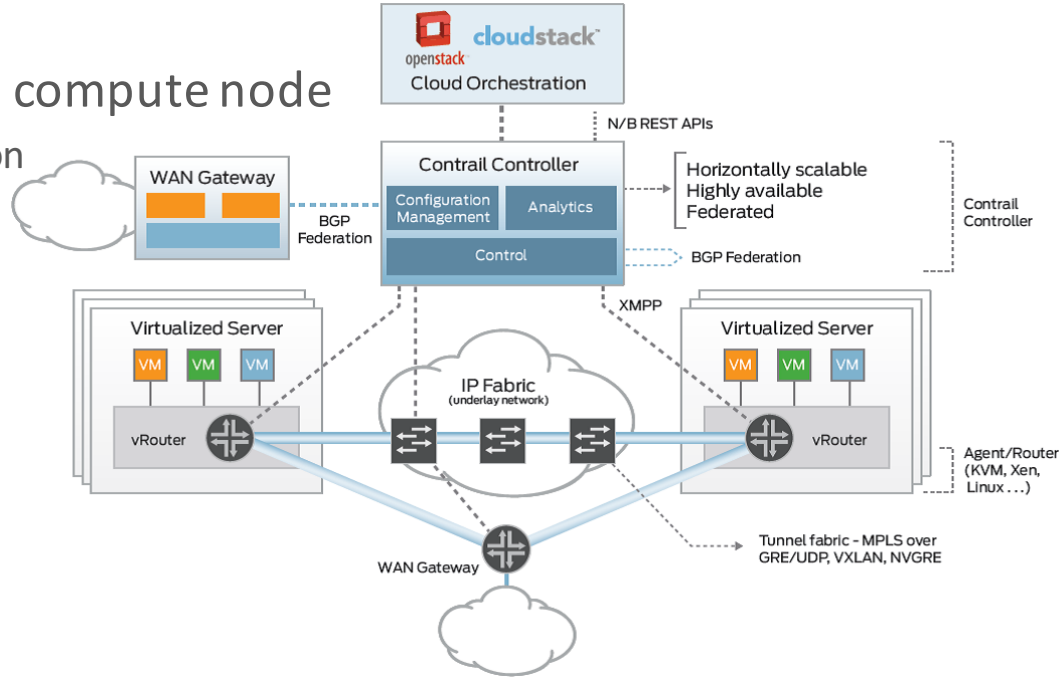
TF and DPDK

ALEX ROSENBAUM @ MELLANOX
[BY EYAL LAVEE @ MELLANOX]

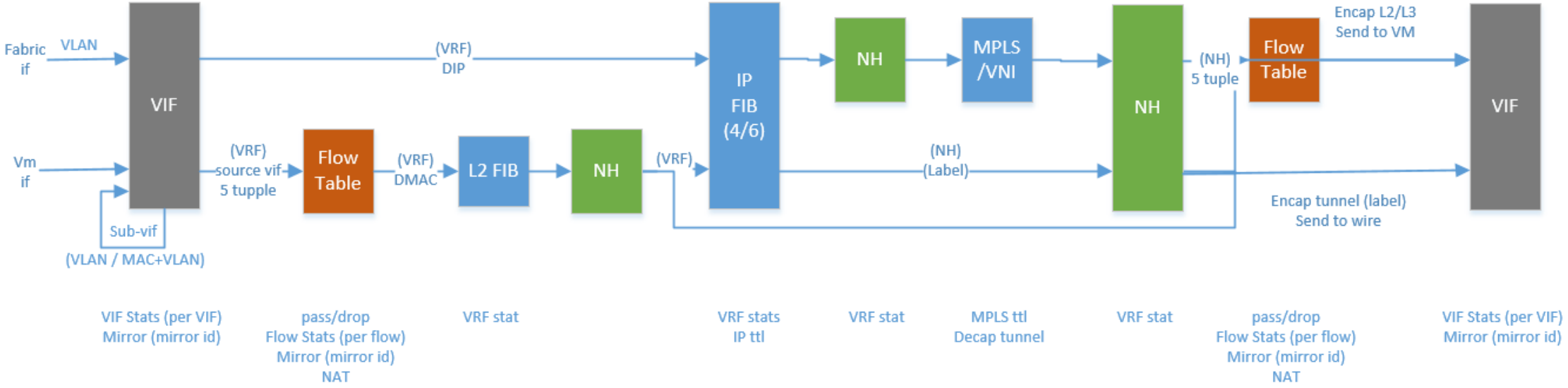
Tungsten Fabric - Overview

- Tungsten Fabric is an SDN-enabled management and control software for simplified service delivery
  - Centralized SDN controller
  - Virtual forwarder (vRouter) on compute nodes
  - Supports both Kernel and User space (DPDK) networking
- Open Source under Linux Foundation Networking Fund (LFN)
- vRouter is the virtual switch/forwarding element on the compute node
  - Similar to OVS in spirit but different in functionality/implementation



Tungsten Fabric – Software Pipeline

- Complex feature-rich multi-stage software pipeline
  - Multi-tenancy support
    - L2 and L3 VPNs (bridging and routing), IPv4/IPv6, various tunnel types (VXLAN, MPLSoGRE, MPLSoUDP)
  - Networking services
    - Security groups/policies, NAT, mirroring, load balancing, …
  - Advanced telemetry/analytics



vRouter Offload – WIP

DPDK
DATA PLANE DEVELOPMENT KIT

- Use rte_flow flow mark
  - Rx Data Path: tag well known tunnels/flows from fabric to VM (RX)
  - Flow spec:
    - Destination IP is vRouter fabric interface/bond IP
    - Match Tunnel Classification:
      - MPLSoGRE or MPLSoUDP and MPLS label
      - VXLAN and VNI
    - Tenant's Flow Classification – Inner 5 tuple
  - Flow action
    - MARK
- Rx data path can bypass software packet classification
  - When mbuf's MARK identification (m->hash.fdir.hi) is available and valid..
  - Retrieve all required forwarding information in single lookup based on MARK
  - Use forwarding information to bypass classification lookup operations
  - All other operations maintained unaltered in existing software data-path

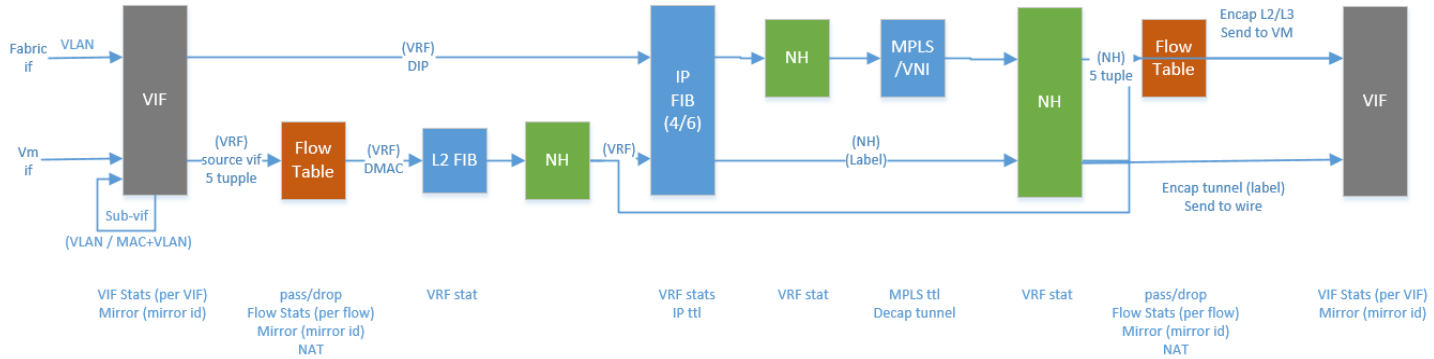
vRouter Offload – WIP

DPDK
DATA PLANE DEVELOPMENT KIT

- Use rte_flow flow RSS
  - Preform RSS on packet's **inner 5 tuple** of **MPLSoGRE** or **MPLSoUDP** tunnel
  - Add DPDK API reference for RSS on tunnels
- Rx data path can bypass software re-distribution, and more packet classification

vRouter Offload - Future steps

- Support SRIOV offloads
  - using rte_flow with DPDK port representors
- Full data path offload to HW
  - Extend rte_flow rules, table/groups, actions
  - Improve update rate



vRouter Offload - Future steps

DPDK
DATA PLANE DEVELOPMENT KIT

- Extend rte_flow
  - Allow offload complex multi-stage pipeline
  - Represent series of match/action tables using groups and JUMP action
  - Introduce “Steering Registers”
    - Pass intermediate state across table & lookups
    - e.g.: VRF from interface needed in FIB lookup
  - Provide hints on each table/group
    - match fields
    - wildcard types
    - LPM – HW offload support for librte_lpm?

vRouter Offload - Future steps

DPDK
DATA PLANE DEVELOPMENT KIT

- Mirror packet
- Split packet processing into two flows:
  - First copy of the packet continues to one set of actions and further processing – e.g.: queue to application
  - Second copy of the packet continues to second set of actions and further processing – e.g.: encap and return to wire
  - rte_flow API: two JUMP actions?

vRouter Offload - Future steps

DPDK
DATA PLANE DEVELOPMENT KIT

- Load balancing / ECMP
  - Indirection table, similar to RSS but goes to forwarding action(s) instead of queuing
  - Provide an array of N action sets (each element of array may be single action or series/list of actions)
  - Provide spec of fields to hash on
  - Performs hash on fields (modulus N) and executes action(s) in matching array element

vRouter Offload - Future steps

DPDK
DATA PLANE DEVELOPMENT KIT

- High rate flow rule and action update
- Statistics collection at high speeds
  - Requirement to read multi-millions of counters per second

DPDK
DATA PLANE DEVELOPMENT KIT

Thanks