Lab 2

I. Experimental Objective

To build a Nebula-based overlay network, enabling devices behind different NAT networks to establish secure and direct connections through a public Lighthouse server, and to test its connectivity and usability.

II. Experimental Environment and Topology

Node 1 (Lighthouse): Cloud server (Ubuntu 22.04.5 LTS), IP: 104.243.20.247, Virtual IP: 192.168.100.1

Node 2 (Client): Personal Windows PC, Virtual IP: 192.168.100.135

Network topology: Star structure, all nodes register and coordinate with Lighthouse as the center.

III. Main Steps and Key Commands

Environment Preparation: Download and extract the Nebula for Windows version on the Windows client.

Certificate Configuration: Use the ca.crt, your_name.crt, and your_name.key provided by the instructor for identity authentication.

Configuration File Modification: Correctly configure config.yaml, specifying the public IP and port of Lighthouse.

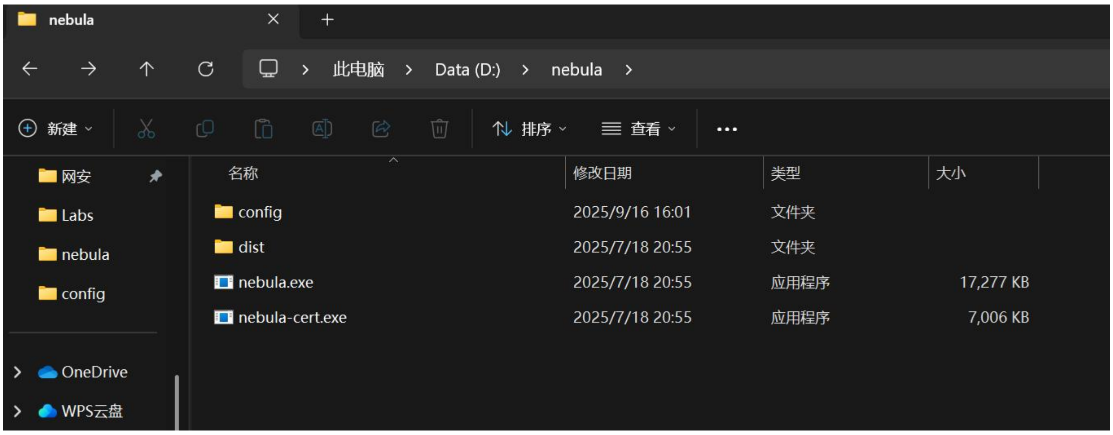
Driver Installation: Install the TAP-Windows virtual network card driver to create a virtual network interface for Nebula.

Starting the Connection: Run .\nebula.exe -config .\config\config.yaml on the client to start the Nebula service.

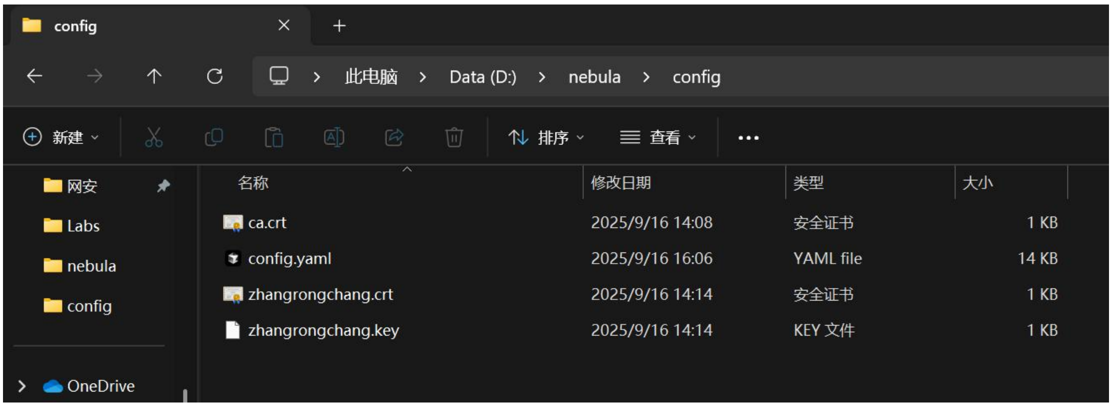
Connectivity Test:

ICMP Test: ping 192.168.100.1

Application Test: ssh nuist@192.168.100.1



名称	修改日期	类型	大小
config	2025/9/16 16:01	文件夹	
dist	2025/7/18 20:55	文件夹	
nebula.exe	2025/7/18 20:55	应用程序	17,277 KB
nebula-cert.exe	2025/7/18 20:55	应用程序	7,006 KB



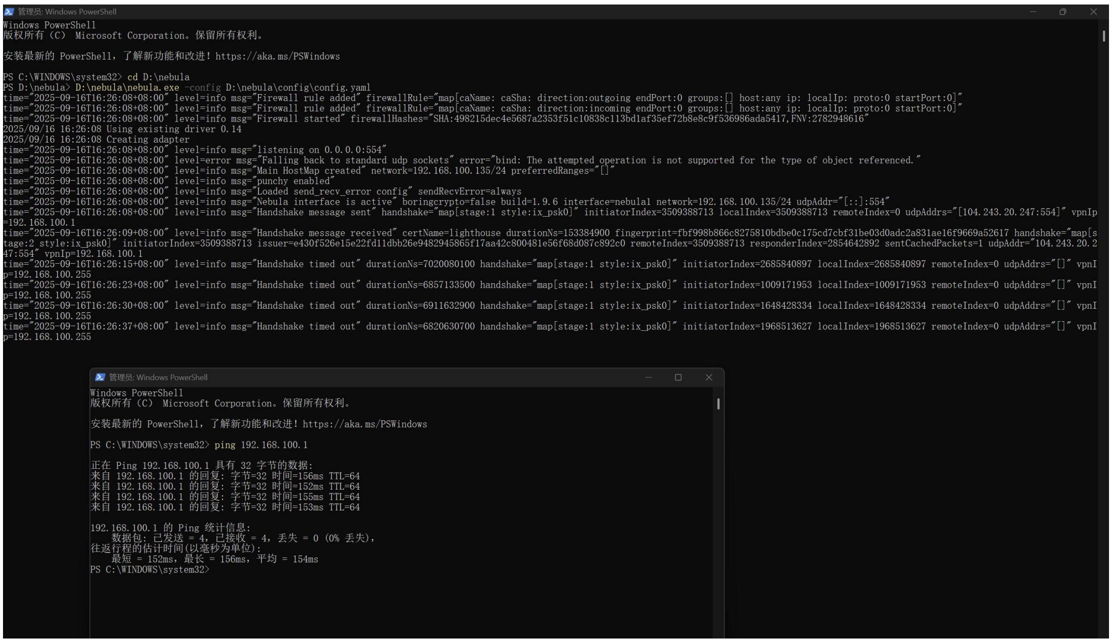
名称	修改日期	类型	大小
ca.crt	2025/9/16 14:08	安全证书	1 KB
config.yaml	2025/9/16 16:06	YAML file	14 KB
zhangrongchang.crt	2025/9/16 14:14	安全证书	1 KB
zhangrongchang.key	2025/9/16 14:14	KEY 文件	1 KB

IV. Experimental Results and Analysis

Result: The experiment was successful. The client was able to ping Lighthouse through the Nebula overlay network and successfully establish an SSH remote connection.

Analysis:

The experimental results show that Nebula effectively penetrated the client's NAT and established a secure P2P tunnel with Lighthouse. The successful SSH connection verified the stability and practicality of the Nebula network, which is sufficient to carry real network applications. The key success factors of this experiment lie in: correct certificate configuration, opening of the Lighthouse firewall port, and correct installation of the TAP driver.



```
Windows PowerShell
版权所有 (C) Microsoft Corporation。保留所有权利。

安装最新的 PowerShell，了解新功能和改进！https://aka.ms/PSWindows

PS C:\WINDOWS\system32> cd D:\nebula
PS D:\nebula> D:\nebula\nebula.exe -config D:\nebula\config\config.yaml
time="2025-09-16T16:26:08+08:00" level=info msg="Firewall rule added" firewallRule="map[caName: caSha: direction:outgoing endPort:0 groups:[] host:any ip: localIp: proto:0 startPort:0]"
time="2025-09-16T16:26:08+08:00" level=info msg="Firewall rule added" firewallRule="map[caName: caSha: direction:incoming endPort:0 groups:[] host:any ip: localIp: proto:0 startPort:0]"
time="2025-09-16T16:26:08+08:00" level=info msg="Firewall started" firewallHashes="SHA:498215dec4e5687a2353f51c10838c113bd1af35ef72b8e8c9f536986ada5417,FNV:2782948616"
2025/09/16 16:26:08 Using existing driver 0.14
2025/09/16 16:26:08 Creating adapter
time="2025-09-16T16:26:08+08:00" level=info msg="listening on 0.0.0.0:554"
time="2025-09-16T16:26:08+08:00" level=error msg="Falling back to standard udp sockets" error="bind: The attempted operation is not supported for the type of object referenced."
time="2025-09-16T16:26:08+08:00" level=info msg="Main HostMap created" network=192.168.100.135/24 preferredRanges="[]"
time="2025-09-16T16:26:08+08:00" level=info msg="punchy enabled"
time="2025-09-16T16:26:08+08:00" level=info msg="Loaded send_recv_error config" sendRecvError=always
time="2025-09-16T16:26:08+08:00" level=info msg="Nebula interface is active" boringcrypto=false build=1.9.6 interface=nebula1 network=192.168.100.135/24 udpAddr="[::]:554"
time="2025-09-16T16:26:08+08:00" level=info msg="Handshake message sent" handshake="map[stage:1 style:ix_psk0]" initiatorIndex=3509388713 localIndex=3509388713 remoteIndex=0 udpAddrs="[104.243.20.247:554]" vpnIp
=192.168.100.1
time="2025-09-16T16:26:09+08:00" level=info msg="Handshake message received" certName=lighthouse durationNs=153384900 fingerprint=fbf998b866c8275810bdbe0c175cd7cbf31be03d0adc2a831ae16f9669a52617 handshake="map[s
tage:2 style:ix_psk0]" initiatorIndex=3509388713 issuer=e430f526e15e22fd11dbb26e9482945865f17aa42c800481e56f68d087c892c0 remoteIndex=3509388713 responderIndex=2854642892 sentCachedPackets=1 udpAddr="104.243.20.2
47:554" vpnIp=192.168.100.1
time="2025-09-16T16:26:15+08:00" level=info msg="Handshake timed out" durationNs=7020080100 handshake="map[stage:1 style:ix_psk0]" initiatorIndex=2685840897 localIndex=2685840897 remoteIndex=0 udpAddrs="[]" vpnI
p=192.168.100.255
time="2025-09-16T16:26:23+08:00" level=info msg="Handshake timed out" durationNs=6857133500 handshake="map[stage:1 style:ix_psk0]" initiatorIndex=1009171953 localIndex=1009171953 remoteIndex=0 udpAddrs="[]" vpnI
p=192.168.100.255
time="2025-09-16T16:26:30+08:00" level=info msg="Handshake timed out" durationNs=6911632900 handshake="map[stage:1 style:ix_psk0]" initiatorIndex=1648428334 localIndex=1648428334 remoteIndex=0 udpAddrs="[]" vpnI
p=192.168.100.255
time="2025-09-16T16:26:37+08:00" level=info msg="Handshake timed out" durationNs=6820630700 handshake="map[stage:1 style:ix_psk0]" initiatorIndex=1968513627 localIndex=1968513627 remoteIndex=0 udpAddrs="[]" vpnI
p=192.168.100.255
```

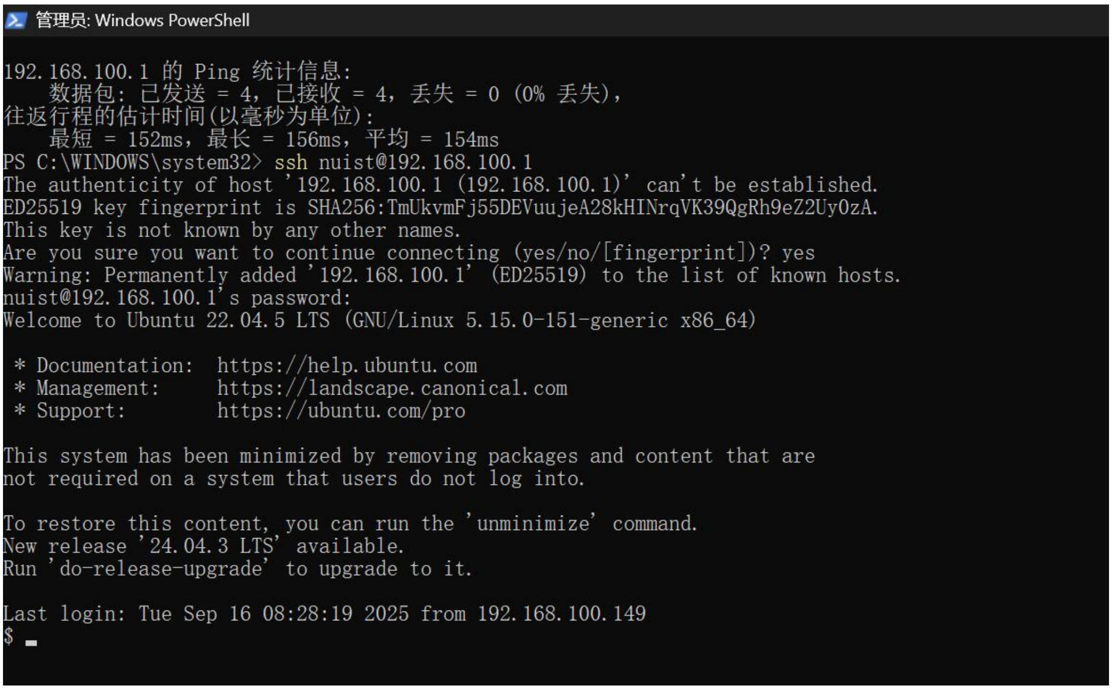
```
Windows PowerShell
版权所有 (C) Microsoft Corporation。保留所有权利。

安装最新的 PowerShell，了解新功能和改进！https://aka.ms/PSWindows

PS C:\WINDOWS\system32> ping 192.168.100.1

正在 Ping 192.168.100.1 具有 32 字节的数据:
来自 192.168.100.1 的回复: 字节=32 时间=156ms TTL=64
来自 192.168.100.1 的回复: 字节=32 时间=152ms TTL=64
来自 192.168.100.1 的回复: 字节=32 时间=155ms TTL=64
来自 192.168.100.1 的回复: 字节=32 时间=153ms TTL=64

192.168.100.1 的 Ping 统计信息:
    数据包: 已发送 = 4，已接收 = 4，丢失 = 0 (0% 丢失)，
往返行程的估计时间(以毫秒为单位):
    最短 = 152ms，最长 = 156ms，平均 = 154ms
PS C:\WINDOWS\system32>
```



```
192.168.100.1 的 Ping 统计信息:
    数据包: 已发送 = 4，已接收 = 4，丢失 = 0 (0% 丢失)，
往返行程的估计时间(以毫秒为单位):
    最短 = 152ms，最长 = 156ms，平均 = 154ms
PS C:\WINDOWS\system32> ssh nuist@192.168.100.1
The authenticity of host '192.168.100.1 (192.168.100.1)' can't be established.
ED25519 key fingerprint is SHA256:TmUkvmFj55DEVuujeA28kHINrqVK39QgRh9eZ2Uy0zA.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.100.1' (ED25519) to the list of known hosts.
nuist@192.168.100.1's password:
Welcome to Ubuntu 22.04.5 LTS (GNU/Linux 5.15.0-151-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.
New release '24.04.3 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Tue Sep 16 08:28:19 2025 from 192.168.100.149
$ 
```

V. Conclusion

Through this experiment, we successfully verified that Nebula, as a software-defined network (SDN) tool, can quickly and securely build overlay networks, solve the interconnection and intercommunication problems in complex network environments, and has high practical value.