WHITE PAPER

Network Throughput in a Virtual Infrastructure

vmware®

Introduction

This paper outlines the considerations that affect network throughput. The paper examines the applications deployed on top of a virtual infrastructure and discusses the factors that affect the network throughput of virtualized applications. Illustrating key points with results obtained with **netperf**, the paper describes the benchmark settings so that the reader can reproduce the experiments.

VMware virtual infrastructure has many network configuration options and allows you to implement a wide variety of network architectures. This paper will not discuss different network configurations. See the ESX Server™ Administration Guide for information on network configuration options. Instead, it will focus on network throughput. The discussion should help you to plan for adequate network performance in your virtual infrastructure.

Networking in VMware ESX Server Architecture

VMware ESX Server is a data center-class virtualization platform. ESX Server runs directly on the system hardware and provides fine-grained hardware resource control

ESX Server virtualizes CPU, memory, storage, networking and other resources. Operating systems running inside virtual machines use virtualized resources, although from the operating system standpoint the resources appear as physical, dedicated hardware.

The key elements of the ESX Server system are:

- The VMware virtualization layer, which provides a standard hardware environment and virtualization of underlying physical resources
- The resource manager, which enables the partitioning and guaranteed share of CPU, memory, network bandwidth and disk bandwidth to each virtual machine
- The hardware interface components, including device drivers, which enable hardware-specific service delivery while hiding hardware differences from other parts of the system

Virtualization

The VMware virtualization layer brings hardware virtualization to the standard Intel server platform.

As with mainframe virtualization, the VMware virtual machine offers complete hardware virtualization; the guest operating system and applications (those operating inside a virtual machine) are not exposed directly to specific underlying physical resources they are accessing, such as which CPU they are running on in a multiprocessor system or which physical memory is mapped to their pages.

The virtualization layer provides an idealized platform that is isolated from other virtual machines on the system. It provides the virtual devices that map to shares of specific physical devices; these devices include virtualized CPU, memory, I/O buses, network interfaces, storage adapters and devices, mouse and keyboard and others.

Each virtual machine runs its own operating system and applications. The virtual machines are isolated from each other; they cannot communicate with each other or leak data, other than via networking mechanisms used to connect separate physical machines. This isolation leads many users of VMware software to build internal firewalls or other network isolation environments, allowing some virtual machines to connect to the outside while others are connected only via virtual networks through other virtual machines.

Network Virtualization

You may define up to four virtual network cards within each virtual machine. Each virtual network card has its own MAC address and may have its own IP address (or multiple addresses) and connects to a virtual network switch. The network switch may be mapped to one or more network interfaces on the physical server. ESX Server manages both the allocation of resources and the secure isolation of traffic meant for different virtual machines—even when they are connected to the same physical network card. Another choice involves binding a virtual network interface to a VMnet, a private network segment implemented in memory within the ESX Server system but not bound to an external network.

Private Virtual Ethernet Networks (VMnets)

VMnet connections may be used for high-speed networking between virtual machines, allowing private, cost-effective connections between virtual machines. The isolation inherent in their design makes them especially useful for supporting network topologies that normally depend on the use of additional hardware to provide security and isolation.

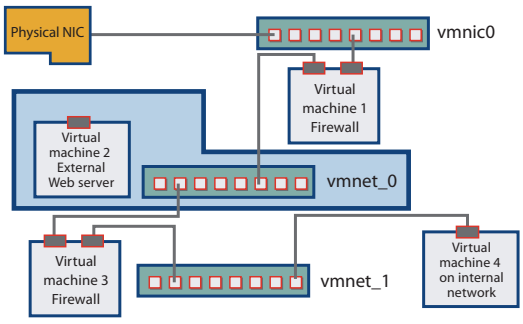

Figure 1

For example, an effective firewall can be constructed by configuring one virtual machine on an ESX Server system with two virtual Ethernet adapters, one bound to a VMnic (giving it a connection to a physical network) and the other bound to a VMnet. Other virtual machines would be connected only to the VMnet. By running filtering software in the dual-homed virtual machine, a user can construct an effective firewall with high-performance virtual networking between the virtual machines without the need for additional hardware.

General Considerations Affecting Network Throughput

Rated network throughput is almost always higher than the actual network throughput.

Even when all the physical elements of the network are rated at the same data throughput, this rate only represents the maximum theoretically possible throughput, rather than something that can be expected from a realistic architecture.

Physical Aspects of a Network

Before undertaking any network optimization effort, the physical aspects of the network need to be well understood. The following are just a few aspects of the physical layout that merit close consideration:

- Are both the client and the server members of a simple local area network (LAN) where the systems are connected to the same switch? If not, how many network hops do the packets need to traverse between the two systems? Do the speed and duplex settings for all the switch and host interfaces match each other?
- What are the types of network cards in interacting machines? Server-class NICs are often able to offer better performance.
- Are both the client and the server configured to use auto-negotiation to set speed and duplex settings? Are they configured for half-duplex or full-duplex mode?
- What size packets are transmitted through the network? Do the packets need to be fragmented or consolidated along the transmission path?

Traffic Patterns

Yet another set of issues impacting networking throughput and latency involve the use patterns of the network. Even traffic generated by the same application can be different depending on the groups that utilize the application and time of day. For example, a CRM application may generate a steady stream of small packets when used by individual sales representatives, but the traffic may become bursty and consist of large packets when the area managers generate activity reports. A few issues to be aware of when considering traffic patterns are:

- Frequency of the transactions and whether packets come in bursts.
- Size of the data packets.
- Sensitivity to data loss; for example, a multimedia streaming application using UDP may still present acceptable media quality to the user even when the data loss is as high as a few percent.
- Traffic directiveness - most of the time, network traffic is substantially asymmetric, with a lot more data transmitted downstream (from the server to the client) than upstream.

Network Stack Implementation and Configuration

Finally, the network protocol stack implementation in the operating system and application performance in processing network transactions often impacts overall network performance. With new network cards and switches now reaching 10Gbps, the bottleneck in processing network traffic often lies with available CPU cycles and system memory, whether it is for processing the transaction on the application level or for executing the operating system's TCP/IP stack. Another important consideration is the size of the network buffer that defines how many packets can be queued for sending or receiving.

Traffic Patterns' Effect on Throughput

As described above, network throughput is highly dependent on the network configuration and the specific application. This section illustrates with experimental data how traffic patterns affect throughput in a typical physical system. The examples below look at peak network throughput that can be achieved with given equipment on three different workloads. A detailed study of how the physical aspects of the network affect performance is beyond the scope of this paper.

- In many cases, a production application such as a Web server does not need high throughput for successful operation. In fact, most VMware customers configure multiple workloads to share the same network adapter with satisfactory network performance.

This paper uses a networking benchmark tool, **netperf,** to approximate three common traffic patterns in order to investigate and compare their associated throughput. **Netperf** is designed around the basic client-server model. The tool consists of two executables—**netperf**, which represents a client process and **netserver**, which represents a server process. The options for traffic patterns are set on the system running **netperf**, while the **netserver** is invoked on the server system.

It should be noted that the **netperf** test tool generates and transmits network packets. In this way, it is able to measure transmission performance independent of the source of the data.

More information about **netperf** including the source code is available from the **netperf** Web site: http://www.netperf.org/netperf/NetperfPage.html

For the purpose of this study, three workloads were investigated:

- **Default** traffic pattern, for approximating a workload of medium size messages
- **File** traffic pattern, for approximating the traffic flow of a file transfer
- **Bulk** traffic pattern, for approximating the traffic flow of bulk data operations

	default	file	bulk
Local send and receive socket buffer sizes (-s)	8192	8192	65536
Remote system send & receive socket buffer sizes (-S)	8192	8192	65536
Local send size (-m)	8192	4096	8192
Remote system receive size (-M)	8192	4096	8192

Table 1: Workload Traffic Patterns Definition

Below is a sample **netperf** command line you can use to reproduce our experiments

```
netperf -H <IP address> -l 60 -t TCP_STREAM -- -m 8192 -M 8192 -s 8192 -S 8192
```

Where•

-H designates the IP address of the system running **netserver**

- -l defines the test duration in seconds, 60 seconds was used for the experiments in this paper
- -t defines the test suite to run, a TCP_STREAM was used for all of the experiments
- -s and –S define send and receive socket buffer size on local and remote systems respectively
- -m and –M define the size of packets for local and remote systems.

All the experiments presented in this section were conducted with physical systems and thus serve as a good illustration of how traffic patterns affect maximum network throughput.

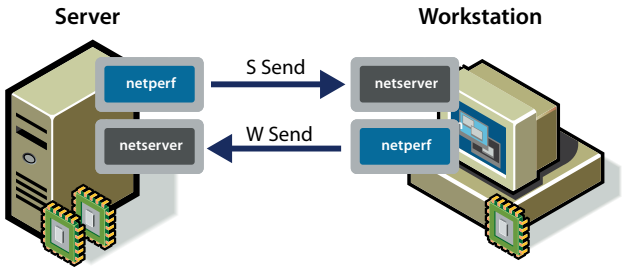

Figure 2: Experimental setup

For all three **netperf** workloads the same two Dell systems were used:

- Dell PowerEdge 1600SC with dual 2.4GHz CPU, 2GB of memory and server class 1 Gbps NIC (Intel Pro/1000 MT Server Adapter) referred to as "Server". The operating system was Windows 2000 Server.
- Dell Precision 340 with single 2.4GHz CPU, 2GB of memory and desktop class 1 Gbps NIC (Intel Pro/1000 MT Desktop Adapter) referred to as "Workstation". The operating system was Windows 2000 Pro.

The two machines were connected directly via a straight cable.

In the experiments designated **S Send Workstation** ran **netserver** while **Server** ran **netperf.** In the experiments designated **W Send**, **Server** ran **netserver** while Workstation ran **netperf**. Each workload was run several times and the maximum throughput value observed reported.

Figure 3 shows that neither the **default** nor the **file** workloads approach maximum rated throughput of 1Gbps NIC. Throughput for the bulk workload is much higher, exceeding throughput on the default workload by 180%. This observation underscores the fact that the pattern of the network traffic represents the most significant factor of network performance. In the bulk workload, the direction of traffic results in almost 300Mbps higher throughput in the **S Send** experiment compared to the **W Send** experiment, most likely due to a better NIC in the **Server.**

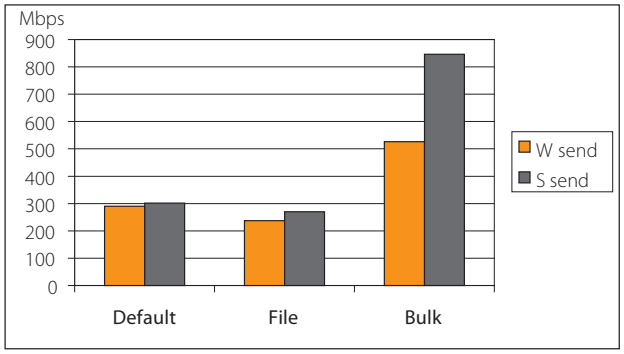

Figure 3: Throughput for different networking workloads on physical systems

Figure 4 shows corresponding CPU utilization for the throughput experiments depicted in Figure 3. The CPU utilization illustrated on the Y-axis of the graph is cumulative CPU utilization by both the sender and the receiver. There is a strong correlation between higher throughput and higher CPU utilization. This is especially clear for experiments with **bulk** workload where both throughput and CPU utilization are higher.

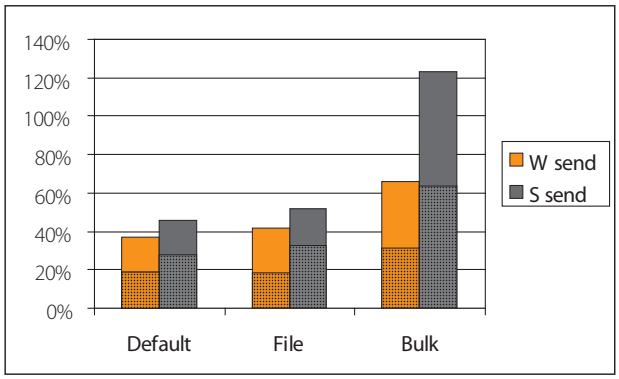

Figure 4: CPU utilization for different workloads. Maximum possible utilization is 300% CPU. Lower portion of the bar represents sender CPU utilization, upper portion of the bar represents receiver CPU utilization.

Virtualization's Effect on Throughput

After establishing how traffic patterns affect network throughput between physical systems, the next experiments show the impact of substituting a virtual machine for a physical system.

Figure 5 shows the new experimental setup. VMware ESX Server version 2.1***** is installed on the 2-CPU system and **netperf** is running inside a uniprocessor virtual machine running Windows 2000 server. The virtual machine was configured to use vmxnet virtual network adapter. The experiments for **default**, **file**, and **bulk** workloads were repeated with this new system setup.

*****Hyperthreading is enabled

The experiments with **netserver** running inside a virtual machines are designated **V Receive**. The experiments with **netperf** running inside a virtual machine are designated **V Send**.

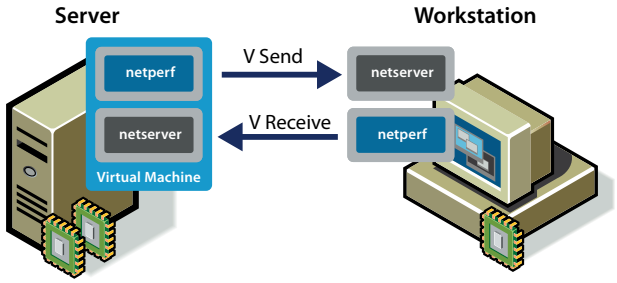

Figure 5: Experimental setup with a virtual machine

As shown in Figure 6, for the same workload the network throughput of the workload running inside a virtual machine is comparable with the network throughput of the workload running directly on a physical server. In fact, the achieved throughput was as high as 678 Mbps for the **bulk** workload sender running inside a virtual machine. You can see that although **V Send** experiments do show somewhat lower throughput, the impact of virtualization is much lower than the effect of the traffic pattern. With **bulk** workload, the throughput of virtualized workload is still higher than the throughput of **W Send** workload which uses a NIC of workstation class.

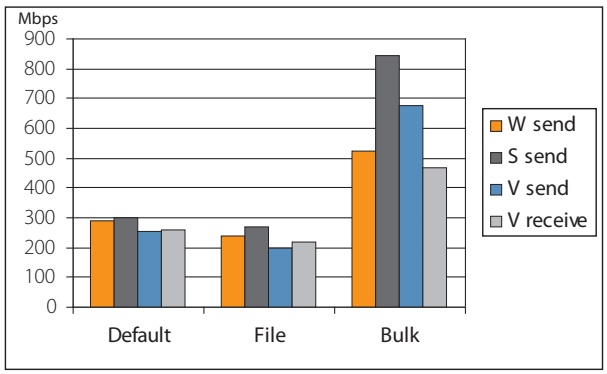

Figure 6: Throughput of different networking workloads on physical and virtual machines

In a physical environment, CPU utilization plays a significant role in reaching acceptable network throughput. To process higher levels of throughput, more CPU resources are needed.

The effect of CPU resource availability on network throughput of virtualized applications is even more significant. Running ESX Server requires a certain amount of fixed CPU resources that depend on the configuration of the server. In addition, because all the elements of the networking from physical to the applica-

tion layer are virtualized, processing network transactions is somewhat more expensive for virtualized applications than for applications running directly on the physical platform. Figure 7 illustrates corresponding CPU utilization for the throughput experiments shown in Figure 6.

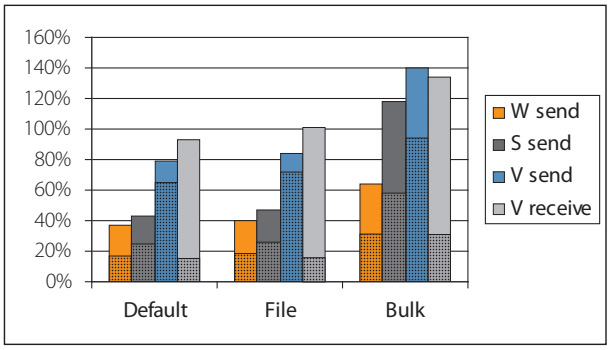

Figure 7: CPU utilization of different network workloads. Maximum possible utilization is 300% CPU. Lower portion of the bar represents sender CPU utilization, upper portion of the bar represents receiver CPU utilization.

Figure 8 summarizes the virtualization cost as reduction in the peak network throughput for both **V Send** and **V Receive** experiments. In most cases, such reduction is less than 20% although it can be higher if the workload is a traffic generator/server as in the case of **file** and **bulk** workloads.

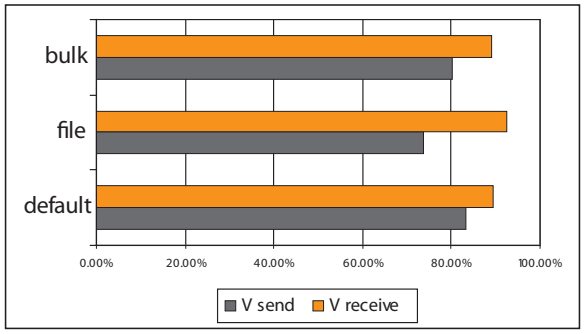

Figure 8: Throughput observed in virtual machines as a percentage of physical machine throughput.

Planning network throughput for virtualized applications is similar to planning network throughput of the applications running on physical platforms. As with physical platforms, the overall performance depends on physical elements of the network, network stack implementation, and, of course, the application itself and the nature of network traffic it generates and/or receives. You also need to take into account the higher CPU utilization required for networking transactions in virtualized applications.

Special Cases in Virtual Infrastructure Networking

One of the great advantages of virtualization is its flexibility and the ability to consolidate multiple servers in virtual machines on a single physical platform. In some cases such virtual machines hosted on the same physical platform need to be networked together.

Virtual infrastructure allows the transparent implementation of such a network configuration. When operating inside a virtual machine, the workloads use the same networking mechanism as they would use when running in physical machine using physical NICs. In many cases, the maximum throughput that can be reached between virtual machines is comparable to throughput between a virtual machine and a physical server. An additional set of experiments that focuses on network performance between two virtual machines on the same server illustrates this concept. Figure 5 shows ESX Server 2.1 installed on **Server** and a virtual machine running Windows 2000 Server which was used in V send and V receive experiments. In addition, we have created a second uni-processor virtual machine running Windows 2000 Pro. Both virtual machines used vmxnet virtual network adapters. ESX Server network buffer settings were tuned as described below in the section on configuration tips. We than ran three network workloads between the two virtual machines on the same server. These experiments are designated **VM-VM**.

Figure 9 shows the observed throughput between the two virtual machines. For default and file workloads, the throughput between virtual machines is similar to the throughput between a physical machine and a virtual machine. This is the case as long as the system running ESX Server has sufficient CPU resources available to process the networking transactions in both sender and receiver virtual machines. Insufficient CPU resources will reduce maximum throughput. In the case of bulk workload, CPU resources cannot meet the network workload requirements for both sender and receiver. The situation is further aggravated by the fact the guest operating system TCP flow control mechanism treats this condition as network congestion. As a result, the observed throughput is lower. It is important to monitor CPU utilization for such high throughput workloads. If the network throughput of such workload becomes an issue in production, the distribution of virtual machines between ESX Servers may need to be modified to address the issue.

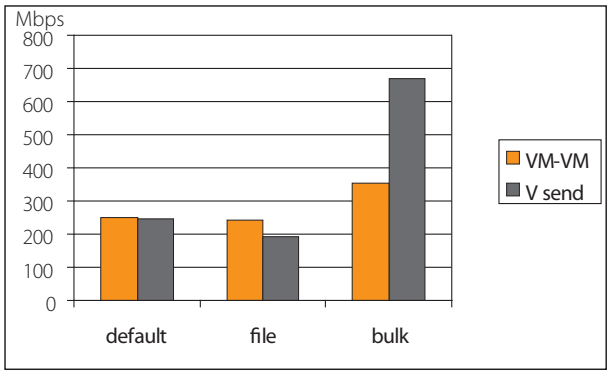

Figure 9: Maximum throughput achieved in virtual to virtual and virtual to physical connections

Another special case that merits consideration is when a virtual machine is configured to use two virtual CPUs with the help of VMware Virtual SMP™ module.

In this final set of experiments, the experimental set-up was the same as **V send** experiments, but the virtual machine running Windows 2000 Server was configured to use two virtual CPUs. These experiments are designated **Vsmp send**. As shown in Figure 10, the throughput between **V send** and **Vsmp send** workloads is comparable. Meanwhile, in the case of SMP virtual machines, CPU utilization is nearly 28% higher as illustrated on Figure 11. All the experiments ran one single-threaded **netperf** instance, so all the workloads are inherently uniprocessor workloads. A uniprocessor workload in an SMP virtual machine cannot make use of the second virtual processor. The second virtual CPU consumes CPU resources without enhancing workload performance and reduces the flexibility of the ESX Server scheduler. In our specific experiments, configuring the three network intensive workloads with virtual SMP did not lead to improved throughput. The virtual machines should only be configured to use multiple CPUs if they are running multi-threaded applications.

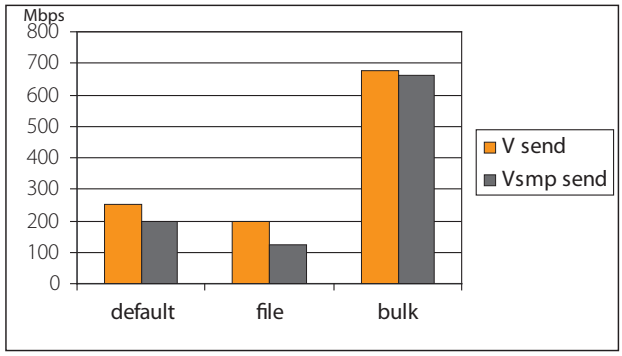

Figure 10: Throughput comparison for uniprocessor and SMP virtual machines

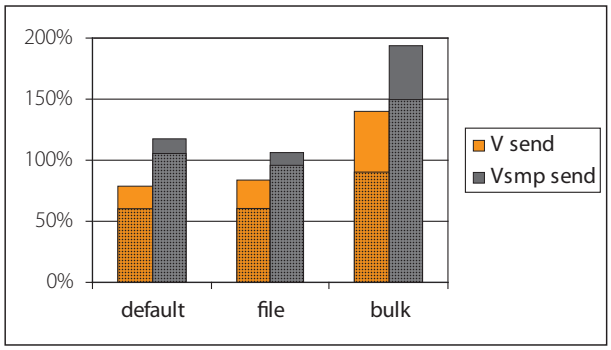

Figure 11: CPU utilization of network workloads in uniprocessor and SMP virtual machines. Maximum possible CPU utilization is 300% CPU. Lower portion of the bar represents sender CPU utilization, upper portion of the bar represents receiver CPU utilization.

Configuration Tips for Achieving Better Networking Performance for Virtualized Applications

When analyzing network throughput of virtualized applications, you need to look at all the elements of the network that would be involved if the application ran on physical servers. In addition, some networking configurations specific to a virtual infrastructure may improve throughput.

To check the current performance of the networking modules in ESX Server go the Network page in ESX Server MUI. The Network page shows network performance information and resources allocated to the virtual machine's virtual network card. The receive and transmit bandwidths indicate how fast data is transferred to and from the virtual machine. The values under Performance indicate throughput averaged over a past five-minute period. The averaging period for these statistics can be modified. The Network page also indicates whether traffic shaping is enabled.

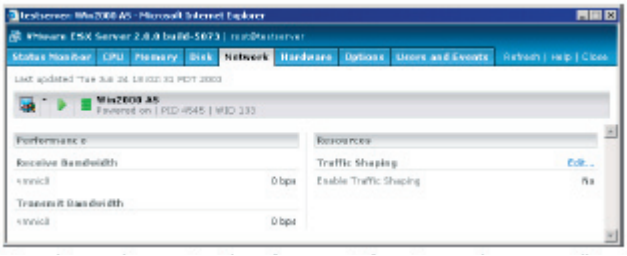

Figure 12

The following are some consideration of virtual infrastructure that may affect the physical elements of your network.

Enabling Traffic Shaping

When network traffic shaping is enabled, outbound network bandwidth is limited according to the values specified here. Because network traffic is bursty, separate parameters are provided to control both the long-term sustainable Average transmit rate and the short-term Peak transmit rate. The Burst parameter controls the amount of data that may be sent in one burst while exceeding the Average rate. The Peak rate limits the maximum bandwidth during such bursts.

To enable traffic shaping specify the average bandwidth. In the Average Bandwidth field, specify the average value for network bandwidth, then specify whether that amount is in Megabits per second (Mbps), Kilobits per second (Kbps) or bits per second (bps) also specify the Peak Bandwidth and the maximum burst size.

Forcing the Network Driver to Use a Specific Speed

The VMkernel network device drivers start with a default setting of Auto-negotiate.

This setting will work correctly with network switches set to Auto-negotiate. If your switch is configured for a specific speed and duplex setting, you must force the network driver to use the same speed and duplex setting.

If you encounter problems—in particular, very low bandwidth—it is likely that the NIC did not auto-negotiate properly and you should configure the speed and duplex settings manually.

To resolve the problem, either change the settings on your switch or change the settings for the VMkernel network device using the VMware Management Interface.

1. Log in to the management interface as root.
2. Click on the **Options** tab.
3. Click **Network Connections**.
4. Locate the device you want to reconfigure and choose the appropriate setting from the drop-down list for Configured Speed, Duplex.
5. Click **OK**.

   Note: Changing the network speed settings only takes effect after a reboot.

Choice of Network Adapters

Configuring virtual machines to use vmxnet virtual network adapters will significantly improve performance. Vmxnet driver-implements an idealized network interface that passes through network traffic from the virtual machine to the physical cards with minimal overhead. Vmxnet network driver is available as a part of VMware Tools installed inside your virtual machine guest operating system.

Network Throughput Between Virtual Machines

In some cases, low throughput between virtual machines on the same ESX Server machine may be caused by TCP flow control misfiring.

Buffer overflows require the sender to retransmit data, thereby limiting bandwidth. Possible workarounds are to increase the number of receive buffers, reduce the number of transmit buffers, or both. These workarounds may increase workload on the physical CPUs.

The default number of receive and transmit buffers is 100 each. The maximum possible for ESX Server 2.1.x is 128. You can alter the default settings by changing the buffer size defaults in .vmx (configuration) files for the affected virtual machines. Refer to the online knowledge base article 1428 by visiting: www.vmware.com/support. This article discusses how to diagnose and address the issue.

Conclusion

Networking sizing and performance considerations in the VMware virtual infrastructure are very similar to networking considerations in physical networks. Such considerations include patterns and intensity of the network traffic, implementation and configuration of network stack, available computing resources for network transaction processing and physical aspects of the network. In most cases, network throughput of virtualized workloads is comparable to the network throughput of the physical workloads. In production, the customers should plan to benchmark the network throughput of their specific network environment to ensure that it will meet their business needs.

V00014-20001205

VMware, Inc. 3145 Porter Drive Palo Alto CA 94304 USA Tel 650-475-5000 Fax 650-475-5001 www.vmware.com

vmware®