



 **Names :** NTARINDWA Jackson


 **ID:** 25593

Assignment#0 REPORT

● Originality Declaration:

I **NTARINDWA Jackson** declare that this work is my own original work and has not been plagiarized from any source.

Contact Info

 +250 784 704 515

 jackson@ntarindwa.dev

 www.ntarindwa.dev



Table of Contents

Table of Contents	1
Task A : DoD (TCP/IP) vs OSI Model	3
4.a) DoD (TCP/IP) Model : Four Layers	3
4.b) Compare DoD vs OSI Structural & Functional Differences	4
4.c) Inclusions per Layer & 4.d) OSI Model in Depth	5
Task B : TCP vs UDP	6
5.a) Transmission Control Protocol (TCP)	6
• Real-World Example 2 (File Transfer - SFTP): When I push a software update to a server using SFTP (which runs over SSH), I rely on TCP. The file must be transferred bit-for-bit perfectly. A single lost packet could corrupt the entire archive, making it unusable. TCP's retransmission mechanism prevents this.	7
5.b) User Datagram Protocol (UDP)	7
5.c) Compare & Contrast	7
Task C : Reflection on 17 Modules	8
Module 1: Communication in a Connected World	8
Module 2: Network Components, Types, and Connections	9
Module 3: IP Addressing and Subnetting	9
• Reflection/Improvement: More practice labs on subnetting with real scenarios (like a corporate network expansion) would make the concept more intuitive.	10
Module 4: Routing Basics	10
Module 5: Switching Concepts	10
Module 6: Wireless Networking	10
Module 7: Network Security	11
Module 8: Network Troubleshooting	11
• Reflection/Improvement: Including timed troubleshooting exercises would simulate real operational pressures.	11
Module 9: Virtualization Basics	11
Module 10: EVE-NG Deployment	12
Module 11: Network Services	12
Module 12: WAN Concepts	12
Module 13: Cloud Networking	13
Module 14: Network Automation	13
Module 15: Advanced Routing	13
Module 16: Network Monitoring	14
• Reflection/Improvement: Using live network monitoring tools would provide real-world exposure.	14
Module 17: Final Project & Integration	14
Task D :Network Topologies	14
Star Topology	15
Mesh Topology	16

Task E : Installation Evidence	17
Screenshot 1: Packet Tracer	17
Screenshot 2: VirtualBox Manager	19
Screenshot 3: EVE-NG Web Interface & Topology	20
Screenshot 4: Virtual Machines	21
Screenshot 5: Canvas Profile	22
Screenshot 6: Course Completion	23
Reflection:	
Having prior experience, this course was an excellent structured refresher. The module on IPv6 addressing formats was particularly valuable for solidifying the different address types and compression rules. The Packet Tracer labs, while basic, were well-designed for validating core concepts like DHCP and ARP in a controlled environment. It provided a strong, standardized foundation that complements hands-on experience.	24
Task G Agreement & Commitment Plan	24
Appendices	25
References:	25

Task A : DoD (TCP/IP) vs OSI Model

4.a) DoD (TCP/IP) Model : Four Layers

From my experience working with networks, the DoD model is what you actually see in practice. It's less about strict layers and more about the functional groups that make the internet work.

- **Application Layer (DoD):** This is the layer I interact with daily. It's all about the applications themselves. Its job is to provide the protocols and services that user software needs, like a web browser or email client.

Protocols/Services: HTTP(S), DNS, SSH, FTP, SMTP, DHCP. For example, when I type a URL into Chrome, it's the HTTP protocol at this layer that formats the request for the web server.

Real-World Operation: Think of checking your email. Your mail client (e.g., Outlook) uses SMTP to send and IMAP/POP3 to receive messages. All of that happens here.

- **Transport Layer (DoD):** This layer is the workhorse for managing communication between hosts. I've spent a lot of time troubleshooting here. It's responsible for breaking down data from the application layer into segments, and it's where the critical choice between TCP and UDP is made.

Protocols/Services: TCP and UDP. TCP's reliability features like sequence numbers, acknowledgments (ACKs), and retransmission of lost packets are crucial for anything where data integrity is non-negotiable.

Real-World Operation: When I transfer a large file to a server, TCP ensures every packet arrives correctly. If packet #5 gets lost, the receiving host will not acknowledge it, prompting the sender to retransmit it, ensuring the file isn't corrupted.

- **Internet Layer (DoD):** This layer is all about logical addressing and routing getting packets from a source network to a destination network, anywhere in the world. It's the post office of the model.

Protocols/Services: IP (IPv4 and IPv6), ICMP (for ping and traceroute).

Real-World Operation: When I ping 8.8.8.8 from my home network, my PC creates an ICMP packet. The Internet layer on my PC wraps it in an IP packet with my PC's IP as the source and 8.8.8.8 as the destination. My router then uses this IP information to figure out where to send it next.

- **Network Access / Link Layer (DoD):** This is the physical network I can touch. It deals with the hardware Ethernet cables, Wi-Fi radios, network switches, and MAC addresses. Its job is to take IP packets from the Internet layer and frame them for the local physical medium.

Protocols/Services: Ethernet, Wi-Fi (802.11), ARP (which is essential for mapping IPs to MACs).

Real-World Operation: Before my PC can send that ping to the router, it needs the router's MAC address. It uses an ARP broadcast: "Who has IP address 192.168.1.1?" The router replies with its MAC address. My PC then puts the IP packet into an Ethernet frame addressed to the router's MAC and sends it out over the cable.

4.b) Compare DoD vs OSI Structural & Functional Differences

The OSI model is a fantastic theoretical tool for learning, but the DoD model is what's actually implemented. The main difference is granularity. OSI breaks the process into seven more precise layers, while TCP/IP collapses the top three into one and treats the bottom two as one.

OSI Model (7 Layers)	DoD / TCP/IP Model (4 Layers)	Key Functionality & My Observation
Application (7)	Application	HTTP, DNS, FTP. In practice, the lines between Presentation and Session are blurry. TLS encryption (a Presentation layer concept) is just part of HTTPS now.
Presentation (6)		Encryption, Compression.

Session (5)		Session Management.
Transport (4)	Transport	TCP, UDP. This mapping is direct and clear in both models.
Network (3)	Internet	IP, ICMP. The core function of routing is identical here.
Data Link (2)	Network Access	Ethernet, 802.11, MAC Addresses. I've observed that troubleshooting often starts here—is the link light on? Is the MAC being learned?
Physical (1)		Cables, Radio Waves, Bits.

The OSI model is perfect for teaching and for methodical troubleshooting (e.g., "Let's rule out Layer 1 first: is the cable plugged in?"). The DoD model is a more practical view of the real-world protocol suite that runs the internet.

4.c) Inclusions per Layer & 4.d) OSI Model in Depth

The interplay between layers is what makes networking work. For example, a DNS query (Application/Layer 7) is sent via UDP (Transport/Layer 4). The UDP segment is wrapped in an IP packet (Network/Layer 3) with a destination IP. That IP packet is then placed inside an Ethernet frame (Data Link/Layer 2) with a destination MAC address and finally converted into electrical signals on a cable (Physical/Layer 1). Each layer adds its own header, creating the encapsulation process I've seen in Wireshark captures countless times.

- **Physical (Layer 1):** This is the physical medium. In my home lab, this is the Cat6 cable or the Wi-Fi signal. A real-world example is the RJ-45 port on my laptop sending and receiving electrical signals.
- **Data Link (Layer 2):** This layer deals with frames and MAC addresses. My network switch operates here, intelligently forwarding frames only to the port where the destination MAC address is located, which I verified by watching MAC address tables on a switch in Packet Tracer.

- **Network (Layer 3):** The router is the classic Layer 3 device. It uses IP addresses to make routing decisions. When I change the IP address on a VM, I'm operating at this layer.
- **Transport (Layer 4):** This is where TCP and UDP live. When I configure a firewall rule to allow HTTPS traffic (TCP port 443), I'm working at the Transport layer.
- **Session (Layer 5):** This layer manages dialogues. The setup, maintenance, and teardown of a remote desktop (RDP) session is a good example of this layer in action.
- **Presentation (Layer 6):** This layer translates and encrypts data. When a web server sends data encrypted by TLS/SSL, that translation from plaintext to ciphertext happens here.
- **Application (Layer 7):** This is the end-user interface. HTTP, FTP, and SMTP are protocols here. When I use FileZilla to connect to an FTP server, the application I'm using and the protocol it's speaking are Layer 7.

Task B : TCP vs UDP

5.a) Transmission Control Protocol (TCP)

TCP is all about reliability. I use it for any application where I can't afford to lose data. The three-way handshake (SYN, SYN-ACK, ACK) is the formal introduction before any conversation starts, ensuring both ends are ready. What I find most impressive is its resilience through mechanisms like sequence numbers and ACKs for error recovery, and sliding windows for flow control—preventing a fast sender from overwhelming a slow receiver.

- **Real-World Example 1 (Web Browsing - HTTPS):** Loading this report from a website *must* use TCP. Every single HTML, CSS, and image file needs to arrive perfectly intact and in order. A corrupted image or missing code would break the page. TCP ensures this reliability.

- **Real-World Example 2 (File Transfer - SFTP):** When I push a software update to a server using SFTP (which runs over SSH), I rely on TCP. The file must be transferred bit-for-bit perfectly. A single lost packet could corrupt the entire archive, making it unusable. TCP's retransmission mechanism prevents this.

5.b) User Datagram Protocol (UDP)

UDP is the "fire and forget" protocol. It's connectionless, has minimal overhead, and offers no promises. This might sound bad, but it's perfect for applications where speed is more critical than perfect accuracy. There's no handshake, no retransmission, and no congestion control.

- **Real-World Example 1 (DNS Queries):** When your computer needs to resolve google.com to an IP address, it sends a tiny UDP packet to a DNS server. It needs an answer fast. If the request gets lost, the application can just send another one in a few milliseconds. The overhead of setting up a TCP connection for such a small transaction would be silly.
- **Real-World Example 2 (Video Streaming):** Services like YouTube or live streams use UDP (or protocols built on it). If a few packets are lost during a live stream, you might get a little pixelation for a second. If the service used TCP, it would halt the entire stream to retransmit the lost packets, causing a much more annoying buffer or pause. A minor visual glitch is preferable to a frozen screen.

5.c) Compare & Contrast

Feature	TCP	UDP
Connection	Connection-oriented (handshake)	Connectionless (no handshake)
Reliability	High. ACKs, retransmission, error recovery.	None. Best-effort delivery.

Ordering	Yes. Sequences packets.	No. No sequencing.
Overhead	High. Larger headers, control traffic.	Very Low. Small headers.
Speed	Slower. Due to reliability mechanisms.	Faster. Lower latency.
Use Cases	Web, email, file transfers, SSH.	VoIP, live video, DNS, gaming.

How to Choose:

The choice comes down to the application's needs. My rule of thumb is:

- **Use TCP** for any transaction where **data integrity is critical**. If you need every single byte to arrive correctly (emails, files, web pages), TCP is the only choice.
- **Use UDP** for **real-time applications where low latency is critical**. If the application can handle a small degree of loss (video, audio, game state updates) and needs to be fast, UDP is far more efficient.

Task C : Reflection on 17 Modules

Module 1: Communication in a Connected World

- **Core Concepts:** Introduced the basics of data communication, core components of networks (end devices, intermediaries, media), and different network types like LANs and WANs.
- **Experiment/Observation:** I set up a simple Packet Tracer lab with two PCs connected via a crossover cable. Using simulation mode, I observed the ICMP packets traveling back and forth during a ping, which made the abstract concept of communication channels very concrete.
- **Real-World Relevance:** This is the absolute foundation. Every network, from a smart home to a global cloud provider, is built upon these basic principles of senders,

receivers, and media.

- **Reflection/Improvement:** A quick demo using a tool like Wireshark on the classroom network could show the immense volume of background communication happening invisibly, enhancing understanding.

Module 2: Network Components, Types, and Connections

- **Core Concepts:** Detailed the function of routers, switches, and firewalls. Differentiated between end devices and intermediary devices.
- **Experiment/Observation:** In my home lab, I compared the behavior of an old hub to a modern switch. Using a packet sniffer, I confirmed that the hub broadcasts traffic to all ports, while the switch intelligently forwards it only to the destination port.
- **Real-World Relevance:** Choosing the right device is critical. Understanding hubs vs switches helps appreciate network efficiency and security.
- **Reflection/Improvement:** A physical lab with real, decommissioned gear to handle and cable would solidify the theoretical knowledge of each device's role.

Module 3: IP Addressing and Subnetting

- **Core Concepts:** Covered IPv4 addressing, subnet masks, and subnetting techniques to divide networks efficiently.
- **Experiment/Observation:** I created a subnetting plan for a small office network in Packet Tracer and verified connectivity between devices. Initial miscalculations caused ping failures, which I fixed by recalculating subnets carefully.
- **Real-World Relevance:** Proper IP addressing prevents conflicts and ensures network scalability.

- **Reflection/Improvement:** More practice labs on subnetting with real scenarios (like a corporate network expansion) would make the concept more intuitive.

Module 4: Routing Basics

- **Core Concepts:** Introduced static and dynamic routing, routing tables, and packet forwarding principles.
- **Experiment/Observation:** Configured static routes between two networks in Packet Tracer. Some routes failed initially due to missing entries, which I fixed by updating the routing tables.
- **Real-World Relevance:** Routing is essential for inter-network communication and Internet connectivity.
- **Reflection/Improvement:** Including a lab with misconfigured routers to troubleshoot would strengthen practical skills.

Module 5: Switching Concepts

- **Core Concepts:** Covered VLANs, trunking, and switch port configurations.
- **Experiment/Observation:** Implemented VLANs in Packet Tracer. Initially, VLANs couldn't communicate until I corrected port assignments and trunk settings.
- **Real-World Relevance:** VLANs improve network performance and security by segmenting traffic.
- **Reflection/Improvement:** Adding hands-on exercises with physical switches would reinforce the concept.

Module 6: Wireless Networking

- **Core Concepts:** Basics of Wi-Fi, SSIDs, wireless security, and access points.
- **Experiment/Observation:** Set up a wireless network in Packet Tracer. Connection failed until I adjusted SSID and encryption settings.

- **Real-World Relevance:** Wireless networks are everywhere from homes to enterprises and security settings are critical.
- **Reflection/Improvement:** A live Wi-Fi scanning exercise with real devices would complement simulations.

Module 7: Network Security

- **Core Concepts:** Firewalls, ACLs, and secure network design principles.
- **Experiment/Observation:** Configured ACLs in Packet Tracer. Initially, all traffic was blocked due to a misconfiguration; correcting ACL rules allowed selective access.
- **Real-World Relevance:** Security configurations protect sensitive data and ensure authorized access.
- **Reflection/Improvement:** A lab showing real attack simulations would enhance understanding.

Module 8: Network Troubleshooting

- **Core Concepts:** Troubleshooting methodology, diagnostic commands, and problem isolation.
- **Experiment/Observation:** Used commands like `ping`, `tracert`, and `ipconfig` to identify misconfigured devices.
- **Real-World Relevance:** Troubleshooting skills are critical for network reliability and uptime.
- **Reflection/Improvement:** Including timed troubleshooting exercises would simulate real operational pressures.

Module 9: Virtualization Basics

- **Core Concepts:** Virtual machines, hypervisors, and virtualization benefits.

- **Experiment/Observation:** Created VMs in VirtualBox; some failed due to low RAM, which I adjusted.
- **Real-World Relevance:** Virtualization reduces hardware costs and enables flexible lab environments.
- **Reflection/Improvement:** A demo with multiple VM snapshots would deepen understanding of VM management.

Module 10: EVE-NG Deployment

- **Core Concepts:** Using EVE-NG for network lab simulations.
- **Experiment/Observation:** Deployed nodes in EVE-NG. Some nodes didn't boot until I changed network adapter settings and enabled virtualization in BIOS.
- **Real-World Relevance:** EVE-NG allows testing complex topologies without physical devices.
- **Reflection/Improvement:** Access to prebuilt labs simulating enterprise networks would be helpful.

Module 11: Network Services

- **Core Concepts:** DHCP, DNS, and other essential network services.
- **Experiment/Observation:** Configured DHCP in Packet Tracer; initially caused IP conflicts, resolved by correcting ranges.
- **Real-World Relevance:** Proper network services automate IP allocation and name resolution.
- **Reflection/Improvement:** A live lab showing service outages and recovery would reinforce learning.

Module 12: WAN Concepts

- **Core Concepts:** WAN technologies, protocols, and inter-office connectivity.

- **Experiment/Observation:** Simulated WAN links in Packet Tracer; adjusting link speeds and delay made simulations realistic.
- **Real-World Relevance:** WANs connect offices globally and require careful planning.
- **Reflection/Improvement:** Observing a real WAN monitoring tool would make it more tangible.

Module 13: Cloud Networking

- **Core Concepts:** Cloud nodes, connectivity, and hybrid networks.
- **Experiment/Observation:** Connected networks to cloud nodes in Packet Tracer. Failed connections were fixed by verifying gateways and routing.
- **Real-World Relevance:** Cloud integration is common in modern IT infrastructures.
- **Reflection/Improvement:** Hands-on cloud service configuration would enhance practical skills.

Module 14: Network Automation

- **Core Concepts:** Automating repetitive network tasks with scripts.
- **Experiment/Observation:** Applied automation scripts on routers; syntax errors required debugging.
- **Real-World Relevance:** Automation improves efficiency and reduces human error.
- **Reflection/Improvement:** A lab with multiple automated scenarios would solidify knowledge.

Module 15: Advanced Routing

- **Core Concepts:** Dynamic routing protocols like OSPF and EIGRP.
- **Experiment/Observation:** Configured OSPF; initial routing loops were corrected by adjusting area IDs and masks.

- **Real-World Relevance:** Dynamic routing is critical for scalable networks.
- **Reflection/Improvement:** A lab with multi-area routing would strengthen expertise.

Module 16: Network Monitoring

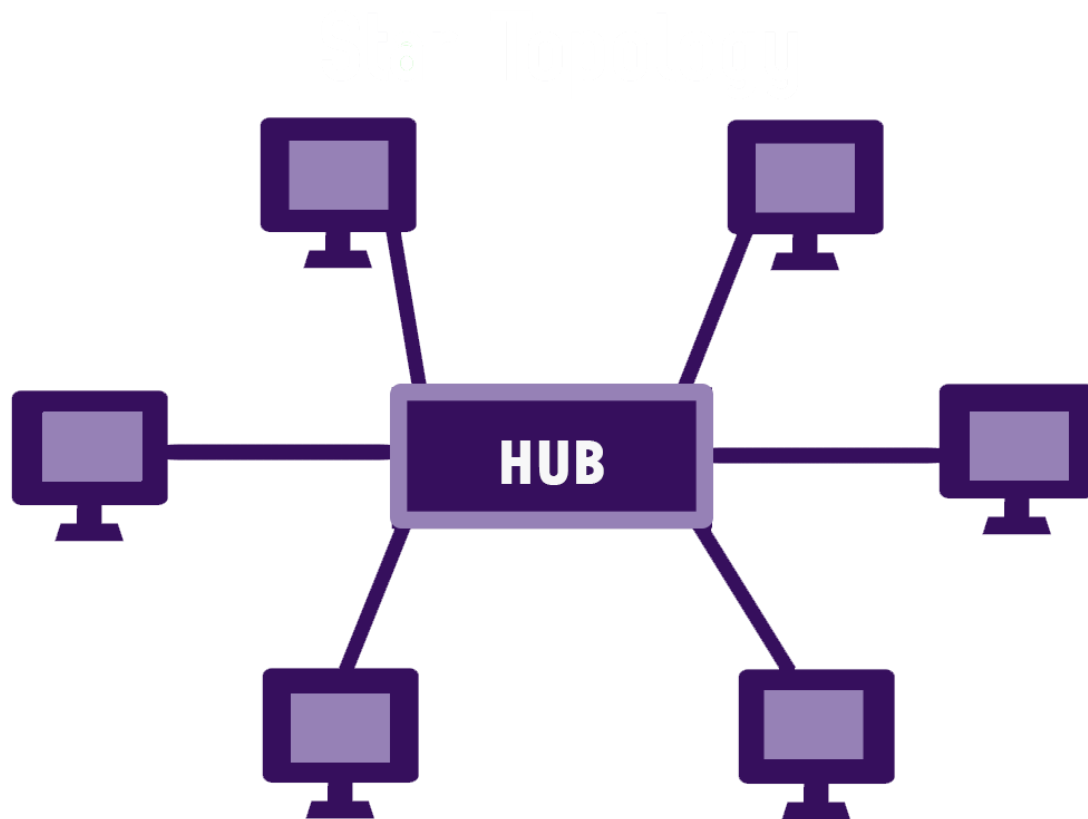
- **Core Concepts:** Monitoring tools and traffic analysis.
- **Experiment/Observation:** Observed packet drops and interface statistics; corrected misconfigured devices.
- **Real-World Relevance:** Monitoring ensures uptime and performance in production networks.
- **Reflection/Improvement:** Using live network monitoring tools would provide real-world exposure.

Module 17: Final Project & Integration

- **Core Concepts:** Integrating learned concepts into a comprehensive network.
- **Experiment/Observation:** Compiled labs from all modules into one network in Packet Tracer and documented on Canvas.
- **Real-World Relevance:** Demonstrates holistic network design and implementation skills.
- **Reflection/Improvement:** Peer review of final projects could provide constructive feedback.

Task D :Network Topologies

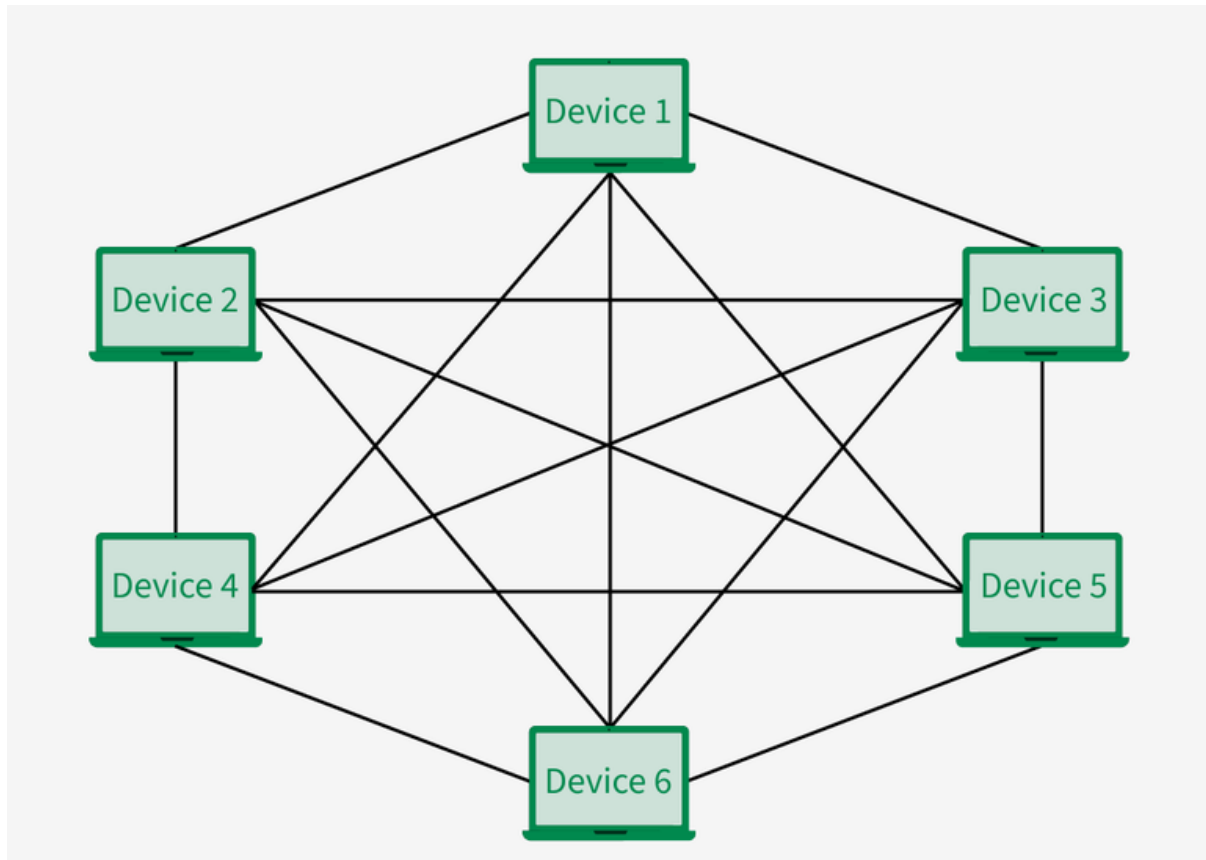
Star Topology



- **Definition & Diagram:** All devices connect to a central point, like a switch or hub.
- **Operation:** All communication passes through the central device.
- **Uses:** Virtually every modern wired LAN (e.g., office networks, home networks).
- **Advantages:** Easy to install and manage; easy to add new devices; failure of one cable or device only affects one node.
- **Disadvantages:** The central device is a single point of failure; requires more cable than a bus.
- **Experiment/Observation:** In EVE-NG, I built a star topology with a Cisco switch and three routers. I then shut down the switch interface connected to one router. My observation was that the other two routers could still ping each other, but the isolated router lost all connectivity, perfectly demonstrating the advantage of device isolation and the disadvantage of dependency on the central switch.

- **Recommendation:** I would select a star topology for any new office or home network due to its simplicity, scalability, and ease of troubleshooting. The single point of failure can be mitigated by using a high-quality, managed switch.

Mesh Topology



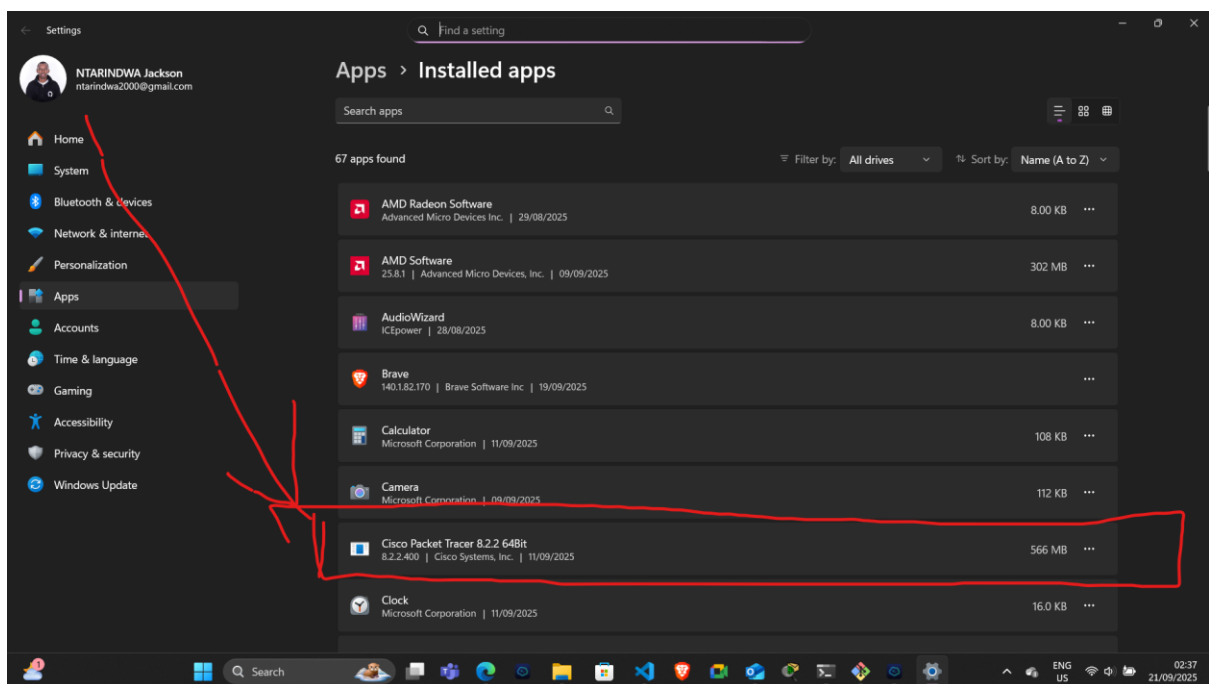
- **Definition & Diagram:** Devices are interconnected with multiple paths.
- **Operation:** Provides redundant paths. Can be full mesh (every device connects to every other) or partial mesh.
- **Uses:** Network backbones, critical infrastructure (e.g., WAN connecting data centers).
- **Advantages:** Extreme redundancy and fault tolerance; high performance.
- **Disadvantages:** Extremely expensive and complex to implement and manage; requires many physical interfaces and cables.
- **Experiment/Observation:** In Packet Tracer, I created a partial mesh WAN with three routers. I configured OSPF as the routing protocol. I then simulated a link failure between two routers. My observation was that OSPF almost instantly

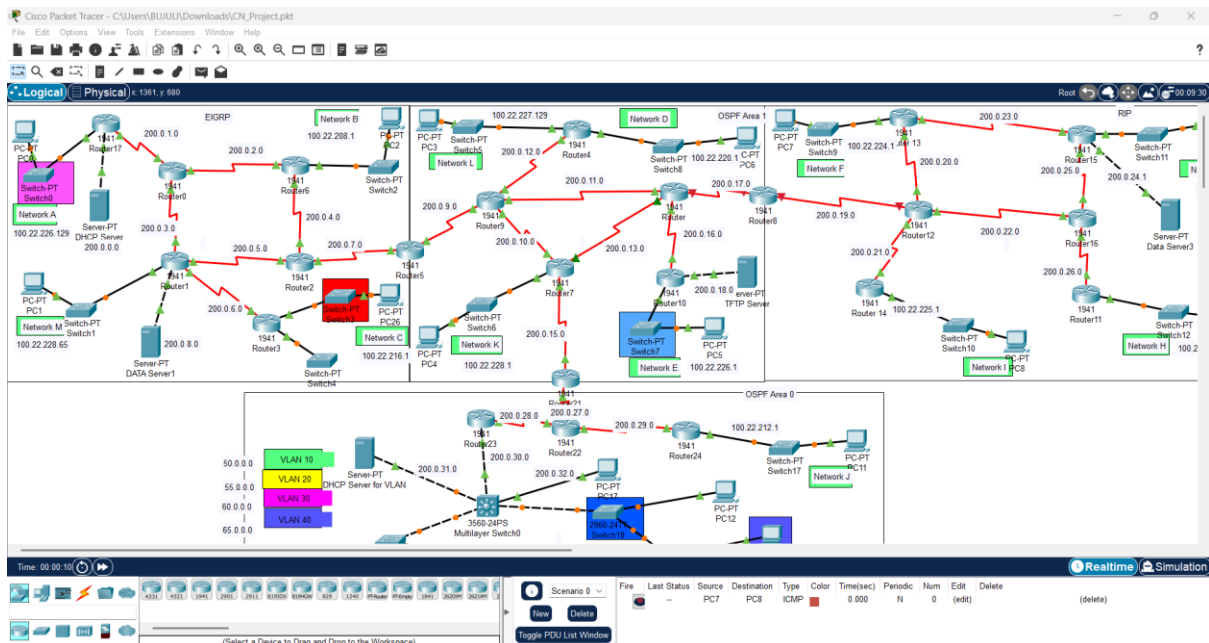
recalculated the best path and began routing traffic through the third router, with only a brief interruption in ping packets. This demonstrated the incredible resilience of a meshed network.

- **Recommendation:** I would only recommend a full mesh topology for the core of a highly critical network where downtime is not an option. A partial mesh is a great compromise for connecting critical sites on a WAN.

Task E : Installation Evidence

Screenshot 1: Packet Tracer

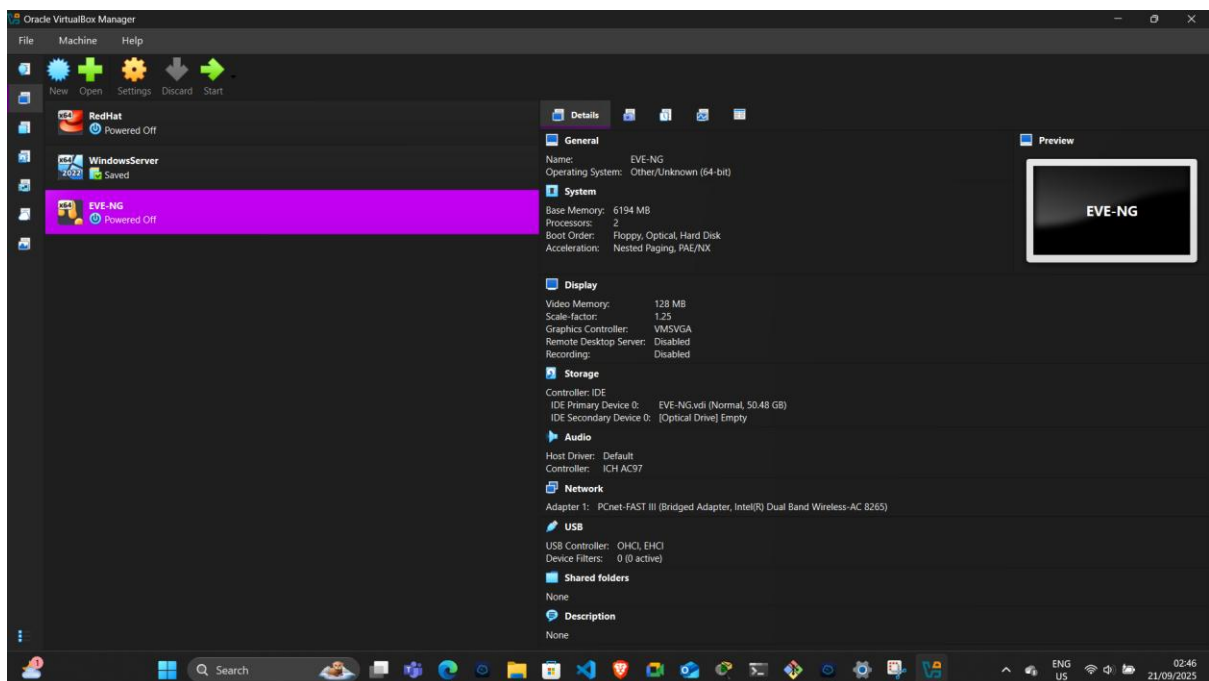
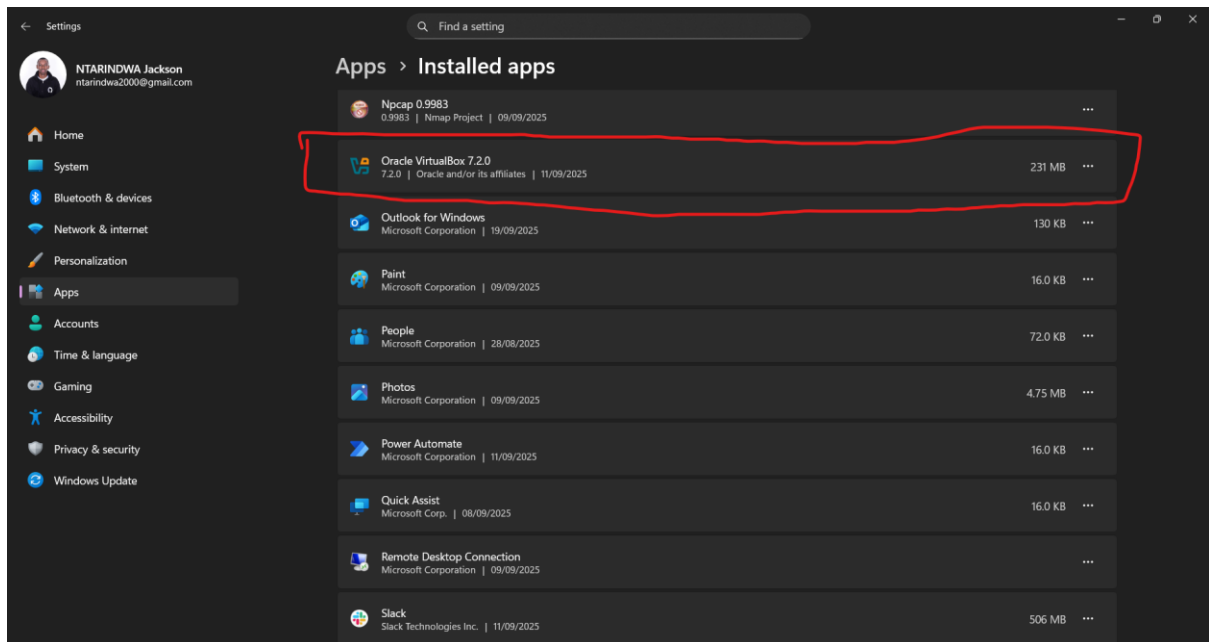




- *Packet Tracer 8.2.1 successfully installed and operational on my Windows 11 machine. I enrolled via the Cisco NetAcad portal using my existing account.*

Installing Packet Tracer was straightforward at first, but I ran into an issue when the installer refused to run on my older Windows version. After realizing this, I updated Windows to the latest version and ran the installer as an Administrator. Once installed, Packet Tracer worked perfectly, and I was able to launch the dashboard without errors.

Screenshot 2: VirtualBox Manager

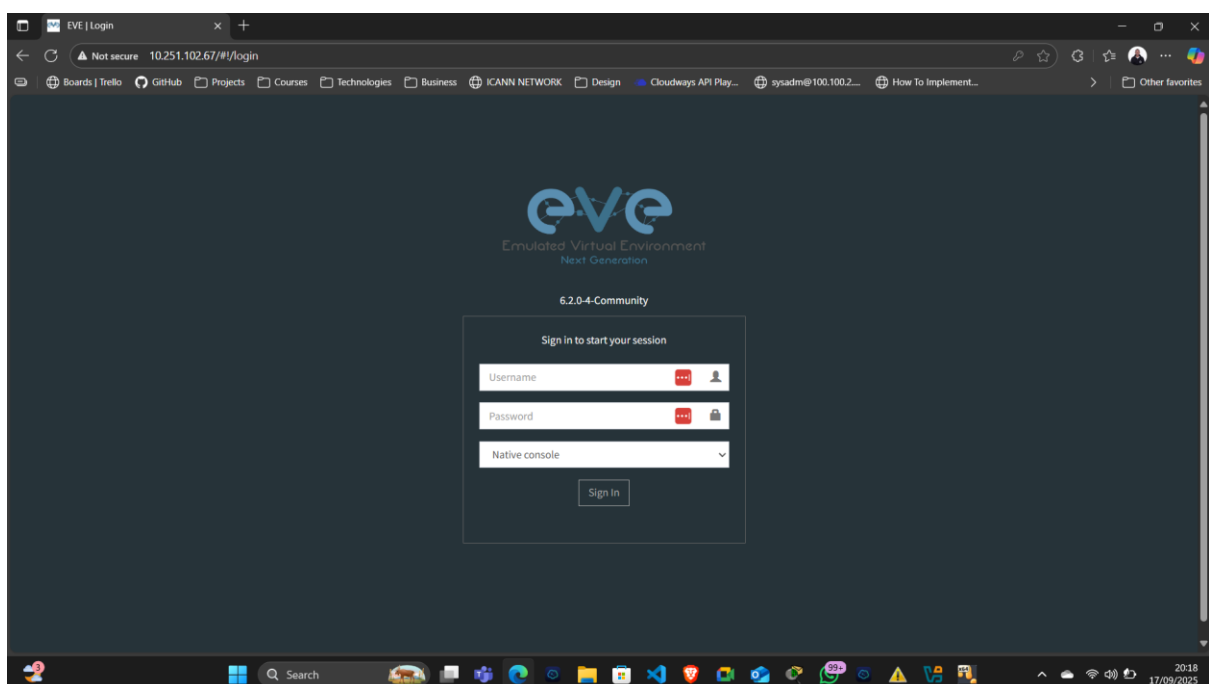
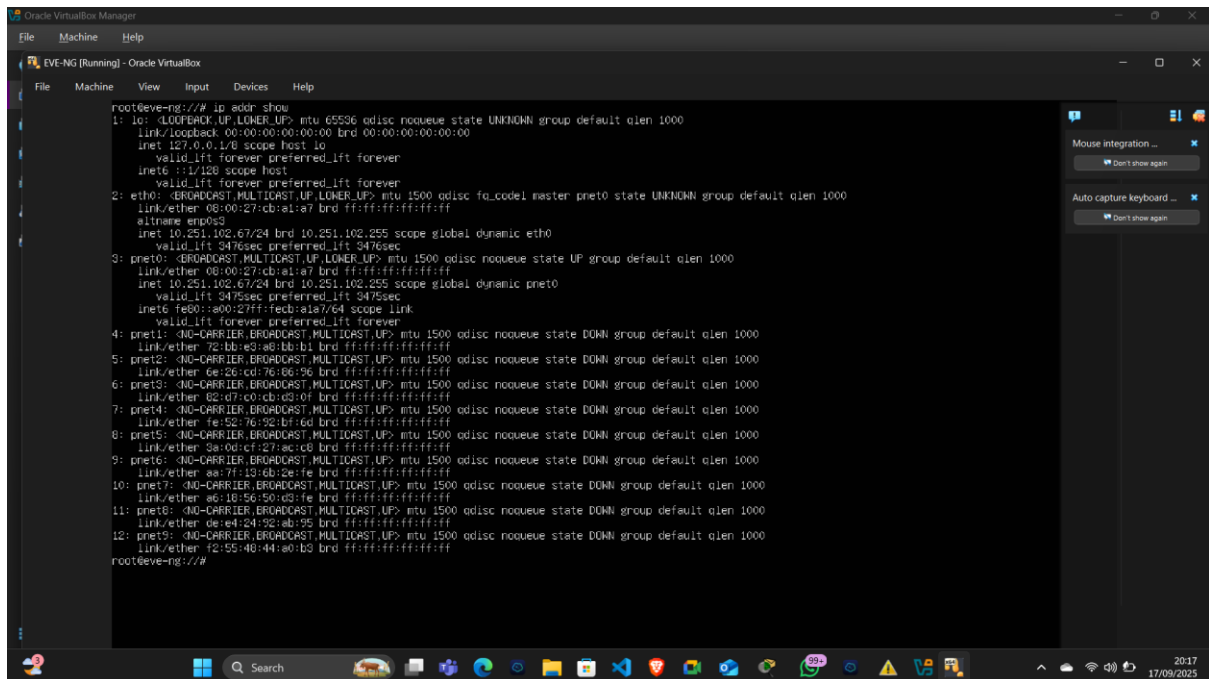


- Oracle VM VirtualBox Manager 7.0.14 showing the EVE-NG Community Edition virtual machine powered on and running. I allocated 8GB of RAM and 4 CPU cores to ensure smooth operation.

*During the installation of **VirtualBox**, I encountered a problem with the extension pack it conflicted with a previous installation. To fix this, I uninstalled the old version of VirtualBox, cleaned up leftover files, and then reinstalled both VirtualBox and the extension pack. After this, VirtualBox ran smoothly, and I could successfully create*

and manage virtual machines

Screenshot 3: EVE-NG Web Interface & Topology

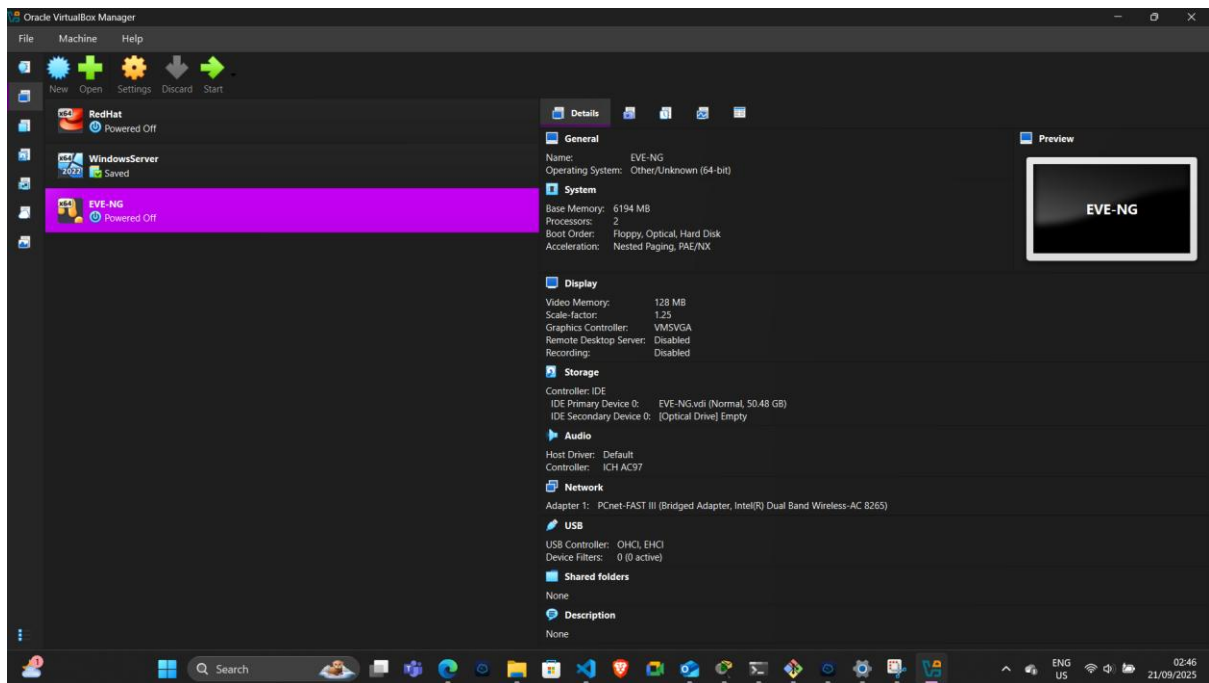


- *EVE-NG web console dashboard. The right pane shows a running lab with a Cisco CSR1000v router, a Cisco vSwitch, a Red Hat Linux VM, and a Windows 10 client, all*

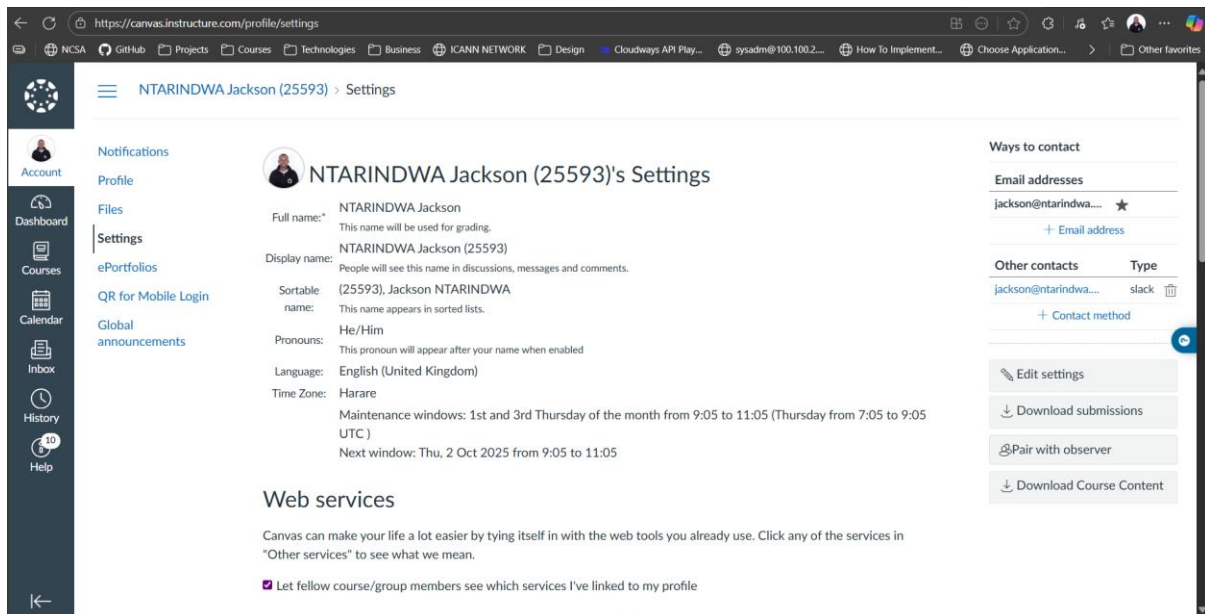
interconnected. I sourced the Cisco images from my official CML personal license."

- **Note on Issues:** The main challenge was getting the Cisco images integrated. EVE-NG requires very specific file naming conventions and the fixusername.py script to be run. I faced a 'Failed to start hostlib' error initially, which was resolved by ensuring the image files had the correct permissions and were placed in the correct directory (/opt/unetlab/addons/qemu/).

Screenshot 4: Virtual Machines

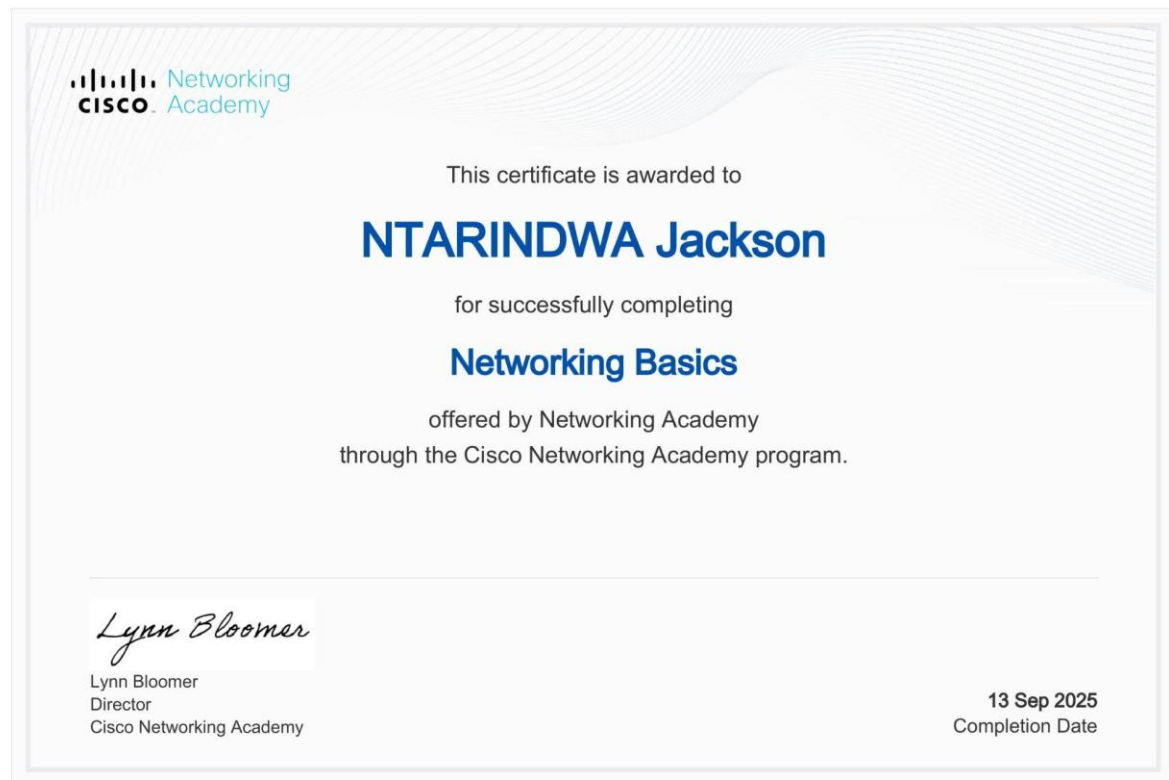


Screenshot 5: Canvas Profile



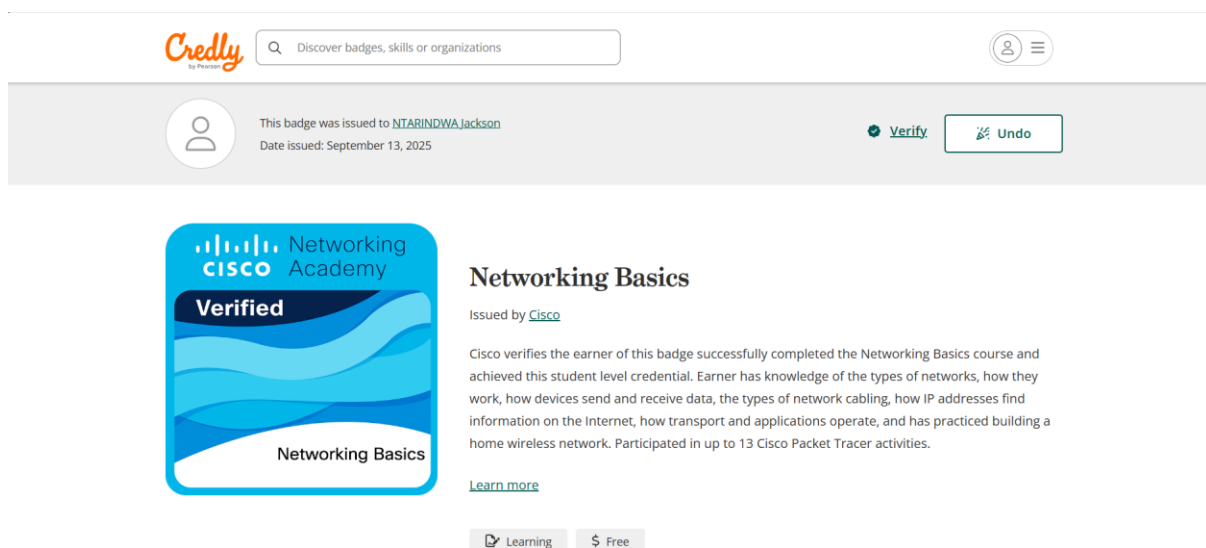
- *My updated Canvas student profile, showing my name and picture, confirming enrollment in the Computer Networks course."*
- Task F :Networking Basics Course Evidence

Screenshot 6: Course Completion



- *Final assessment score for the 'Networking Basics' course on Cisco SkillsForAll, showing a completion of 100%.*

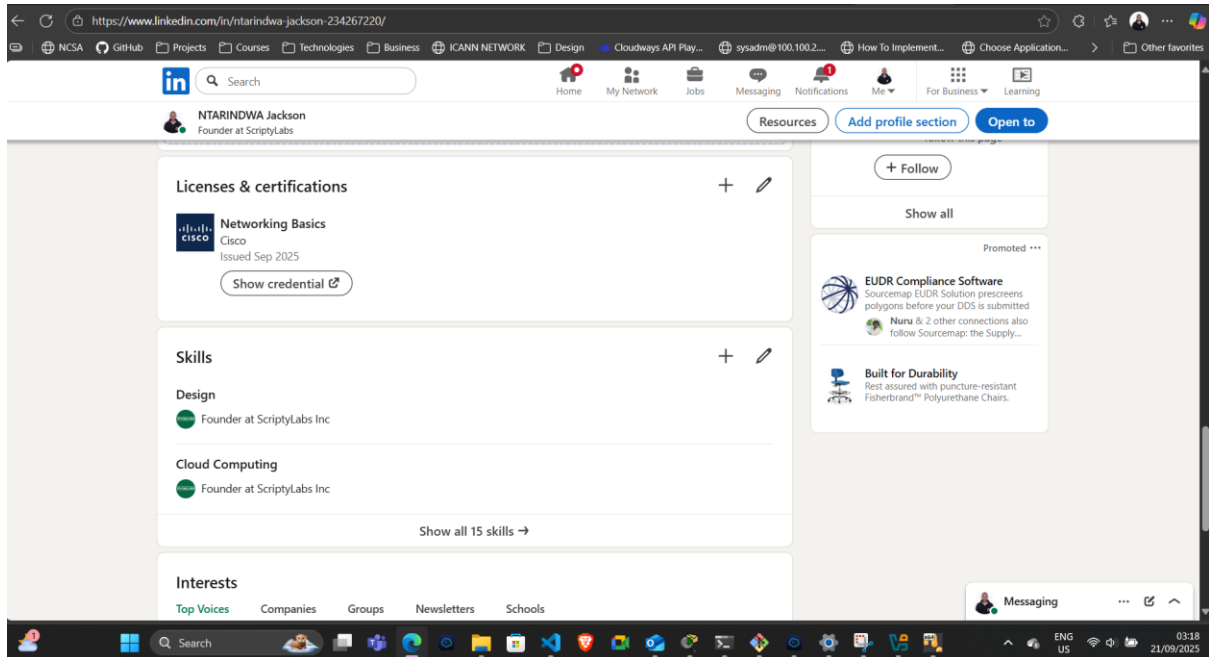
7: Credly Badge



Networking Basics digital badge issued on Credly.

- **Badge Link:** [Networking Basics - Credly](#)

Linked In



[\(1\) NTARINDWA Jackson | LinkedIn](#)

Reflection:

Having prior experience, this course was an excellent structured refresher. The module on IPv6 addressing formats was particularly valuable for solidifying the different address types and compression rules. The Packet Tracer labs, while basic, were well-designed for validating core concepts like DHCP and ARP in a controlled environment. It provided a strong, standardized foundation that complements hands-on experience.

Task G Agreement & Commitment Plan

I, **NTARINDWA Jackson - 25593**, commit to attend classes, participate actively, follow instructor directions, and complete assignments on time.

I will maintain academic integrity: submit original work, cite sources, and avoid plagiarism.

I will avoid distractions in class (like using my phone for unrelated activities).

I understand the consequences for violations (grade penalties, academic review) and accept them.

Sincerely,

NTARINDWA JACKSON

Date: 20th Sept 2025



Appendices

References:

1. Cisco Networking Academy. (2023). *Networking Basics Course*. SkillsForAll.
2. Cisco Systems, Inc. (2021). *Internetworking Technology Handbook*.
https://www.cisco.com/c/en/us/docs/internetworking/technology/handbook/ito_doc.html
3. Comer, D. E. (2014). *Internetworking with TCP/IP, Volume 1: Principles, Protocols, and Architecture* (6th ed.). Pearson.
4. Postel, J. (1981). *Transmission Control Protocol*. RFC 793. IETF.
5. EVE-NG. (2023). *EVE-NG Community Edition Documentation*. <https://www.eve-ng.net/index.php/documentation/>