U-4 (SDN).

* Northbound Application Interface - Q1 -

→ Interface / set of protocols that allows appln or s/w sys to interact with a higher-level component in n/w architecture.

→ Interface through which appl^n communicate with SDN controller.

→ SDN controller → central brain of n/w → controls how n/w traffic flows
↳ maves decisions on how packets are forwarded.

→ Northbound API → defines methods, data structures & protocols use to exchange info with SDN controller
→ allows appear to programming retreive n/w info, seb policies, configure n/w paths.

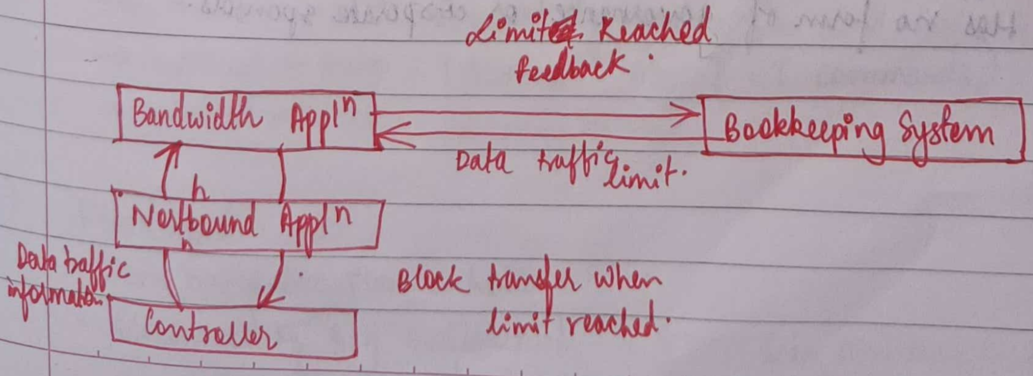
→ Northbound API → provides standarized way for apped to send commands.

→ By using Northbound API → appl^n can leverage capabilities & flexibility of SDN.
→ To implement n/w services'
→ optimises traffic flows..
→ monitor n/w performance
→ performs other n/w related tasks.

→ API abstracts → underlying complexities of SDN controller.
↳ provides a simplified & standarized interface for appl^n)

→ specific design & features of N API can vary depending on SDN controller implementation.

→ common examples of Northbound APIs used in SDN include. OpenFlow, NETCONF, RestFull APIs.


~~Limit~~ Limited Reached feedback.
Bandwidth Appl^n
Bookkeeping System
Data traffic limit.
Northbound Appl^n
Data traffic information.
Controller
Block transfer when limit reached.

✱ Current Language & Tools -

① Frenetic -

- domain specific language for programming OpenFlow networks-
- allows new operators to program n/w as a whole than manually configuring configuring each connected n/w device.
- introduces set of purely functional abstractions enable module program
- defines high level, programming centric, packet processing operators & eliminates many of difficulties of 2 tier prog model.
- embedded in python → 2 level abstraction.
  1) High level abstraction.
  2) Modular constructs.
  3) Portability
  4) Rigorous semantic foundation-

② Procera -

- high level n/w control language that allows new operators to express reactive n/w control policies, without having to resort.
- networks. devices n/w intelligence solns based on deep packet inspection technology.

③ Ryu -

- highly modular, small SDN controller → Python.
- core of Ryu is smaller than controllers, every feature is implement.
- supports multiple OpenFlow versions, along with related protocols-
- Has no form of governance or corporate sponsors.

DOMS Page No.
Date / /

④ Open Daylight -

- modular controller written in Java.
- supports large no. of networking protocols OpenFlow, BGP, LISP.
- has a large amt of backing from diff companies.
- uses Apache Karaf framework from diff. companies
- config & logging handled by Karaf.
- Karaf has no. of components of available for use, most important a web server used by northbound interface of OpenDay light.

⑤ NetKAT -

- n/w language for SDN programming has matematical & semantic foundation.
- applies its complete equational theory & provides techniques.
- appl^n in n/w include → checking reachability, isolating traffic b/w programing.

⑥ Mininet (Tool) -

- network emulator capable of creating virtual networks with 100s of host & switches on single computer. ~~Bo~~
- Based on Linux process virtualization to run nodes in OS kernel.
- mininet can create a realistic virtual n/w or any type of machine.
- provide an inexpensive soln & streamlined development running in line.
- Basic commands -

→ sudo mn -c (after logging into error, start mininet reset to initiate.
→ sudo mn (start mininet w/o cleaning it up)
→ mininet > net (to list typology type)
→ mininet > help (display mn CLI commands).
→ mininet > nodes (display nodes)

⑦ Floodlight -

- Java based OpenFlow controller
- developed by Big Switch Network & used in commercial switches
- compatible with h/w & virtual switches as OVS.

~~*~~ Composition of SDN -

① Flow table -

→ fundamental data stovchue in SDN devices

→ FTs allow device evaluate incoming packets → takes apt action based on conttents of packet.

→ Resides on h/w drivee + consists of series of flow entries + action to perform when a packet matching flow arrives at device.

→ When SDN device packet, consuts in flow tables in search of match

→ If doesn't find match, switch can either drop packet or pass it to controller.

→ Each flow table entry contains -

ⓐ Match fields -

ⓑ Priority -

ⓒ Counters -

ⓓ Instructions -

ⓔ Timeouts -

ⓕ cookie -

② SDN s/w switches -

① SDN s/w device implementations found in s/w based sh/w devices, such as hypervisors of a virtualization system.

② Implemetation of SDN devices in s/w is simplest means of creating an SDN device - flow tables, flow entries & match fields involved are easily mapped to s/w data structures such as sorted arrays & hash tables.

③ Hardware SDN devices –.

① H/w devices utilise special hardware designed to faciciate the inspection of incoming packets & subsequent decisions that follow based on packet matching opelation.

② H/w includes layer 2 & layer 3 forwarding tables Usually implements using-

CAMS – content addressable memories.

TCAMS – Ternary content addressable memories.

* NFV (Network Functions Virtualization) –

- way to virtualization h/w services, such as routers, firewalls load balancers (traditionaly h/w or propriety h/w).
- services are packaged as VMs on commodity h/w it allows service providers to run their n/w on standard servers.
- NFV architecture consist of.

ⓐ VNF – (Virtualized n/w func).

- software appln that derives n/w func such as file sharing, directory services & IPconfig.

ⓑ Network Func Virtualization Infrastructure (NFVI).

- consists of infra components, compute, storage, networking on a platform to support s/w such as hypervisor sive KVM or a container mgmt platform neded to run new apps.

ⓒ Management & Automation & N/w Orchestration (MANO) –

- provides framework for managing NFV infrastructure & provishig new VNFs.

(4) Critical goals of NFVarch -

① Decomposition of phys n/w elements into virtualized h/w fns to allow operators to choose best implementation from range of vendors.

② Portability of UNFs to diff h/w platforms & hypervisors.

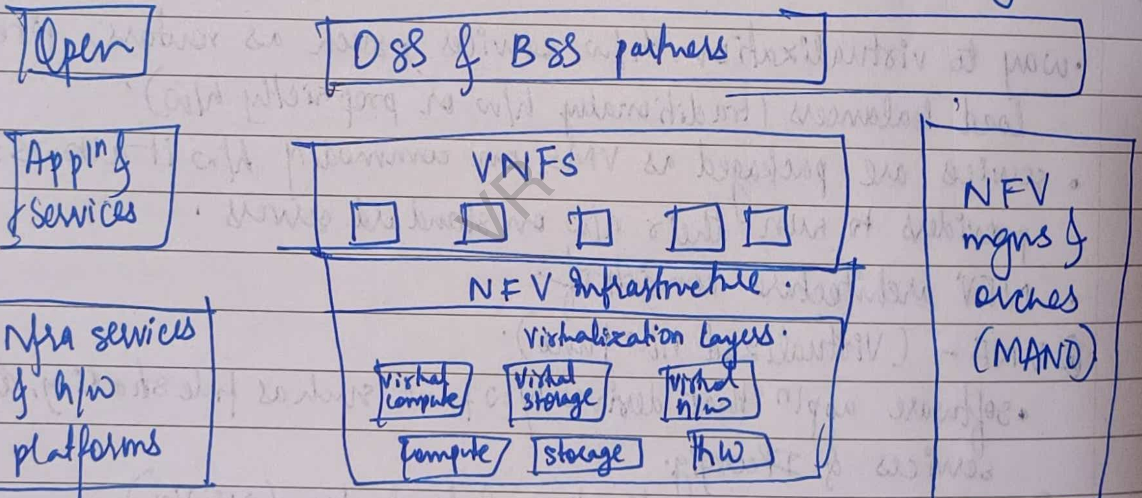
③ Rapid S/w based service delivery.

④ standard, open interfaces for improved interoperability in multivendor NFV soln.

⑤ use of low-cost, commercial off shelf (COTS) h/w

⑥ MANO provides overarching mgmt & orchestraction of VNFs in NFVarch

⑦ MANO instanties n/w services through automation, provisioning & coordination of workflows to VIM & VNF managers.



* SDN Implementation -

1) Define a use case 2) Assemble a cross fuctional team

3) Test in less critical network area 4) Review your test case

Q5/ 5) Gain maturity before expanding deployment.

- No. of prog lang. for SDN & majority can be applied to open Flow.
- OpenFlow - 1st & only standardized interface b/w control & infrastructure layer.

* Appln → ① Data Centers ② Enterprise Networks.
③ optical n/w ④ Bandwidth mgmt
⑤ content availability.

Q12.