An Evolving Architecture for Network Virtualization

Shicong Ma(✉), Baosheng Wang, Xiaozhe Zhang, and Tao Li

College of Computer, National University of Defense Technology,
Changsha 410073, Hunan Province, China
{msc91008,wangbaosheng}126.com, {nudtzhangxz,taoli.nudt}@gmail.com

Abstract. Network virtualization realizes new possibilities for the evolution way to future network by allowing multiple virtual networks over a shared physical infrastructure. In this paper, we discuss the shortcoming of current network virtualization and propose our approach, a framework that improves current infrastructure by extending link virtualization with a new component which we call Multi-Hop Virtual Link. At last, we present preliminary design of our proposal.

Keywords: Network Virtualization · Virtual Link · Router Virtualization Architecture

1 Introduction

In a past three decades, Internet has become critical infrastructure for supporting multitude distributed systems, applications, and a widely various networking technology. Just as many successful technologies, Internet has been suffering the adverse effects of inertia [1], and recent efforts show that it is hard to design a one-fit-all network architecture [19]. Network virtualization is considered to be an effective way to fend off this problem [2]. With network virtualization, multiple isolated virtual networks with potentially different routing algorithms, network protocols and data process can share the same physical infrastructure [3].

Network virtualization decouples network functionalities from those physical realization by separating the role of the traditional Internet Service Providers (ISPs) into two parts: infrastructure providers (InPs), who manage the physical network infrastructure, and virtual network operators (VNOs), who create virtual networks by aggregating resources form more than one InP and offer network services based on users needs[5].

In a network virtualization scenario, InPs should maintain a network environment supporting network virtualization which must allow the coexistence of multiple virtual networks with different architectures and protocols over a shared physical infrastructure. Thus, the deployment of network virtualization introduces new requirements in relation to what the architecture of network infrastructure and how virtual network are provisioned, managed and controlled under

V.C.M. Leung et al. (Eds.): TridentCom 2014, LNICST 137, pp. 379–386, 2014.
DOI: 10.1007/978-3-319-13326-3_36

the condition of virtualization. In conventional network virtualization architecture, the physical network infrastructure is divided into a plurality of slices that are assigned to different users to guarantee resource and support multiple virtual networks [19].

Unfortunately, current networking infrastructure cannot fully meet any of these goals, which causes significant ossification for network operators who want to provide network virtualization services for end-to end users or applications[9]. To fend off this inability, network research communities have made lots of meaningful works to design a framework of infrastructure which can satisfy the needs of network virtualization[4].

In this paper, we begin with revisiting the previous conception of network virtualization and discuss limitations and explain how current works would fall short of our goals. We then explore how we might improve current network virtualization architecture. Roughly, our idea is to be dubbed as two improvements for the existing work, including, i) extend current abstraction mechanism of a network with a more detailed link virtualization abstraction; ii) present a preliminary design of a network virtualization platform, describing how to leverage virtualization primitives in current general-propose hardware. Our modified approaches introduce a new modularity in network infrastructure which we think is necessary to achieve our goals.

We begin this paper at Section 2 by reviewing the basics of current network technologies. We then, in Section 3, introduce our proposal and preliminary design thought. We end with a discussion of the implications of this approach in section 4.

2 Background of Network Virtualization

As shown in Figure 1, previous researches have proposed conventional virtual network architecture running on the sharing infrastructure which must consist of two basic components, virtual node and virtual link [1]. In a networking environment that is capable of supporting network virtualization, each underlying node should be able to contain multiple virtual nodes and bearers more one virtual links[12]. By far, most use cases demand a one-to-one mapping between each virtual node of a virtual network and a physical node. On the other hand, a virtual link is corresponding with a physical path which may consist of more than two physical nodes.

In a common view, virtual node is often considered as virtual router, and data plane virtualization is the basic component of router virtualization[4]. Thus, earlier researches on node virtualization focus on isolation, reconfiguration and partitioning of underlying hardware resources, such as CPU, memory, IO bandwidth, through recent developments in para-virtualization (such as XEN) and operating system virtualization (e.g. OpenVZ). Link virtualization is another central component of network virtualization. Compared with node virtualization, link virtualization is used for forming a virtual network through interconnecting multiple virtual nodes according to the topology of the virtual network. From

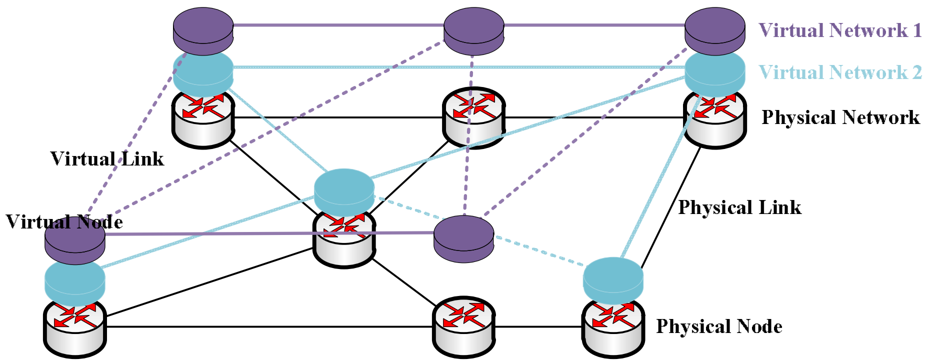

Fig. 1. Basic components of network virtualization

Figure 1, it is easy to see that link virtualization need realize a mapping virtual links between a physical link and guarantee link bandwidth of each virtual link. Particularly, Due to the difference between physical network topology and virtual network topology, there may be some virtual links that traverse more than two physical nodes (in this paper, we call it **multi-hop virtual link**. Due to the existing of multi-hop virtual links, each physical node in a virtual network must play one of two roles, one is one of the two nodes who terminate the virtual link, we call it **Link End Node**; the other is intermediate node which is traversed by the virtual link, we call it **Link Intermediate Node**. At the aspect of implementing link virtualization, some researches argue that time-division multiplexing (TDM) and wavelength division multiplexing (WDM) can be used for isolation and partitioning of link bandwidth and todays common network technology (e.g. MPLS, ATM) can be suitable for building a virtual link.

3 Network Virtualization Extending Design

3.1 Requirements Analysis

While previous works have permitted significant advances in terms of fairness and preference, we consider that there is still room for improvement through recent advances in other domains. In this section we will explore how our proposal in network virtualization framework might be extended to better meet the goals listed in the introduction. There are relevant improvements we consider here:

First, previous studies do not pay enough attention for multi-hop virtual links. Packet processing of multi-hop links on Link Intermediate Node only requires switch hardware to lookups over a small table consisted of multi-hop virtual link tags. Therefore, it unnecessarily couples the whole requirements on Link End Node to packet processing on Link Intermediate Node.

Second, while node virtualization based on operating system virtualization has permitted significant advances in terms of fairness and isolation, but this may introduce additional overhead, especially in support of forwarding plane virtualization with IO-bound feature. Moreover, the environment of forwarding plane virtualization does not require so strict isolation as operating system virtualization. We argue that it is necessary to implement a lightweight forwarding plane virtualization hypervisor without operating system virtualization.

Thus, our proposal is to extend network virtualization architecture in a way that improves some aspects yet still retains current benefits. Then, we will present a preliminary design and prototype implementation of our proposal, describing how to leverage existing virtualization primitives in todays general hardware to achieve ideal network virtualization.

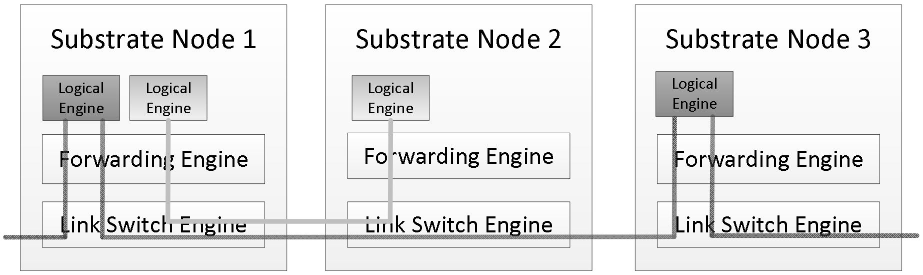

Fig. 2. Proposed Physical Node Model

3.2 Architecture Overview

In this section we explore how the Network Virtualization architectural framework might be extended to better meet the goals listed in the introduction. Our proposal involves two changes for physical nodes in network virtualization environment, which is based on our discussions last section.

Figure 2 shows an example of the proposed architecture. Compared to existing network virtualization, we introduce a new component in physical node supporting network virtualization, which we call Link Switch Engine. Link Switch Engine carries out a simple and fast packet switch process like MPLS, when the physical node acts as a Link Intermediate Node. In this way, we separate packet process of Link End Node and Link Intermediate Node into two separate processes and guarantee individual packet processing ability for multi-hop virtual links. A seemingly natural choice to implement multi-hop virtual link would to refer to simplified hardware, such as OpenFlow or MPLS. For the implementation of Link End Node, there are many solutions based on general-purpose server ,programmable hardware[17], such as FPGA[11][14][16] and additional accelerator, e.g.GPU[15]. Other researchers use IO optimization mechanism[7]

In order to achieve reconfiguration, flexibility and isolation, Link End Nodes should host instances of logical engines which carry out customized forwarding processing of packets that flow through them and require isolation among logical instances. Recent promising technologies proposed by Intel show that software data plane using general-purpose high-preference processors may be a replacement of traditional hardware-based data plane, since it is able to support rates of 10 Gbps[9]. By these results, forwarding hardware could get more improvements in terms of reconfiguration and flexibility. As for isolation, direct mapping each logical instance on an individual core could avoid additional overhead introduced by operating system virtualization [10], which we plan to advance packet processing in Link End Node in future.

3.3 Multi-hop Virtual Link Service Model

Under our modified model, packet processing on a multi-hop virtual link can be represented as a process of packet switch rather than packet forwarding, since it does not need intelligence, just a relatively dumb, but very simple, fast fabric, which we call Link Switch Engine.Even if someone believes that in-network processing will introduce more and more complexity and underlying infrastructure needs to become more flexible, it does not mean that packet processing in a Link Intermediate Node of a multi-hop virtual link and it only need a minimal set of switch primitives without internal forwarding processing. The design of the additional component in network device for Link Switch Engine is a reasonably good analogy for a network virtualization architecture that includes multi-hop virtual links. In a network virtualization environment with multi-hop virtual links, Link End Node will implement network policies and finish encapsulating and decapsulating virtual link tags.

The complexity in the environment lies in mapping the Link End Nodes and Link Intermediate Nodes to physical nodes according requirement of the virtual network for network resources and topology. By the processing of mapping, we embed a virtual network into a given network. There are also four primary mechanisms for this:

Identifying a Multi-hop Virtual Link. As we have described above, packet processing in a Link Intermediate Node differs from in a Link End Node, which is not required to use source or destination addresses for switching. Thus, one option would be to use the identification of the virtual network, since a virtual network only has a multi-hop virtual link instance in a physical node.

Lookup Tables. A Link Intermediate Node will maintain a small table consisted of all multi-hop virtual links through it, and Link Switch Engine delivers a packet to its output interface based on this table. Our proposal is designed around MPLS so we adopt a simple and generalized table structure which can be built around a pipeline of hardware.

Packet Processing in Link Switch Engine. In our proposed architecture, every packet will be identified by virtual network ID. Received packet is firstly checked by Link Switch, judging whether the packet should be processed by it. If not, then, the packet will be delivered to packet forwarding engine. If yes, Link Switch Engine looks up its forwarding table ,make switch decision for the received packet and deliver it to corresponding outport.

Switch Ability Isolation. In a network virtualization environment, isolation of processing resource is a key question for resources guarantee for each virtual network[13]. Therefore, in our proposal, a multi-hop virtual link may be implemented as a tunnel which traverses multi-hop physical nodes. Thus, if a physical node bearer more than one virtual link, Link Switch Engine would be required for strong isolation of switch ability offered to different virtual links.

Beside what we listed above, there are also many practical challenges in implementing such a mechanism we will not cover in this paper. These include control architecture of multi-hop virtual links, the detailed definition of the virtual link tag, and the virtual link tags distributing protocol .etc. We aim to address these challenges in our next phase.

3.4 Feasibility Analysis

There are a multitude of approaches which would be suitable for implementing proposed design presented in the previous section. In what follows, we describe our implementation which we would build as a prototype. While we have proposed the design thought of our implement. Then, we present some of practical issues we have considered in our system.

Multi-hop Virtual Link Switch Engine. In our design, we plan to use a changed MPLS to provide fast switching for multi-hop virtual links, and try to realize it on NetMagic[8], a programmable platform deployed for network innovation. NetMagic can offer multiple simultaneous hardware resources for each multi-hop virtual link and easily reconstructed.

Isolation among Multiple Virtual Links. In order to provide switch ability isolation for multiple virtual links on Link Switch Engine, we prefer to allocate separate hardware to each virtual link, which can facilitate strong isolation among multiple virtual links. For a further work, we are ready to propose a simple but effective scheduling algorithm to enhance isolation among multiple virtual links.

4 Discussion

The approach proposed in this article is conceptually quite simple: rather than requiring forwarding planes to support Link End Node and Link Intermediate Node at the same time, our proposal depends on mapping each of them on an individual element. By separating packet processing element into Packet Forwarding Engine and Link Switch Engine, there will be significant benefits for network infrastructure designed for supporting network virtualization as follow:

Two Free-Running Packet Processing. Independent Link Switch Engines may relieve contradiction between requirements for flexible functions and high preference by mapping the two targets on a flexible packet processing engine and a simple but fast Link Switch Engine. Link Switch Engine only requires the minimal set of packet switch and is suitable for a hardware-based appliance. And flexible packet processing engine could be realized by general-purpose multi-core processors. Based on optimizations on packet I/O and buffers, preference of software packet processing may catch or even exceed hardware-based appliance. Thus, network devices designed based on our proposal will be easy to build, operate and accommodate future innovation.

More Flexible Virtual Network Mapping Model. In traditional network virtualization architecture, the packet processing capacity of a physical node is always described by packet forwarding ability, and in this mapping model, a multi-hop virtual require all Link Intermediate Nodes on its corresponding physical path with forwarding capacity which is equal to its two Link End Nodes. However, in our proposal, by introducing description of Multi-hop Virtual Link and Link Switch Engines, we can get a more flexible virtual network mapping model. Separation of packet processing changes the ability description of physical node into Link Switch capacity and Packet Forwarding capacity. By this approach, our proposal can provide more choices for virtual network mapping.

5 Conclusion

In this paper, we discuss the shortcoming of current network virtualization architecture and give a more detailed description for network virtualization by describing multi-hop virtual link ,a virtual link pass through more than two physical routers. Based on what our description, we propose out extended network virtualization architecture and represent a preliminary design. In order to support multi-hop virtual links, we introduce a new packet component called "Link Switch Engine" used to bear packet switch in a Link Intermediate Node of a multi-hop virtual link. At last, we give our realization idea of our proposal. Although we have discussed a lot of things of our proposal, we clearly know that it is more complex than we consider to give a complete description and realization idea. This approach merits further investigation, as we have only begun to scratch the surface of this idea.

Acknowledgments. Our work is supported by Program for National Basic Research Program of China(973 Program) "TestBed of Flexible Architecture of Reconfigurable Infrastructure", Specialized Research Fund for the Doctoral Program of Higher Education of China(20114307110006)"Research on Technology of Network Quality of Service based on Network Virtualization", and Research on Trustworthy Cross-Domian Information Exchanging Technology Based M-TCM KJ-12-07.

References

1. Carapinha, J., Jimnez, J.: Network virtualization: a view from the bottom. In: Proceedings of the 1st ACM Workshop on Virtualized Infrastructure Systems and Architectures. ACM (2009)
2. Chowdhury, N.M., Boutaba, R.: A survey of network virtualization. Computer Networks 54(5), 862–876 (2010)
3. Anderson, T., et al.: Overcoming the Internet impasse through virtualization. Computer 38(4), 34–41 (2005)
4. Casado, M., Koponen, T., Ramanathan, R., et al.: Virtualizing the network forwarding plane. In: Proceedings of the Workshop on Programmable Routers for Extensible Services of Tomorrow. ACM, p. 8 (2010)
5. Schaffrath, G., Werle, C., Papadimitriou, P., et al.: Network virtualization architecture: proposal and initial prototype. In: Proceedings of the 1st ACM Workshop on Virtualized Infrastructure Systems and Architectures, pp. 63–72. ACM (2009)
6. http://www.edac.org/downloads/resources/profitability/HandelJonesReport.pdf
7. http://www.dpdk.org
8. www.netmagic.org
9. Costa, P., Migliavacca, M., Pietzuch, P., et al.: NaaS: Network-as-a-Service in the Cloud. In: Proceedings of the 2nd USENIX Conference on Hot Topics in Management of Internet. Cloud, and Enterprise Networks and Services, Hot-ICE, vol. 12, p. 1 (2012)
10. Egi, N., Iannaccone, G., Manesh, M., et al.: Improved parallelism and scheduling in multi-core software routers. The Journal of Supercomputing **63**(1), 294–322 (2013)
11. Unnikrishnan, D., Vadlamani, R., Liao, Y., et al.: Scalable network virtualization using FPGAs. In: Proceedings of the 18th Annual ACM/SIGDA International Symposium on Field Programmable Gate Arrays, pp. 219–228. ACM (2010)
12. Bhatia, S., Motiwala, M., Muhlbauer, W., et al.: Hosting virtual networks on commodity hardware. Georgia Tech. University., Tech. Rep. GT-CS-07-10 (2008)
13. Wu, Q., Shanbhag, S., Wolf, T.: Fair multithreading on packet processors for scalable network virtualization. In: ACM/IEEE Symposium on Architectures for Networking and Communications Systems (ANCS), pp. 1–11. IEEE (2010)
14. Anwer, M.B., Motiwala, M., Tariq, M., et al.: Switchblade: a platform for rapid deployment of network protocols on programmable hardware. ACM SIGCOMM Computer Communication Review **41**(4), 183–194 (2011)
15. Han, S., Jang, K., Park, K.S., et al.: PacketShader: a GPU-accelerated software router. ACM SIGCOMM Computer Communication Review **41**(4), 195–206 (2011)
16. Xie, G., He, P., Guan, H., et al.: PEARL: a programmable virtual router platform. IEEE Communications Magazine **49**(7), 71–77 (2011)
17. Shimonishi, H., Ishii, S.: Virtualized network infrastructure using OpenFlow. In: 2010 IEEE/IFIP Network Operations and Management Symposium Workshops (NOMS Wksps), pp. 74–79. IEEE (2010)
18. Martins, J., Ahmed, M., Raiciu, C., et al.: Enabling fast, dynamic network processing with clicko. In: Proceedings of the Second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking. ACM, pp. 67–72 (2013)
19. Turner, J.S., Taylor, D.E.: Diversifying the internet. In: Global Telecommunications Conference, GLOBECOM 2005, vol. 2(6), p. 760. IEEE (2005)