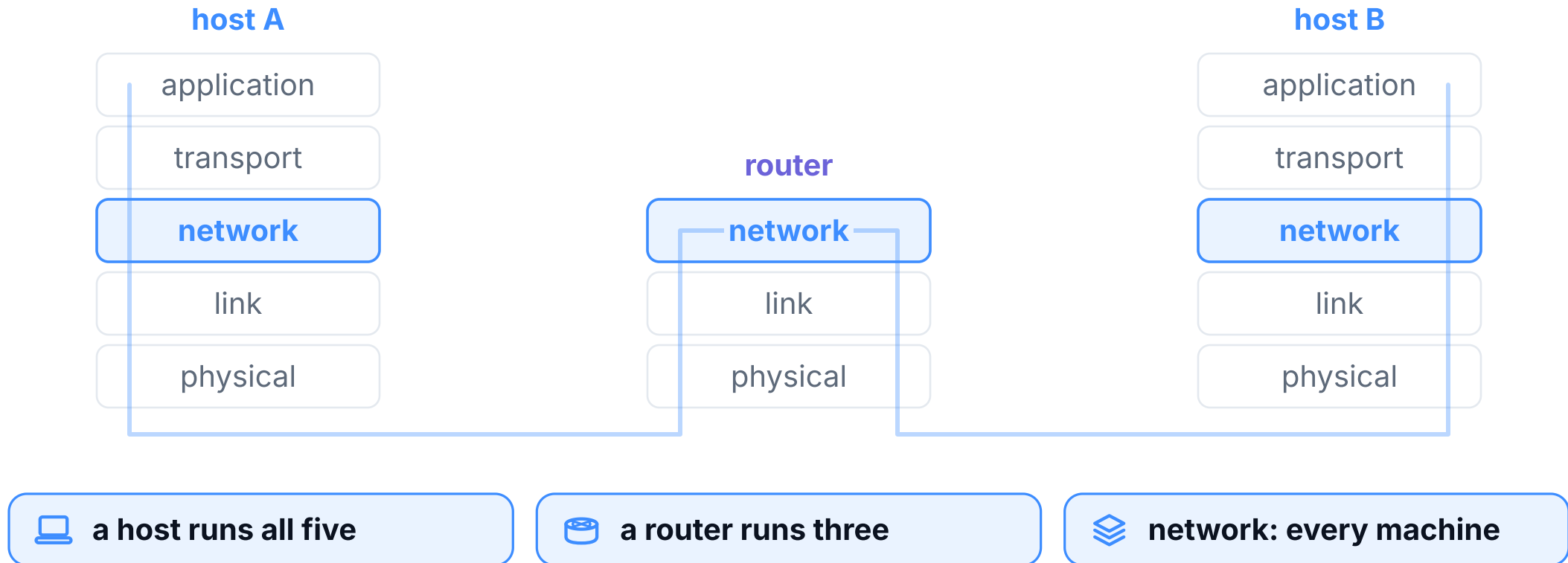
The network layer carries a datagram host to host, across the whole network.

1 The layer

2 Two jobs

3 Two planes

Which machines run the network layer?



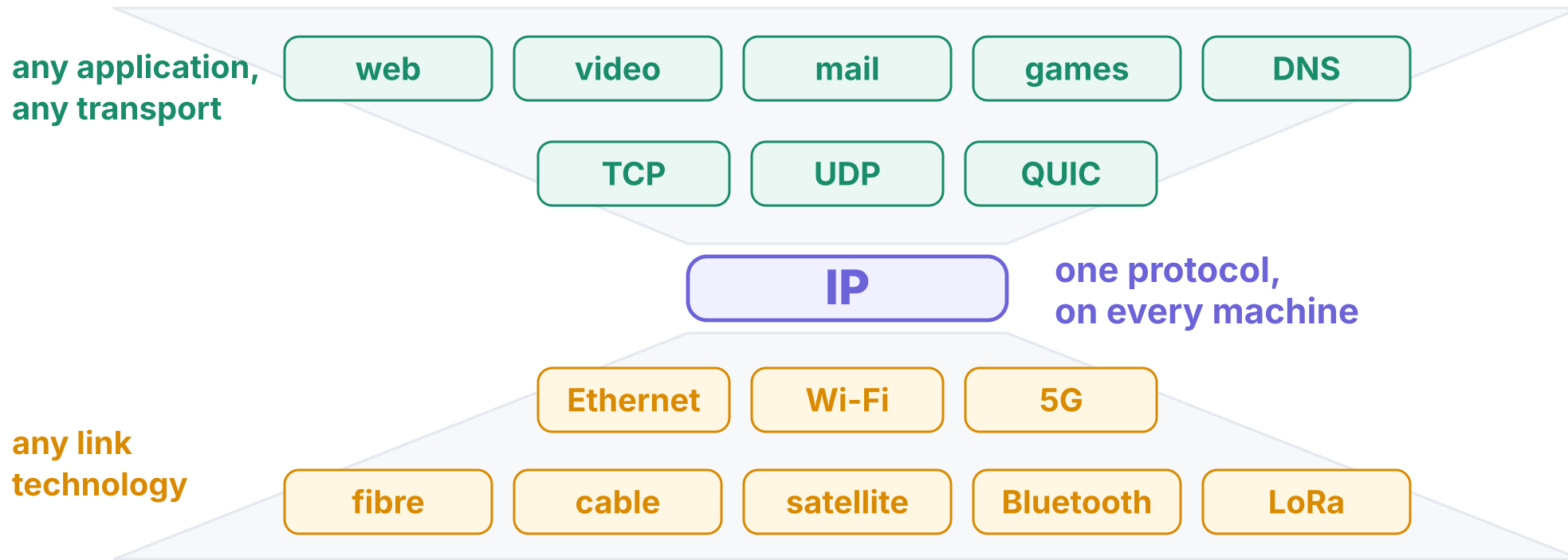
Transport runs on the two ends. The network layer runs on every machine on the path.

1 The layer

2 Two jobs

3 Two planes

What do all these machines have in common?



IPv4 was fixed in 1981 (RFC 791). Everything above and below it has been replaced several times since. IP has not.

Everything above and below varies. IP is the one thing every machine shares.

1 The layer

2 Two jobs

3 Two planes

What does IP promise?



- ✗ Will it arrive? no promise
- ✗ In the order it was sent? no promise
- ✗ Within some bound on time? no promise

best effort

that is the whole promise

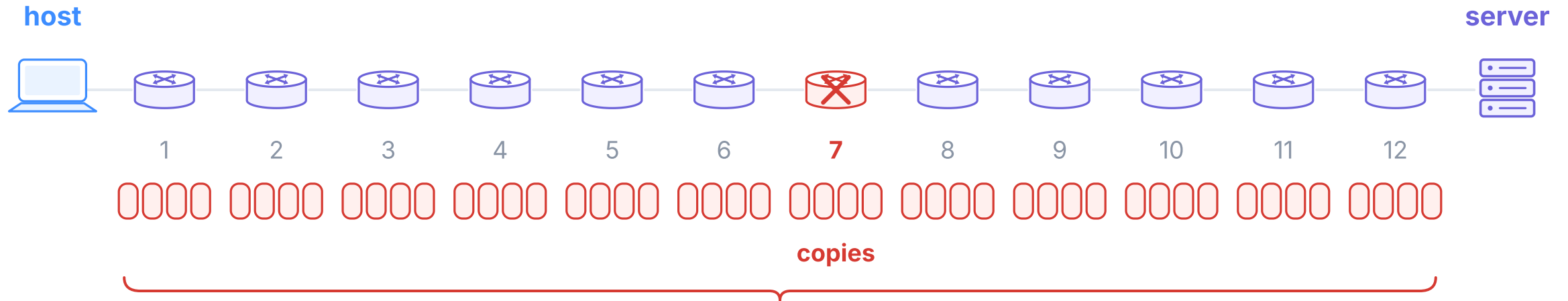
Best effort: IP tries, and promises nothing. Loss, reordering and delay are all allowed.

1 The layer

2 Two jobs

3 Two planes

Suppose IP did promise delivery. Who resends a lost packet?



and so would all twelve, for every connection passing through them: millions at a time
memory and bookkeeping per connection, in every router, growing with the network

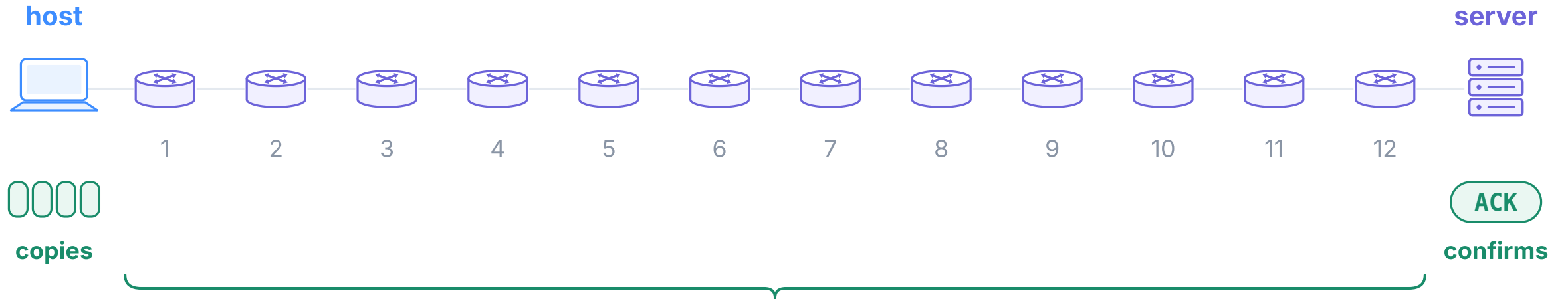
A promise inside the network means every router remembers every connection.

1 The layer

2 Two jobs

3 Two planes

Put the promise at the ends instead



host A keeps the copies and resends; host B confirms. That is TCP, Chapter 3.

the twelve in between remember nothing: no memory per connection, no bookkeeping, no limit on how many

TCP and IP were one protocol until 1978. Splitting them put the promise into TCP, at the ends, and left IP with nothing to remember.

Two machines do the work; the routers stay simple. That is why the Internet scales.

1 The layer

2 Two jobs

3 Two planes

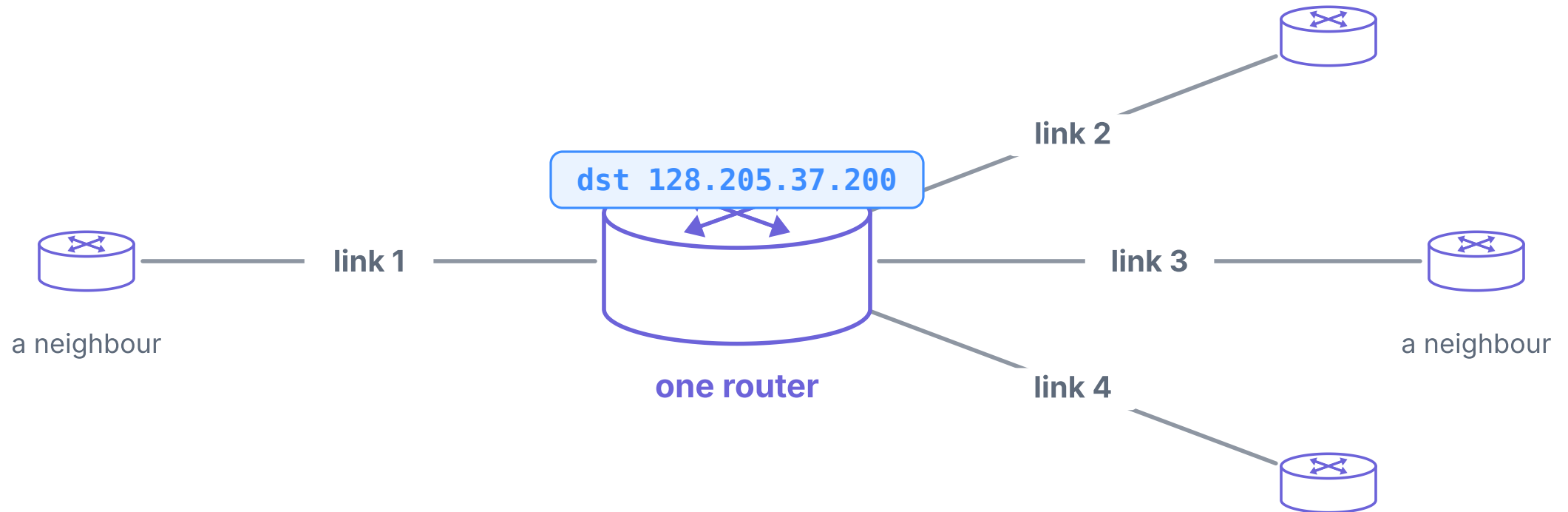
What happens to a packet inside one router?

02

Two jobs

A packet arrives at a router on one link and has to leave on one of several. This part is about the two jobs behind that: choosing the link for each packet, and working out beforehand which link is the right one.

A router has several links. A packet arrives on one of them.



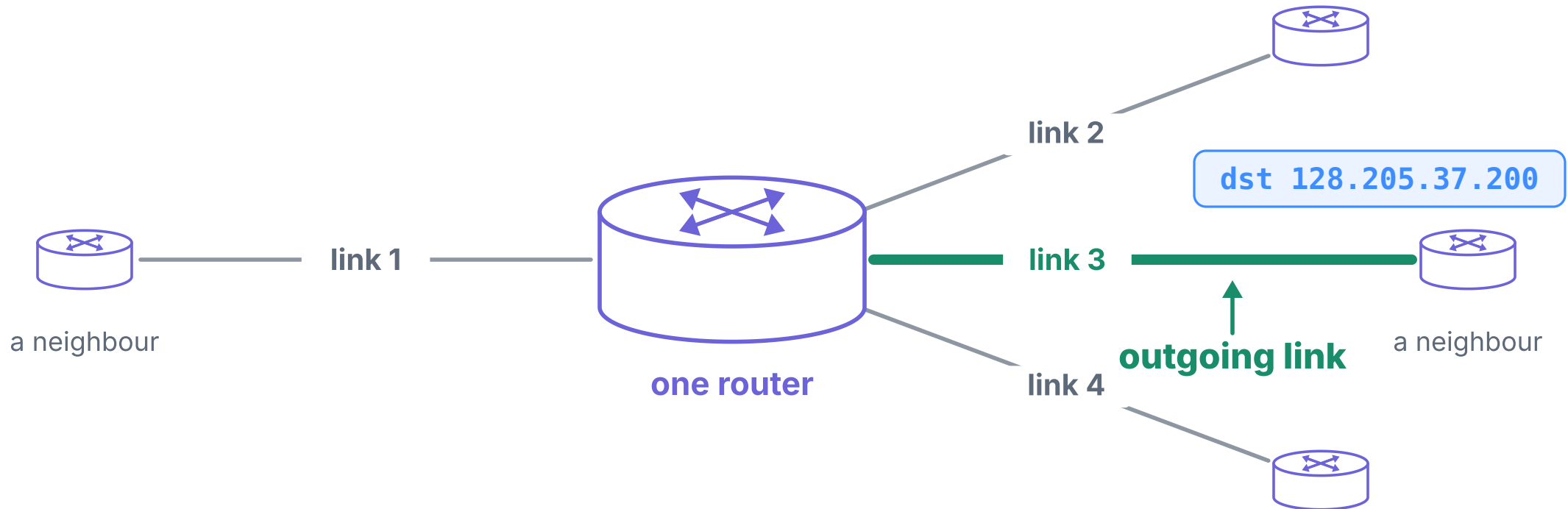
Four links, one packet. The router has to put it on one of the other three.

1 The layer

2 Two jobs

3 Two planes

The router's job: destination IP in, outgoing link out



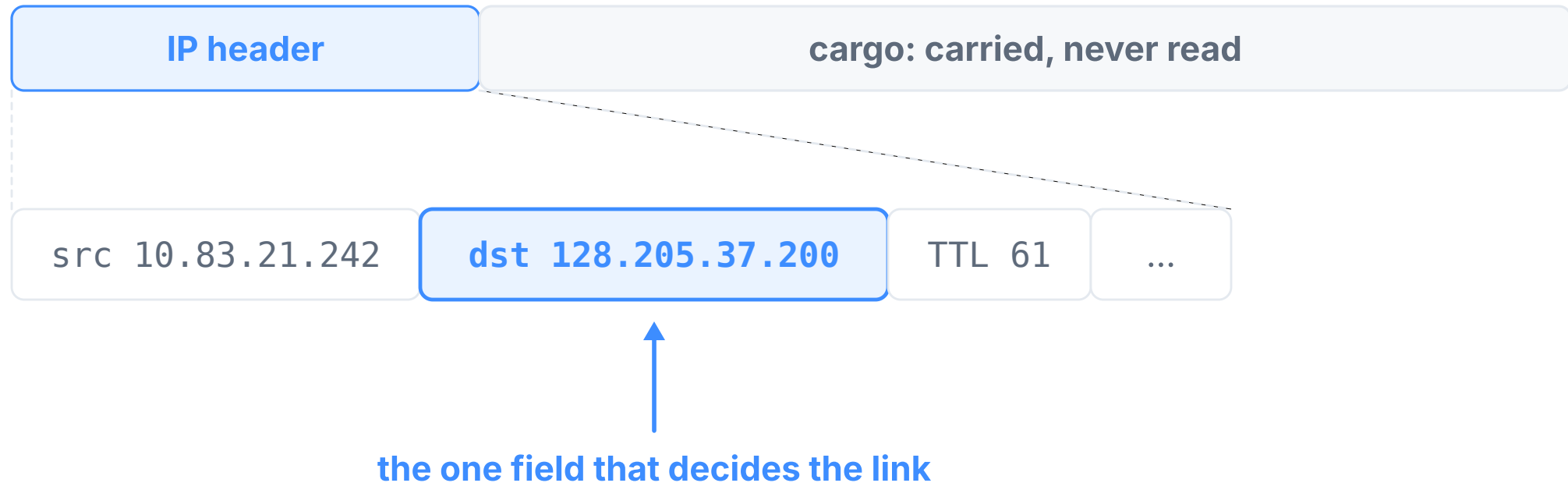
Destination IP in, outgoing link out: that is all a router does with a packet.

1 The layer

2 Two jobs

3 Two planes

What does the router have to go on?

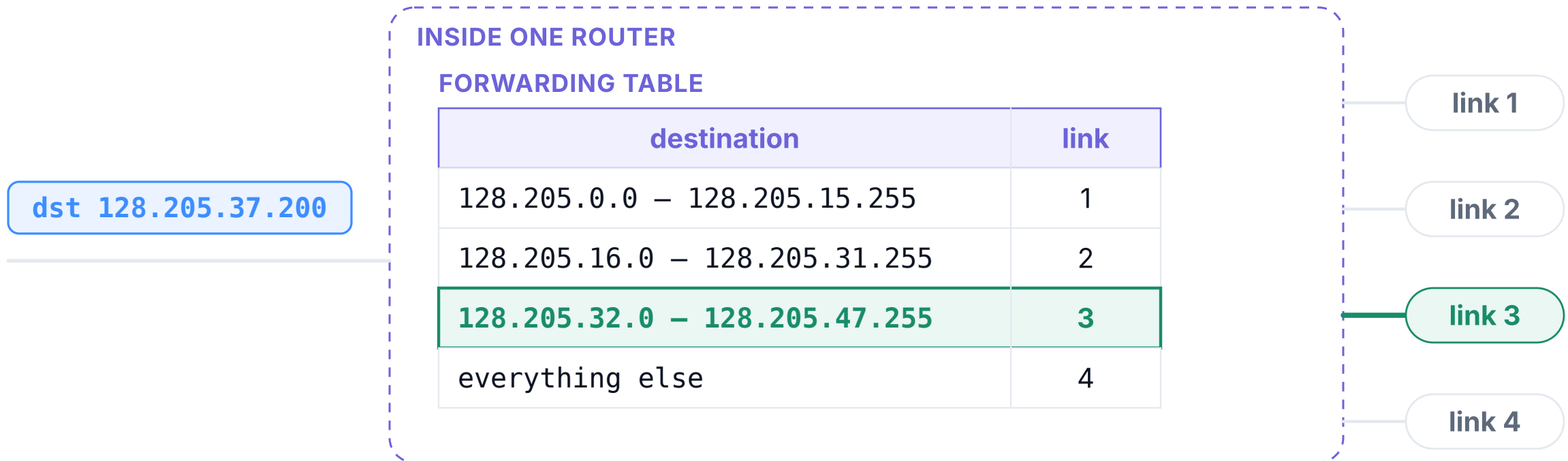


Like a post office: it reads the address on the envelope and never opens the letter.

One field decides the link: the destination address. Everything behind it is cargo.

- 1 The layer
- 2 Two jobs
- 3 Two planes

The forwarding table turns an address into a link



Forwarding: look the destination up in the table, send the packet out on the link it names.

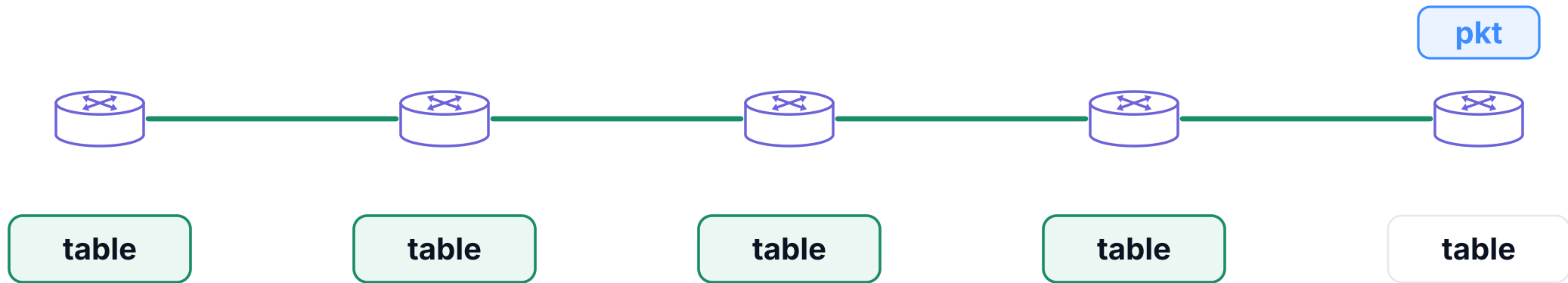
1 The layer

2 Two jobs

3 Two planes

Every router does the same, one hop at a time

ONE PACKET, FIVE ROUTERS · PRESS → FOR EACH HOP



each router reads only its own table, then forgets the packet

No table holds the path. The path is a chain of independent next-hop decisions.

1 The layer

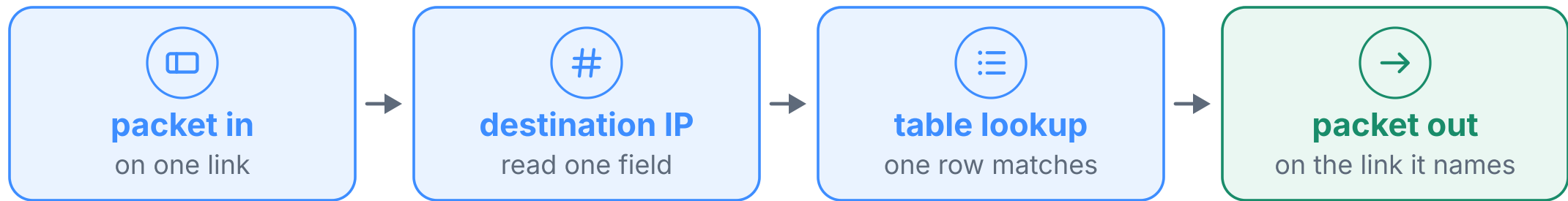
2 Two jobs

3 Two planes

So this is forwarding

Forwarding

for each packet, pick the outgoing link from its destination address



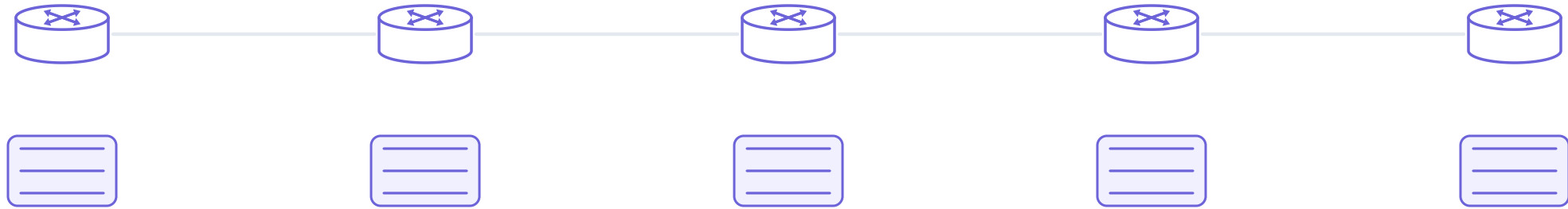
local: a router looks at its own table and its own links, nothing beyond

Forwarding is local: one router, one packet, one lookup, one outgoing link.



Who filled the tables in?

THE SAME FIVE ROUTERS · PRESS → TO WATCH THEM TALK



each router works out its own rows from what it heard

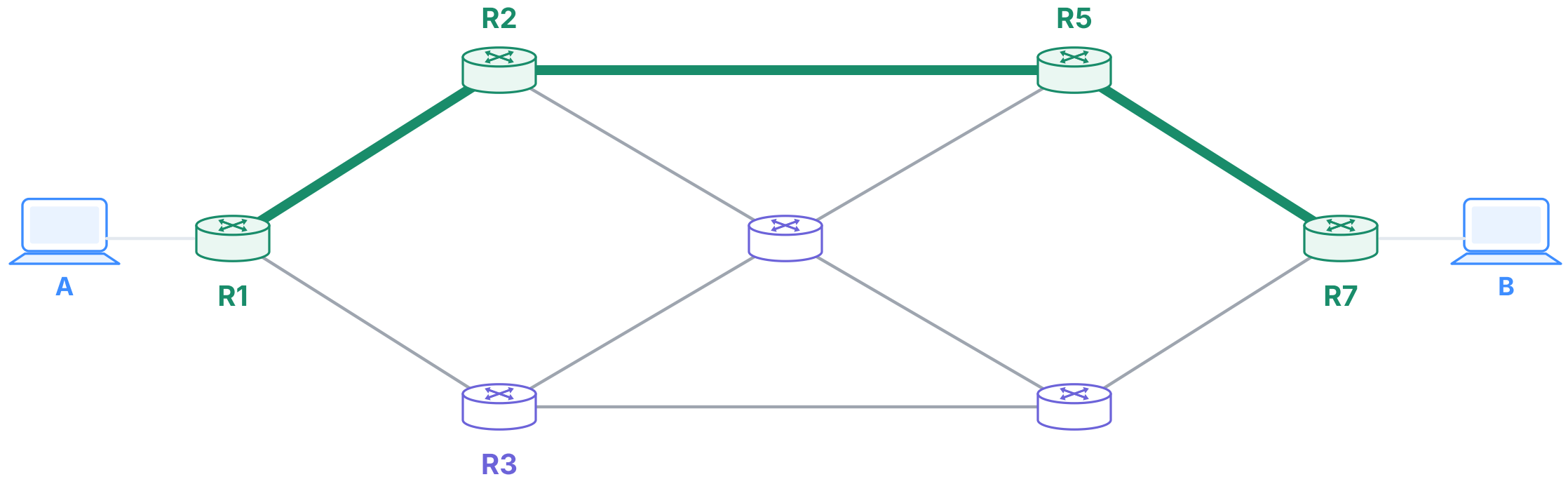
Routing: routers tell each other what they know, and each computes its own table.

1 The layer

2 Two jobs

3 Two planes

What must routing see that forwarding never does?



only then can it pick the path, and fill in: "for B, send to R2"

Forwarding sees one packet and one table. Routing has to see the whole network.

1 The layer

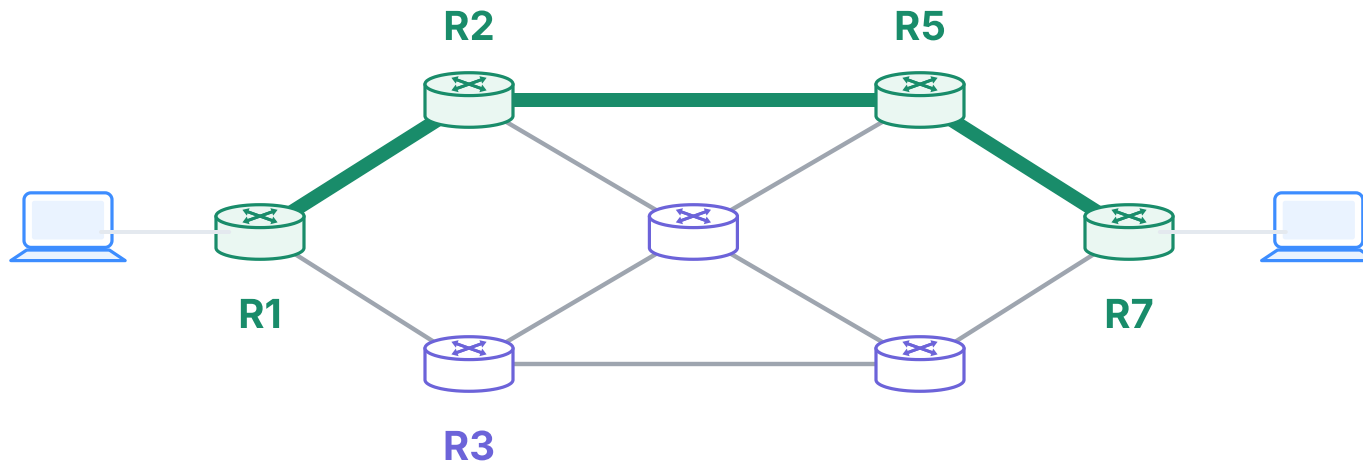
2 Two jobs

3 Two planes

So this is routing

Routing

choose the path a packet should take from its source to its destination



the goal

a good path: short, and around failed links



the scope

the whole network, not one router (nor the sender)



the answer

one next hop in every table along the path

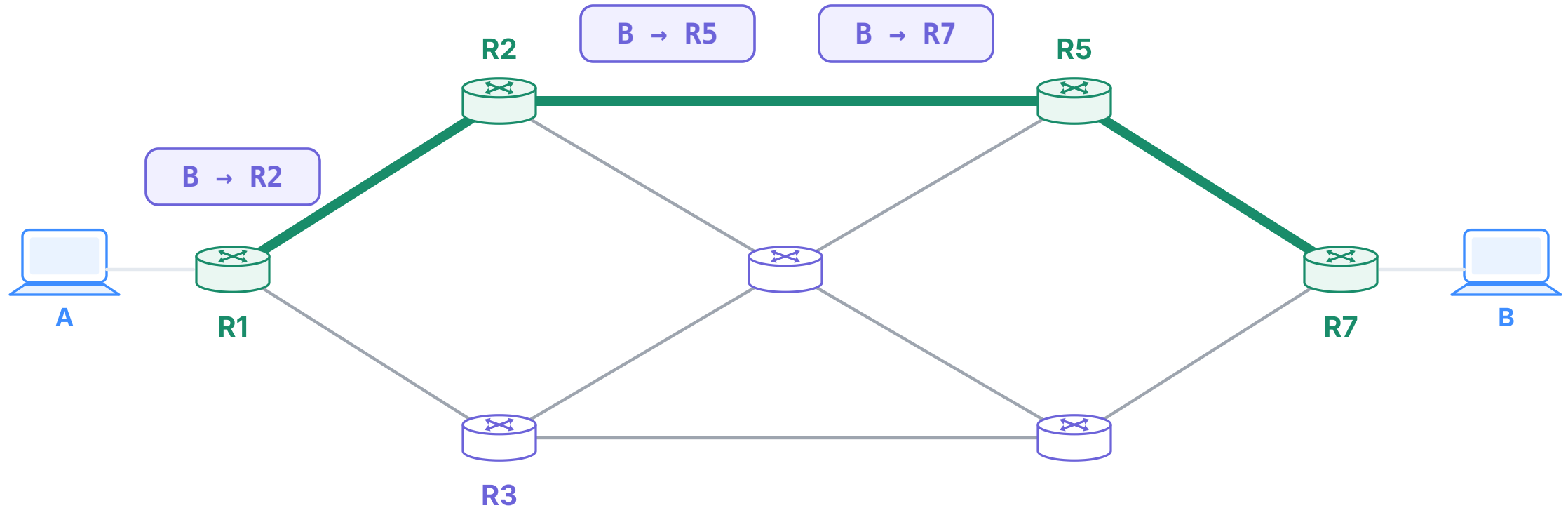
Routing is global: it needs the whole map, and its answer is written into the tables.

1 The layer

2 Two jobs

3 Two planes

Routing writes the path down. Forwarding walks it.



forwarding reads one row at a time, and the packet walks the path

Routing stores the path as next hops; forwarding reads one of them per router.

1 The layer

2 Two jobs

3 Two planes

Two jobs, two very different scales

	forwarding	routing
looks at	 one packet, one table	 the whole network
runs	 for every packet, in nanoseconds	 when something changes, in seconds
built as	 hardware	 software

Forwarding: one packet, one router, nanoseconds. Routing: the whole network, seconds.

1 The layer

2 Two jobs

3 Two planes

The same split on the road



routing: plan the trip on the whole map
once, before leaving



forwarding: at each exit, read the sign
every time, at full speed

Routing plans the trip once. Forwarding takes one exit at a time, without stopping.

1 The layer

2 Two jobs

3 Two planes

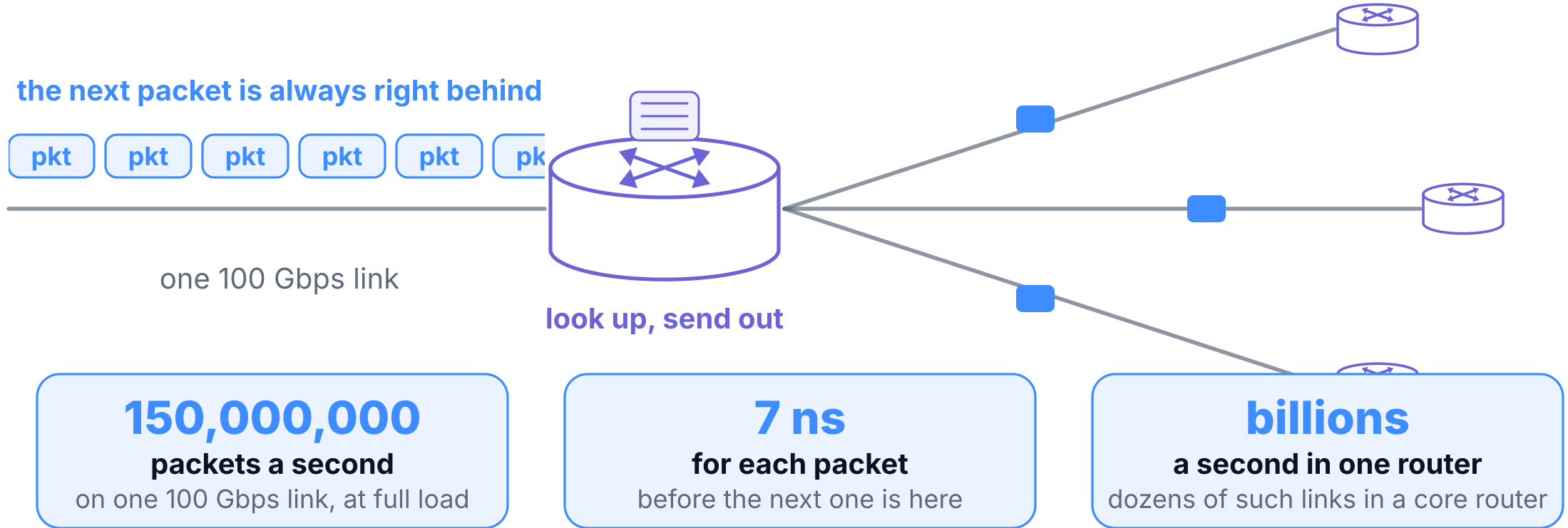
Two jobs in one router. Can the same machine do both?

03

Two planes

Forwarding is needed for every packet, right away. Routing runs in the background, on a clock of seconds and minutes. This part is about what follows from that difference.

Forwarding: every packet, right away



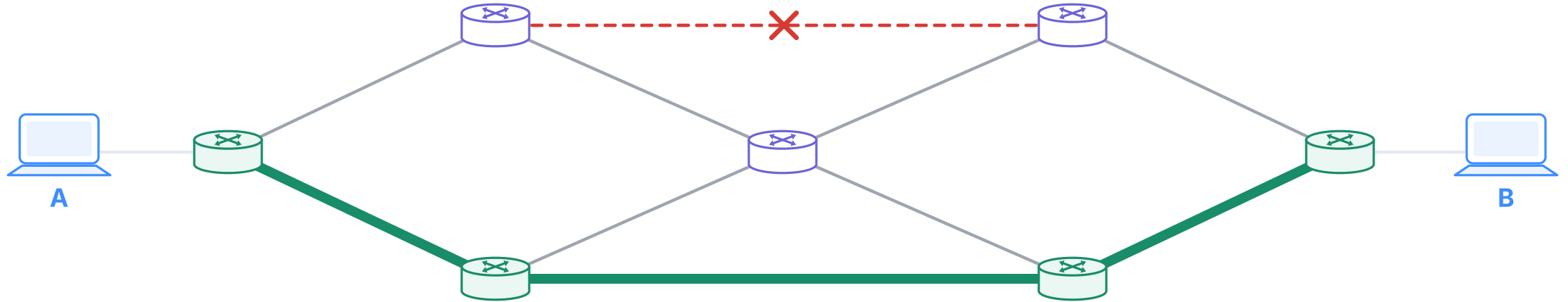
Forwarding has to be fast: a lookup per packet, 150 million times a second, no waiting.

1 The layer

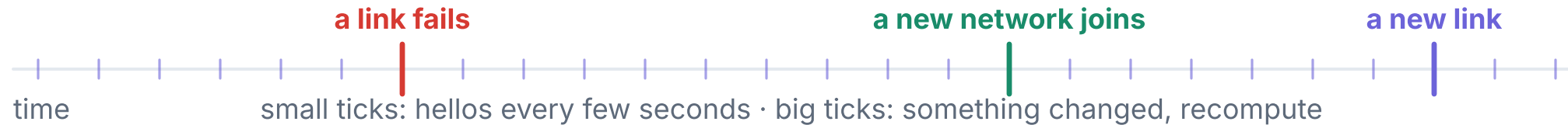
2 Two jobs

3 Two planes

Routing: in the background, on its own clock



a link fails: new paths are worked out and the tables rewritten, in seconds



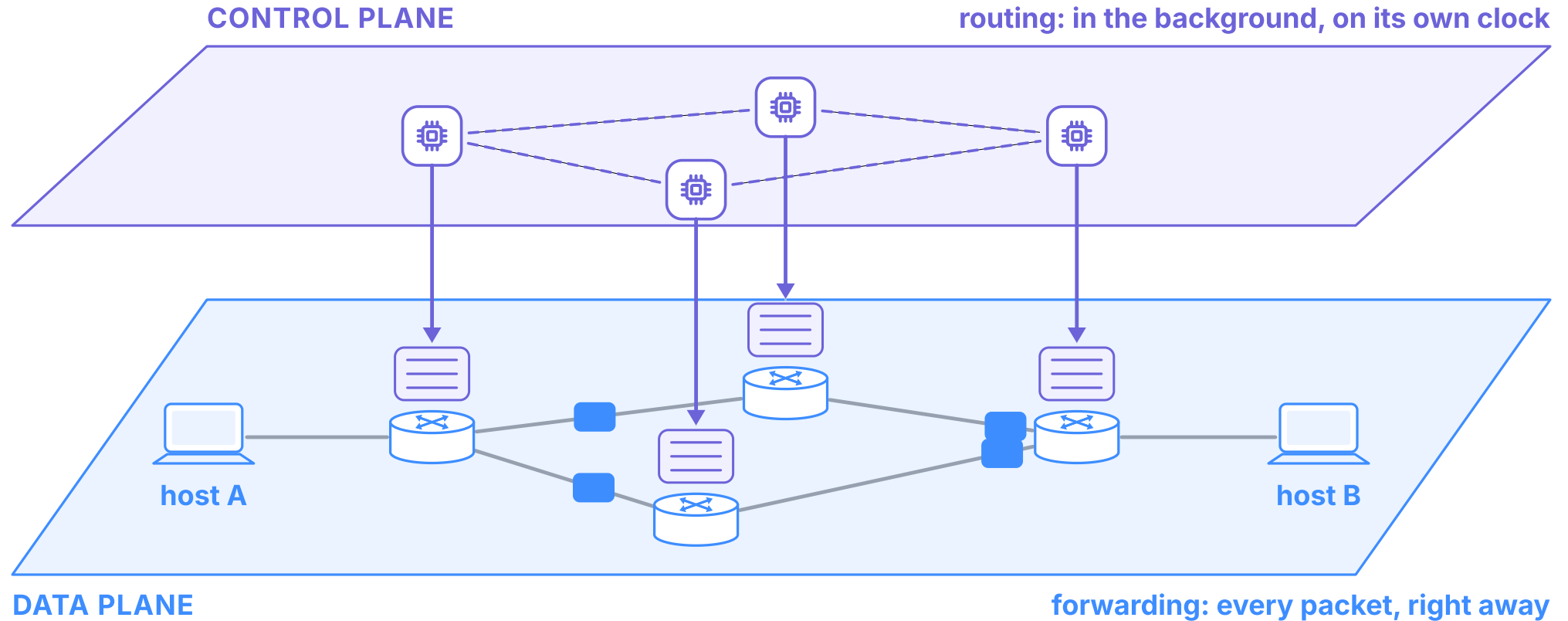
Routing works all the time, on a clock of seconds and minutes, never per packet.

1 The layer

2 Two jobs

3 Two planes

Two very different jobs, so two planes



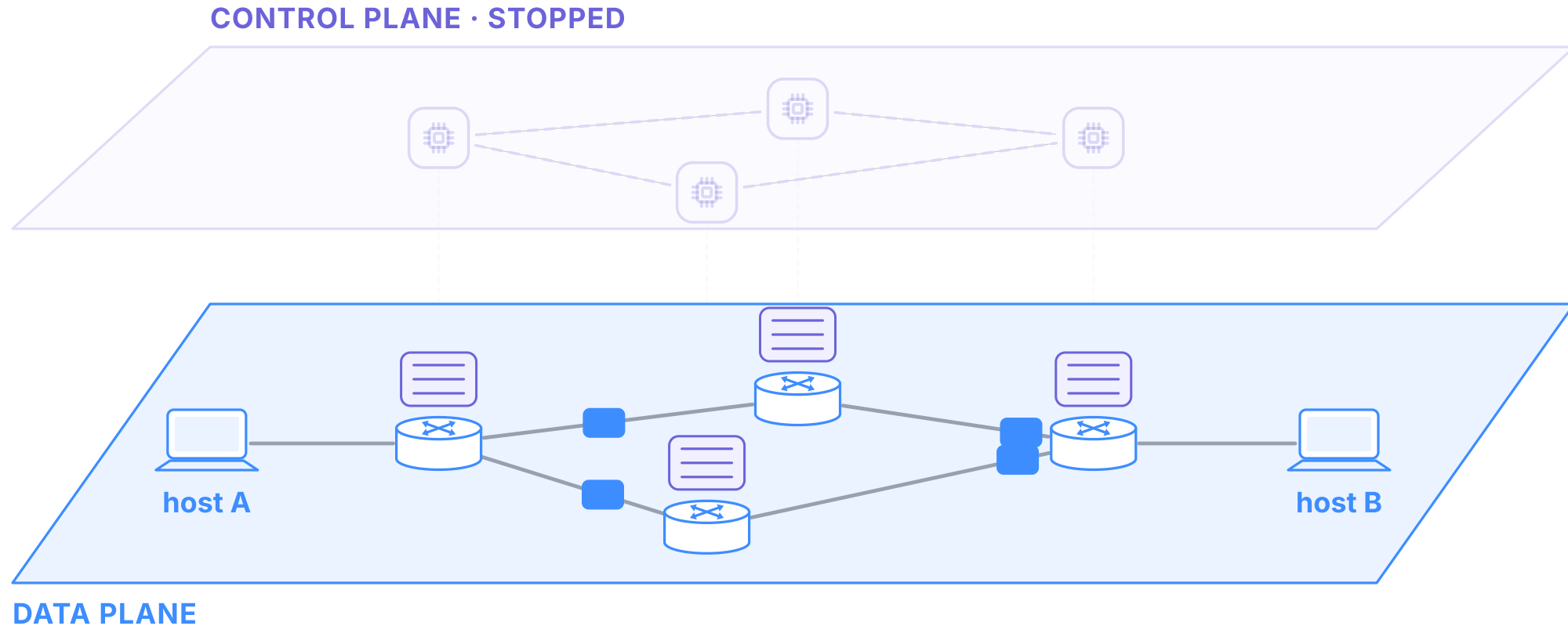
Data plane: forwarding, fast and simple. Control plane: routing, the management.

1 The layer

2 Two jobs

3 Two planes

What if the control plane stops?



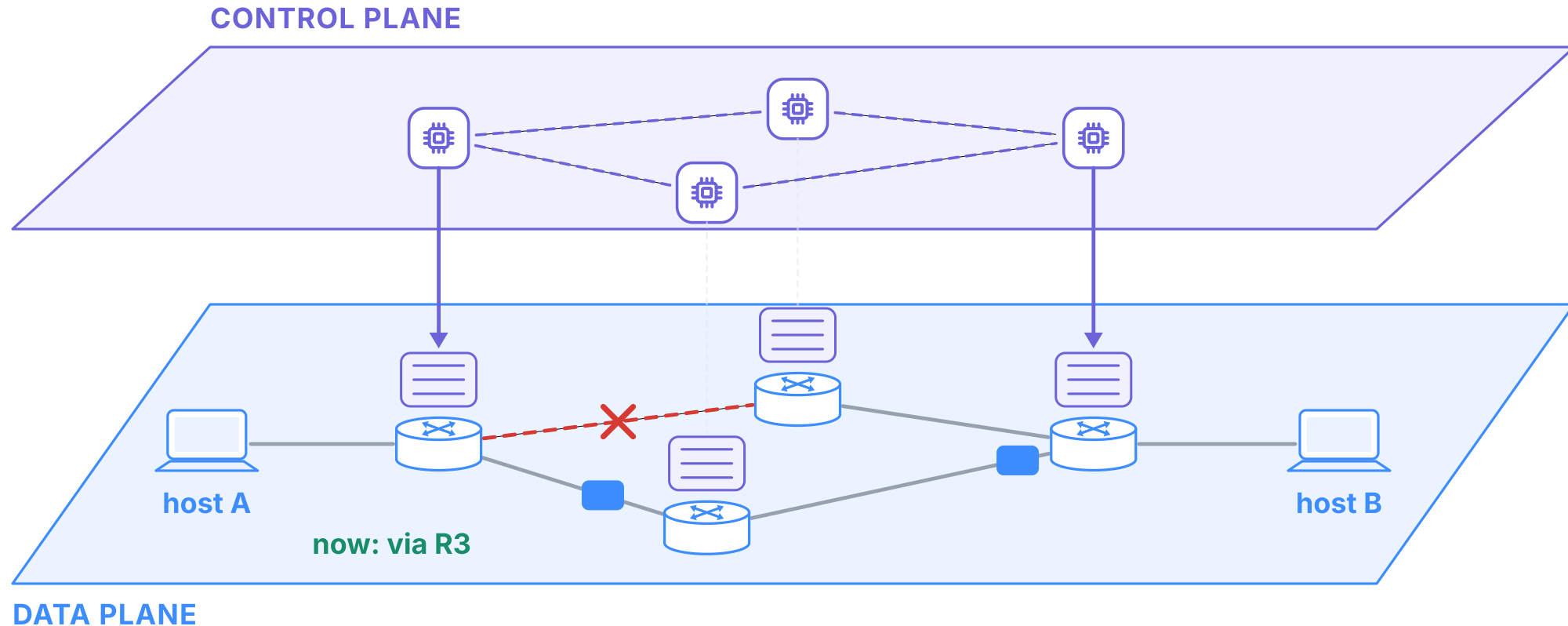
Packets keep flowing on the tables they have. Control is not in the packet's path.

1 The layer

2 Two jobs

3 Two planes

What if a link fails?



Forwarding follows the table, even into a dead link, until routing rewrites it.

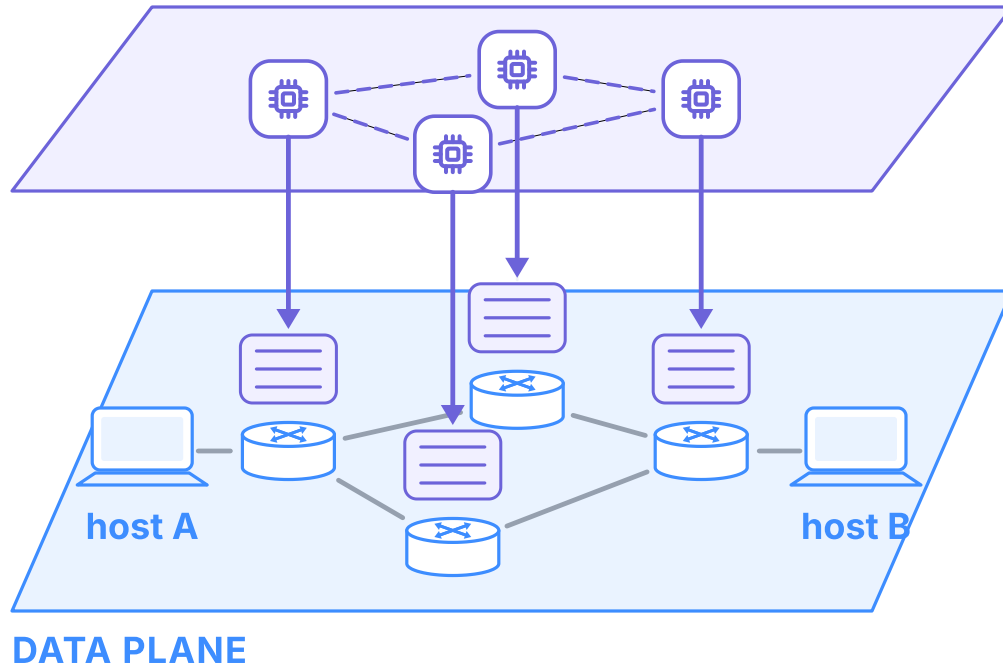
1 The layer

2 Two jobs

3 Two planes

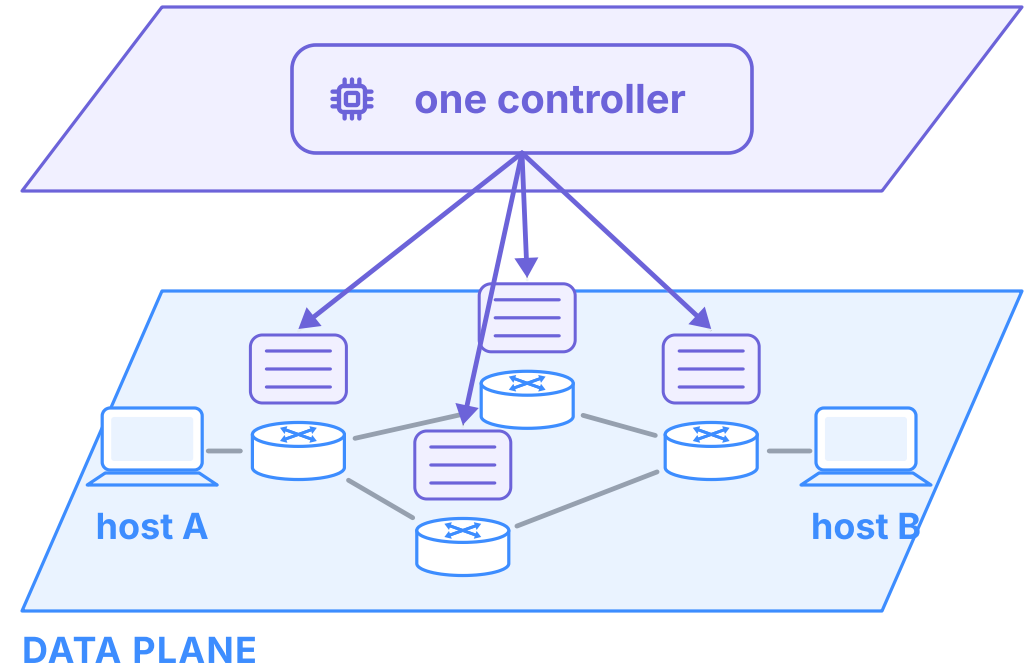
Where does the control plane run? Two answers

A ROUTING PROCESS IN EVERY ROUTER



the Internet today: the next chapter

ONE CONTROLLER FOR ALL OF THEM



software-defined networking (SDN)

Google's data-centre backbone has run this way since 2013

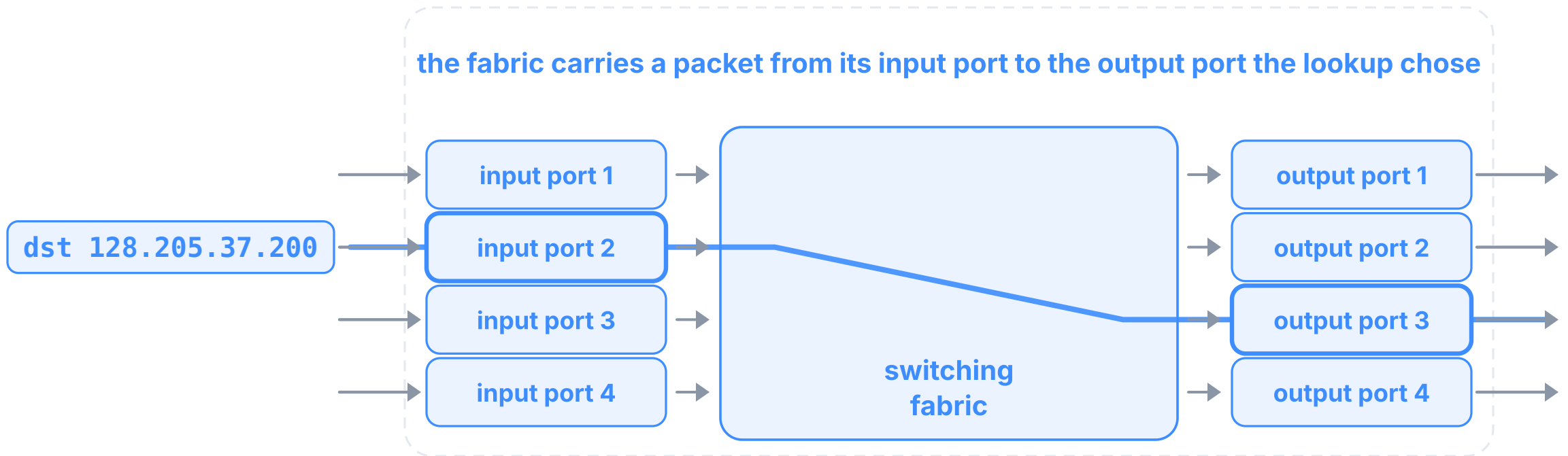
Either way the data plane is identical. Only who computes the table changes.

1 The layer

2 Two jobs

3 Two planes

Inside the box: ports, and a fabric between them



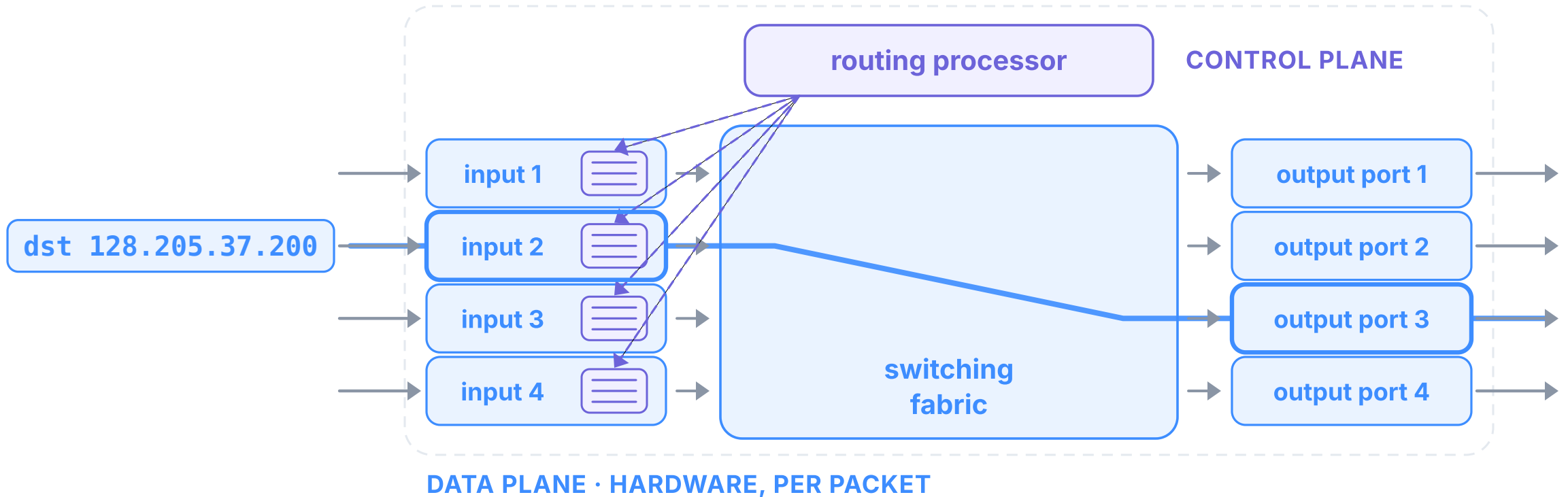
Four links, four ports. A packet comes in on one port and the fabric moves it to another.

1 The layer

2 Two jobs

3 Two planes

The two planes inside one router



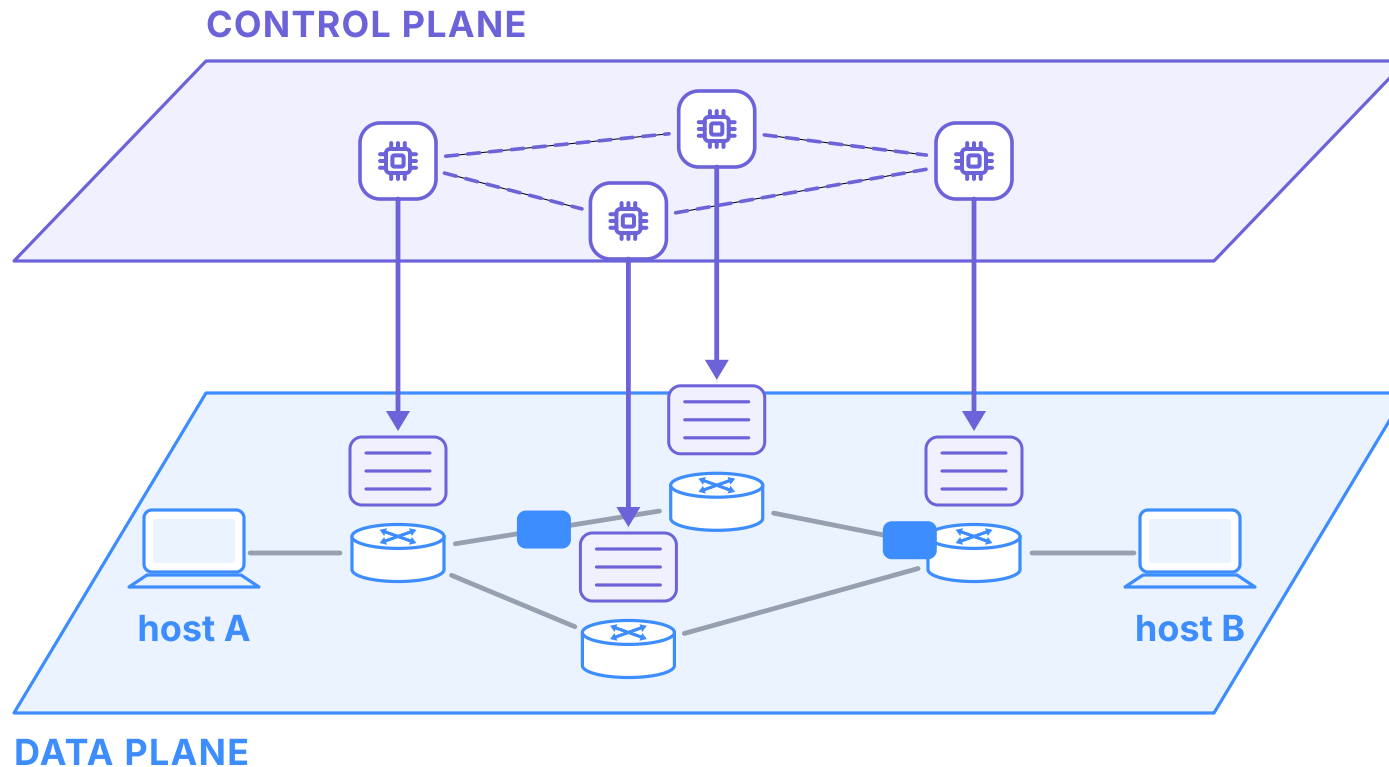
Software writes the table once per change; hardware reads it once per packet.

1 The layer

2 Two jobs

3 Two planes

Which lecture opens which part



CHAPTER 5 · CONTROL PLANE

Link state · Distance vector

two ways to compute a table

OSPF

routing inside one network

BGP

routing between networks

THE REST OF CHAPTER 4 · DATA PLANE

IP

the header and the address

Forwarding

the lookup in the table

Router

the hardware a packet crosses

The rest of Chapter 4 is the data plane. Chapter 5 is the control plane.

1 The layer

2 Two jobs

3 Two planes

Three things to carry out of this lecture

01



The layer

host to host, on every machine
one shared protocol: IP
which promises only best effort

02



Two jobs

forwarding reads the table,
once per packet;
routing fills it, once per change

03



Two planes

the data plane reads,
the control plane writes;
the table is the only interface

Next: Forwarding. How does a table of a few rows answer for four billion addresses?