TRILL

is really cool...

Andy Davidson
UKNOF, Edinburgh

NetSumo / LONAP
7th September 2010

Agenda

- What is the problem at Layer 2?
- How does TRILL work to solve this?
- Use Cases
- Other bits/reference material

Agenda

- What is the problem at Layer 2?
- How does TRILL work to solve this?
- Use Cases
- Other bits/reference material

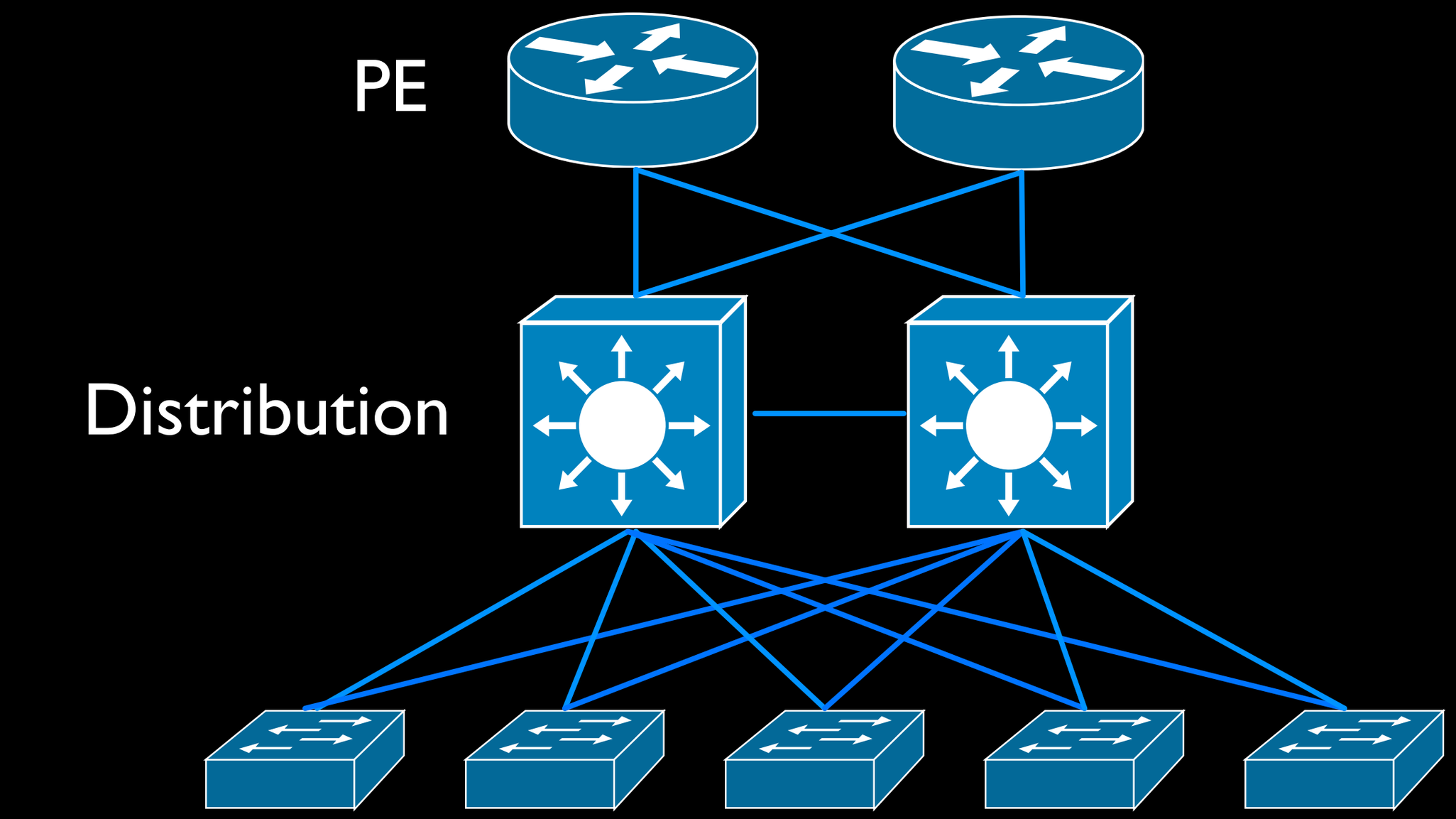
I think that I shall never see
a graph more lovely than a tree.

A tree whose crucial property
is **loop-free** connectivity.

Algorhyme, Radia Perlman

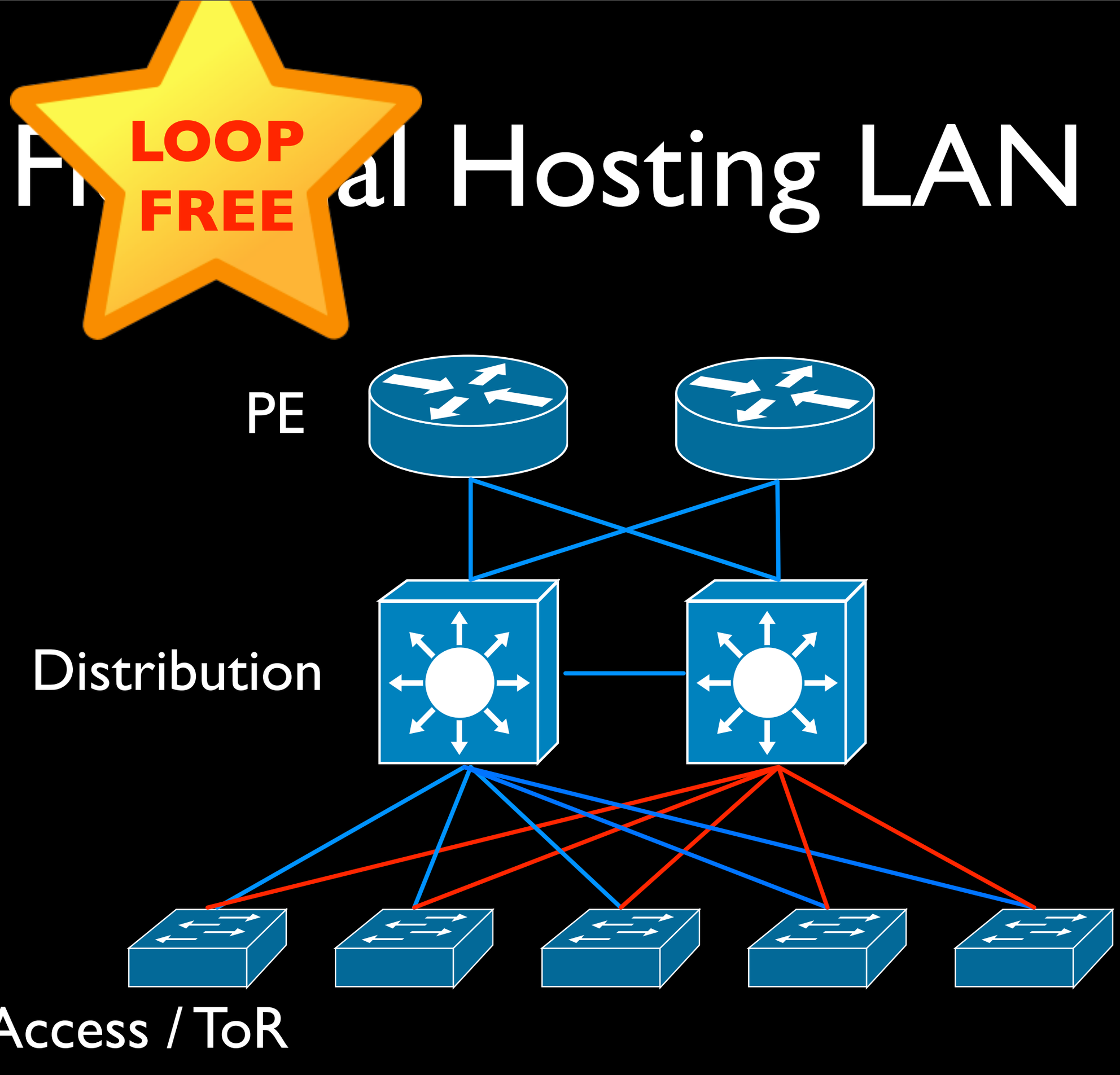
Ethernet segments MUST be loop free in order to function. Severe disruption is guaranteed if they are not. However, many networks are constructed that rely on looped topologies in order to provide diversity or reliability. In such circumstances, protocols are overlayed on a switched network, that cause some of the links to block, to prevent the loops. Spanning tree is just one example of such a protocol, but all rely on blocking certain links to prevent loops.

Fictional Hosting LAN



Here is a typical hosting segment. The provider wants to give some diversity to their hosting customers, so connects their access switches to diverse distribution switches. This allows one distribution switch to fail, or permits one to be taken down deliberately to be maintained, and service to continue. If cabling between the distribution switches and racks is diverse, it allows one path to break also. However, an Ethernet loop has been created, so there must be a protocol configured that blocks some of the links.

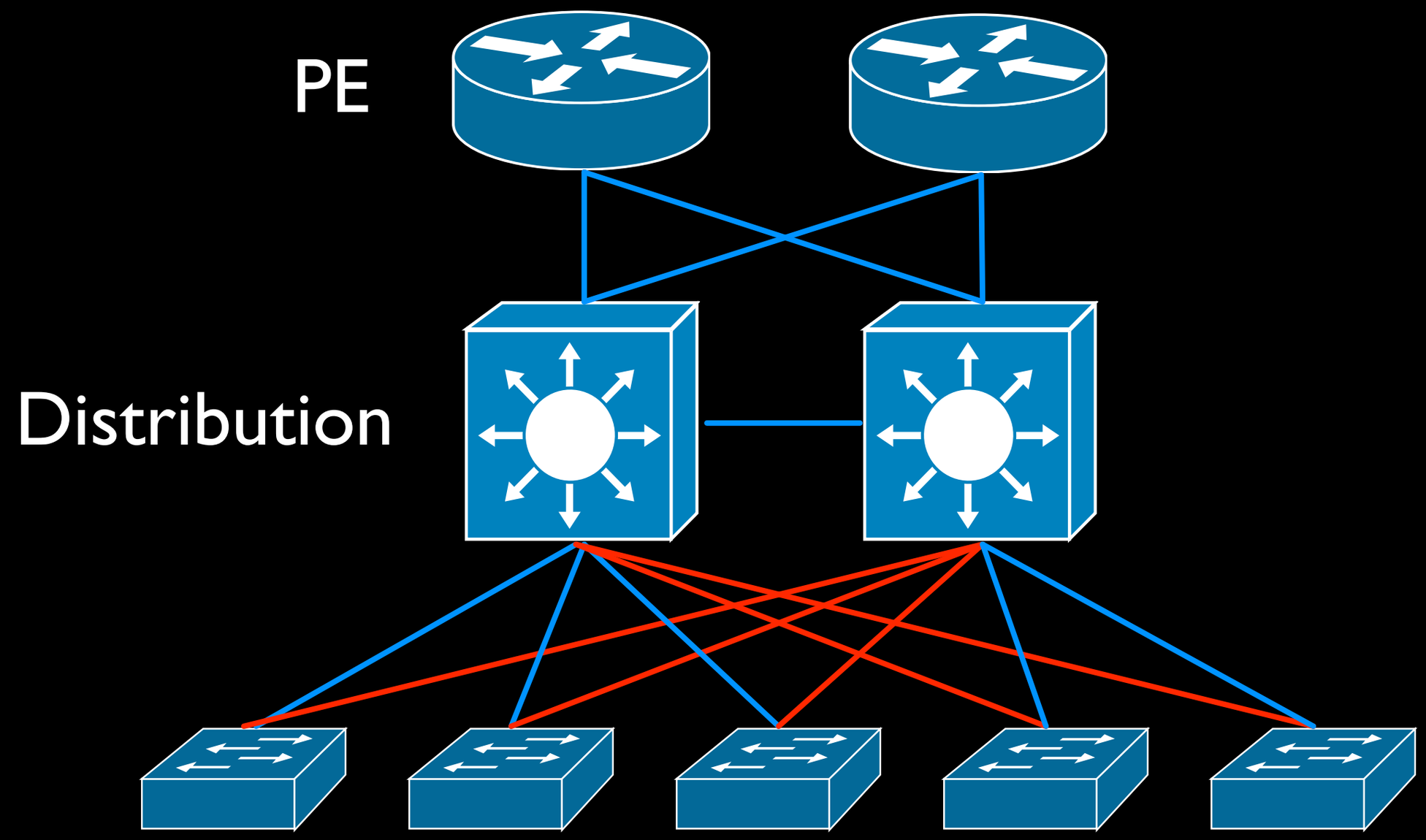
Fi al Hosting LAN



The red links represent blocked links.

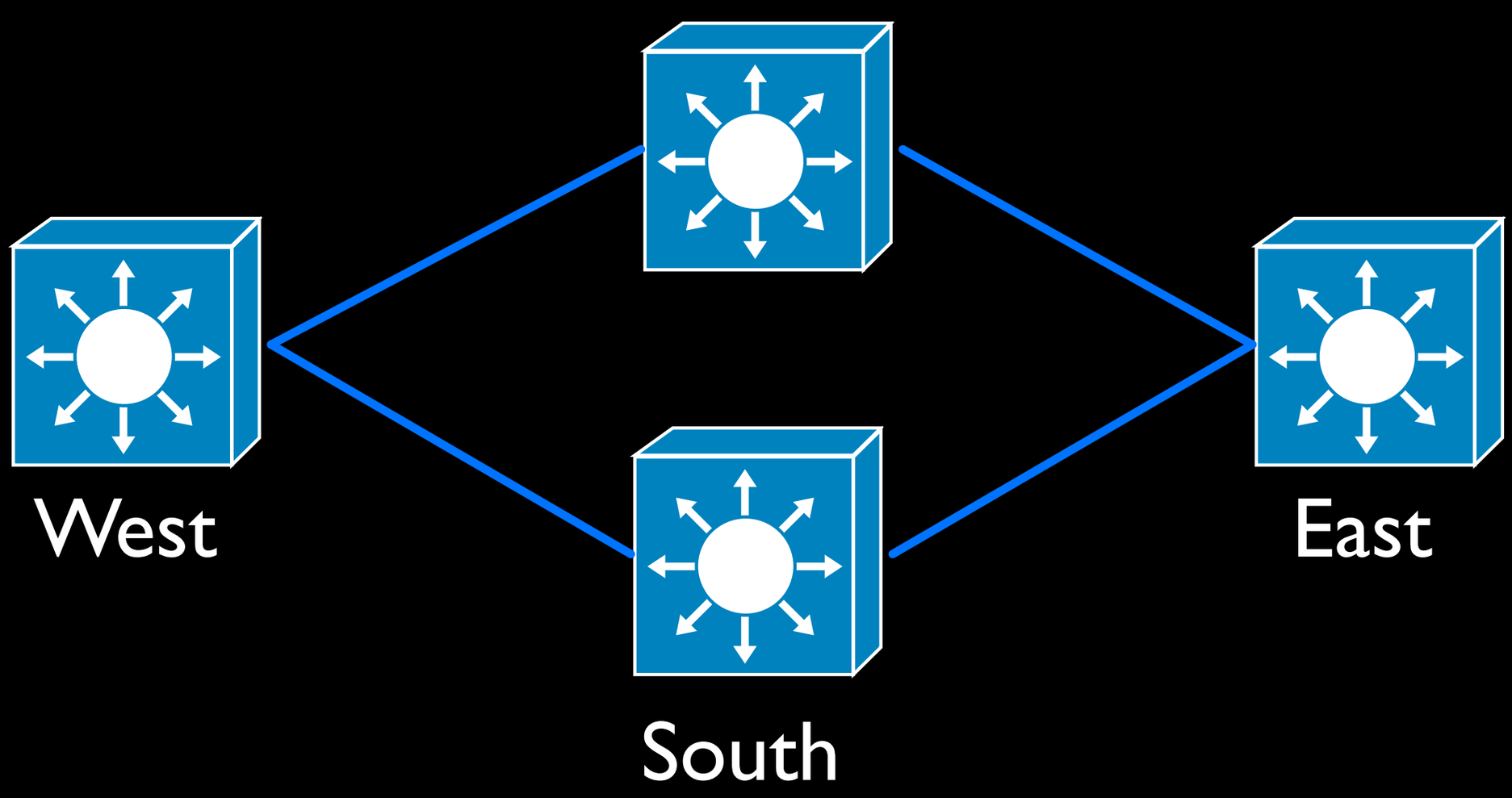
Here the network has deployed spanning tree to prevent an ethernet loop. However, this is not a beautiful arrangement, because the distribution switch on the right normally provides very little return on investment. It is also very hard to scale this network, either the owners must build several islands which look like this, or more powerful distribution switches, when the traffic volume increases.

Fi al Hosting LAN



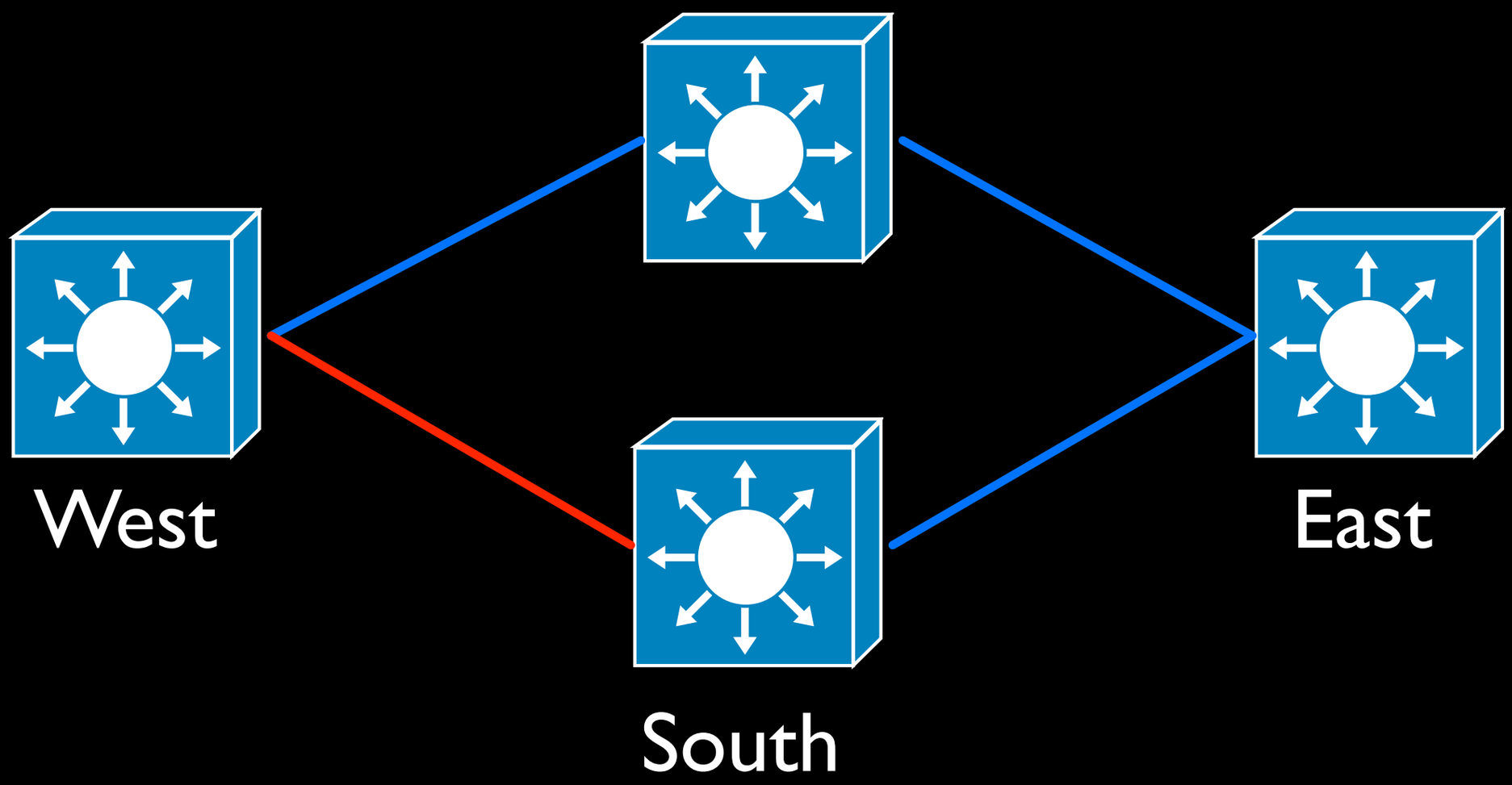
This scenario is almost no better at all, because even though the distribution switch on the right has got some work to do now, both distribution switches must be able to cope with all of the traffic on the network in the event of failure, so the theoretical maximum amount of traffic that this network can cope with is the peak throughput of one distribution switch. If this amount is exceeded, then this network is doubling, rather than halving, the chance of failure.

Fictional Metro Ethernet Service



Here, an ISP provides Metro Ethernet services between four points of presence in a city. A loop is created so that the (frequent) inter-POP link failures are mitigated, and to limit the effects of switch maintenance on customers.

Fictional Metro Ethernet Service



The loop has to be mitigated, so Spanning Tree disables the loop between the South and West POPs. This means that traffic between the South and West POPs must traverse the entire ring, even though they are directly connected. In these terms, the interlink between these two POPs is not providing good value for money.

If a significant, but temporary volume of traffic was to be delivered between the West and East POPs, the extra capacity via the South POP could not deliver any of this traffic, as the link is blocked.

Also, this provider is disincentivised from connecting the North/South switches, as it would by definition require another link becoming blocked.

(In reality, multiple paths may exist for different VLANs, but these have to be configured manually, so scales badly, and could also hurt debugging methodology).

Summary of Layer 2 Woe

- Spanning tree can lead to:
  - cost implication from blocked links / idle infra
  - inefficient paths
  - no multipath support
  - confusion, with many STP modes
  - complex debugging - no deterministic failure mode
  - Slow re-convergence after failover - partitioning

As well as there being many spanning tree modes (mst, pvst, rst), there are many vendor specific ‘standards’ - mrp/eaps, etc.
The choice of a layer 2 loop prevention protocol can limit topology options or feature availability, e.g. 802.1ad, q-in-q.
A quote from RFC5556 “There are a number of features of a modern layer 3 routing protocol which would be beneficial if available at layer 2”

Agenda

- What is the problem at Layer 2?
- How does TRILL work to solve this?
- Use Cases
- Other bits/reference material

A network where RBridges can
Route packets to their target LAN.
The paths they find, to our elation,
Are least cost paths to destination!
With packet hop counts we now see,
The network **need not be loop-free**!

draft-ietf-trill-rbridge-protocol-16

Ray Perlner, Algorhyme v2

Definitions

- TRILL is the TRansparent Interconnection of Lots of Links

- RBridges are devices (normally switches) that implement the TRILL protocol

We'll use these phrases a lot in the last half of this presentation, so it's important to agree definitions.

TRILL overview

- Link State protocols run between RBridges
- All Rbridges know all other Rbridges
- Optimal paths converged for unicast destinations
- Loopless distribution trees for unknown destinations
- Multipath delivery is supported
- Transit only RBridges do not learn end station MACs
- Transparency is assured - to end nodes and intermediate switches that do not implement TRILL

I'll demonstrate these in a moment.

RBridges must:

- Participate in TRILL-IS-IS to build the topology overview
- Encapsulate frames with TRILL headers, when they should deliver through another RBridge
- Decapsulate frames when they arrive for a locally connected end node

We're going to look at these roles in detail in a moment

Link State protocol in TRILL

- TRILL-IS-IS not compatible with IS-IS, its Layer3 brother.
- Role is to elect a Designated RBridge for each **link** on the network.
- Max MTU of control frames = 1470
  (any size is permitted on the LAN segment)

TRILL-IS-IS control frames are delivered to all RBridges, as they are addressed to the Multicast address All-IS-IS-RBridges. Switches in the ethernet cloud which do not support Trill simply forward the packets. End nodes drop the packets.

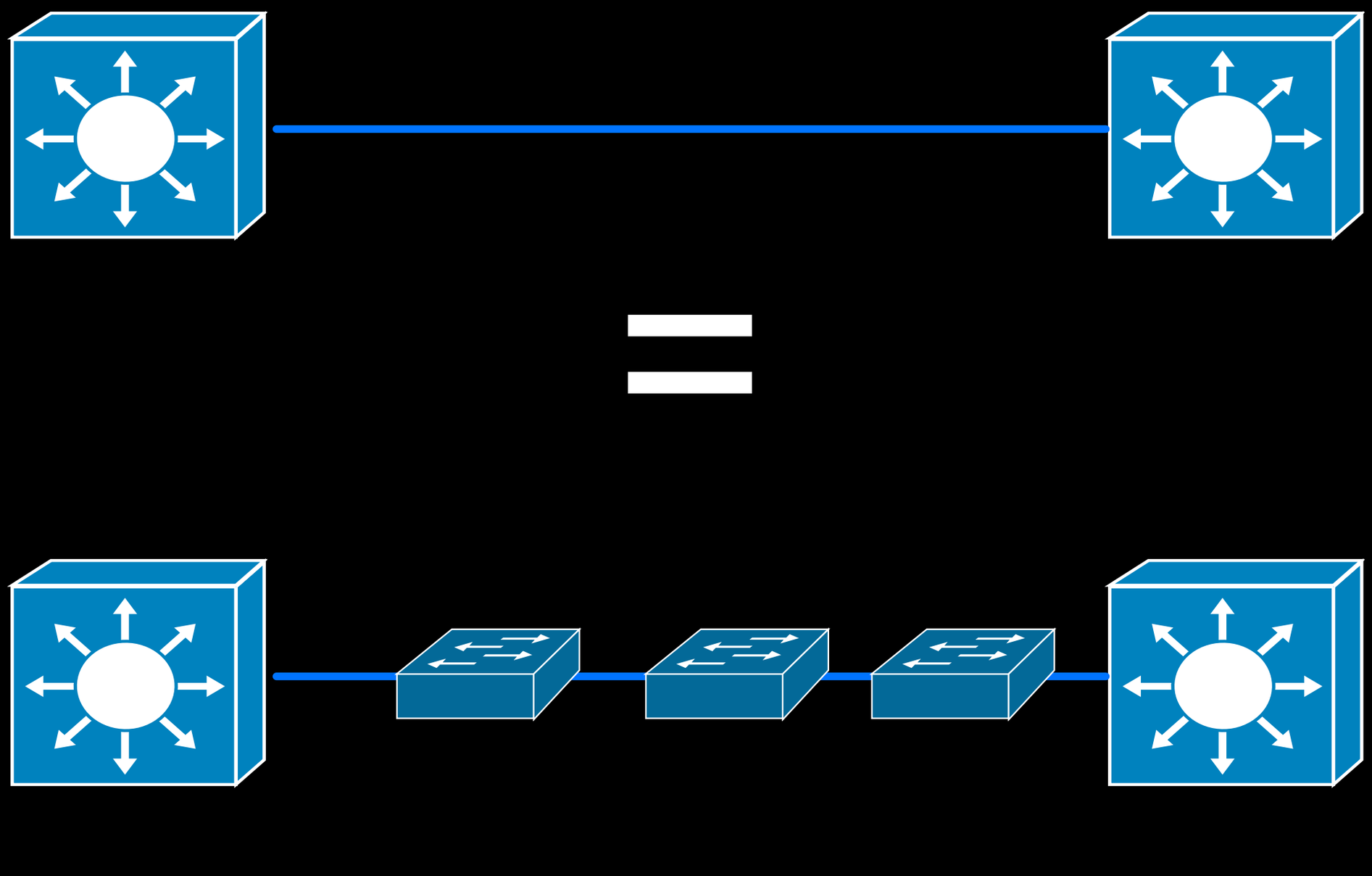
Although TRILL-IS-IS is not compatible with IS-IS, it borrows a lot of positive features (zero configuration, no ip address configuration, fast convergence).

A lower than 1500 MTU is used to try to limit the danger of having TRILL-IS-IS frames dropped by a misconfigured device in the middle of the ethernet cloud between two RBridges.

Link State protocol role

- Collect:
  - RBridge Nickname
  - Connectivity between RBridges
  - VLAN topology
  - Multicast users
  - Options supported
- Build:
  - “My” Forwarding path to all other RBridges
  - Old fashioned tree for multiple destination frames

Transparency

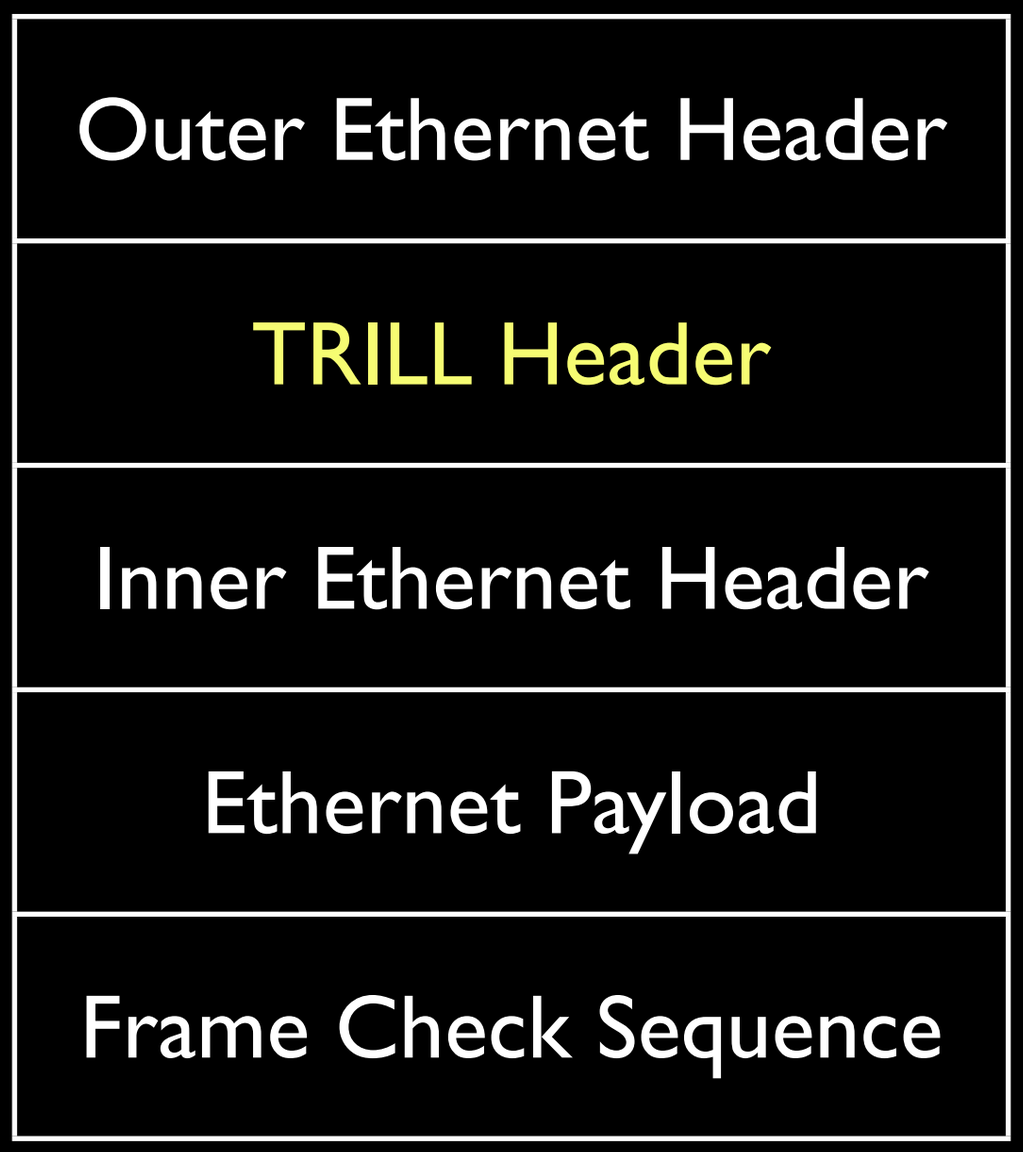


=

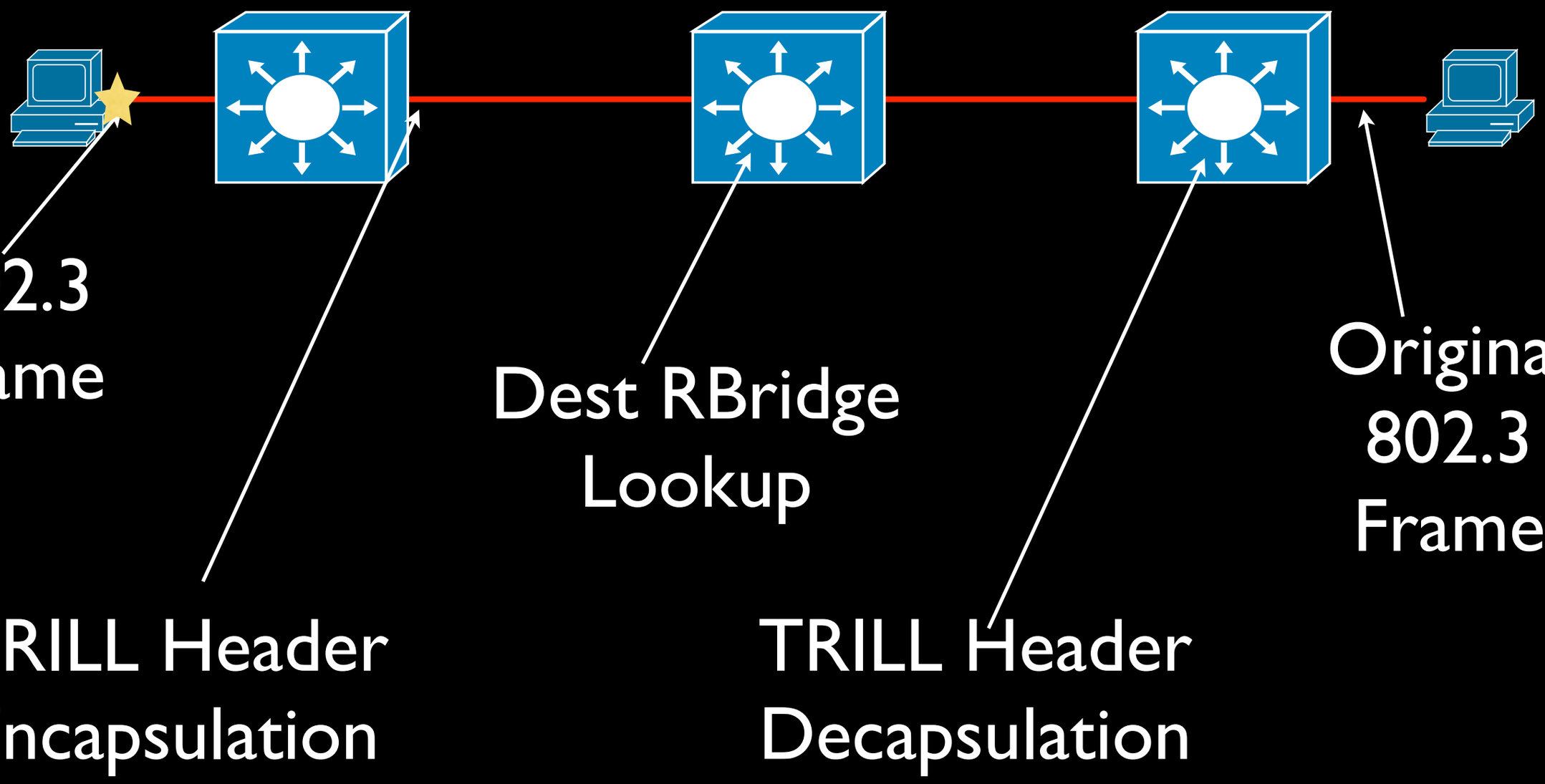
The RBridges see no difference between an ‘ethernet cloud’ of devices that do not support TRILL, and a direct interconnect between two TRILL speakers.

Frame Encapsulation

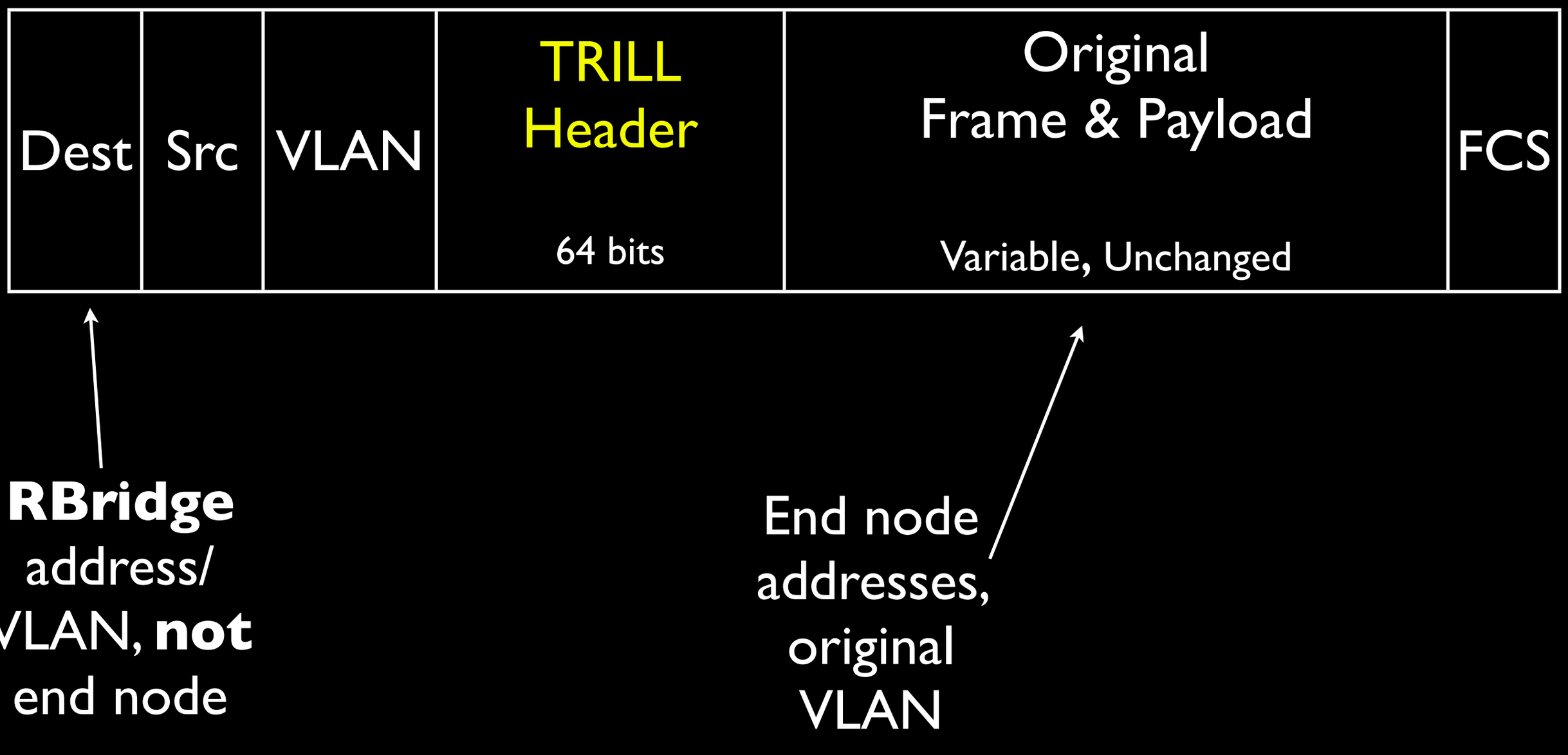
- Frames on Inter-switch links encapsulated with a header.
- It details the exit RBridge name to signal to other RBridges the frame's destination



Encapsulated Frame Quest



Frame Contents



TRILL header Contents

TRILL Ethertype	Vers 'n	Res	Multi Dest Flag	Opts	Hop Count
Egress RBridge Nickname	Ingress RBridge Nickname				

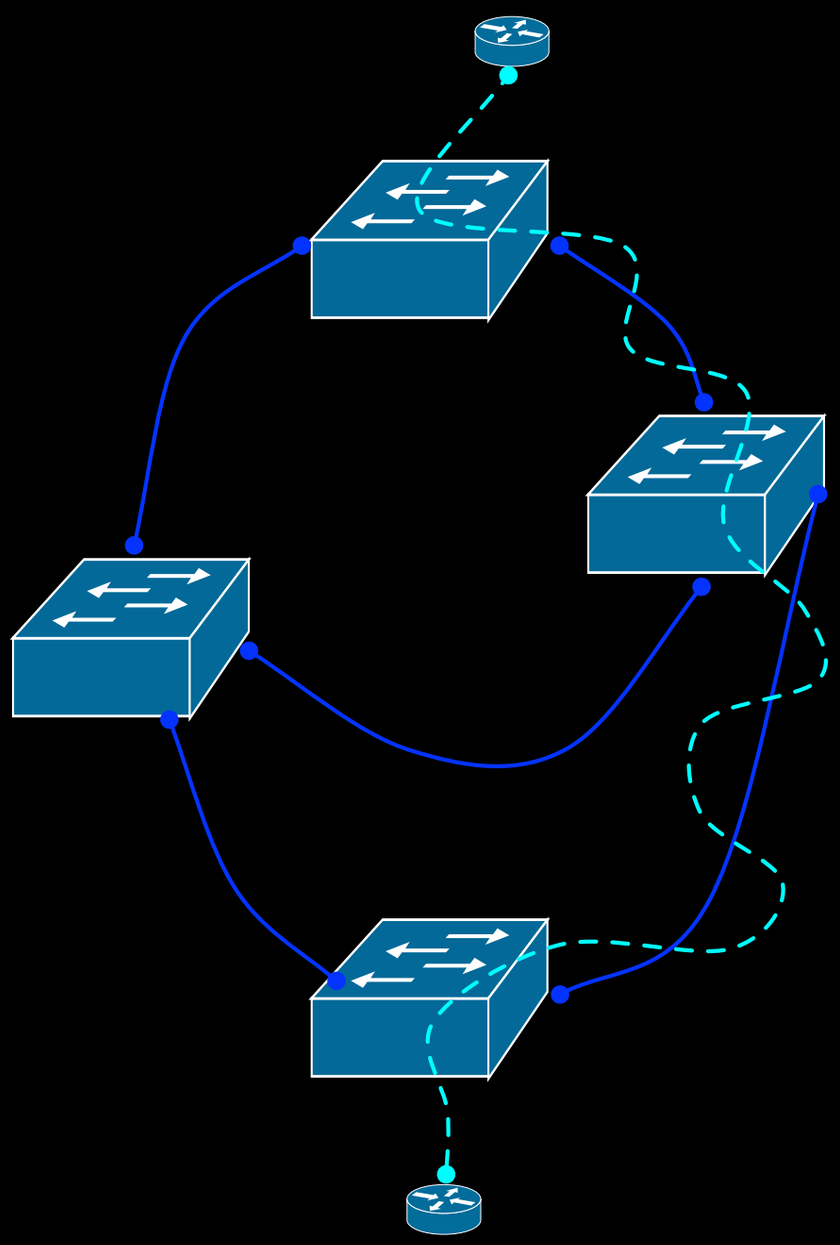
Not quite to perfect scale ;-)

Encapsulation Method

- Based on RFC3032 (MPLS Label Stack Encoding)
  - Making a method just for TRILL would have led to poor availability (expensive, risky)
  - Just nesting 802.x tags easy, but no TTL field
  - IP encapsulation inappropriate for layer2 service
- But this is NOT an MPLS feature

The ability to encapsulate and decapsulate frames in this way is built into the hardware capabilities of switches made by many different vendors today, and piggy-backing this established methodology means that TRILL's eventual deployment is much more likely to be a software upgrade for many users.

Known Unicast Forwarding



- Defined by the presence of a frame with known unicast destination MAC
- First RBridge encapsulates the frame with a TRILL header, that identifies the exit RBridge
- Forwarded “hop by hop”
- Exit RBridge pops the encapsulation and delivers native frame, unchanged

Multi-destination

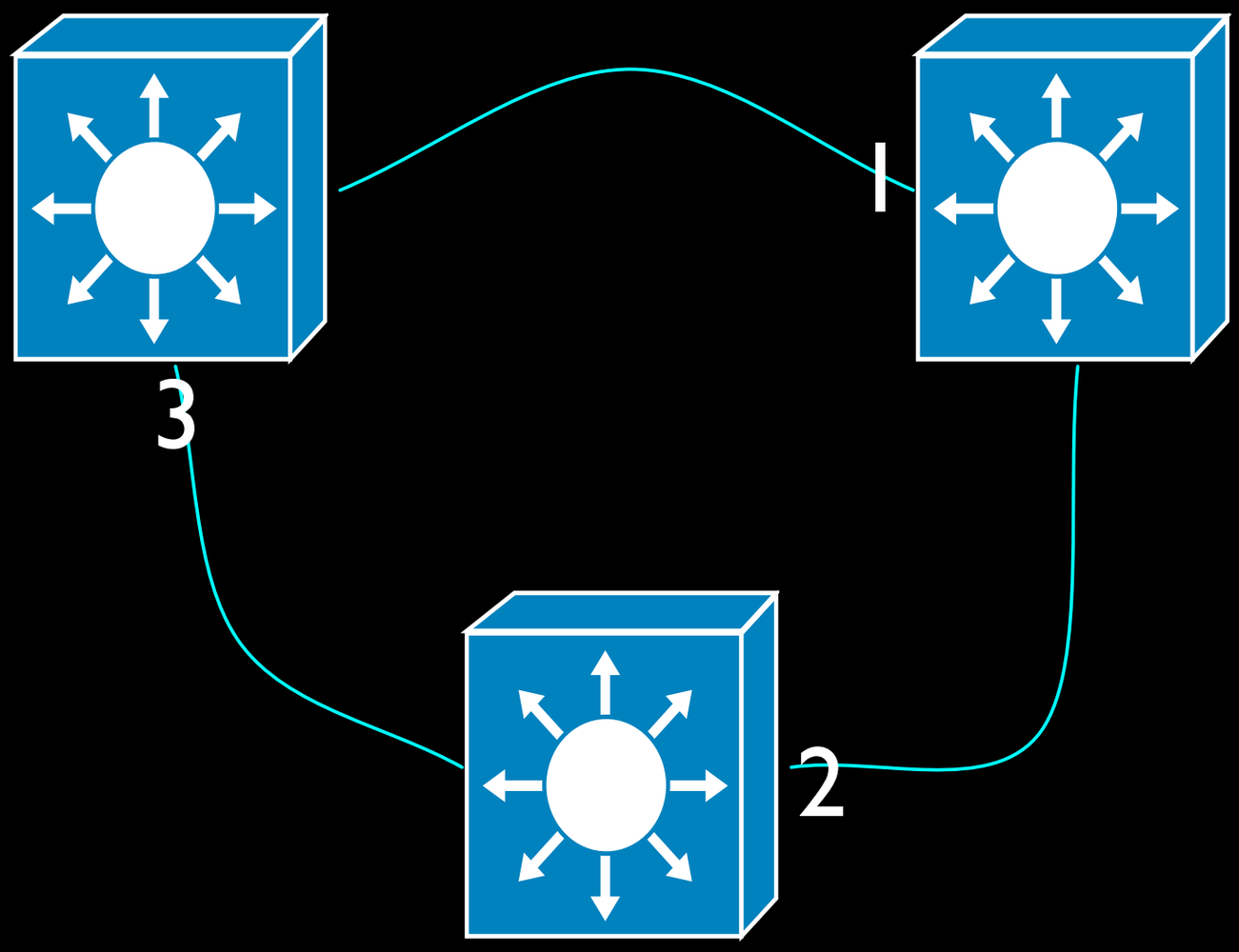
- Broadcast, multicast, or unknown unicast MACs.
- One or more bidirectional trees calculated and nicknamed
- SPF calculation, not Spanning Tree
- Links with no downstream nodes are pruned.
- Forwarding, generally, is handled by delivering frames to adjacent, downstream RBridges, according to the tree nickname specified in the TRILL header.

Loops

- Same issues imported from Layer 3....
- Theoretically could occur during re-convergence?

Loops - Hop Count

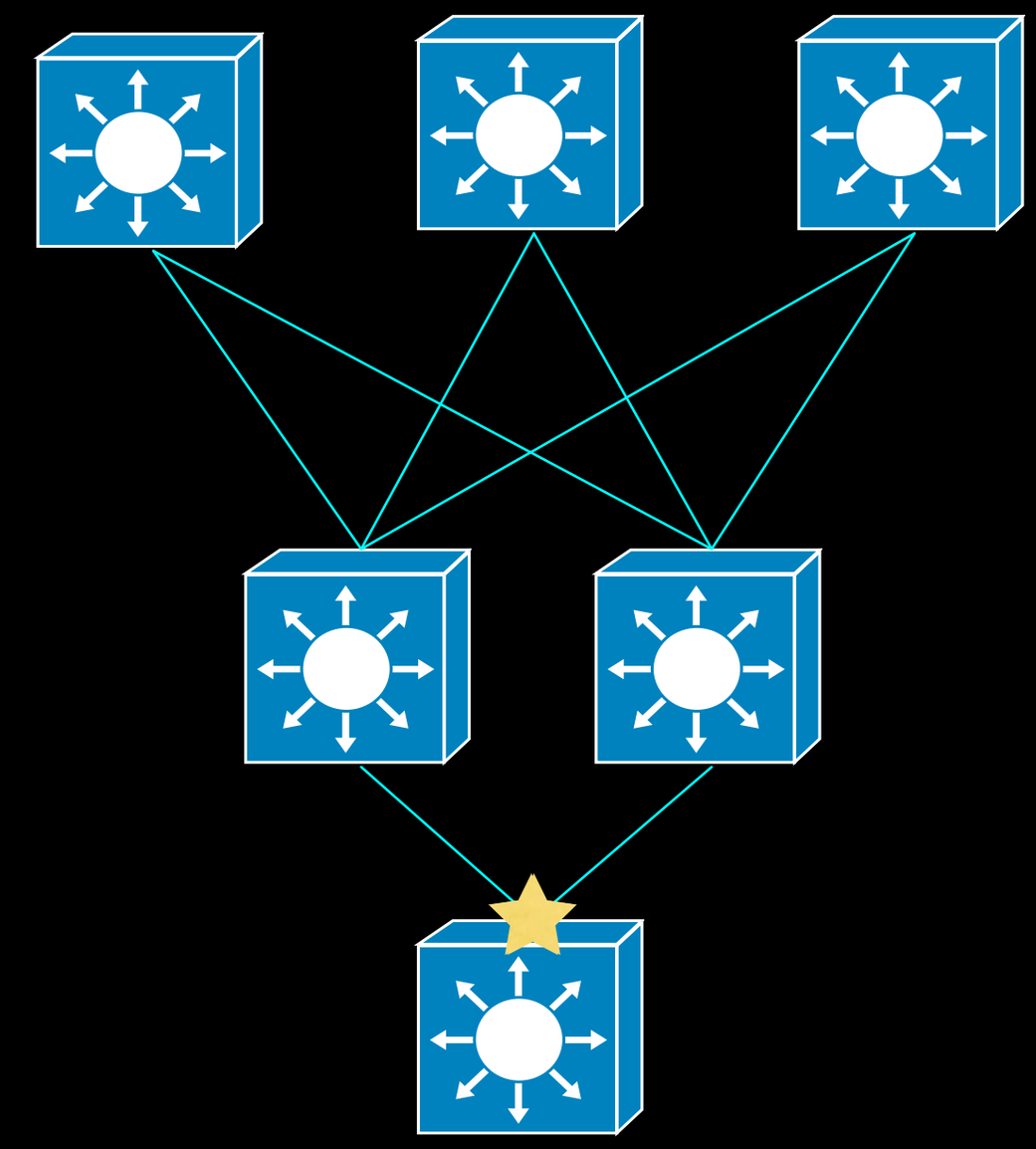
- Reconvergence woes mitigated with hop count



The TRILL header contains a hop count/time to live, and each RBridge should reduce the counter each time the packet traverses the bridge. Once the time to live counter has been exceeded, the packet is discarded. On small lan segments, this means that a loop could still be distruptive, and certainly leads to unwanted traffic. But reduces the risk of network death!

Loops - Multi Destination

- Packets should only be learned from ‘uptree’ switches

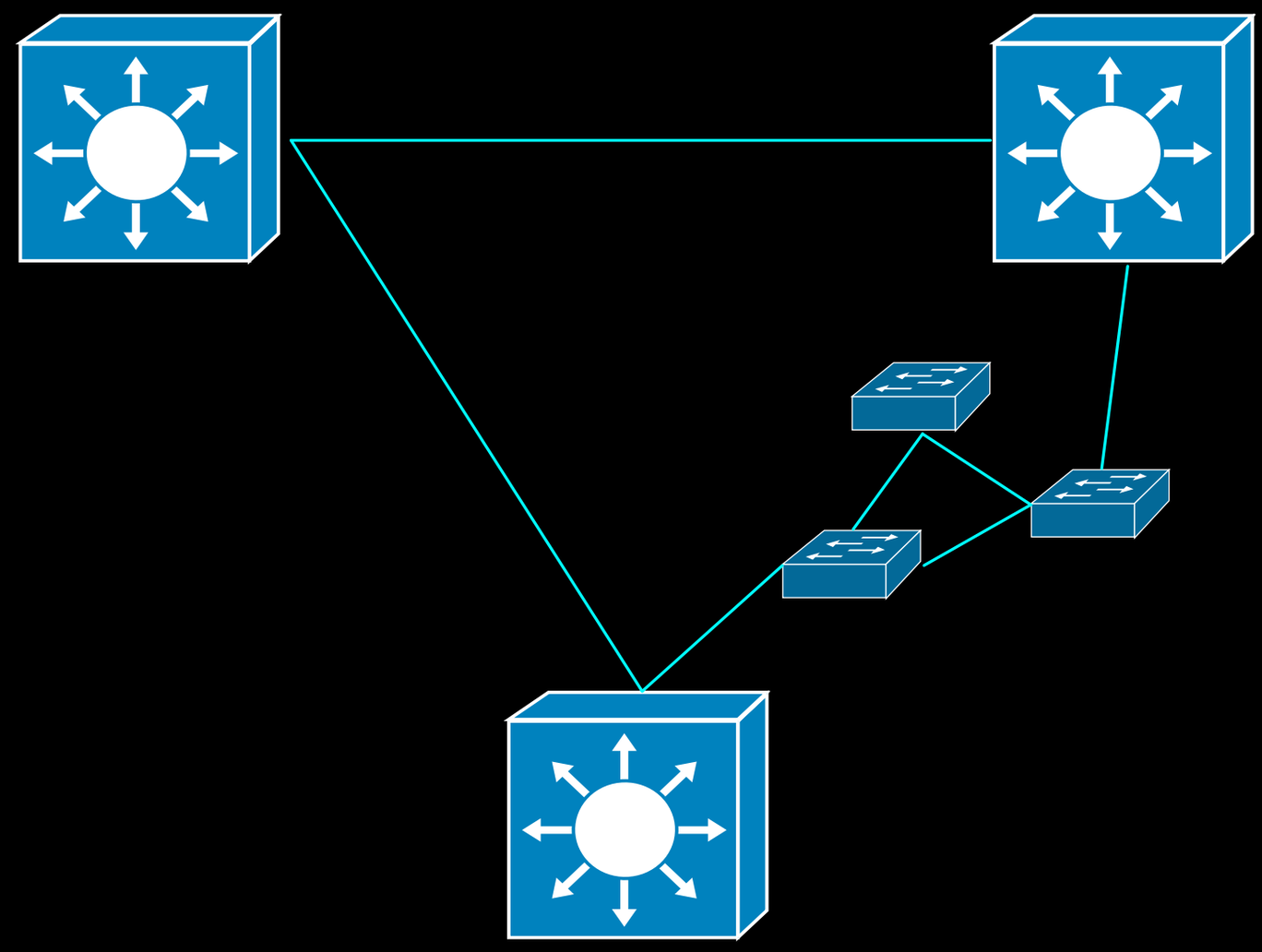


Packets with multiple destinations are delivered along a tree like structure. All RBridges know, from the point of view of the converged tree, whether a switch is ‘up tree’ or ‘down tree’. If a packet is received on an uptree switch from a downtree switch, it is discarded. There can be many simultaneous trees.

Loops - Middle Cloud

No Cute Animations:

This would just suck



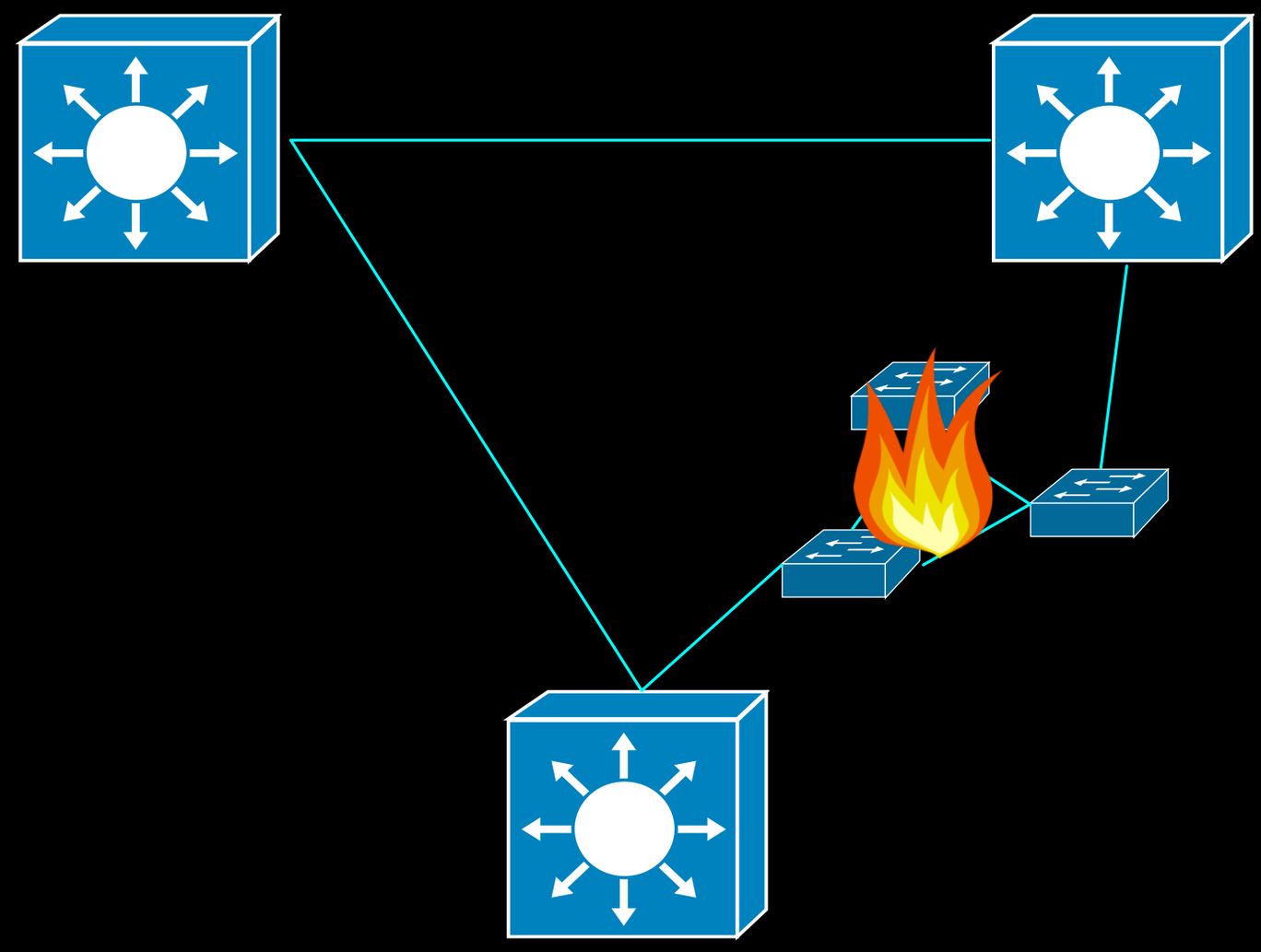
Ethernet clouds in between RBridges need to be loop free, otherwise all bets are off.

Loops - Middle Cloud

No
A s:

JOKING

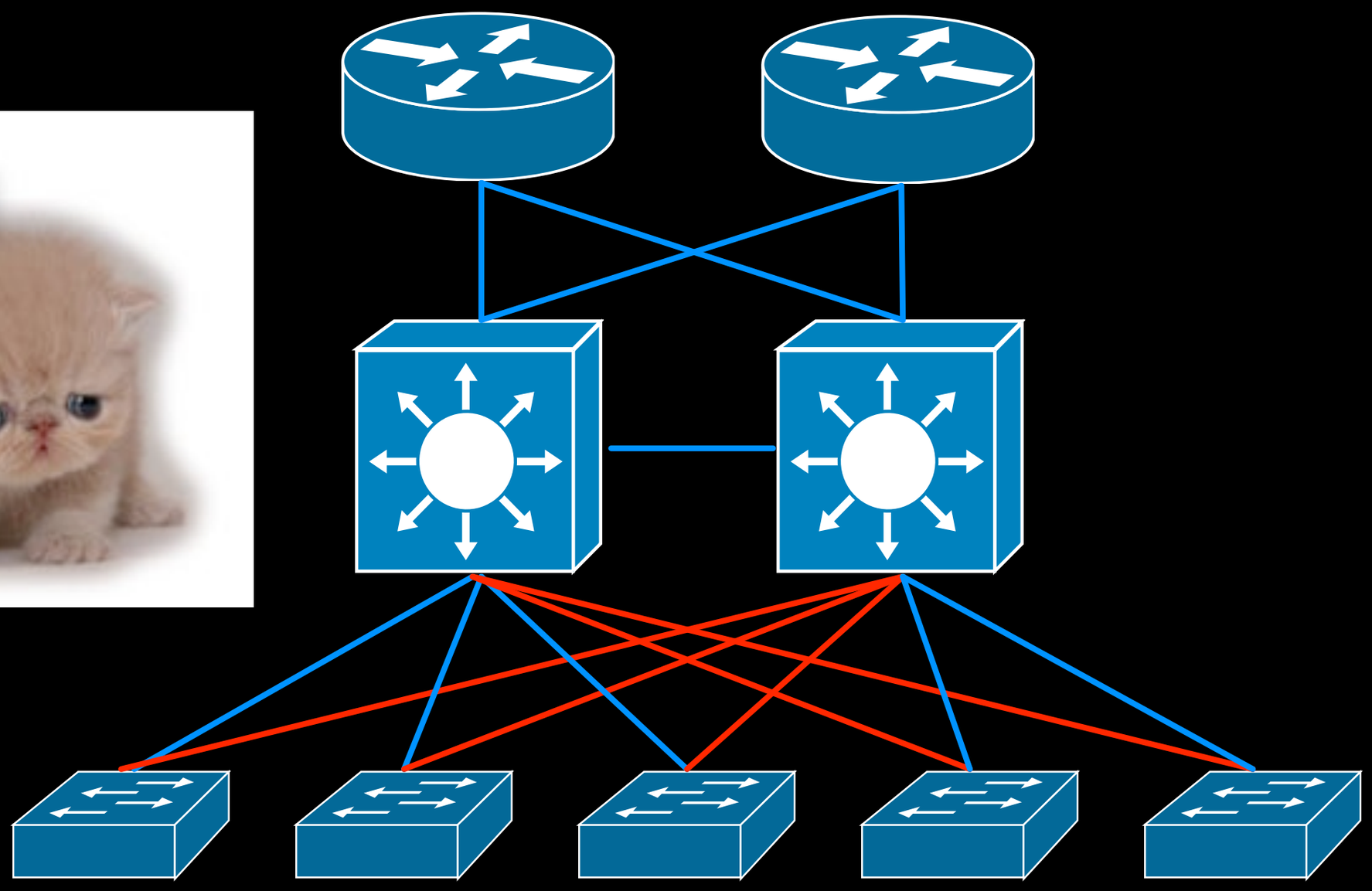
This would just suck



Agenda

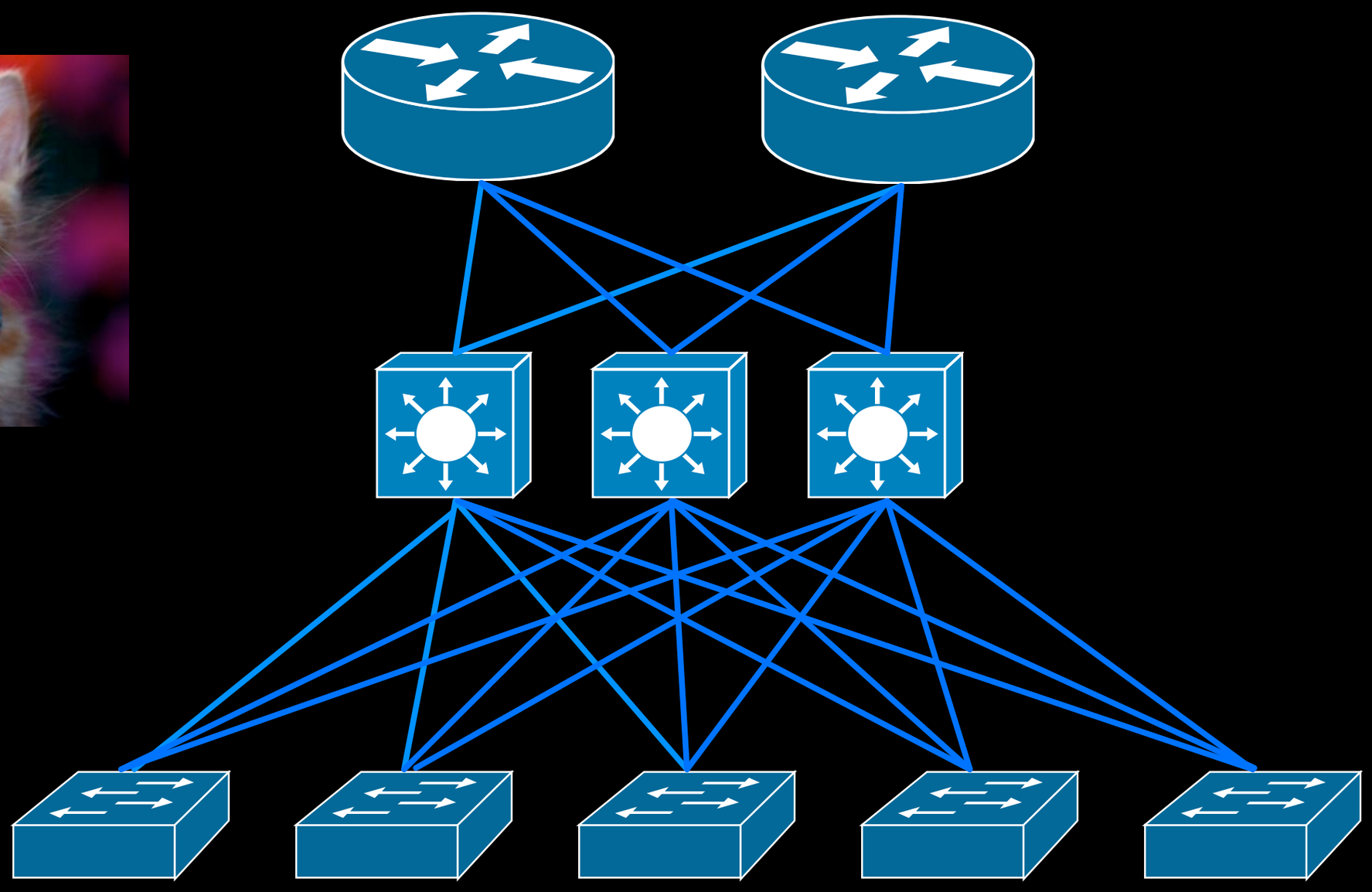
- What is the problem at Layer 2?
- How does TRILL work to solve this?
- Use Cases
- Other bits/reference material

Fictional hosting networks revisited



So, we're back to the original network, which had two distribution switches, a loop to every top of rack access switch, and spanning tree blocking ports that would have caused problems. The distribution switches can not handle the volume of traffic that they need to, and need to be upgraded. Network kitten is sad.

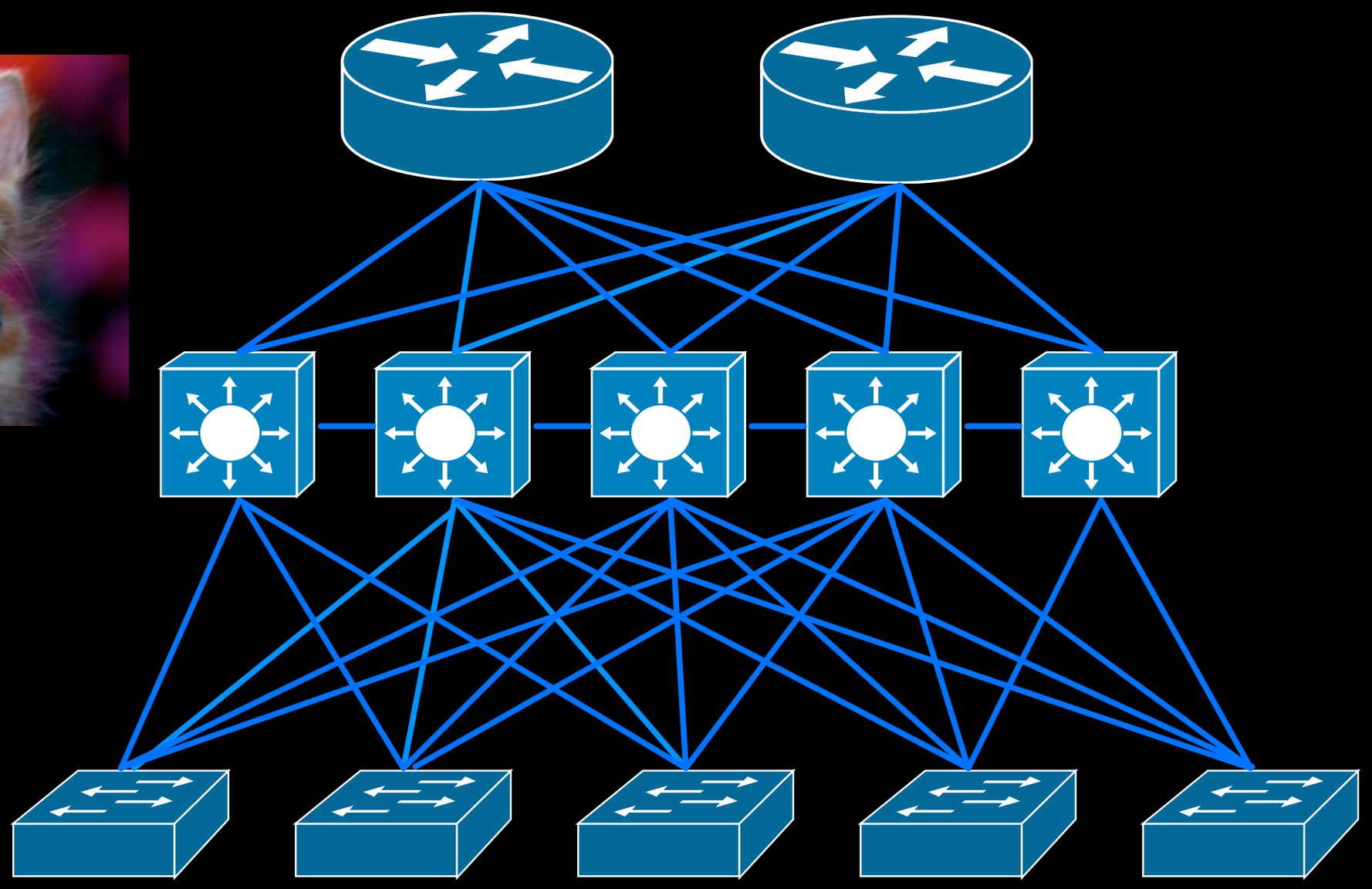
Scaling up with TRILL..



Deploying TRILL means that network kitten can add further distribution switches to scale traffic, and multipath traffic when there are peak loads....

It also becomes possible to interconnect top of rack switches, in the event that large volumes of traffic has to pass directly between two local switches. Doing this with only spanning tree protocol increases the risk that traffic will take a very suboptimal path.

Scaling up with TRILL..



... and more, and more distribution switches (as the requirement for top of rack switch volume grows.)

It is not mandated to fully mesh distribution switches with top of rack switches.

Benefits

- Shorter Layer 2 paths, with meshing
- Therefore improved latency
- RoI from resilient links
- No fu g loop prevention protocol faults
- Multipath forwarding to handle increased traffic volumes

Agenda

- What is the problem at Layer 2?
- How does TRILL work to solve this?
- Use Cases
- Other bits/reference material

Reference Implementation

- OpenSolaris RBridges
- Actually an end-host implementation!
  - Enhanced Quagga IS-IS
  - Xen/Virtualisation community interest
    - Better virtual machine mobility

Vendor Progress

- Brocade - Virtual Cluster Switching (“to ship 2011”)
- Force10 - List it as a ‘Key Emerging Data Centre Standard’
- Cisco - Are a TRILL protocol author

See Further

- Full Protocol Specification (in RFC Editor Q) https://datatracker.ietf.org/doc/draft-ietf-trill-rbridge-protocol/
- Problem Statement http://www.ietf.org/rfc/rfc5556.txt
- IETF workgroup https://datatracker.ietf.org/wg/trill/
- Solaris Implementation http://hub.opensolaris.org/bin/view/Project+rbridges/
- Initial vision, Radia Perlman http://www.usenix.org/events/usenix06/tech/slides/perlman_2006.pdf

Any Questions?

Andy Davidson

andy.davidson@netsumo.com

+44 20 7993 1702