Dual Mode, Multi-Protocol, Multi-Instance (DMM)

Contents

- Introduction
- Current challenges of transport layer protocol
- Proposal
- DMM framework design
- Road map & Scope

DMM Introduction

DMM (Dual Mode, Multi-protocol, Multi-instance)

- DMM is a concept to provide a running multi-protocol/multi-instance framework for the applications where applications can
  - Work with both user and kernel mode network stack
  - Use different network protocol stack based on their functional and performance requirement (e.g. QOS)
  - Work with multiple instance of transport protocol stack
- DMM also provides a development interface for userspace protocol stack developers.
  - Easy to integrate a new userspace protocol stack by the DMM stack interface
  - Easy to customize the algorithms of a available userspace protocol stack

Current Challenges

- Applications bringing different network requirements for QOS & SLA to Transport protocol
  - Often TCP protocol is a performance killer i.e) during packet loss/congestion results in very low link utilization
  - Transport layer protocol capability, which is traditionally provided by OS networking stack and monolithic in design and inflexible
  - Legacy TCP is best effort based and provides no performance guarantee
- One-fits all protocol or algorithm becomes less feasible, and difficult to integrate the new algorithm into current protocol stacks.
- Complicated and Heterogeneous Network Environments
- Growing concern on network security built on a specific protocol
- Utilize different kinds of protocols for apps in **one** host, or even in **one** app

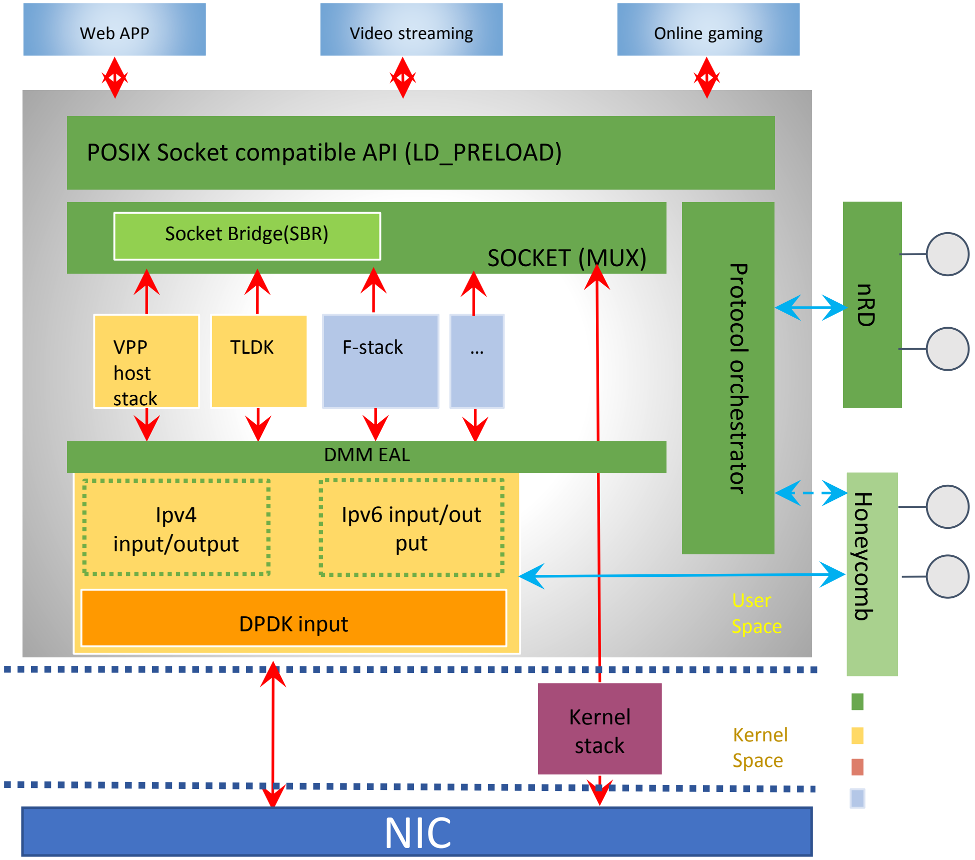
Proposal

We would like to propose DMM frame work to support for multiple protocols and multi-instances on top of VPP that has achieved great success on package forwarding.

We propose a generic framework to support multiple protocols and at the same time is agnostic to applications. This framework including:

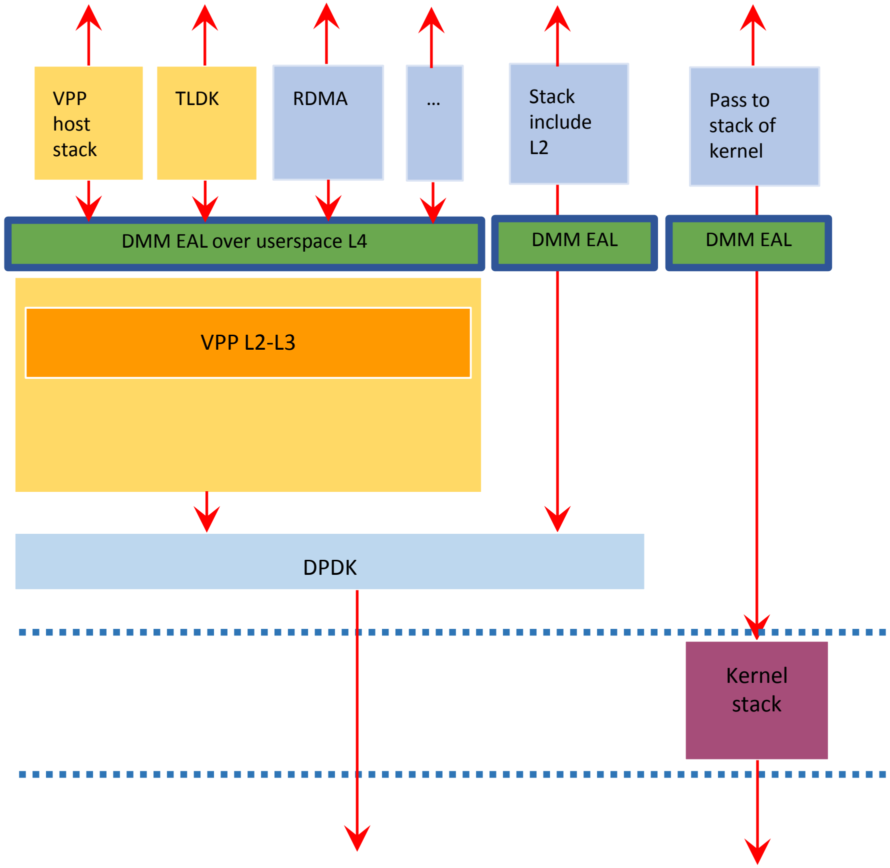
- Protocol orchestrator (L-RD and C-RD)
- Socket bridge layer
- EAL for protocol stack

DMM Framework



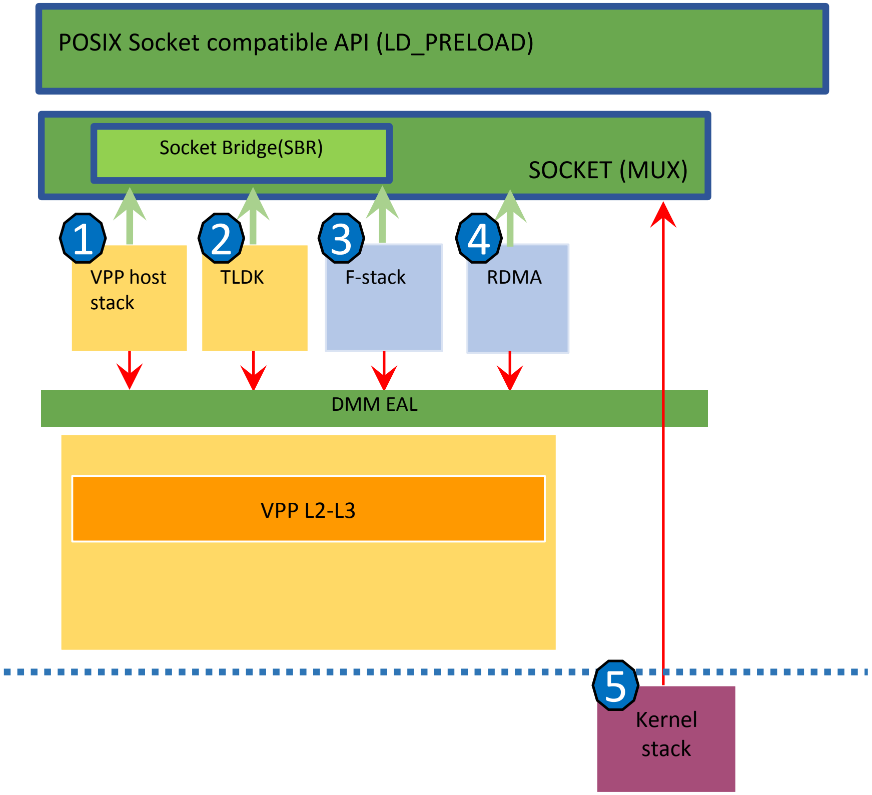
- Existing applications or new applications use Posix compatible and uniform socket API library (via system hijacking or LD_PRELOAD).
- Various stacks integrated in, like vpp hoststack, TLDK, F-stack and etc.
- Based on protocol lookup the corresponding protocol stack is chosen.
- Neuro Resource Discovery learn the set of protocol policies to route the packet. It can be like controller continuously monitor the network state and can update the policy rules dynamically on need basis.

EAL in DMM



- EAL(env abstract layer) in DMM is responsible for isolate the interface of the IO.
- Customed protocol stack will send/receive the pkgs from DMM EAL.

Protocol stacks in DMM



5 Protocol stacks example listed,
1 kernel stack, and 4 userspace stack

- Multi-Protocol, Multi-instance
- SBR will provide a plugin to integrate for any new protocol stack
- VPP hoststack can work as one specific userspace protocol stack.
- People can leverage this socket layer to add their own customized stacks for different scenarios.

Road map & Scope

Step1

- Initial DMM Framework
- Interface document
- Socket plugin mechanism
- Protocol Policy(L-RD part) through CLI
- Dual protocol state choice (kernel&userspace)

Step2

- Protocol stacks inside DMM based on VPP L3
- Integrate vpp hoststack, TLDK
- Support for P2P communication between L-RD’s.
- Performance optimization, reduce the middle layer consumption

Step3

- Multi Instance support.
- Customized protocol algorithm plugin interface
- Support more stacks

Thank you!