Week 31

\*ARCHITECTURE of PARES.

PARES resides between the front-end console portal/API and physical data center. Consider an example of a tenant requesting a VM instance deployment through the cloud service provider’s front-end. The front-end processes authentication and access control, then virtual network configuration requests are sent to PARES through its APIs. PARES acts as an OpenFlow controller to program edge switches dynamically, based on tenant requests.

B. PARES packet handling layers

Most of the PARES logic is for its OpenFlow proto- col control module, which programs flow tables of edge switches (data plane) upon the given network configuration, and dynamically updates data planes upon the front-end’s request. In essence, PARES programs the edge switches of SDN fabric in three layers: end host solicitation, network function and routing layer. The operation at these layers is elaborated below:

1) End-host solicitation layer: Separating names from location is a quintessential requirement for MTDCs, as providers and tenants require elasticity and VM migration. MTDCs typically place a shim layer inside each server, or run service nodes on the hypervisor to intercept location solicitation packets in order to prevent broadcast traffic from flooding the data center. Different from these approaches, PARES places these functionalities of end-host solicitation on edge switches in the network fabric. This layer is the first layer a packet encounters when ingressed on the switch. If the packet is matched to any end-host solicitation protocol (ARP, DNS or DHCP), whole packets are forwarded to the PARES OpenFlow controller, and PARES replies (by looking up its directory) and injects the reply back to the original port.

2) Network function layer: All traffic other than end- host solicitation is forwarded to this layer. PARES places the network function layer before the routing layer, such that packet rewriting can be completed before routing. For example, in case of load balancing, destination transport address usually needs to be rewritten at network function layer before its final routing at routing layer. It is important to note that network functions can be independent or require sequential order among each other. For example, network traffic from north-south traffic first requires firewall then load balancing. Network functions are diverse in nature; in this paper, we limit our scope of network functions only for multi-tenancy (tunneling and address isolation). After processing each network function, the packets are passed to the routing layer.

3) Routing layer: After the packet modification in network function layer, switching/routing decision is made in this layer. If the underlay network is Layer 2 fabric, this layer performs mac-learning process, and switching is performed based on MAC address. While flood-and-learn semantics of Layer 2 nature provides easy deployment, STP (spanning tree protocol) prohibits exploiting possible redundant multiple parallel paths to increase overall bisection bandwidth. Although TRILL and LISP can be solutions for exploiting multipath in Layer 2 data center, it does not solve the problems of limited scalability caused by the number of TCAM entries in switches. As a result, enterprise/multi-tenant data center networks have evolved to shift from Layer 2 to Layer 3 fabric, due to reasons outlined below. With the introduction of the multi-root fat tree network topology, which places all switch and servers in Layer 3 and exploits the existence and predictable latency of multiple parallel paths amongst all nodes, a pure Layer 3 data center network topology has become popular, especially in massively large data center. Additionally, it maximizes the bisection traffic in data center network using commodity switches without the requirement of expensive high capacity switches at the core. If Layer 3 fabric is used as an underlay network, switching is performed based on IP address.