Wireshark Student Lab Guide

Hands-on packet analysis aligned with common Windows CMD networking commands

Version: 12 Mar 2026

What you will learn
- Capture live traffic safely
- Use display filters (ICMP, DNS, ARP, TCP, TLS)
- Match CMD commands to packet-level evidence
- Interpret hop-by-hop routing and basic troubleshooting
- Write a short lab report with screenshots and answers

Ethics & Safety Notice: Capture traffic only on networks and devices you are authorized to use. Do not capture passwords or private content. Stop capture when the lab tasks are completed.

Contents

1. Setup & Environment

This guide assumes you are using Windows 10/11. If you use macOS or Linux, the commands differ, but the Wireshark concepts remain the same.

Required software

- Wireshark (installed with Npcap).
- Google Chrome or any modern browser.
- Windows Command Prompt (CMD) or PowerShell.

Before you start

- Close unnecessary apps (cloud sync, downloads) to reduce background noise.
- Connect to one network interface only (Wi-Fi or Ethernet).
- If you use a VPN, disconnect for the lab (VPN changes routing and DNS).

Deliverables

- A short lab report (1–3 pages) with screenshots and answers.
- Your capture file (.pcapng) for Labs A–D (optional if your instructor requests).

2. Wireshark Basics

Wireshark captures packets (frames) from a network interface. A capture is like a microscope: it shows what actually moved across the wire.

2.1 Choose the correct interface

- Open Wireshark.
- On the home screen, select the interface that shows live traffic (moving graph).
- Typical names: Wi-Fi or Ethernet.

2.2 Start and stop a capture

- Click Start Capturing Packets (shark fin icon).
- Let it run only while you perform a task (10–30 seconds).
- Click the red Stop button when done.

2.3 Save your capture

- File → Save As…
- Name it using: Lastname_Firstname_LabA.pcapng

2.4 Useful columns

If your packet list is hard to read, you can add columns. Right-click a field in the Packet Details pane and choose Apply as Column.

Column	Why it matters
Time	Shows sequence and delays between events
Source / Destination	Who talked to whom
Protocol	Which protocol (ICMP, DNS, ARP, TCP, TLS)
Info	Quick summary of the packet purpose

3. Display Filters You Must Know

Display filters do not change what you captured. They only change what you see.

Goal	Filter
Show ICMP (ping)	icmp
Show DNS only	dns
Show ARP only	arp
Show TCP only	tcp
Show TLS only	tls
Show traffic to/from a specific IP	ip.addr == 8.8.8.8
Show HTTPS traffic	tcp.port == 443
Show DNS queries for a name	dns.qry.name == "google.com"

Tip: Use multiple conditions with and, or.

```
Example:
(ip.addr == 8.8.8.8) and icmp
```

4. Lab A — ICMP (ping) and RTT

Objective: Observe ICMP Echo Request / Echo Reply and compare Wireshark timestamps with ping RTT.

CMD task

```
ping 8.8.8.8
```

Wireshark steps

- Start a new capture.
- Run the ping command (4 packets by default).
- Stop capture after ping finishes.
- Apply display filter: icmp.

What to find (evidence)

- At least 4 Echo (ping) request packets.
- At least 4 Echo (ping) reply packets.
- Match requests and replies using the Identifier and Sequence number fields.

Questions (write answers in your report)

- What is the source IP and destination IP for the Echo Request?
- What is the TTL value in the Echo Reply? What does TTL mean in general terms?
- Pick one request/reply pair and estimate RTT using Wireshark timestamps. Is it close to the ping output?

Screenshot requirement

- One screenshot showing the ICMP filter and both request + reply selected (packet list visible).
- One screenshot of Packet Details where Identifier/Sequence fields are visible.

5. Lab B — DNS (nslookup) name resolution

Objective: Observe DNS query and response, and identify the resolved IP addresses.

CMD task

```
nslookup google.com
```

Wireshark steps

- Start a new capture.
- Run nslookup as shown above.
- Stop capture.
- Apply display filter: dns.

What to find (evidence)

- A DNS Standard query for google.com (A and/or AAAA).
- A DNS Standard query response containing answers.
- Identify the DNS server IP (it is the destination of the query).

Questions

- What DNS server IP did your PC query?
- How many IP addresses were returned for google.com? List at least two.
- Did you see A records (IPv4), AAAA records (IPv6), or both?
- Explain why one domain might return multiple IP addresses (hint: load balancing/CDN).

Screenshot requirement

- One screenshot showing a DNS query and its response (dns filter applied).
- One screenshot of the Answer section in Packet Details.

6. Lab C — ARP (arp -a) and MAC resolution

Objective: Observe how a host finds the MAC address for an IP on the local network (often the default gateway).

CMD tasks

```
ipconfig
arp -a
```

Wireshark steps

- Start a new capture.
- If there is no ARP traffic, open any website once (this often triggers ARP).
- Stop capture.
- Apply display filter: arp.

What to find (evidence)

- An ARP request like: Who has 192.168.0.1? Tell 192.168.0.X
- An ARP reply like: 192.168.0.1 is at aa:bb:cc:dd:ee:ff
- Compare the gateway IP from ipconfig with the ARP packets.

Questions

- What is your default gateway IP (from ipconfig)?
- In the ARP request, which device is asking, and which IP is it trying to resolve?
- What MAC address is returned in the ARP reply?
- Why is ARP needed even though we already have IP addresses?

Screenshot requirement

- One screenshot showing ARP request and ARP reply (arp filter applied).
- One screenshot of your `arp -a` output in CMD (or a text copy in report).

7. Lab D — TCP and TLS (opening a website)

Objective: Observe the TCP 3-way handshake and TLS packets when opening an HTTPS website.

Browser task

Open a fresh browser tab and visit: https://example.com (or another allowed site).

Wireshark steps

- Start a new capture.
- Open the website once (wait until it loads).
- Stop capture.
- Apply display filter: tcp.port == 443 (for HTTPS).

What to find (evidence)

- A TCP handshake: SYN → SYN, ACK → ACK.
- TLS packets (Client Hello / Server Hello may appear) if you filter by tls.
- The server IP and port (destination should be 443).

Questions

- Which packet is the SYN? What flags do you see?
- What is the destination port for HTTPS?
- Why can’t you easily read website content in Wireshark for HTTPS? (hint: encryption)

Screenshot requirement

- One screenshot showing the TCP handshake packets (tcp.port==443 filter).
- One screenshot showing TLS packets (tls filter).

8. Lab E — Traceroute (tracert) and TTL / Time Exceeded

Objective: Connect the idea of hop-by-hop routing to packet behavior. Traceroute works by manipulating TTL so routers respond when TTL reaches zero.

CMD tasks

```
tracert -d google.com
```

Wireshark steps (advanced)

- Start a new capture.
- Run tracert -d google.com.
- Stop capture.
- Try these display filters (one at a time): icmp and udp and ip.ttl.

Notes (important)

Depending on Windows version and network policies, you may or may not clearly see ICMP Time Exceeded messages. Firewalls and routers can block traceroute responses. If you do not see them, write that observation in your report.

What to find (if visible)

- ICMP Time-to-live exceeded messages from intermediate routers.
- A sequence of packets with increasing TTL values (sometimes visible in IP header).

Questions

- Why does tracert show multiple ‘hops’?
- What does TTL protect the network from (conceptually)?
- If some hops show '* * *', give two possible reasons.

9. CMD + Wireshark Quick Comparison Sheet

Task	CMD command	Wireshark filter	What you should observe
Connectivity to an IP	ping 8.8.8.8	icmp	Echo Request/Reply, RTT, TTL
Name resolution	nslookup google.com	dns	Query + response, A/AAAA answers
Local MAC resolution	arp -a (after traffic)	arp	Who-has / is-at mapping
HTTPS website	(open https://example.com)	tcp.port==443 or tls	TCP handshake + TLS packets
Route (hops)	tracert -d google.com	icmp (maybe) / ip.ttl	Time Exceeded or TTL changes

Instructor tip: Do the CMD command first (students see the symptom), then show the packets in Wireshark (students see the mechanism).

10. Submission Checklist & Report Template

Use this section as your lab report structure.

10.1 Submission checklist

- All questions answered (Labs A–D required; Lab E optional unless assigned).
- At least 6 screenshots total (as specified per lab).
- Filters are visible in screenshots (top filter bar).
- Screenshots show the relevant packet selected and Packet Details expanded.
- File naming is correct: Lastname_Firstname_LabReport.pdf (or .docx).

10.2 Report template

Title: Wireshark Packet Analysis Lab
Name: _____ ID: _____ Date: _____

Lab A (ICMP)
1) Source IP: ____ Destination IP: ____
2) TTL value observed: ____ Meaning of TTL: ____
3) RTT estimate from Wireshark: ____ ms; ping RTT: ____ ms; Comment: ____
Screenshot(s): [A1], [A2]

Lab B (DNS)
1) DNS server IP: ____
2) Returned IP addresses (list): ____
3) Record types observed (A/AAAA): ____
4) Why multiple IPs? ____
Screenshot(s): [B1], [B2]

Lab C (ARP)
1) Default gateway IP: ____
2) ARP request: who-has ____? tell ____
3) ARP reply MAC: ____
4) Why ARP is needed: ____
Screenshot(s): [C1], [C2]

Lab D (TCP/TLS)
1) Destination port for HTTPS: ____
2) Handshake sequence (SYN, SYN-ACK, ACK): ____
3) Why content is not readable (HTTPS): ____
Screenshot(s): [D1], [D2]

Lab E (Traceroute - optional)
Observations: ____
Answers: ____
Screenshot(s): [E1]

Academic integrity: Do your own capture. Your IP addresses and hop paths will differ from other students.