**AINFV: Analysis of Isolation (memory/packet) in Network Function Virtualization**

***Abstract:***

***Introduction:***

The starting days of networks, it was used to designed to send packets between two nodes. As the size of network increased, technology evolved and many network services were introduced from time to time i.e. routing, forwarding, security etc. Traditionally these network services were deployed using hardware middleboxes i.e. firewall, routers etc. These traditional hardware middleboxes are in the market for very long time and still serving their purposes. But there are disadvantages of using this approach, such as mentioned in [**Noval approach**] (a)inflexibility: unable to modify the network services, proprietary issue; (b) Non-Scalability: one needs to buy the new middlebox if the load of the network increases for certain period of time and the load stabilizes after some time, then the new middlebox is useless; and (c) Cost: expensive in terms of upgrading the network components by replacing old middleboxes with latest middleboxes to get the maximum throughput. These disadvantages encouraged the ETSI[**ref**] (European Telecommunication Standards Institute) in 2012 and the idea of NFV (Network Function Virtualization) was proposed. The idea was to replace the hardware middleboxes with software defined network services and deploy these network services as VM (Virtual Machine) on commodity servers. ETSI proposed that NFV will help the service providers as (a) swift deployment of network services; (b) comparatively cheap, by using the commodity servers; (c) more flexibility, upgrading of network service is in software. As mentioned in [**panda thesis**]NFV provide the blueprint of developing the network’s dataplane, that allows the developer to program every packet forwarding in the network. Same in SDN (Software Defined Networking), that provides the blueprint of managing the controlplane, i.e. allows developer to define the custom routing, managing network failures etc. NFV framework provides the following features[**panda thesis**]:

**-Multiplexing:** NFV framework should ensure that the NF (Network Functions) should be hardware independent, this helps in scaling of NF without changing the hardware.

**-Isolation:** NF deployed in virtualized share the under the underlying hardware, NFV framework should ensure the memory and packet isolation without affecting the performance

**-High Performance:** NF connected in series working as NF chains should have maximum throughput or equal to as of hardware middleboxes. NFV framework should ensure this throughput, as there is a major overhead of copying packets from one NF to other.

**-Efficiency:** Framework should ensure the minimal hardware utilization as the aim of NFV is to utilize the commodity servers in effective way.

**-Simplify NF Development:** Framework should ensure the simplicity in development of NF, by separating the tasks into two categories i.e. user defined functionality and preprocessing tasks. All of this should be automated.

**-Rapid Deployment:** Framework should ensure the rapid deployment by production ready NFs (i.e. NF testing and deployment in production environment on the go, to improve the performance) . This safes a lot of time.

***Problem statement***

***Available Solutions***

***Background***

***NFV***

***VNF***

***NF Chains***

***Programming VNF***

***Isolation***

***Memory Isolation: Vms/Conatainers***

***Packet Isolation: High performance I/0 libraries for packet processing zero copy Isolation***

***Proposed Framework***

***Overview***

***Main Components***

***Framework***

***Development Model***

***Execution Model***

***Testing***

***Analysis Tool***

***Test-bed***

***Achieved Results***

***Overheads***

***Framework Evaluation***

***Performance analysis of framework based different Nfs***

***Throughput/MPPS***

***Security Analysis***

***NF vs NF Isolation***

***Comparison of Frameworks***

***Based on Packet/memory Isolation***

***Discuss new & old Framework introduced***

***Conclusion drawn***

***Future Work***