IP Networking fundamentals-3

The “clever” bits Part 2

Apologies for the Hiatus!

It became quite clear that part 1 was running on a bit! Part 2 gets a (tiny bit) more abstract (initially).

Let’s focus on the “three-layer hierarchical model” first. At the bottom, we have the “access” layer. This is almost always a switch. Above this we have the “distribution” layer. This is often switched, and this distributes VLANs over 802.1Q “trunks”. It may be a routed layer, whereby the VLANs may be considered IP Subnets, and the distribution layer may be done as “internal routes”. This will depend entirely upon the size of the broadcast domains. Above this is the Core. Whilst this is also often switched, the Inter Core links (to cores at other places) is almost always Routed.

By contrast we have the Spine/leaf concept. This is much more dependent on routing. Once again, we find our old friend IPv4 struggling on with this task. My introduction to spine/leaf was somewhat awkward. It was very reliant on a tech salesperson who expressed themselves from a position of knowledge, but lacked the fundamental answers to migration. This is fine, and I think, through independent testing, I have a vague understanding. Spine/leaf uses the idea of overlay and underlay networks. This can be made to sound very flash; indeed, however it is another way of expressing encapsulation. I appreciate that I haven’t covered encapsulation yet, but let’s press on, and then all will be revealed! With that statement, I feel duty bound to keep this brief, and as accurate as I can! An Underlay network is the layer where the data travels between devices. It can be via IPv4 or IPv6. In a spine/leaf network, we remove broadcast traffic, and the broadcast messages in the overlay can be carried as selective multicast between devices, currently the general datacentre practice is to use VXLAN over EVPN, this is a wonderful expansion. The Overlay network is much more like a network as we would understand it. This is part of the beautiful de coupling of the infrastructure from the data payload itself.

VXLAN? Isn’t that VLAN with an X?

Functionally, this is true. This is the way to stretch a l2 broadcast domain over an entirely unicast underlay network, using a technique called ingress replication. This is the method used in big data centres for “multi-tenanting”. VXLAN has not just the “small” number of 4094 useable VLANS it has 24bits of “VNI”. We can use a “VTEP” which is where VXLAN terminates (normally in our switch) and from there we can map a VNI to a VLAN locally! This means that if I want to share a particular VLAN with another organisation, I can do this via VXLAN over BGP. I can communicate the IP Subnet containing my VLAN through my BGP advertisement. I can decide whether I wat to establish a BGP “peering” arrangement with an external entity by sharing a cryptographic key with them. The nice thing about VXLAN and VNI is that I may have my devices in VLAN 3 (for example) my VTEP maps this to a vni (24-bit number), this travels over BGP to my client organisation, who see the incoming 24 bit number, and can map that VNI to whichever VLAN they deem correct for their purposes. By using ingress replication, my devices and their devices appear to be in the same L2 network!

You hate Broadcast traffic? How does this work over unicast?

One of the best things about VXLAN over a spine/leaf network is that it is sent from the VTEP (switch) as unicast (actually as udp) to other VTEPS (switches). Because this is a standard udp transaction, it is unidirectional, this means that a “broadcast storm” is limited to only existing in a single VNI, just like it does in a VLAN segmentation, Thus limiting the “blast radius” of a patching error. Because, in the overlay, the VNI-VLAN mapping very really “stretches the L2 broadcast domain” (potentially into other sites! There is still a need for Spanning tree at the edge, to guard against patching errors.

VTEPS are used to bridge the data traffic that flows over the underlay, de-encapsulate it, then, relying on their VNI to VLAN mapping table, forward downstream as “tagged” 802.1q frames. The beauty of this approach is that the underlay network behaves as infrastructure. The job of the Underlay is to forward traffic (in this case udp vxlan communication) to the devices needing it. This does create some duplication. Especially where multicast (and yes broadcast) transactions occur. For this reason, it is common to provision quite huge capacity links between leaf nodes and the spine. This is because bandwidth is now abundant, and the processing capacity of spine/leaf capable switches is vast. In fact it is probably not appropriate to use the term switches anymore. The more common word is “node” as these devices are equally at home transacting frames, as they are encapsulated frames, whilst running serious application driven traffic like BGP, VXLAN and PTP to name but a few.

This is likely to be the reason why most modern “expensive” nodes look very similar to each other. The commoditisation of fabric has started a move into value-add OEMs buying of a small (and largely hitherto unknown) set of ODMs. Before, a switch of substance relied upon FPGA gateware an FPGA (field programmable gate array) is a beautiful and flexible platform where “hardware” meets “software”. Typically programmed in Varilog, or VHDL these devices can have their hardware behaviour changed in software. They make excellent prototyping devices, and in the mid 2000’s up to about 2012 they were economically attractive when compared with the capital risk of committing to an ASIC (Application Specific Integrated Circuit). As Microprocessors, and their more ubiquitous cousin the microcontroller has become more powerful, for most computational load, this “software defined hardware” has become the norm not the exception.

Does this mean I can go and deploy a network?

The information contained in here is not an attempt to cover the myriad details of a modern data network. The purpose is to present modern practices in the context of where they can be very useful for an entertainment practitioner. I have avoided re-capitulating ip subnetting and a detailed explanation of routing protocols. Having the foundational essential knowledge, will hopefully aid in understanding, and this brief set of articles will hopefully aide.

The next topics to cover are:

Ptp – ieee1588 is/has become the high accuracy time reference for many things, including audio/visual networks.

BGP – the Internet’s routing protocol (and probably when adopting spine/leaf, the only routing protocol needed (phew)

IPv4 subnetting – worth learning!

DHCP – definitely worth understanding, especially dispelling the myth of the “problem of multiple DHCP servers” (a bug fixed a long time ago).

IGMP and PIM – vital if wanting to do multicast routing in IPv4 its IPV6 equivalent MLD is a thing of beauty.