An Experimental Analysis of Round-Trip Time Under Varying Resource Workloads in a VMware-Based Virtualized Network Environment

***[1]John Joshua B. Sazon, [2]Angelo Cylo Real, [3]Roxanne Caroll SP. Baoy, [4] Jayson A. Batoon**

[1,2,3,4]La Consolacion University Philippines, Malolos Bulacan

DOI: https://doi.org/10.51584/IJRIAS.2026.11100046

Received: 18 October 2026; Accepted: 23 October 2026; Published: 05 November 2026

ABSTRACT

Round-trip time (RTT) is a practical measure of communication responsiveness because it records the time required for a probe packet to travel to a destination and for a response to return. In virtualized environments, packet timing may be affected by competition for CPU processing, storage operations, and network capacity. This study examined how selected resource workloads were associated with RTT in a VMware-based virtual network. RTT was measured using a Windows PowerShell script that sent up to 1,000 ICMP echo requests to a target address and recorded response times, packet loss, and descriptive statistics. Twenty test conditions covered connectivity verification, baseline operation, CPU workload, disk activity, network traffic at increasing stream counts, recovery periods, and combined workloads. The reported baseline mean RTT was 1.84 ms. Under CPU load, mean RTT increased to 3.27–3.41 ms; under disk activity, it ranged from 2.48 to 2.67 ms; and under network load, it rose from 4.10 ms with one stream to 7.84 ms with four streams. The highest reported mean, 8.92 ms, occurred during combined CPU and network load, accompanied by 1.1% packet loss. Mean RTT returned close to baseline during the reported recovery conditions. These descriptive results show an association between resource workload and increased RTT in the tested environment. Because hardware specifications, VM allocations, and raw packet-level data were not fully documented in the available record, the findings should be interpreted as environment-specific rather than generalized to all virtualization platforms.

Keywords: round-trip time, virtualization, VMware, resource contention, network latency, packet loss, CPU workload, disk I/O

INTRODUCTION

Virtualization is a core technology in contemporary computing because it allows multiple operating-system environments and services to share a physical host. It supports flexible resource allocation, consolidation, testing, and deployment. However, virtual machines do not operate independently of the physical resources beneath them. They share processor time, memory bandwidth, storage systems, and network interfaces. When several operations compete for these resources, packet processing may be delayed and communication timing may become less predictable.

Round-trip time (RTT) is the elapsed time between transmitting a probe packet and receiving its corresponding response. The ping utility commonly reports RTT in milliseconds and can also reveal packet loss. RTT is not identical to one-way latency: it combines the outward and return paths and may include endpoint processing, scheduling delays, virtual networking overhead, and other transient effects. As a result, RTT is useful for observing responsiveness but does not by itself identify the precise location or cause of a delay.

The issue is particularly relevant in virtualized systems. A VM experiencing CPU pressure may wait longer to be scheduled; disk-intensive tasks may compete for host I/O resources; and heavy traffic may increase queueing on a virtual or physical network path. The extent of these effects depends on the hypervisor, operating system, resource allocation, workload intensity, network configuration, and the location of the workload relative to the measured communication path.

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume XI Issue X October 2026

The practical problem addressed in this study is that a virtualized network can appear responsive during idle operation but behave differently when resource-intensive tasks run concurrently. Without measurements under

distinct workload conditions, administrators and students may overlook how CPU, disk, and network activity coincide with changes in RTT and packet loss. The study therefore measures RTT under baseline, individual, recovery, and combined workload conditions in one VMware-based environment.

Research Objectives

• To measure baseline RTT in the tested virtualized environment.

• To describe changes in mean RTT, median RTT, observed minimum and maximum, standard deviation, and packet loss under CPU, disk, network, and combined workloads.

• To compare increasing network-stream conditions using the available summary measurements.

• To examine whether reported recovery measurements return toward baseline values after workloads are removed.

• To identify methodological limitations and practical implications for interpreting RTT measurements in virtualized environments.

Scope and Delimitation

This study describes reported RTT summaries for baseline, CPU, disk, network, recovery, and combined workloads. Configuration notes assign Sender to 192.168.56.101 and Responder to 192.168.56.102; however, the embedded ping screenshots show replies from 192.168.56.103. The actual historical Sender address and the role of .103 remain unverified. Consequently, the measured endpoint cannot yet be identified conclusively.

Significance of the Study

The measurements provide a small-scale empirical example for network and systems students learning how to collect and interpret RTT data. For administrators, the findings illustrate why a single idle-state ping result may not characterize behavior under load. The results can also inform future controlled tests by identifying the workload conditions that produced the largest changes in this particular environment.

Conceptual Framework

The independent condition in this study is the workload state applied during each test. The observed outcomes are descriptive RTT statistics and packet loss. Host specifications, VM allocations, hypervisor settings, background activity, and timing of workload execution can also affect the observations; where these factors were not recorded, they remain potential confounders rather than controlled variables.

Study component	Operational meaning	Recorded indicators
Workload condition	Baseline, CPU, disk, network, recovery, or combined activity	TC-01 to TC-20 labels
RTT central tendency	Typical response time in the test window	Mean and median (ms)
RTT spread	Variation and observed extremes	Standard deviation, minimum, maximum (ms)
Delivery reliability	Probe requests without a recorded response	Packet loss (%)
Recovery behavior	Measurements after load removal	Recovery-condition mean and loss

Table 1. RTT Performance Indicators Across Workload Conditions

Figure 1 is an embedded historical ping screenshot, not a conceptual framework. It shows replies from .103 and a summary of 1,000 sent and received probes. The Sender address, condition ID, date, and run ID are not

visible; the screenshot is not a complete packet log.

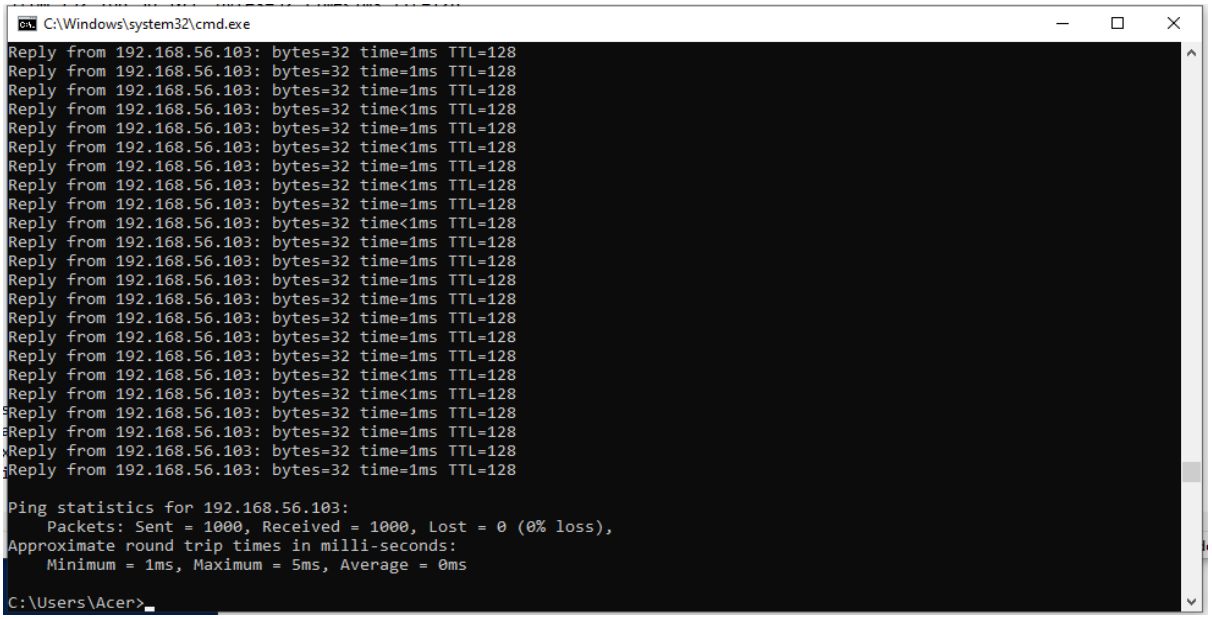


Figure 1. Historical ping screenshot with responding address 192.168.56.103 and unresolved VM role

LITERATURE REVIEW

Round-trip time (RTT) is the elapsed time between sending a packet and receiving its response. It reflects the combined effects of propagation, transmission, queuing, operating-system scheduling, packet processing, and return-path delay. In virtualized environments, packets may pass through a guest operating system, virtual network interface, hypervisor or virtual switch, and physical network interface. RTT can therefore change when the virtual machine or host is busy, when the network data path changes, or when packet-processing resources are shared.

The performance of virtualized network environments has become an important subject of investigation because virtualization introduces additional software layers between applications, operating systems, and physical hardware. These layers may influence network communication, particularly when computing resources are shared among multiple workloads. One relevant experimental investigation examined RTT in KVM virtual machines and LXC and Docker containers under different operating conditions. The researchers considered several factors, including CPU affinity, virtual network drivers, CPU load, disk input/output, and network load. Their findings demonstrated that RTT performance varies according to workload conditions and virtualization configurations, establishing the importance of evaluating network latency under controlled experimental circumstances. This study provides the primary foundation for the present research because it directly investigates RTT under different resource and virtualization conditions (de Oliveira Filho et al., 2022).

In addition to virtualization architecture, operating-system mechanisms can influence the time required to process network packets. Research on Kernel Busy Poll (KBP) investigated kernel-level enhancements designed to reduce network latency in virtual machines and containers without requiring modifications to application software. The study examined how improvements in packet-processing mechanisms could affect latency under virtualized networking conditions. Its findings highlight the importance of considering software-level processing overhead when analyzing communication delays. Although the present study does not implement kernel-level optimization, this research provides supporting evidence that RTT can be affected by mechanisms operating within the virtualized networking stack (Fujimoto et al., 2022).

The influence of network configuration on RTT was further investigated through an experimental evaluation of Single Root I/O Virtualization (SR-IOV) and virtualization. The researchers examined packet RTT under different network data-plane configurations to determine how network interface access and virtualization mechanisms affect communication performance. Their findings demonstrated that the configuration of the

network data path can influence packet delay. This research is particularly relevant because it establishes that measured RTT is not determined by virtualization alone; the underlying network configuration must also be considered when interpreting experimental results. Accordingly, the present study documents its VMware-based network environment and maintains consistent network settings throughout the workload experiments (de Oliveira Filho et al., 2023).

A related investigation evaluated SR-IOV in Docker containers using RTT measurements. The study compared network configurations involving SR-IOV with a traditional Linux bridge approach and reported RTT improvements under the tested SR-IOV configuration involving PCI passthrough. These findings indicate that the method through which a virtualized workload accesses network resources can influence communication latency. Although Docker networking differs from the VMware environment examined in the present study, the research provides a useful comparison by demonstrating how network architecture and access mechanisms may affect RTT. It also reinforces the importance of specifying the networking configuration when evaluating latency in virtualized systems (de Oliveira Filho et al., 2024). Beyond direct RTT investigations, research comparing native and virtualized system performance provides additional context for understanding resource-related overhead. An evaluation of system virtual machines examined performance across several categories, including CPU, memory, disk, and network operations. The study addressed differences between native and virtualized execution and highlighted the importance of considering multiple resource dimensions when assessing virtualization performance. Although these measurements do not directly establish how RTT changes under specific workloads, they support the premise that virtualization overhead may vary depending on the resource being used. This is relevant to the present study because CPU-intensive and disk-intensive activities may compete with network processing for resources within the experimental environment (Silva et al., 2023).

Similarly, research evaluating containers for low-latency packet processing in virtualized network environments examined the suitability of container-based approaches for communication tasks that require minimal delay. The findings indicated that containerized processing can achieve latency performance comparable to bare-metal processing under appropriate conditions. The study also highlighted the influence of hardware characteristics and workload conditions on latency behavior. These findings suggest that network performance should be assessed in relation to the complete execution environment rather than attributed exclusively to the virtualization technology. This provides a useful basis for examining both average RTT and its variability during the present study's workload experiments (Wiedner et al., 2024).

The relationship between virtualization arrangements and resource performance was also examined in an experimental assessment comparing standalone virtual machines, containers, and containers operating within virtual machines. The evaluation covered several performance dimensions, including CPU, memory, disk, and network operations. This research is relevant because running containers inside virtual machines introduces additional architectural layers that may affect resource utilization and execution performance. Although the present study focuses on RTT within a VMware-based virtualized network rather than directly comparing nested containers, the findings support the importance of identifying the virtualization arrangement and evaluating performance under clearly defined conditions (Aqasizade et al., 2025).

Zbořil and Svatá (2025) compared cloud virtual machines provided by Amazon Web Services, Microsoft Azure, and Google Cloud Platform using the Phoronix Test Suite and a selection of benchmarks covering major system-performance areas. Their findings showed differences in benchmark performance among the providers. Although the study does not directly establish RTT behavior in a locally configured VMware environment, it illustrates that results may vary across virtual-machine platforms and configurations. This supports the need to document the experimental environment and interpret the present findings within the limits of the tested configuration (Zbořil & Svatá, 2025).

The reviewed studies indicate that RTT and related performance measures may vary with virtualization architecture, network configuration, packet-processing mechanisms, and resource use. Research directly examining RTT provides the closest basis for this study, while studies of virtual-machine and container performance offer supporting context (de Oliveira Filho et al., 2022, 2023, 2024; Fujimoto et al., 2022; Wiedner et al., 2024). However, the reviewed studies use different platforms, workloads, and configurations, so their results cannot be assumed to apply directly to the VMware-based environment examined here. The

present study therefore describes RTT measurements under CPU, disk, network, and combined workloads using mean, median, standard deviation, observed maximum RTT, and packet loss.

Recent literature includes direct RTT experiments in KVM and Docker as well as broader studies of latency in containers, NFV, Kubernetes, and accelerated network data planes. However, these platforms and methods differ from the VMware environment examined here. The specific contribution of this work is a transparent set of baseline and workload measurements in its own VMware-based environment, with CPU, disk, network, and combined workloads reported using consistent summary statistics. The results therefore provide a local experimental profile rather than a claim that VMware is universally faster or slower than other platforms.

The configuration notes describe Sender, Responder, and Background roles. Their address assignments are provisional until reconciled with original interface records and logs. Five Background allocations are documented, but no condition-level result is linked to an allocation; their performance cannot be compared.

MATERIALS AND METHODS

A quantitative experimental approach was used. RTT was measured repeatedly under predefined workload conditions, and summary statistics were compared across the recorded sequence. The workload condition served as the principal explanatory factor. Outcomes included mean RTT, median RTT, minimum and maximum RTT, standard deviation, and packet loss. The analysis is descriptive because raw packet-level observations and sufficient independent runs for inferential testing were not supplied in the results document.

Experimental Environment

The reported topology comprises a Sender, a Responder, and a Background workload VM. Table 2 records the assignments in the configuration notes rather than verified historical addresses. The embedded screenshots identify .103 as the responding endpoint and do not establish the Sender address. Obtain contemporaneous ipconfig /all output, VM network settings, and original command transcripts before finalizing the topology.

The configuration notes describe VMware Workstation Pro on a Windows 10 Pro 64-bit host with an Intel Core i5 processor, 8 GB physical RAM, and SSD storage, and three Windows 10 64-bit guests. These specifications have not been independently corroborated. The exact processor model and core/thread count, SSD model/capacity, VMware version/build, guest editions/builds, virtual NIC models, and virtual disk/controller settings require author verification. Sender and Responder are listed at 1 vCPU and 2 GB RAM each; Background allocations are listed separately in Table 2. Whether all five allocations were actually tested is not established.

The VM configuration notes list the Sender VM at 192.168.56.101 (1 vCPU, 2 GB RAM), the Receiver VM at 192.168.56.102 (1 vCPU, 2 GB RAM), and Background VM configuration options A–E with their respective IP addresses and vCPU/RAM allocations. The reported results do not contain separate RTT summaries for each configuration, so no configuration-specific performance comparison is made.

VM / configuration	IP address	vCPU	RAM	Role / notes
Sender	192.168.56.101	1	2 GB	Sends RTT measurement probes
Receiver / Responder	192.168.56.102	1	2 GB	Receives RTT probes from Sender
Background – Config A	192.168.56.103	1	1 GB	Background workload; .103 is intended workload VM
Background – Config B	192.168.56.104	1	2 GB	Background workload
Background – Config C	192.168.56.105	2	2 GB	Background workload
Background – Config D	192.168.56.106	1	3 GB	Background workload
Background – Config E	192.168.56.107	2	3 GB	Background workload

Table 2. Virtual Machine Configurations and Network Addresses

The notes specify a host-only VMnet1 network, subnet 192.168.56.0/24, mask 255.255.255.0, and no required default gateway. Verify these settings against Virtual Network Editor and each VM adapter record, including DHCP versus static addressing, MAC addresses, additional adapters, MTU, and firewall rules. Record whether all VMs shared one physical host and whether Background configurations ran sequentially or concurrently.

Throughout this revision, Responder denotes the endpoint receiving RTT probes. The provisional .102 assignment must not be substituted into historical logs: the .103 endpoint shown in the screenshots must first be explained and mapped to its actual VM role.

Component	Documented information	Information not available in supplied record
Virtualization	VMware-based environment	Exact product/version and host configuration
Measurement host	Windows PowerShell ping script	Full OS build and hardware specifications
Target	Unverified; screenshots show 192.168.56.103	Reconcile with provisional Responder .102 and confirm Sender address
Measurement	Up to 1,000 ICMP requests; one-second pause	Raw per-packet export for all test conditions
Workloads	CPU, disk, network, and combined conditions	Exact CPU utilization, disk throughput, and network throughput achieved

Measurement Instrument and Script

The described measurement procedure attempts 1,000 ICMP echo requests per main test, with a stated one-second pause. The exact PowerShell script, version, timeout, payload, timestamp method, and interval implementation are not preserved here. The plain ping command below is a documented command example and does not establish that the described script generated the summary statistics.

The available results consist of condition-level RTT summary statistics. Complete original packet-by-packet RTT logs for every condition were not available for independent recalculation. For reproducibility, future logs should include the condition ID, run number, timestamp, target IP, sent/received/lost counts, and per-probe response times.

The manuscript lists ping -n 1000 192.168.56.102, whereas the screenshots show replies from 192.168.56.103. Neither address is accepted as the verified historical Responder without the original run records.

For a new verified run, replace the placeholder before execution: ping -n 1000 <verified-responder-IPv4> > C:\RTT_Test\TC06_Run01.txt. This is a prospective logging example, not an original experimental record. Preserve the actual command and interface configuration with each run.

RTT results are summarized by test condition using descriptive statistics. The results are interpreted in relation to the defined Sender-to-Receiver path and the workload generated by the Background VM.

Statistic	Excel formula example	Purpose
Mean RTT	8	Average of extracted successful RTT values
Median RTT	8	Middle value of sorted RTT values
Minimum RTT	1	Smallest successful RTT
Maximum RTT	15	Largest successful RTT
Sample standard deviation	#NAME?	Spread of observed successful RTT values
Packet loss (%)	#NAME?	Share of attempted probes with no reply

Table 3. RTT Statistical Measures and Calculation Formulas

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume XI Issue X October 2026

Workload Conditions

The workload descriptions specify CPU square-root calculations for up to 20 minutes, repeated write/read/delete operations on a 50 MB temporary file, and iPerf3 with one, two, or four parallel streams. The executable scripts, versions, worker counts, disk path/cache settings, and actual workload start/stop times require verification. These descriptions establish intended activity, not achieved resource utilization.

CPU, disk, and network workloads were applied through the Background VM according to the selected test condition. Complete time-aligned CPU utilization, disk activity, and network-throughput series were not available for every RTT probe; these resource measurements are discussed only where supporting quantitative records are available.

Condition	Verification method	Information supplied	Evidence still needed
Baseline	Task Manager / iPerf3	Idle reference; network traffic described as near 0 Mbps	Actual idle CPU and disk readings, if recorded
CPU load		Target approximately 90–100% CPU utilization	Actual utilization log or screenshot for each relevant run
Disk load		Monitor Active Time %, read speed (MB/s), and write speed (MB/s); activity expected to rise versus idle	Actual Active Time %, read speed, and write speed
Network load	iPerf3 between Background VM and Responder VM	iPerf3 reports throughput in Mbit/s	Actual throughput output for one-, two-, and four-stream tests
CPU + network	Task Manager and iPerf3	CPU target approximately 90–100%; network throughput measured by iPerf3	Paired CPU and throughput readings from the same trial

Table 4. Workload Verification Methods and Evidence

The intended network workload runs an iPerf3 client on Background and a server on Responder. Historical scenario notes instead identify .103 as the client target, conflicting with the provisional .102 Responder assignment. Resolve the endpoint mapping before reporting the traffic path. For a new verified run, use iperf3 -s on Responder and iperf3 -c <verified-responder-IPv4> -t 1200 -P 1 -J on Background, changing -P to 2 or 4 as required. These prospective examples do not establish the commands executed historically or the throughput achieved.

Test Procedure

• Verify that the sender can reach the target address using a short connectivity check.

• Run a pilot baseline to check that the measurement script returns usable RTT values.

• Collect the full baseline and a repeated baseline measurement.

• Apply each documented workload and collect RTT measurements using the same script.

• Repeat selected workload conditions and record recovery measurements after workload removal.

• Run combined workload conditions, then perform the final baseline verification.

• Compile the reported summary statistics into a common comparison table.

TC-01 to TC-20 are condition identifiers, not independent repetitions per condition. The summary contains two baseline entries (TC-03/04), two CPU entries (TC-06/07), and two four-stream entries (TC-15/16). TC-11 is labelled a disk repeat but its correspondence to TC-09 or TC-10 is unresolved. TC-20 is a later baseline check. Original dated run records are required to establish independence and the actual run count for every condition. No claim of 20 runs or 20,000 probes per condition is supported.

The test procedure consists of establishing connectivity, recording a baseline, applying the specified Background VM workload, measuring RTT during each condition, and collecting recovery measurements after workload activity. The analysis uses the available condition-level RTT summaries and does not infer unrecorded resource metrics.

Condition entries	Entries in summary	Verified independent runs
Baseline TC-03/04	2	Not established by original logs
CPU TC-06/07	2	Not established by original logs
Four streams TC-15/16	2	Not established by original logs
Disk TC-09/10/11	3; repeat mapping unresolved	Not established by original logs
Other TC identifiers	1 entry per identifier	Not established by original logs

Table 5. Documented condition entries and verification of independent runs

The reported summaries are preserved as supplied. Because packet-level logs were not provided for all conditions, the calculations could not be independently recomputed from the original observations. The script calculates standard deviation across successful packet RTTs within a run; this within-run standard deviation is not the same as variability across independent runs. For publication-quality replication, report the number of independent runs separately from the number of probes within each run and analyze run-level summaries when multiple independent repetitions are available.

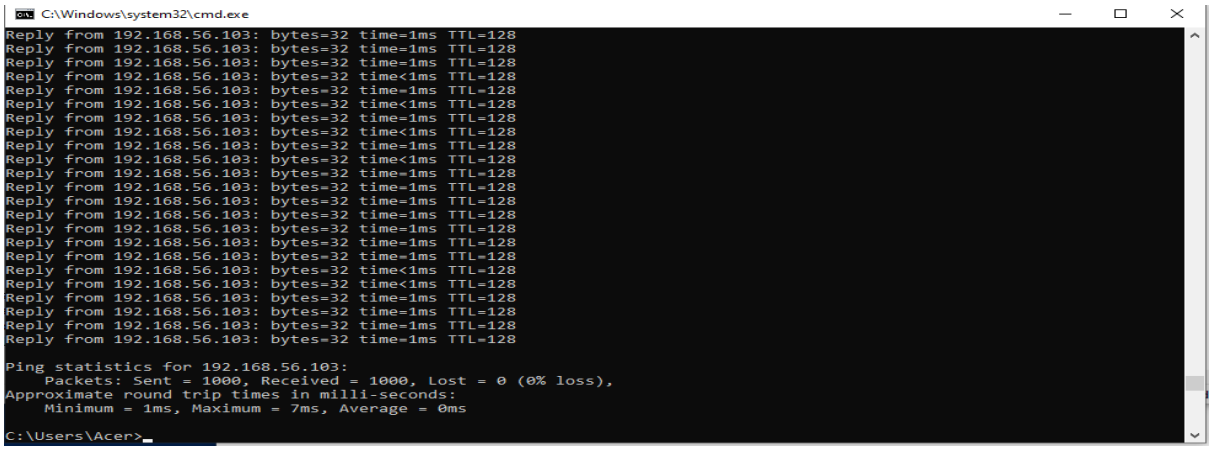


Figure 2. Historical ping screenshot with responding address 192.168.56.103 and no verified condition or run identifier

Variables and Operational Definitions

Variable	Role	Definition
Workload condition	Independent / grouping variable	Baseline, CPU, disk, network, recovery, or combined workload state
Mean RTT (ms)	Outcome	Arithmetic average of successful recorded responses
Median RTT (ms)	Outcome	Middle value of sorted successful response times
Minimum / maximum RTT (ms)	Outcome	Smallest and largest recorded successful response times

Standard deviation (ms)	Outcome	Reported spread of response times around the mean
Packet loss (%)	Outcome	Percentage of probes for which no response was recorded
Host/VM configuration	Potential confounder	Hardware, guest allocation, hypervisor settings, and network topology; incompletely documented

Table 6. Experimental Variables and RTT Performance Measures

Data Processing and Statistical Analysis

Descriptive statistics were used to summarize the reported results for each experimental condition, including mean, median, minimum, maximum, sample standard deviation, and packet-loss percentage where available. These statistics describe the successful packet responses recorded in each reported condition or run. Because complete raw RTT logs were unavailable, the reported calculations could not be independently verified.

Packet-loss percentage was interpreted as the proportion of attempted probes that did not produce a recorded response. TC-01 was treated as a short connectivity check, while TC-02 was treated as a pilot baseline. These tests were not considered equivalent in duration or evidential strength to the full 1,000-probe conditions. TC-03 and TC-04 represented the documented full and repeated baseline conditions.

The analysis remained descriptive and focused on comparing the available condition-level summaries. No claims of statistical significance, confidence intervals, or causal effect estimates were made. The 1,000 probes collected within a run were treated as packet-level observations rather than 1,000 independent experimental runs. More advanced statistical analyses would require complete raw packet-level data, clearly documented independent repetitions, and an appropriate method for distinguishing within-run variability from between-run variability. Therefore, the findings should be interpreted as descriptive comparisons of the reported results.

RESULT AND DISCUSSION

Overview of the Test Sequence

Twenty condition-level summary entries are reproduced in Tables 7 and 8. The appended scenario table labels the same figures Sample Actual Result. Their provenance as actual experimental measurements therefore requires author confirmation and original logs. The numerical discussion is conditional on that confirmation; it is not independent validation of the figures. No table entry can presently be assigned to Background configuration A, B, C, D, or E.

ID	Condition	Mean (ms)	Median (ms)	Min	Max	SD	Loss %
TC-01	Connectivity check	—	—	—	—	—	0
TC-02	Pilot baseline (100 packets)	1.72	—	1	5	—	0
TC-03	Full baseline	1.84	1.72	1	7	0.92	0
TC-04	Baseline repeat	1.91	1.8	1	8	0.97	0
TC-05	Background VM idle	1.96	1.82	1	8	1.01	0
TC-06	CPU load	3.27	3.05	1	14	1.86	0.2
TC-07	CPU load repeat	3.41	3.12	1	15	1.94	0.3
TC-08	CPU recovery	2.02	1.88	1	8	1.05	0
TC-09	Disk write load	2.48	2.3	1	11	1.32	0.1
TC-10	Disk read/write load	2.67	2.42	1	12	1.45	0.1

Table 7. Results for TC-01 through TC-10. A dash indicates a statistic not reported for that condition.

ID	Condition	Mean (ms)	Median (ms)	Min	Max	SD	Loss %
TC-11	Disk load repeat	2.59	2.38	1	12	1.39	0
TC-12	Disk recovery	1.99	1.85	1	8	1	0
TC-13	Network load: one stream	4.1	3.74	1	19	2.31	0.2
TC-14	Network load: two streams	5.26	4.91	1	24	2.88	0.4
TC-15	Network load: four streams	7.84	7.12	1	36	4.17	0.8
TC-16	Network load: four-stream repeat	7.69	7.01	1	35	4.02	0.7
TC-17	Network recovery	2.06	1.91	1	9	1.08	0
TC-18	Combined CPU + disk	4.36	4.02	1	21	2.37	0.3
TC-19	Combined CPU + network	8.92	8.14	1	42	4.88	1.1
TC-20	Final baseline verification	1.93	1.79	1	8	0.98	0

Table 8. Results for TC-11 through TC-20. The summary values are transcribed from the supplied results report.

Baseline RTT

The full baseline condition (TC-03) reported a mean RTT of 1.84 ms, median of 1.72 ms, maximum of 7 ms, standard deviation of 0.92 ms, and no packet loss. The repeated baseline (TC-04) reported a mean of 1.91 ms, median of 1.80 ms, maximum of 8 ms, standard deviation of 0.97 ms, and no loss. The final baseline verification (TC-20) reported a mean of 1.93 ms and median of 1.79 ms, with maximum RTT of 8 ms, standard deviation of 0.98 ms, and no packet loss. The close range among the three full baseline means—1.84 to 1.93 ms—suggests that the measured RTT returned to a similar level in these baseline windows. It does not prove that the underlying system was perfectly stable throughout the entire experiment, because the report does not provide raw time-series plots or repeated baseline runs at every point in time.

CPU Workload

During CPU load, TC-06 reported mean RTT of 3.27 ms, median of 3.05 ms, maximum of 14 ms, standard deviation of 1.86 ms, and 0.2% packet loss. The repeat (TC-07) reported mean RTT of 3.41 ms, median of 3.12 ms, maximum of 15 ms, standard deviation of 1.94 ms, and 0.3% packet loss. Compared with TC-03 baseline, the TC-06 mean increased by approximately 77.7% (from 1.84 to 3.27 ms). The maximum rose from 7 to 14 ms, and standard deviation approximately doubled. The repeated CPU test showed a similar direction. These figures indicate higher observed delay and variability during the documented CPU workload, although the experiment does not isolate the precise internal mechanism.

Disk Workload

The disk-write condition (TC-09) reported mean RTT of 2.48 ms, median of 2.30 ms, maximum of 11 ms, standard deviation of 1.32 ms, and 0.1% loss. The disk read/write condition (TC-10) reported mean RTT of 2.67 ms, median of 2.42 ms, maximum of 12 ms, standard deviation of 1.45 ms, and 0.1% loss. A repeated disk condition (TC-11) had mean RTT of 2.59 ms and standard deviation of 1.39 ms, with no loss. The disk-load means were above the 1.84 ms full baseline, but below the reported CPU-load and network-load means. The recovery test after disk activity (TC-12) returned to a mean of 1.99 ms, with no loss. The available measurements therefore show a moderate increase during disk activity and a return close to the baseline level during the recorded recovery window.

Network Workload and Stream Count

The network-load conditions show a graded increase across the reported stream counts. TC-13, using one stream, had a mean RTT of 4.10 ms; TC-14, using two streams, had a mean of 5.26 ms; and TC-15, using four

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume XI Issue X October 2026

streams, had a mean of 7.84 ms. Their standard deviations were 2.31, 2.88, and 4.17 ms, respectively. Packet loss increased from 0.2% to 0.4% and 0.8%.

The four-stream network repeat (TC-16) produced a reported mean of 7.69 ms, standard deviation of 4.02 ms, and packet loss of 0.7%, compared with 7.84 ms, 4.17 ms, and 0.8% for TC-15. The two means are similar in this recorded sequence. However, only selected repeats are documented, and there are no iPerf3 throughput logs to quantify the traffic actually achieved.

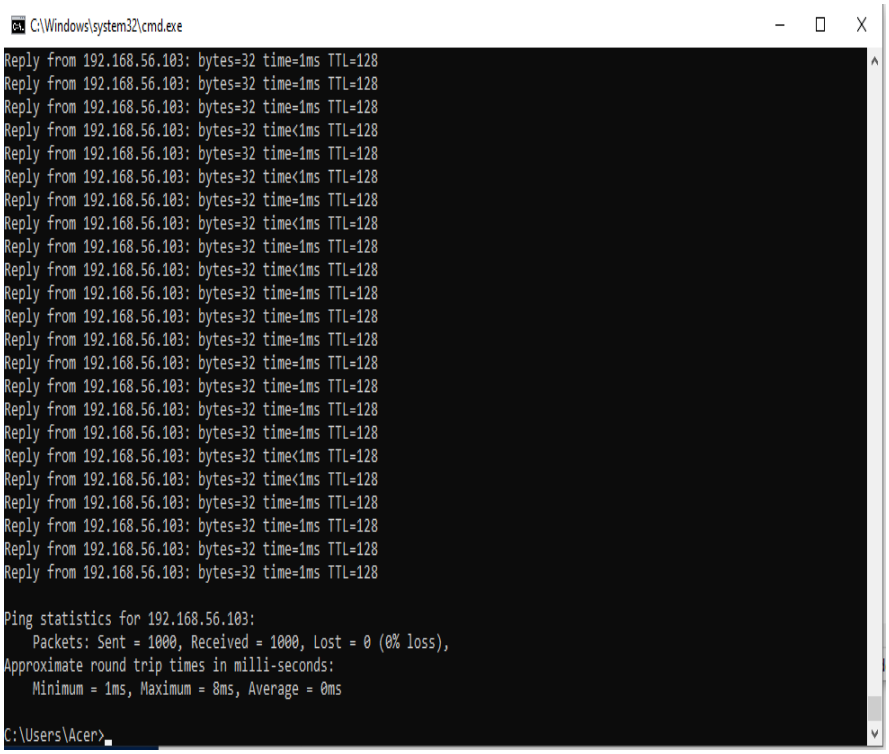


Figure 3. Historical ping screenshot with responding address 192.168.56.103 and no verified condition or run identifier

Recovery Conditions

The recorded recovery means were 2.02 ms after CPU load (TC-08), 1.99 ms after disk activity (TC-12), and 2.06 ms after network load (TC-17). All three recovery conditions reported zero packet loss. These values are close to the full baseline mean of 1.84 ms and final baseline mean of 1.93 ms. The pattern suggests that measured RTT moved back toward its baseline level after the applied workloads were removed.

Recovery observations are useful because they provide a within-sequence comparison. Still, the record does not specify the exact recovery duration before each measurement, and the workload process termination and system settling time were not independently verified. Therefore, the results show a return toward baseline in the reported measurements rather than a formally established recovery time.

Combined Workloads

The combined CPU–disk condition (TC-18) reported mean RTT of 4.36 ms, median of 4.02 ms, maximum of 21 ms, standard deviation of 2.37 ms, and 0.3% packet loss. The combined CPU–network condition (TC-19) reported mean RTT of 8.92 ms, median of 8.14 ms, maximum of 42 ms, standard deviation of 4.88 ms, and 1.1% packet loss. These are the highest reported mean and loss values among the listed conditions, but workload resource utilization and achieved throughput were not measured in a complete synchronized record.

TC-19's reported mean RTT of 8.92 ms is approximately 4.85 times the TC-03 baseline mean of 1.84 ms, a difference of 7.08 ms (about 384.8% above baseline). This is a descriptive comparison of the supplied summaries rather than a causal estimate.

Summary of Principal Observations

Comparison	Observed pattern	Interpretation limited to this dataset
Baseline vs. CPU	Mean 1.84 ms vs. 3.27–3.41 ms	Higher RTT and spread during CPU workload
Baseline vs. disk	Mean 1.84 ms vs. 2.48–2.67 ms	Moderate increase during disk activity
Network stream count	4.10 → 5.26 → 7.84 ms	Mean RTT rose with the tested stream count
Four-stream repeat	7.84 vs. 7.69 ms	Similar high mean in the repeated condition
Recovery checks	1.99–2.06 ms	Means returned close to baseline
Combined CPU + network	8.92 ms; 1.1% loss	Highest mean and loss among recorded conditions

Table 9. Summary of RTT Patterns Across Workload and Recovery Conditions.

Packet Loss

Packet loss was zero in the full baseline, repeated baseline, background-idle, recovery, and final-baseline conditions. Small losses were reported during some workload tests, including 0.2–0.3% under CPU load, 0.1% under disk activity, 0.2–0.8% under network load, 0.3% under CPU–disk load, and 1.1% under CPU–network load. The largest reported loss therefore coincided with the highest mean RTT condition.

RTT and loss should be interpreted together. The mean RTT is calculated from successful replies and excludes packets that did not return. When loss increases, the successful-packet mean alone may understate the overall service impairment. The report does not provide enough raw data to determine whether loss occurred in bursts or randomly distributed across each run.

DISCUSSION OF MAIN FINDINGS

The results show that Round-Trip Time (RTT) varied across the workload conditions in the tested VMware-based virtualized network environment. During the full baseline test, the mean RTT was 1.84 ms, while the repeated baseline test recorded 1.91 ms. These measurements provide a reference for examining changes under additional resource workloads. The small difference between the two baseline measurements suggests that the observed RTT remained relatively similar during these tests. However, the results describe only the tested environment and do not establish that the same performance pattern will occur in other virtualized systems.

Under CPU workload conditions, the mean RTT increased to 3.27 ms in the first test and 3.41 ms in the repeated test. The standard deviation also increased to 1.86 ms and 1.94 ms, respectively, compared with 0.92 ms and 0.97 ms in the two baseline tests. These results indicate that RTT was higher and more variable while the CPU workload was active. One possible explanation is that CPU-intensive activity increased competition for processing resources, potentially affecting the scheduling and handling of network requests. Nevertheless, because detailed CPU utilization measurements and raw packet-level data were unavailable, the results should be interpreted as an observed association rather than definitive proof that CPU contention alone caused the increase. During the CPU recovery test, the mean RTT decreased to 2.02 ms, approaching the baseline range.

Disk activity was also associated with an increase in RTT. The disk write workload produced a mean RTT of 2.48 ms, while the combined disk read/write workload recorded 2.67 ms. These values were higher than the full baseline mean of 1.84 ms. The repeated disk workload produced a mean of 2.59 ms, showing a similar pattern. The findings suggest that disk-intensive activity coincided with higher RTT in the tested environment, although the increase was smaller than that observed during the heavier network workload and combined CPU–network condition. Possible explanations include competition for system resources or other effects of concurrent disk activity, but the available measurements do not establish the specific mechanism responsible.

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume XI Issue X October 2026

After the disk workload was removed, the mean RTT decreased to 1.99 ms, which was close to the baseline measurements.

The network workload showed the clearest increasing pattern among the individual workload conditions. With one network stream, the mean RTT was 4.10 ms; with two streams, it increased to 5.26 ms; and with four streams, it reached 7.84 ms. The repeated four-stream test recorded a similar mean RTT of 7.69 ms, suggesting that the elevated measurements were reasonably consistent across the two recorded tests. The standard deviation also increased from 2.31 ms under one stream to 4.17 ms under four streams, indicating greater variation in response times as the number of streams increased. Packet loss rose from 0.2% under one stream to 0.8% under four streams, while the repeated four-stream test recorded 0.7% loss. This pattern is consistent with the possibility that increasing network activity placed additional demands on the tested network path or its processing resources. However, because achieved throughput, network utilization, and queueing behavior were not independently measured, the results cannot identify the precise cause of the increase. Following the removal of the network workload, the mean RTT decreased to 2.06 ms, returning close to the baseline range.

The combined workload tests further demonstrated that RTT differed according to the workload conditions applied. The CPU–disk condition recorded a mean RTT of 4.36 ms, a standard deviation of 2.37 ms, and packet loss of 0.3%. The CPU–network condition produced a higher mean RTT of 8.92 ms, a standard deviation of 4.88 ms, and packet loss of 1.1%. The CPU–network condition had the highest mean RTT, greatest reported variation, and highest packet-loss percentage among the recorded test conditions. Compared with the full baseline mean of 1.84 ms, its mean RTT was approximately 4.85 times higher. This result suggests that the simultaneous presence of CPU and network workloads coincided with more substantial performance changes than those observed during baseline operation or the CPU–disk condition. However, the experiment does not provide enough evidence to determine whether the increase resulted primarily from CPU activity, network traffic, or an interaction between them.

Overall, the findings indicate that RTT was not constant across the recorded experimental sequence. Individual CPU, disk, and network workloads were associated with higher mean RTT than the baseline, while the network tests showed an increasing pattern as the number of streams increased. The combined CPU–network condition recorded the highest mean RTT and packet loss. In contrast, the recovery measurements and final baseline verification returned close to the initial baseline range, with the final baseline mean recorded at 1.93 ms and no packet loss. These observations suggest that the measured performance changes were associated with the workload conditions rather than representing a consistently elevated RTT throughout the entire test sequence. Nevertheless, the study is descriptive and limited to the documented VMware-based environment. Further experiments with detailed resource-utilization measurements, raw packet-level observations, and additional independent repetitions would be needed to test these patterns more rigorously and determine the mechanisms behind them.

Technical interpretation of the combined CPU–network condition (additional discussion): the reported mean increased from 1.84 ms at baseline to 8.92 ms under combined CPU and network load. The difference is 7.08 ms; relative to baseline, this is approximately a 384.8% increase, and the combined-load mean is approximately 4.85 times the baseline mean. These calculations describe the reported values and do not by themselves identify the cause. Resource contention is one possible explanation because the virtual machines share the host's physical processor and networking resources. CPU scheduling delay is another possibility: under heavy demand, a virtual CPU may wait longer to be scheduled on the host processor, potentially delaying packet generation or response processing. Network queueing may also contribute if heavy iPerf3 traffic causes packets to wait in virtual NIC, virtual-switch, operating-system, or host networking queues.

The supplied materials include workload scripts but do not provide complete time-aligned CPU utilization, disk activity, or iPerf3 throughput measurements for the RTT runs. No VMware CPU-ready time, virtual-switch queue depth, interrupt latency, or other hypervisor-level metrics were supplied. These measurements are therefore not presented as collected variables in the results.

Implications of the Findings

For systems administrators, the results suggest that baseline ping measurements alone may not characterize performance during simultaneous workloads. If a service is sensitive to latency, testing should include representative CPU, disk, and network activity, and should record packet loss as well as RTT. A practical monitoring protocol should also record host utilization, VM allocation, network throughput, and workload start/stop times so that changes can be linked to specific system events. For students and researchers, the experiment illustrates the value of repeating baseline measurements, including recovery conditions, and reporting variation rather than only a mean. It also demonstrates why summary statistics should be accompanied by the raw observations or distributions whenever possible.

Administrators should correlate RTT with CPU utilization, RAM usage, disk activity, network throughput, and packet loss rather than interpret RTT in isolation. High RTT together with high CPU usage may indicate CPU contention; high RTT with high network throughput may indicate congestion or queueing; high RTT with packet loss may signal substantial resource pressure or network saturation; and high RTT during intense disk activity may justify investigating competition for shared storage resources. These combinations are diagnostic clues, not proof of a single cause.

The results can inform follow-up decisions such as adjusting VM CPU or RAM allocations, moving workloads to another VMware host, avoiding excessive co-location of resource-intensive VMs, scheduling heavy workloads during off-peak periods, applying bandwidth management, and investigating packet loss. Such changes should be made only after monitoring confirms the relevant bottleneck; the current results do not establish that any single allocation change will necessarily reduce RTT.

LIMITATIONS OF THE STUDY

The study has several limitations. First, complete packet-level Round-Trip Time (RTT) logs were not available for all experimental conditions, preventing independent recalculation of every reported summary statistic. Second, complete synchronized measurements of CPU utilization, disk activity, and iPerf3 network throughput were unavailable for all test conditions, limiting the analysis of relationships between resource utilization and RTT.

Third, the available records do not establish the number of independent replicate runs for every condition, and the experimental sequence was not documented as randomized. The exact VMware version/build and detailed host and guest hardware and software specifications were also not fully documented, which may affect reproducibility. In addition, separate RTT results for Background VM configurations A–E were unavailable; therefore, the effects of varying CPU and RAM allocations cannot be evaluated from the reported results.

Finally, the experiment was conducted in a small-scale VMware-based virtualized environment using Windows 10 virtual machines. Consequently, the findings are limited to the documented experimental environment and tested workload conditions and may not be directly generalizable to enterprise-level virtualization platforms or other network environments.

CONCLUSION

This study examined Round-Trip Time (RTT) under baseline, CPU, disk I/O, network, combined-workload, and recovery conditions in a VMware-based virtualized network environment. The reported descriptive statistics showed higher mean RTT under several workload conditions compared with the full baseline mean of 1.84 ms. The highest reported mean RTT was 8.92 ms under combined CPU–network activity, while the final baseline mean was 1.93 ms. These findings describe the recorded experimental sequence and do not establish a general causal relationship between workload intensity and RTT.

The recorded results indicate that several workload conditions were associated with increased RTT and packet loss. However, the findings should be interpreted descriptively because complete raw RTT logs and synchronized resource measurements were unavailable for independent verification. Furthermore, independent

run counts were not established for all conditions, and separate results for Background VM configurations A–E were not available. Consequently, the study does not establish which resource configuration provides the best RTT performance.

This study provides a foundation for conducting RTT experiments in a small-scale VMware-based laboratory environment. Future experiments should preserve raw RTT logs, collect time-aligned CPU utilization, disk activity, and iPerf3 throughput measurements, conduct clearly documented independent repetitions, and report results separately for each Background VM configuration. These improvements would support more reliable comparisons and a clearer assessment of how resource allocation and workload type relate to RTT.

RECOMMENDATIONS

Future experiments should document the host CPU model and core/thread count, physical memory and storage specifications, VMware product and version/build, guest operating system editions/builds, individual VM vCPU/RAM allocations, virtual NIC types, and VMnet1 network settings. The output of ipconfig /all should be recorded for each VM to ensure consistency among the network topology, measurement targets, commands, tables, and figure captions.

For every experimental condition, researchers should preserve the complete RTT output and maintain a machine-readable log containing the condition ID, run number, timestamp, target IP address, packets sent, received and lost, and individual probe RTT values. Time-aligned measurements of CPU utilization, disk active time, disk read/write throughput, and complete iPerf3 client/server output should also be collected. The number of independent runs per condition should be specified before testing, with a clear distinction between repeated probes within a run and independent experimental runs.

If configurations A–E were tested, report a separate row for every configuration, workload, and independent run, including run ID, probe counts, RTT statistics, loss, and paired resource measurements. No allocation effect can be inferred from the current unmapped summaries. Randomize or counterbalance future condition order and distinguish within-run packet variation from between-run variation.

REFERENCES

1. Aqasizade, H., Ataie, E., & Bastam, M. (2025). Experimental assessment of containers running on top of virtual machines. IET Networks. https://doi.org/10.1049/ntw2.12138
2. de Oliveira Filho, A. T., Freitas, E., do Carmo, P. R. X., Sadok, D. H. J., & Kelner, J. (2022). An experimental investigation of Round-Trip Time and virtualization. Computer Communications, 184, 73–85. https://doi.org/10.1016/j.comcom.2021.12.006
3. de Oliveira Filho, A. T., Freitas, E., do Carmo, P. R. X., Sadok, D. F. H., & Kelner, J. (2023). Measuring the impact of SR-IOV and virtualization on packet round-trip time. Computer Communications, 211, 193–215. https://doi.org/10.1016/j.comcom.2023.09.013
4. de Oliveira Filho, A. T., Freitas, E., do Carmo, P. R. X., Souto, E., Kelner, J., & Sadok, D. F. H. (2024). Analysis of SR-IOV in Docker containers using RTT measurements. Computer Communications. https://doi.org/10.1016/j.comcom.2024.107961
5. Fujimoto, K., Kaneko, M., Matsui, K., & Akutsu, M. (2022). KBP: Kernel enhancements for low-latency networking for virtual machine and container without application customization. IEICE Transactions on Communications, E105-B(5), 522–532. https://doi.org/10.1587/transcom.2021EBT0004
6. Silva, D., Rafael, J., & Fonte, A. (2023). Virtualization maturity in creating system VM: An updated performance evaluation. International Journal of Electrical and Computer Engineering Research, 3(2), 7–17. https://doi.org/10.53375/ijecer.2023.341
7. Wiedner, F., Helm, M., Daichendt, A., Andre, J., & Carle, G. (2024). Performance evaluation of containers for low-latency packet processing in virtualized network environments. Performance Evaluation, 166, 102442. https://doi.org/10.1016/j.peva.2024.102442
8. Zbořil, M., & Svatá, V. (2025). Performance comparison of cloud virtual machines. *Journal of Systems and Information Technology, 27*(2), 197–213. https://doi.org/10.1108/JSIT-02-2022-0040

Evidence category	Available information	Reporting note
RTT summaries	Condition-level statistics for TC-01–TC-20	Report the supplied values as summary results.
Measurement procedure	PowerShell/ICMP procedure described in the methodology	Keep commands consistent with the defined Sender-to-Receiver path.
Workload procedure	CPU, disk, and network workload scripts described	Do not infer unrecorded resource-utilization values.
Resource metrics	Complete synchronized CPU, disk, and iPerf3 series are not available for every condition	Discuss only quantitative measurements supported by records.

Configuration	Documented allocation	RTT result by configuration available?
A	1 vCPU, 1 GB RAM	Not provided in supplied results
B	1 vCPU, 2 GB RAM	Not provided in supplied results
C	2 vCPU, 2 GB RAM	Not provided in supplied results
D	1 vCPU, 3 GB RAM	Not provided in supplied results
E	2 vCPU, 3 GB RAM	Not provided in supplied results

TC ID	Test Scenario	Test Steps	Expected Result	Sample Actual Result
TC-01	Connectivity Check	1. Start Sender and Responder. 2. On Sender run historical record: ping 192.168.56.102 (target role unresolved).	Sender should receive replies from Responder with no timeout.	Passed. 4 packets sent, 4 received, 0% loss.
TC-02	Baseline Pilot	1. Background VM OFF. 2. On Sender run historical record: ping -n 100 192.168.56.102 (target role unresolved). 3. Record results.	100 packets should complete with low RTT and ideally 0% packet loss.	Mean 1.72 ms, Min 1 ms, Max 5 ms, Loss 0%.
TC-03	Full Baseline	1. Background OFF. 2. Run historical record: ping -n 1000 192.168.56.102 (target role unresolved). 3. Save output. 4. Calculate Median and SD.	Establish normal RTT with no background resource contention.	Mean 1.84 ms, Median 1.72 ms, Min 1 ms, Max 7 ms, SD 0.92 ms, Loss 0%.
TC-04	Baseline Repeat	1. Wait 2–5 minutes. 2. Keep Background OFF. 3. Repeat 1,000 pings.	Results should be reasonably close to TC-03.	Mean 1.91 ms, Median 1.80 ms, Min 1 ms, Max 8 ms, SD 0.97 ms, Loss 0%.
TC-05	Background VM Idle	1. Start Background VM. 2. Do not run workload. 3. Run 1,000 pings from Sender.	An idle Background VM should have little effect compared with baseline.	Mean 1.96 ms, Median 1.82 ms, Max 8 ms, SD 1.01 ms, Loss 0%.

TC-06	CPU Load	1. Start CPU stress on Background. 2. Verify CPU usage in Task Manager. 3. Run 1,000 pings.	CPU contention may increase RTT and variability compared with baseline.	Mean 3.27 ms, Median 3.05 ms, Min 1 ms, Max 14 ms, SD 1.86 ms, Loss 0.2%.
TC-07	CPU Load Repeat	1. Restart CPU stress. 2. Run another 1,000 pings. 3. Record results.	Results should show a similar trend to TC-06 if the effect is repeatable.	Mean 3.41 ms, Median 3.12 ms, Max 15 ms, SD 1.94 ms, Loss 0.3%.
TC-08	CPU Recovery	1. Stop CPU stress with Ctrl+C. 2. Wait 2–5 minutes. 3. Run 1,000 pings.	RTT should move back toward baseline after workload is removed.	Mean 2.02 ms, Median 1.88 ms, Max 8 ms, SD 1.05 ms, Loss 0%.
TC-09	Disk Write Load	1. Start repeated disk write workload on Background VM. 2. Verify disk activity. 3. Run 1,000 pings.	Disk activity may increase RTT slightly if host storage becomes contended.	Mean 2.48 ms, Median 2.30 ms, Min 1 ms, Max 11 ms, SD 1.32 ms, Loss 0.1%.
TC-10	Disk Read/Write Load	1. Run write/read/delete loop on Background. 2. Run 1,000 pings from Sender.	Read/write contention may produce more RTT variability than baseline.	Mean 2.67 ms, Median 2.42 ms, Max 12 ms, SD 1.45 ms, Loss 0.1%.
TC-11	Disk Load Repeat	1. Repeat disk workload. 2. Run another 1,000 pings.	Results should be broadly consistent with TC-09/10.	Mean 2.59 ms, Median 2.38 ms, Max 12 ms, SD 1.39 ms, Loss 0%.
TC-12	Disk Recovery	1. Stop disk workload. 2. Wait 2–5 minutes. 3. Run 1,000 pings.	RTT should return close to baseline.	Mean 1.99 ms, Median 1.85 ms, Max 8 ms, SD 1.00 ms, Loss 0%.
TC-13	Network Load – 1 Stream	1. On Responder run iperf3 -s. 2. On Background run historical command target: 192.168.56.103 (role requires verification) -t 1200 -P 1. 3. Run 1,000 pings from Sender.	RTT should increase while Responder handles additional network traffic.	Mean 4.10 ms, Median 3.74 ms, Max 19 ms, SD 2.31 ms, Loss 0.2%.
TC-14	Network Load – 2 Streams	1. Start iPerf server. 2. Run Background with -P 2. 3. Run 1,000 pings.	More traffic should generally create equal or higher RTT than one stream.	Mean 5.26 ms, Median 4.91 ms, Max 24 ms, SD 2.88 ms, Loss 0.4%.
TC-15	Network Load – 4 Streams	1. Start iPerf server. 2. Run historical command target: 192.168.56.103 (role requires verification) -t 1200 -P 4. 3. Send 1,000 pings.	Heavy network load is expected to cause the greatest network-related RTT increase.	Mean 7.84 ms, Median 7.12 ms, Max 36 ms, SD 4.17 ms, Loss 0.8%.

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume XI Issue X October 2026

TC-16	Network Load Repeat	1. Repeat TC-15 using four streams. 2. Record 1,000 RTT samples.	Result should show a similar high-load pattern.	Mean 7.69 ms, Median 7.01 ms, Max 35 ms, SD 4.02 ms, Loss 0.7%.
TC-17	Network Recovery	1. Stop iPerf3. 2. Wait 2–5 minutes. 3. Send 1,000 pings.	RTT should return toward baseline after network load stops.	Mean 2.06 ms, Median 1.91 ms, Max 9 ms, SD 1.08 ms, Loss 0%.
TC-18	CPU + Disk Combined	1. Start CPU stress. 2. Start disk read/write load. 3. Run 1,000 pings.	Combined resource contention may create higher RTT and variability than individual CPU or disk tests.	Mean 4.36 ms, Median 4.02 ms, Max 21 ms, SD 2.37 ms, Loss 0.3%.
TC-19	CPU + Network Combined	1. Start CPU stress. 2. Start iPerf3 4-stream traffic. 3. Run 1,000 pings.	Combined CPU and network contention may produce the highest RTT among the test conditions.	Mean 8.92 ms, Median 8.14 ms, Max 42 ms, SD 4.88 ms, Loss 1.1%.
TC-20	Final Baseline Verification	1. Stop all workloads. 2. Wait 5 minutes. 3. Run final 1,000 pings.	Final baseline should be reasonably close to the initial baseline, indicating the VM environment recovered.	Mean 1.93 ms, Median 1.79 ms, Min 1 ms, Max 8 ms, SD 0.98 ms, Loss 0%.