There Layer Campus Design is an enterprise network design that depicts various layers of nodes, theire functions and position witein an enterprise network.

Typically it consists of Core nodes, Distribution and Access nodles.

It features or works similarly like a tree would:

Core: root

Distribution: aggregate branches

Access: Branches.

(i) Access layer: these is comprised of local clients like end-users, Servers and their traffic is backhauled via ~~a~~ switch that has connection to the distribution node.

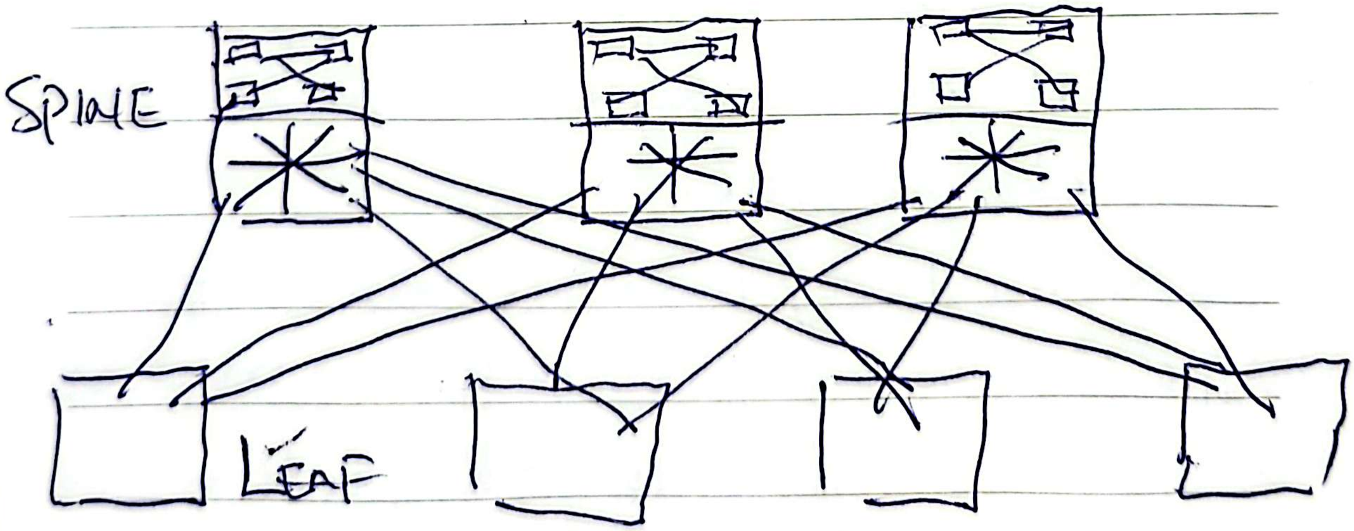
(2) Distribution layer; these are nodles that recieve traffic from several access nodes and backhauls to the core node, they also handle functions like routing, filltering, QoS etc.

(3) Core layer! these are nodes to interconnect to other networks, high speed backbone with fast reliable transport.

SPINE - LEAF DESIGN

Spine-leaf design was introduced to eliminate the limitations of a 3-tier design which consists of Core, Distribution and Access nodes which increases latency when a clients traffic is backhauled as it has to go through Access - Distribution and to Core.

However, spine-leaf design integrates both the Core nodes and Distribution nodes of 3-tier into one layer called Collapsed core design introduced to reduce latency,



A payload only has to travel through leaf and Spine layer to reach it destination as against 3-tier design.

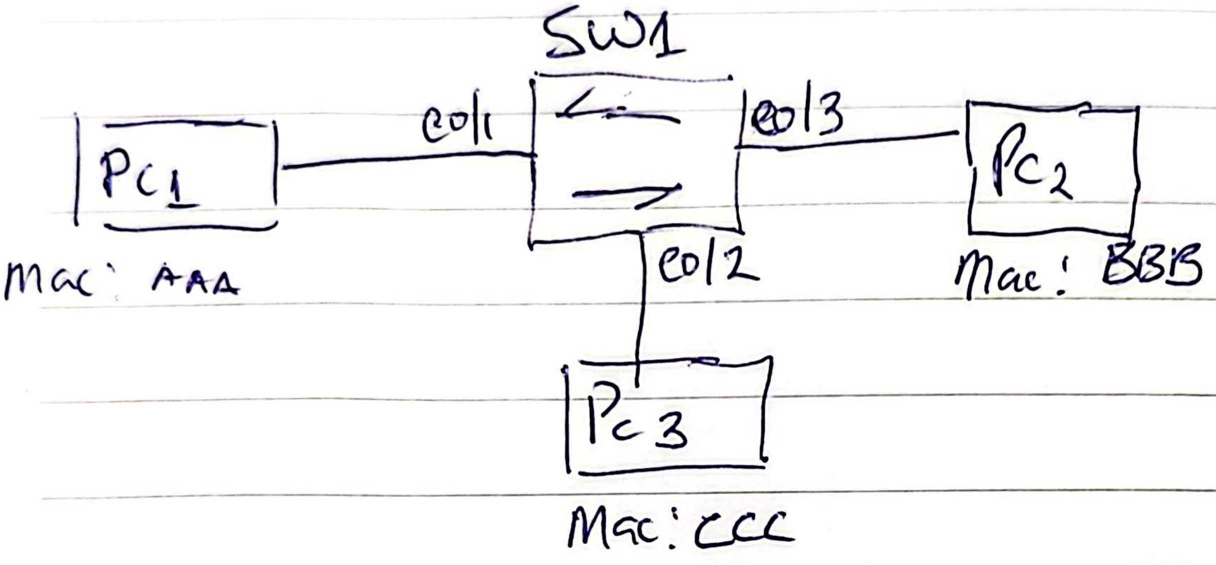
CORE COLLAPSED DESIGN

Collapsed core design is the combination of Distribution and core nodes to function as one, Collapsed Core nodes has the capability to route & swith packets at the same time thereby reducing latency and cost of acquiring multiple nodes which differs in functions and capability.

Collapsed core node are found in spine-leaf network designs in campuses.

HOW DOES SWITCHES LEARN MAC ADDRESS

Case study Topology:



Pc1 sends a frame to switch 1 asking to reach Pc2, it will send a frame with source/destination mac-address AAA | BBB and sends to Sw1, when Sw1 receives the frame it will add the mac of Pc1 to its mac-address table, looks into it mac-address table to locate the mac of Pc 2 but does not find it will FLOOD every port with the mac-~~add~~ of PC2 except the port it originally received it from, when Pc2 receives the frame from Sw1 and sees it own mac it will send a frame with its mac as source and destination mac of Pc1 and when Sw1 recieves the frame from PC2 it will add the source mac of Pc2 to it mac-address table then forwards the packet to PC1 since it already Knows how to get to PC1