Voice over IP (VoIP) Performance Evaluation on VMware vSphere® 5

Performance Study

TECHNICAL WHITEPAPER

vmware®

Table of Contents

Introduction

The majority of business-critical applications such as Web applications, database servers, and enterprise messaging systems have been successfully virtualized, proving the benefits of virtualization for reducing cost and streamlining IT management. However, the adoption of virtualization in the area of latency-sensitive applications has been slow partly due to unsubstantiated performance concerns. By taking VoIP service as an example, this paper demonstrates that vSphere 5 brings the same virtualization benefits to latency-sensitive applications, and vSphere 5 does this while driving good performance. In particular, vSphere 5 delivers excellent out-of-the-box performance in terms of voice quality when running VoIP service.

VoIP applications are characterized by latency-sensitivity that dictates audio data be delivered at regular intervals to achieve good voice quality. Irregular delivery may lead to packet drops, severely deteriorating user experience. Therefore, timely processing and delivery of audio data is critically important to VoIP service. In the virtualized environment, however, meeting this requirement for VoIP applications is more challenging due to the additional layer of scheduling virtual machines (VMs) and processing network packets.

Despite such challenges, vSphere 5 is able to achieve great performance for VoIP applications thanks to the following reasons. First, vSphere 5 facilitates the highly optimized networking stack and paravirtualized device drivers to minimize virtualization overhead, adding little variance in packet delivery[1]. The overhead is usually in the order of tens of microseconds that are negligible, especially to VoIP applications, where packets need to be delivered at intervals of tens of milliseconds. Second, vSphere 5 gives each VM a fair share of CPU[2], ensuring the predictable processing of audio data even under high CPU contention when running multiple VMs. Finally, the Network I/O Control (NetIOC) feature allows VoIP traffic to be isolated by partitioning physical network bandwidth. This helps to achieve the intended voice quality when VoIP traffic competes for shared network resources.

This paper illustrates that:

- Excellent out-of-the-box VoIP performance is achieved with a large number of users served by a commercial VoIP media server hosted on vSphere 5.
- vSphere 5 is able to maintain great VoIP performance when running a large number of instances of VoIP server; results showed that vSphere 5 provided good performance even when running 12 instances configured with a total of 48 vCPUs on a system with 8 cores, utilizing more than 90% of the physical CPUs.
- With Network I/O Control (NetIOC), vSphere 5 is able to preserve voice quality under high contention for network resources.

VoIP Performance: Voice Quality and Jitter

When a VoIP media server is streaming voice traffic, audio data is periodically processed and sent out over the network. Packets are then received by the user at regular intervals. During the process, delayed packets may need to be dropped so as not to disrupt real-time playback. Since voice quality is very sensitive to packet loss (even 1% packet loss can jeopardize voice quality[3]), such packet drops caused by delayed packet delivery can result in degraded voice quality. In order to properly measure the timeliness of packet arrivals, *jitter* is defined to express the statistical variance of packet inter-arrival times. More formally, jitter is defined as the absolute difference between the expected inter-arrival time (usually 20 milliseconds in VoIP systems[4]) and actual inter-arrival time. It is further processed and smoothed out using a low-pass filter (as indicated by RFC 3550[5]). Performance tests in this paper use jitter as the main metric to evaluate VoIP performance.

If packets are delivered to the user at the desired interval of 20 milliseconds, jitter is zero. But if packets are delivered late or in burst, jitter is non-zero. To mitigate voice quality problems caused by high jitter and the resulting packet drops, a *jitter buffer* is often used. The jitter buffer reduces jitter by temporarily storing packets before forwarding them for further processing. With jitter buffers, voice quality may not be affected by a few milliseconds of isolated jitter spikes. The downside of using a jitter buffer is that additional end-to-end delay may be introduced.

Since human ears are known to easily endure up to 150 milliseconds of delay[6], voice quality is not significantly sensitive to end-to-end delay as long as it is maintained below the threshold. In a LAN environment where the evaluation in this paper was conducted, the end-to-end delay is dominated by a constant overhead, such as delays due to jitter buffers and CODEC operations. Combined together, they can attribute to more than tens of milliseconds depending on the buffer size and the CODEC type. vSphere adds little to this constant delay unless there is excessive resource contention; for example, by severe overcommitment. When this happens, it also should be manifested as an increased jitter, since packet delivery can be delayed. As jitter is sufficient to capture voice quality changes on our test setup, end-to-end delay was not measured in evaluation.

Audio CODECs also affect voice quality but tests exclude them, because the purpose of this performance study was not to evaluate different types of CODECs. We used only one type of CODEC in evaluation.

Evaluation of VoIP Performance on vSphere 5

This section evaluates the performance of commercial VoIP media servers on vSphere 5. Experiments were conducted by increasing 1) the number of users served by a single streaming media server instance and 2) the number of media server instances.

Test Configuration

The test configuration (shown in Figure 1) consisted of a Dell PowerEdge R710 server running vSphere 5, and two client machines that generated voice traffic. The server was configured with dual-socket, quad-core 3.4GHz Intel Xeon X5690 processors, 48GB of RAM, and one 1GbE Broadcom NIC. Hyper-threading was used to closely match a typical user environment.

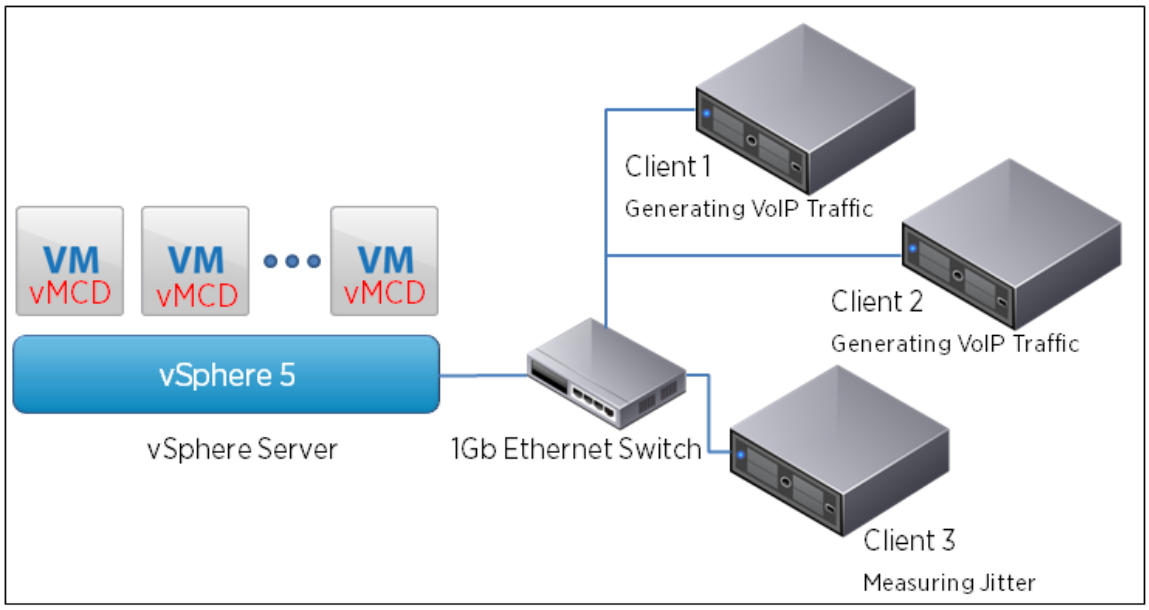

Figure 1. Test Bed Setup

The server hosted virtual Mitel Communications Director (vMCD)[7]. vMCD is software that provides call control and multi-party audio conferencing services. Tests used audio conferencing to measure jitter since call control operations are not latency-sensitive in comparison to streaming (that is, audio conferencing). Our tests used a G.729 CODEC that is often preferred in the wide area network over other CODECs such as G.711, because it compresses audio data and hence consumes less bandwidth. However, this also means G.729 is more CPU-intensive. vMCD was configured with 4 vCPUs and 2GB of RAM to properly support G.729 streams. VMXNET3

was used for virtual NICs. A Gigabit Ethernet switch was used to interconnect the vSphere host and client machines. The detailed hardware setup is described in Table 1.

	vSphere Host	Client1	Client2	Client3
Make	Dell PowerEdge R710	Dell PowerEdge 2970	Dell PowerEdge 1950	Dell PowerEdge 1950
CPU	2X Intel Xeon X5690 @ 3.40GHz, 4 cores per socket	6X AMD Opteron 2427 @ 2.20GHz, 1 core per socket	2X Intel Xeon X5355 @ 2.66GHz, 2 cores per socket	2X Intel Xeon 5160 @ 3.00GHz, 2 cores per socket
RAM	48GB	12GB	16GB	8GB
NICs	1 Broadcom NetXtreme II 1Gbps adapter	1 Broadcom NetXtreme II 1Gbps adapter	1 Broadcom NetXtreme II 1Gbps adapter	1 Broadcom NetXtreme II 1Gbps adapter

Table 1. Hardware Setup Details

Method

Two dimensions of evaluation were conducted. First, load was increased by increasing the number of audio conferences on a single instance of vMCD in order to see performance trends with regard to the number of users. Each conference call consisted of four participants. The number of conferences was varied from 5 to 30, which corresponds to 20 to 120 voice streams.

Second, VoIP performance was evaluated with multiple instances of vMCD to better understand how well vSphere 5 scales in terms of voice quality as the number of instances increases. Each vMCD instance served a demanding load of 120 G.729 streams. The number of vMCD instances was varied from 1 to 12, which increased the total number of streams from 120 to 1440. With 12 instances, the CPU of the ESXi host was almost fully utilized, at around 90%.

Three client machines were used to generate load and measure jitter. Two machines were used to generate load with a given number of voice streams. When multiple instances of vMCD were used, the same numbers of voice streams were created for each instance. The remaining machine was used to measure jitter by initiating an additional audio conference. The purpose of using a separate machine (and a separate conference) for jitter measurement was to reduce measurement variance. This also mimics the user environment where a single call is usually made. When multiple vMCD instances were used, jitter was measured for only one vMCD instance, as all instances were identical. All results shown are trimmed averages of 5 runs with a duration of 5 minutes each.

Single VoIP Media Server Instance Performance

This section evaluates the VoIP performance of a single vMCD instance by varying the number of voice streams. No manual tuning to the host was made.

The results are shown in Figure 2. As shown in the figure, the mean jitter was 0.4 millisecond with 20 G.729 streams and did not increase significantly as vMCD handled more streams. For example, with 120 streams, the mean jitter was just 0.54 millisecond. While the average behavior was good, we were also interested in the tail of the distribution, as jitter should be bounded to enforce the timeliness of audio streaming. The 99th-percentile jitter was maintained around 1 millisecond with the number of streams increased. Considering jitter buffers are usually used in several places in the VoIP network such as the voice gateway and the user, 0.54 and 1 millisecond of mean and 99th-percentile values are exceedingly low to achieve good voice quality. No packet loss was observed across all test configurations. These results with a commercial media server and a CPU-intensive CODEC clearly demonstrate that vSphere 5 adds a minimal overhead, and drives excellent out-of-the-box VoIP performance.

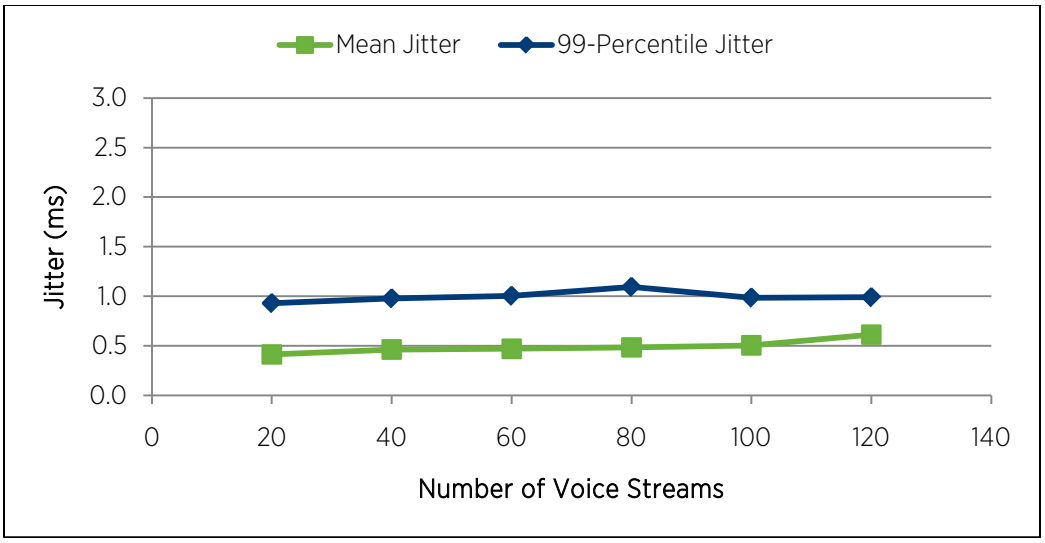

Figure 2. Mean and 99th-Percentile Jitters

Multi-Instance Performance

This section presents the result of VoIP performance with multiple instances of vMCD running on the host. Each vMCD instance ran 120 G.729 streams and the number of instances was increased from 1 to 12. Figure 3 shows the mean and 99th-percentile jitter values respectively. As seen in the figure, both the mean and 99th-percentile values increased with more vMCD instances running together (that is, more voice streams were being served); a slightly accelerated rate of increase in the 99th-percentile jitter was observed from four to eight instances (that is, 16 vCPUs to 24 vCPUs) as overcommitment occurred. This is expected due to increased CPU contention among VMs. However, the overall rate of increase both in the mean and 99th-percentile jitter is not steep, indicating that good voice quality is maintained. Even with 12 instances, the mean and 99th-percentile jitter values were very low, 0.80 and 1.93 milliseconds respectively, while the CPU was almost fully utilized, reaching around 90% (see Figure 4) with a total of 48 vCPUs running at the same time. The total packet rate with 12 instances was 72K packets per second, which consumed 45Mbps over a 1GbE link. No packet loss was observed regardless of the number of instances running.

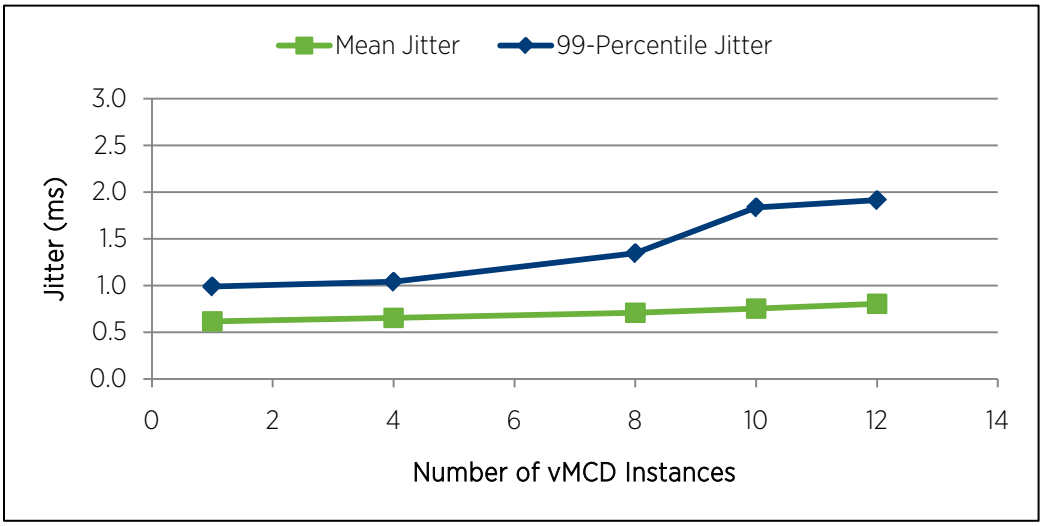

Figure 3. Mean and 99th-Percentile Jitters

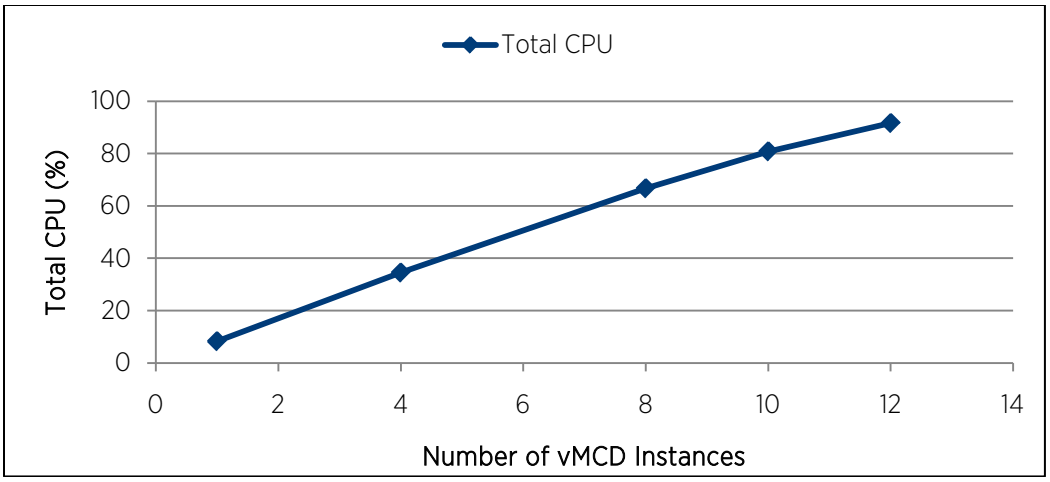

Figure 4. Total CPU Usage

The results show that good voice quality was maintained under CPU contention with multiple vMCD instances running. Effective fair-share-based CPU scheduling and low-overhead virtual networking in vSphere 5 prevented unbounded packet processing delays. In addition, flexible co-scheduling in vSphere 5 allowed a high CPU utilization of 90% to be achieved, with a total of 48 vCPUs running. Observe that the CPU usage increases almost linearly with the number of vMCD instances. This illustrates that per-VM scheduling and packet-processing overhead, if any, remained constant against the number of vMCD instances, further demonstrating efficient multi-VM management.

Isolating VoIP Traffic Using NetIOC

This section demonstrates how the Network I/O Control (NetIOC) feature in vSphere 5 helps to preserve voice quality under high network contention.

Test Configuration

The test configuration is described in Figure 5 and the same hardware setup described in Table 1 was used except that only two client machines were used for the test. In this evaluation, two different configurations were compared to see the benefits of NetIOC on VoIP applications: one system with NetIOC enabled and one without NetIOC support. For the configuration without NetIOC support, the same resource pool was used for the VoIP media server and 4 VMs generating competing network traffic. This simulated a condition where voice traffic was not differentiated from other traffic and packets were managed in a FIFO manner.

In the configuration with NetIOC support, separate resource pools were configured for the VoIP server and the other VMs. The other four VMs were put into one resource pool. The two resource pools were then configured to have the same priority (shares). NetIOC schedules packet transmission among multiple resource pools according to the bandwidth shares set by the user. In this case, NetIOC will ensure the same bandwidth to be allocated to both resource pools. For more details about NetIOC, see *VMware Network I/O Control: Architecture, Performance and Best Practices*[8].

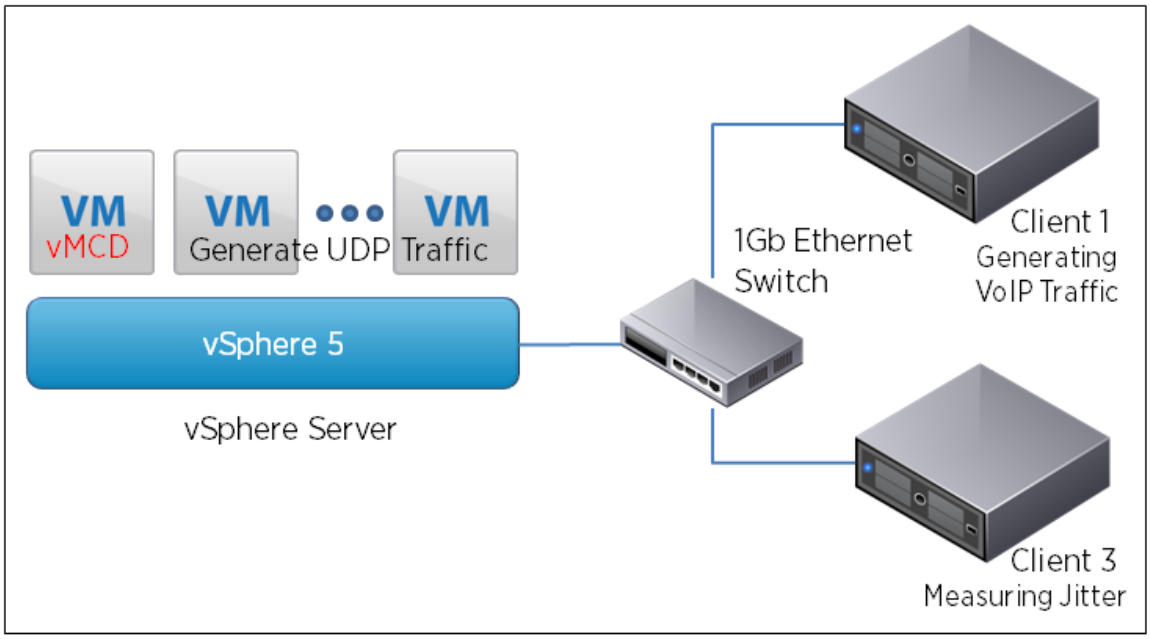

Figure 5. Hardware Setup

Method

At the start of the experiment, 120 G.729 streams were generated from a client machine. After 5 minutes, UDP traffic was created by the other four VMs and sent out to the remaining client machine in order to disrupt the voice traffic. The packet rate generated by those VMs was very high at around 390K packets per second, occupying approximately 880Mbps, whereas the packet rate of the VoIP traffic was much smaller at 6K packets per second using less than 4Mbps. The duration of the entire test was 10 minutes.

Results

Figure 6 shows the comparison of the configurations with and without NetIOC support. Without NetIOC support (that is, with the same resource pool configured for voice traffic as other VMs), voice packets were indiscriminately dropped after the UDP traffic became active; packet loss rate was constantly over 500 packets per second. This is more than 8% of the total packets generated and such a high packet loss rate is detrimental to voice quality[3].

On the contrary, with NetIOC support (that is, with a separate resource pool configured for the voice traffic), no packet loss was seen, which clearly demonstrates the benefits of NetIOC. NetIOC effectively differentiated the voice traffic such that it received all the required bandwidth, preventing any packet loss under high network contention.

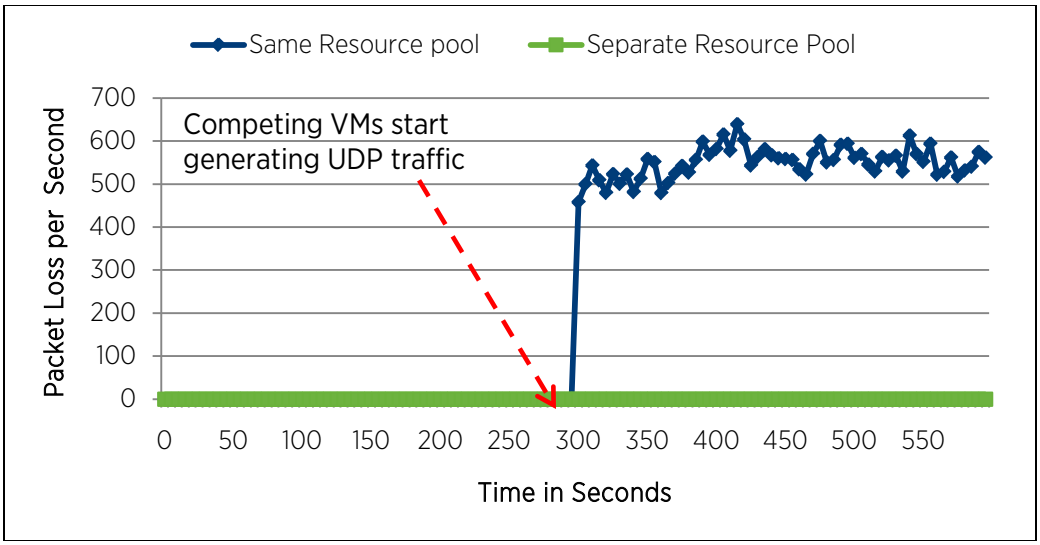

Figure 6. Comparison of the Number of Packet Losses

Table 2 shows mean and 99th-percentile jitters for the same experiments. As shown in the table, the numbers are small for both cases and there is no significant difference between the two. Jitter, for the case without NetIOC, is also small because either a voice packet is sent out without waiting too much time in the queue or it is simply dropped (the packet did not make it to the queue because it's full) before jitter is noticeably affected.

	Mean Jitter (ms)	99th-Percentile Jitter (ms)
Without NetIOC	0.64	2.00
With NetIOC	0.62	1.79

Table 2. Jitter Comparison With and Without NetIOC

NetIOC Usage Recommendation

Unless there is significant contention, NetIOC may not be necessary to provide good voice quality. Packet drops in vSphere happen only when the packet rate generated by (competing) VMs is extremely high and the network link is shared. Should packet loss occur due to such reasons, NetIOC greatly helps to sustain the desired voice quality.

When using NetIOC, assigning a separate resource pool with "normal" shares is sufficient in most cases, since the packet rate of VoIP traffic is not typically high and it consumes a relatively small amount of bandwidth. Doing so isolates VoIP traffic from competing traffic and guarantees the amount of bandwidth needed for achieving good voice quality.

NetIOC also provides the ability for packets of a given resource pool to be tagged with 802.1p tags when those packets leave the vSphere host into the physical switches. This allows appropriate prioritization to be enforced in physical switches with proper configuration in those switches.

Conclusion

This paper evaluated VoIP performance using a commercial media server to demonstrate that vSphere 5 is readily able to host VoIP service, delivering excellent out-of-the-box VoIP performance. To compare voice quality of different configurations, jitter was used as the main metric that describes the timeliness of voice packet arrival. The evaluation results demonstrated that good voice quality was maintained when the number of users (number of voice streams) and media server instance increased, while fully utilizing CPU. For example, vSphere 5 was able to maintain great VoIP performance even when running 12 instances of VoIP media server configured with a total 48 vCPUs on a system with 8 cores. It is further shown that the NetIOC feature was able to prevent packet loss successfully, thereby helping to preserve voice quality under severe contention for network.

References

1. *VMware vSphere 4.1 Networking Performance*. VMware, Inc., 2011. http://www.vmware.com/files/pdf/techpaper/Performance-Networking-vSphere4-1-WP.pdf.

2. *VMware vSphere: The CPU Scheduler in VMware ESX 4.1*. VMware, Inc., 2011. http://www.vmware.com/files/pdf/techpaper/VMW_vSphere41_cpu_schedule_ESX.pdf.

3. "Packet Loss," *A reference guide to all things VOIP*. Voip-Info.org, 2011. http://www.voip-info.org/wiki/view/QoS.

4. ITU-T Recommendation G.114, "One-way transmission time." International Telecommunication Union, 2003. http://www.itu.int/rec/dologin_pub.asp?lang=e&id=T-REC-G.114-200305-I!!PDF-E&type=items.

5. *Quality of Service Design Overview*. Cisco Press, 2004. http://www.ciscopress.com/articles/article.asp?p=357102.

6. Schulzrinne, H., S. Casner, R. Frederick, and V. Jacobson. *RTP: A Transport Protocol for Real-Time Applications*. The Internet Society, 2003. http://www.ietf.org/rfc/rfc3550.txt.

7. *Mitel Communications Director*. Mitel Networks Corporation, 2011. http://www.mitel.com/DocController?documentId=32750.

8. VMware Network I/O Control: Architecture, Performance and Best Practices. VMware, Inc., 2011. http://www.vmware.com/files/pdf/techpaper/VMW_Netioc_BestPractices.pdf.

About the Author

Jin Heo is a Member of Technical Staff in the Performance Engineering group at VMware. His work focuses on improving network performance of VMware's virtualization products. He has a Ph.D. in Computer Science from the University of Illinois at Urbana-Champaign.

Acknowledgements

The author would like to thank Boon Seong Ang, Guolin Yang, Haoqiang Zheng, Julie Brodeur, Lenin Singaravelu, Scott Goldman, Shilpi Agarwal, and Yong Wang (in alphabetical order) for reviews and contributions to the paper. The author would also like to thank Mitel for supplying their vMCD products used for evaluation, and in particular Jean-Yves Patry for offering his expertise on VoIP systems.

vmware

VMware, Inc. **3401 Hillview Avenue Palo Alto CA 94304 USA Tel 877-486-9273 Fax 650-427-5001 www.vmware.com**