CS120 PROJ REPORT

Lu Yuzhou 2021533001
Cheng Dinghao 2020533144

Designs

baseline: C++ + JUCE

PJ1

Handle node1 and node2 with physical interface, i.e. sound wave.
We use PHASE-ctrl method. We design a header sinewave of around 2000Hz for 480 samples, the carrier is a 4000Hz sinewave, each bit will take 24 samples, containing 2 cycles. When decoding the recordings, we use carrier to multiply the range of samples one by one, always choose the exactly largest volume sample as the begin of a header. For the info bits, after successfully matching the header, we match the position of the first peak of each sample, take the average>0 to judge if it is 1. For the ECC, we choose hanmming 7 of 4.

PJ2

Only Task1 & Task2. Get a package then resend, never imple sliding windows.
We use a new set of PHYSICAL layer. Header: (4*1 + 4*-1) *6. Carrier: (+1,+1,-1,-1)*2.
When recognizing the header, for the speed reuqirements, we choose to set a threhold based on the cable connection and resistance in the whole system.
When judging the bits, we choose to only consider the first cycle, judge ([0]+[1] >[2]+[3]). In the implementation of this project, we never imple a model with directly-divided layers, just BOUND each needed info as ints and strings to combine together and translate into bits to convey.
You can check our package design in 'utils.h'.

PJ3

Link layer similar to PJ2, pls check PJ2 parts. Compared to PJ2, we disigned more things on LINK layer, SEQ number, IP int32, Payload string, Identifier HEX-string... All to the similar solutions to PJ2.
Use PYTHON files to solve NAT and router functions on NODE2.
'scapy' here can help us solve many problems related to TCP and ICMP fares. It can efficiently reform or hack a package then send to the Internet, also listen the package on the WLAN to transfer to the Aethernet. Quite useful.
We use scapy to send ICMP packages and sniff the reply, then write to a file as the information transfer from python to c++ process.
For routers, catch the package receive on Aethernet, change the appropriate addr then send to the destination.
Using python here is much less complicated than using socket module in c++.
You dont need to send ACK!

PJ4

For dns enquiry module, use python module 'dnspython' to acquire dns from the selected domain,, then modify the Aethernet package to transfer domain as payload, and the dns result as int32.
For http curl function, one implematal way is to manully imple tcp/ip sliding windows and 3-shaking, 4-waving scheme. You can use scapy here to realize editing SYN number upto your requirements.
This part is the most complicated part for the PJ4. Thus is most playable.

Challenges

PJ1

The physical soundwave is too reliable to provide a 100% correct solution. One reason is the lack of our tool chain and code work, the other is the junk devices, for example, the soundcard, speaker and microphones are highly varied and need much work to adjust them to the best. Some groups may also benefit from their expensive, self-provided divices, thus not fair to others.
The soundwave here can be said is not productive in any consequence.
This PJ is the most tiring assignments of the whole semester, for it never worths productive work except only adjusting digits and fighting with junk devices.

PJ2

Lack of usable nodes.
For my teammates never participate in certain impletation, I only use my laptop and desktop for development. Even when evaluating, I was noticed that the nodes should be separated, then urgently borrowed one of my classmates' laptop. Stupid course.
Thankfully it didn't have much difference. Lack of implementation instructions.
All resoureces I can rely on is the teachings from the past CS120ers. From the lecturers? not any. I hope they can give more instructions abt the projects.

PJ3

The encode and decode of utf-8 and hex-strings on the linklayer. We should use our own implementation of BYTE to bit conversion here. And it should stay consisitent when decoding.
The pack up of a ICMP package in scapy also need reading documents.

PJ4

Complicated TCP scheme relisation, such as sliding windows and 3-shake-4-wave.

Shortcut Solutions

PJ1

No, Never.

PJ2

Distinct from others and myself?
Use process ID of Windows system.
We carry process ID in every package, so even I can only test the solution on one computer, I can easily avoid self packages.

PJ3

Use logs to get test infomation to speed up your development.
You can use std::fstream like this : logout << endl; Thus give you easy and reliable status report and broke.
Use scapy to sniff and send packages. You can almost modify everything, even mimic another device.

PJ4

for task2, never transfer all http contents on the Aethertnet.
Take Node 1 as a display of status for Node 2. Only trnasfer SYN num and HTTP status only to pass the check.

Stories

I would like to complain about the work division balance in this teamwork.
It could be a nightmare for a non forth-year student to team up with a forth-year student.
For third-year students, no matter how hard you are trying to find a partner, NEVER choose a forth-year one.
I highly appreciate some suggestions given by my teammate. But except those suggestions, he had not done any codework or designs in this project for the whole semester. Even both two nodes were provided by myself for he was too tired to get his laptop runable for the solution.
Anyway, I think it is a little over-workload for a third-year student, but you can handle it only by yourself.
Having a teammate to throw jobs on is a delay for your project, although you understand he will never do any job.

Suggestions for the lecturers

PJ1

1. change to electical wires instead of sounds.
2. give specific microphones & speakers.

PJ2

1. more instructions.
2. usable public laptops.

PJ3

1. give more instruction on virtual network adapter.
2. give more time when ping 1.1.1.1 .
3. should check package loss rate during the ping and routing process.

PJ4

I prefer ftp problems in contract with simple HTTP impletation.