6. Project 4. Above IP

Important

Please read the following instructions carefully:

1. This project is to be completed by each group individually. Use your own code to complete the main parts of the tasks.
2. This project contains 6 tasks. 2 of them are optional (8 points + 2 points).
3. The task score does not reflect the implementation difficulty but rather the recommended level.
4. Suggested workload is 2~6 FULL days. Manage your time.
5. Submit your code through Blackboard. The submission due date is **Jan. 11, 2026.**
6. Each group needs to submit the code once and only once. Immediately after TAs' checking. The submission is performed by one of the group members.
7. Tasks with "Optional" tag are optional tasks. Their due date is the end of the semester. Optional tasks are graded based on performance criteria and the results of the code review.
8. Unless otherwise mentioned, the instructor is responsible for grading optional tasks. Contact the instructor to check if you have finished one or more of them.
9. *Task 5, and 6* are graded by TAs.
10. Tasks are graded according to their hierarchy. The hierarchy of this project is: *Task 1* < *Task 5*. A full score of one task automatically guarantees the full score of non-overdue preceding tasks (left side of "<").
11. Free to use any tools for debugging, but only the provided toolkit is permitted in performance assessment.
12. During the performance assessment, any task can only be attempted up to 5 times.

6.1. Overview

At this stage, we have built a basic yet functional network using audio signals. Above the IP layer lies a vast, rich, and open network realm. Before wrapping up the course project, adding some decorations to your piece can make it more appealing.

6.2. Task 1: (4 points) DNS

DNS (Domain Name System, RFC 882) associates user-friendly string names with IP addresses. This task revisits Project 3: Task 3. Upgrade the NAT in `NODE2` to support TCP and UDP. Configure the local DNS server on `NODE1` to `NODE2_3.IP` (need to implement a local DNS server on `NODE2`), 10.15.44.11, or 1.1.1.1, so that `NODE1` can resolve domain names.

Performance Assessment

The setup is similar to Project 3: Task 3. The group provides NODE1 and NODE2, and connects them with the toolkit according to Figure 4.2. Connect NODE2 to the campus LAN and disconnect NODE1 from the campus LAN.

In the Aethernet program or the system terminal of NODE1, enter the following content:

```
# NODE1
ping some-name-dot-com -n 10
```

- Objective 1 (1 point). TAs provide the domain name `some-name-dot-com`. TAs check the reachability but do not count the RTT.
- Objective 2 (3 points). TAs use Wireshark on NODE2 to monitor the DNS query and response packets. TAs should verify that NODE2 functions as a local DNS server, providing recursive resolution on behalf of NODE1: NODE2 performs step-by-step iterative queries to external DNS servers.

6.3. Task 2: (1 points) HTTP

HTTP (Hypertext Transfer Protocol, RFC 1945) defines a way to distribute text and multimedia information over the Internet. Implement a simplified TCP handshake and sliding window scheme on NODE1 to allow it to visit public HTTP resources.

Tip

- Thanks to those who host public HTTP servers for testing purposes: example.com, 101.pku.edu.cn, guozhivip.com.
- Developer tools of web browsers can be used to inspect the web content
- TCP tools, such as telnet and nping, can be used to bypass TCP implementations when debugging.
- If you have not finished the Project 3: Task 6, you cannot directly use the system's curl tool. Therefore, you need to implement a curl-like command within the Aethernet program to fetch HTML pages.

Performance Assessment

The setup is similar to Project 3: Task 3. The group provides NODE1 and NODE2, and connects them with the toolkit according to Figure 4.2. Connect NODE2 to the campus LAN, and disconnect NODE1 from the campus LAN.

The group specifies the initial sequence numbers the TCP connections of NODE1 from the set: 0x $H_1H_2H_3H_4H_5H_6H_7H_8$, where, $H_i \in \{1,2,3,4,5,6,7,8\}$ and $H_i \neq H_j$, e.g., 0x1234 5678, 0x8765 4321, 0x4321 8765.

In the system terminal OR the Aethernet program of NODE1, enter the following or equivalent content:

```
# NODE1
curl http://www.example.com
```

curl fetches the HTML page of the requested URL via HTTP. TAs verify the returned HTML content. TAs verify the specified sequence number of the TCP SYN segment from `NODE1` through Wireshark on `NODE2`.

6.4. Task 3: (3 points) Project Report

Please summarize your four course projects in a report (approximately 2000 words). Include your stories, designs, challenges, solutions, and any suggestions you may have for the course. The teaching team highly values your feedback and is continually working to improve the course. Submit the report through a separate link on Blackboard. Its deadline slightly later than the code submission deadline.

6.5. Part 4. (0 point) Relaunch

Please return the borrowed toolkit to the TAs after completing this project. The toolkit is important to students in upcoming semesters, so please ensure it is in perfect condition. Returning the toolkit is a prerequisite for the TAs to upload the group’s project grades.

6.6. Part 5. (Optional, 1 point) Browsing the Web

Performance Assessment

In Task 2, use any web browser to open and render the web content, e.g.,:

```
# NODE1
start chrome www.example.com
```

TAs verify the rendered HTML content.

6.7. Part 6. (Optional, 1 point) Project Report

We will make the project report in Task 3 publicly available. Please explicitly add the following information below your report title so that we can comply with your preferences.

Subtitle	Disclosure Preference	Percentage Earned
your name (and affiliation)	disclose content with names	100%
“Anonymous Authors”	anonymous disclosure	90%
blank or other content	keep it offline	0%