Enhancing Data Center Agility with Network Virtualization Overlays

Executive Summary

Software Defined Infrastructure (SDI) primarily defines IT infrastructure managed by software to deliver unprecedented agility and responsiveness to business.

Transitioning to Software Defined Infrastructure is a long-term process, and Network Virtualization Overlays (NVO) can provide important abstraction and virtualization functions to support Software Defined Networking (SDN), which is a core component of Software Defined Infrastructure.

"VLANs are now giving way to Network Virtualization Overlay (NVO) technology to deliver network agility for data centers."

– Dawn Moore
GM Networking Division, Intel

Networking and Software Defined Infrastructure

In an SDI environment, business defines application and operational policies while orchestration software provisions and configures infrastructure according to defined business needs. Services are monitored and resources automatically allocated according to defined policies.

The capability of orchestration and management software, combined with the extensive software control of resources, provides a foundation for significantly reduced operational costs, increased utilization of hardware resources, and improved agility.

Transitioning to a Software Defined Infrastructure is a long-term process, not a short- to medium-term infrastructure project. It is necessary to address Compute – Software Defined Compute (SDC), Network – Software Defined Networking (SDN) and Storage – Software Defined Storage (SDS) infrastructure to ensure alignment and availability to function within a software controlled environment.

At the core of all of these infrastructure changes is the drive to abstract hardware configuration complexities and shift control to intelligent software implementations that can be used for infrastructure provisioning and management.

NVOs can provide abstraction and virtualization functions to support SDN. There are a number of network protocols and extensions to consider for network infrastructure planning and upgrade purposes.

Table of Contents

LAN Network Evolution

Ethernet has become the primary means of moving data in the LAN environment. Reviewing the standards produced by the IEEE 802.3 working group shows a consistent increase in the Ethernet speeds from 100MbE in 1995 to 1GbE in 1998, 10GbE in 2002 and 40GbE, and 100GbE in 2010.

The available Ethernet bandwidth and corresponding networking components, including the range of Ethernet Converged Network and Server Adapters from many vendors supporting 10GbE, 25GbE, 40GbE, and 100GbE connections, provides a dependable foundation for planning for exceptional LAN bandwidth demands.

Today, many switched Ethernet network use Layer 2 VLANS as a means of segmenting the network. Layer 2 VLANs were originally implemented based on proprietary switch port allocation and progressed to supporting 802.1Q VLAN tagging as part of the IEEE 802.3ac standard in 1998.

This included the insertion of VLAN Identifier in the Ethernet Frame that would be used by the Ethernet switch when forwarding to other devices within the same VLAN. VLAN tagging, as it has become known, became a prominent means of segmenting Ethernet LAN environments as its adoption increased from 2005 onwards.

Layer 2 Switched Ethernet environments generally implement the IEEE 802.1D (Functions now incorporated in IEEE802.1Q) Spanning Tree Protocol to support redundant LAN links in the data center where network loops must be avoided.

However, there is a challenge in the way many LANs are segmented in current Ethernet networks.

Server Virtualization

Server virtualization is essential in most data center implementations. This places demand on what can be now considered traditional network infrastructure implementations.

- **Extending LAN Boundaries**
  Geographical distance between physical and required virtual server mobility between these locations demands the extension LAN boundaries. Migrating VMs or supporting High Availability VM Clusters across geographically separate data centers, can require significant work-arounds based on dependence on Layer 2 networking with the challenge of extending LAN boundaries within the data center.
- **Rapid Service Deployment**
  Server virtualization enables rapid service deployment. However, legacy Level 2 VLAN implementations may impede deployment flexibility and speed based on network planning and reconfiguration times.
- **Tenant Isolation**
  Cloud computing services that support multiple tenants within the same data center are growing rapidly. In large, multi-tenant environments, the number of VLANs required to effectively isolate tenants and services may exceed the 4096 VLAN identifiers supported in a Layer 2 VLAN environment.
- **Spanning Tree Protocol**
  Spanning Tree Protocol (STP) is used extensively to support LAN link redundancy, but when combined with VLAN implementations and assessed in the context of current data center scale, there are potential L2 switch related constraints.

  In a Layer 2 network, inter-switch connections are generally configured as VLAN trunks to transport traffic for multiple VLANS between switches. Where a VLAN intersects with a physical switch port, a logical port is defined. A trunk with 250 associated VLANS has 250 logical ports.

There is a physical limitation on the number of logical ports supported and this may vary across switching hardware.

When STP is evaluated in the context of a large data center environment, the spanning tree implementation may not operate within required stability and convergence requirements.

- **MAC Address Tables**
  Virtualized server infrastructure can increase the ratio of MAC addresses per switch port dramatically. A physical server with four NIC ports, and hardware management enabled would typically create five MAC addresses.

  This number will increase in a virtualized environment many Virtual NICS (VNICs) may be used for hypervisor kernel functions, virtual switches and VMs on the server.

  With current server compute and storage capabilities, supporting 16 VMs on a host is quite feasible. If each server is assigned two VNICs, that would result in an additional 32 MAC addresses. Add up to four additional MAC addresses for the hypervisor kernel and four for virtual switching and that can add an additional 40 MAC addresses for that physical server instance.

  If this single server instance is scaled out to a large data center environment, its feasible that many current L2 switches may not support the number of MAC entries in their MAC tables.

Network Virtualization Overlays

Network virtualization using encapsulation protocols is not new. Virtual Private Networking (VPN) using Point-to-Point Tunneling Protocol (PPTP) and Layer Two Tunneling Protocol (L2TP) have long been used for secure point-to-point tunnels across various intermediary networks.

Typically, a packet intended for transmission from a network device would be encapsulated as a data payload in an additional packet that includes a header specific to the protocol implementation and additional headers required depending on the underlying network transport protocols being used. The payload is transported to a defined end-point where the encapsulation data is stripped and the original packet is forwarded to the intended node.

While encapsulation as a concept may be simple enough to understand and many types of protocols exist to support encapsulation, it is essential to understand the network traffic being encapsulated and corresponding demands placed on hardware and software infrastructure to support effective virtualization through encapsulation.

A Layer 2 Switched Ethernet environment has specific frame size, broadcast, multicast, forwarding, and MAC address learning requirements that can introduce significant bandwidth and compute overhead in large, virtualized environments.

The network communication between virtual hosts may be virtualized, but the physical transport requirements remain an essential component of a successful virtual network implementation.

Network Virtualization Considerations

To be effective, the Network Virtualization Overlay should:

- Support the logical segmentation of LANs in a manner that provides equivalent or more effective network segment isolation including tenant traffic and address space isolation between each tenant, and overlay network address space isolation from tenant address space
- Support a significantly higher number of Virtual LAN segments
- Function within the Ethernet and IP network construct for implementation in existing environments
- Promote a scalable, flexible, and manageable network infrastructure capable of providing the network throughput required
- Be fully standardized as a protocol to support hardware and software vendor interoperability
- Support efficient implementation in hardware to support efficient packet processing and potential participation with the encapsulation protocol being implemented.

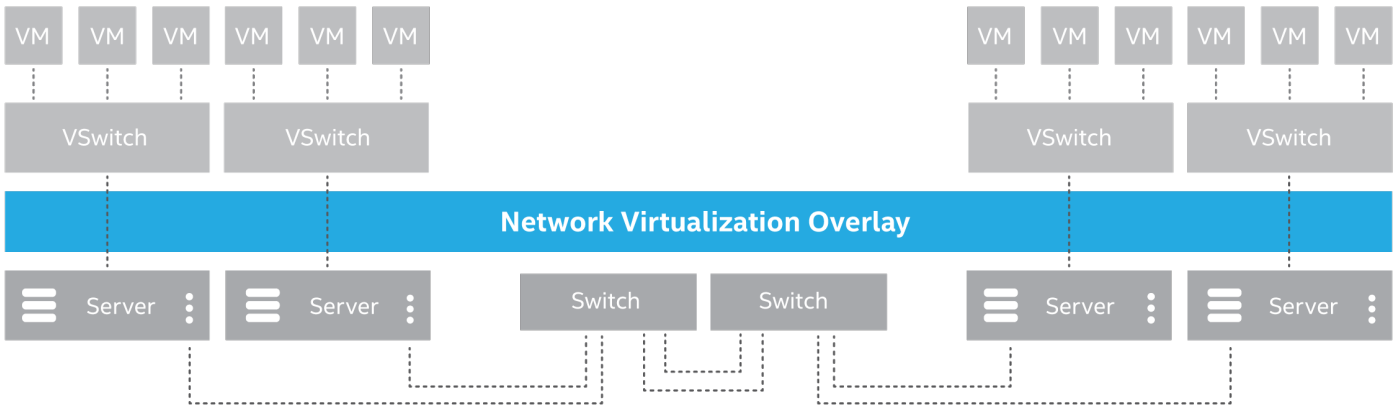

Figure 1. Network Virtualization Overlay

Virtualization may abstract specific hardware and software interactions, but it does not remove dependence on underlying hardware to support compute, store, and transmit functions.

Similar considerations apply to server and network virtualization technologies from a hardware design and implementation planning perspective including:

- Specific hardware and software interactions being abstracted
- Associated increase in compute function to support the abstraction
- Distribution of compute tasks between components
- Specific technologies designed to improve efficiencies for added compute function

From a server virtualization perspective, the underlying server hardware will dictate the scope of hardware resource that a host can support.

In addition, implementing hardware that incorporates technologies to improve virtualization can have a significant impact on hardware resource utilization.

Intel has a credible record of developing technologies that have a significant positive impact on the way server compute resource is utilized.

As an example, I/O virtualization features were developed to facilitate offloading o multi-core packet processing to network adapters as well as direct assignment of virtual machines to virtual functions including disk I/O.

These include Virtual Machine Device Queues (VMDq), Single Root I/O Virtualization (SR-IOV) and Intel® Data Direct I/O Enhancements (Intel® DDIO).

The same causal relationship between hardware and software for facilitation of efficient and effective virtualization applies to Network Virtualization Overlays.

Using encapsulation to enable Layer 2 network function over a Level 3, IP data plane adds processing and packet overhead to the underlying server and network hardware.

The choice of Network Virtualization Overlay protocol combined with the appropriate underlying network and server hardware can significantly enable or impede the realization of a flexible, scalable network that meets business demands.

Network Virtualization Protocols

When evaluating a protocol for implementation purposes, it is essential that it is specified as a standard to enable hardware and software interoperability between different vendor products.

Once that is established, it is important to evaluate the protocol in terms of function, efficiency, and device support.

The Network Virtualization Overlays (NVO3) Working Group has the charter to develop protocols and/or protocol extensions specifically for data centers with underlying IP networks.

The protocols listed below fall within the scope of NVO3.

Network Virtualization using Generic Routing Encapsulation (NVGRE)

NVGRE provides a network overlay solution for virtualizing Layer 2 networks over IP infrastructure. It provides a method for tunneling Ethernet Frames in IP in GRE using a 24 bit Tenant Network Identifier (TNI) that represents a logical network. The tunneling mechanism is stateless but issues such as IP fragmentation may need to be addressed in implementation.

An NVGRE endpoint functions as a gateway between the logical and physical network. This endpoint performs the de-encapsulation and encapsulation of Ethernet frames. GRE is a widely implemented protocol that should be widely supported with most current networking hardware. However, the increase in the payload additional payload header may require increasing MTU sizes between endpoint devices to prevent packet fragmentation.

NVGRE does not use TCP or UDP for transport which prevents the implementation of TCP/UDP load balancing of flows between networks.

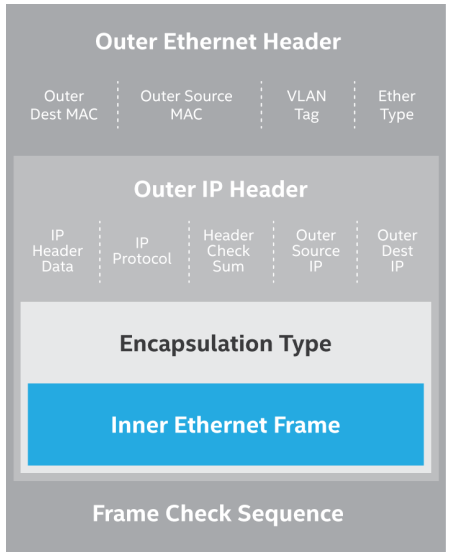

Figure 2. Encapsulation Concept

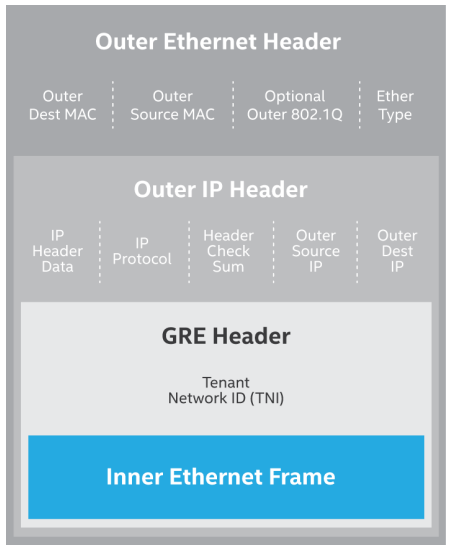

Figure 3: NVGRE Frame Structure

Virtual Extensible LAN (VXLAN)

VXLAN provides the framework for overlaying virtualized Layer 2 networks over Layer 3 networks.

It comprises a stateless tunneling protocol that wraps layer 2 packets in a Layer 3 packet and adding a VXLAN header that includes a Virtual Network Identifier (VNI) for a specific virtual LAN assignment.

At this point it is important to introduce the Network Virtualization Edge (NVE) which resides at the endpoint of a virtual segment and is located in the hypervisor or hardware/ software switch.

The NVE performs all encapsulation/ de-capsulation and is responsible for the VM MAC to VNI association.

An NVE has two external interfaces:

- **Tenant System Facing**
  On the Tenant System facing side, an NVE interacts with the hypervisor (or equivalent entity) to provide the NVO3 service.
- **Data Center Network Facing**
  On the data center network facing side, an NVE interfaces with the data center underlay network, sending and receiving tunneled TS packets to and from the underlay.

While NVO technologies like VXLAN can achieves these goals, it can come at the cost of significant performance penalties introduced by the VXLAN encapsulation/de-capsulation process done by a software NVE, and/or from the loss of existing hardware accelera-tions and offloads. This is addressed later in this paper.

It is important to note further developments with the VXLAN protocol.

Generic Network Virtualization Encapsulation (GENEVE) and the Generic Protocol Extension for VXLAN (VXLAN-GPE) are intended to extend the VXLAN capabilities to support multiprotocol encapsulation, operations, administration and management (OAM) signaling and explicit versioning via changes to the VXLAN header.

This change is significant as it provides native support for Network Service Header (NSH) implementations using VXLAN-GPE as the tunnel transport.

Generic Network Virtualization Encapsulation (GENEVE)

GENEVE provides a framework for data-plane structure that only defines the encapsulation data format. It does not define control-plane functions and hardware support is limited.

Part of its stated intention is to provide a framework for tunneling to support network virtualization without being prescriptive about the whole system within which it functions.

This level of flexibility may be considered beneficial for some implementations, however, the lack of defined control plane functions may inhibit broader adoption in current environments.

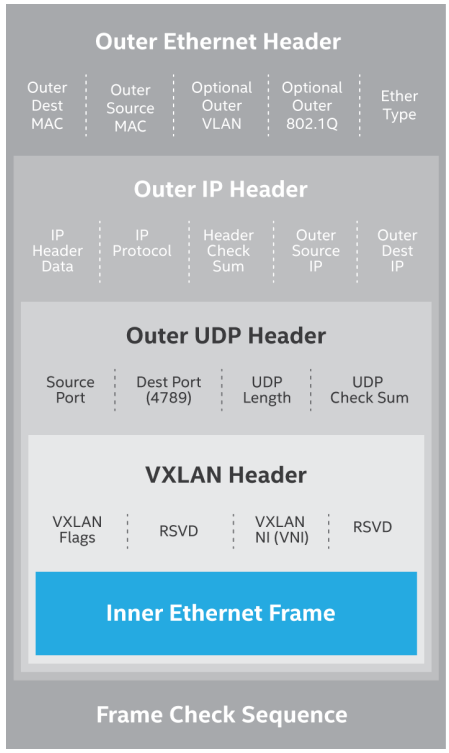

Figure 4: VXLAN Frame Structure

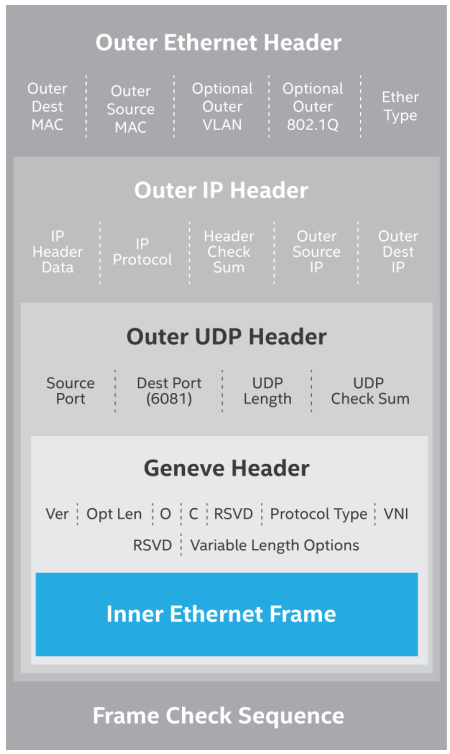

Figure 5: Geneve Frame Structure

Generic Protocol Extension for Virtual Extensible LAN (VXLAN-GPE)

VXLAN-GPE is an emerging protocol that adds a protocol type field within a VXLAN header. This is intended to support functions such as Network Service Chaining that requires specific classifiers in the packet being forwarded through the network.

VXLAN-GPE is backwards compatible with VXLAN, but VXLAN is not forwards compatible with VXLAN-GPE. While both protocols can use the same ports, it is recommended that different ports are used in mixed environments.

Network Service Header (NSH)

NSH is a data-plane protocol intended to support the growing evolution towards Network Service Chaining.

It is added as an additional header to a packet and contains a Service Classifier that is used by a Service Function Forwarder (SFF) to forward the packet according to prescribed policies.

Intel® Technologies for NVO

Intel has been implementing various acceleration technologies for Network Virtualization Overlays starting with the Intel® Ethernet Controller X540 and Intel® 82599 Ethernet Controller Family, with further enhancements in the Intel® Ethernet Controller XL710 and X550 for its family of Intel® Ethernet Converged Network Adapter XL710 for 10/40 GbE network-interface cards. Choosing the appropriate Intel Ethernet Product can have a significant impact on the distribution of NVO related compute overhead between processor and network adapter resources as well as the network throughput realized.

The Intel® Ethernet Converged Network Adapters X520 and X540 incorporate acceleration for VXLAN traffic implemented as Receive Side Scaling (RSS) for VXLAN traffic. When configured correctly with the correct software drivers, it spreads traffic over multiple CPU cores thereby reducing processing delays and possibly related cache efficiencies.

The Intel® Ethernet Converged Network Adapter XL710 and X710 incorporate additional accelerations with inner-header Stateless Offloads and advanced traffic steering and CPU core alignment with Intel® Ethernet Flow Director or Receive Side Scaling (RSS) based on the inner-header for NVO traffic. When configured correctly in supported Linux* kernels with the correct software drivers installed, computation tasks associated with tunnel packet processing in an IP network that would normally consume CPU resource, are processed by the network adapter. In many cases, this has a direct impact on increasing network throughput and reducing associated CPU overhead in high bandwidth environments. If a further CPU utilization or increased network throughput is required, the NVE encapsulation process can be moved into the network adapter hardware.

This method does require much higher levels of coordination between the hypervisor, virtual switch, and network adapter but it also has shown the ability to achieve close to wire speed throughput with minimal CPU overhead.

Network Virtualization Overlays use encapsulation to create tunnels for data flows within the network and many are tightly coupled with Ethernet networking. The additional compute overhead associated with the processing of tunnel packets can consume additional processor and memory resources.

Ethernet network adapters can therefore fulfill a critical function in supporting NVO protocols by accelerating associated tunnel packet processing.

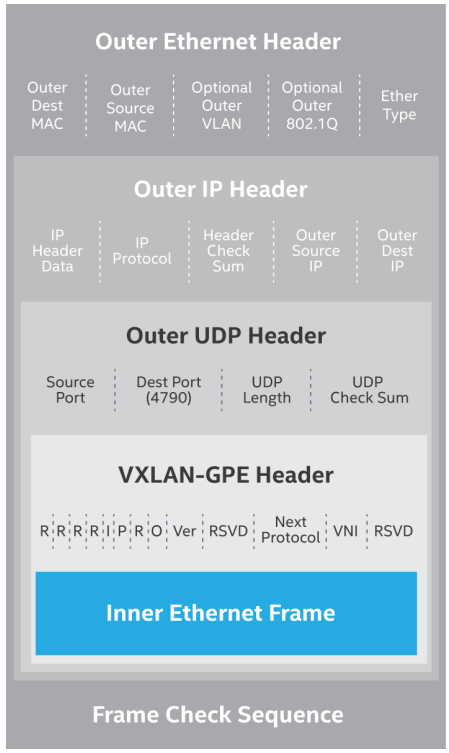

Figure 6: VXLAN-GPE Frame Structure

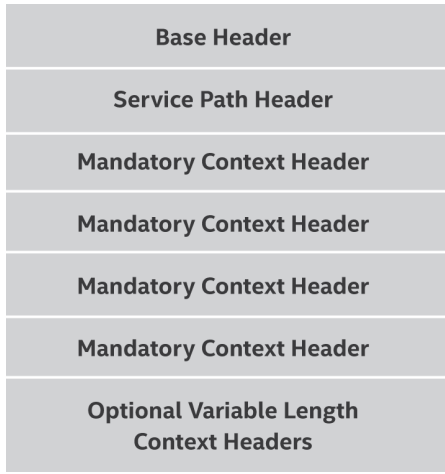

Figure 7: Network Service Header

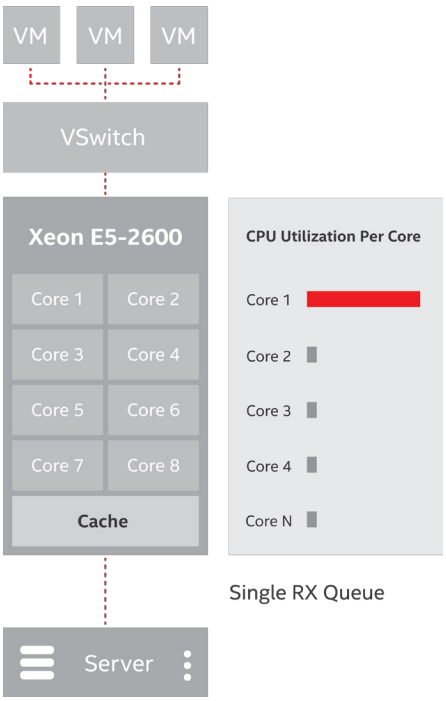

Figure 8: VXLAN Network Virtualization Optimizations without Receive Side Scaling

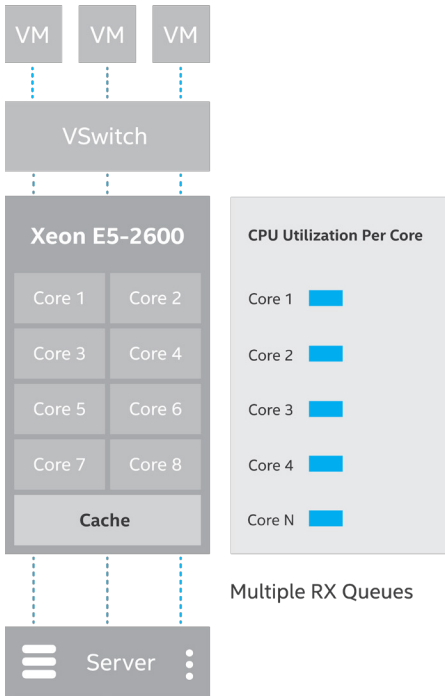

Figure 9: VXLAN Network Virtualization Optimizations With Receive Side Scaling

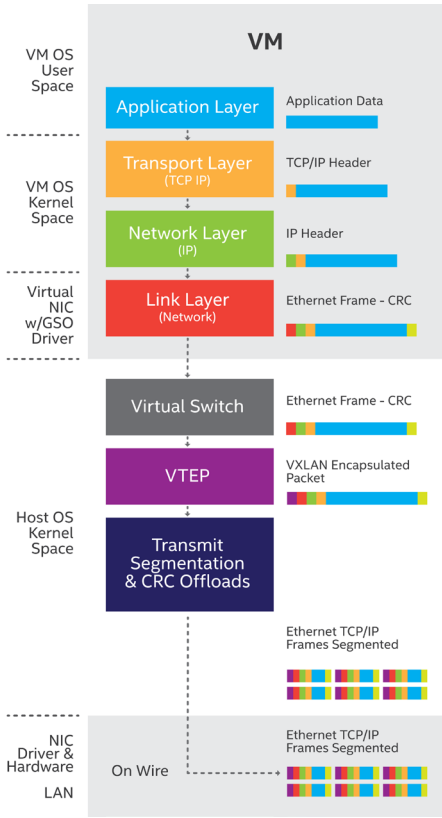

Figure 10: Software-Based Segmentation

Recommended Network Adapter Feature for Software-based NVE

The shift to Network Virtualization Overlays to support Software Defined Networking, network infrastructures require solutions with the flexibility to scale across multiple deployment models while also providing exceptional performance. With a single architecture for application, control, and data plane workloads, Intel® architecture-based platforms simplify application development and decrease time to market without sacrificing performance. Ingredients like the Data Plane Development Kit (DPDK), Intel® QuickAssist Technology, and HyperScan optimize throughput for SSL, IPsec, compression, and DPI workloads while also enabling maximum scalability across physical and virtual deployments. Intel® platforms also streamline development and testing, reducing cost and time-to-market for network equipment providers and operators, and helping provide faster and better services to end users. It is essential that network infrastructure consistently meets bandwidth demands without placing too much overhead on server compute resource. Network adapters should not be viewed solely as hardware components that provide a physical connection to the network.

The right network adapter can improve work throughput while reducing use of server compute resource. The recommended network adapter features for Software-based NVE include:

- Stateless offload support for tunneled traffic—e.g., Checksum, TCP Segmentation Offload (TSO)
- DPDK PMD support
- Multi-queue support for packet steering to different CPU cores

Good
Receive Side Scaling (RSS) distributing traffic based on outer header.

Better
Receive Side Scaling (RSS) distributing traffic based on inner header.

Best
Intel Ethernet Flow Director distributing traffic based on flows and CPU core alignment.

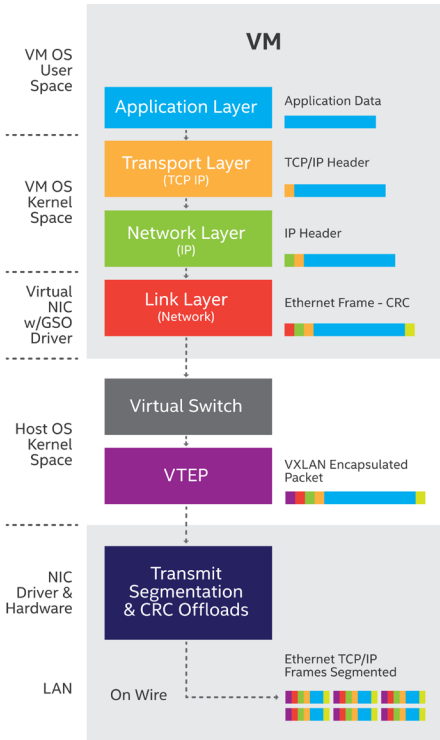

Figure 11: VXLAN Traffic Lso/Tso Offload

Additional Recommended Network Adapter Feature for Hardware Tunneling Engine NVE

- Encapsulation and de-capsulation engine that can operate at line speed for VXLAN, GENEVE, NVGRE, and VXLAN-GPE
- Network Service Chaining support for NSH and GEVENE
- Control Plane handled by SDN controller or OvS
- DPDK PMD support

Summary

Virtualizing the network is a major milestone in the path to Software Defined Infrastructure.

Data center virtualization is in part, focused on application and infrastructure agility and efficiency. Network Virtualization is designed to deliver this capability to the network fabrics that connect the data center.

Intel is focused on leading the network virtualization revolution with support for the evolving NVO standards in the Intel Ethernet Controller X710 and Intel Ethernet Controller XL710.

It is essential that stakeholders and influencers providing strategic direction, and supporting tactical implementation of network infrastructure, thoroughly investigate and evaluate Network Virtualization Overlay technologies as a key component of Software Designed Infrastructure. In addition, it is imperative to choose the appropriate network adapters and configure correctly to promote a smooth, efficient Network Virtualization Overlay implementation.

For more information on Intel® Ethernet Converged Network Adapters, visit www.intel.com/go/ethernet

Please Recycle 334399-001US