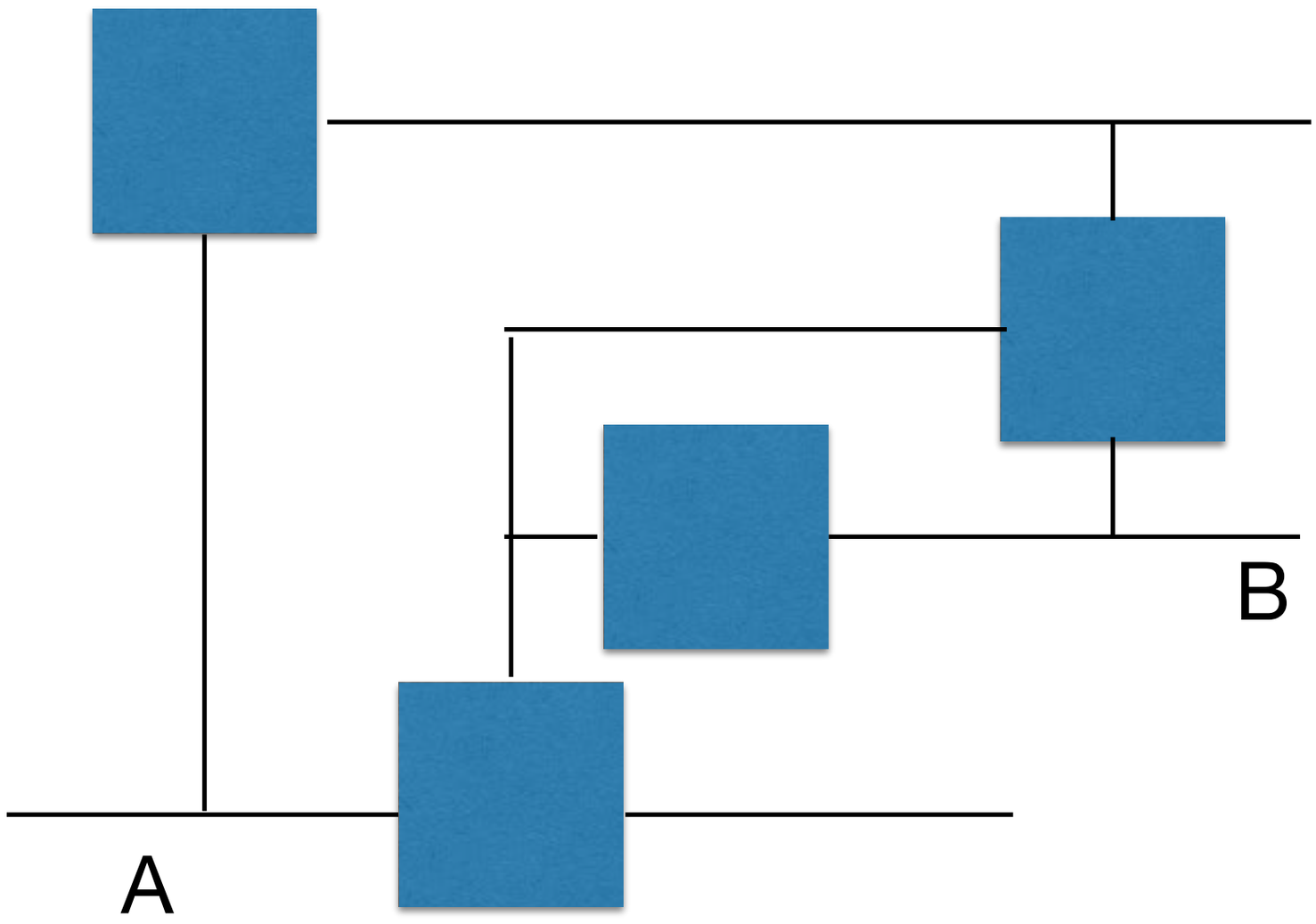
Networks 19: Routing Protocols

i.g.batten@bham.ac.uk

The problem



IP Routing

- Network doesn’t do it for us: we are responsible for knowing the best “next hop”

- We can’t (in 2017) dictate the route beyond the next hop

- There is no global (or local) protocol for co-ordinating routing. Loops can arise.

IP Resilience

- There are various ways to spare routers
  - VRRP, HSRP, proprietary N+1 inside or outside switches
- There are various ways to spare links at layer two
  - LCP and Link Agg, “Spanning Tree”, proprietary or media-specific buddying systems
- But you really want complete resilience in the event of major failures

Static Routing

- In simple network, you set routing tables up by hand
- “This network goes to this router, this network goes to this router, everything else goes to the Internet via this router”.
- No resilience, but unlikely to go wrong so long as all the links stay up

Interior v Exterior Routing

- Different mechanisms used for routing inside the enterprise and outside the enterprise.

- “Interior” protocols are used when all the equipment is to some greater or lesser extent under one management

- “Exterior” protocols are used between enterprises

Interior

- Far fewer networks (10s is a lot, 100s is rare, 1000s very unusual)
- More trust between equipment and effective sanctions available if it goes wrong
- Potentially a central management authority or system
- Reasonable to maintain a full set of routing tables

Exterior

- In the limit, needs to be able to handle the entire Internet (2^24 routes and in the future up to 2^64)
- Partial routing tables a reasonable response
- No central authority and no effective sanctions, so needs to be robust and secure (FSVO “secure”).

Objectives

- In simple networks, there is only one sensible path between points and the task is to find it
    - Spanning tree which doesn't change (and there's a unique solution)
- In more complex networks, there are multiple paths between points and the task is to find the best
    - Minimal spanning tree which changes rarely (possibly need to choose between alternatives)
- In the largest networks, there are multiple paths between points and a high rate of change, so the task is to find the best **now**.
    - Minimal spanning tree with requirements for stability (lots of options, lots of change)

Metrics

- Minimal spanning trees require costs associated with edges
- Those costs normally expressed as simple integers
- Problem in networks is that we might be interested in bandwidth, latency, reliability, security, cost…
  - OS routing tables don’t deal with workload-specific routing, and protocols which purport to handle this usually reduce to a single integer by magic formulae.
  - Interesting research topic (“policy based routing”)

Distance Vector

- Simplest and earliest algorithm
- Doesn’t require computation of a spanning tree by any one party
- Quick and easy to implement
- Gives quick and dirty results

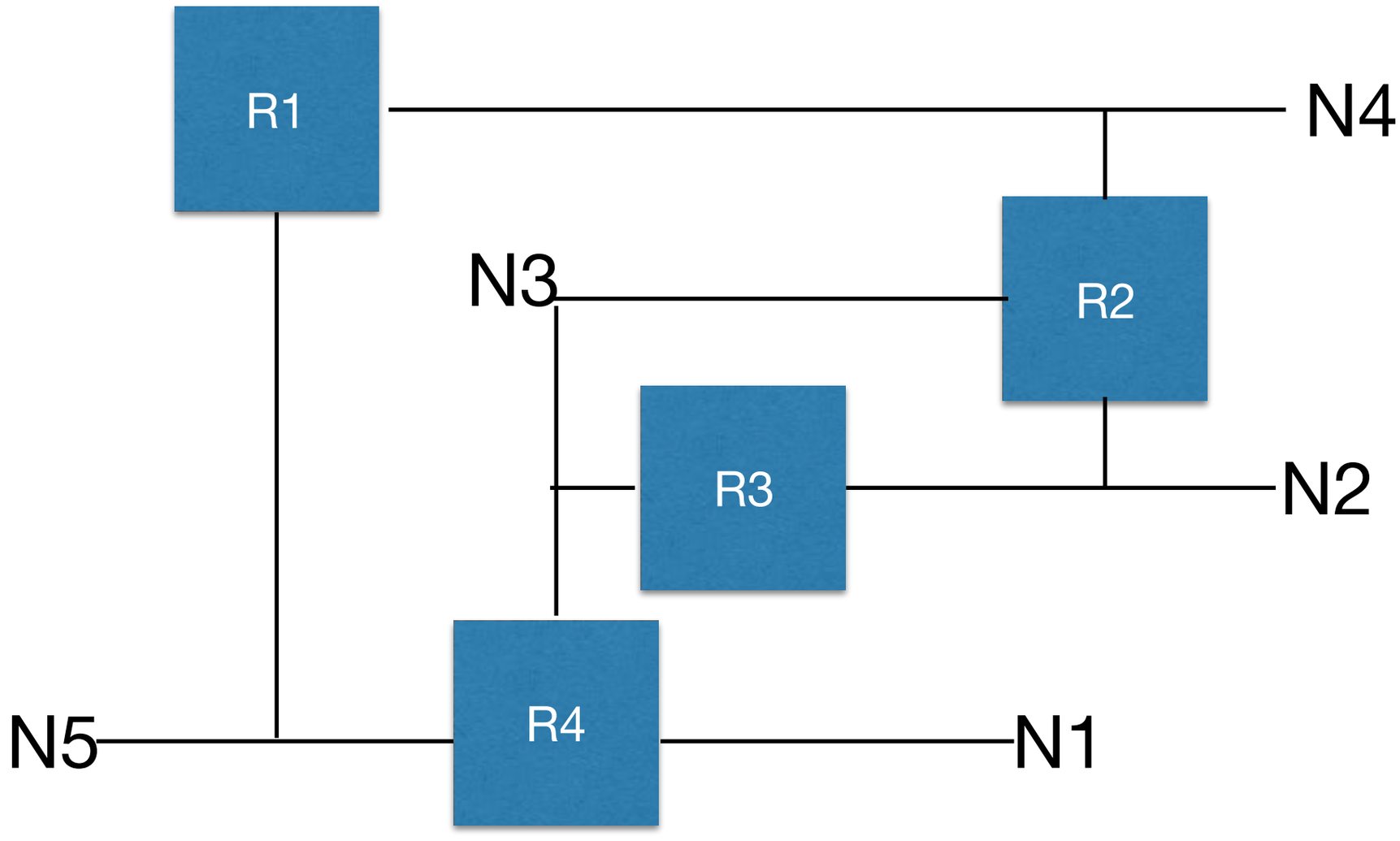
RIP

- Each connected network is one “hop” away
- Each node broadcasts all the networks it knows how to reach, with their “hop count”
- When you receive a routing update, you add one to the included “hop counts” and add it to the routing database.
- Best options are used for kernel routing tables

RIP

- Metrics are only 0 to 16, 16 = unreachable
- Sets maximum diameter on network
- Limits ability to use metrics >1 to indicate slow or unreliable links

The problem, labelled



As a matrix

	N1	N2	N3	N4	N5
R1				X	X
R2		X	X	X	
R3		X	X		
R4	X		X		X

Consider Routers 1 and 2

- R1 broadcasts “I can reach N4 and N5, directly connected”
- R2 broadcasts “I can reach N2, N3, N4, directly connected”
- R2 learns a route to N5 via R1 over their shared N4
- R1 learns a route to N2 and N3 via R2 over their shared N4.
- R1 now broadcasts “I can reach N4 and N5, connected, and N2 and N3, 1 hop away”.
- R2 now broadcasts “I can reach N2, N3, N4, connected, and N5, 1 hop away”.

Exercise

- Spend five minutes working out the routing tables everyone gets

- Assume the only metric in use is hop count

Computed Routes

	N1	N2	N3	N4	N5
R1	R4	R2	R4/R2	X	X
R2	R3	X	X	X	R1
R3	R4	X	X	R2*	R4
R4	X	R3/R2	X	R1	X

Problems with Distance Vector

- Routing loop
  - Router1 believes it has a route to Network1 via Router4.
  - Router4’s interface to Network1 goes down
  - Router4 will believe an update from Router1 saying “I know how to get to Network 1, two hops away”
  - Takes some time to damp this problem down

Problems with routing loops

- Packets are sent forwards and backwards until the packet TTL hits zero

- Initial TTL in packets has been increasing in recent years as diameter of Internet increases: could be as much as 60

- In process generates lots of ICMP messages, triggers IDSes, etc, etc, etc, as well as burning bandwidth.

R4 i/f N1 breaks at t=10

Announcement for Network 1	Router 1	Router4
t=0	1 Hop	Directly Connected
t=30	3 Hops	2 Hops
t=60	5 Hops	4 Hops
t=90	7 Hops	6 Hops
t=240	Removed	Removed

Solution 1: Flash Update

- Standard RIP sends a packet every 30s
- Therefore can take four minutes for routes to converge after a topology change
- Instead, send a packet on every update which makes changes, and send a packet on every change of local interface status
- Same progression, but much quicker

Solution 2: Split Horizon (now mandatory)

- Instead of making same announcement out of each interface, only announce from each interface routes reachable from other networks

  - Don’t announce routes which involve turning packets around and sending them back out of the interface they arrived over

- Prevents simple routing loops like this, but more complex analogues exist

- Requires more book-keeping and more computation (but not very much)

R4 interface N1 breaks at t=10: split horizon

Announcement for Network 1	Router 1	Router4
t=0	1 Hop	Connected
t=30	1 Hop	Nothing
t=60	1 Hop	Nothing
t=180 (or 210)	Nothing	Nothing

Solution 3: Poison Reverse

- When a link fails, immediately send an update with metric 16

- Then proceed as normal

R4 interface N1 breaks at t=10, Poison Reverse

Announcement for Network 1	Router 1	Router4
t=0	1 Hop	Directly Connected
t=30	Broken	16 Hops (Broken)
t=60		
t=90		
t=240		

RIP with mod cons

- RIP with split horizon, poison reverse and flash updates works tolerably well on small networks
- Still limited by small limit on diameter and crude metrics
- Topology changes spread quite slowly
- Modern implementations are more complex (“hold down timers”, for example)
- Attempts to change protocol timers lead to wild instability unless implemented network-wide at the same time

RIPv2

- RIPv1 is from when dinosaurs walked the earth, and therefore is “classful”
  - Networks starting 0 are /8, neworks starting 10 are /16, networks starting 110 are /24
  - Impossible to use in modern networks with subnetting
- RIPv2 incorporates subnet masks, and will run over multicast rather than broadcast
- Not widely adopted

RIP Security

- RIPv2 introduces MD5-based authentication, in the same style as SNMPv3
  - Insert string, hash packet, put hash where the string was.
  - Sequence numbers used to prevent replay attacks
- Used by…almost no-one.

RIPng

- RIPv2 modified to handle IPv6
- Networks simple enough to work with RIPng are simple enough to not need RIPng, I suspect
    - Much more layer 2 switching means fewer layer 3 networks.

Link State Protocols

- Instead of sending out reachability information, devices in an *area* exchange information about all links that are active.

- Every device on network can therefore calculate a minimal spanning tree

- Link information is flooded throughout the area, so convergence is quick: devices switch from one consistent set of routing tables to another

OSPF

- Open Shortest Path First
- Link State Protocol
- Usable on very large networks, and has additional features to help with this
- Supports authentication (same method, and caveat, as RIPv2)

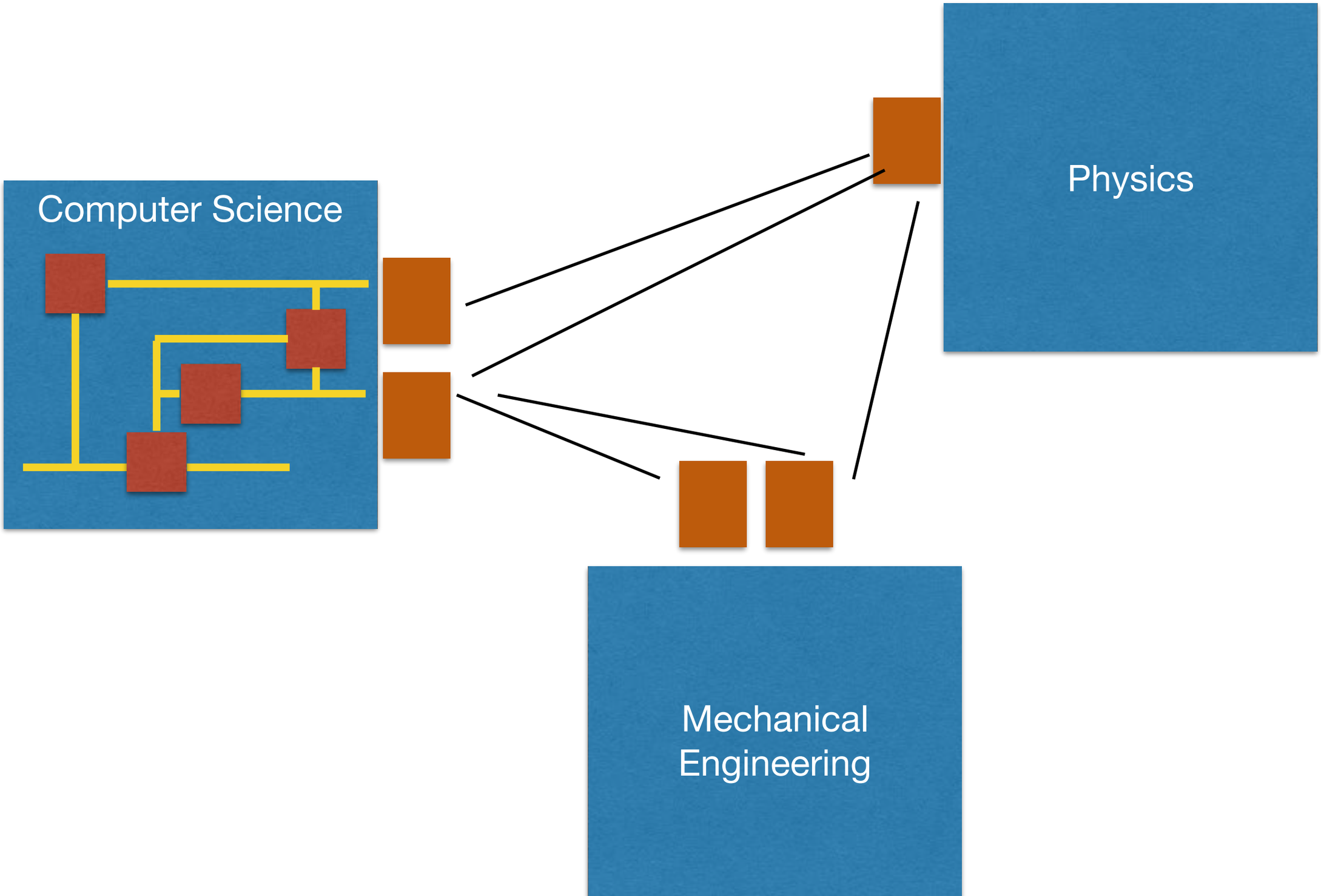
OSPF Basic Operation

- Devices on a subnet exchange HELLO packets to learn about their local neighbours
- They elect a “Designated Router” and a “Backup Designated Router” from the devices that have multiple interfaces (based on preferences pre-configured into the routers, then router ID as a tie-break)
- The DR and BDR exchange link state advertisments (LSA) with neighbouring routers
- When the LSA information changes, the routing algorithm is used to recompute a set of routing tables
- The DR and BDR announce a complete set of non-local routes to other systems on the local network

OSPF Areas

- Most networks have a small group of highly connected core routers, with networks connected to one of the core routers
    - For example, core router per department, meshed, with departmental networks connected to departmental router
    - Pointless to send updates about changes inside a department campus wide, as always reached via departmental network
- OSPF can divide networks into Areas, with “out of area” routes summarised rather than re-calculated by everyone

OSPF Areas



OSPF Areas

- So Computer Science's routers will advertise all the internal networks as part of the area they provide access, and other routers will compute routes to an area that includes all those networks as a group.

- Reduces opportunities for fine-grained decisions about which core link is "closer" to individual networks in an area

- But as we have an n*log*n problem computing the minimal cost tree, reducing n is always good

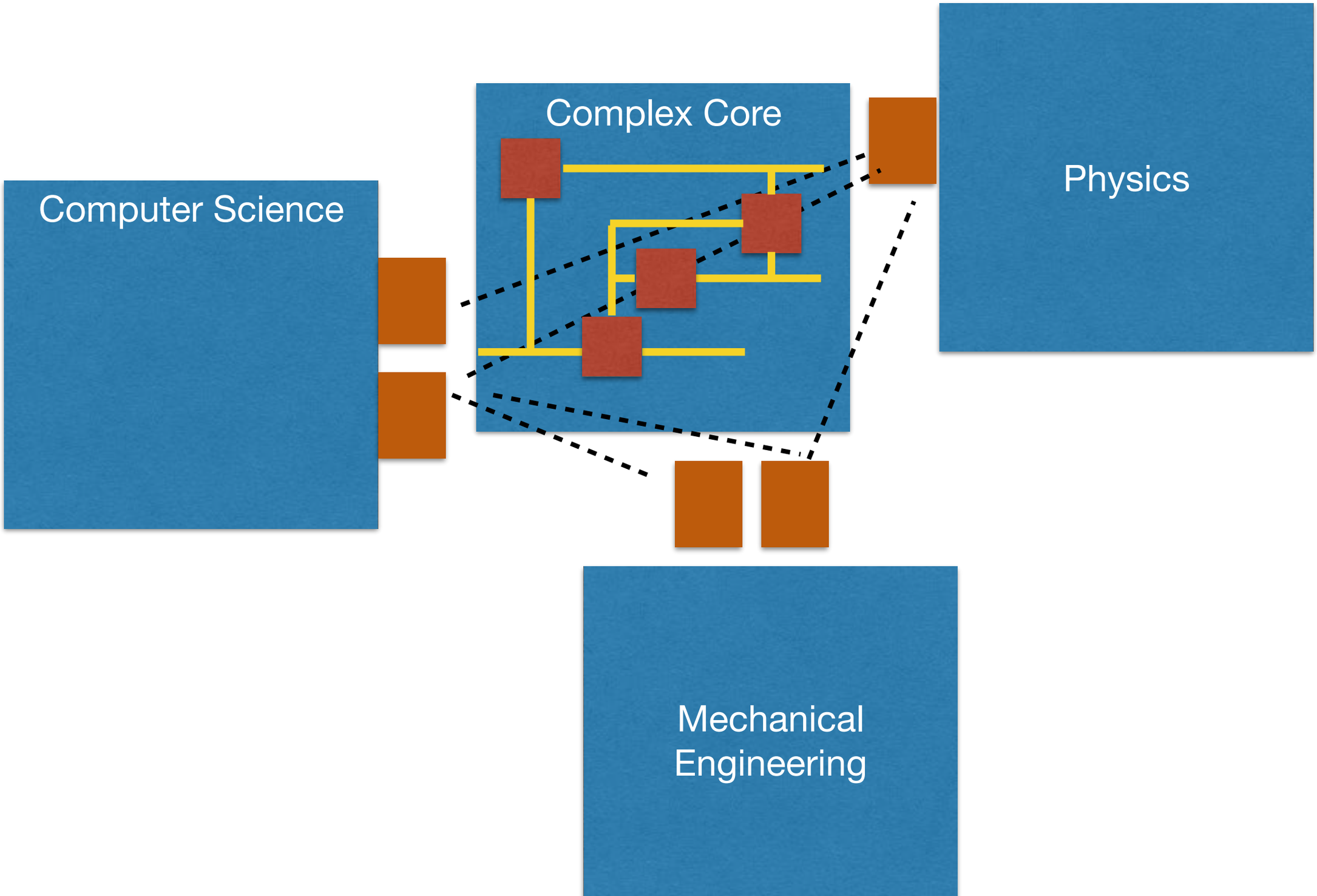
OSPF Areas

- OSPF doesn’t really support a hierarchy of areas: the intention is that there is a “backbone” network which is simply connected, and each area connects to a backbone network.

- In more complex cases, the backbone can rely on transit through a more complex area with “virtual links”

  - Core routers know about each other, and pretend they are directly linked

OSPF Virtual Links



Advantages of OSPF

- Converges within a few seconds of topology change, as DR and BDR routers exchange LSAs on demand
  - Routers can pass on LSA information before they have performed the recomputation: DV protocols pass on “cooked” information, while LS protocols pass on “raw” information
- Routing loops are rare or short-lived, because everyone is working from the same information

Problems with OSPF

- (Historically) CPU intensive, with solutions ("Inter-area announcements") quite difficult to get right.
- Complex to configure correctly
- Shortage of good open-source implementations ("gated" had complex licensing, "zebra" and "quagga" niche and difficult to use).
- Proprietary alternatives sometimes preferred

Alternatives

- IGRP (classful) and EIGRP (classless) from Cisco
- Proprietary, but widely reverse-engineered; standard released in 2013
- Distance Vector protocol with more sophisticated metrics and additional information
    - Does incremental updates rather than sending the whole routing table each time
- Usable for large, complex networks, but tricky if it is heterogenous.

The real world

- OSPF is required for large, complex networks
  - Rate of change, even manual change, too high for static routing
- In un-spared networks, static routing probably better than RIP
- OSPF Areas allow statically routed departmental networks with dynamic core routing

Load Balancing

- Implementations can load balance over links with the same metric (RIP) or with metrics that are within some bound of each other (OSPF, EIGRP).

- Layer-3 load balancing can play very badly with firewalls and NAT (“asymmetric routing”)

- Probably better ways to do it today.

Summary

- RIP is easy to understand, easy to use and usually the wrong answer

- OSPF is complex to understand, very complex to use correctly and very effective

- Static routing isn’t cool, but is worth sticking with as long as you can.

External Routing

- Autonomous Systems
  - Large networks on the Internet are called “autonomous systems”
  - They have an AS Number to identify them
  - Every routable network is a member of exactly one AS

Autonomous System

- Typical AS is “an ISP and all of its small and medium customers”
- But what happens when you want to have multiple ISPs, for reasons of resilience, negotiating power, geography, etc?
- Answer: get an AS Number and some provider-independent IP address space (not easy these days)
    - Only went from two-byte to four-byte ASNs in 2010

AS a bit like OSPF Area

- Assumption that everyone inside the AS knows how to reach everyone else inside the AS, and that there is a small number of entry/exit points to the AS from which those networks can be reached.

- In fact, BGP sometimes used an an Interior Routing Protocol by very large organisations for whom OSPF doesn’t scale

BGP

- Routers peer over **TCP**
    - Makes dealing with message loss and node failure easier
- Updates are incremental
    - Routing tables changed on each update, rather than all at once
- Uses slightly different approach to RIP or OSPF

Path Vectors

- Conceptually, BGP is a distance vector algorithm: the route to a network reached by another router has a cost equal to the cost that router advertises, plus the cost of getting to the router.

- BGP has extensive support for policy-based decisions, load spreading and so on

AS Path

- But each route also carries with it a vector containing an ordered list of all the ASes that the route passed through

- So if AS45 peers with AS90, and AS90 advertises that route to its peer AS135, which advertises it to AS200, the AS Path (45, 90, 135, 200) will be known to AS200.

- Allows easy detection of routing loops

Pro Tip

- I once asked the architect of the UK's first ISP, who went on to found the London Internet Exchange (LINX), whether I should get an ASN and start talking to two ISPs

- "You'll have more trouble from that than your single ISP will cause you: get redundant links to a single ISP and talk OSPF to them".

- Ten years later, as I was leaving, we got an ASN and two ISPs to satisfy a customer requirement and started talking BGP

- From what I hear, Keith was absolutely right…