Instituto Tecnológico y de Estudios Superiores de Occidente

Reconocimiento de validez oficial de estudios de nivel superior según acuerdo secretarial 15018, publicado en el Diario Oficial de la Federación del 29 de noviembre de 1976.

Departamento de Electrónica, Sistemas e Informática
MAESTRÍA EN DISEÑO ELECTRÓNICO

USB Through Network Performance and Cost Analysis of Solution in Data Center Applications

---

TRABAJO RECEPCIONAL que para obtener el **GRADO** de **MAESTRO EN DISEÑO ELECTRÓNICO**

Presenta: María Fernanda Villagrán López

Director Dr. José Luis Chávez Hurtado

Tlaquepaque, Jalisco. Mayo 2025.

Dedicatoria:

A mi madre, a mi padre y a mi hermano: quienes jamás dudaron.

Summary

This work focuses on evaluating the performance, scalability, and usage model of USB over Ethernet solutions in data center applications. The research aims to assess how these solutions impact data transfer speeds and latency, determine their adaptability to growing data center needs, identify security implications, compare costs and complexity, investigate integration with existing infrastructure, and explore practical use cases and limitations.

The study involved extensive testing and analysis using various benchmarking tools, including Atto benchmark, and DiskSPD. The findings revealed that direct motherboard connections offer the best performance for USB devices, as expected, yet the SEH UTN Server Pro provides a comparable and slightly better alternative to front panel connections. Additionally, the research highlighted the importance of selecting devices with robust security and manageability features to fully leverage their capabilities in a network environment.

Contents

1 Introduction

1.1 Background

Connecting to devices from any location is crucial for modern businesses. Network-based data retrieval and equipment management streamline operations, saving both time and money. USB turns out to be one of the most popular options for smart devices and peripherals, offering a low-cost, flexible interface for communication with more complex devices. It simplifies direct connections to computers, finds the right drivers and applications and can provide power using the same connector.

USB is a protocol that is being continually updated and back-wards compatible that ensures straightforward and effective device connectivity. As this technology keeps growing in demand, new challenges and solutions arise. (Digi International Inc. 2025)

USB over Ethernet is a technology that allows USB devices to be shared and accessed over a network. USB signals are converted into Ethernet-compatible signals using software drivers or hardware adapters, allowing devices to communicate as if they were directly connected via USB. Key components of this technology include the USB device, adapter, network server, and client software. Benefits and applications of USB over Ethernet technology include increased flexibility, convenience, scalability, remote access, device sharing, and virtualization.

Enabling USB peripheral devices to connect over a Local Area Network (LAN) using Ethernet, cuts the need for a direct attachment to a local host computer when the setup must be replicated in multiple separate locations. A USB hub would ensure robust and reliable connections between peripheral devices and a PC, free from the usual physical constraints. Another benefit to having a USB hub in lieu of a physical host is that you can enhance the security of your host PC and network by relocating the PC to a more secure and protected environment, while using USB over ethernet solutions to extend connectivity to your peripheral devices. Some of the applications in various industries include testing, validation, logistics, operations, retail, and software physical key management, among others.

Figure 1. AI generated image depicting USB over ethernet concept, with USB devices connecting through the network via a USB hub.

With more USB devices being deployed as part of business solutions (see Fig.1 with a generated image to exemplify the vast number of devices and ways of connecting), the pain of scaling starts to arise with the distance that needs to be covered for devices to operate optimally both from a bandwidth and power perspective. Typically, adding extra ports to connect more USB devices would mean installing additional computers close to the equipment, given the protocol specifications: USB 2.0 devices must be placed within 5 meters of host, and USB 3.0 are recommended to be placed within 3 meters of host. Which brings the added challenge of having to deploy and manage all the hosts used to interface with the USB devices. As an example, if we observe Fig. 2, it depicts a layout where users can connect to hosts deployed on a network (represented by a cloud, which would include the networking components required for necessary interconnects) which are respectively physically routed to the USB devices, which means that user's hosts as well as local systems need to be managed and interconnected to be able to access all different resources. On the other hand, if we look at Fig.3 we see the USB-To-Network server hub functioning as interface for all USB devices, thus no longer needing additional local hosts to connect and maintain.

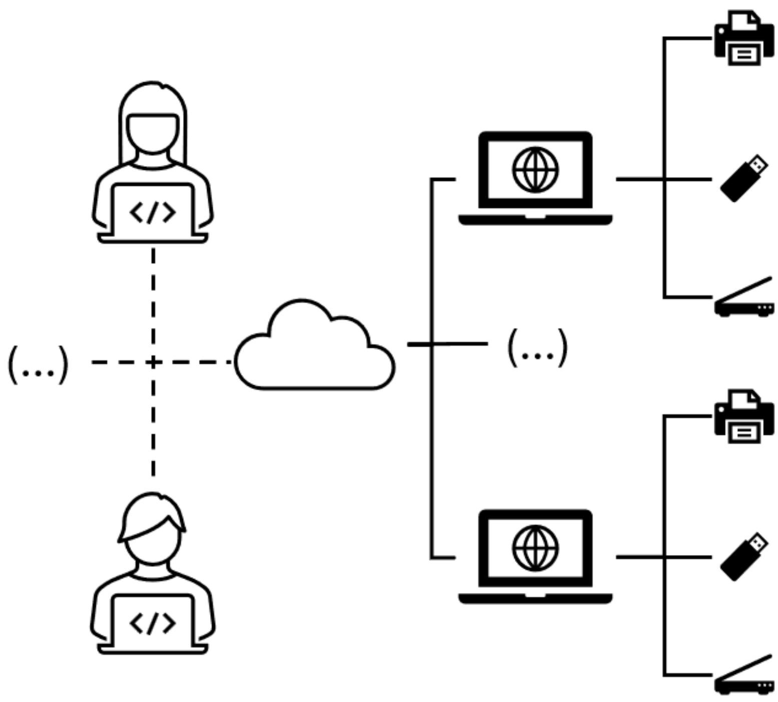


Figure 2. Typical layout with hosts in physical proximity of USB device and requiring an added PC to access remotely.

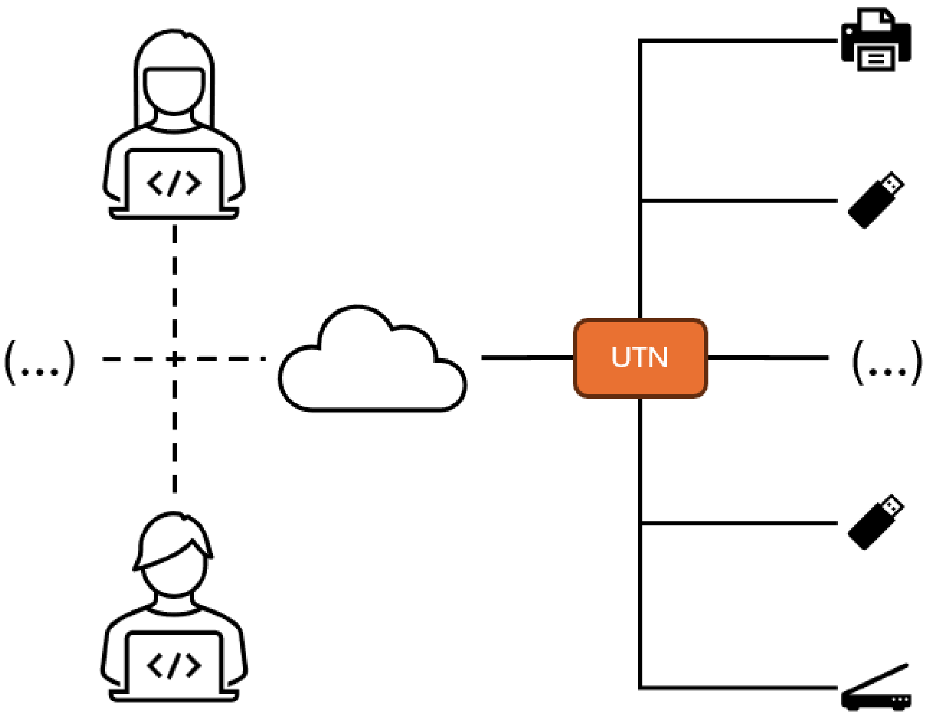


Figure 3. Proposed layout with UTN (USB Through Network) Server that allows for remote access to multiple USB devices.

1.2 Objectives

The objectives of this work focus on evaluating the performance, scalability, security, cost, and compatibility of USB over Ethernet solutions, focusing on data center applications. The research aims to assess how these solutions impact data transfer speeds and latency, determine their adaptability to growing data center needs, identify security implications, compare costs and complexity, investigate integration with existing infrastructure, and explore practical use cases and limitations.

- Evaluate Performance:
  - Assess how USB over Ethernet solutions impact data transfer speeds, latency, and overall performance compared to traditional USB connections.
- Analyze Scalability and Flexibility:
  - Determine the extent to which USB over Ethernet can be scaled and adapted to meet the growing and changing needs of data centers.
- Examine Security:
  - Identify potential security implications and benefits associated with using USB over Ethernet, including data protection and network security.
- Assess Cost and Complexity:
  - Compare the costs and complexity of implementing USB over Ethernet solutions versus traditional USB connections, including hardware, setup, and maintenance costs.
- Review Integration and Compatibility:
  - Investigate how well USB over Ethernet solutions integrate with existing data center infrastructure and their compatibility with various USB devices and systems.

- Identify Use Cases and Limitations:
    - Explore practical applications of USB over Ethernet in data centers and any limitations or challenges associated with these solutions.

1.3 Scope

The work conducted was based on a setup designed specifically for proof-of-concept measurements. The setup was based on a usage model for server validation, where the typical workspace would have a PC (also referred to as host)

Application focus of this research corresponds to Data Center Applications, where some of the devices used include:

- SPI programmers
- FPGA programmers
- Protocol analyzers (Beagle/Aardvark),
- extract crash dumps (ITP),
- read AC/DC signals,
- or remotely interact with jumpers.

For a comprehensive cost analysis of a larger-scale solution, it is essential to recalculate the calculations according to vendor specifications. Additionally, other factors such as network topology and various implementation details could significantly impact on the overall feasibility and effectiveness of the solution.

2 Technology Overview

2.1 USB over Ethernet Technology Overview

USB Over Ethernet is a technology that allows USB devices to be shared and accessed over a network. To do this, USB signals are converted into Ethernet-compatible signals using SW drivers or HW adaptors to then transmit over the network, allowing devices to communicate as if they were directly connected via USB.

Key Components

- USB Device: The peripheral device (e.g., USB flash drive, printer) connected to the network.
- Adapter: Hardware or software that converts USB signals to Ethernet signals.
- Network Server: Hosts the USB over Ethernet device and manages the network connection.
- Client Software: Installed on the client device to access and control the shared USB device.
    - For our POC the SEH UTN Server encompasses the last 3 components of the solution by being adapter, server and offering all SW required.

Benefits & Applications

- Increased Flexibility: Devices can be shared and accessed from anywhere on the network.
- Convenience: Eliminates the need for physical proximity between devices and users.
- Scalability: Supports multiple devices and users over a network.
- Remote Access: Access USB devices from remote locations.
- Device Sharing: Share USB devices among multiple users or systems.
- **Virtualization**: Use USB devices in virtual environments.

2.2 Solutions available in the market

For the scope of the project, a compact and low-cost server is sought, and options were researched and discarded, to come up with a brief list of 2 top runners: SEH UTN Server Pro vs. Silex USB Device Server

2.2.1 Silex USB Device Server DS-510

One of the lowest cost options found with required features is the Silex USB Device Server. (SILEX TECHNOLOGY 2025).

Specifications:

- **Ports**: 2 USB 2.0 ports
- **Network**: 10/100/1000 LAN (Gigabit Ethernet)
- **Data Throughput**: High-speed data transfer rates
- **Security**: Supports secure device connections over the network
- **Compatibility**: Compatible with various USB devices, including printers, scanners, and storage devices
- **Management**: Silex SX Virtual Link software for device management
- **Processor**: 800 MHz
- **Form Factor**: Desktop
- **Dimensions**: 1" height, 3.9" width, 3.9" depth
- **Weight**: Approximately 3.39 oz
- **Warranty**: Typically, 2 years
- **Price**: $124.91

But there are a couple of drawbacks:

1. Only USB 2.0
2. No documentation about security in the application – risky

2.2.2 SEH UTN Server Pro

Another option is the UTN Server by SHE Technology. (USB over Network 2024)

- **Ports**: 2 USB 3.2 Gen 1 (SuperSpeed) Ports
- **Network**: Gigabit Ethernet (10BaseT/100BaseTX/1000BaseT)
- **Data Throughput**: Up to 80 MB/s
- **Security**: Encrypted connections via LAN, VPN, VLAN, and the Internet
- **Compatibility**: Supports a wide range of USB devices, including RDX drives, scanners, medical devices, and more
- **Management**: SEH UTN Manager software for easy device management
- **Additional Features**: Isochronous USB mode for audio-video data transfer, point-to-point connection, and energy-saving options
- **Warranty**: 36 months (extendable to 60 months)
- Additional features:
    - SEH UTN Manager (connection management) for:
        - Microsoft Windows (32/64-Bit; Windows 10 or later, Windows Server 2016 or later)
        - macOS: macOS 12 (Monterey), 13 (Ventura), 14 (Sonoma) or later
        - Linux (Ubuntu 22.04, Debian 12, Oracle 9, CentOS Stream 9, Open-SUSE Leap 15.5) [1]
    - SEH Product Manager (network device management) for:
        - Microsoft Windows (32/64-Bit; Windows 10 or later, Windows Server 2016 or later)
        - macOS 12 (Monterey), 13 (Ventura), 14 (Sonoma) or later
    - utnserver Control Center
        - integrated user interface for configuration, administration, monitoring and maintenance via browser
    - Configuration services: Email, SNMP
    - Regular software updates, technical support worldwide
    - Encryption: TLS 1.0/1.1/1.2/1.3, HTTPS
    - Authentication: 802.1X (EAP-MD5, EAP-TLS, EAP-FAST, EAP-TTLS, PEAP)

- Certificate management: self-signed certificate, certificate request, CA certificate, PKCS#12 ceritificate, S/MIME-certificate
  - SNMPv3
  - VLAN (802.1Q)
  - TCP-port access control
  - Device access control (password protection, session login)
  - USB port access control (port key control, device assignment, USB port deactivation)
  - USB HID class blocking
- **Price**: Approximately $257[1]

Comparison:

Feature	**SEH UTN Server Pro**	**Silex USB Device Server (DS-700)**
Ports	2 USB 3.2 Gen 1 (SuperSpeed) Ports	2 USB 2.0 ports
Network	Gigabit Ethernet (10BaseT/100BaseTX/1000BaseT)	Gigabit Ethernet (10/100/1000 Mbps)
Data Throughput	Up to 80 MB/s	High-speed data transfer rates
Security	Encrypted connections via LAN, VPN, VLAN, and the Internet	Secure device connections over the network
Compatibility	Wide range of USB devices, including RDX drives, scanners, and more	Various USB devices, including printers, scanners, and storage devices
Management Software	SEH UTN Manager	Silex SX Virtual Link
Additional Features	Isochronous USB mode, point-to-point connection, energy-saving options	Isochronous transfer mode, ideal for audio and video applications
Warranty	36 months (extendable to 60 months)	Typically 2 years
Price	$257	$124.91

Key differences:

- USB Ports available: SEH UTN Server Pro has USB 3.0+ Ports.
- **Data Throughput**: SEH UTN Server Pro specifies up to 80 MB/s, while Silex focuses on high-speed transfers without a specific rate.
- **Security**: Both offer secure connections, but SEH UTN Server Pro provides more detailed security features.

- Price: $100+ higher price on SEH UTN manager.

In conclusion, given the fact that it has USB 3.0 ports, much detailed documentation, security features, and having collaborated with it before, the SEH UTN manager was selected as the solution to use for this project.

3 Methodology

HW setup will be as follows, the diagram depicts the connections per experiment.

3.1 Experimental setups:

1. Local USB 3.0 connection has Benchmark running on the Desktop PC and measuring the performance of the USB drive attached to the USB 3.0 port of the motherboard.

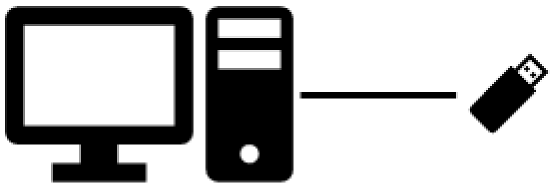


Figure 4

2. Local USB 2.0 connection has Benchmark running on the Desktop PC and measuring the performance of the USB drive attached to the USB 2.0 port attached to the front panel.

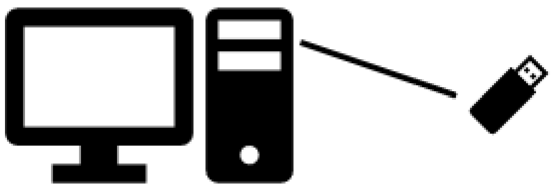


Figure 5

3. UTN Local Connection describes the setup where the USB is physically plugged to the SEH UTN server, the UTN server is wired through ethernet to local Modem, with a reserved IP, and the Benchmark runs from the Desktop PC wired through ethernet to the same Modem and with a reserved IP within the same local network.

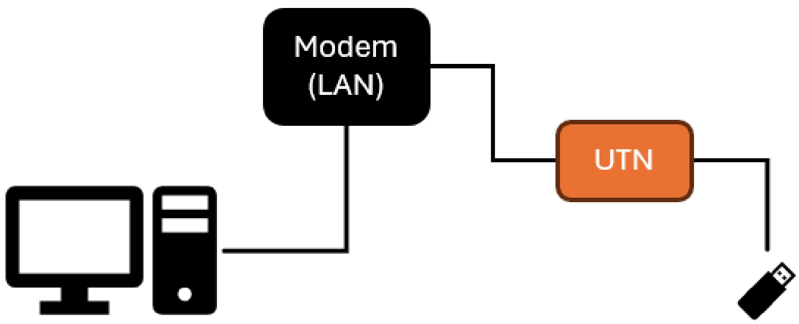


Figure 6

4. UTN Wi-Fi Connection describes a setup where the USB is physically plugged to the SEH UTN server, the UTN server is wired through ethernet to local Modem, with a reserved IP, and the Benchmark runs from a laptop with wireless Wi-Fi connection to the internet within the same local network.

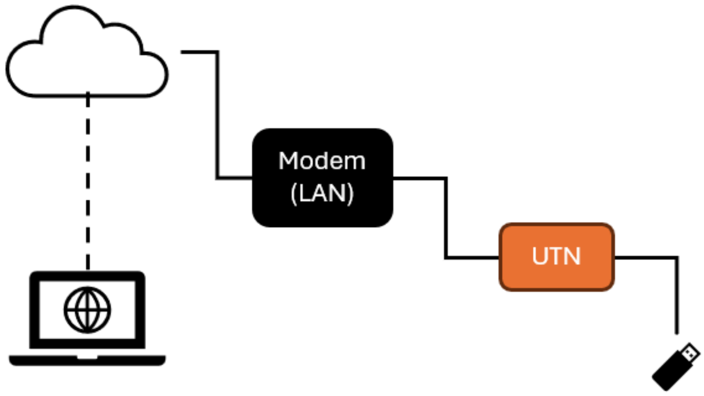


Figure 7

It is particularly important to note that there is significant difference between a USB port on the motherboard and one on the front panel primarily lies in their physical location and potential performance characteristics.

A USB port on the motherboard is directly integrated into the main circuit board of the computer. This integration often results in more stable and faster data transfer rates because the connection is shorter and more direct. Motherboard USB ports are typically used for devices that require consistent and high-speed data transfer, such as external hard drives or high-performance peripherals.

On the other hand, a USB port on the front panel is connected to the motherboard via internal cabling. This can sometimes lead to slightly slower data transfer rates due to the longer connection path and potential interference from other internal components. Front panel USB ports are convenient for connecting devices that are often plugged in and unplugged, such as USB flash drives or temporary peripherals.

4 Performance metrics analyzed

To measure the impact of the network on storage performance, you should focus on three key metrics: IOPS (Input/Output Operations Per Second), throughput, and latency. Here's a brief overview of each:

- IOPS: This measures the number of read and write operations a storage device can perform per second. High IOPS indicates the system's ability to manage a high volume of data transactions. (Engelbert 2024)
- Throughput: This refers to the amount of data transferred to and from the storage medium within a given period, typically measured in megabytes per second (MB/s). Higher throughput means the system can manage larger data transfers more efficiently.
- Latency: This is the delay before a data transfer begins following an instruction for its transfer. Lower latency results in faster system responsiveness, which is crucial for real-time data access. (Carter 2024)

To effectively measure these metrics, you can use benchmarking tools and software designed for storage performance analysis. Some popular tools (among many others, the ones used can be found in section 6) include:

- Iometer: A versatile benchmarking tool that can measure IOPS, throughput, and latency.
- FIO (Flexible I/O Tester): tool for testing storage performance, widely used in Linux environments.
- CrystalDiskMark: effective for measuring storage performance on Windows systems.

By analyzing these metrics, you can identify performance bottlenecks and optimize your network and storage configurations for better performance. (Dancuk 2023)

5 Disk Benchmarking

A disk benchmark works by performing a series of controlled read and write operations to a storage device, measuring the time it takes to complete these operations, and then using that data to calculate the disk's performance metrics like read/write speed, IOPS (Input/Output Operations Per Second), and access time, allowing for comparison between different drives or to identify potential performance issues.

Key aspects of a disk benchmark:

- Test patterns. Different patterns of data access are used to simulate various real-world workloads, including sequential reads/writes (large continuous data transfers) and random reads/writes (accessing data in scattered locations on the disk).
- Queue depth. This setting simulates how many read/write requests can be queued up at once, which is important for understanding performance under heavy load.
- File size. The size of the data chunks used in the test can impact results, with smaller file sizes often testing random access performance while larger test files sustained transfer rates.
- Test duration. Benchmarking tools typically run for a set duration to ensure enough data is transferred for accurate results, while also considering potential caching effects.

For this use case, to measure the impact of the network on storage performance, we should focus on three key metrics: IOPS (Input/Output Operations Per Second), throughput, and latency.

6 Benchmarking and other SW Tools Used

6.1 UTN Software requirements:

SEH UTN Manager (SEH Computertechnik GmbH 2025)

Version link: SEH UTN Manager Version 4.0.7 for Windows; Released: 2024-05-23

All management and accessing of USB dongles via the Dongle Server are done using the 'UTN Manager' software tool. The UTN Manager is installed on all clients from which access to USB dongles is intended.

6.2 ATTO Disk Benchmark Windows

ATTO created a Disk Benchmark for Windows to help measure storage system performance. As one of the self-proclaimed top tools in the industry, Disk Benchmark helps evaluate performance in hard drives, solid state drives and storage systems.

The following article[1] was used as reference to pinpoint the expected performance of the storage device (USB key) that would be used for the scope of this project using similar tools.

ATTO Disk Benchmark for Windows. (ATTO 2025)

Widely accepted Disk Benchmark for Windows to help measure storage system performance. As one of the top tools in the industry, Disk Benchmark identifies performance in hard drives, solid state drives and storage systems. Use the tool that top drive manufacturers use to build and assess drives to tune your storage system for optimal performance.

[1] (T S 2022)

6.3 DISKSPD

DISKSPD is an I/O generating command-line tool designed for micro-benchmarking. This tool is particularly useful for individuals setting up a cluster or physical server, whether for web hosting environments or virtual desktops for employees. Before deploying an actual application, it is often beneficial to simulate a test scenario, which can be challenging in real-world conditions. DISKSPD addresses this need by allowing users to customize synthetic workloads and test applications prior to deployment. The tool provides flexibility to configure and adjust parameters to create scenarios that closely resemble real workloads, offering insights into system capabilities before deployment. Fundamentally, DISKSPD performs a series of read and write operations.

While DISKSPD may struggle with emulating complex workloads, it excels in scenarios where the workload is not closely approximated by a single-threaded file copy. It serves as a simple tool that produces acceptable baseline results for various applications. (Microsoft 2025)

7 Test Results with ATTO Benchmark

This test does iterations on different I/O sizes (seen in vertical axis from 512B through 64MB) used to transfer the file (which had same size through all the tests: 32GB) to calculate the transfer rate for reads and writes (represented by gray and red bars respectively) in two different speeds: Bytes/s and IO/s.

7.1 Direct connection in PC USB port in motherboard (see Fig.4 in section 3).

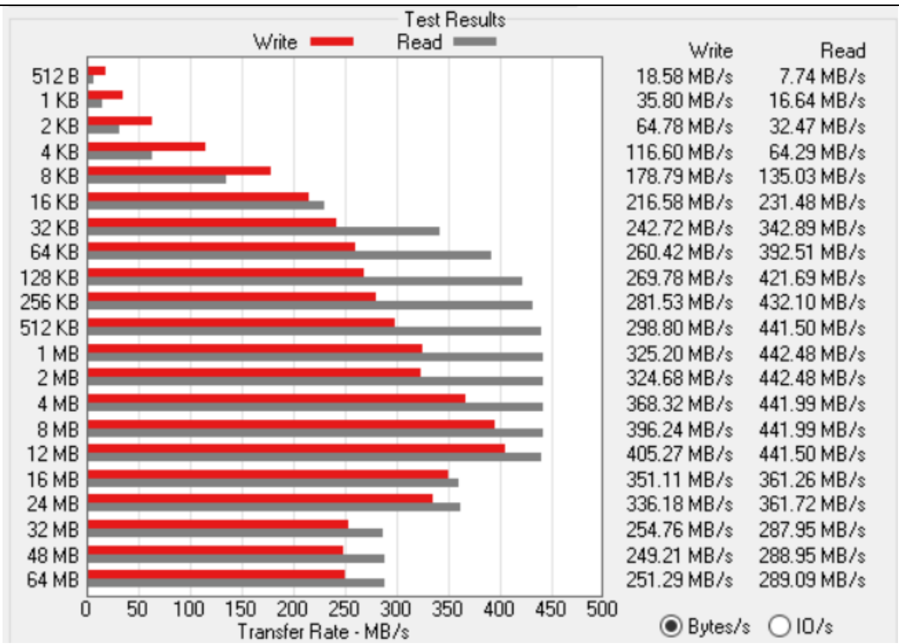


Figure 8. Read (gray bars) and Write (red bars) speed in Bytes/s with a file size of 32GB.

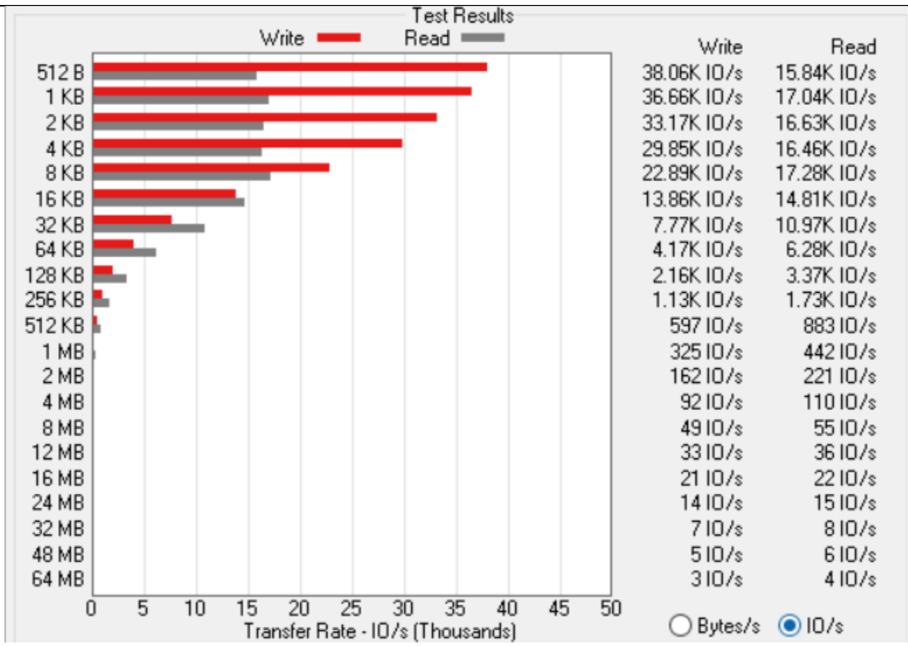


Figure 9. Read (gray bars) and Write (red bars) speed in IO/s with a file size of 32GB.

7.2 Connection to USB port in front panel (see Fig.5 in section 3)

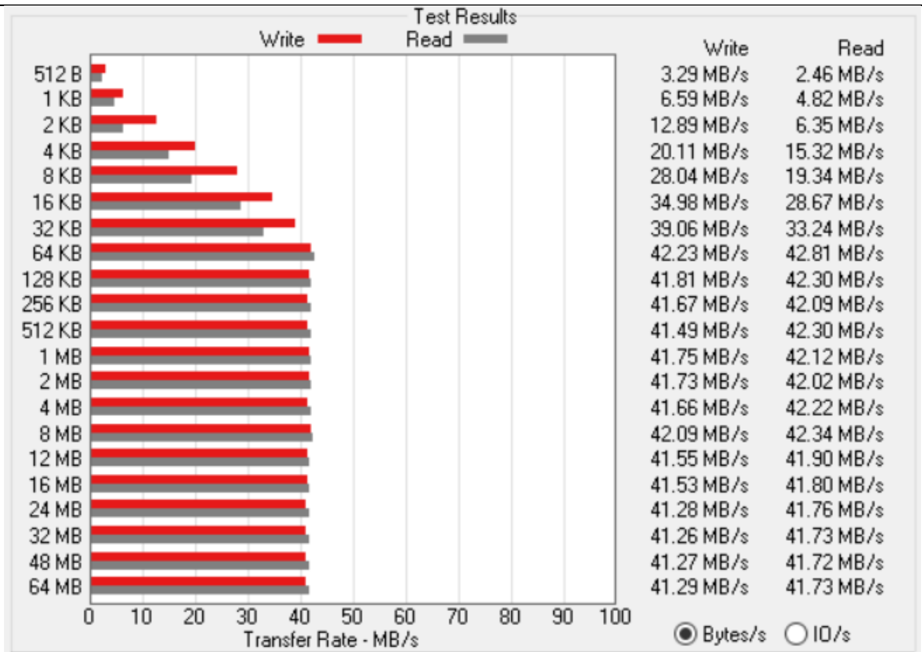


Figure 10. Read (gray bars) and Write (red bars) speed in Bytes/s with a file size of 32GB.

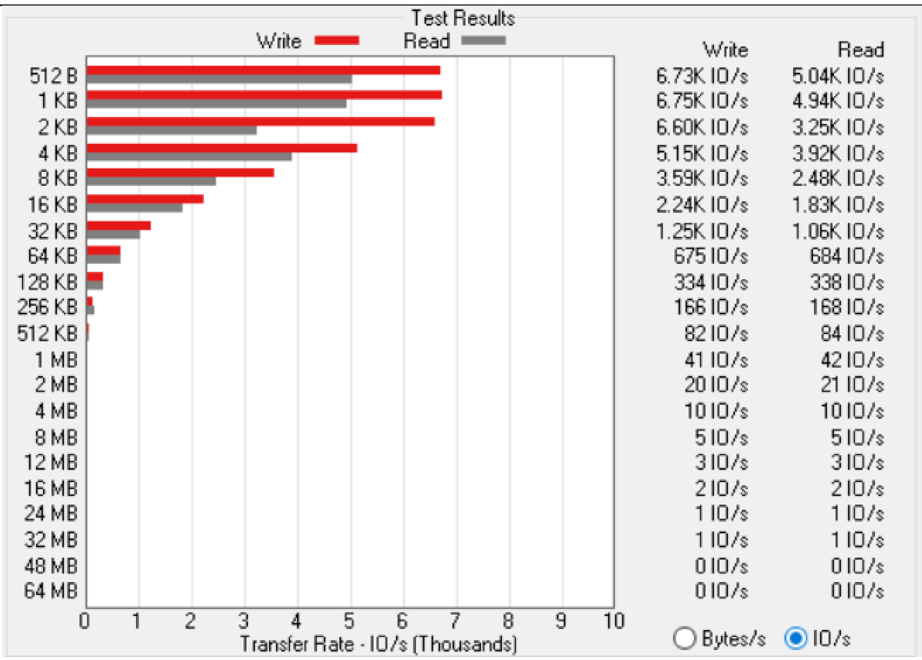


Figure 11. Read (gray bars) and Write (red bars) speed in IO/s with a file size of 32GB.

7.3 Connection in utnserver Pro from system connected via ethernet to same router

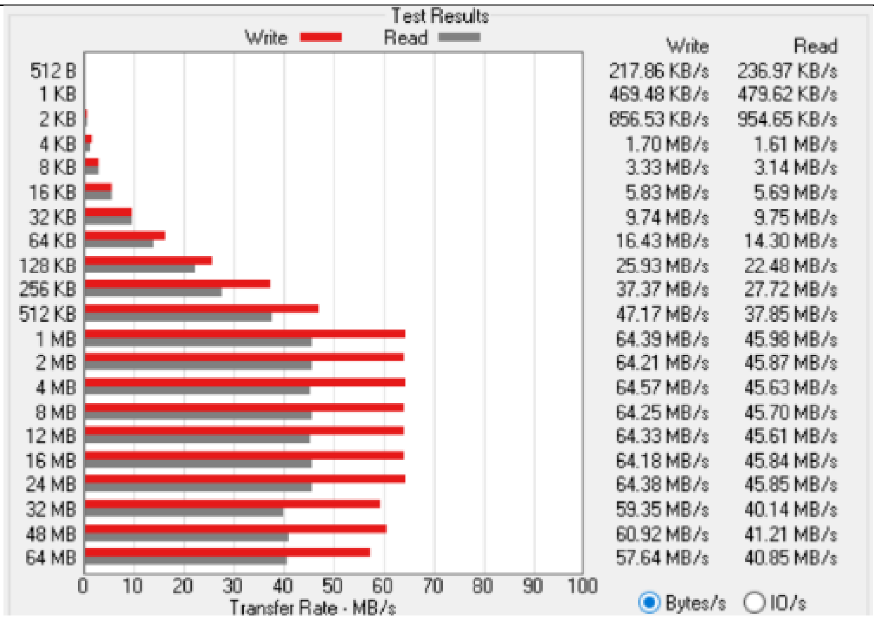


Figure 12. Read (gray bars) and Write (red bars) speed in Bytes/s with a file size of 32GB.

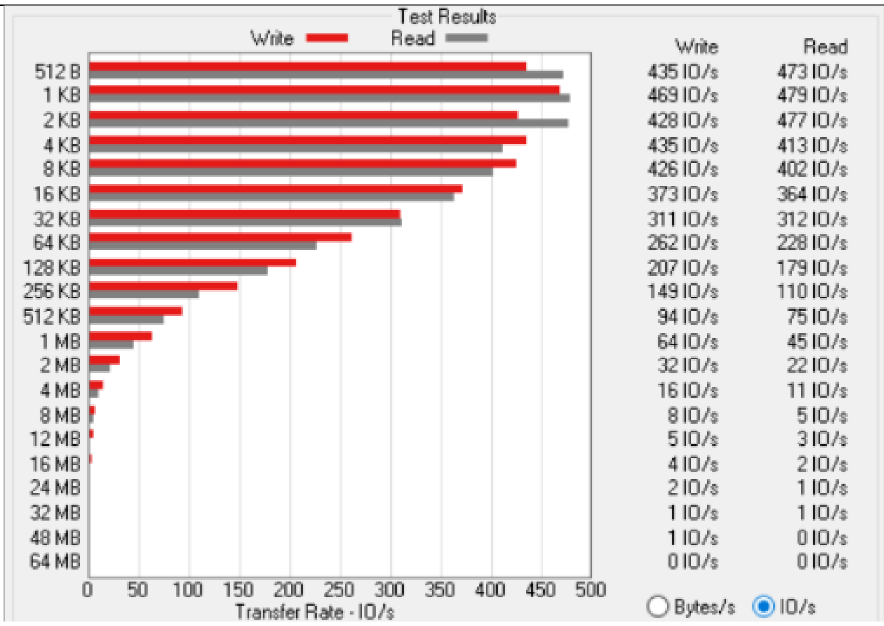
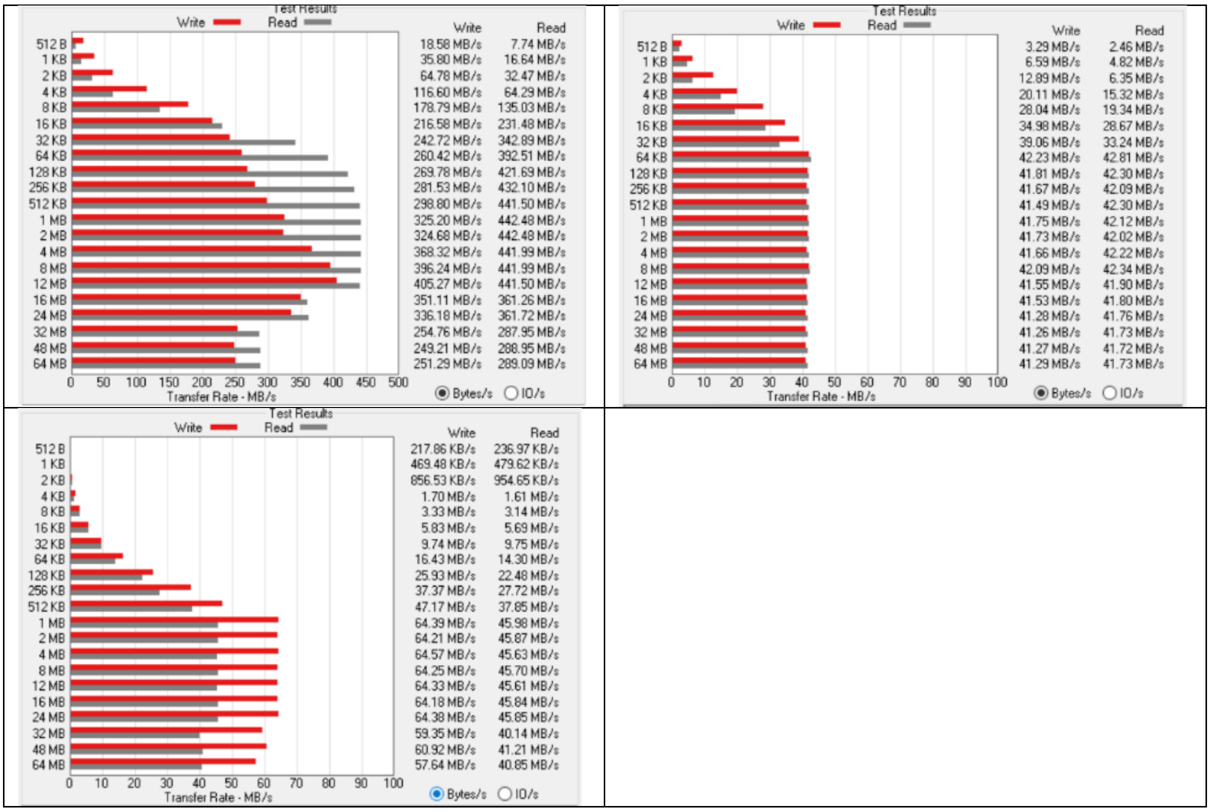


Figure 13. Read (gray bars) and Write (red bars) speed in IO/s with a file size of 32GB.

7.4 Side-to-side Comparison



Figures from tests with Direct connection in motherboard, front panel and via ethernet. See Figures 8 (top left), 10 (top right) & 12 next to each other. Just by looking at the results it stands out by just analyzing the scale on the X-axis, that the transfer rate in Bytes/s is in a scale of 500 MB/s for Fig. 8 corresponding to a direct connection to the motherboard, whereas for Fig. 10 & 12 the scale goes only up to 100 MB/s.

Similarly, rate in IO/s for Fig.9 is up to 60 thousands, yet for the rest goes only to 10 or 5 thousand IO/s. Alas, the results as they are it is not easy to compare directly, as the tool doesn't provide a way to digest and post-process data, which is why additional testing was done with DiskSpd (see next section.)

7.5 Connect through WiFi to utnserver from system in local network

It is worth noting that when Wi-Fi was used to measure the storage performance, as suggested in Fig. 7 of the Methodology setup the test was not viable. The performance is so low that the utility was unable to initialize test file to obtain any comparable results, even when file size was reduced to be able to load and initialize, the test was not able to finish for all the IO transfer sizes required.

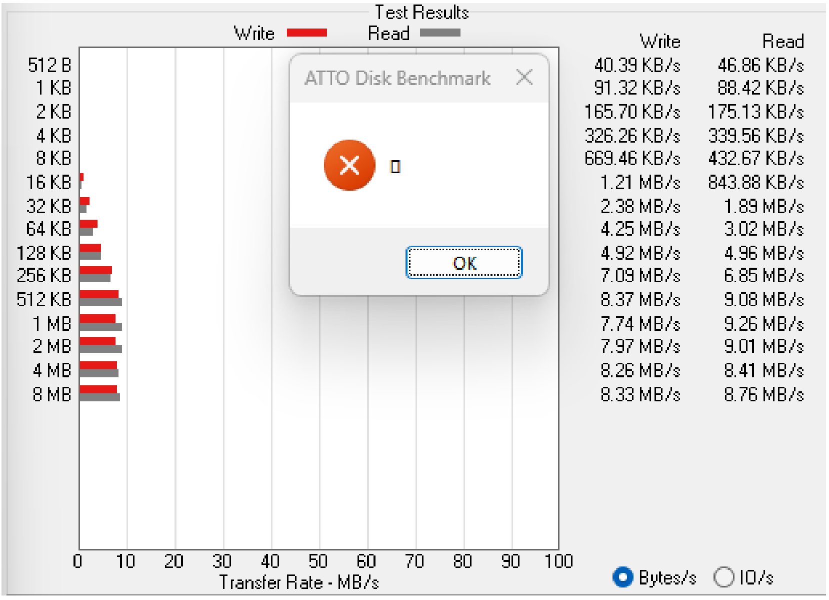


Figure 14. Failed test though Wi-Fi setup. (see Fig. 7)

8 Test Results with DiskSpd

Testing was done using the basic installation of DiskSPD. (Microsoft 2025)

8.1 Test Parameters:

The commands used was, with iterations running on different block sizes to gather expected plots.

*diskspd.exe -t1 -o4 **-b4K** -r4k -w50 -d60 -Sh -D -L -c256M D*

Where the parameters indicate:

- **-t1**: number of threads per target/test file, in this case 1 thread was specified.
- **-o4**: number of outstanding I/O requests per target per thread, also known as queue depth.
- **-b4K**: block size in bytes, KiB, MiB, or GiB. In this case, 4K block size was used to simulate a random I/O test.
- **-r4K**: This indicates the random I/O aligned to the specified size in bytes, KiB, MiB, Gib, or blocks. Common 4K byte size was used across.
- **-w50**: This specifies the percentage of operations that are write requests (-w0 is equivalent to 100% read). In this case, 50% read/writes were used.
- **-d600**: This specifies the duration of the test, not including cool-down or warm-up. The default value is 10 seconds, but we recommend using at least 60 seconds for any serious workload. In this case, 60 seconds were used.
- **-Suw**: Disables software and hardware write caching (equivalent to **-Sh**).
- **-D**: Captures IOPS statistics, such as standard deviation, in intervals of milliseconds (per-thread, per-target).
- **-L**: Measures latency statistics.
- **-c256M**: Sets the sample file size used in the test. It can be set in bytes, KiB, MiB, GiB, or blocks. In this case, a 5 GB target file was used.
- **D**: to point test at D: drive

8.2 Consolidated results in tabular form:

See Appendix 1 for sample results, test run, and scripts used to consolidate.

	Direct MB		Front Panel		UTN	
Block Size	Total IO MiB/s	Total IO IOPS	Total IO MiB/s	Total IO IOPS	Total IO MiB/s	Total IO IOPS
1KiB	16.26	33332.89	5.14	10515.22	0.73	1494.17
2KiB	32.45	33262.95	10.08	10317.1	1.48	1511.35
4KiB	64.54	33072.54	16.68	8533.35	2.34	1193.94
8KiB	122.57	31409.8	23.76	6077.48	4.75	1211.47
16KiB	203.9	26113.36	32.87	4203.55	8.74	1117.11
32KiB	277.49	17757.96	37.57	2395.78	14.35	917.34
64KiB	331.48	10601.17	41.61	1327.31	20.94	669.22
128KiB	391.94	6263.19	41.63	665.58	33.97	543.36
256KiB	433.3	3466.51	41.21	328.99	44.65	356.53
512KiB	436.11	1739.52	41.44	165.58	51.56	207.03
1MiB	442.51	881.6	41.47	83.06	51.44	103.59
2MiB	443.15	442.93	41.4	41.22	51	50.21
4MiB	443.34	222.33	41.46	20.06	52.07	25.45
8MiB	443.57	111.56	41.59	10.28	51.86	12.8
16MiB	443.62	54.64	41.6	5.13	52.26	6.4
32MiB	443.64	27.23	41.6	2.62	52.26	3.08
64MiB	442.65	13.58	41.6	1.17	52.26	1.57

8.3 Plotted results

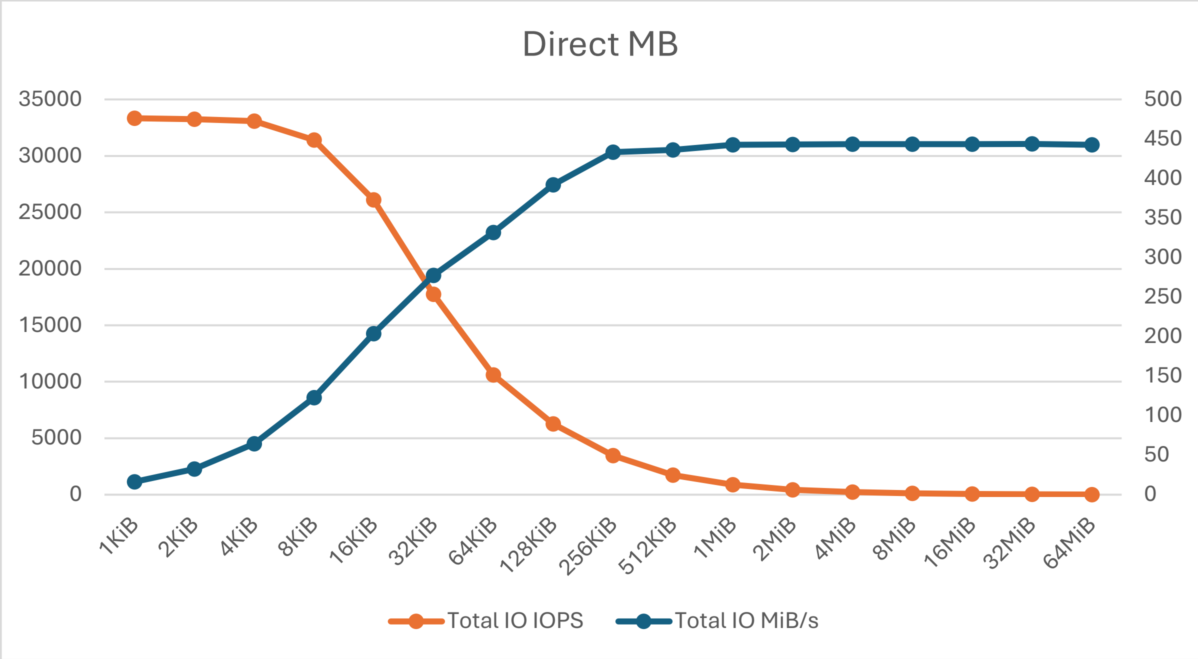


Figure 15. Total IO (Read + write) for Direct connection to MB, analog to Figs 8 & 9.

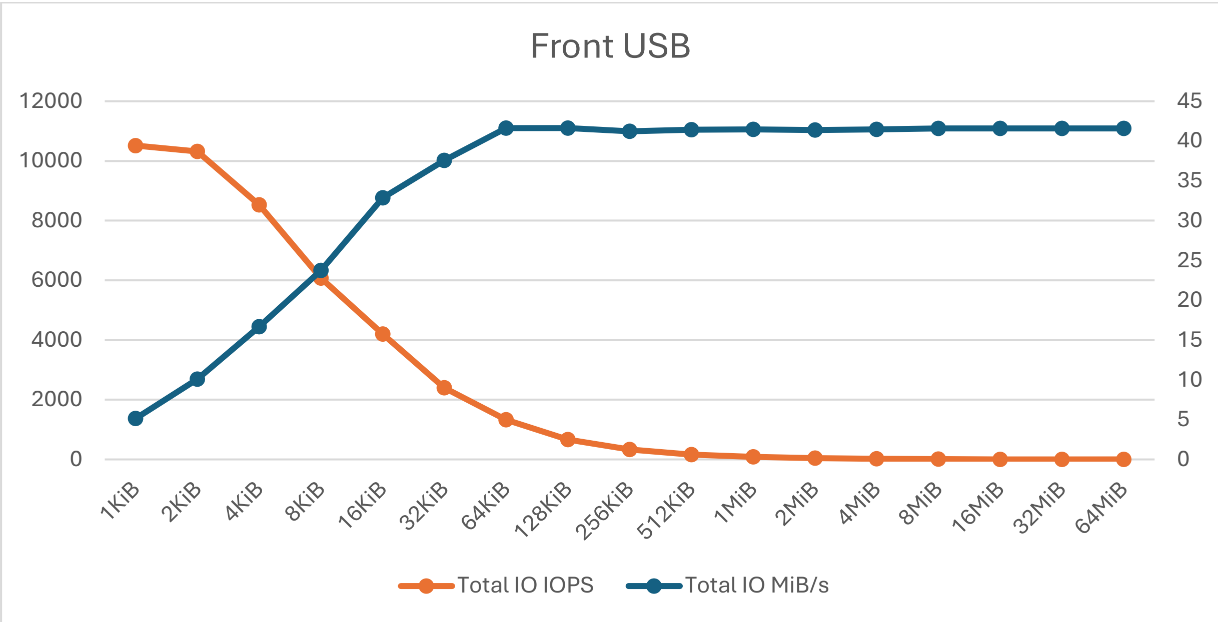


Figure 16. Total IO (Read + write) for USB connection through front panel, analog to plots Figs 10 & 11.

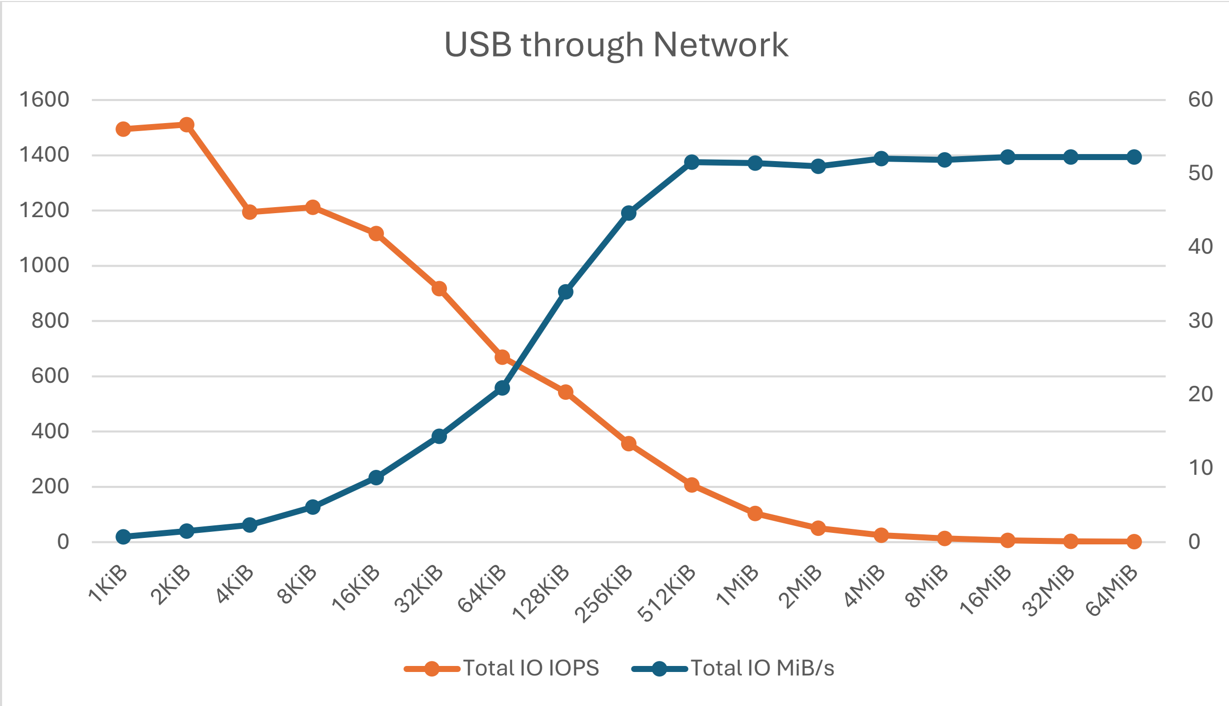


Figure 17.. Total IO (Read + write) for connection via UTN, analog to Figs 12 &13.

See section 9.2 for the analysis of these results.

9 Conclusions

9.1 Key Findings

9.1.1 Hardware Setup

The process of defining the test setup for this research presented significant challenges, both from a hardware and software perspective. The initial phase involved an extensive search for the optimal device capable of managing USB devices over the network. This search revealed a plethora of USB-to-Ethernet servers and extenders available on the market. However, it became evident that many of these extenders lacked the necessary security and manageability features required to fully leverage capabilities for various applications.

The findings of this research underscore the importance of selecting the right device for managing USB devices through the network. While numerous USB-to-Ethernet extenders exist, they often fall short in providing comprehensive management and security features. This limitation can hinder the effective utilization of these devices in real-world scenarios, where robust security and manageability are paramount.

9.1.2 Features Required

Also linked with the device selection lies the need for a device that not only facilitates the connection of USB devices over the network but also offers detailed security features and manageability options. The SEH UTN Server Pro emerged as the preferred solution due to its superior security features, detailed documentation, and proven performance in managing USB devices over the network. This device was selected for its ability to provide encrypted connections, comprehensive device management, and compatibility with a wide range of USB devices.

Overall, the conclusions drawn from this research emphasize the critical role of security and manageability in the selection of USB-to-Ethernet extenders. The SEH UTN Server Pro stands out as a reliable and effective solution, offering the necessary features to harness the full potential of USB devices in networked environments. Future research should continue to explore and evaluate the performance, scalability, and security implications of USB over Ethernet solutions to further enhance their application in various industries. (See Apendix 10.2 for an example for TCOS analysis)

9.1.2 Benchmarking Software

Another issue encountered during the test setup was finding the proper benchmarking software to obtain and post-process the results. Initially, tools such as Atto benchmark and IOmeter, which have graphical user interfaces, were used. However, these tools were unable to provide results in a format suitable for post-processing. Consequently, testing was conducted using various software until DiskSPD, a Windows application, was identified as the optimal solution. DiskSPD allowed for the output of results to a CSV format, which could then be processed to generate the same plots observed in the GUIs of other software tools. The post-processing scripts used for this purpose have been included in the annex section of this work.

9.2 Results Analysis

9.2.1 Overall performance

The analysis of the plots in section 8.3 reveals several critical insights into the performance of USB devices connected through various methods. Firstly, it is evident that the performance for Total IO in MiB/s (represented by the blue plot lines) is significantly better when the USB device is connected directly to the motherboard. Specifically, the performance is observed to be ten times (10x) better compared to using a port in the front panel of the chassis or via the UTN server. This finding underscores the superiority of direct motherboard connections in terms of data transfer rates and overall efficiency, making it the preferred option across all testing scenarios.

9.2.2s Front panel vs UTN

Yet interestingly, the performance between a USB device plugged into the front panel and one connected via the UTN server is not only comparable but proven to be better when using the UTN server. The plots show that while the front panel connection reaches a performance plateau at slightly above 40 MiB/s, the UTN server connection achieves a plateau above 50 MiB/s for Total IO. This suggests that the UTN server can provide enhanced performance compared to front panel connections, making it a viable alternative for scenarios where direct motherboard connections are not an option.

The front USB port on the ATX chassis was demonstrated to have a slower transfer rate and less stability compared to motherboard ports which might be due to:

- Cable Quality: The cables connecting the front panel USB ports to the motherboard headers can introduce signal degradation, especially if they are of lower quality.
- Power Delivery: Front panel USB ports might not provide as consistent power as the direct motherboard ports, which can affect performance, particularly for high-power devices.
- USB Version Compatibility: Ensure that both the motherboard and the front panel ports support the same USB version (e.g., USB 3.2 Gen 2) for optimal performance.
- Interference: The longer path and potential interference from other components can also impact the performance of front panel USB ports.

9.2.3 UTN caveats

Additionally, the analysis highlights a slight hiccup in the IOPS measurement of the UTN plot, particularly around the block sizes of 1 KiB to 8 KiB. This anomaly reflects network performance instabilities, which may be attributed to the inherent complexities of network-based USB connections. Despite this, the overall performance of the UTN server is still robust, offering a practical solution for managing USB devices over the network.

9.1 General Conclusions

In conclusion, the findings from this analysis emphasize the importance of selecting the appropriate connection method for USB devices based on performance requirements. Direct motherboard connections offer the best performance, while the UTN server provides a comparable and slightly better alternative to front panel connections. Future research should continue to explore and address the network performance instabilities observed in the UTN server to further optimize its application in various industries.

References

[ATTO-25] ATTO, "*ATTO Disk Benchmark Window",* February 2025. https://www.atto.com/download/atto-disk-benchmark-windows/.

[Carter-24] Carter, Michael, "*How to Benchmark Data Systems Using Storage Performance Metrics",* August 2024. https://sharearchiver.com/benchmark-storage-performance-metrics/.

[Dancuk-23] Dancuk, Milica, "*Measuring Storage and IOPS Performance (Windows, Linux, vCenter)",* November 2023. https://phoenixnap.com/kb/measuring-storage-and-iops-performance.

[Digi-25] Digi International Inc., *"USB Over IP White Paper",* Digi.com. 2025. https://www.digi.com/resources/library/white-papers/usb-over-ip.

[Digi-25] Digi International Inc., "*USB-Over-IP",* 2025. https://www.digi.com/products/networking/infrastructure-management/usb-connectivity/usb-over-ip/anywhereusb.

[Engelbert-24] Engelbert, Chris, "*IOPS vs Throughput vs Latency – Storage Performance Metrics",* April 2024. https://www.simplyblock.io/blog/iops-throughput-latency-explained/.

[Garcia-16] García Bedoy Torres, Jorge, "*Improving Harmonic Balance Performance Via Parallelization*", Instituto Tecnológico y de Estudios Superiores de Occidente. September 2016. https://rei.iteso.mx/server/api/core/bitstreams/14f8ffa1-409e-4e46-aa01-29b6ebaf18a5/content.

[iTechStorm-11] iTechStorm, "*Iometer Tutorial and Introduction",* 5 October 2011. https://www.itechstorm.com/iometer-tutorial-introduction.

[Microsoft-25] Microsoft, "*Use DISKSPD to test workload storage performance",* March 2025. https://learn.microsoft.com/en-us/azure/azure-local/manage/diskspd-overview?view=azloc-24112#quick-start-install-and-run-diskspd.

[SEH-25] SEH Computertechnik GmbH, "*UTN Manager",* 2025. https://www.seh-technology.com/services/dongle-server-maintenance/utn-manager.html.

[SILEX-25] SILEX TECHNOLOGY, "*Silex USB Device Server, 2x USB 2.0, 10/100/1000 LAN, US Power Supply",* 2025. https://www.pcnation.com/silex-technology-ds-510-us-zg1635?mkwid=s_dc&pcrid=524019285072&pkw=&pmt=&plc=&gad_source=1&gclid=EAIaIQobChMInsT8nrfIiAMV3g-tBh1eRgSpEAQYAiABEgIBU_D_BwE.

[T-22] T S, Ganesh, "*Kingston DataTraveler Max UFD Series Review: New Type-A Thumb Drive Retains NVMe Performance",* July 2022. https://www.anandtech.com/show/17486/kingston-dtmax-ufd-series-review.

[USB-25] USB over Network, "*How to Create a USB Server?",* April 2024. https://www.net-usb.com/usb-server/.

10. Appendix

10.1 post-processing of results for DiskSPD

To install and run DISKSPD, open PowerShell as an admin on your management PC, and then follow these steps (Microsoft 2025):

1. Download and expand ZIP file.
2. Add DISKSPD to $PATH environment variable.
3. Then run using the parameters provided:
    a. Diskspd [INSERT_SET_OF_PARAMETERS] [INSERT_CSV_PATH_FOR_TEST_FILE] > [INSERT_OUTPUT_FILE.txt]
    b. Example:
    diskspd.exe -t1 -o4 -b64M -r4k -w50 -d60 -Sh -D -L -c256M D: > test01.txt

Where a sample of the results is:

```
Command Line: C:\DISKSPD\AMD64\diskspd.exe -t1 -o4 -b64M -r4k -w50 -d60 -Sh -D -L -c256M D:

Input parameters:
	timespan:   1
	-------------
	duration: 60s
	warm up time: 5s
	cool down time: 0s
	measuring latency
	gathering IOPS at intervals of 1000ms
	random seed: 0
	path: 'D:'
		think time: 0ms
		burst size: 0
		software cache disabled
		hardware write cache disabled, writethrough on
		performing mix test (read/write ratio: 50/50)
		block size: 64MiB
		using random I/O (alignment: 4KiB)
		number of outstanding I/O operations per thread: 4
		threads per file: 1
		using I/O Completion Ports
		IO priority: normal

System information:
	computer name: DESKTOP-IS6T870
	start time: 2025/03/14 23:30:04 UTC

	cpu count:		6
	core count:		6
	group count:		1
	node count:		1
	socket count:		1
	heterogeneous cores:	n

	active power scheme:	Balanced (381b4222-f694-41f0-9685-ff5bb260df2e)
```

```
Results for timespan 1:
*******************************************************************************

actual test time:   60.00s
thread count:                   1

CPU |  Usage |  User  | Kernel |  Idle
-------------------------------------------
   0|   7.60%|   2.94%|   4.66%|  92.40%
   1|  10.21%|   8.15%|   2.06%|  89.79%
   2|   4.97%|   3.85%|   1.12%|  95.03%
   3|   3.91%|   2.71%|   1.20%|  96.09%
   4|   3.18%|   2.21%|   0.96%|  96.82%
   5|   2.99%|   1.90%|   1.09%|  97.01%
-------------------------------------------
avg.|   5.48%|   3.63%|   1.85%|  94.52%

Total IO
thread |       bytes     |     I/Os     |    MiB/s   |  I/O per s |  AvgLat  | IopsStdDev | LatStdDev |  file
------------------------------------------------------------------------------------------------------------------
     0 |     27850178560 |          815 |     442.65 |      13.58 |  293.977 |       3.95 |   288.590 | D: (476.94GiB)
------------------------------------------------------------------------------------------------------------------
total:       27850178560 |          815 |     442.65 |      13.58 |  293.977 |       3.95 |   288.590

Read IO
thread |       bytes     |     I/Os     |    MiB/s   |  I/O per s |  AvgLat  | IopsStdDev | LatStdDev |  file
------------------------------------------------------------------------------------------------------------------
     0 |     27850178560 |          415 |     442.65 |       6.92 |  577.301 |       0.28 |     1.632 | D: (476.94GiB)
------------------------------------------------------------------------------------------------------------------
total:       27850178560 |          415 |     442.65 |       6.92 |  577.301 |       0.28 |     1.632

Write IO
thread |       bytes     |     I/Os     |    MiB/s   |  I/O per s |  AvgLat  | IopsStdDev | LatStdDev |  file
------------------------------------------------------------------------------------------------------------------
     0 |               0 |          400 |       0.00 |       6.67 |    0.028 |       3.97 |     0.018 | D: (476.94GiB)
------------------------------------------------------------------------------------------------------------------
total:                 0 |          400 |       0.00 |       6.67 |    0.028 |       3.97 |     0.018

Total latency distribution:
  %-ile |  Read (ms) | Write (ms) | Total (ms)
----------------------------------------------
    min |    565.081 |      0.005 |      0.005
   25th |    576.768 |      0.015 |      0.023
   50th |    577.531 |      0.023 |    573.258
   75th |    577.835 |      0.033 |    577.542
   90th |    579.029 |      0.055 |    577.949
   95th |    579.739 |      0.065 |    579.078
   99th |    580.856 |      0.073 |    580.513
3-nines |    586.471 |      0.159 |    586.471
4-nines |    586.471 |      0.159 |    586.471
5-nines |    586.471 |      0.159 |    586.471
6-nines |    586.471 |      0.159 |    586.471
7-nines |    586.471 |      0.159 |    586.471
8-nines |    586.471 |      0.159 |    586.471
9-nines |    586.471 |      0.159 |    586.471
    max |    586.471 |      0.159 |    586.471
```

To consolidate the results, files were organized by IO size and then consolidated using the following PowerShell script to do the parsing into a consumable CSV format:

```
# Check if the input file path is provided as a command-line argument
if ($args.Length -lt 1) {
  Write-Host "Usage: .\parse.ps1 <inputFilePath>"
  exit
}

# Define the input and output file paths
$inputFilePath = $args[0]
$outputFilePath = "C:\Users\mafev\Documents\diskspd_tests\raw_results_MB.csv"
# Initialize variables to store the extracted data
$commandLine = $null
$blockSize = $null
$duration = $null
$totalIoMibS = $null
$totalIoIops = $null
$readIoMibS = $null
$readIoIops = $null
$writeIoMibS = $null
$writeIoIops = $null
# Read the input file line by line
$lines = Get-Content $inputFilePath
for ($i = 0; $i -lt $lines.Length; $i++) {
  $line = $lines[$i]
  if ($line -match "Command Line:") {
    $commandLine = $line.Split(":")[2].Trim()
    Write-Host "Command Line: $commandLine"
  }
  elseif ($line -match "block size:") {
    $blockSize = $line.Split(":")[1].Trim()
    Write-Host "Block Size: $blockSize"
  }
  elseif ($line -match "duration:") {
    $duration = $line.Split(":")[1].Trim()
    Write-Host "Duration: $duration"
  }
  elseif ($line -match "Total IO") {
    if ($i + 5 -lt $lines.Length) {
      $totalLine = $lines[$i + 5]
      $parts = $totalLine -split "\|"
      $totalIoMibS = $parts[2].Trim()
      Write-Host "Total IO MiB/s: $totalIoMibS"
      $totalIoIops = $parts[3].Trim()
      Write-Host "Total IO IOPS: $totalIoIops"
    }
  }
  elseif ($line -match "Read IO") {
    if ($i + 5 -lt $lines.Length) {
      $totalLine = $lines[$i + 5]
      $parts = $totalLine -split "\|"
      $readIoMibS = $parts[2].Trim()
      Write-Host "Read IO MiB/s: $readIoMibS"
      $readIoIops = $parts[3].Trim()
      Write-Host "Read IO IOPS: $readIoIops"
```

```
        }
      }
      elseif ($line -match "Write IO") {
        if ($i + 5 -lt $lines.Length) {
          $totalLine = $lines[$i + 5]
          $parts = $totalLine -split "\|"
          $writeIoMibS = $parts[2].Trim()
          Write-Host "Write IO MiB/s: $writeIoMibS"
          $writeIoIops = $parts[3].Trim()
          Write-Host "Write IO IOPS: $writeIoIops"
        }
      }
    }
    # Create a custom object with the extracted data
    $data = [PSCustomObject]@{
              "Command Line" = $commandLine
      "Block Size" = $blockSize
      "Duration" = $duration
      "Total IO MiB/s" = $totalIoMibS
      "Total IO IOPS" = $totalIoIops
      "Read IO MiB/s" = $readIoMibS
      "Read IO IOPS" = $readIoIops
      "Write IO MiB/s" = $writeIoMibS
      "Write IO IOPS" = $writeIoIops
    }
    # Check if the CSV file exists
    if (Test-Path $outputFilePath) {
      # Append the data to the existing CSV file
      $data | Export-Csv -Path $outputFilePath -NoTypeInformation -Append
    } else {
      # Create a new CSV file and write the data
      $data | Export-Csv -Path $outputFilePath -NoTypeInformation
    }
    Write-Host "The test results have been consolidated into '$outputFilePath'."
```

To further simplify another tiny script was written to iterate running the parser on all the files contained in a folder:

```
# Check if the input file path is provided as a command-line argument
if ($args.Length -lt 1) {
  Write-Host "Usage: .\wrapper.ps1 <Folder_Name>"
  exit
}

# Define the input and output file paths
$Folder_Name = $args[0]

# Iterate on every item in folder
# $folderPath = "C:\$Folder_Name"
$items = Get-ChildItem -Path $Folder_Name

foreach ($item in $items) {
  ./parse.ps1 "$Folder_Name\$item"
}
```

10.2 Total Cost of Ownership (TCOS) Analysis of Scaled-out Solution

Based on the following comparison depicted earlier in this document, a cost analysis could be drawn for a scalability scenario within a rack:

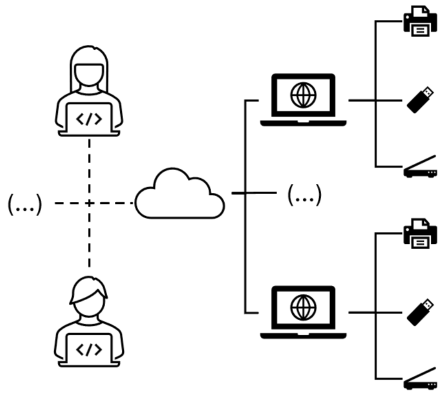


Figure 18. Typical layout with hosts in physical proximity of USB device and requiring an added PC to access remotely.

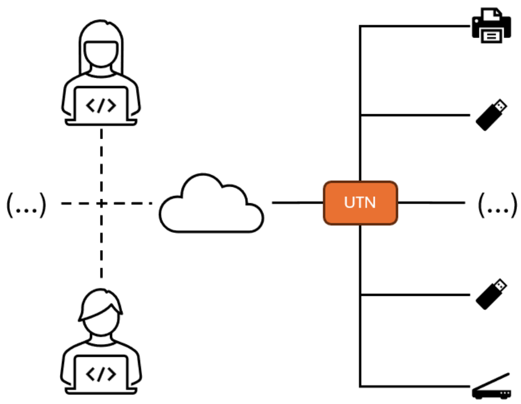


Figure 19. Proposed layout with UTN (USB Through Network) Server that allows for remote access to multiple USB devices.

For simplicity, we can take as an example 1 UTN replacing a single mini-PC to give a cost example. The UTN used in this case has an average retail price of $257 vs a mini-PC that could average on the low end about $500 depending on size, requirements, and availability, where each would have 2-4 (both PC and UTN) available USB ports depending on model. So roughly slicing cost in half for the same solution (along with security and manageability). Other models also offer 4 USB ports in UTN for around 100$ (such as Coolgear 4 Port USB Device Server Over Ethernet), in some cases which would make it significantly cheaper if other features can be disregarded for security and manageability.

Another example could be using a larger UTN Manager for 24 ports with a cost of around $2,400 (such as Digi AnywhereUSB 24 Plus USB/Ethernet Combo Hub - 24 USB 3.1 Port(s) - PC AW24-G300) could replace up to 6 mini PCs (assuming 4 USB devices per hub) which would go $500 a piece for a total of $3,000 but without accounting power consumption, HW and SW management, among many other points of failure that would be added to the solution. In both cases the UTN provides a more compelling TCOS as well as manageability for solutions.