IMPLEMENTATION OF A SECURED SCALABLE FILE SERVER SYSTEM

ABSTRACT:

This research paper proposed and developed a blueprint for a secured scalable file server system for organisations that can handle large volumes of data, accommodate numerous users with varying levels of access and permissions and adhere to all security and compliance standards of organisations. With the implementation of this secured and scalable file server system, users of all levels will be able to access, manipulate, transfer files easily devoid of any data lost, file corruption or fear of getting infested with virus when sharing using flash drives. System administrators will be able to backup files easily to forestall any unforeseen circumstances and finally the organisation's Management will be able to have full control of all data that is used in the organisation. The proposed system is a scalable file server designed for digital file management. This secured and scalable file server system provided a secure and user-friendly interface allowing organisation's staff to efficiently store, retrieve, and share files. The proposed system at its heart, provide strong data storage capabilities that will easily meet data growing needs.

KEYWORDS: DHCP, Networking, CISCO, Router, Switch, Packet tracer, Topology

INTRODUCTION

Various organisations and individuals regularly need to move their documents from one workstation to another in the same Unit or different Units. However, these transfer need to be executed in a more secure manner, nevertheless remaining simple to operate. The above can be achieved with the use of a file server. A file server allows multiple users to access the same files simultaneously while using centralized authentication to grant access to users. It also makes it very simple to create backups. Files can be backed up to an onsite or offsite platform so that they can be readily restored in the event of file damage. The dangers of file corruption, file viruses, and stolen storage media (which can hold important data) are also eliminated with File Server system. These difficulties usually come from the transfer of files between computers with storage media such as external hard drives and removable media.

In this digital era, the electronic file transfers which is the focus of this research paper, would be managed by using a server, which is both computer software and a hardware. Although most organisations are currently sharing some data through a web server using various approaches inclusing websites, anyone who visits the website can view these data. Sensitive information cannot be exchanged on the web platform as a result. This research paper aims to create a file server blueprint that will impose controlled access on users. The blueprint will also be implemented with scalability capabilities. As the organisations grows, more files are created, stored, and distributed across its numerous departments and units. File transfers between various units are currently done largely through removable drives. Additionally, sensitive files are shared using email services like Gmail, the corporate mail and the social media among others. These files should only be available at the same time to only the intended recipients, which is not the case

when some of the options listed above are used. When new offices or branches are created in the near future, provisions should be made so that they can quickly connect to the file system.

FILE SERVER

According to Sommerville (2016), a file server is a fundamental component of a networked environment that serves as a central repository for storing, managing and sharing digital files. It facilitates efficient data management and sharing within various units in an organization. Some benefits derived from the use of file servers according to Sommerville (2016) are:

a) Dedicated storage infrastructure that allows users to store various types of digital files such as documents, images, videos etc.
b) File permissions management that grants the right access to users to control access to files.
c) Allows multiple users within a network to access and work on the same files simultaneously.
d) Encryption of data in transit.
e) Backup mechanisms and
f) Fault tolerance features.

DISTRIBUTED HASH TABLE (DHT) APPROACH

Muthitacharoen et al (2002) describe Ivy, a read-write peer-to-peer file system that provides scalability, fault tolerance and effective storage. Ivy offers an original solution to the problems of scalability and fault tolerance in conventional file systems by utilizing a distributed hash table infrastructure and a decentralized file system interface. Due to centralization, traditional file systems are prone to performance bottlenecks and single points of failure. Ivy overcomes these restrictions by utilizing a peer-to-peer network's combined resources to enable effective data storage and retrieval. Ivy's distributed hash table (DHT) infrastructure enables quick file lookup and storage as its key innovation. The authors in Muthitacharoen et al (2002) described how the DHT converts file names to a distinctive identifier, enabling the location and access of files over a network. Constructed on top of the Ivy architecture is IvyFS, the File System interface. The authors described the many operations enabled by IvyFS such as file access, peer sharing and file creation, modification and deletion. Additionally, Muthitacharoen et al (2002) detailed how to resolve disputes and guarantee data consistency in a decentralized setting.

SCALABLE NETWROK FILE SYSTEM

From the point of view of Zhang (2019) in their work titled “Network file systems that are scalable”, traditional file systems are constrained in terms of performance, capacity and fault tolerance as data quantities continue to increase exponentially. The author in Zhang (2019) suggested novel strategies that took advantage of distributed architectures' strengths and advanced scalability techniques to get beyond these constraints. Data distribution, metadata management and data access protocols are few of the important parts of their proposed system. They stressed on the significance of load balancing, data replication and caching techniques to improve system performance and provide fault tolerance. Additionally, Zhang (2019) suggested a clever data placement technique that optimizes data distribution across dispersed nodes to reduce access latency and increase throughput. In brief, Zhang (2019) addressed the difficulties of managing sizable, distributed storage systems by presenting a thorough analysis on scalable network file

systems to improve performance, scalability and fault tolerance. The suggested design incorporates distributed file systems, parallel computing, and fault tolerance methods.

SCALABLE FILE SERVER ARCHITECTURE

A research paper by Brown (2020) titled "Scalable File Server Architecture for Large University Campuses" published in the International Journal of Advanced Networking and Applications discussed the development of a scalable file server architecture. The project aimed at addressing the difficulties major University campuses encountered while managing enormous volumes of data. The paper explored the technical details of the file server design, focusing on components that guarantee scalability, performance, and fault tolerance. The suggested architecture employed a distributed file server strategy to effectively divide the data load and offer redundant storage for increased data availability and system resilience. Simulated and practical tests were used in the research to assess the performance and scalability of the suggested design. The findings showed that the scalable file server design has improved performance and is able to efficiently meet the expanding data requirements of huge university campuses.

DRAWBACKS OF REVIEWED SYSTEMS

Table 1 presented in this research paper summarizes the drawbacks identified in the Muthitacharoen et al (2002), Zhang (2019) and Brown et al (2020).

Table 1: Drawbacks of reviewed systems

CASE STUDY	AUTHOR(S)	DRAWBACKS
CASE 1	Muthitacharoen et al. (2002)	1. Limited Consistency Guaranteed: Because Ivy is decentralized, it only offers flimsy consistency assurances. In some situations, this can result in inconsistent data. 2. Lack of Full Fault Tolerance: The lack of full fault tolerance in Ivy's design means that some failure types may not be fully recovered from. 3. Complexity of Maintenance: Ivy is a peer-to-peer file system, as such implementation and maintenance calls for specialized knowledge and continuous monitoring.
CASE 2	Zhang (2019)	1. Complexity in Implementation: It is difficult to construct and maintain such systems because scalable network file systems may have complicated architectural designs and algorithmic implementations. 2. Increased Network Traffic: Due to data duplication and dissemination, distributed file systems may result in increased network traffic, which could affect overall network performance. 3. Data Integrity and Consistency: Data consistency and integrity over dispersed servers can be difficult to ensure and poor implementation can result in inconsistent data.
CASE 3	Brown et al. (2020)	1. Network Dependency: The scalable file server architecture is heavily reliant on the network

		infrastructure, making it vulnerable to problems and bottlenecks with the network. 2. Increasing Hardware Requirements: Additional hardware purchases is necessary to grow file server systems for large university campuses. 3. Maintenance and Management: Operating such a design need constant monitoring and maintenance for peak performance.

PROPOSED SECURED SCALABLE FILE SERVER

Burback (1998) indicated that methodology refers to the set of practices, processes, and tools that are used to develop and maintain software. These methodologies provide a structured approach for organizing and managing the software development life cycle, from the initial conception and requirements gathering, through design, implementation, testing, and deployment. The model adopted in this study is the waterfall model. Lutkevich (2022), waterfall model is characterized by a series of distinct phases, with each phase building upon the previous one which leads to the final delivery of the software product. Waseem (2022) revealed that there are three principles of the waterfall model which are the sequential structure, robust documentation, and minimal customer involvement which are some of the reason why this model is adopted. This research paper used Cisco Packet Tracer. As the paper is centered on the design of file server which is network related, Cisco Packet Tracer is used to design and simulate the network environment.

Cisco Packet Tracer is a tool used to simulate network configuration, including routers, switches, virtual servers etc. With it, the research work can test how the file server interacts with the network infrastructure and make sure it runs without a problem within an organisation’s current network configuration by emulating the network environment. Additionally, it offers a secure and controlled environment for testing and validating, minimizing the possibility of network disruption while the development phase is ongoing. Overall, the combination of the Waterfall methodology and Cisco Packet Tracer as a development tool brings structure and precision to the system development process, ensuring that the scalable file server meets standard organisation's requirements.

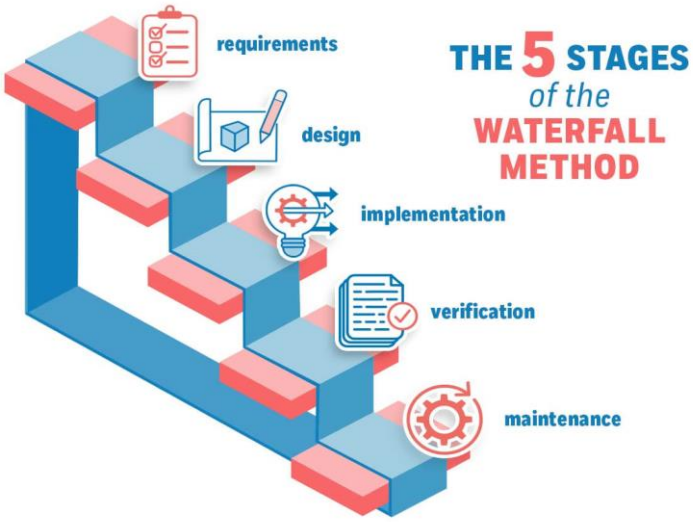

Figure 1: Waterfall model (Source: https://management.org/waterfall-methodology)

The proposed system is a scalable file server designed for digital file management. This secured and scalable file server system provided a secure and user-friendly interface allowing organisation's staff to efficiently store, retrieve, and share files. The proposed system at its heart, provide strong data storage capabilities that will easily meet data growing needs. A system for access control ensures that information is kept secured and is only available to users who have been granted permission to access. Additionally, strict data integrity controls, such as automated and secure backup procedures are put in place to defend against data loss due to hardware breakdowns or other unforeseen circumstances. The proposed file server system, offers scalability, performance, security, and user-friendliness, basically forms the basis of organisation's data management infrastructure. The system is ready to support a more productive and friendly academic and administrative environment while meeting the organisation's present and future needs.

PROPOSED SYSTEM'S USE CASE AND DATA FLOW DIAGRAMS

The use case diagram shown in figure 2 depicts how the various actors of the file server will interact with the file server. The actors are:

- users which is made up administrative staff whilst the
- system administrators which are the technical staff who manages the file server.

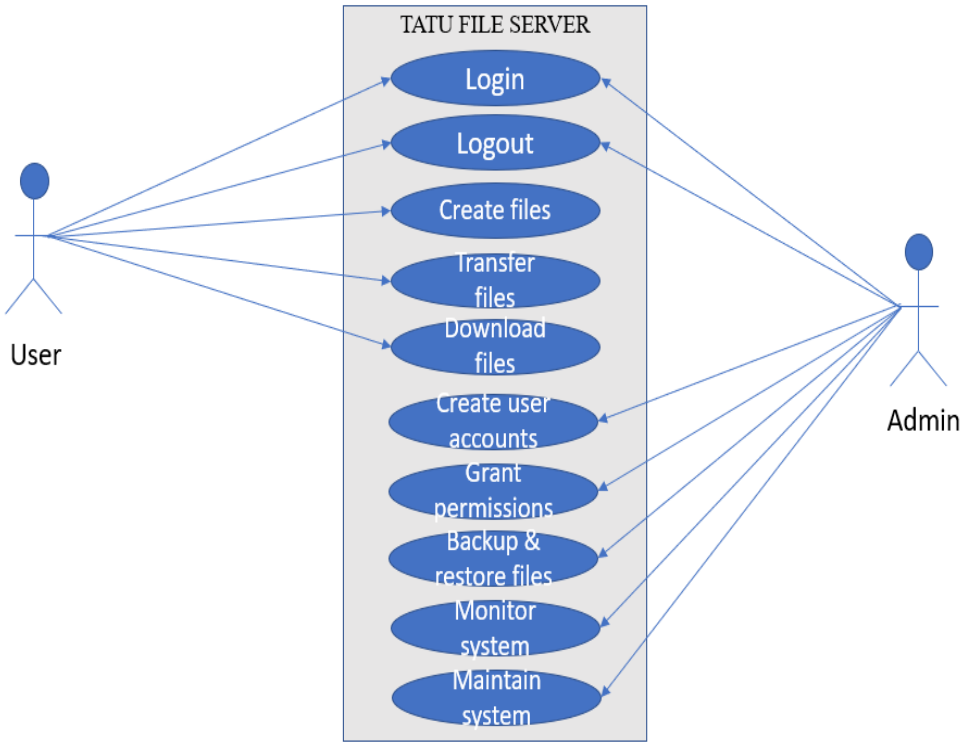

Figure 2: Use case diagram

A data flow diagram (DFD) highlights operations, data flows, data storage, and external entities to illustrate how data travels inside a system. A DFD can help provide a clear picture of how data is processed and stored within the system in terms of the Scalable File Server Implementation project.

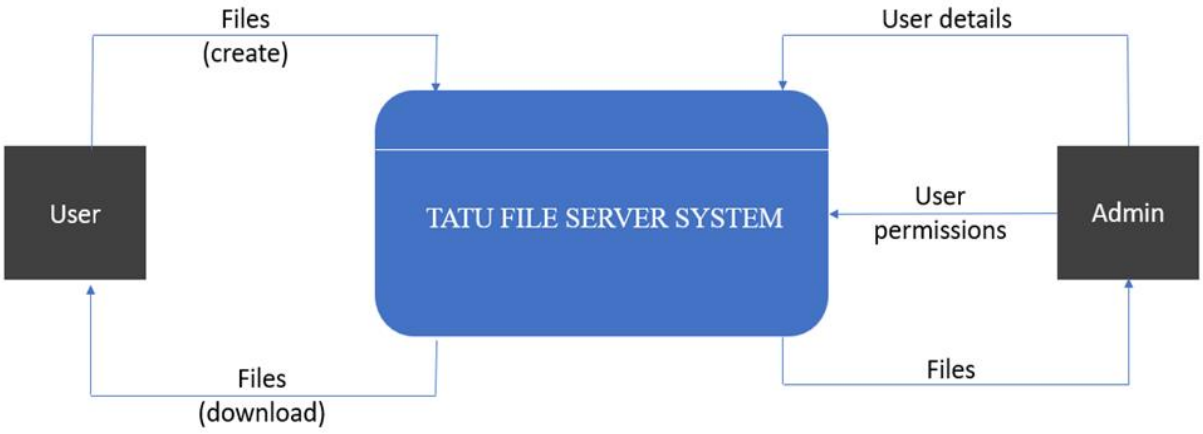

Figure 3: Data Flow Diagram

The Scalable File Server System's internal data flow is depicted in detail using this data flow diagram. It provides a clear picture of the data flow and processes involved in the system's

functioning and explains how users, administrators, and the system's many processes and data stores interact with one another.

PROPOSED NETWORK TOPOLOGY

The figure 4 show how the entire proposed file server structure look like showing currently four subnets which are active. When the load increases, the remaining four subnets can easily be added to accommodate the increased load and this was implemented using the Tamale Technical University (TaTU) network topology.

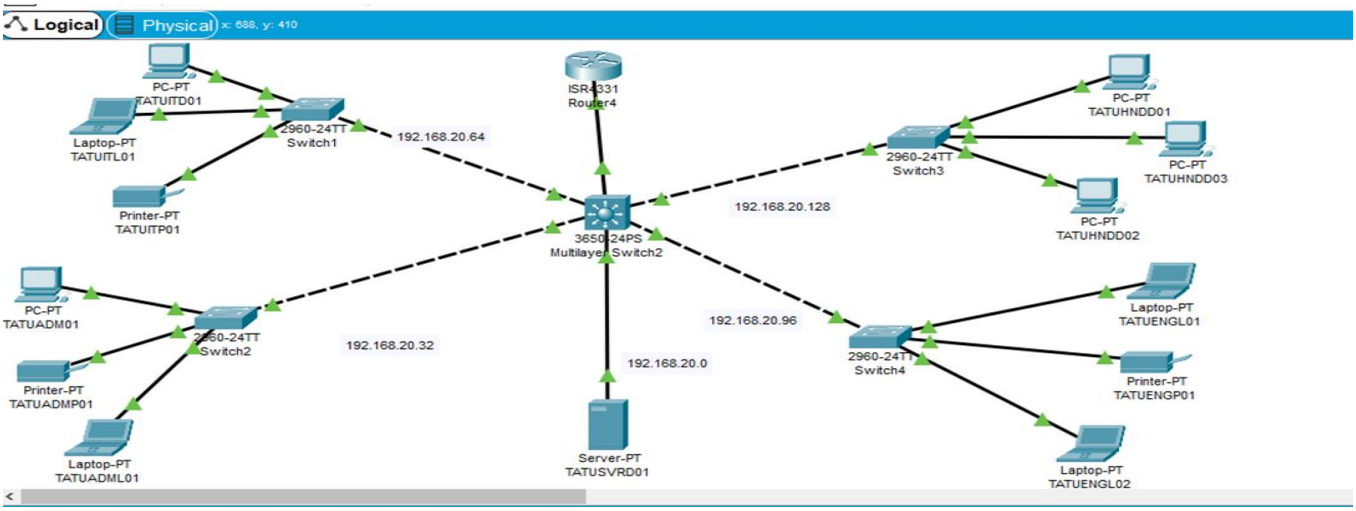

Figure 4: Proposed Network topology

Class C IP address is chosen and sub-netted into 8 that is

192.168.20.0 = 192.168.20.0/27

Subnet mask = 255.255.255.224

This makes room for 32 hosts in each network.

The table 2 shows more detail of the 8 subnets.

Table 2: Subnet Address Details

BLOCK/ LOCATION	NETWORK ADDRESS	GATEWAY ADDRESS	HOST RANGE	BROADCAST ADDRESS
SVR ROOM	192.168.20.0	192.168.20.1	192.168.20.2-30	192.168.20.31
ADMIN	192.168.20.32	192.168.20.33	192.168.20.34-62	192.168.20.63
IT	192.168.20.64	192.168.20.65	192.168.20.66-94	192.168.20.95

ENG	192.168.20.96	192.168.20.97	192.168.20.98-126	192.168.20.127
HND	192.168.20.128	192.168.20.129	192.168.20.130-158	192.168.20.159
x	192.168.20.160	192.168.20.161	192.168.20.62-190	192.168.20.191
x	192.168.20.192	192.168.20.193	192.168.20.194-222	192.168.20.223

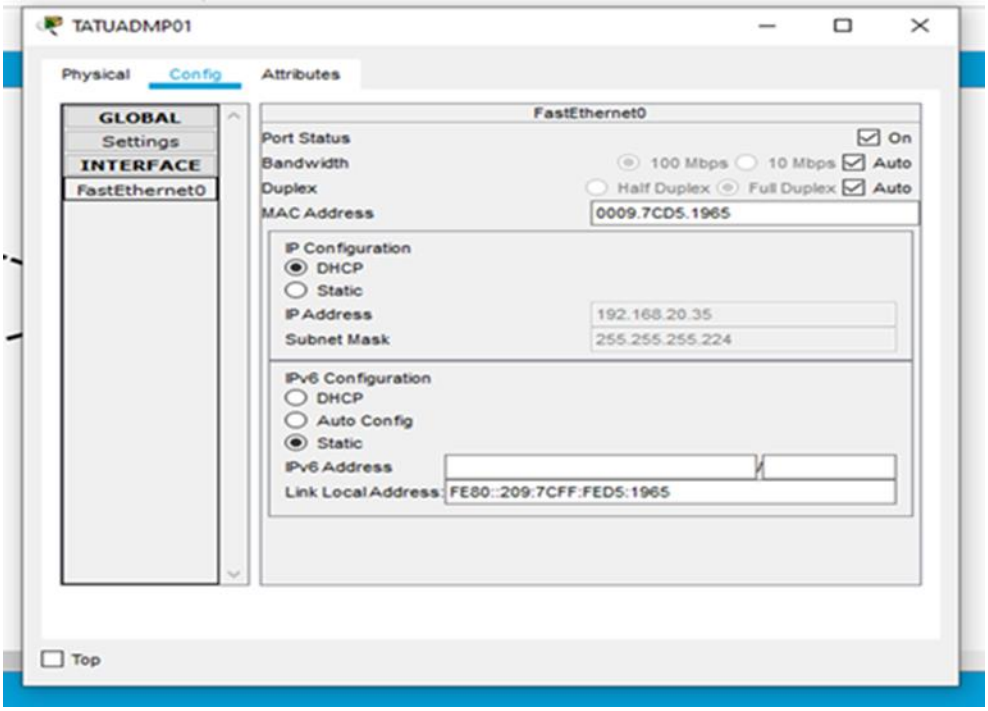

Figure 5: Successful Configured Host on Admin Host

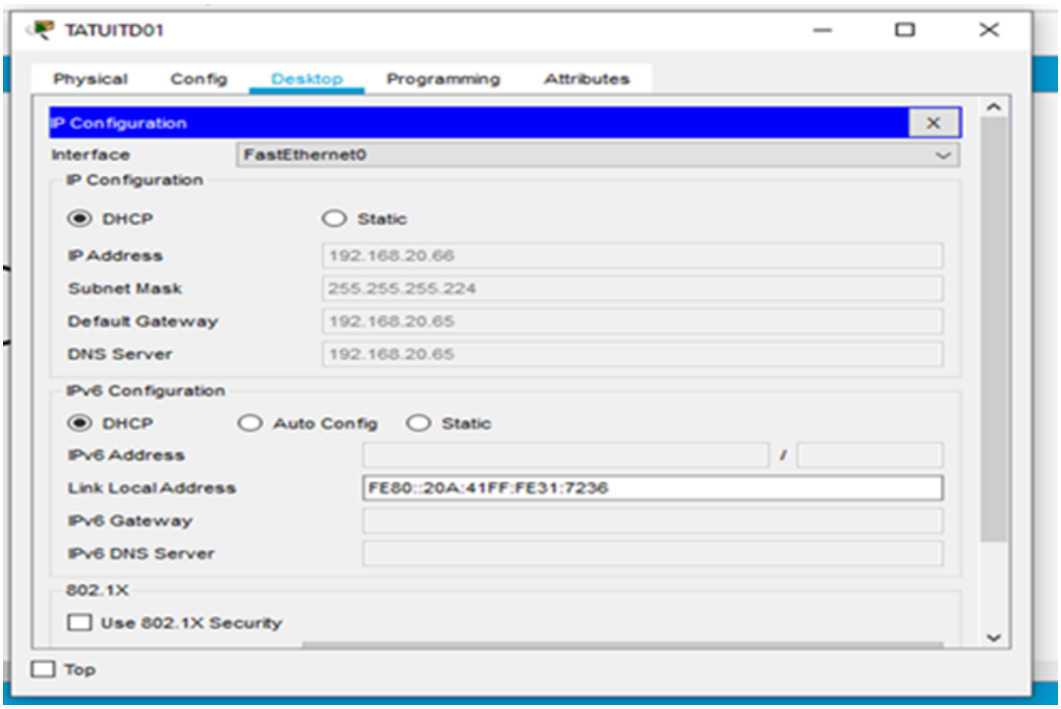

Figure 6: Successful Configured Client on Admin Host

PROPOSED SYSTEM TESTING

The research work conducted tests to be sure that any device connected on a subnet and set to pick an IP address automatically, picks an IP address that falls within that subnet. Also, a ping of all nodes on each subnet to make sure that data packages are delivered successfully was conducted. Figure 7 is a screenshot of a ping result from the first device to the other two devices on the Admin Subnet.

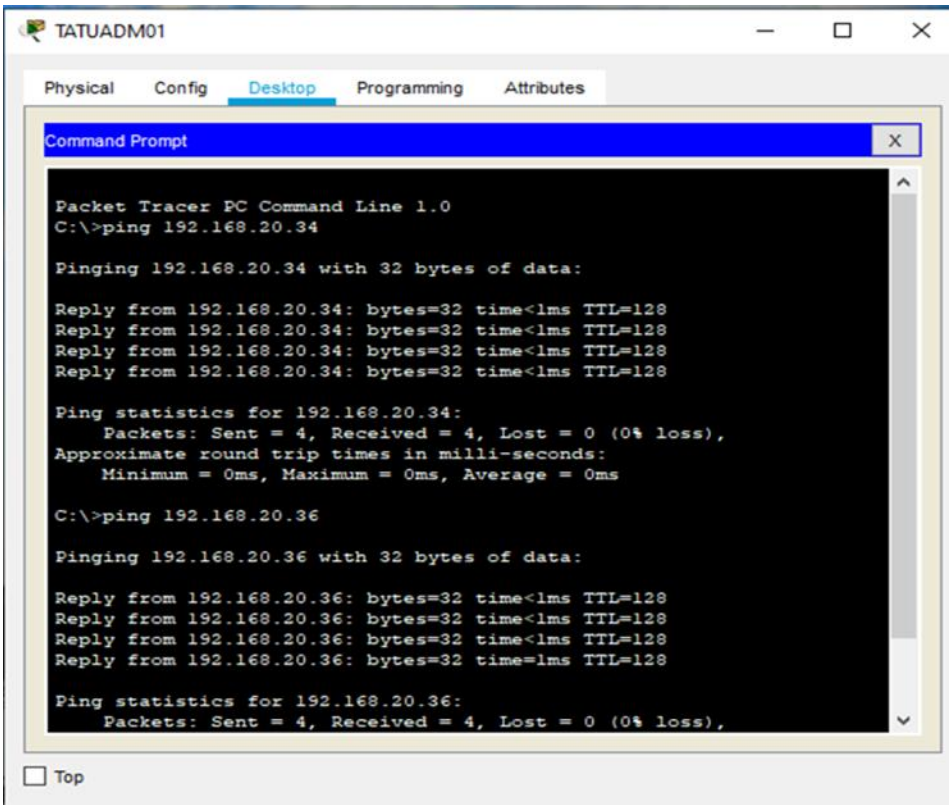

Figure 7: Ping Result on Admin Subnet during Unit Testing

The four configured subnets were tested simultaneously by sending packets to the server at the same time. All packets were delivered successfully. The figure 8 shows packages coming from the four subnets.

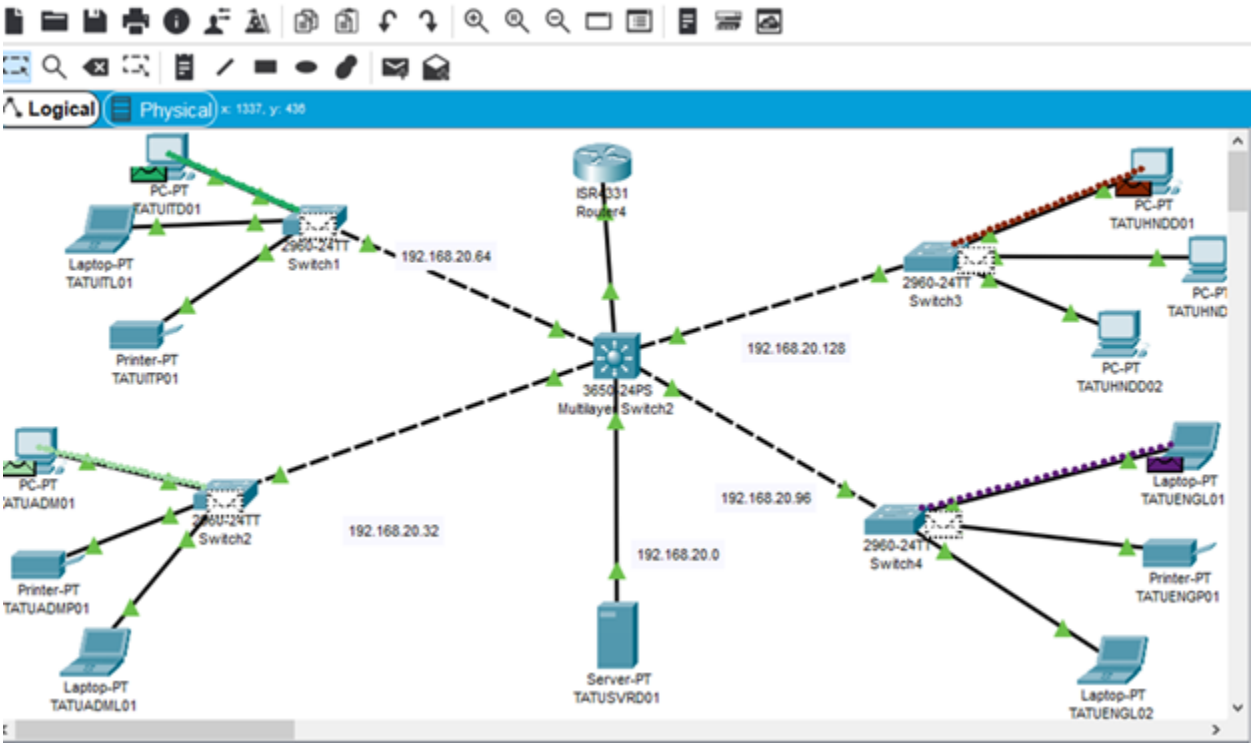

Figure 8: Packages sent from all 4 subnets

PROPOSED FILE SERVER SECURITY AND PERMISSION

The proposed file server system presented in this research work ensures security guarantees recommended by both the traditional and state-of-the-art file systems as stated by Kadekodi et al (2019). The researchers in Kadekodi et al (2019) identified three properties that needs to be satisfied. These properties are the need for the file server meta data to always be updated by a trusted source, the corruption of data are restricted to data processes that requires write permissions and only permitted or legal writers can update file's data. These properties identified by Kadekodi et al (2019) was also proposed by Kwon et al (2017) and the proposed file server system presented in this research work has satisfied all the three properties proposed by both Kadekodi et al (2019) and Kwon et al (2017). File permission checks in the implementation of the proposed file server system was also performed by relying on the Operating Systems (OS) for data operations. The research work ensured the trusted Operating System (OS) is able to generate a 128-bit random unique identifier for every process and this is used to update the credential table as one of the requirements. The security and performance of the proposed file server system is attributed to the innovative and faster hardware storage and effective access interface as argued by Bjørling, M., González, J., & Bonnet, P. (2017). Data integrity is one of the most essential features that requires special attention as stated by Lagoze (2014) and that has been considered and addressed during the implementation and testing of the proposed file server system.

CONCLUSION AND RECOMMENDATIONS

The secured and scalable file server presented in this research paper is a step in ensuring that the organisation's data, which are very essential to its financial and administrative operations, is organized, accessible, and well-protected. The suggestions made in this research paper are meant to act as a guide for the continuous maintenance, improvement, and appropriate use of the file server. Organisations in this digital era have reached important turning point where a successful implementation of a scalable file server is required. It is crucial in solving the organisation's expanding file management needs, ensuring that data remains an asset rather than a liability as scalability, data integrity and security, improved performance, and easy file backup are few advantages of this new proposed system. The backbone of an organisation's administrative operations is a well-organized file server, which makes it easier to share documents among Units and encourage collaboration. It boosts performance and creativity by enabling users to work more efficiently and ensures that data is accessible when needed.

However, it is important to remember that the proposed system is not completed after implementation. Regular updates, on-going maintenance, and user support are needed if the scalable file server is to keep up with its promises. Monitoring the server's health, installing important updates and patches, and taking preventative measures to resolve any problem are all part of the maintenance tasks. Additionally, the file server must be secured against intruders because cybercriminals are now well-organized and know what to do with stolen data.

Finally, educating end-users about best practices on data, security, and system utilization is important. It ensures that they can fully harness the capabilities of the file server while also safeguarding the integrity of the data it hosts. These post implementation steps are very important to keep the file server robust and a reliable asset.

This research paper recommends the implementation of File replication to allow user files to be stored on individual workstations and then synchronized with server. This will allow users to have access to their files even if the server is down and when it comes up, the files are synchronized with the server. The file screening will prevent users from saving some types of file on the server/desktop which will help to manage disk space.

REFERENCES

Brown. (2020). Scalable File Server Architecture for Large University Campuses.

Burback, R. (1998, December). *Software Engineering Methodology*. Retrieved from http://infolab.stanford.edu/~burback/watersluice/watersluice.pdf

Lutkevich, B. (2022, November). *Waterfall Model.* Retrieved from TechTarget: https://www.techtarget.com/searchsoftwarequality/definition/waterfall-model

Muthitacharoen et al, .. (2002). Ivy: A Read/Write Peer-to-Peer File System. *The USENIX Association.* Boston, Massachusetts, USA.

Sommerville, I. (2016). *Software Engineering (10th ed.).* . Pearson Education Limited.

Tanenbaum & Van Steen. (2007). *Distributed Systems: Principles and Paradigns.* Upper Saddle River, New Jersey: Pearson Prentice Hall .

Waseem, A. (2022, November 23). *Waterfall Methodology*. Retrieved from https://management.org/waterfall-methodology

Zhang, Y. W. (2019). Scalable Network File Systems. *17th International Conference on Distributed Computing Systems, ,* (pp. 123-134.).

Ren Y, Min C, Kannan S. (2020). CrossFS: A Cross-layered Direct-Access File System. In14th USENIX Symposium on Operating Systems Design and Implementation, OSDI 20, pp. 137-154

Kadekodi, R., Lee, S. K., Kashyap, S., Kim, T., Kolli, A., & Chidambaram, V. (2019). SplitFS: Reducing Software Overhead in File Systems for Persistent Memory. In A. EditorLastName (Ed.), Proceedings of the 27th ACM Symposium on Operating Systems Principles, SOSP ’19 (pp. 494–508). New York, NY, USA: Association for Computing Machinery.

Kwon, Y., Fingler, H., Hunt, T., Peter, S., Witchel, E., & Anderson, T. (2017). Strata: A Cross Media File System. In A. EditorLastName (Ed.), Proceedings of the 26th Symposium on Operating Systems Principles, SOSP ’17 (pp. page range).

Bjørling, M., González, J., & Bonnet, P. (2017). LightNVM: The Linux Open-channel SSD Subsystem. In A. EditorLastName (Ed.), Proceedings of the 15th Usenix Conference on File and Storage Technologies, FAST’17 (pp. page range). Santa Clara, CA, USA.

Lagoze, C. (2014). Big Data, data integrity, and the fracturing of the control zone. Big Data & Society, 1(2).