University of Plymouth, Centre for Information Security and Network Research
Wednesday 2nd April 2008

A Closer Look at OSPF and EIGRP

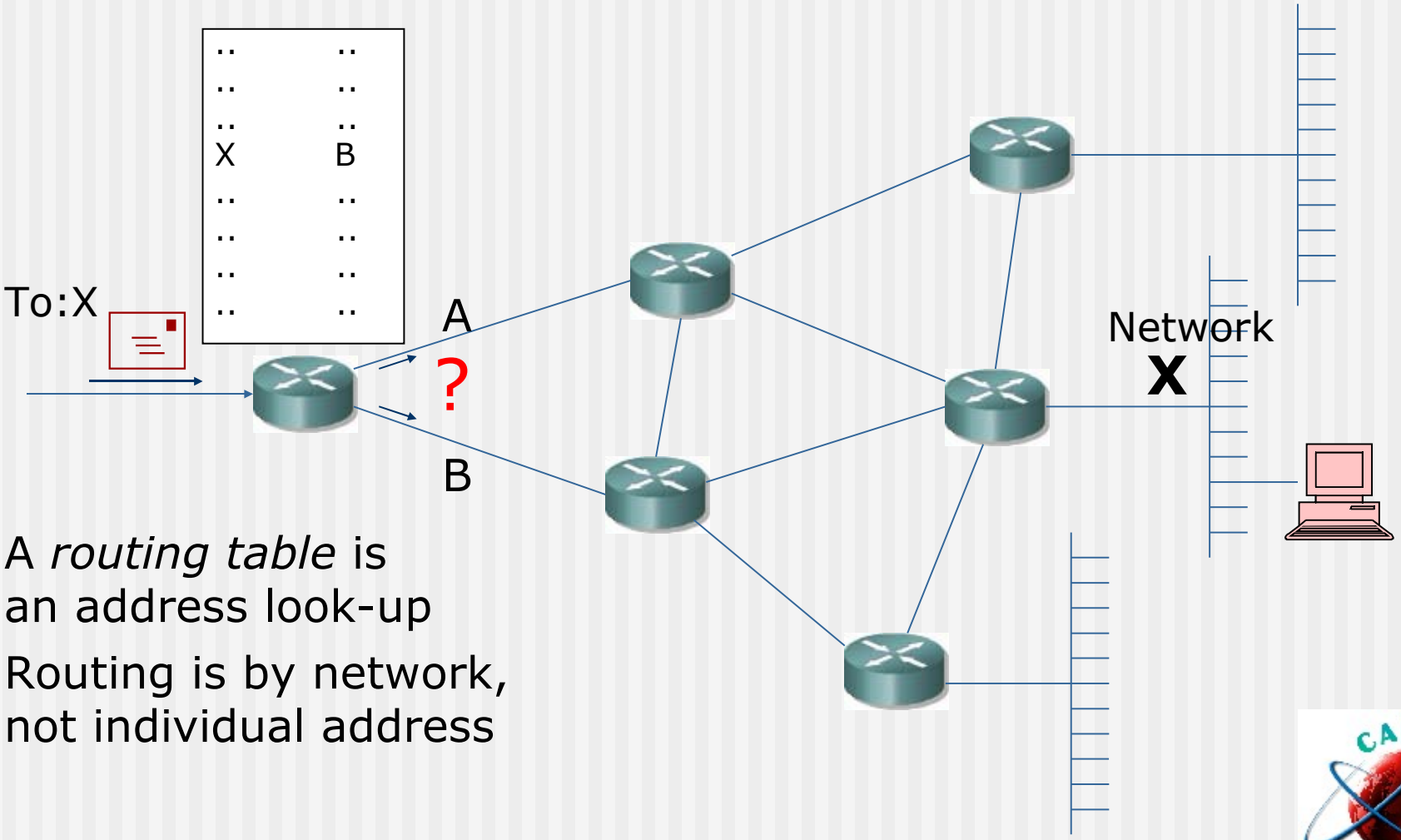
Vic Grout

Centre for Applied Internet Research (**CAIR**)
University of Wales
NEWI Plas Coch Campus, Mold Road
Wrexham, LL11 2AW, UK

v.grout@newi.ac.uk

www.cair-uk.org

Routing Tables



A *routing table* is an address look-up

Routing is by network, not individual address

Definitions

- A *routed* protocol, such as IP, allows packets to find their way to their destination in a packet-switched system
- They do this by looking up the address of each packet in each router's *routing table*, which tells the router where to send it next
  - the *next hop*
- **But where do these *routing tables* come from?**
- Routing tables can be tens of thousands of lines long. Some routes might be configured by hand (static routes) but most won't!

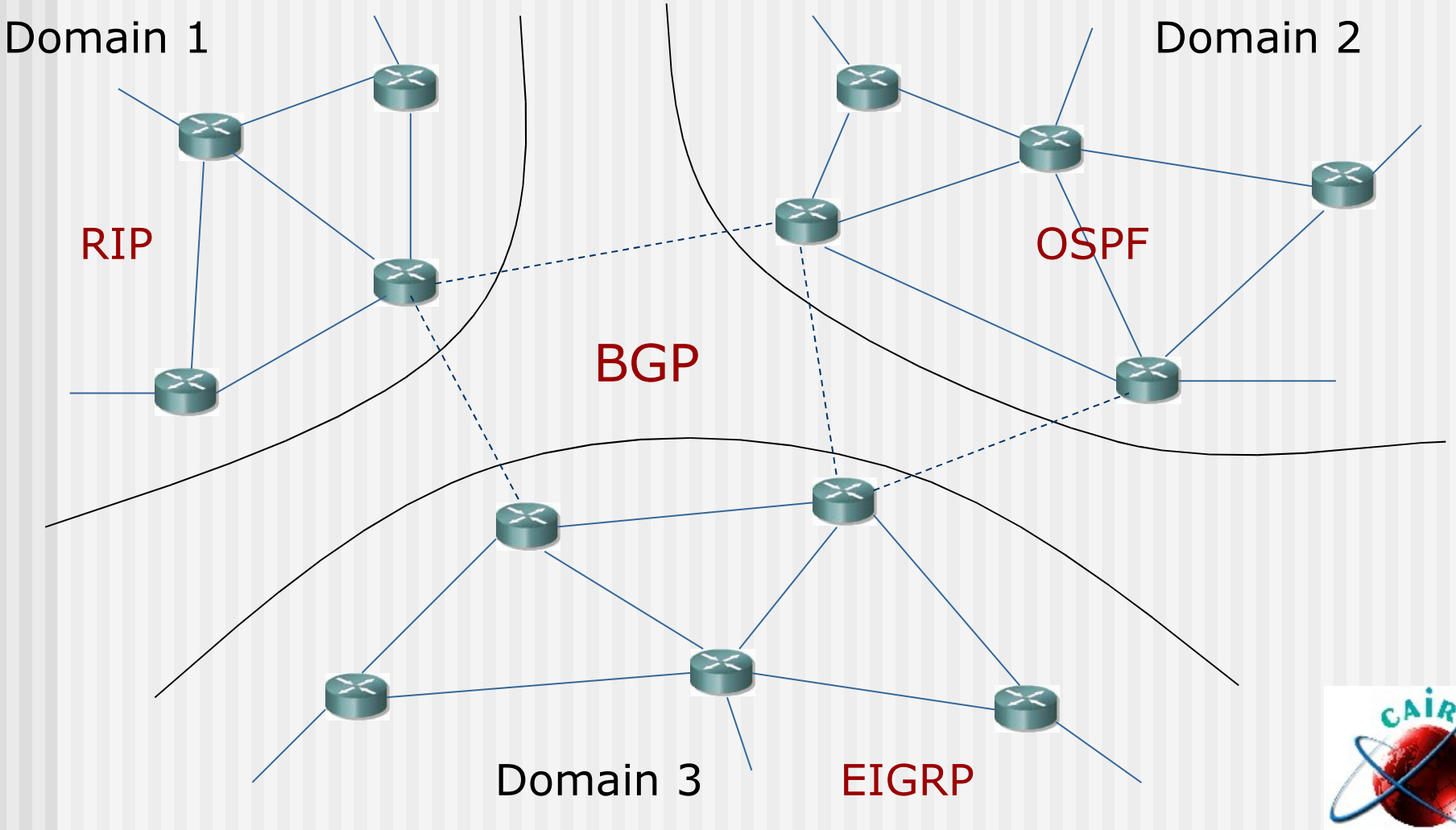
Definitions

- A *routing* protocol, such as RIP, OSPF, IS-IS, EIGRP, BGP, etc. allows routers to share what they know about the network topology …
- … so that each builds up their own *routing table*
- Topologies change (links fail, etc.) from time to time so this form of routing is *dynamic*
- **The relationship is simple: *routed* protocols need *routing* protocols to work**
- There are many different routing protocols and a number of different types

Types of Routing

- *Interior* Routing (IRPs)
  - within a *domain*
    - routers under a common administration
    - that trust each other
  - eg, RIP, OSPF, IS-IS, EIGRP
  - algorithmic
- *Exterior* Routing (ERPs)
  - between domains
    - a form of protection
  - eg, BGP
  - procedural/rule-based

Interior and Exterior Routing



Exterior Routing

- Almost entirely BGP (*Border Gateway Protocol*)
- A ‘protection mechanism’
- Configured as a set of precedent rules
  - what traffic to accept
  - what routes to advertise
- Nothing particularly clever in the protocol …
- No real route calculations …
- In fact, not really a protocol
- So we’re not interested in these!

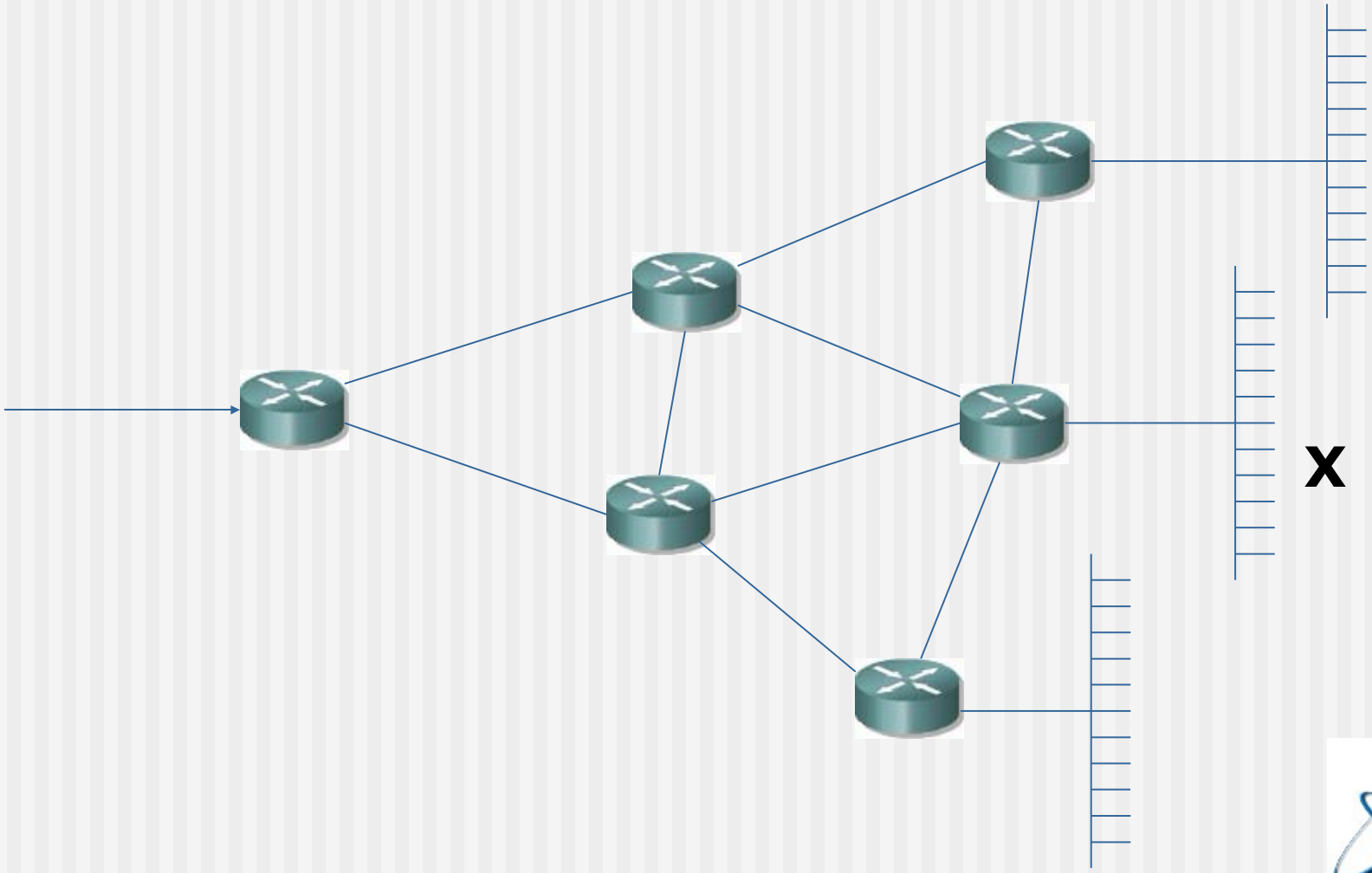
Interior Routing

- A ‘cooperation mechanism’
- Once set up, the protocol takes care of
  - routing information exchange
  - route calculation
    - Sometimes separately (eg OSPF)
    - sometimes not (eg RIP)
- Three basic types
  - *Distance Vector*
    - eg, RIP
  - *Link-State*
    - eg, OSPF
  - *Hybrid* (?)
    - eg, EIGRP

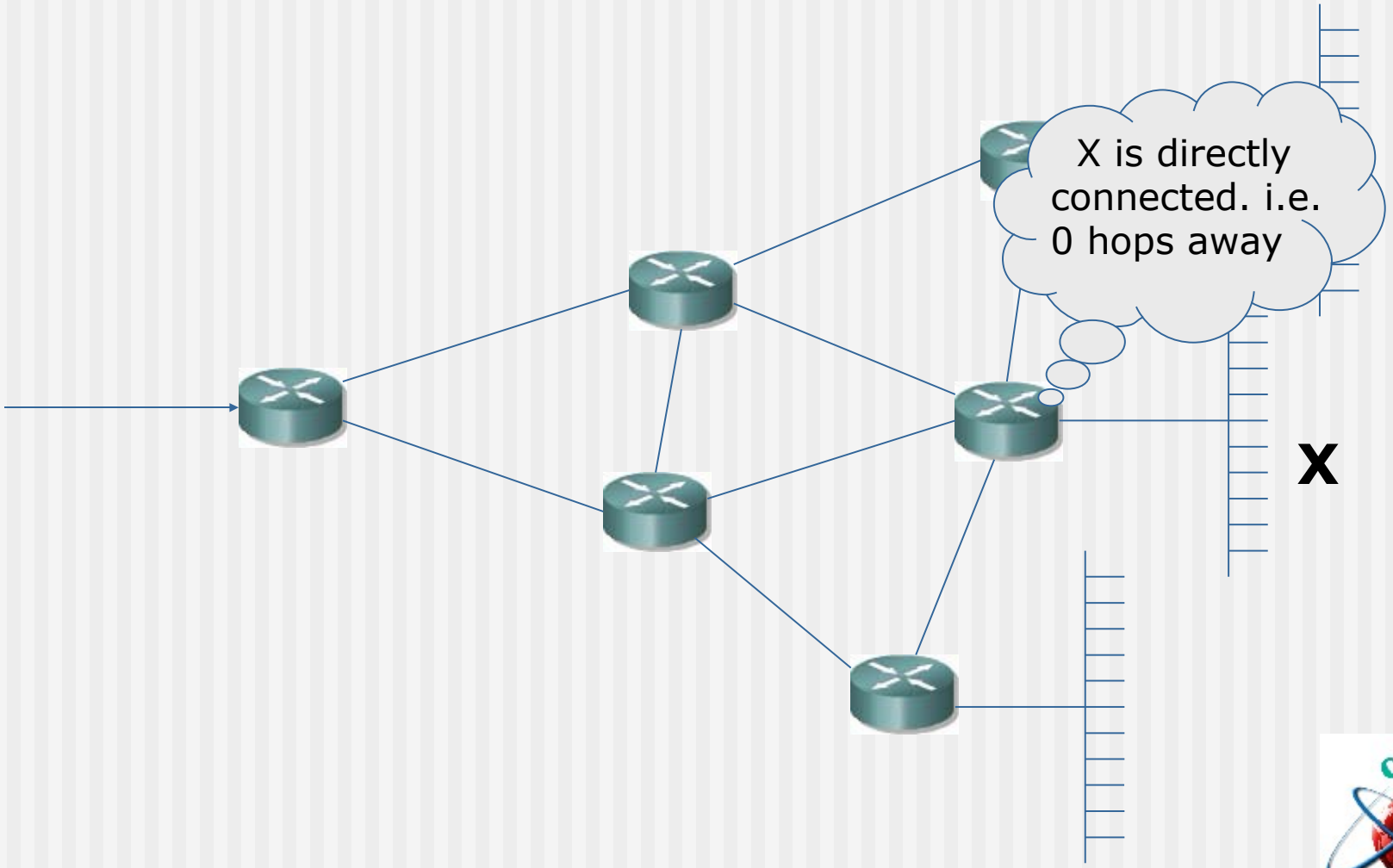
Distance Vector Protocols

- eg, RIP (*Routing Information Protocol*)
- Routers running a DV protocol …
  - measure distance (doesn't have to be hops)
  - exchange routes (doesn't have to be periodic)
  - and so, know …
    - how far to a remote network (distance)
    - which direction/interface to take to get to that network (vector)
- DV protocols are …
  - inefficient (?)
  - distinctly sub-optimal (?)
  - slow to *converge* across the domain (?)

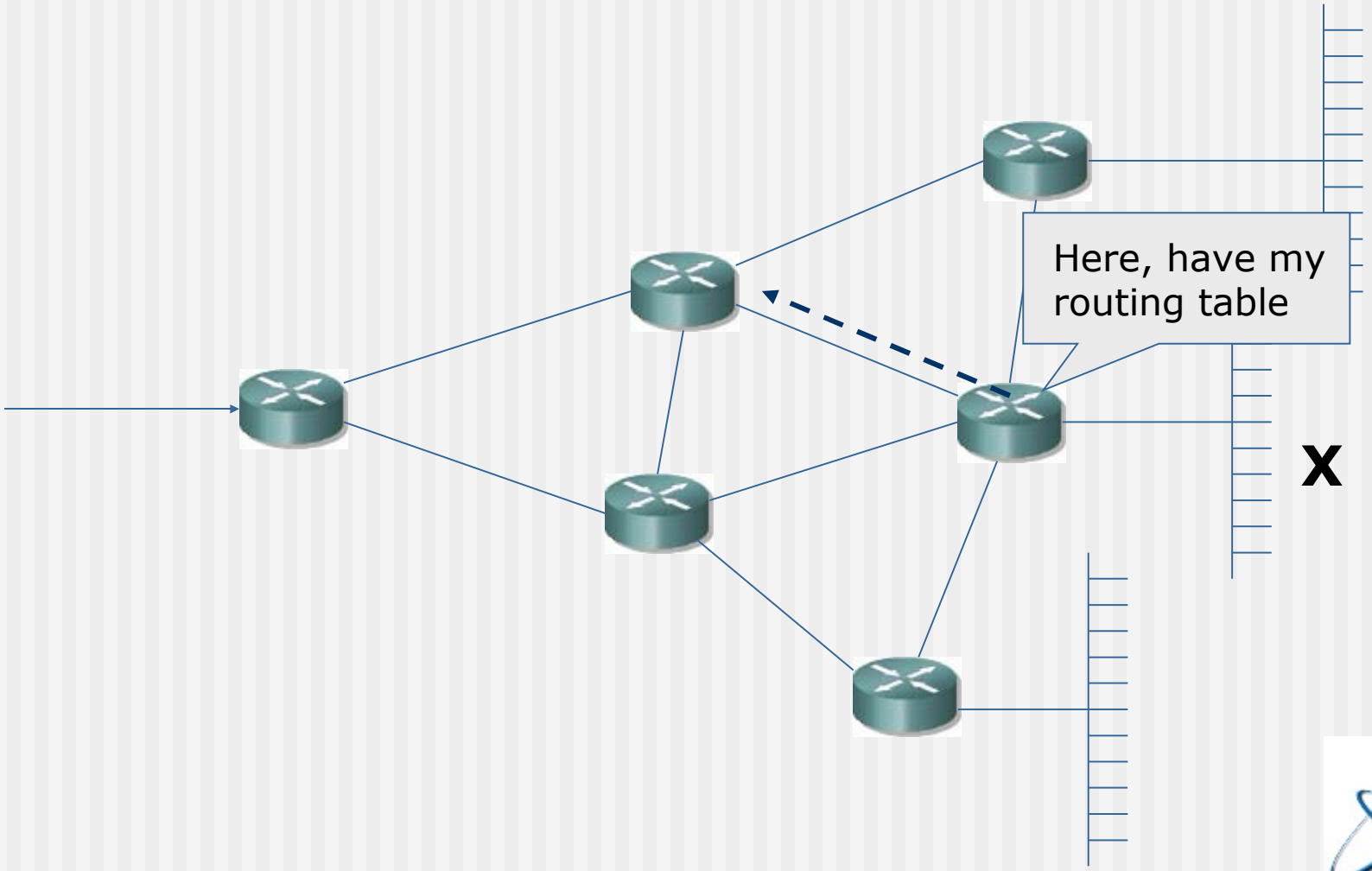
Distance Vector Routing



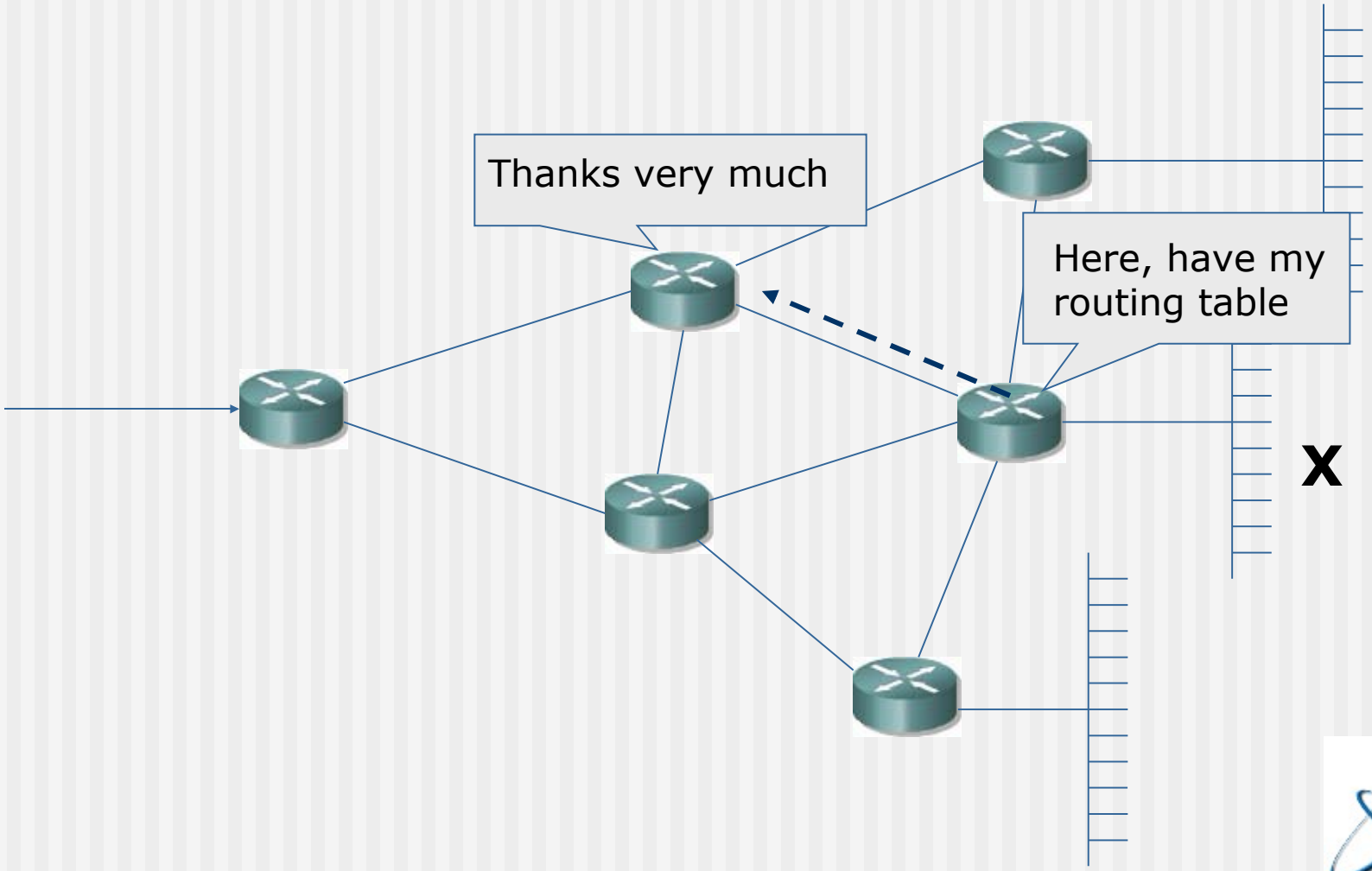
Distance Vector Routing



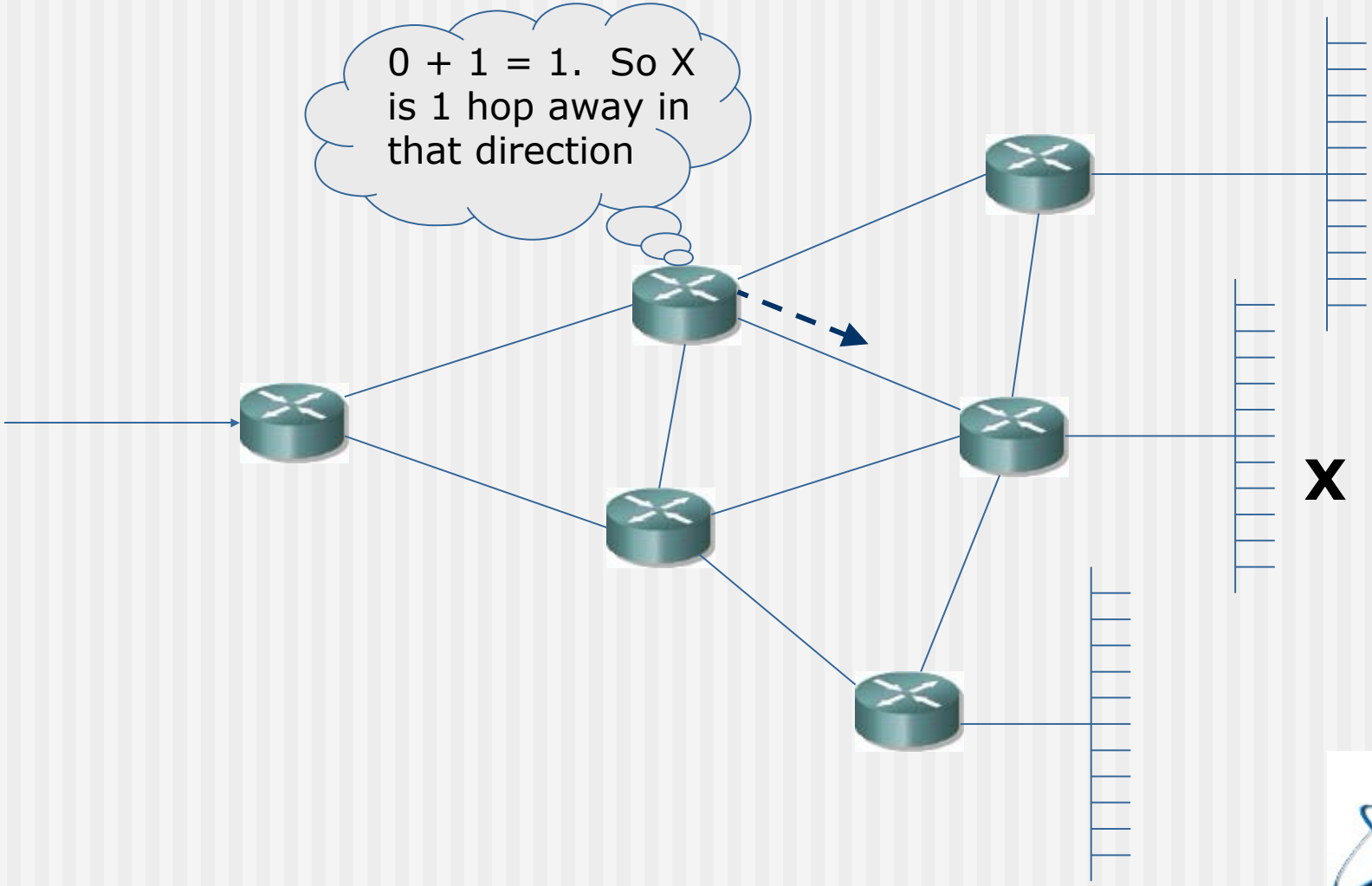
Distance Vector Routing



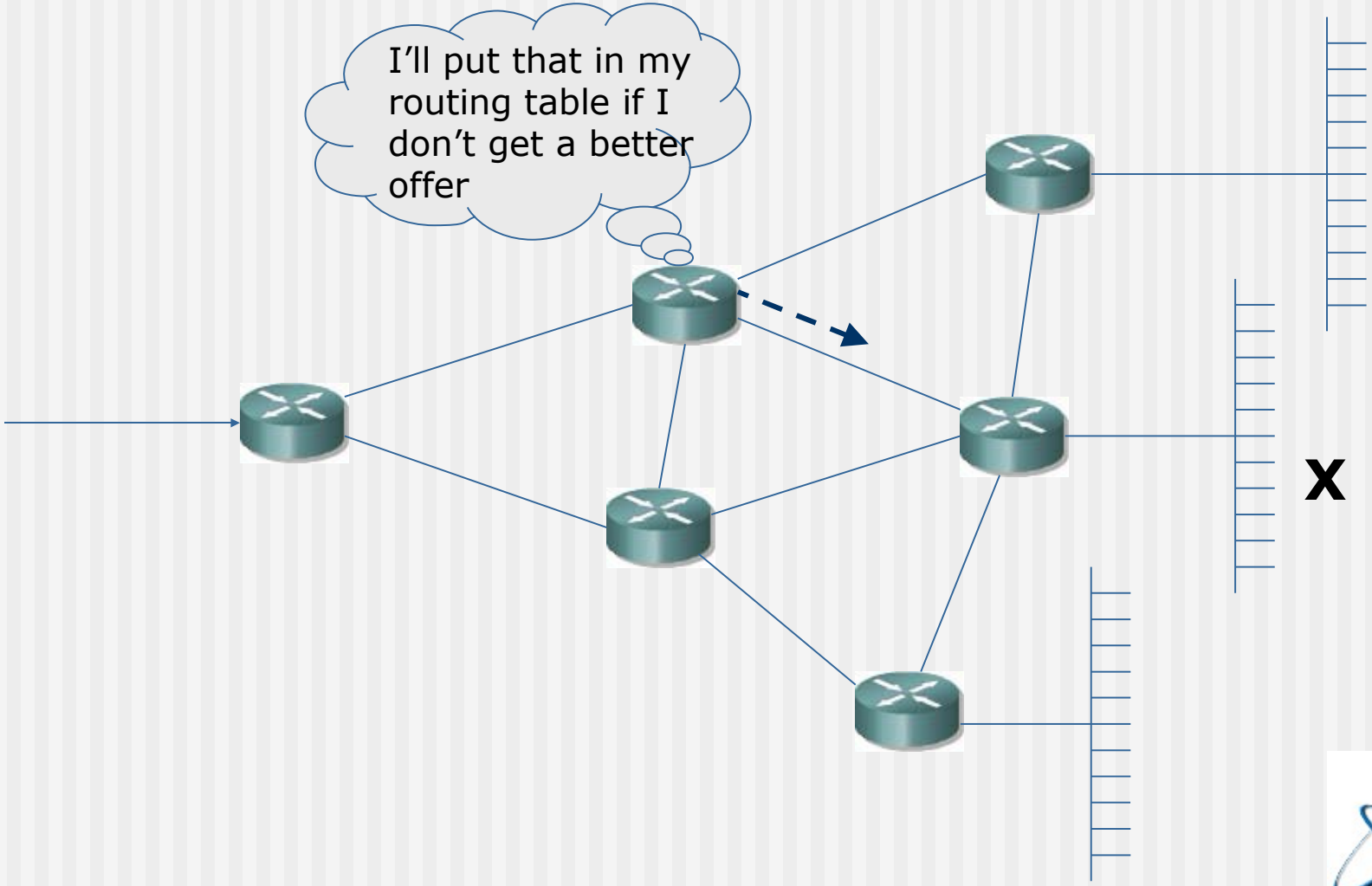
Distance Vector Routing



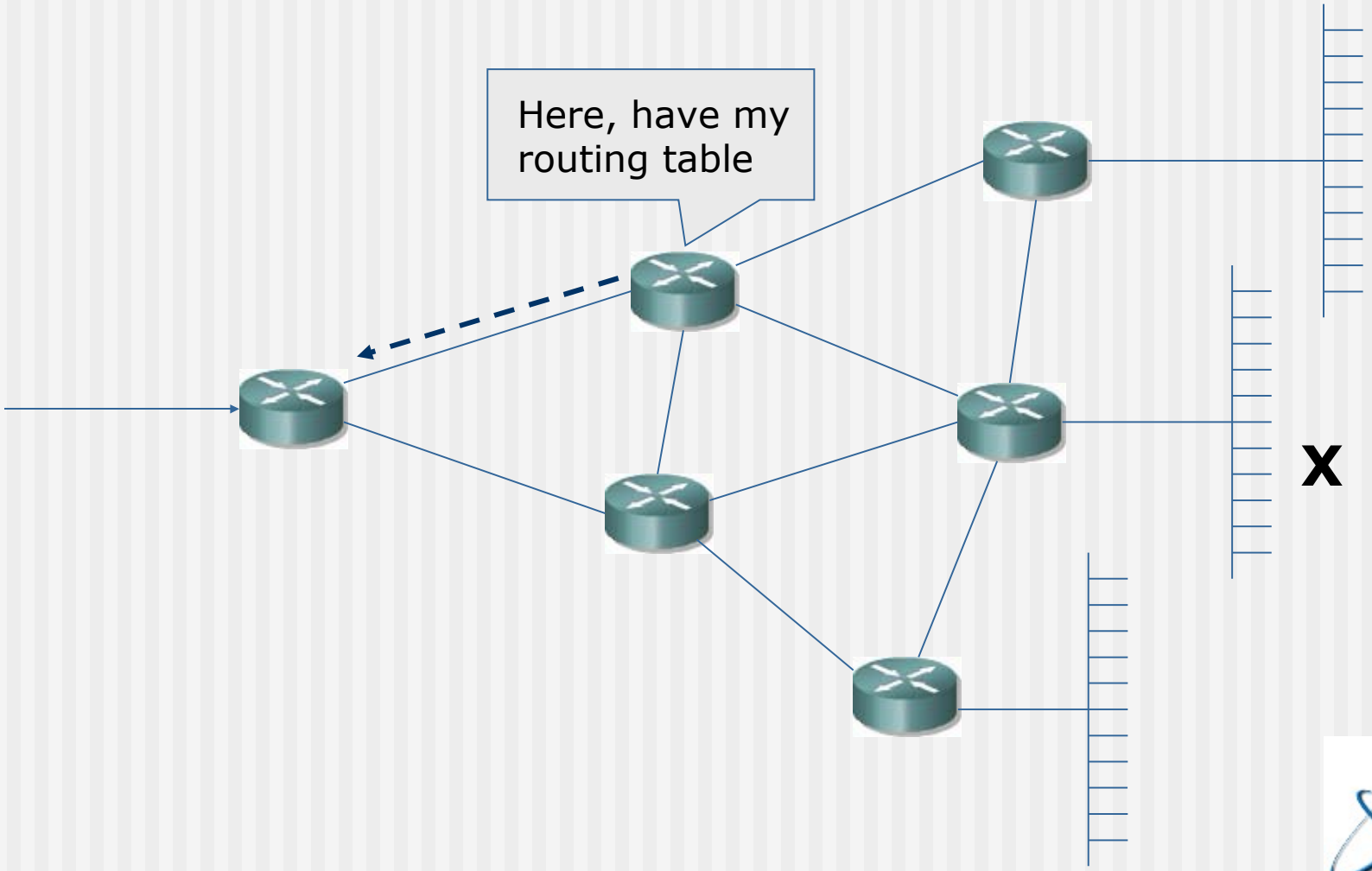
Distance Vector Routing



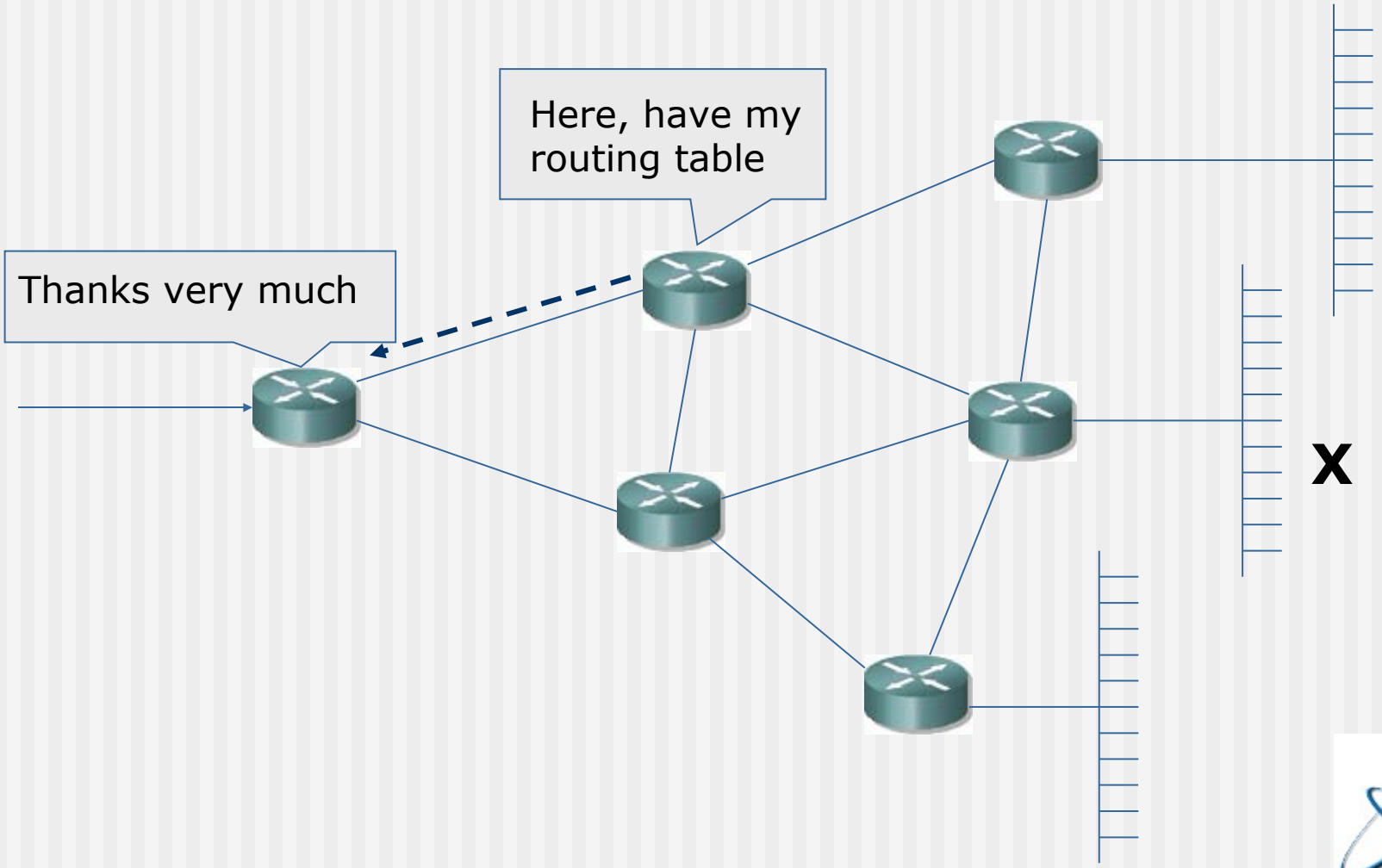
Distance Vector Routing



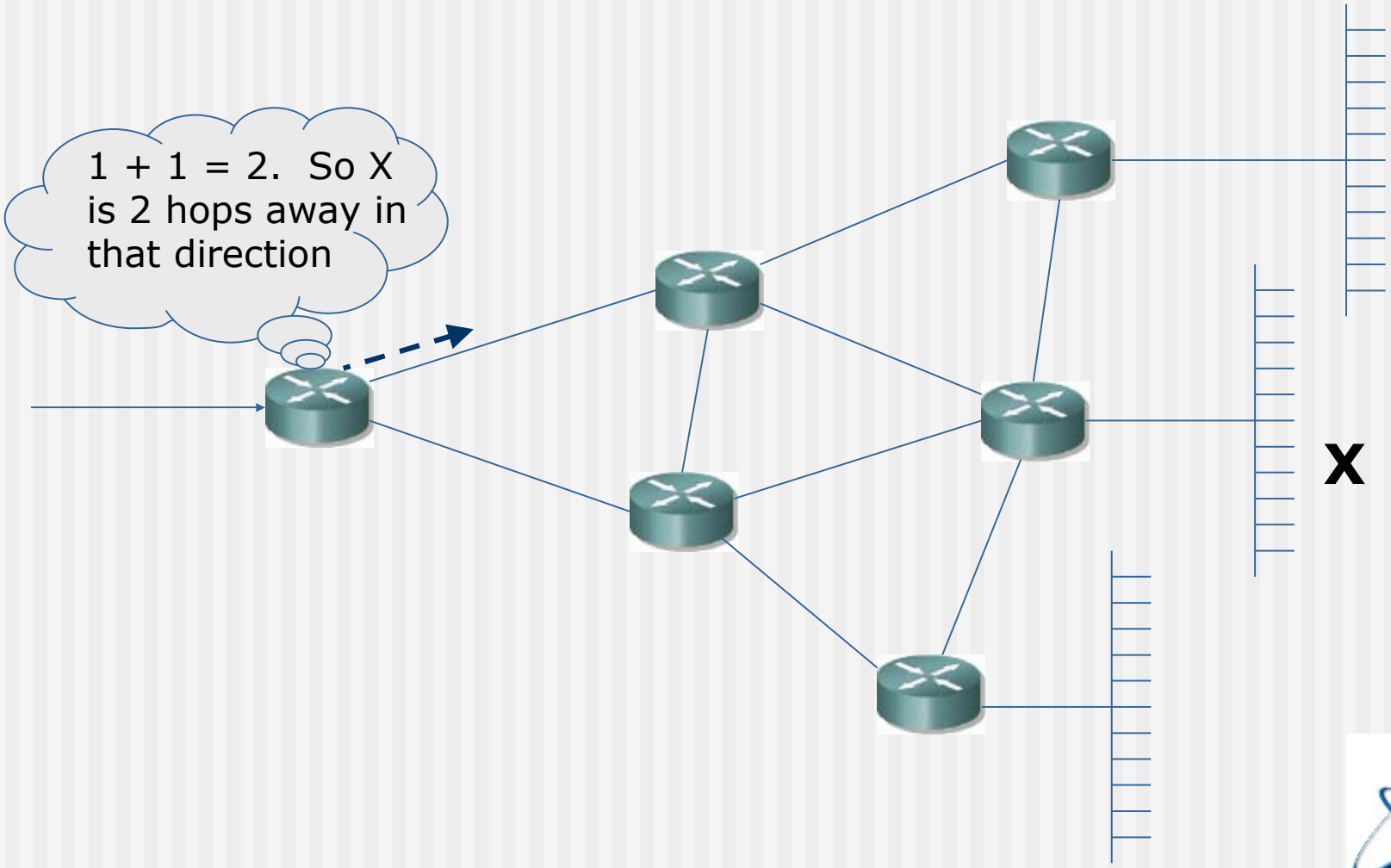
Distance Vector Routing



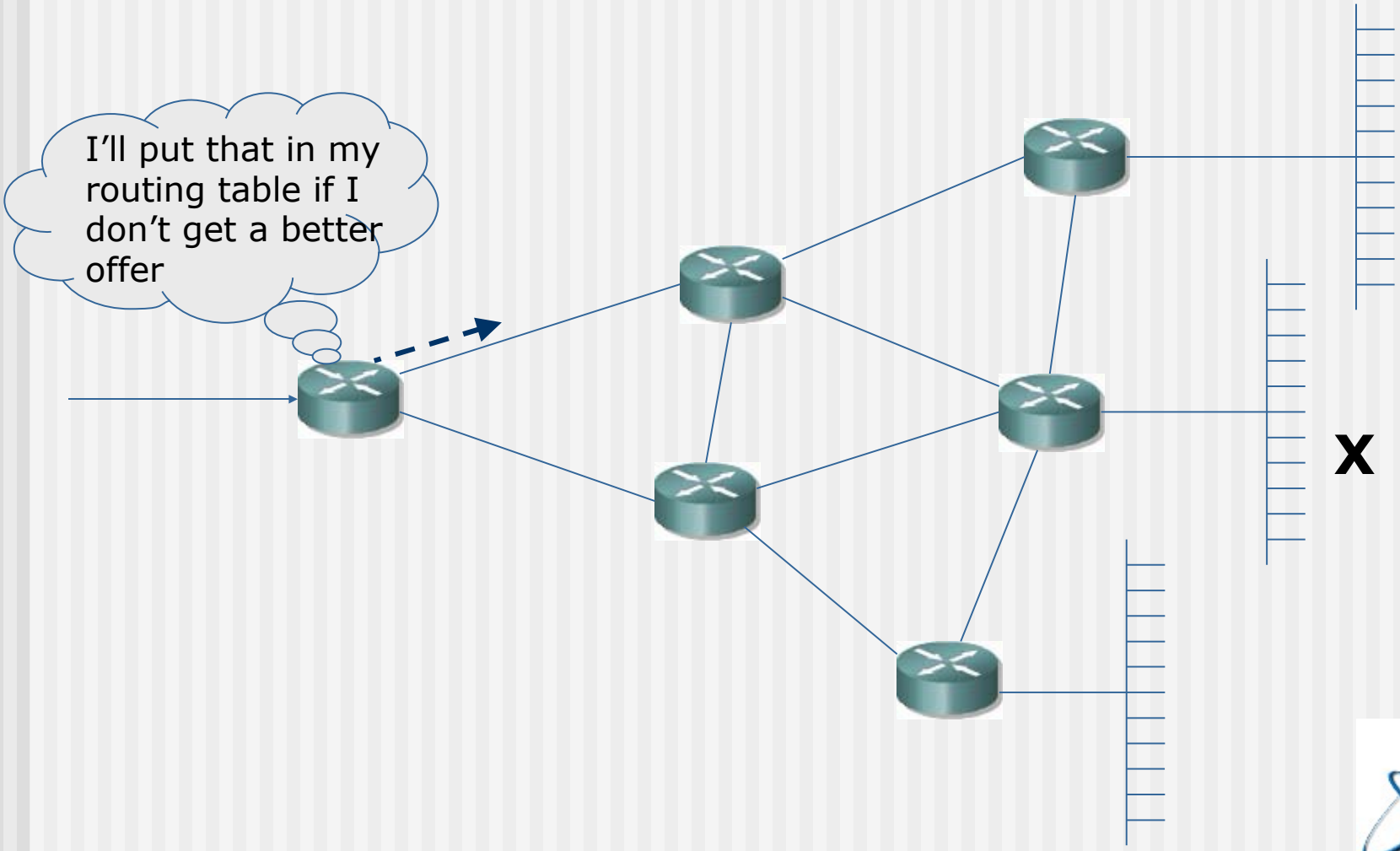
Distance Vector Routing



Distance Vector Routing



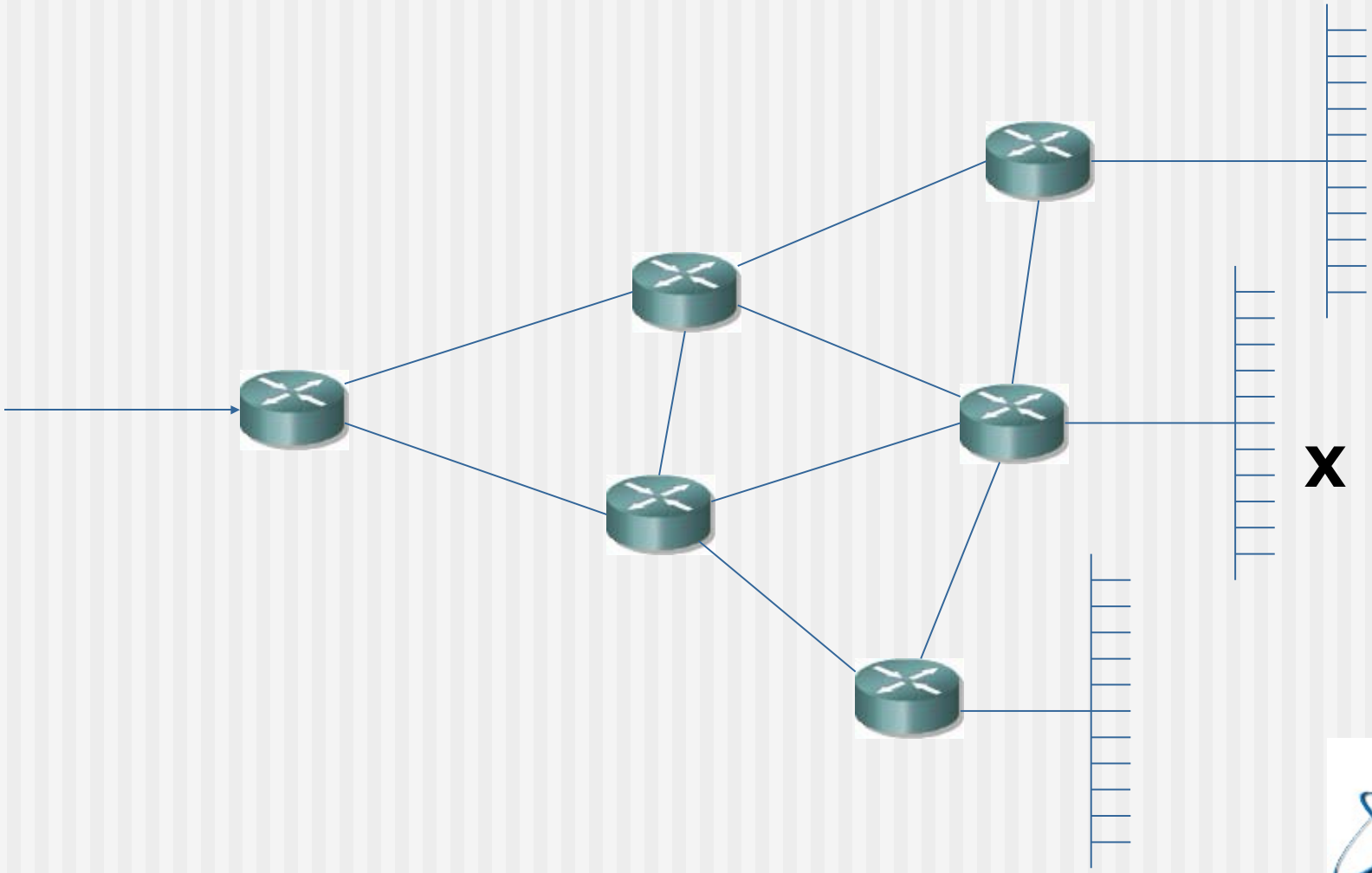
Distance Vector Routing



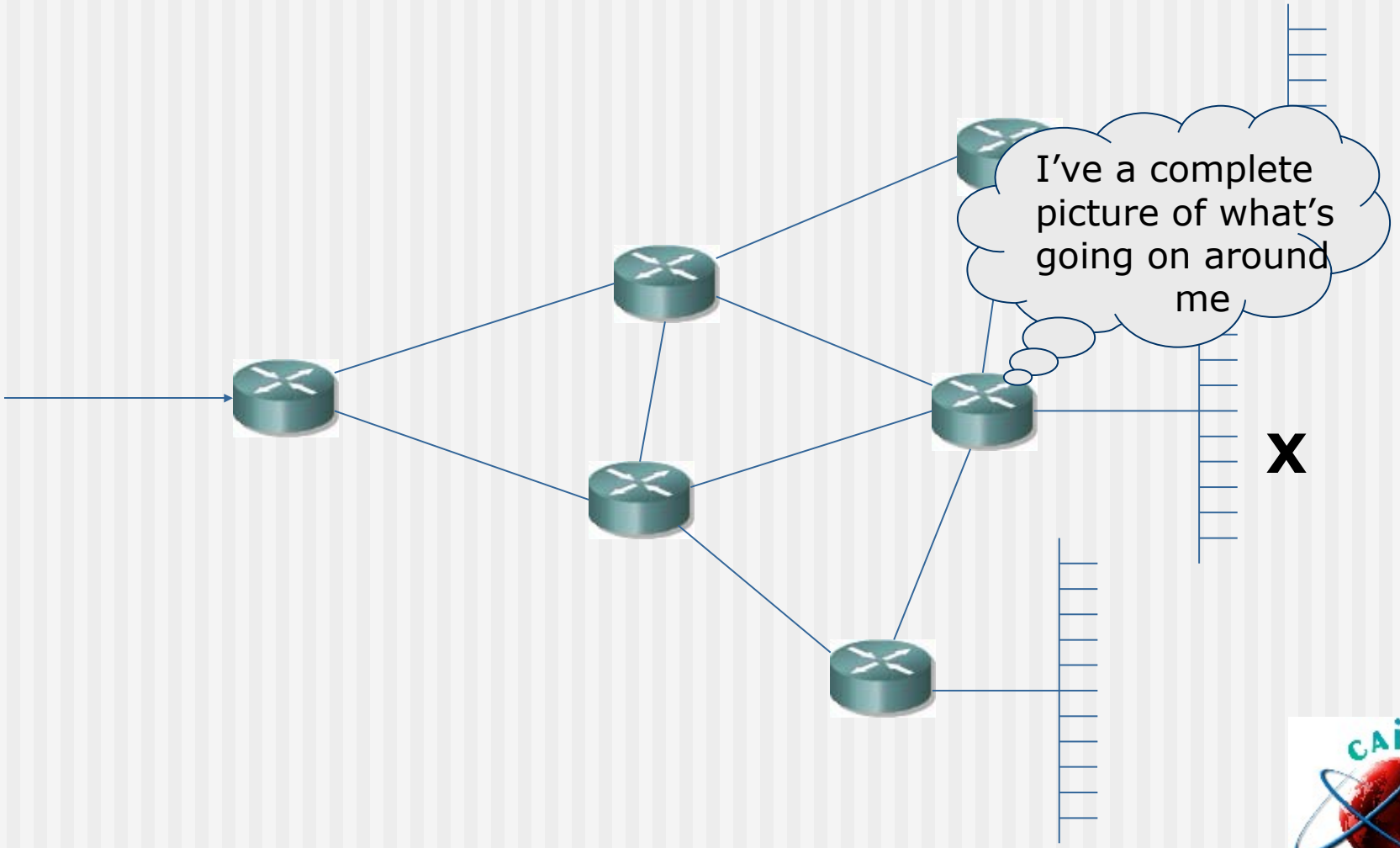
Link-State Protocols

- eg, OSPF (*Open Shortest Path First*)
- Routers running an LS protocol …
  - measure the *cost* of each link
    - eg, $C = 10^8 / Bandwidth$
  - exchange *routing updates* as when needed (probably)
  - and so, can …
    - calculate cost of each path to a remote network
    - run a shortest path algorithm (eg, Dijkstra) to find the 'best' route to put in their routing table
- LS protocols are …
  - more efficient (?)
  - closer to optimal (?)
  - faster to converge across the domain (?)

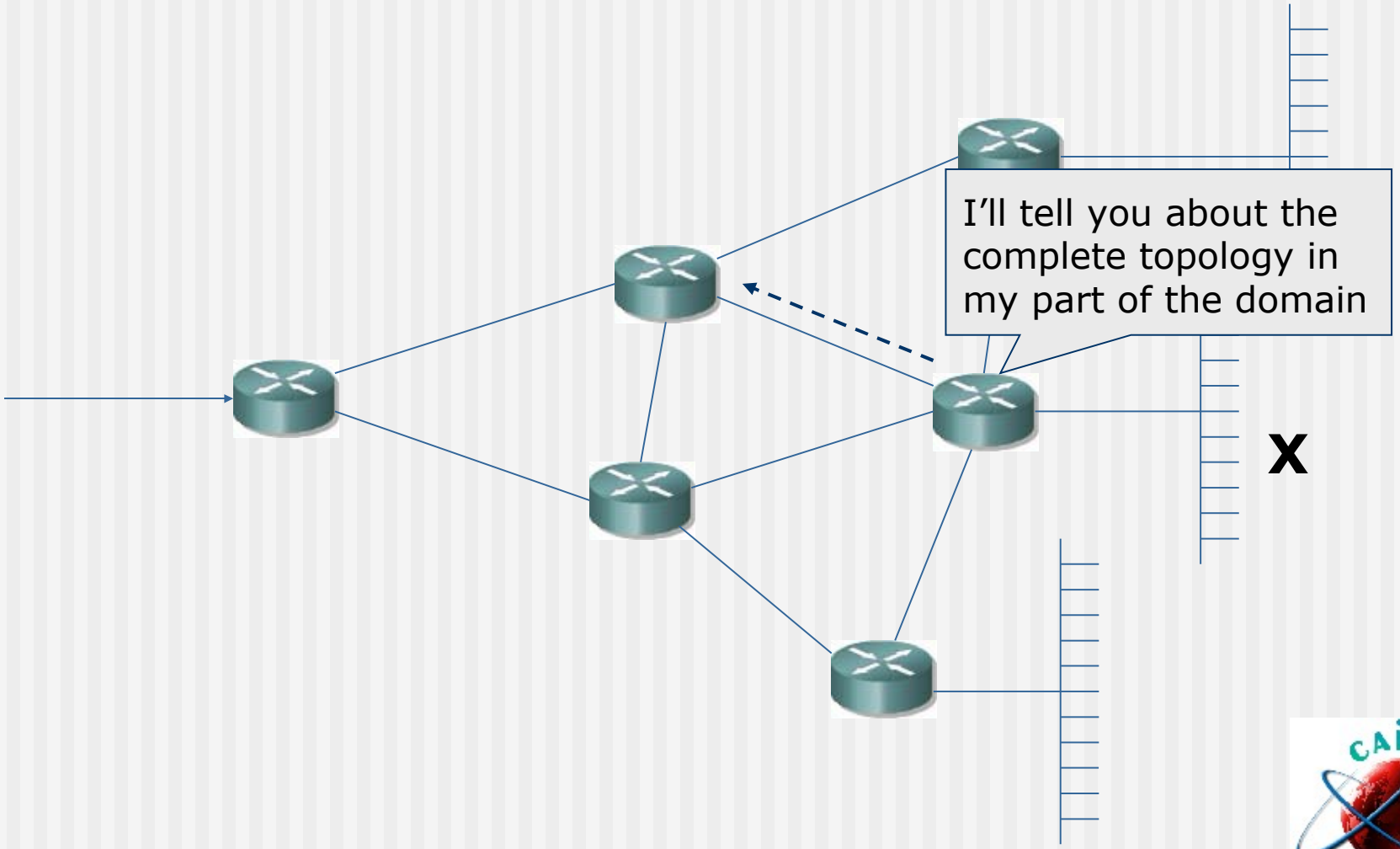
Link-State Routing



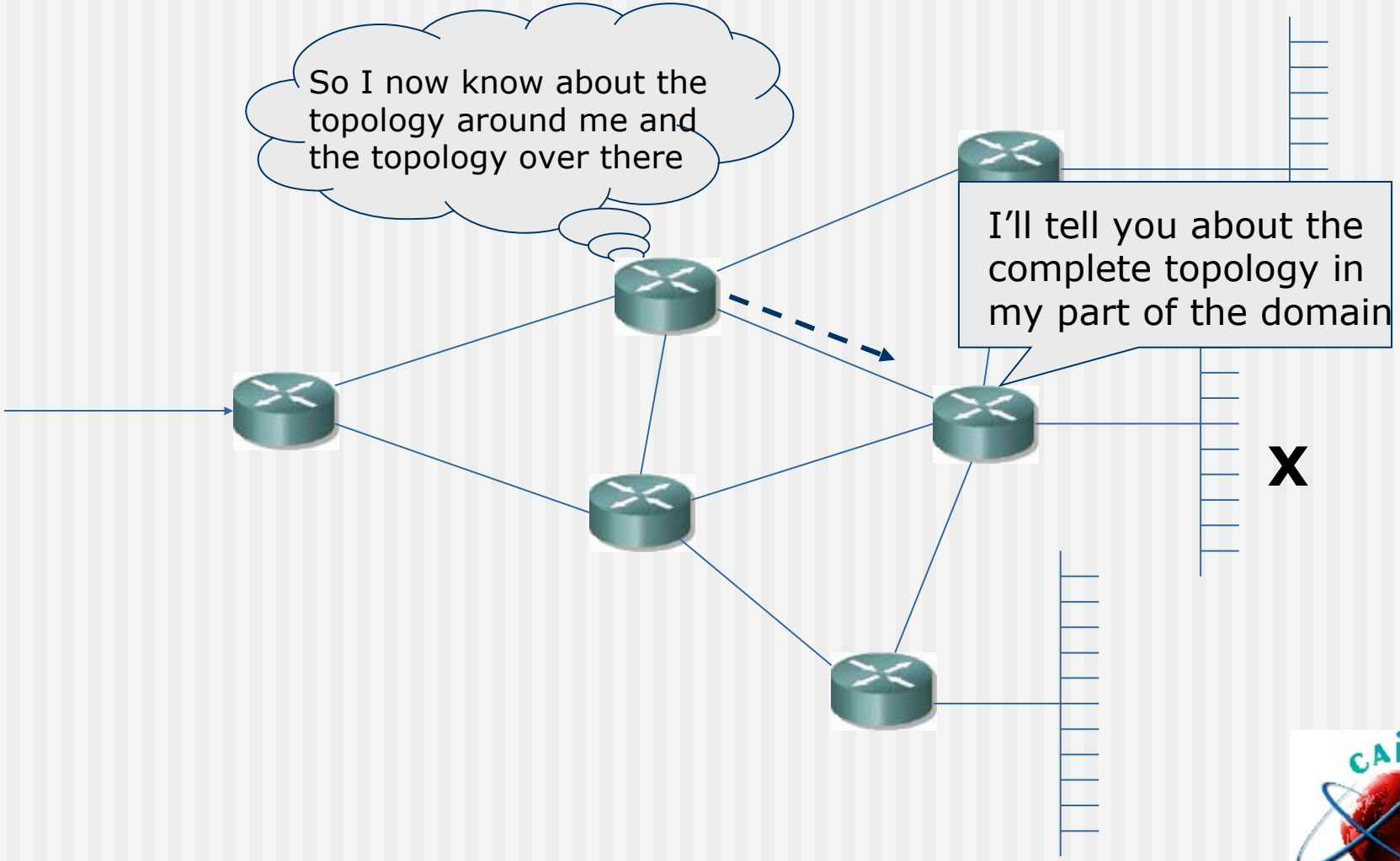
Link-State Routing



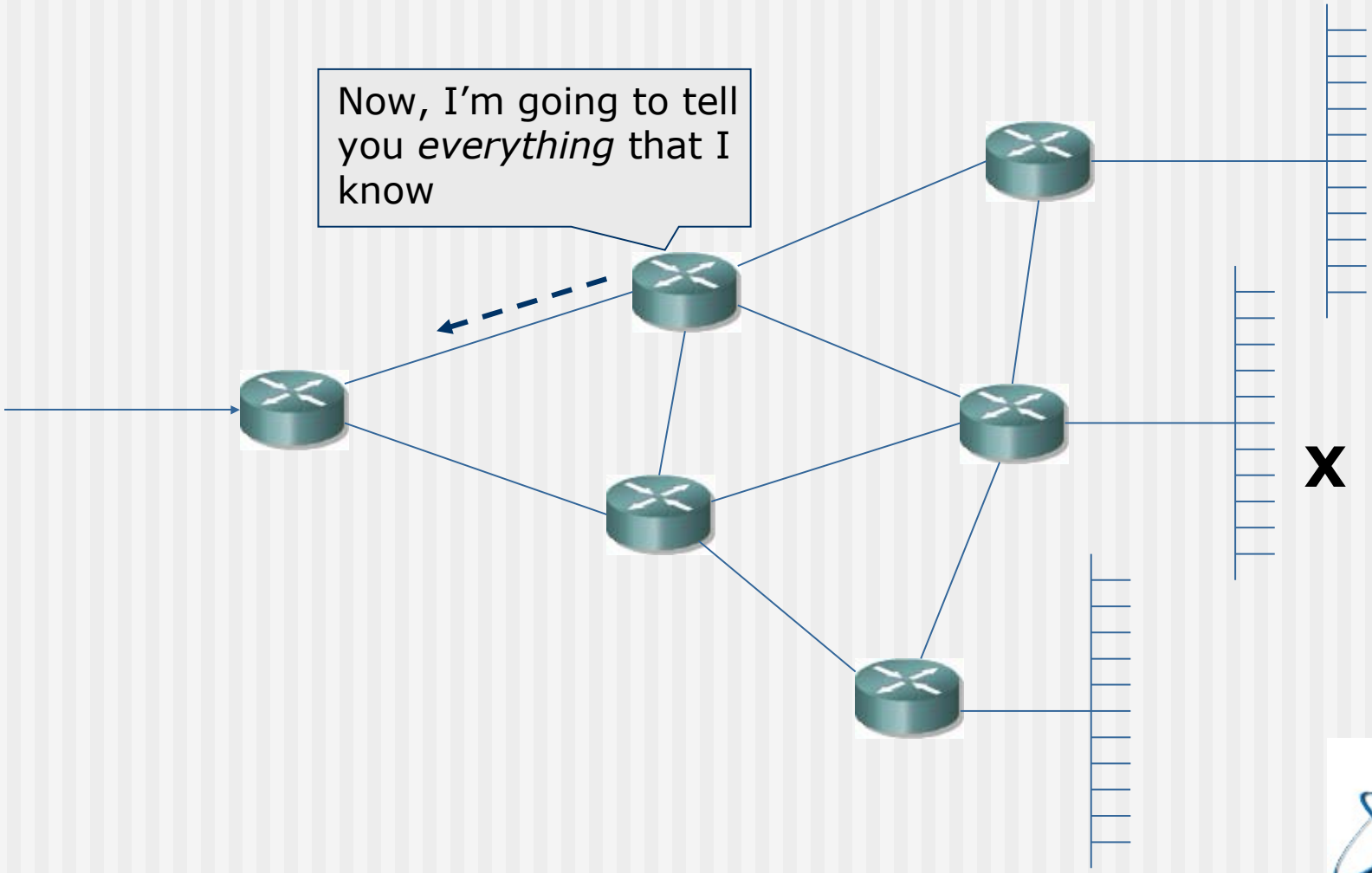
Link-State Routing



Link-State Routing

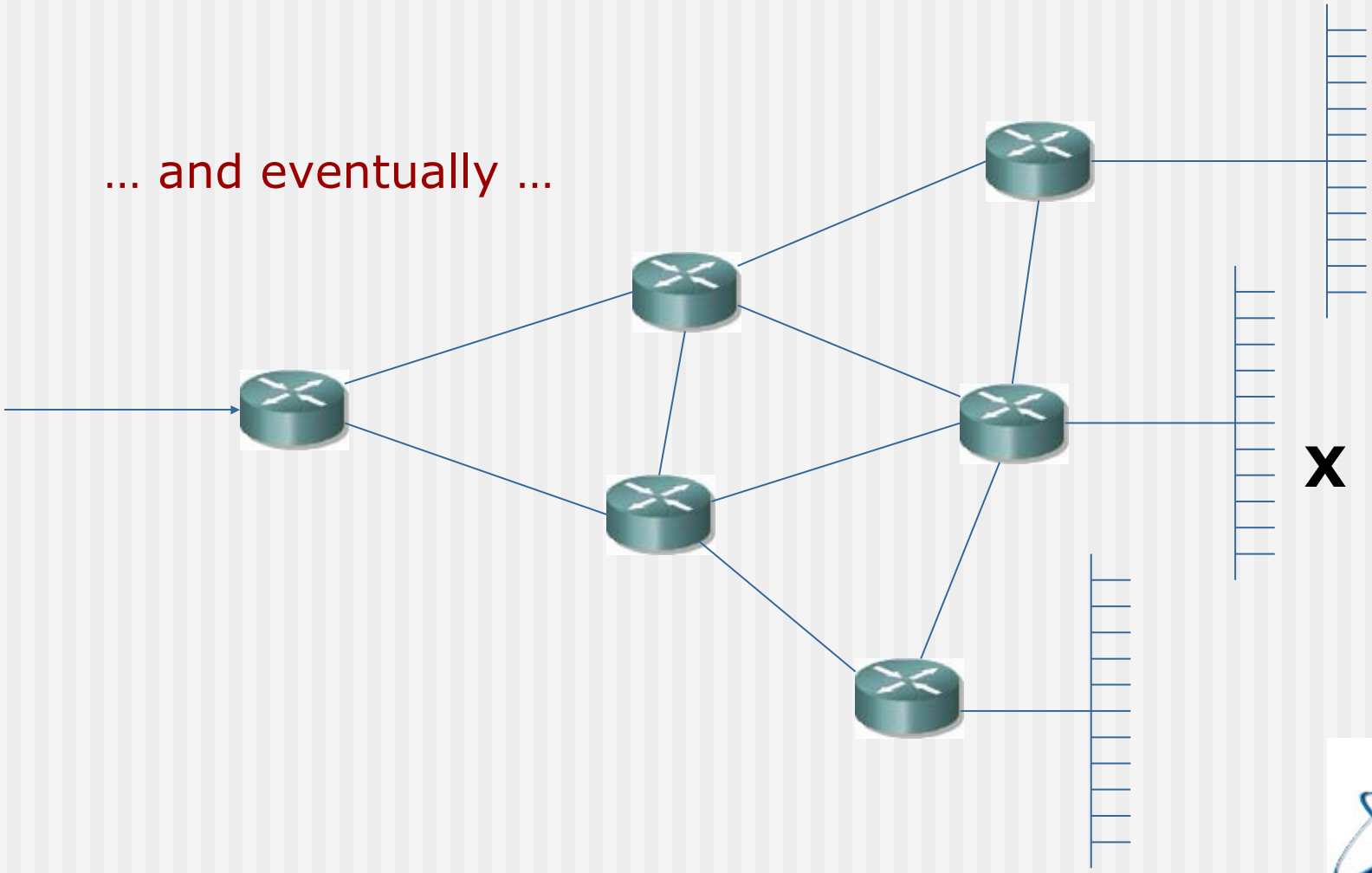


Link-State Routing

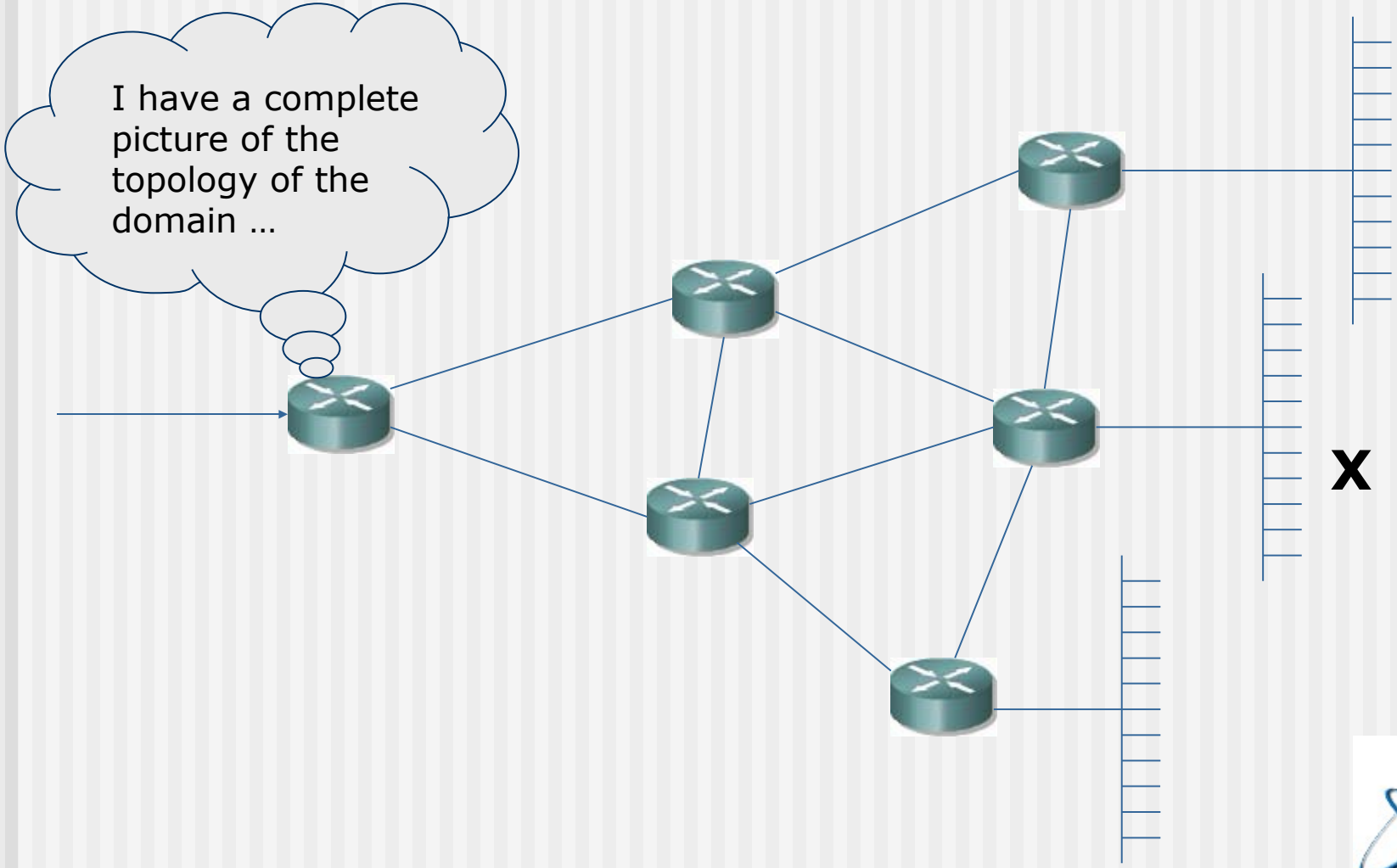


Link-State Routing

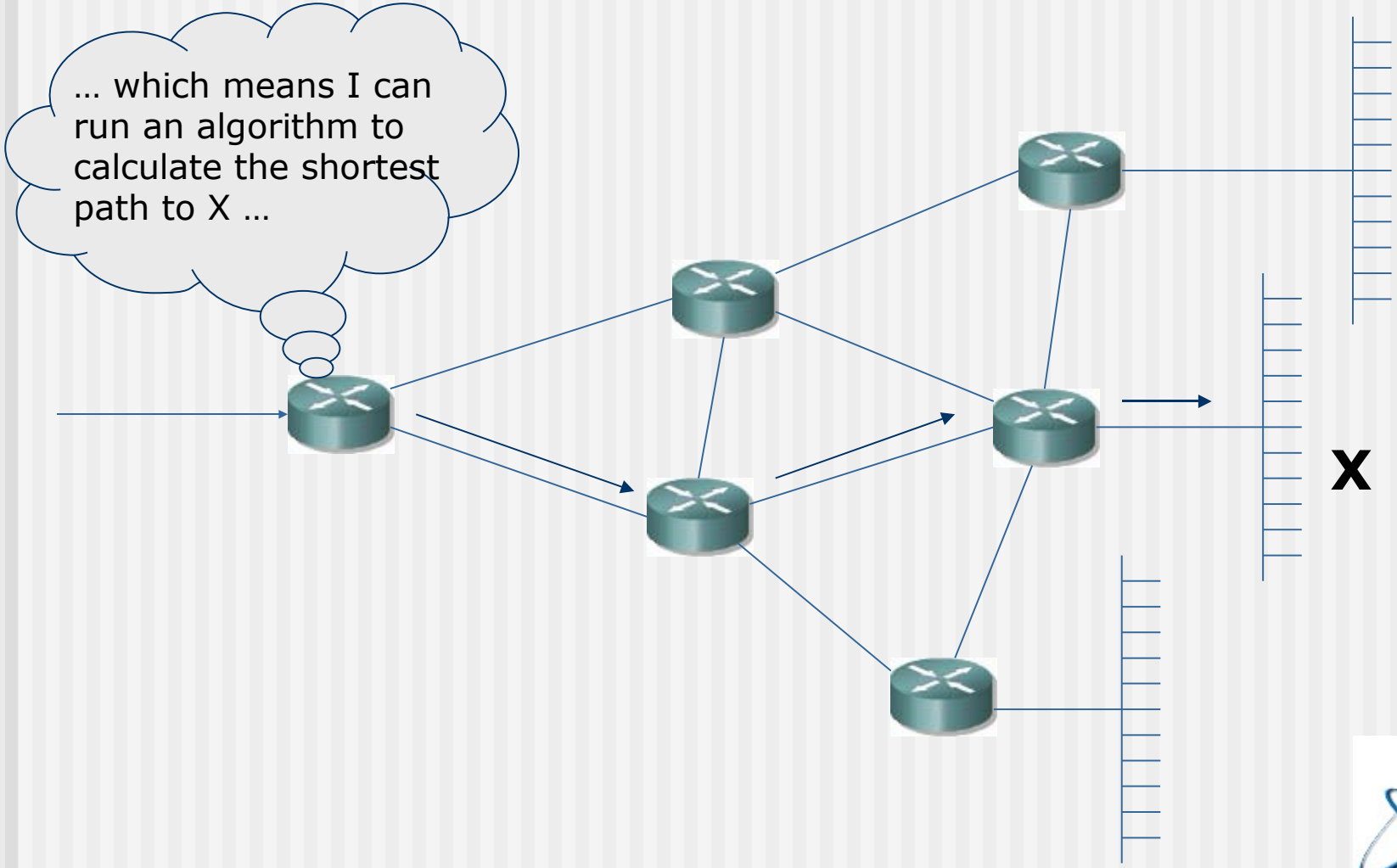
… and eventually …



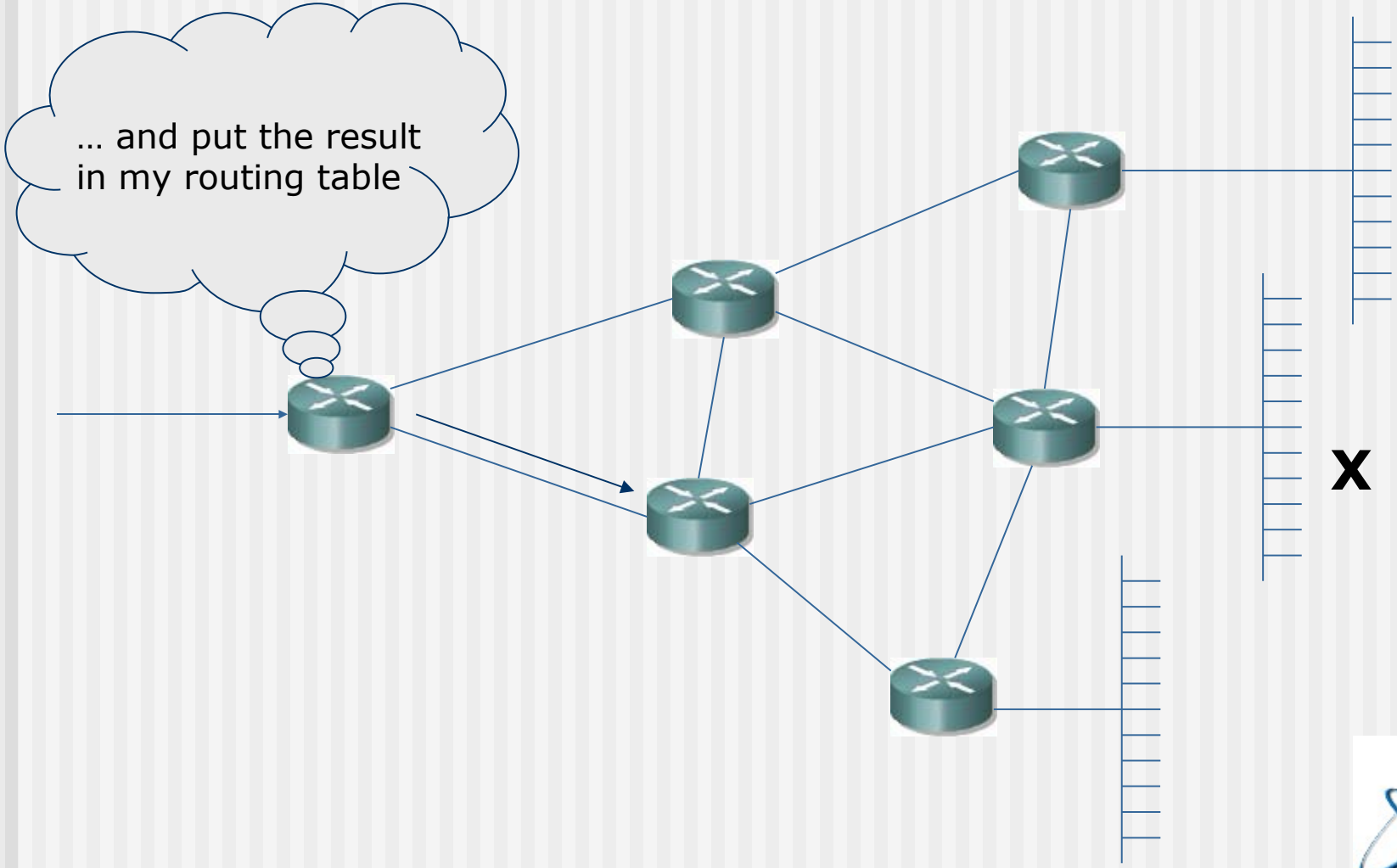
Link-State Routing



Link-State Routing



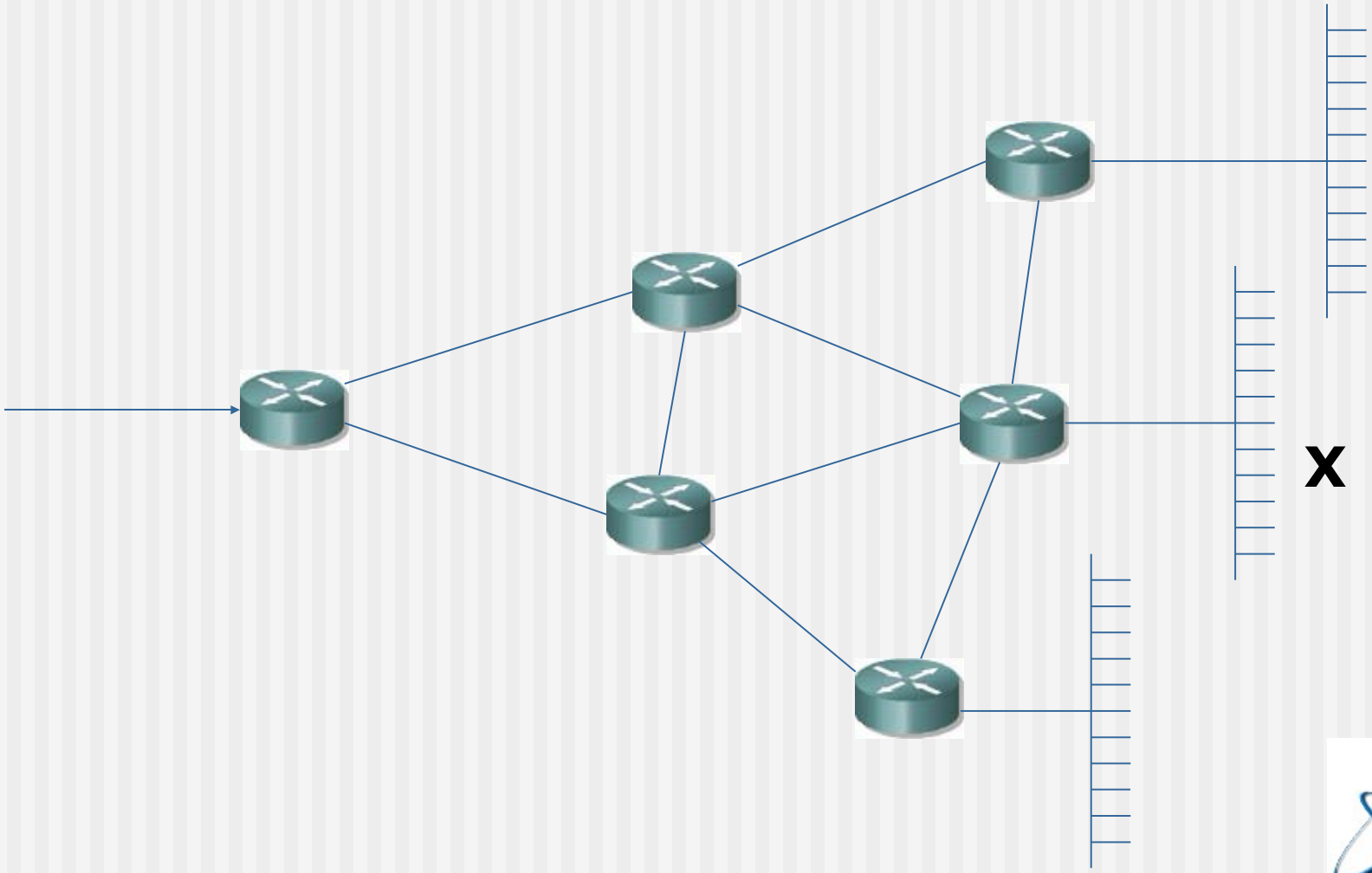
Link-State Routing



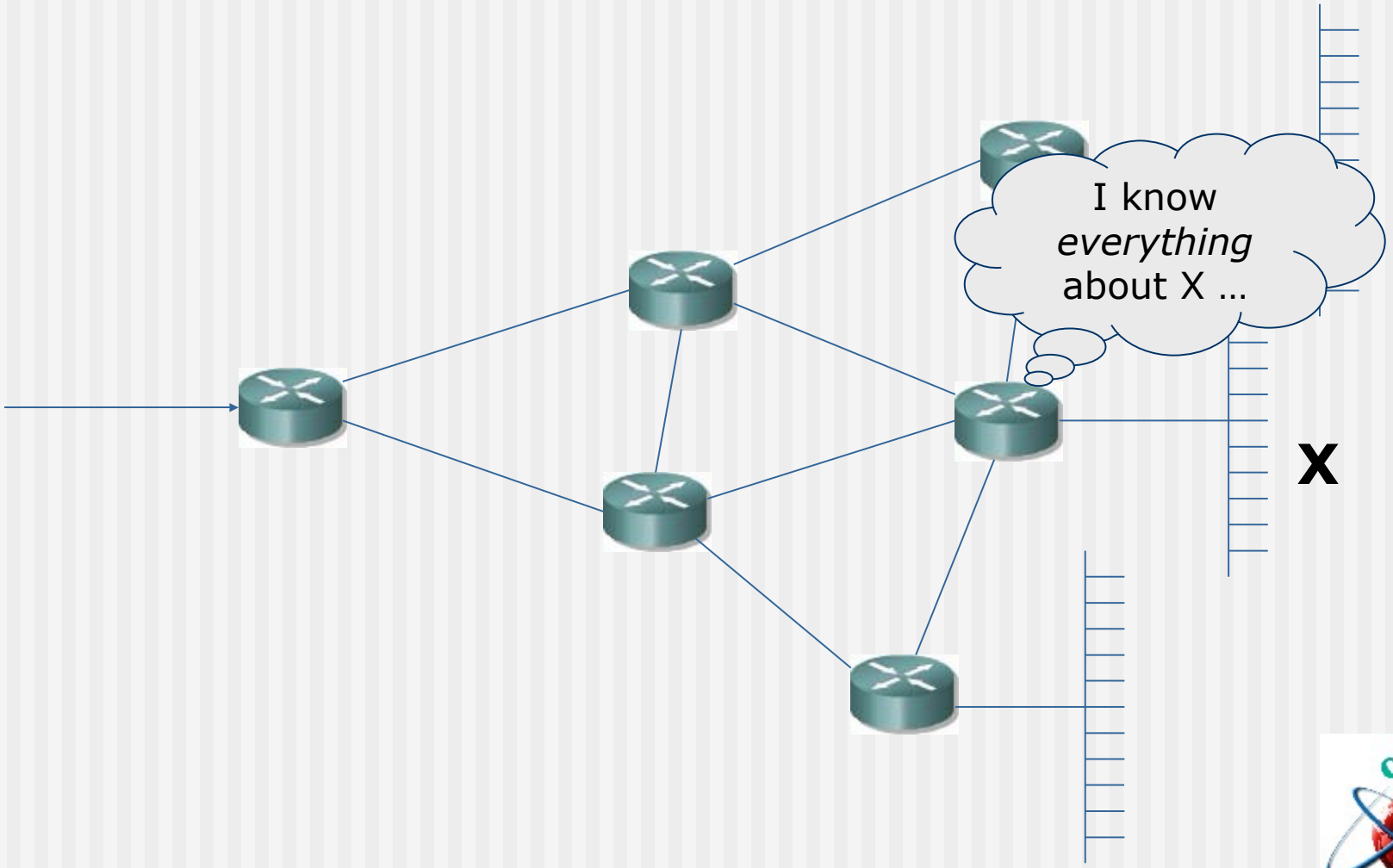
Hybrid Protocols

- eg, Cisco's EIGRP (Enhanced Interior Gateway Routing Protocol)
- Routers running a hybrid protocol …
  - (may) have a sophisticated cost calculation
    - bandwidth, delay, load, reliability, etc.
    - like an LS protocol, or better
  - (may) exchange information like a (multi-part) DV protocol
  - but only as required, like an LS protocol (probably)
- Hybrid protocols are …
  - very efficient (?)
  - even closer to optimal (?)
  - even faster to converge across the domain (?)

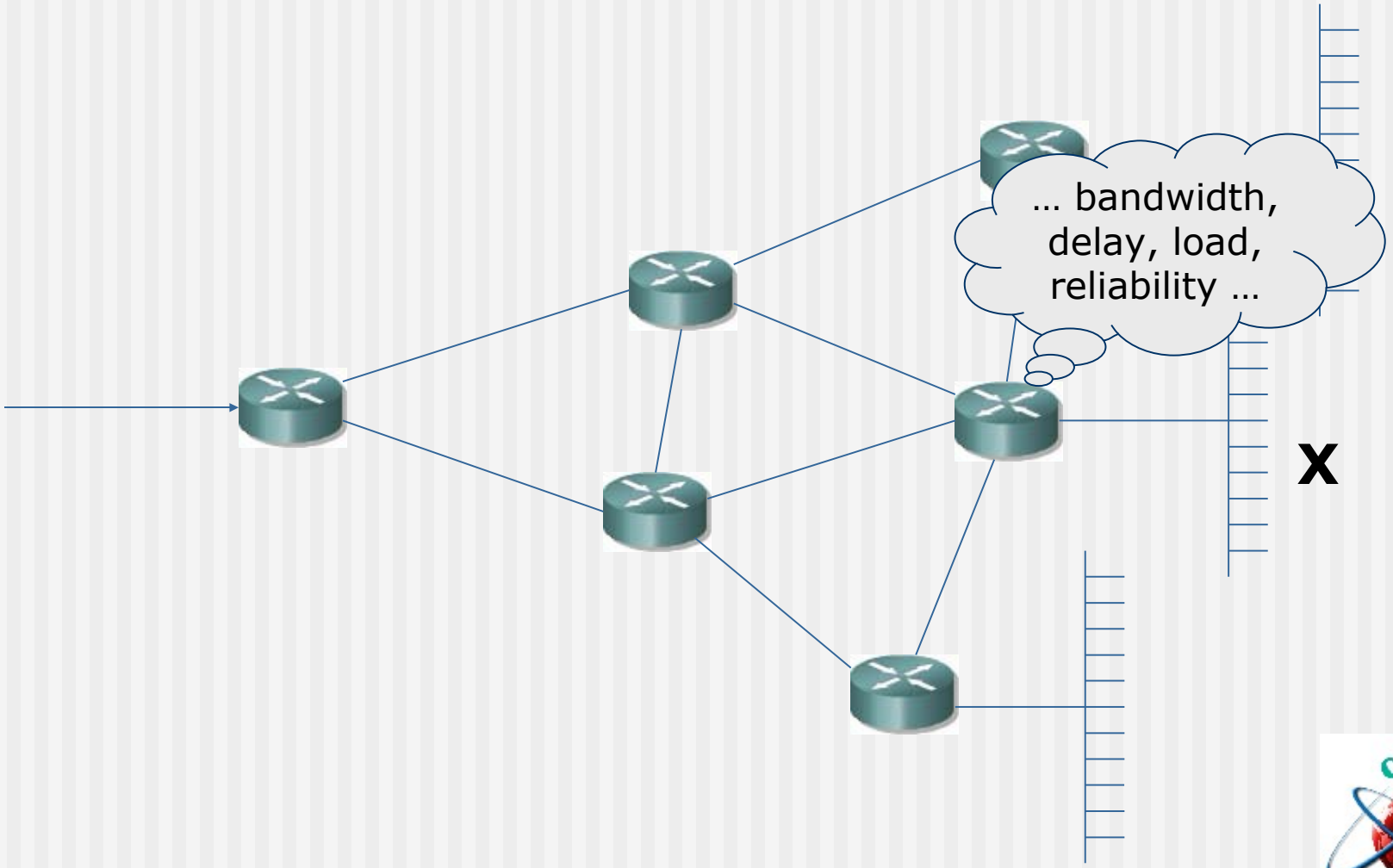
Hybrid Routing



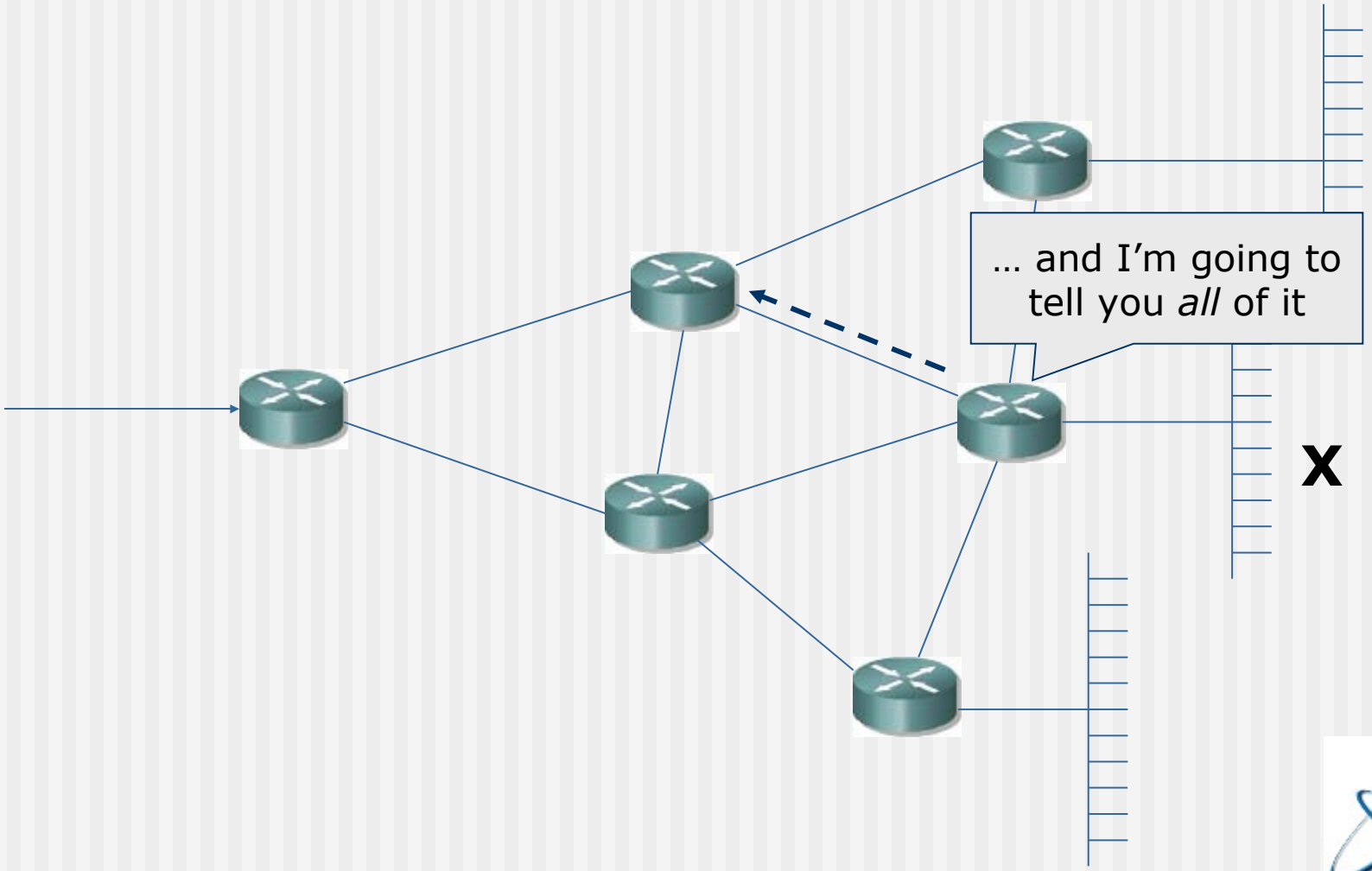
Hybrid Routing



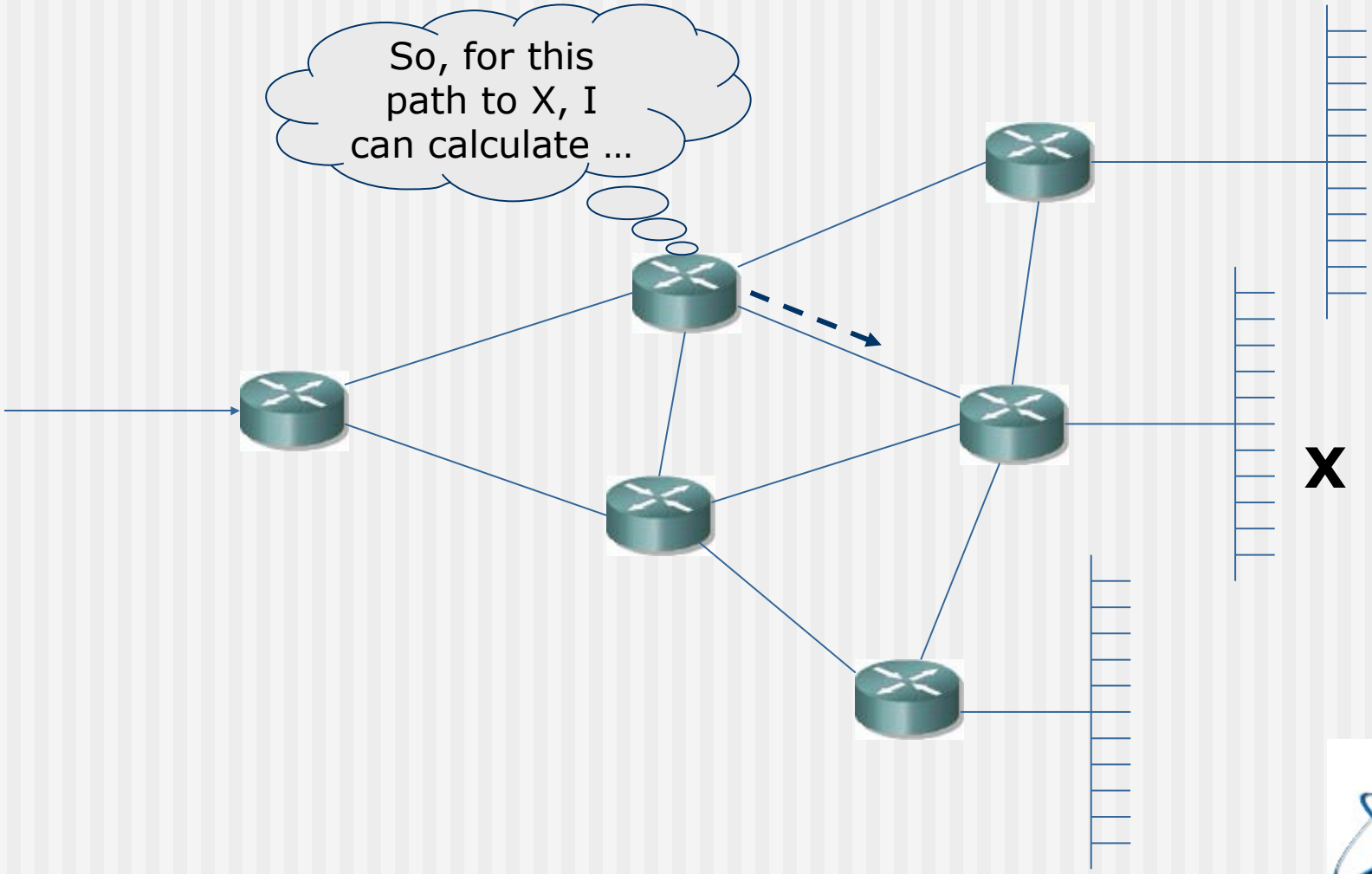
Hybrid Routing



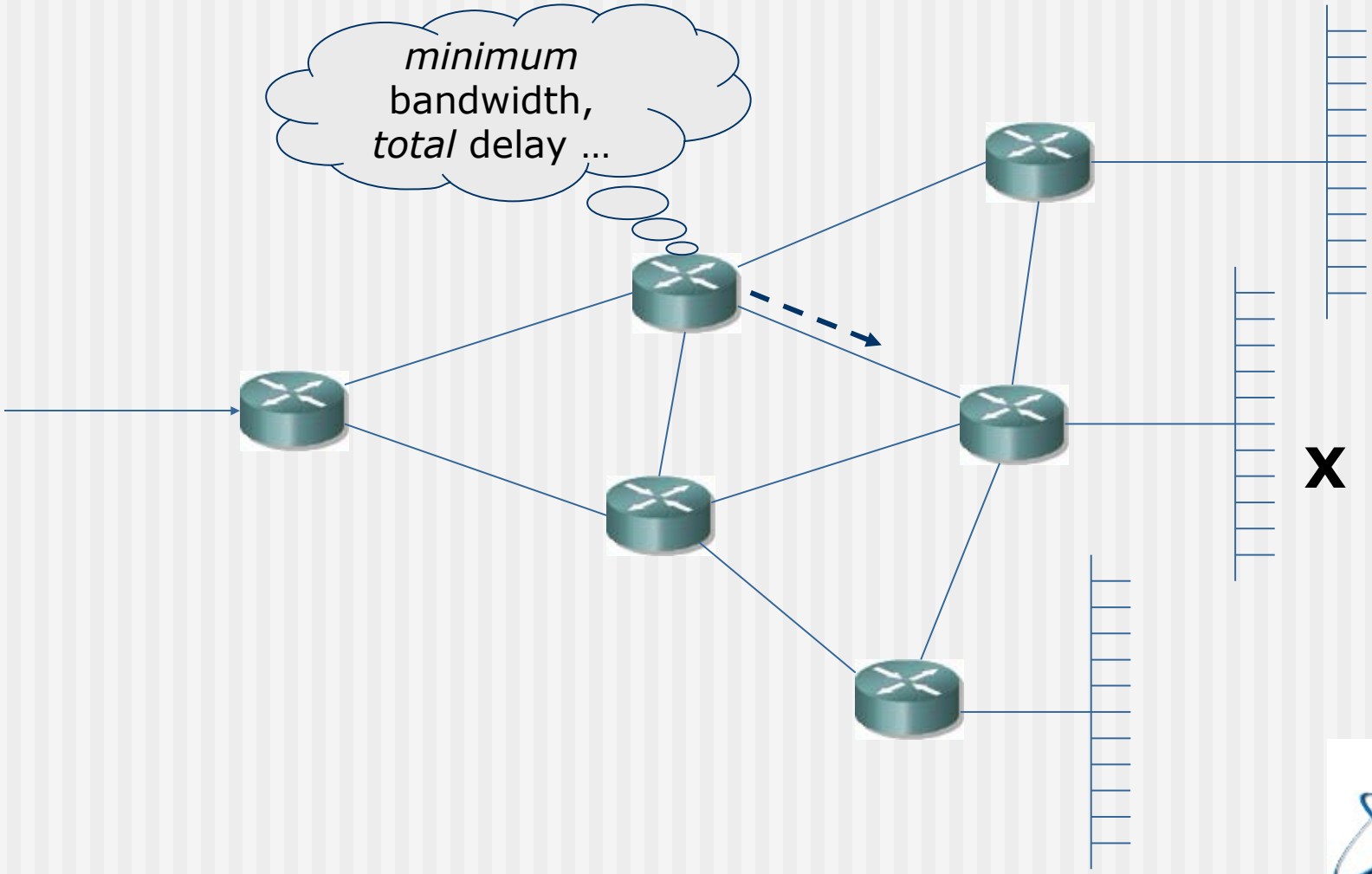
Hybrid Routing



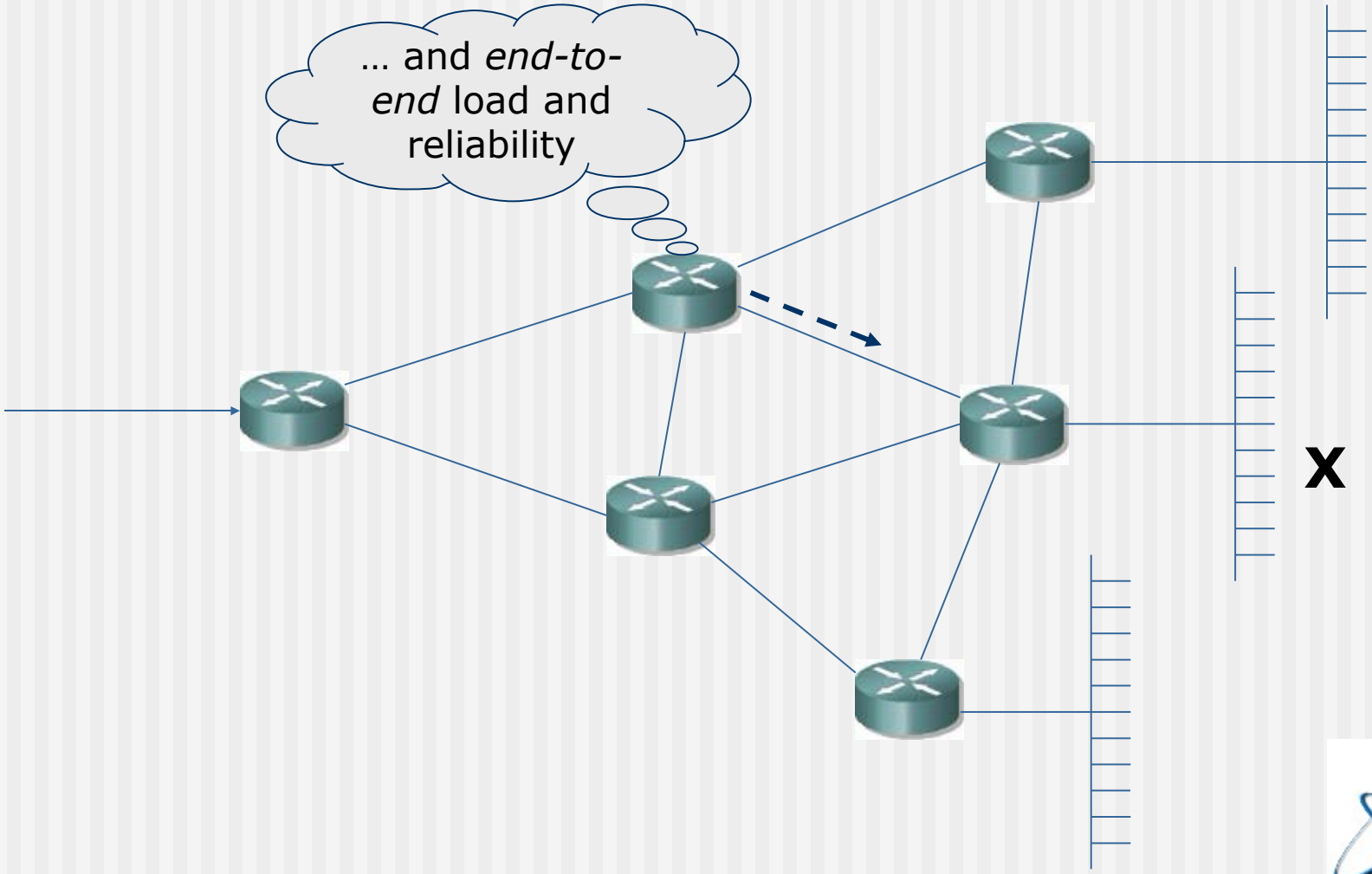
Hybrid Routing



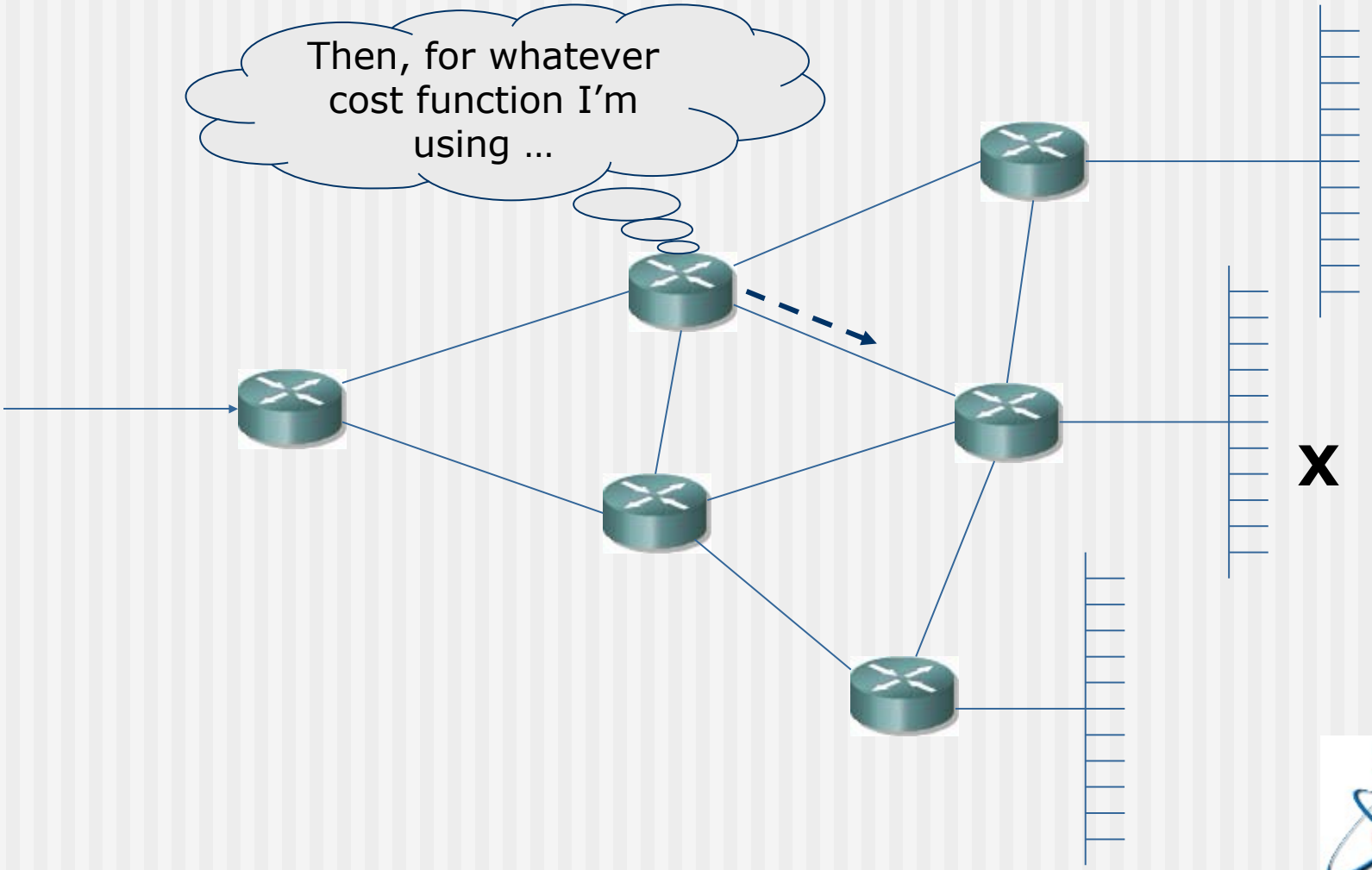
Hybrid Routing



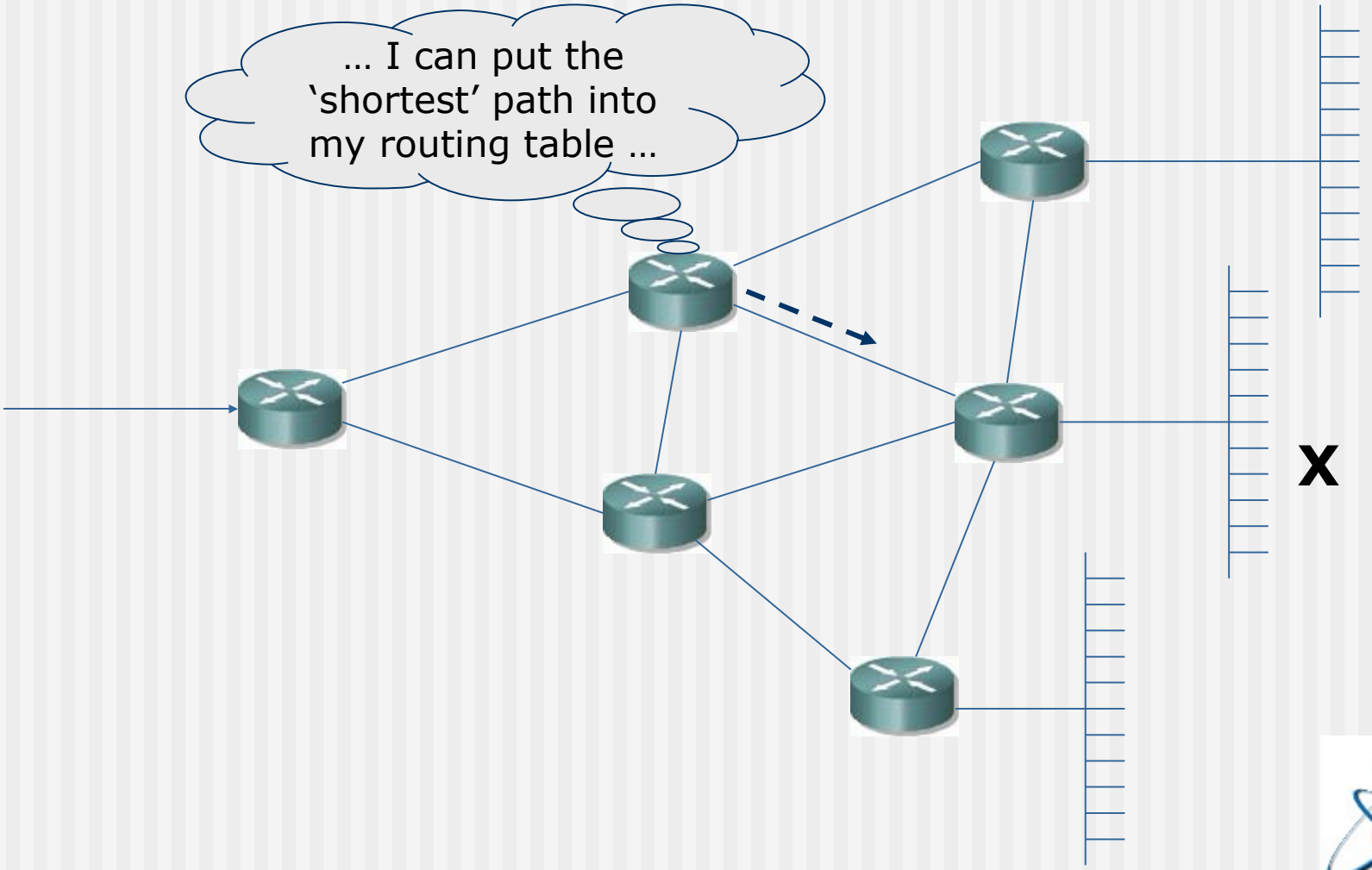
Hybrid Routing



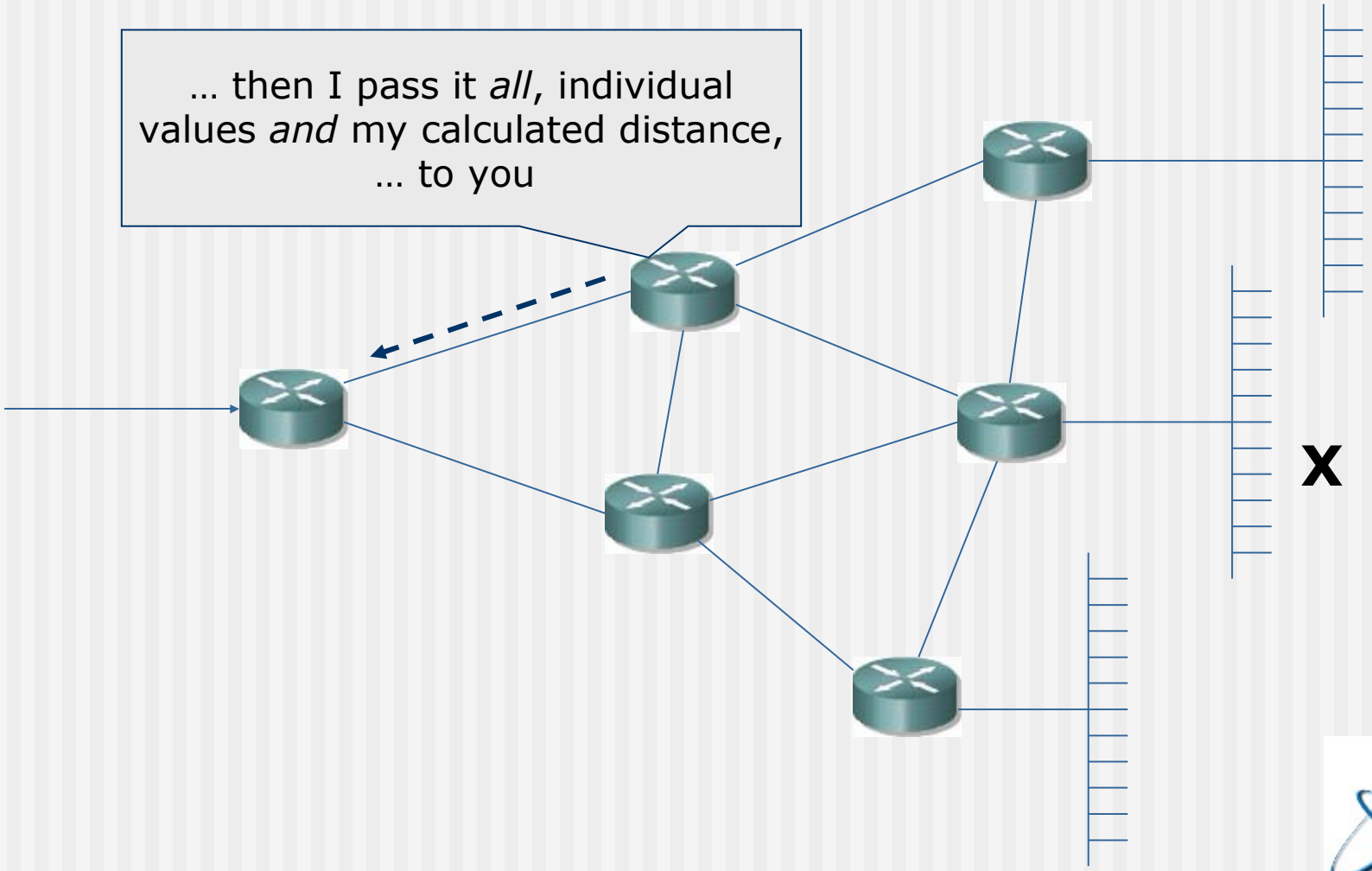
Hybrid Routing



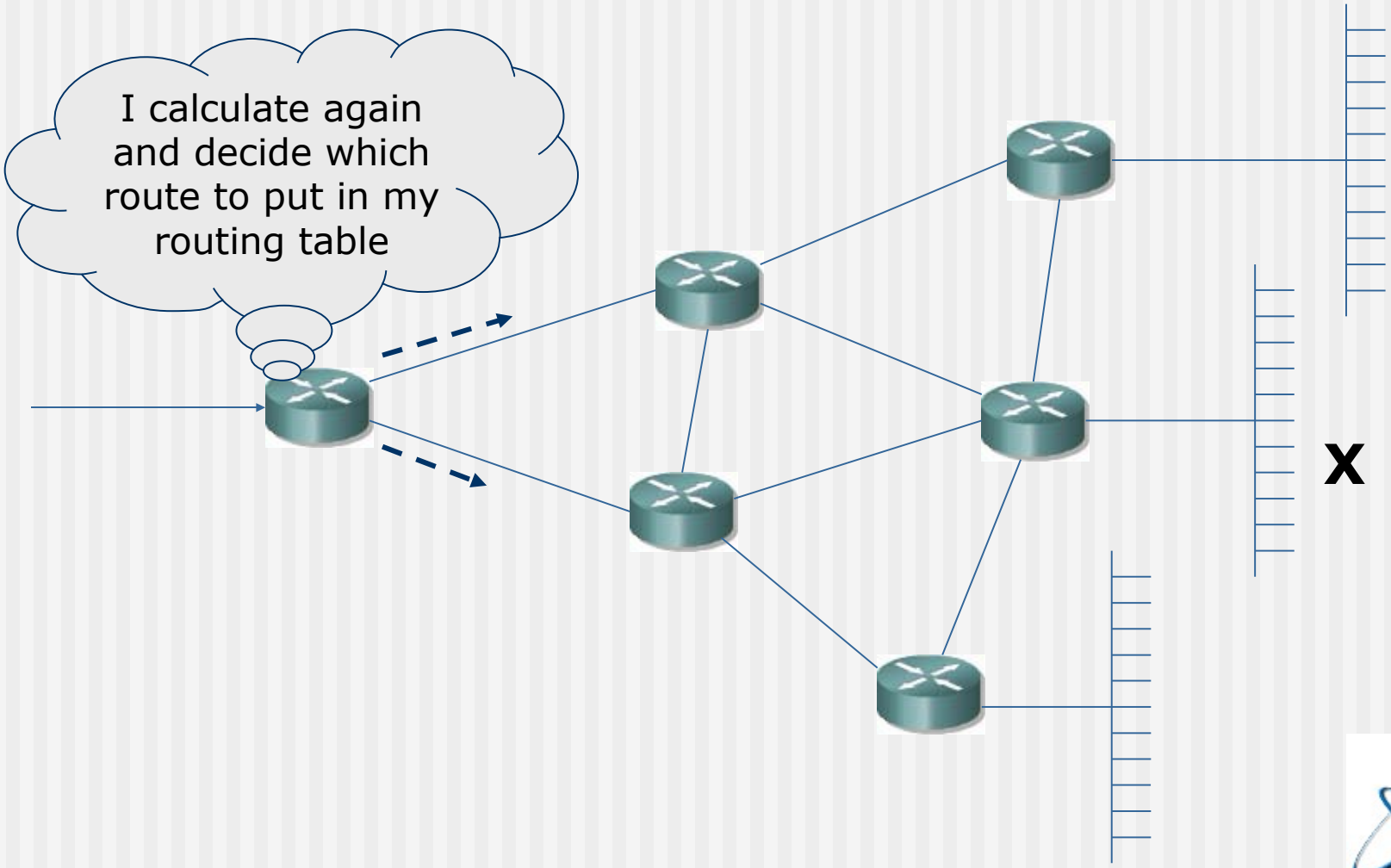
Hybrid Routing



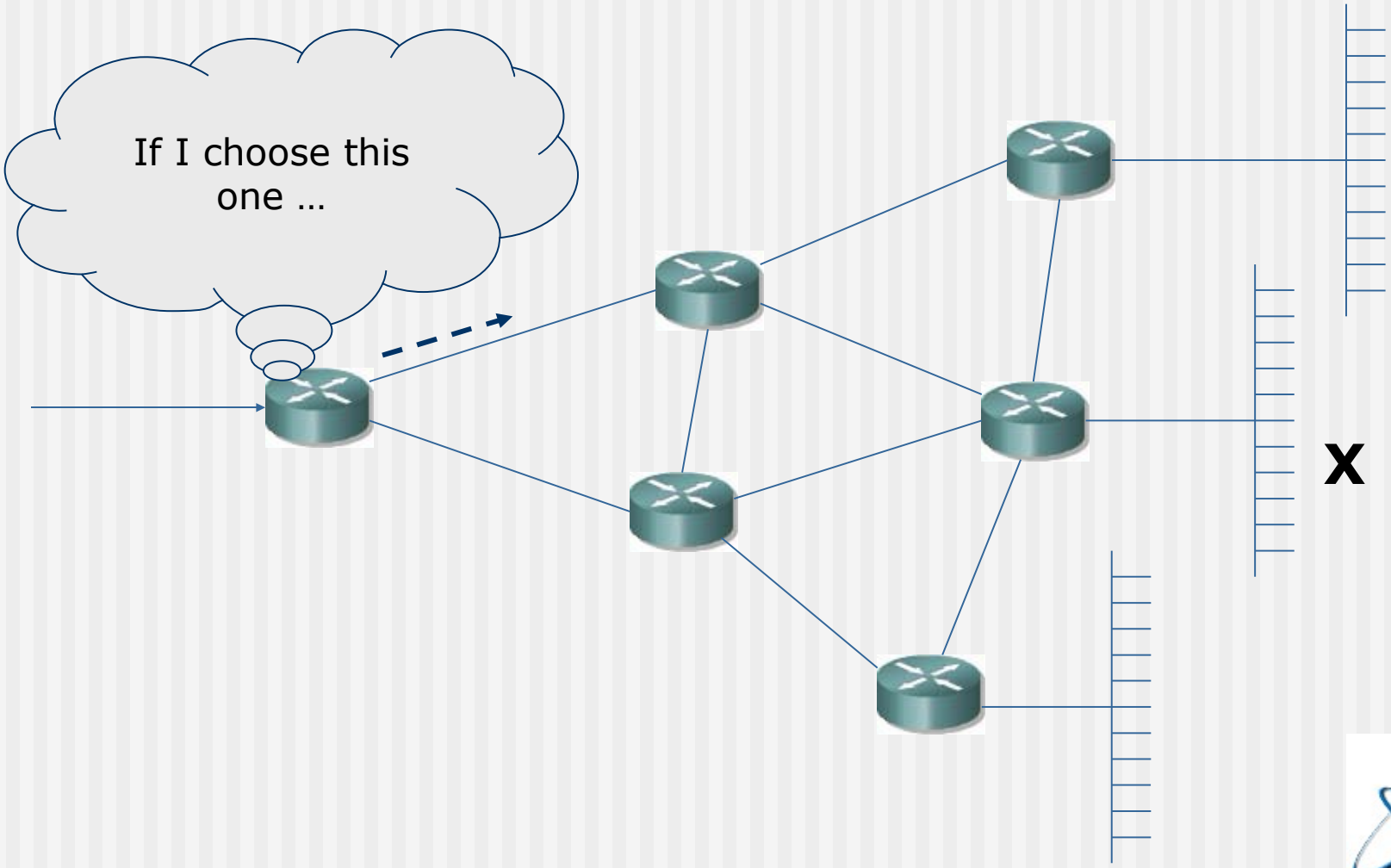
Hybrid Routing



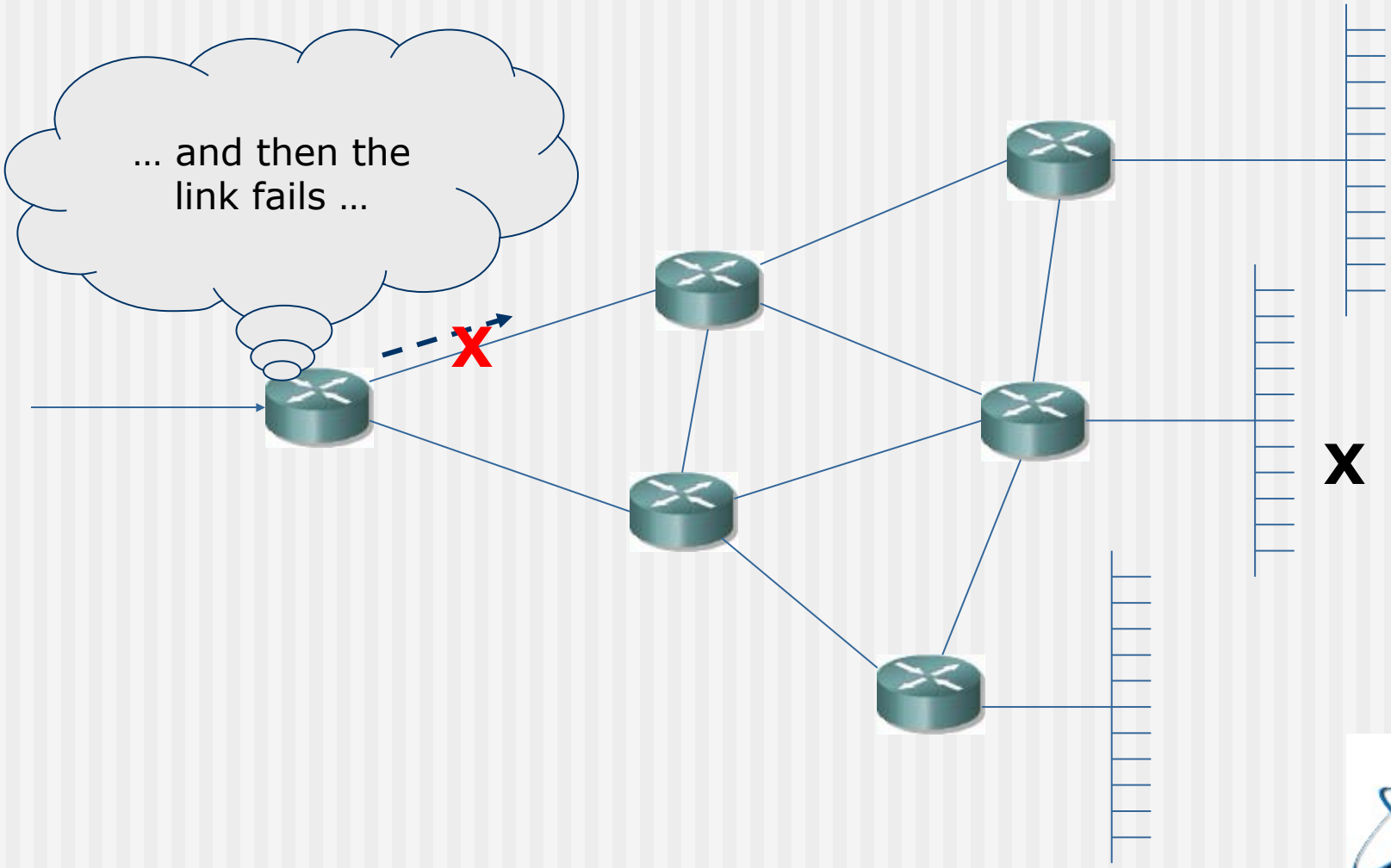
Hybrid Routing



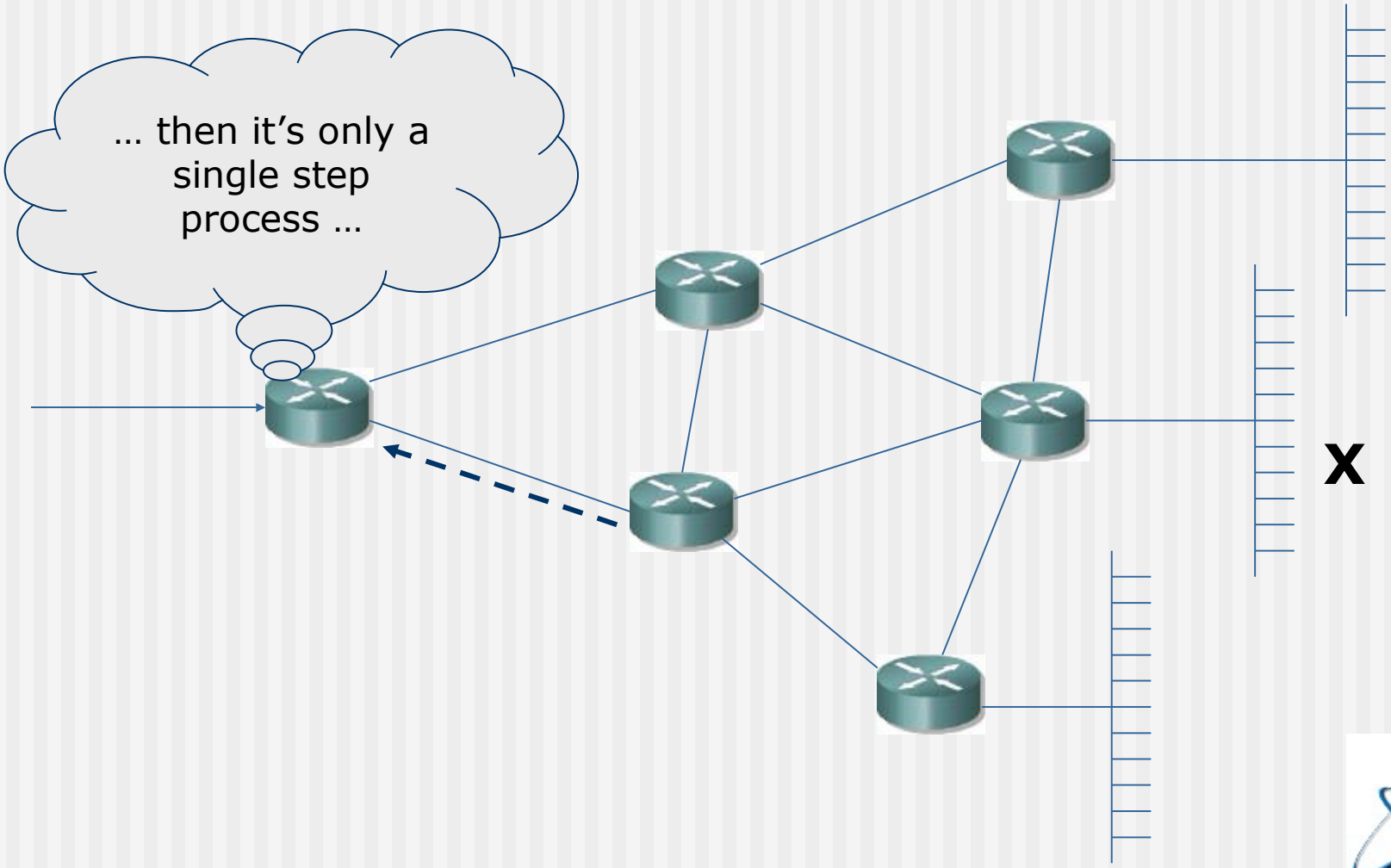
Hybrid Routing



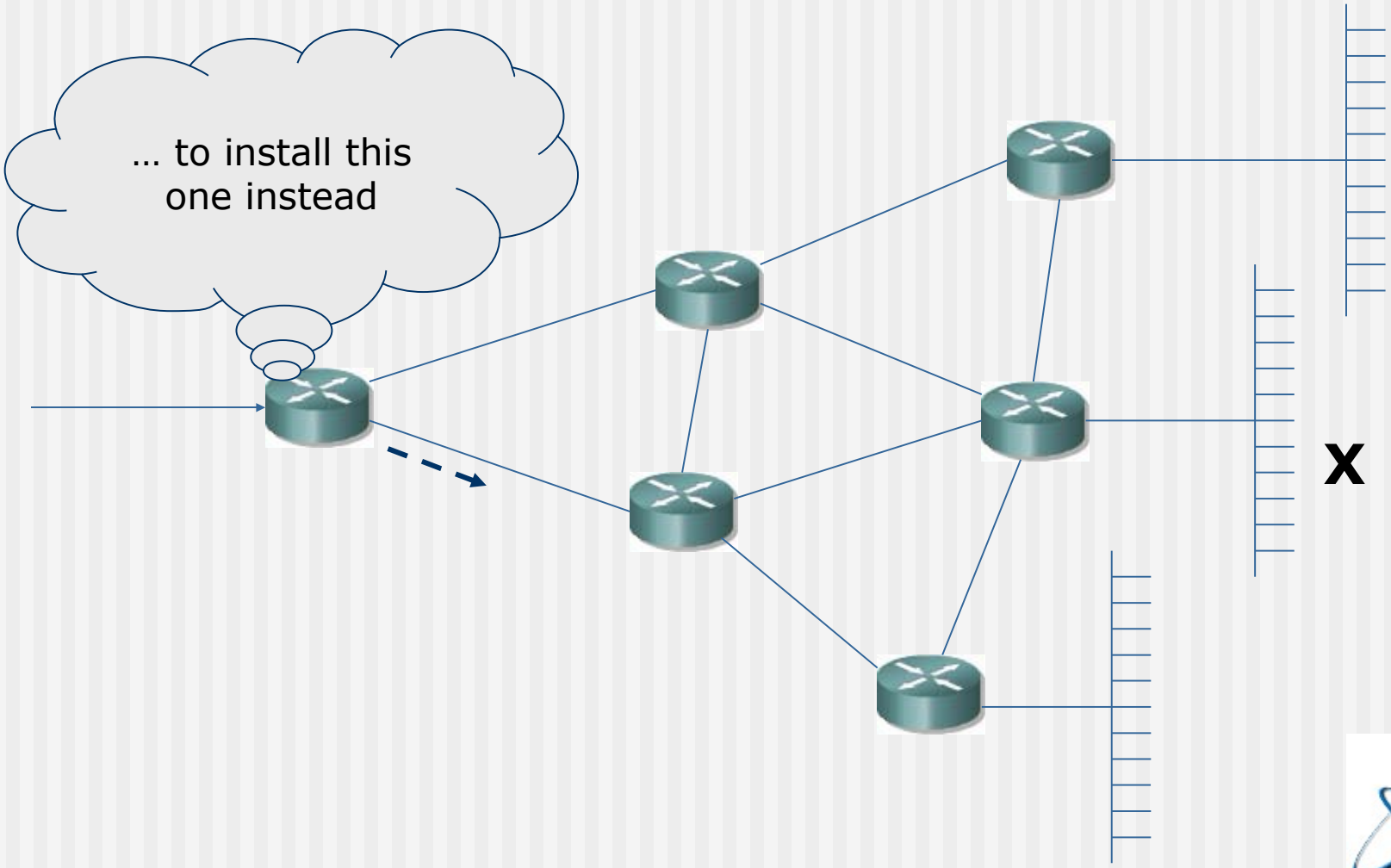
Hybrid Routing



Hybrid Routing

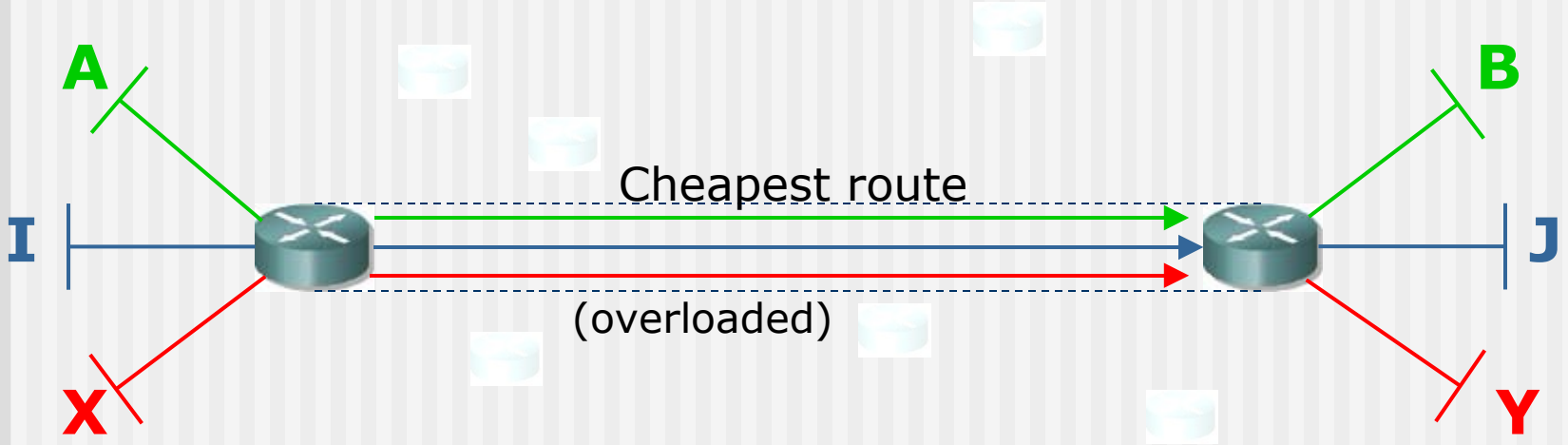


Hybrid Routing



A Problem with All Routing

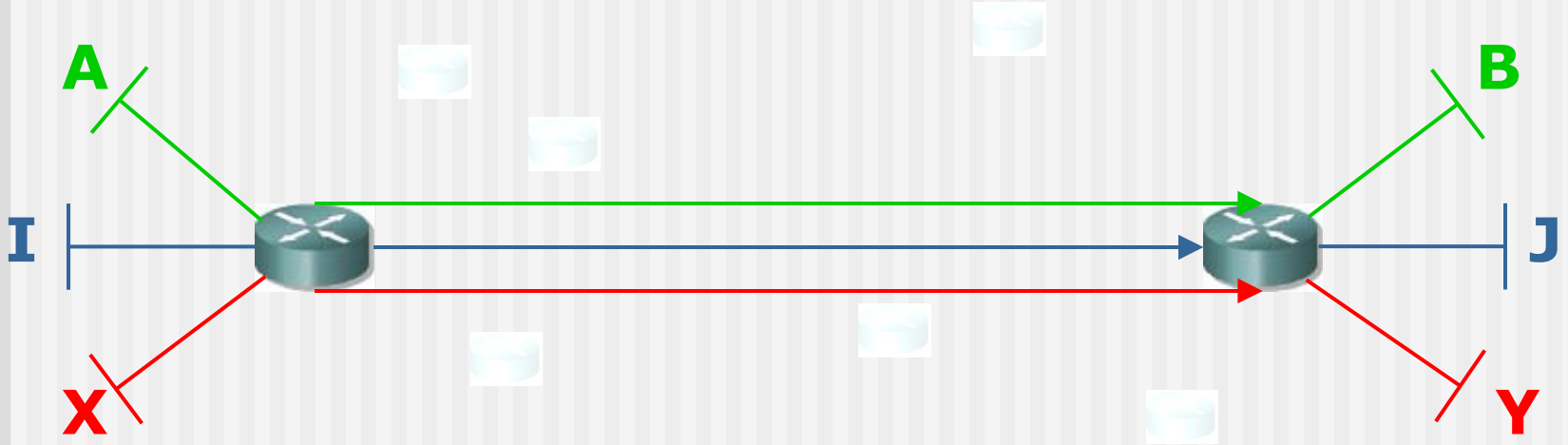
- *All* these routing protocols/algorithms look to optimise routes *individually*
- *Individually* optimised routes do not give the best overall *domain* routing



Individually optimised routes

A Problem with All Routing

- *All* these routing protocols/algorithms look to optimise routes *individually*
- *Individually* optimised routes do not give the best overall *domain* routing



Domain optimised routes

The Route-Finder Analogy

A shortest path algorithm will find the 'best' route between two points … just like a routing protocol.

The Route-Finder Analogy

A shortest path algorithm will find the 'best' route between two points … just like a routing protocol.

The Route-Finder Analogy

A shortest path algorithm will find the 'best' route between two points … just like a routing protocol.

But what if many cars want to take the same journey? Might it be better to send some a different way?

The Route-Finder Analogy

A shortest path algorithm will find the 'best' route between two points … just like a routing protocol.

But what if many cars want to take the same journey? Might it be better to send some a different way?

The Route-Finder Analogy

Actually, some of the more advanced protocols (such as EIGRP) already do this …

'Load balancing'

But load balancing only applies to different routes to the same destination

The Route-Finder Analogy

Actually, some of the more advanced protocols (such as EIGRP) already do this …

'Load balancing'

But load balancing only applies to different routes to the same destination

The Route-Finder Analogy

What about all of the other routes?

How can we take into account how all this traffic will

- *interact*
- *compete*

for the available links?

The Route-Finder Analogy

What about all of the other routes?

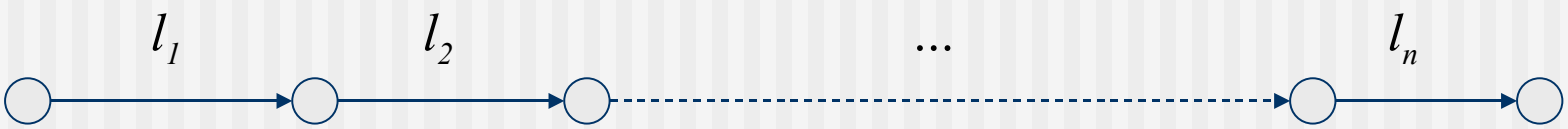
How can we take into account how all this traffic will

- *interact*
- *compete*

for the available links?

There is no simple answer! (NP-complete)

A Closer look at OSPF



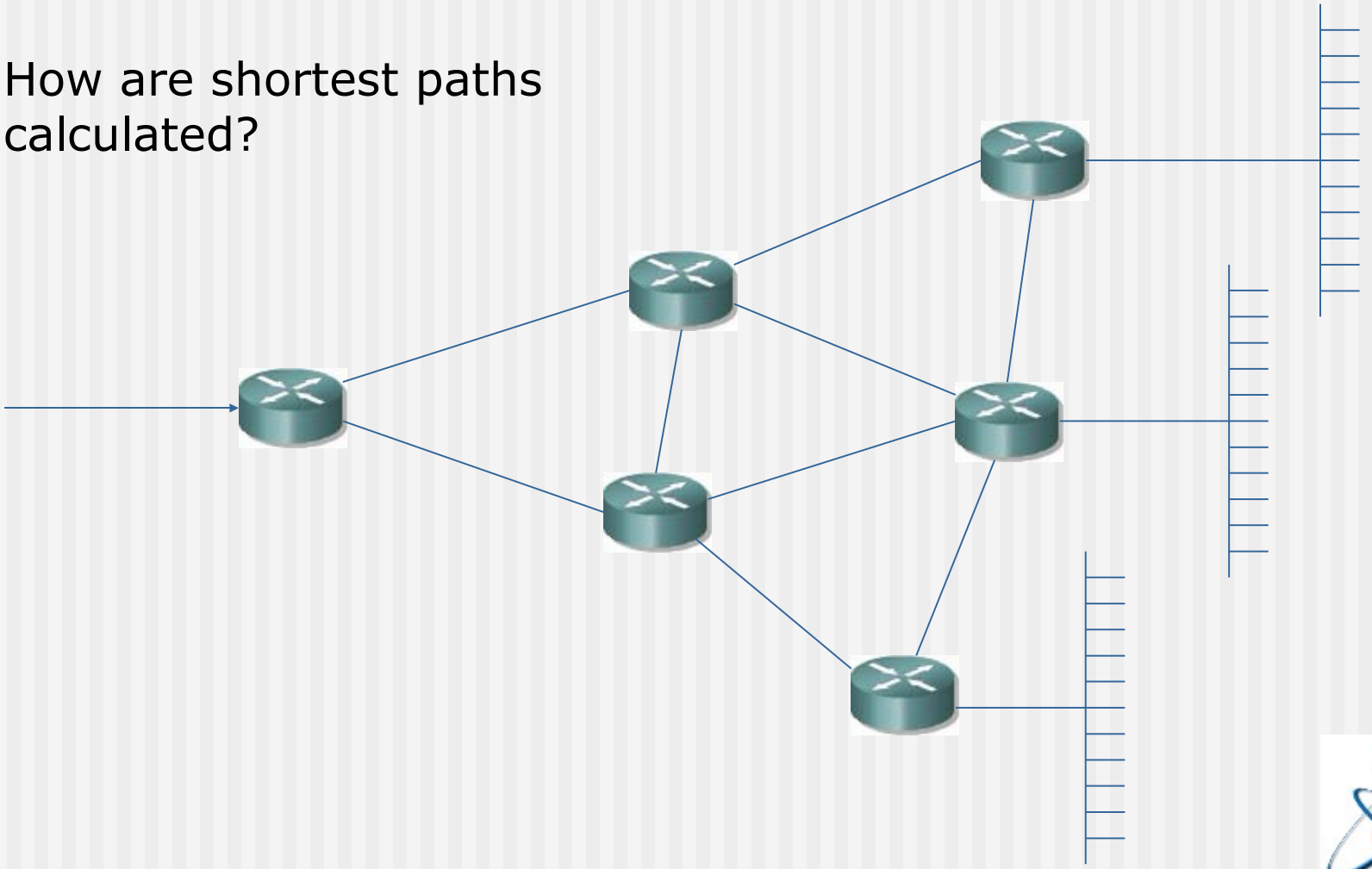
- Cost of route $l_1, l_2 \ldots l_n$ given by bandwidths:

$$\frac{1}{b_1} + \frac{1}{b_2} + \ldots + \frac{1}{b_n}$$

- Doesn't actually make much sense!
- Could argue that a low bandwidth link dominates the calculation but, in that case, why add them at all?
- Why not just measure the *minimum* bandwidth?
- Actually there are several possible implementations

A Closer look at OSPF

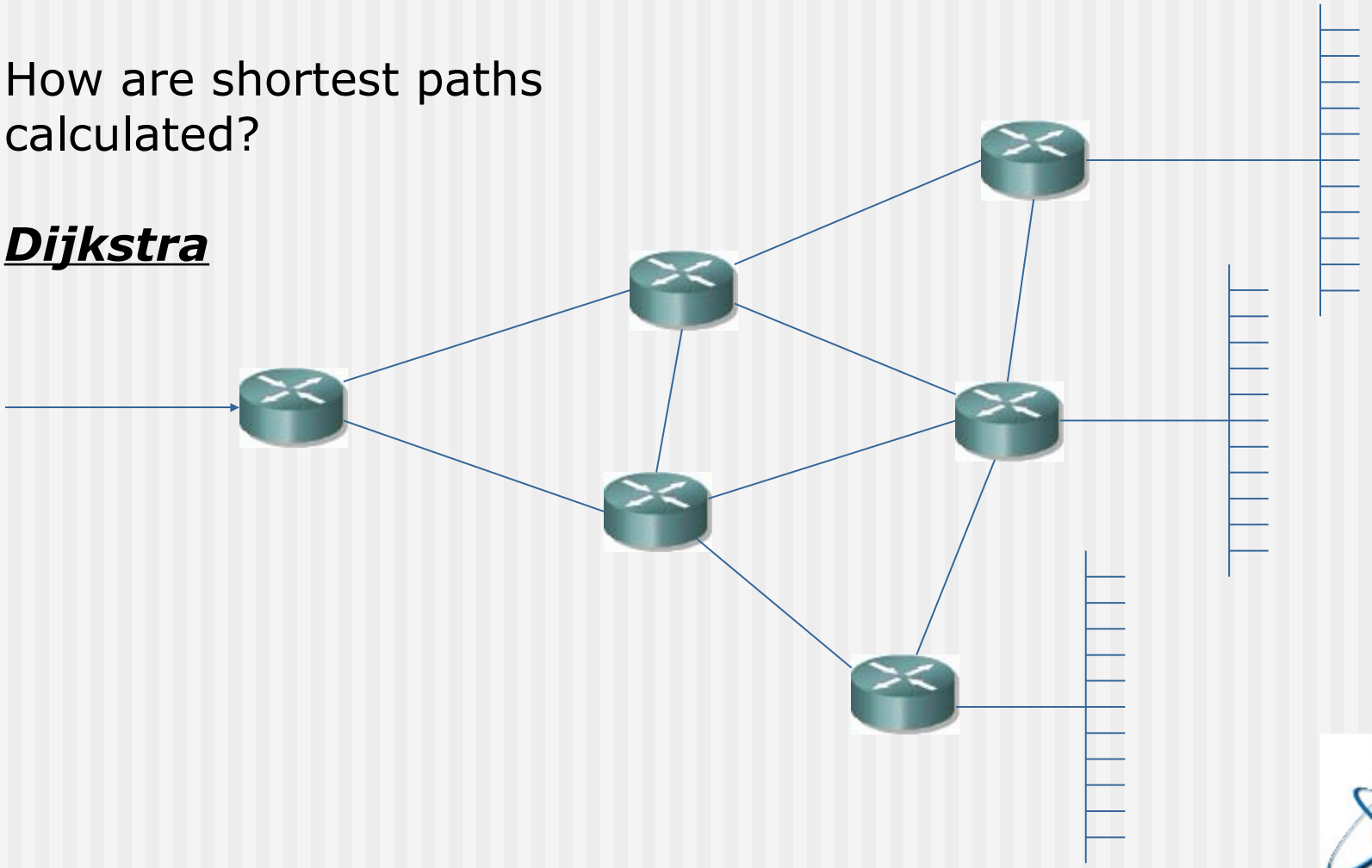
How are shortest paths calculated?



A Closer look at OSPF

How are shortest paths calculated?

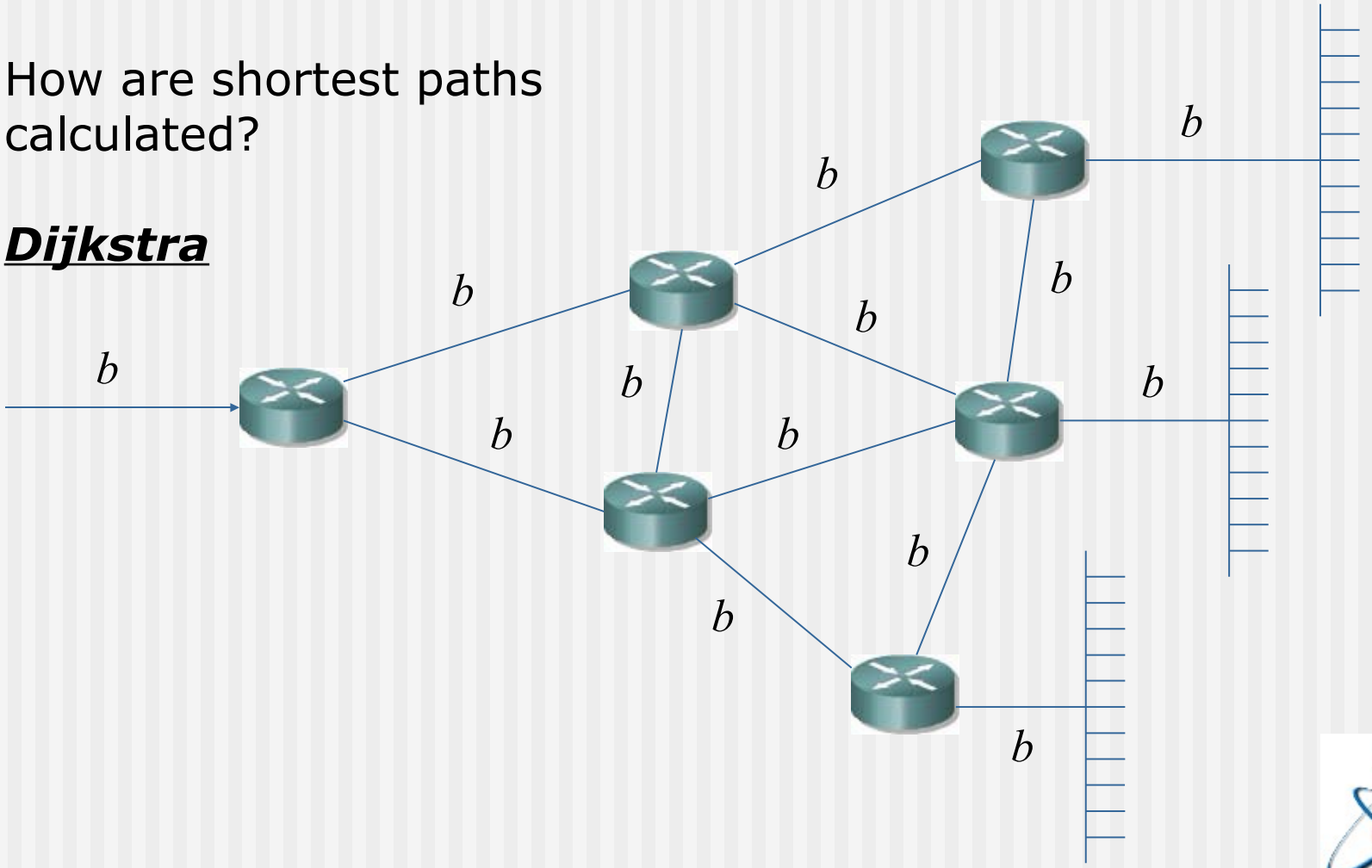
Dijkstra



A Closer look at OSPF

How are shortest paths calculated?

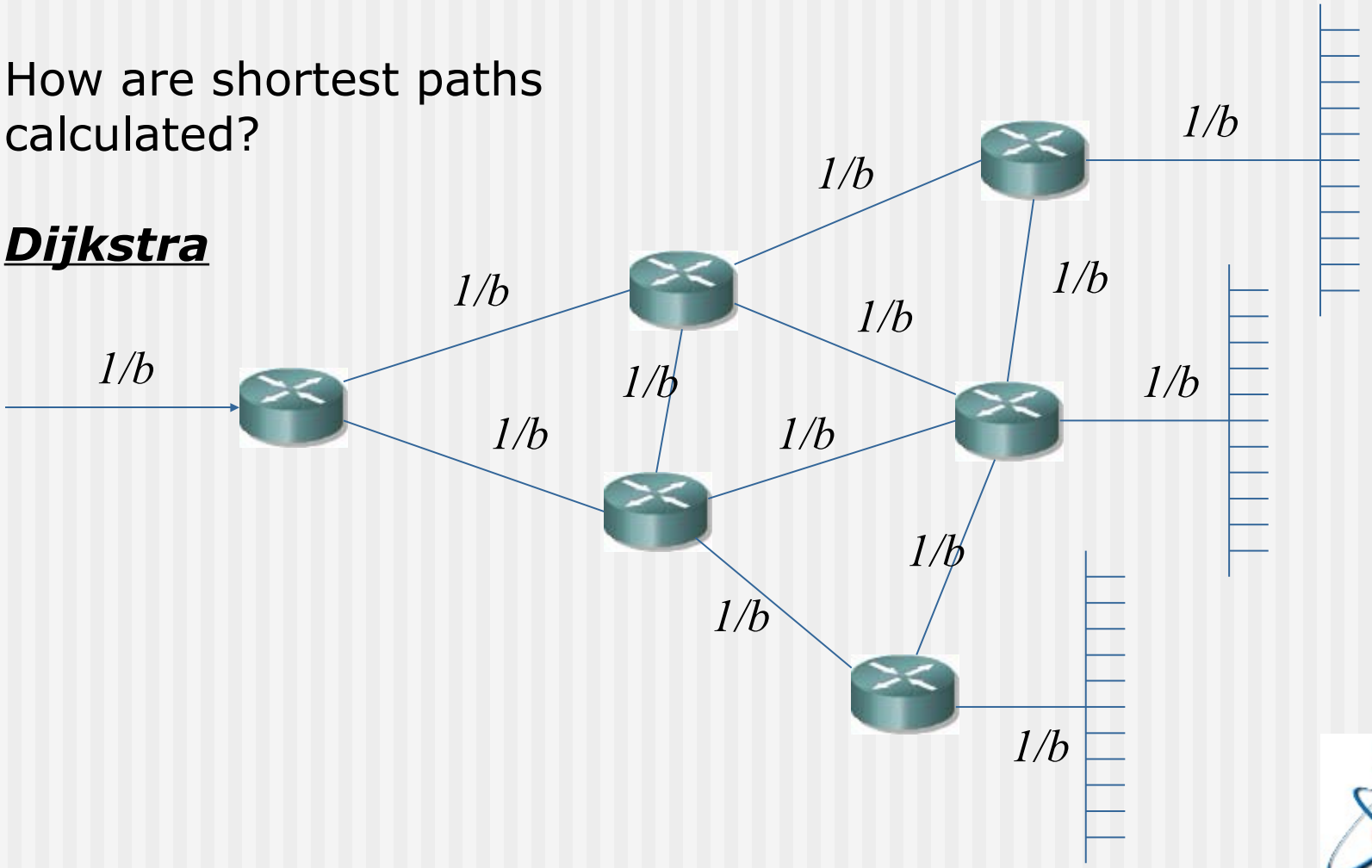
Dijkstra



A Closer look at OSPF

How are shortest paths calculated?

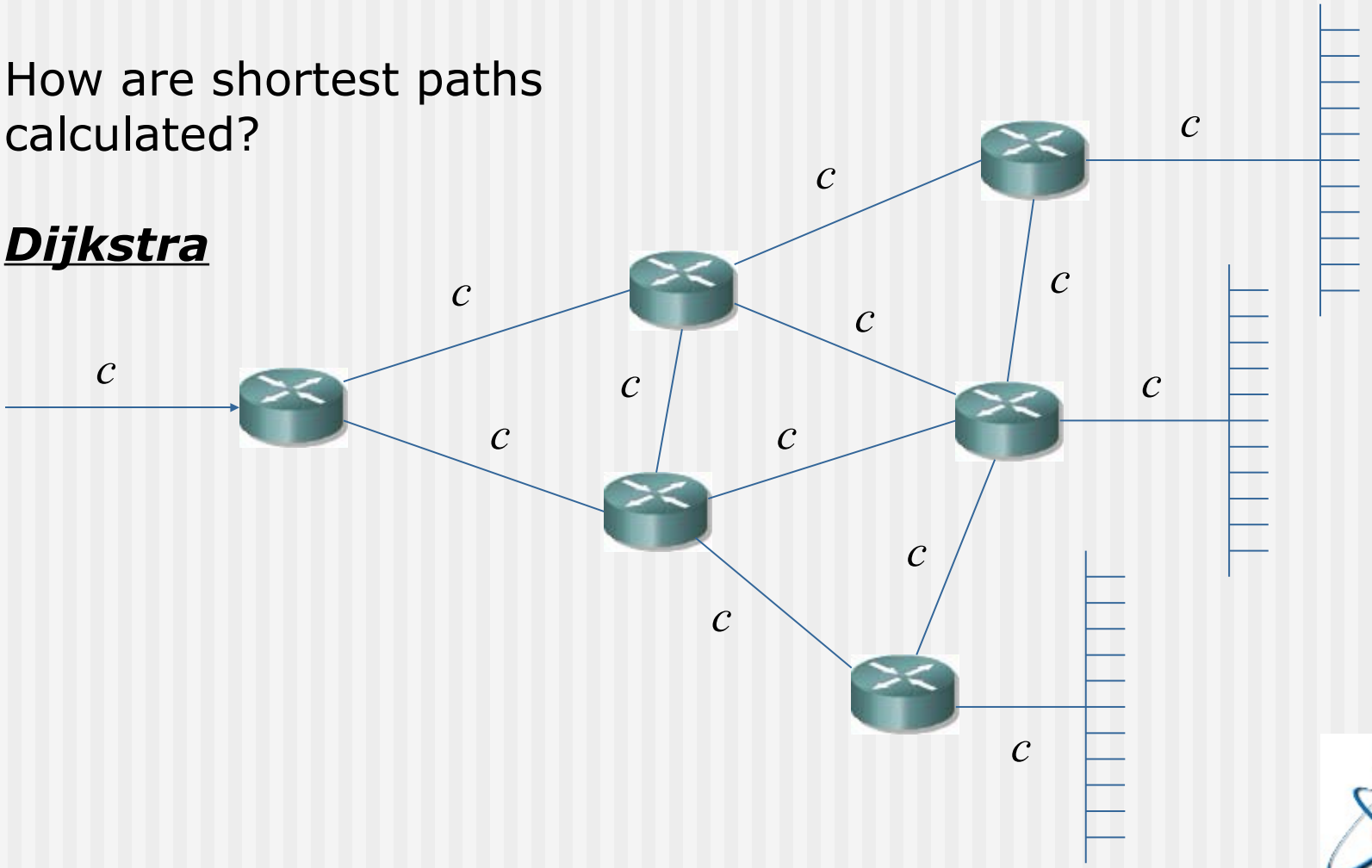
Dijkstra



A Closer look at OSPF

How are shortest paths calculated?

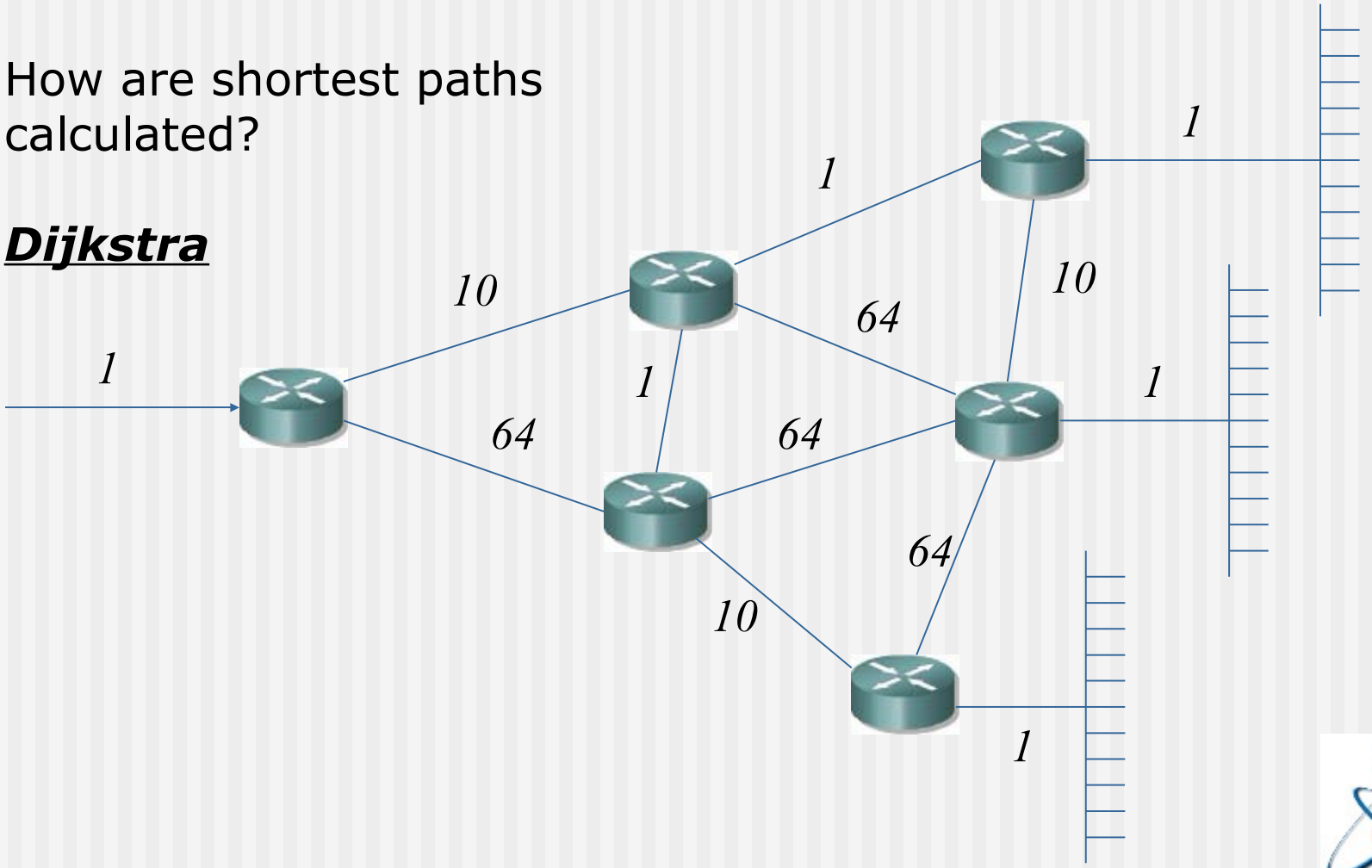
Dijkstra



A Closer look at OSPF

How are shortest paths calculated?

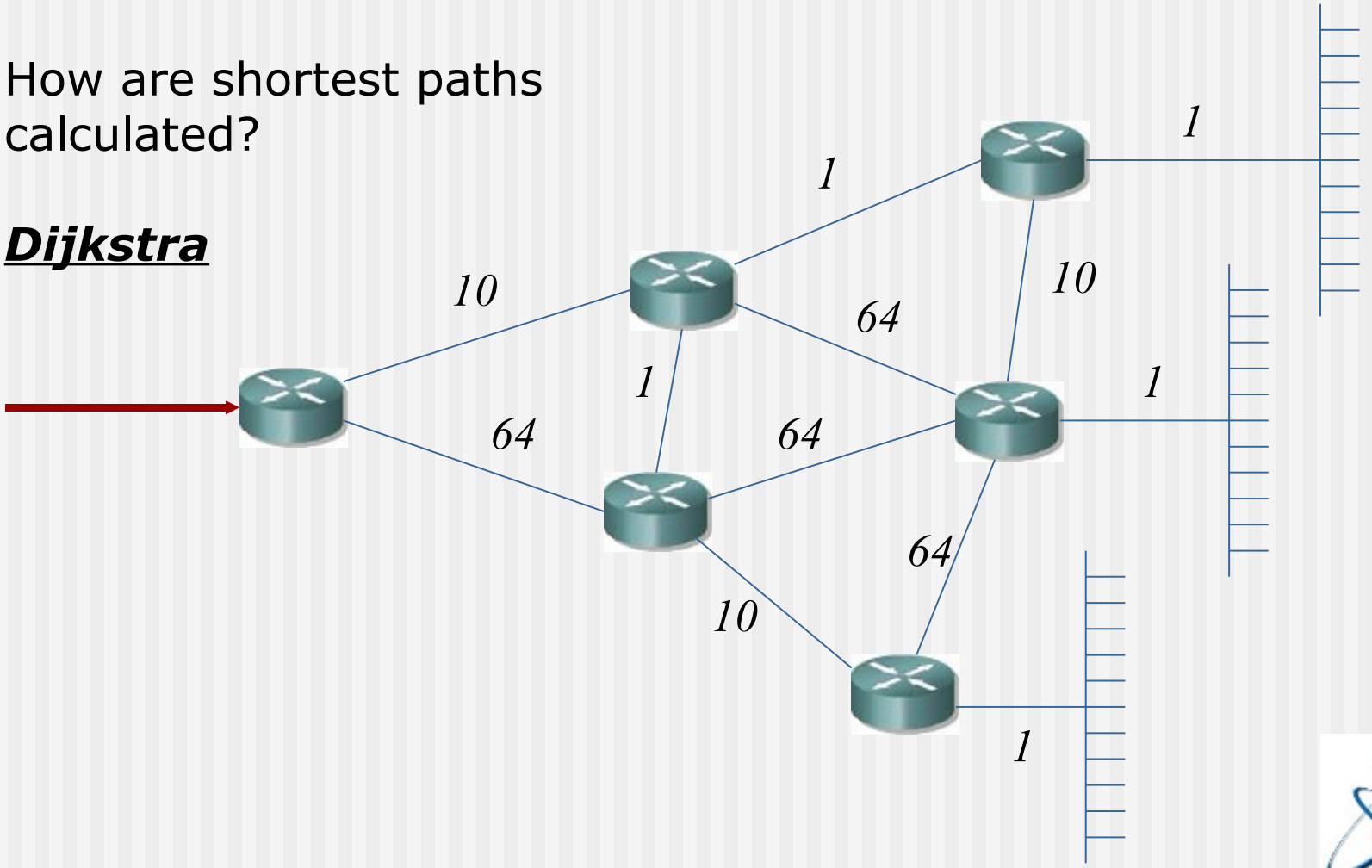
Dijkstra



A Closer look at OSPF

How are shortest paths calculated?

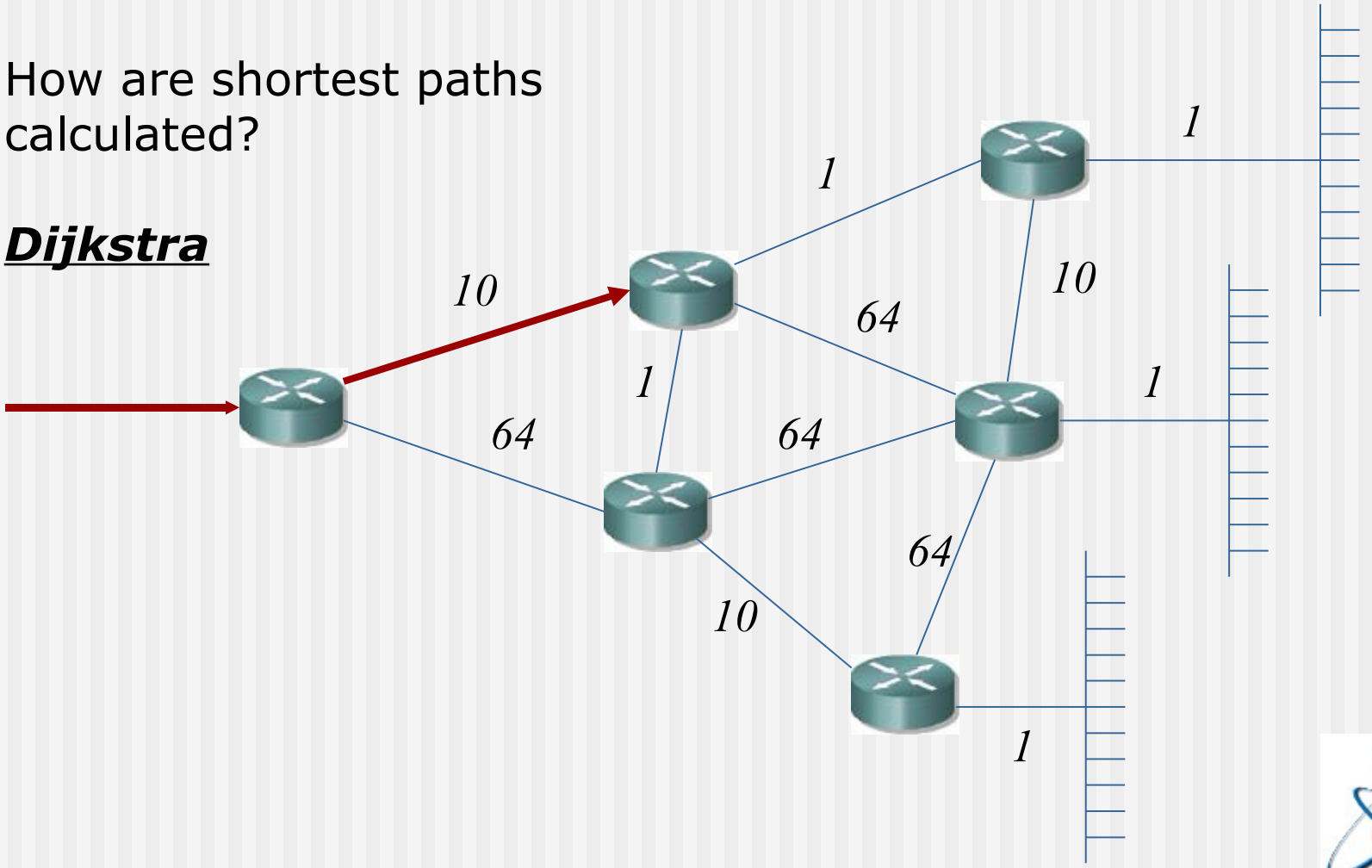
Dijkstra



A Closer look at OSPF

How are shortest paths calculated?

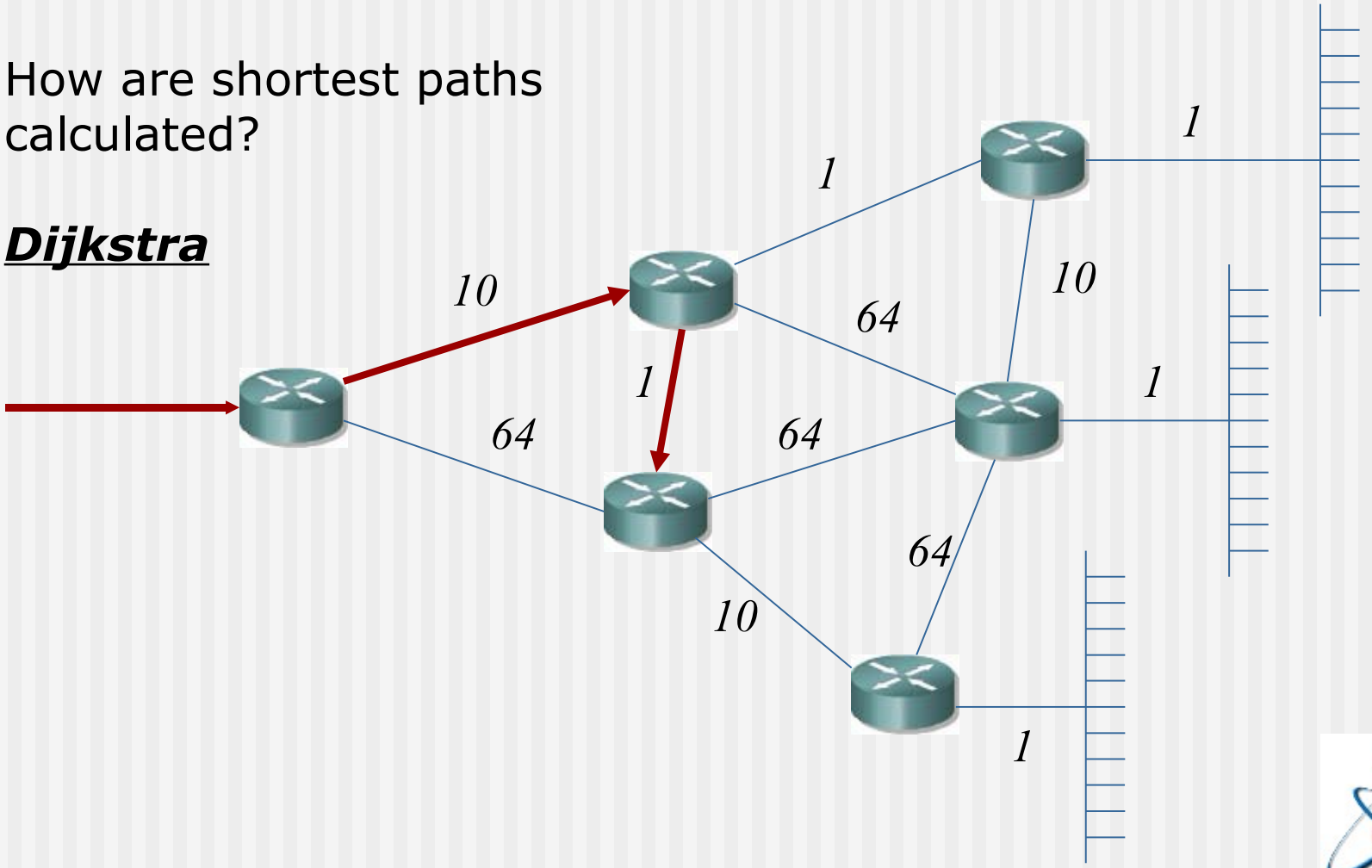
Dijkstra



A Closer look at OSPF

How are shortest paths calculated?

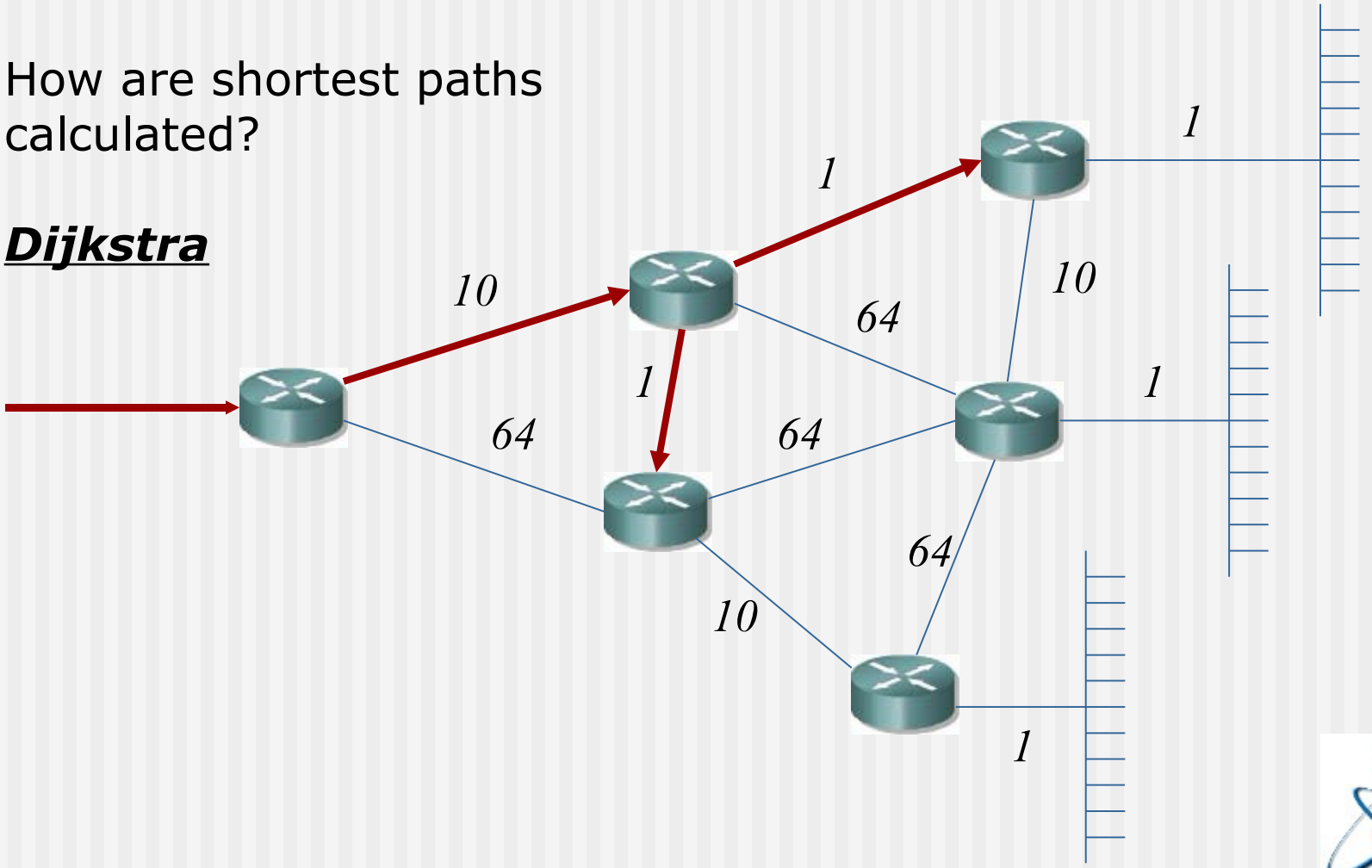
Dijkstra



A Closer look at OSPF

How are shortest paths calculated?

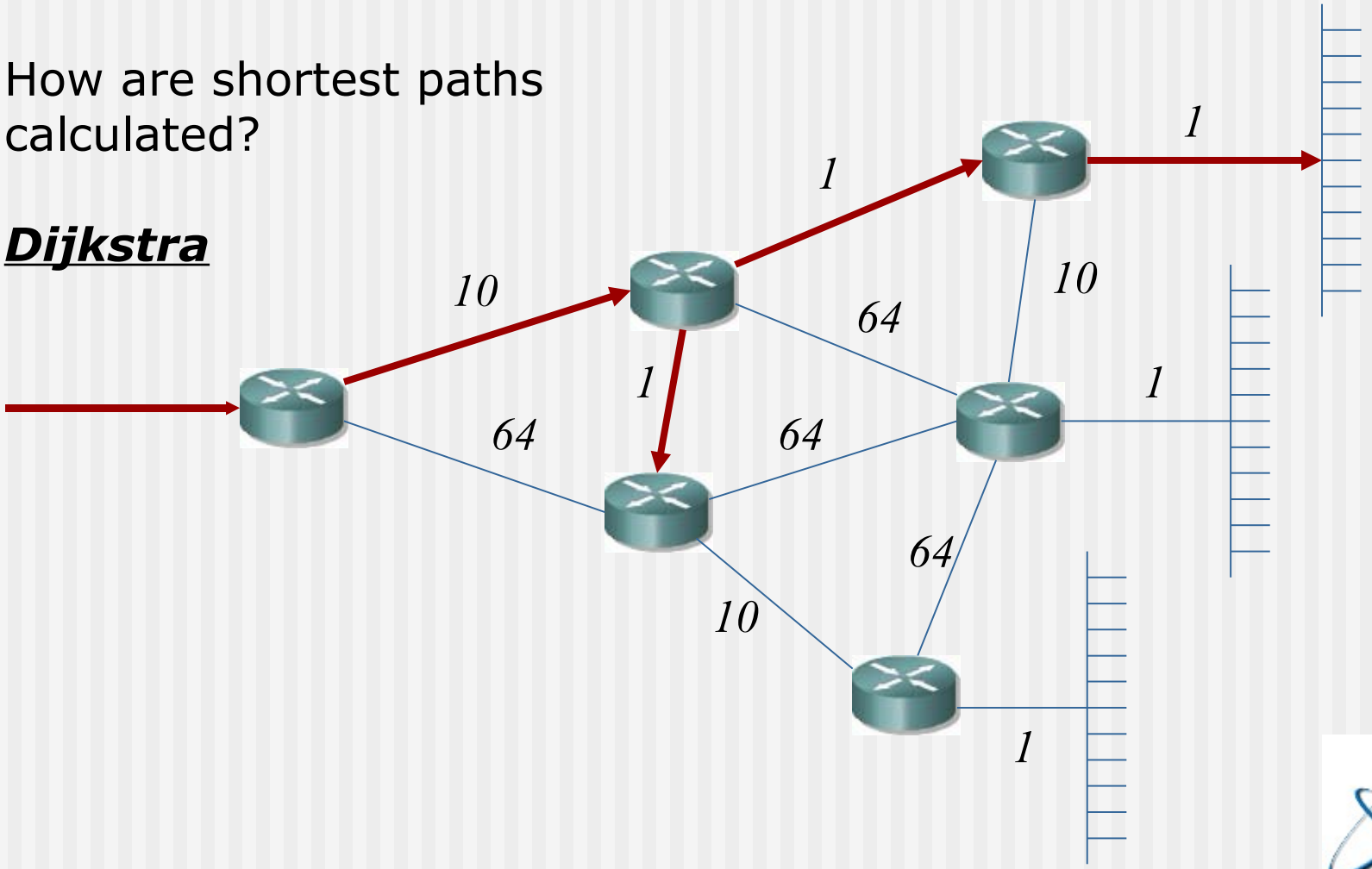
Dijkstra



A Closer look at OSPF

How are shortest paths calculated?

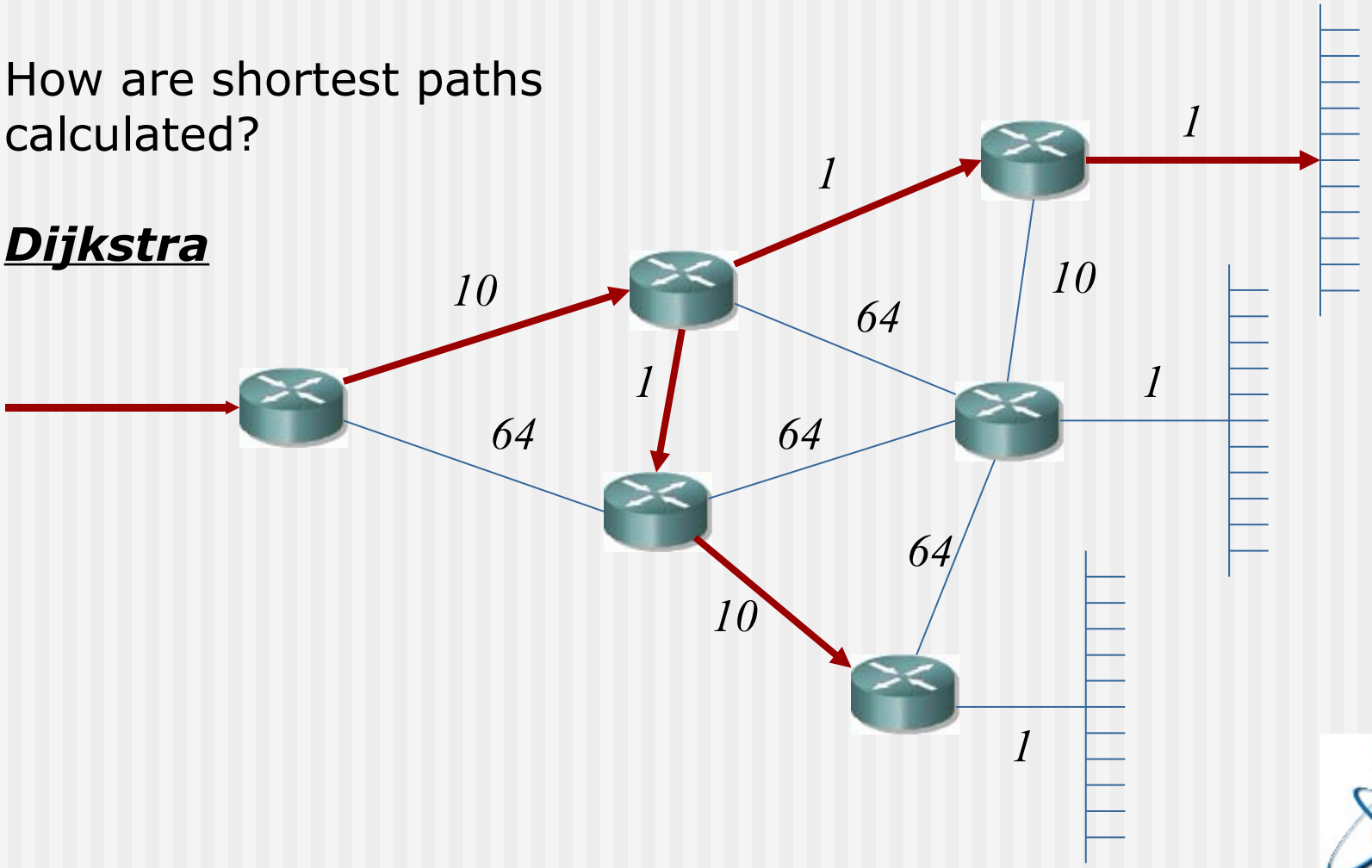
Dijkstra



A Closer look at OSPF

How are shortest paths calculated?

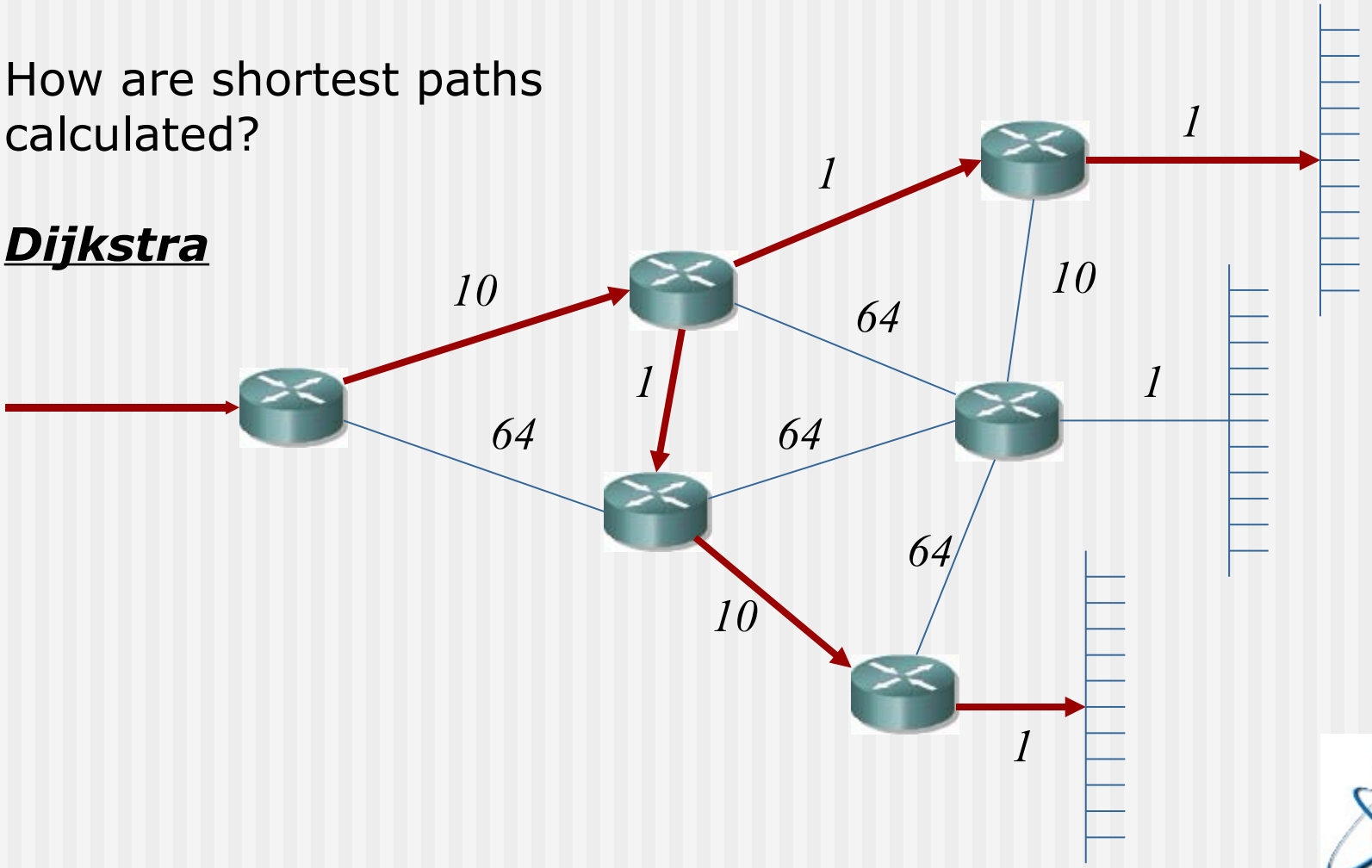
Dijkstra



A Closer look at OSPF

How are shortest paths calculated?

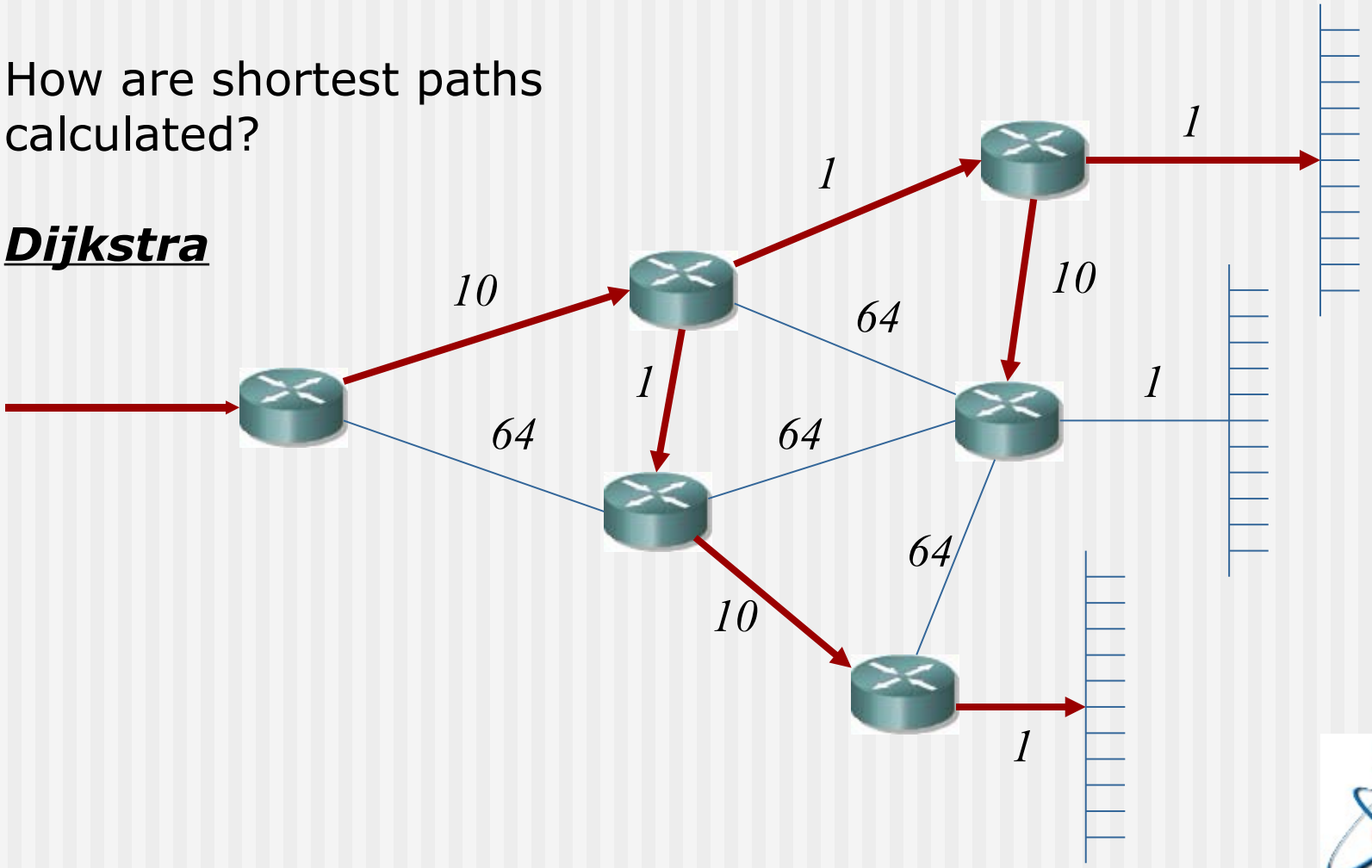
Dijkstra



A Closer look at OSPF

How are shortest paths calculated?

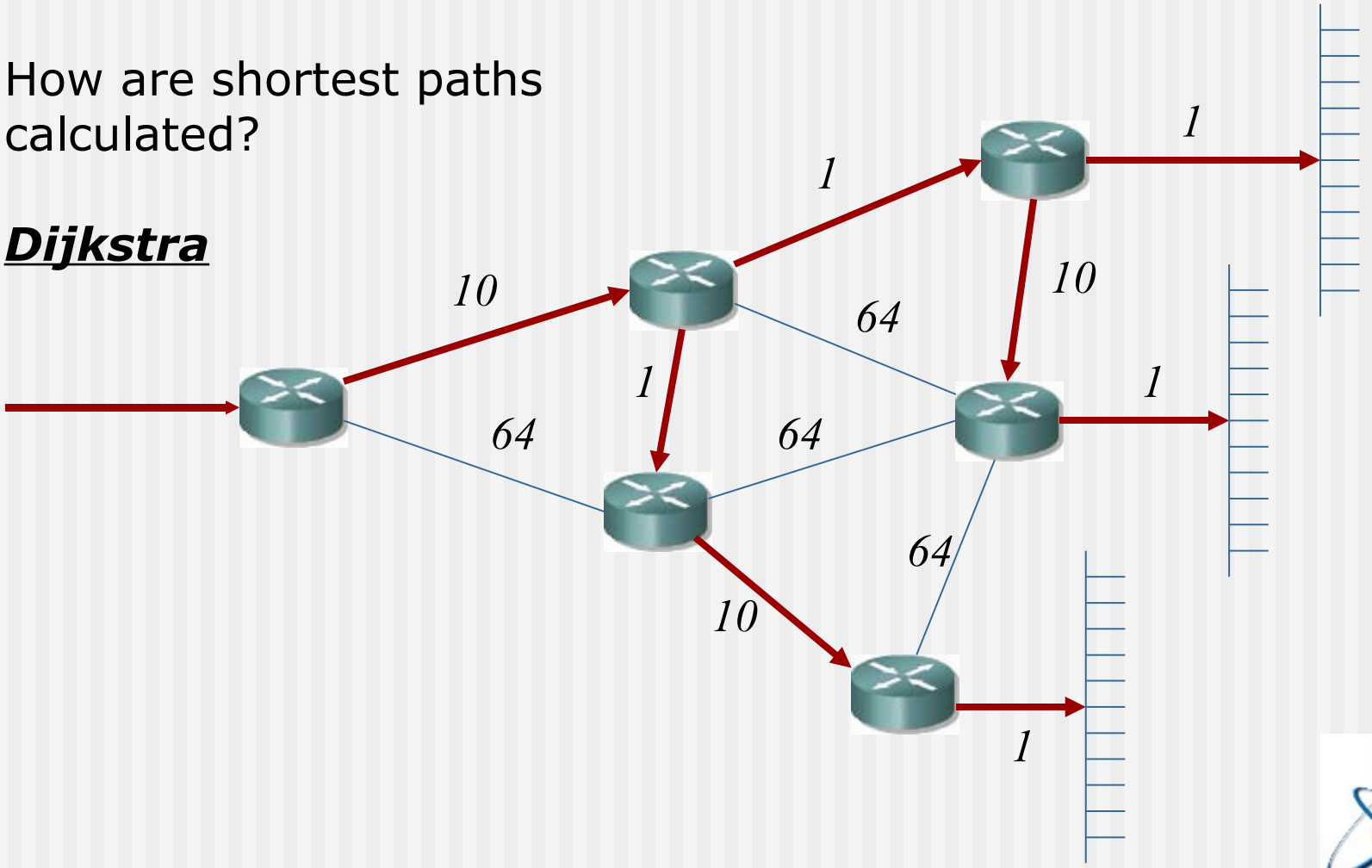
Dijkstra



A Closer look at OSPF

How are shortest paths calculated?

Dijkstra



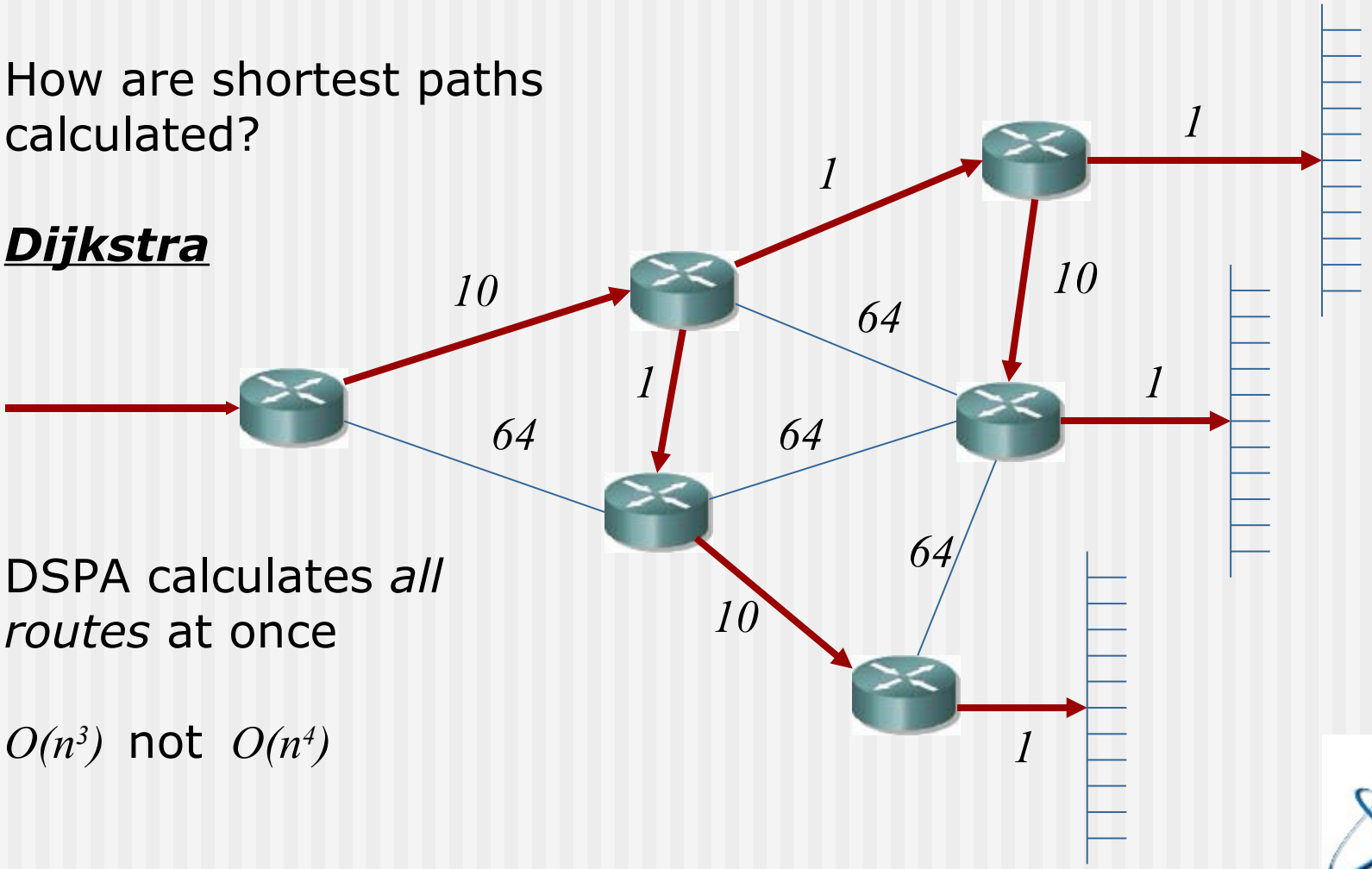
A Closer look at OSPF

How are shortest paths calculated?

Dijkstra

DSPA calculates *all routes* at once

$O(n^3)$ not $O(n^4)$



A Closer Look at EIGRP

- Cost function:

$$C = \left(k_1 b + \frac{k_2 b}{256 - l} + k_3 d \right) \left(\frac{k_5}{r - k_4} \right) \qquad \text{(if } k_5 > 0\text{)}$$

or

$$C = \left(k_1 b + \frac{k_2 b}{256 - l} + k_3 d \right) \qquad \text{(if } k_5 = 0\text{)}$$

  - b: bandwidth, d: delay
    l: load r: reliability … EIGRP 'metrics'

"Don't change k-values!!!!!"

A Closer Look at EIGRP

- Cost function:

$$C = \left(k_1 b + \frac{k_2 b}{256 - l} + k_3 d \right) \left(\frac{k_5}{r - k_4} \right) \qquad \text{(if } k_5 > 0\text{)}$$

or

$$C = \left(k_1 b + \frac{k_2 b}{256 - l} + k_3 d \right) \qquad \text{(if } k_5 = 0\text{)}$$

  - b: bandwidth, d: delay
    l: load r: reliability, k_1, k_2, k_3, k_4, k_5 configured by administrator (must match on neighbour routers)
    - default: $k_1=1, k_2=0, k_3=1, k_4=0, k_5=0$)
- Metrics and costs distributed by *DUAL* (*Diffusing Update Algorithm*) … but not truly *dynamic*!
- Very effective/efficient … but not perfect!

A Closer Look at EIGRP

- Cost function:

$$C = \left(k_1 b + \frac{k_2 b}{256 - l} + k_3 d \right) \left(\frac{k_5}{r - k_4} \right) \qquad \text{(if } k_5 > 0\text{)}$$

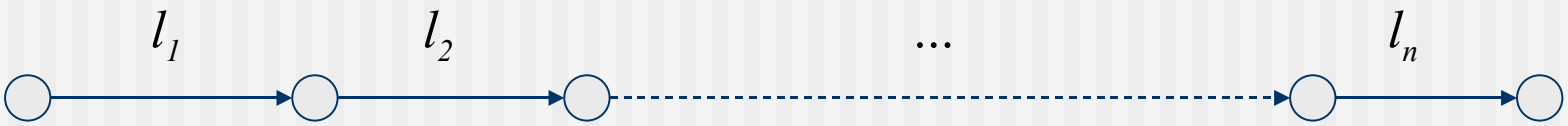
or

$$C = \left(k_1 b + \frac{k_2 b}{256 - l} + k_3 d \right) \qquad \text{(if } k_5 = 0\text{)}$$

  - b: bandwidth, d: delay
    l: load r: reliability … EIGRP 'metrics'

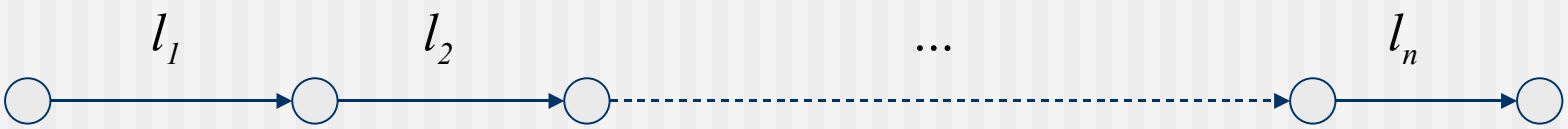
"Don't change k-values!!!!!"

Using EIGRP 'Metrics'



- How should metrics be combined across a path?
- 'End-to-End' path costs?
- Delay?
- … add?
- Bandwidth?
- … minimum?
- Load?
- … maximum?
- Reliability?
- … product?

Using EIGRP 'Metrics'



- Is this what really happens?
- Add Delays?
- … yes! (well, sort of!)
- Minimum Bandwidths?
- … yes!
- Maximum Loads?
- … who knows!!!
- Multiply Reliabilities?
- … who knows!!!

EIGRP Metrics and DUAL

$$C = \left(k_1 b + \frac{k_2 b}{256 - l} + k_3 d \right) \left(\frac{k_5}{r - k_4} \right)$$

- Bandwidth and delay appear linearly in the EIGRP cost function
- Load and reliability do not
- (In fact bandwidth becomes non-linear if load is introduced)
- DUAL checks to see if 'cost' + 'additional cost' > 'cost' for 'feasible successors'
- A linear relationship
- DUAL may not work with changed k-values!

EIGRP Metrics in Practice

$$C = \left(k_1 b + \frac{k_2 b}{256 - l} + k_3 d \right) \left(\frac{k_5}{r - k_4} \right)$$

- Where do the four EIGRP metric values come from?
- Bandwidth – measured from interface
  - As it should be!
- Delay – taken from lookup tables! (from Bandwidth)
  - $C = f(b,d) = f(d,g(b)) = h(b)$
- Load and Reliability – who knows??!!
  - Fixed? $C = f(b,d,l,r) = f(d,g(b),constant) = h(b)$
  - Why can't we change them??!!

A Closer Look at EIGRP

- Cost function:

$$C = \left(k_1 b + \frac{k_2 b}{256 - l} + k_3 d \right) \left(\frac{k_5}{r - k_4} \right) \qquad \text{(if } k_5 > 0\text{)}$$

or

$$C = \left(k_1 b + \frac{k_2 b}{256 - l} + k_3 d \right) \qquad \text{(if } k_5 = 0\text{)}$$

  - b: bandwidth, d: delay
    l: load r: reliability, k_1, k_2, k_3, k_4, k_5 configured by administrator (must match on neighbour routers)
    - default: $k_1=1, k_2=0, k_3=1, k_4=0, k_5=0$)
- Metrics and costs distributed by *DUAL* (*Diffusing Update Algorithm*) … but not truly *dynamic*!
- Very effective/efficient … but not perfect!

The ***BIG*** Problem!

- Routing is an optimisation problem
- Optimisation generally involves trying to maximise or minimise something
- What are we trying to maximise/minimise here?
- … dropped/timed-out packets
- … overall throughput?
- … average/maximum delay?
- … customer satisfaction?
- … company profits?

$$P = f(b) \qquad ?!?!$$

- ***<u>There is no single solution to all these at once!</u>***

Conclusions

- Yes, OSPF and EIGRP are fairly sophisticated routing protocols
- But they're a long way off perfect!
- (What should an *ideal* routing protocol be looking to achieve?)
- Both OSPF and EIGRP have some failings and uncertainties
- Can we ever build a *perfect* routing protocol?
  - Will we ever understand the problem well enough?
  - Will we have enough processing power to run it?
  - Are IS-IS or 'Ant routing' the answer?
- The answer to all these questions is …
- … probably not!

Thank you

Any questions?

Vic Grout

Centre for Applied Internet Research (**CAIR**)
University of Wales
NEWI Plas Coch Campus, Mold Road
Wrexham, LL11 2AW, UK

v.grout@newi.ac.uk

www.cair-uk.org