A Remote Network Design Project That Turned Into a Contract

@Hays Tech #NetworkEngineering

#RoutingAndSwitching

#CiscoNetworking

#EnterpriseNetworking

#NetworkSecurity

#InfrastructureDesign

#Layer3

#PacketTracer

https://www.linkedin.com/pulse/remote-network-design-project-turned-contract-natospf-fadipe-vheje

The project started off as a remote technological exercise for an unidentified company. There was only a problem statement, a deadline, and the expectation that I could design, explain, and defend an entire enterprise-style network from scratch—no actual site visit or equipment rack in front of me.

This seemed like a straightforward lab, but it soon became evident that its true goal was to see if I could think like a network engineer, effectively explain my analysis, and troubleshoot in real time while being watched, rather than whether devices could ping each other.

Designing the Network Before Touching the CLI

I described the design I want to create before launching Packet Tracer or entering a single command.

I went with a hierarchical architecture because it emulates how actual networks are constructed, not because it looks good. I divided the architecture into distinct layers: Layer-2 access switches for end devices, multilayer switches for high-speed inter-VLAN routing, internal routers for segmentation and resilience, and an edge router for Internet access and security.

This established the atmosphere right away. Every device on the network had a function, and nothing was positioned at random, so the company could see that the network had intent.

Why an Edge Router Matters

The edge router was located at the top of the topology. Its function was purposefully restricted to **NAT** and traffic routing to and from the ISP. There were no internal VLANs there, and no pointless services were activated. This router served as a clear security barrier between private networks from the open Internet.

I clarified that troubleshooting is made easier, security policies are made clearer, and the internal network is shielded from external exposure by centralizing NAT and Internet connectivity on the edge.

Building Redundancy Into the Core

I positioned two internal routers behind the edge router. The purpose of these routers was robustness, not complexity. Traffic would still be delivered upstream even if one path failed. **OSPF** strengthened this redundancy, enabling dynamic network adaptation without the need for manual reconfiguration.

Two multilayer switches, which served as the network's central component, were located beneath them. Each VLAN's SVIs were hosted by these MLS devices, which also offered fast inter-VLAN routing. I clarified that this method offloads work from the routers while keeping routing near to the users.

Structured Addressing With Purpose

I subnetted the 10.0.0.0/8 private address space for IP addressing in a clear and deliberate manner: 10.60.16.0/20, 10.60.32.0/20, 10.60.48.0/20 ,10.60.64.0/20.

Every subnet matched a certain VLAN and purpose. Readability, scalability, and ease of troubleshooting were more important than simply having a enough number of IP addresses. Anyone could see where traffic belongs at a look.

VLANs, Trunks, and Access in Action

I set up VLANs 100, 200, 300, and 400 as the access layer developed, designating uplinks as trunks and user ports as access ports. Across trunks, only necessary VLANs were permitted, enhancing efficiency and security.

I demonstrated how, despite sharing physical infrastructure, this design keeps departments cognitively distinct and avoids needless broadcast traffic.

Turning on the Brain of the Switch

I turned the multilayer switches from basic switches into Layer-3 forwarding devices when I activated **IP routing** on the SVIs. This instance was significant because it illustrated the distinction between a network that appears configured and one that actually routes traffic.

The network came to life after routing was turned on.

OSPF Bringing the Network Together

Routes spread dynamically when OSPF was configured across routers and MLS devices. I described how the edge router injected a default route so that all internal devices could automatically figure out how to get to the Internet.

I used popular OSPF verification and troubleshooting commands like these during the demonstration:

- `show ip route`
  `show ip ospf neighbor`
  `show ip ospf interface`

This demonstrated that I understood how to confirm and debug routing behavior rather than just following instructions.

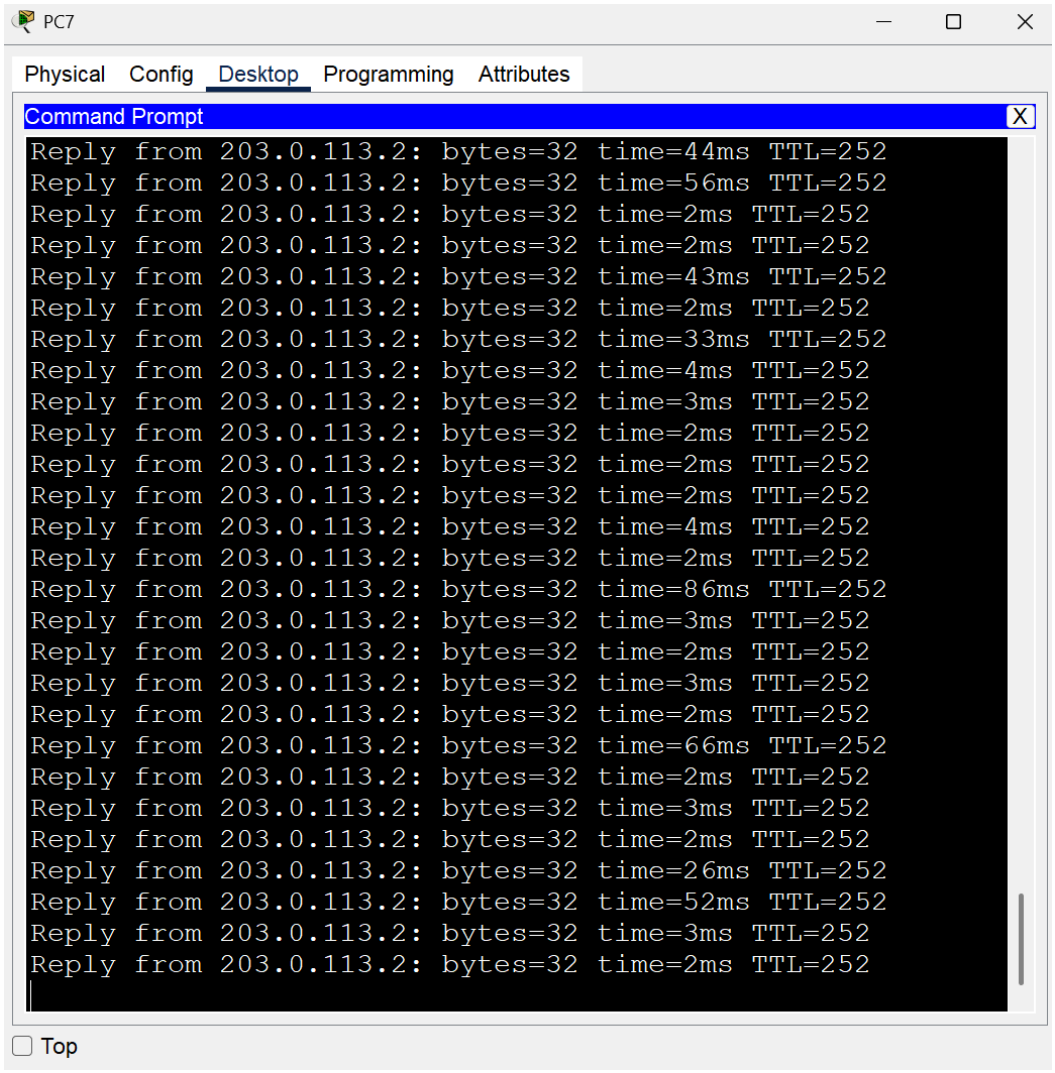
NAT: The Moment Everything Connected

PAT (NAT overload) was the project's pivotal moment. Internal private addresses were converted into a single public IP address for the ISP after being configured on the edge router.

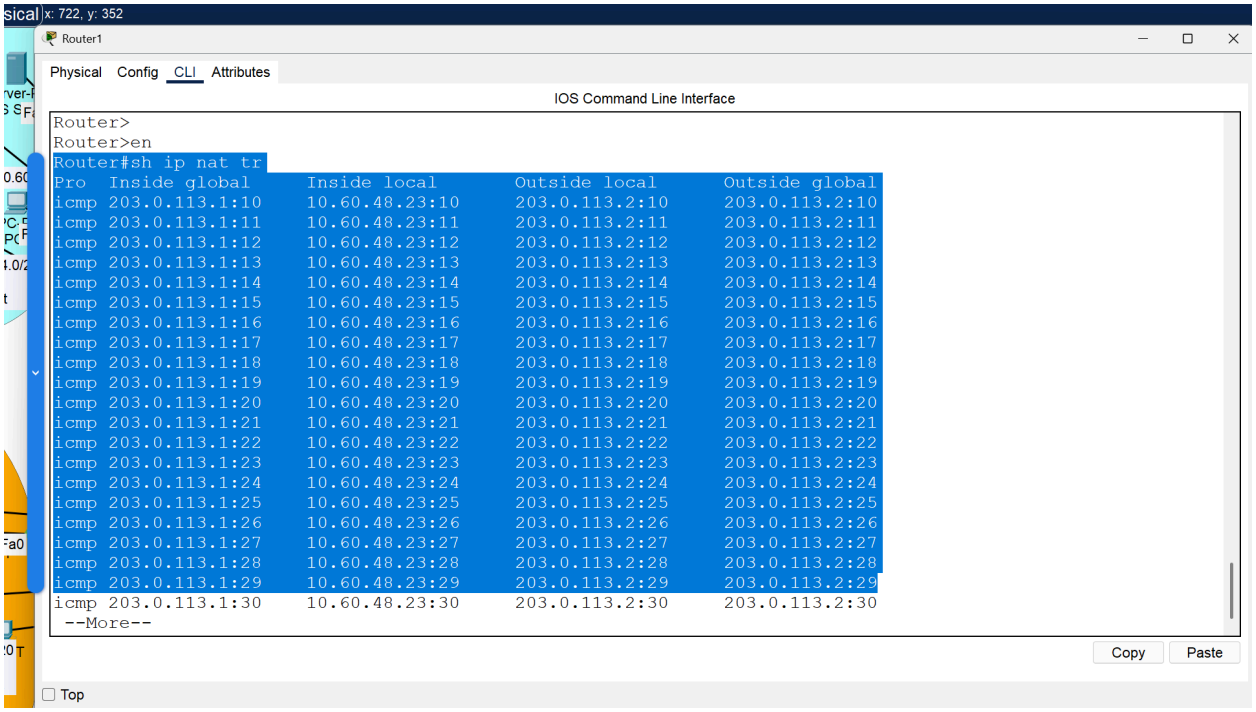
I used a PC to generate traffic and then show the NAT translation table to demonstrate this in real time. The design became tangible when a private address, such as 10.60.48.23, was converted to a public address. It was now practical rather than just academic.

Screenshots supporting this moment showed:

- Successful pings from PC to ISP

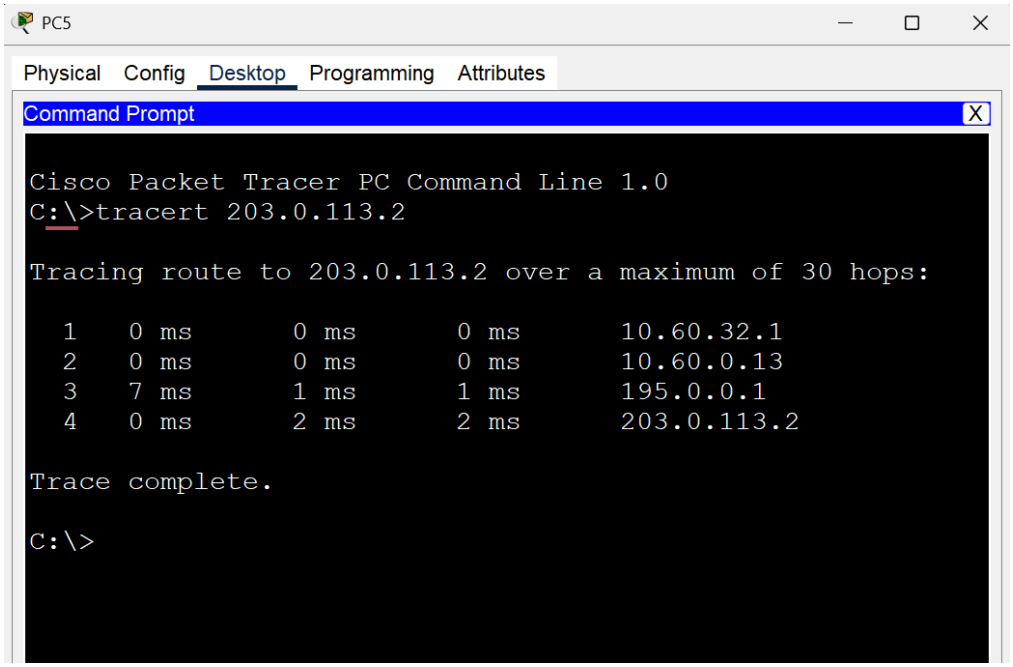


- NAT translation entries



The PC (10.60.48.23) sends a ping, R1 translates it to its public IP (203.0.113.1), sends it to the ISP (203.0.113.2), and keeps track using ICMP ID's.

- End-to-end connectivity



Completing the User Experience

In order to complete the network, I turned on DHCP, which allowed devices to automatically receive IP addresses along with the appropriate gateways and DNS details. Users just connected and got to work without any manual setup.

This reaffirmed the notion that an effective network is both usable and functional.

The Outcome

What started off as a remote project evolved into a comprehensive technical assessment. By the end, the organization had a comprehensive, comprehensible, and defendable network design rather than merely a Packet Tracer file.

The combination of:

- Clear hierarchical structure
- A distinct security perimeter at the edge
- OSPF dynamic routing
- Resilience and redundancy
- Effective PC-to-ISP communication and NAT-based secure Internet access
- Subnet and VLAN architecture
- Self-assured troubleshooting

Result: **the project was accepted and the contract was given.**

More significantly, they observed self-assurance, structure, and leadership. Not only was the network constructed, but it was also justified, verified, and explained. In the end, that combination won the contract and strengthened my standing as a **solution engineer and team leader.**

My Final Reflection

This remote project stands as one of my proudest achievements because it combined technical depth with real-world impact. It wasn't about configuring devices—it was truly about designing trust, reliability, and security into a system that people could depend on. Project was successfully delivered.