

U-2 (SDN)

DOMS

Page No.

Date

/

/

* Matching process in SDN (Flow table format)

- Packet arrival
- Header Parsing
- Flow table lookup.
- Match Found (Success) → Action set defined in that entry.
- No Match (Failure) → default behaviour programmed.

- flow tables → multiple entries with diff Match fields & Action sets.
- order of entries → switch prioritizes earlier entries during lookup process.
- controller dynamically adds, modifies or deletes flowtable entries to manage network traffic based on requirements.

* Ports in SDN

- 1) Physical ports → Real hardware ports on network device.
 - Includes Ethernet ports & tunnel ports.
 - handles forwarding of data packets.
 - connect to end device or other network switches for data transmission.
- 2) Logical ports → switch defined ports don't match up with switch's ^{hw} interfaces.
 - s/w concept within controller to manage physical ports.
 - could have additional pipeline called Tunnel-ID.
- 3) Reserved ports → generic forwarding behaviours specified by OpenFlow reserved ports including learning, controller, flooding or forwarding improperly utilising non openflow methods.

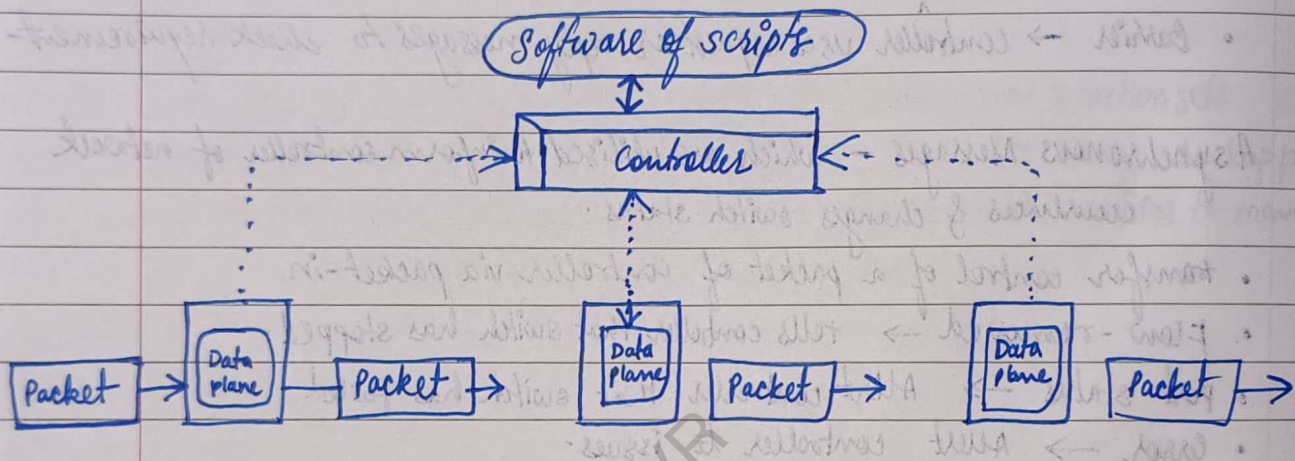
* Types of msg in SDN -

- a) Control switch messages - initiated by ~~SDN~~ controller & used to direct manage or inspect switch.
 - Features → switch must as for identify
 - modify state → flow mod.
 - statistics → Read-states
 - Barrier → controller uses request or reply messages to check requirement.
- b) Asynchronous Messages - which are utilised to inform controller of network occurrences & changes switch status.
 - transfer control of a packet of controller via packet-in
 - Flow-removed → tells controller that switch has stopped.
 - port status → Alert controller that switch has failed
 - error → Alert controller to issues.
- c) Symmetric msg → sent on their own initiative, either by switch or controller.
 - hello - keep alive message b/w controller & switch
 - Echo - sending an echo from either switch or controller verifies link.
 - experimenter - openflow switch to provide more functionality inside openflow message type.



Working on SDN controller -

- software defined networking controller is an applⁿ & SDN architecture that manages Flow control for improved network management & applⁿ performance.
- typically runs on a server & uses protocols to tell switches where to send packets.



- In SDN, all networking devices should be connected to controller in order that all it will regulate information planes of all devices once drawing design of network, typically network designer places networking devices below controller.
- Southbound interface is associated in missing interface between a program on controller & a program on networking layer.



Proactive Data Flows

- established in advance
- immediate availability
- low latency
- preemptive installation

network devices

Routers (routing table)

- Adv →
 - fast forwarding
 - scalability
 - security control

- Disadv →
 - overhead
 - inefficiency
 - slow convergence

- eg → large corporate network

Reactive Data Flows

- setup on demand
- delayed availability upon packet arrival
- higher latency due to setup delay
- dynamic installation upon packet arrival
- Routers, switches (learned paths)

- Adv →
 - resource efficiency
 - fast convergence
 - scalability

- Disadv →
 - potential delays
 - security concerns
 - ~~security~~ complexity

- eg → cloud computing environment



OpenFlow Protocol

- Is a standardized protocol that enables communication b/w the control & data planes of an SDN architectures
- allows centralized control over network devices by separating control logic from forwarding hardware.

Key components

- Controller (central entity responsible for managing network)
- Switches (network devices support OpenFlow & controlled by controller)
- Protocol Msg. (Msg b/w ~~the~~ the controller & switches to configure)

Limitations -

- Maturity
- Performance
- Control plane
- Vendor lock in
- Limited protocol support
- complexity of deployment
- lack of support for certain advanced networking features
- limited security features.

Benefits

- easier implementation
- vendor independence
- automatic discovery
- flexibility & scalability
- Traffic engineering
- security concerns



Role of SDN controller

- Centralized Management
- Global Network View
- Flow table Configuration
- Traffic engineering
- Policy Enforcement
- Dynamic Adjustment
- Integration with Applications

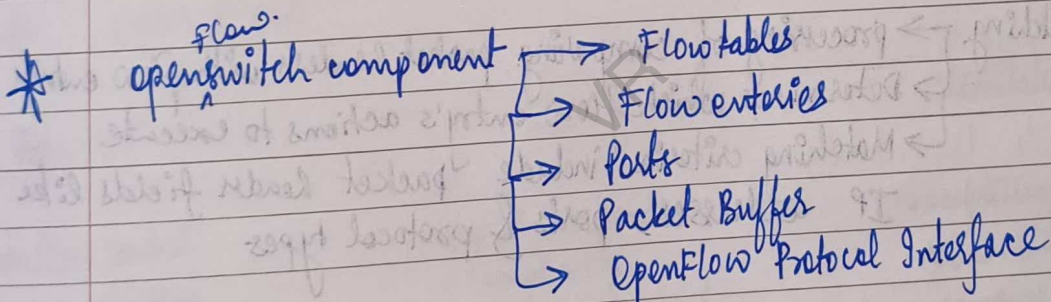
Open Flow Pipeline processing

- Packet arrival
- Header Parsing
- Flowtable matching
- First Match Wins
- Action Taken
- No match (default behaviour programmed)

* Openflow switch with controller. & diff openflow switch component

- 1) Packet arrives at switch
- 2) switch extracts header info.
- 3) Switch searches flow tables for matching query Match Fields
Action Set
- 4) Match Found → Trigger corresponding action
- 5) No match → Default behavior or notify controller.

- 1) Initialization 2) Topology Discovery 3) Flow Table Configuration.
- 4) Packet Processing 5) Controller Interaction. 6) Asynchronous Message.



* Define terms related to open Flow.

- 1) Flow table → data structure in OpenFlow switches
 - stores flow entries
 - determines packet processing & forwarding
- 2) Instructions → directives within flow tables
 - define how flow entries should be processed
 - specify actions for matched packets or table-miss.
- 3) Action set → collection of actions for matched packets
 - actions include forwarding, modifying headers, dropping or sending to controller
- 4) Matching → processing of comparing packet headers with flow entries
 - Determines which flow entry's actions to execute
 - Matching criteria include packet header fields like IP addresses, ports & protocol types

* Open Flow Architecture -

- 1) Controller → manages & controls network.
execute network control appln.
Implement network policies.
• commⁿ b/w openflow switches using openflow
• provides a centralized point for network management.
- 2) openflow switches → divided into data plane & control plane.
→ forwards packets based on instructions from controller.
→ execute actions specified by controller.
→ communicate with controller to exchange info.
→ provides network forwarding based on controller decisions.
- 3) Communication Protocols → communication b/w controller & switches.
→ defines msg types such as packet in, flow-mod & port mod.
→ enables controller to send instructions to switches & receive info.
→ includes various msg types such as packet-in, flow-mod & port-mod.

* open Flow

Adv →

- Centralized Control
- Dynamic Routing
- Security Enforcement
- Automation
- Vendor Independence
- Flexibility & programmability
- Simplified Network Management
- Scalability
- Traffic engineering
- Innovation

Disadv →

- Vendor Dependence
- Complexity
- Centralized Control
- Maturity
- Performance overhead
- Limited Control Plane Functionality

Limitations →

- scalability concerns
- Vendor lock in
- security risks
- complexity
- protocol support
- performance overhead

* diff timers used in SDN

- 1) Idle timeout → Removes flow entries after a period of inactivity
- 2) Hard timeout → sets max. lifetime of flow entries regardless of activity.
- 3) Packet in timeout → limits time for buffering packet in msg before dropping.
- 4) Controller Response Timeout → defines max. time for controller response to packet in messages.
- 5) Echo Request Timeout → sets duration for waiting for a controller response to echo request.
- 6) Connection Timeout → specifies max. idle time for maintaining connections b/w switch & controller.
- 7) Miss send length Timeout → controls duration for buffering packets before sending them controller for processing

* Drawbacks of OpenSDN

- Vendor Dependency
- Scalability concerns
- complexity.
- single point of failure
- Security Risks.
- Performance overhead
- Limited Protocol Support
- Young Technology (evolving standards).