**Toward A Software Based Network**

**Austin Carter**

**Computer Networks**

**Scott Kelly**

**4/12/20**

***T****oward a Software Based Network: Integrating Software Defined Networking and Network Function Virtualization* attempts to explain the need for a change in how networks handle traffic to keep up with the rapid advances in both software and virtualization. The authors’ proposed solution for keeping up with advancing technology is a virtual server platform concept called NetVM. Research conducted to assist in explaining the need for these changes is relayed through testing done to both compare and explain the differences between the throughput capabilities of current networking types such as single root I/O virtualization (SR-IOV), to NetVM. Regarding background research done, it’s difficult to determine what pieces of information the authors’ had to research before composing the article, considering the authors’ biographies express strong academic / professional backgrounds in cloud computing, networking, and virtualization. There are three primary sections to this article: an overview of current network architectural elements and their pros / cons, an overview of the proposed NetVM platform and its performance, and a detailed proposal for combining software defined networks with function virtualization to create the NetVM platform.

**W**ithin the first overview section, the authors discuss traditional networking hardware, new advancements in virtualization, and advancements in software based networking. One of the primary points is that currently, networking hardware is often designed for a particular purpose, and can even include ASICs for functions that are normally software based, such as proxies, firewalls, and caches,” due to ASICs offering increased performance in the data plane. However, with businesses becoming more reliant on cloud based services, there have been advances in both virtualization and software based networking that allows for control of network traffic this is far more dynamic than an ASIC setup. Furthermore, the primary need for these new technologies stems from the undeniable fact that the amount of traffic in our networks is only going to increase as time continues. In addition to the total amount of traffic increasing, the traffic is also going to become increasingly more heterogeneous in that the amount of traffic in a network might vary drastically over short periods of time. This said, the previous information helps emphasise the need to shift from having hyper-performing hardware capable of executing tasks under strict conditions to having powerful off-the-shelf hardware capable of having software integrated into it that can be changed / improved if the need arises.

***I***n the second section, the authors discuss their results from testing the NetVM platform alongside a simple software based linux router and an SR-IOV style VM. To reiterate, the primary objective of this plan is to shift from using hardware designed for a specific purpose to using software that allows powerful off-the-shelf hardware to act as a host to any number of virtual machines capable of performing different network functions as necessary. The tests conducted include testing the capability of the setups to keep up with output for increasing inputs and testing the capability of the VM setups to maintain line speed when having to chain packets through multiple VMs on the same host. When looking at the performance of NetVM, the authors show that NetVM is capable of achieving throughput many times greater than that of using a software router in Linux and can manage twice the throughput of an SR-IOV system which has degrading output as input reaches a certain point and even worse degradation if the packets need to be chained through multiple VMs on a host. NetVM is reportedly capable of maintaining a line rate of 10Gb/s even with 60% of packets needing to be chained through multiple VMs on host. However, it seems like the authors propose that this 60% limit could be improved further on a host with more CPU cores.

**T**hroughout the final section, the authors describe in further detail their plan to implement NetVM. Currently, there are three roadblocks preventing this implementation. Firstly, networking packets do not arrive at specified times and are unable to be predicted which results in the operating system having to rely on almost constant interrupts to know when data is ready to be processed. The second issue is related to the input and output in a virtual environment and its increase in overhead from requirements such as needing to use a single network interface controller to handle traffic for multiple virtual machines. The final problem is that most current operating systems store inbound packets into a secure kernel before writing them into space available to the application using the data. Executing these read / write functions requires a large amount of system resources that could otherwise be spent bettering the flow of network traffic . Although the first two issues do not yet have a defined solution, the final issue of excessive reading and writing from kernel space to user space can be solved by shared space, where NetVM will store data that can be accessed by all virtual machines running on the host operating system, rather than write to each individual VM. Another idea that NetVM brings to the table is in having these networking functions, such as proxies and compression, take the form of software that can be placed by service providers only where those functions are necessary along the packet flow, rather than have routing and packet policy put together.   
  
**A**lthough there were a few pieces of this article that I had a difficulty fully understanding, such as the benefits of the “*Service* *Naming*” section and some of the stuff involving the data and control planes (I did some googling on the planes and partially understand the basics of what both planes are, but I do not fully understand how the changes discussed by the authors would affect each them), there were some interesting things that I did learn. As an example, I did not fully understand how capable VM technology was at hosting such a wide array of functionality with the write software. I also learned that there are setups that actually use GPUs to process packets which I thought was pretty interesting considering I would imagine something built for efficiency in processing graphical data wouldn’t be nearly as efficient processing packet data Lastly, I would say the most important thing that I learned / realized was just how important it is going to be to continue improving and adapting the framework that the **Internet of Things** relies on because of the fact that the way we use the IOT is always changing and becoming more dynamic as consumer need continues to do the same.

**References**

1. Wood, Timothy, et al. *Toward a Software-Based Network: Integrating Software Defined Networking and Network Function Virtualization*. IEEE Network, 2015.