

## [Note] A good debugging test for ARP response

**Details:** It's very hard to see where possibly your code has gone wrong (or if it's working properly). Well here's a slight debugging test. Before running any of the routers, add `print_hdrs(packet)` before processing the packet in `handle packet`. Then add `print_hdrs(your created packet)` after your ARP response is created. When you run the router you should get something like:

```
vagrant@vagrant:/vagrant$ ./router
-- 11/12/17 02:18:16.784 ./router: Network: trying to establish tcp connection to 127.0.0.1:8888
-- 11/12/17 02:18:16.785 ./router: Network: tcp connection established
    local address = 127.0.0.1:42846
    remote address = 127.0.0.1:8888
Resetting SimpleRouter with 3 ports
sw0-eth1 (192.168.2.1, 2e:d2:dc:04:f2:1d)
sw0-eth2 (172.64.3.1, e2:2e:f3:7f:a9:df)
sw0-eth3 (10.0.1.1, 4a:e6:0d:75:ab:1d)
-- 11/12/17 02:18:16.789 ./router: Network: attempting to bind to tcp socket :::65500
-- 11/12/17 02:18:16.789 ./router: Network: listening for tcp connections at :::65500
    local interfaces: 10.0.2.15, 10.0.1.100, 127.0.0.1, 0:0:0:0:0:0:1
-- 11/12/17 02:18:16.790 ./router: Network: published endpoints for object adapter `Tester':
    tcp -h 10.0.2.15 -p 65500:tcp -h 10.0.1.100 -p 65500
-- 11/12/17 02:18:16.790 ./router: Network: accepting tcp connections at :::65500
```

In your mininet, type command “client ping 10.0.1.1” (this will send ICMP packets from client to the router on 10.0.1.1 face. We are not coding ping for this project but for Ping to work, it sends an ARP request first)

The moment the first packet is dropped CTRL + c on the mininet ssh window to stop the ping. Your router window should then display the ARP requests packet client has sent and the ARP response packets your router has sent.

When you then run “client arp 10.0.1.1” you will get the arp cache of the client for entry 10.0.1.1 like so: (it should be filled out correctly if successful)

```
mininet> client ping 10.0.1.1
PING 10.0.1.1 (10.0.1.1) 56(84) bytes of data.
^C
--- 10.0.1.1 ping statistics ---
1 packets transmitted, 0 received, 100% packet loss, time 0ms

mininet> client arp 10.0.1.1
Address      Hwtype      Hwaddress    Flags Mask      Iface
10.0.1.1     ether       4a:e6:0d:75:ab:1d  C              client-eth0
mininet>
```

---

## [Ans] How do I safely exit the SSH connections & Close Vagrant?

**Details:** Hi! How do I close the ssh connection to mininet/vagrant when I'm done? Also, when I try shutting down my computer I get the notification that there are still active connections to VirtualBox? How do I shut down the VM from vagrant?

**Answer:** To exit the ssh connections, you just type “exit” in the console/command prompt. To shutdown the VM started by vagrant, use “vagrant halt”. It will safely shutdown the virtual machine but please make sure ALL ssh connections (including the one to vagrant) are closed before shutting down the machine.

---

## **[Ans]** Vagrant couldn't find the vm box “boxcutter/ubuntu1604”

### **Details:**

The box 'boxcutter/ubuntu1604' could not be found or could not be accessed in the remote catalog. If this is a private box on HashiCorp's Vagrant Cloud, please verify you're logged in via `vagrant login`. Also, please double-check the name. The expanded URL and error message are shown below:

URL: [<https://vagrantcloud.com/boxcutter/ubuntu1604>]

Error: The requested URL returned error: 404 Not Found  
(I am on Windows 10, Virtual Box 5.1.28)

**TA Answer:** Change the vagrant file to: config.vm.box="ubuntu/xenial64". It should run correctly on MacOS and Linux with this line. The first Vagrant up will take a while but after that, each vagrant up will be faster (not instantaneous but several minutes still). If on Windows (or any OS), better solution might be to update to VirtualBox 5.2 and see if it works with the new VirtualBox.

---

## **[Ans]** With vagrant up, get 403 Forbidden error

### **Details:**

default: Downloading:

<https://vagrantcloud.com/box-cutter/boxes/ubuntu1604/versions/2.0.26/providers/virtualbox.box>

An error occurred while downloading the remote file. The error message, if any, is reproduced below. Please fix this error and try again.

The requested URL returned error: 403 Forbidden

**Update/Comment:** Are you running vagrant up with administrator permissions? If so, then it might be the same boxcutter issue. If it is the boxcutter issue, please check out “Vagrant couldn't find the vm box”

---

[Ans] What ARP requests should be ignored?

Detail: The first bullet point in the requirement part of the ARP section of our spec says we should ignore “other ARP” requests. What kind of ARP requests belong to this type? Namely, how do we know if an ARP request is asking the MAC address of our router instead of MAC address of other hardware?

**Answer:** Although the pseudo code does not explicitly specify ignoring packets with non-matching IP address, you should drop such packets in your code. Thus, once you process the Ethernet header, next step would be to check if IP address matches the corresponding network interface and to drop the packet if there is a mismatch.

---

[Ans] Error running ‘vagrant up’ on Ubuntu 16.04

**Details:** There was an error while executing `VBoxManage`, a CLI used by Vagrant for controlling VirtualBox. The command and stderr is shown below.

Command: ["startvm", "5f1def71-81f7-4105-9286-5eafcf055aab", "--type", "headless"]

Stderr: VBoxManage: error: The virtual machine 'cs118-proj1\_default\_1510216064452\_5618' has terminated unexpectedly during startup with exit code 1 (0x1)

VBoxManage: error: Details: code NS\_ERROR\_FAILURE (0x80004005), component MachineWrap, interface IMachine

**Followup:** What version of VirtualBox are you using and what is your OS? I'm thinking it might be a VirtualBox issue and you should download the latest VirtualBox that's if your OS is Windows (that I know for sure of).

**Answer:** This is using VirtualBox 5.1.30 on a Ubuntu 16.04 OS. I figured out how to solve it if anyone else running Linux runs into the same problem: Use your package manager's version of VirtualBox instead of the download link provided in the project spec.

---

### [Ans] Still having file structure issue

**Details:** I loaded a new copy of the repo but in the core folder, main is there. Main has a problem where it cannot find simple-router.hpp. What files belong in core folder and what files belong outside of it?

**Answer:** That is intended. If you are using Visual Studio community please do not make a “solution” for it. The only reason why we “recommended” it was for the C++ editing benefits :) (Don’t overthink it. Trust the python code. Live with the python code) (And I now realize this after I look at my own code ^^)

---

### [Ans] Error message and output routing table

The provided code in `handlepacket()` print some error message and the routing table. When we handle errors, do we need to print error messages? If we do, should everyone’s message be consistent? Also, is printing routing table mandatory every time we handle a packet? Since the code for printing the routing table is given in the code, I assume we just start writing our code after that.

**Answer:** I believe we are not grading you on what your system outputs in error messages (just that we might grade you that it does output an error message in some cases). That doesn’t mean you can write inappropriate error messages though. Pretend as if you’re writing this code to give to a customer for their Software Defined Network (SDN) and so when they go to debug the router or want to know what happened, they have a clear enough message of what went wrong.

I don’t think you need to print the routing table each time we handle the packet, but check with Seungbae on that and update here when he gets back to you. (I believe we’re just checking that certain transfers work and others fail...)

---

### [Ans] Output interface for ARP reply

After an ARP request is received, we need to reply it. I want to make sure that the interface we reply it should just be the interface which received the request. Or we should check the routing table by with the source ip to find the output interface? I think these two should end up with same interface.

**Answer:** You can do either way :) The fastest is the interface we received the request on but you may want to add a check that the ARP request was for that interface :/ .

---

### [Ans] Usage of cksum

I feel confused about the usage of function cksum. Especially the argument **len**, what should be the length when do cksum to the ipv4 header? If we use the size of the ip\_hdr data type, what should be the field ip\_sum before we know the cksum of this header?

**Answer:** The specs of the IPv4 checksum is kind of confusing. The Len parameter of the function is the size of the IPv4 header. However, the checksum is computed WITHOUT the checksum field filled in. So ip\_sum is originally all 0s. To read up on it (since I think we haven't gotten there yet): [https://en.wikipedia.org/wiki/IPv4\\_header\\_checksum](https://en.wikipedia.org/wiki/IPv4_header_checksum)

---

### [Ans] Minimum length of IPv4 packet

What is the minimum length of an IPv4 packet that we need to check every time we receive one?

**Answer:** hint would be the sizeof() function. Minimal stuff needed for an IPv4 packet to be an IPv4 packet is just the IPv4 header.

---

### [Ans] When to use ntohs and htons

I think we may need to use ntohs and htons sometime so that the data received can be compared with local variables. However, are there any rules we can follow to make sure all variables are consistent?

**TA Answer:** I gave in... and Apparently you have to use htons. (got a bunch of packets dropping...) Unfortunately I can't tell you where to use htons since that is a large part of the project but the hint would be check out the printed packet hdrs and the respective header in the protocol file. Where in the console it looks different than the respective header constant (e.g. 2054 vs 0x0800) that's probably where you need htons.

As for when to use ntohs, I'm not too sure. It depends on your implementation.

>> I've updated this to answered as after asking the other TAs, it is required to use htons to get it working.

---

[Ans] What is class Buffer?

**Details:** For the function `handlePacket(const Buffer& packet, const std::string& iface)` I can't locate the class Buffer to see how it is laid out. Or is it just a char array?

**TA Answer:** Buffer is defined in `core/protocol.hpp`. Yes, it is a vector (a special array) of characters.

---

[Ans] How can we determine the router address ?

**Detail:** the above ^

**TA Answer:** We have provided `findIfaceByName`, `findIfaceByIP`, and `findIfaceByMac`. Given Name, IP addr, or MAC addr (respectively) it will return the Interface of the router if the name, IP addr, or MAC addr belongs to the router. The router is configured by the given `RTable` file in the repo. But this file may change when we are running the test script => do not hardcode the IP addresses or name of each interface. The MAC addr are randomly generated upon each bootup of mininet.

You do not need to write the code for Client or Server (if that was the main question...). (Those are already written in the nice exe files we have provided in the repo :) )

---

[Ans] Mininet not picking up Router code?

**Details:** I've coded my router but strangely when I run `make ./router`, it doesn't seem like mininet is picking up my new router code?

**Answer:** The command is "make" (hit the enter key and it will compile your code). When it stops, then run `./router`. You should see the resetting simple router message on the ROUTER ssh connection (not the mininet one). All of the error messages etc. will appear on the router window.

---

[Ans] What addresses should we change when forward

**Details:** I assume every time we forward a packet, we should update the source MAC address in the ethernet header but should keep the source ip address in the IPv4 header. Is this the correct understanding?

**Answer:** Yes :) So when you forward a packet through the network, the MAC addresses change (including the target MAC addresses). The only addresses that stay constant (or at least should stay constant, except in the case of NAT but that's beyond the scope of the project) are source and destination IP addresses

---

### [Ans] Submission files

**Details:** Does source code includes \*.py and \*.ice? Can we just include \*.cpp together with Vagrantfile, Makefile and README.md?

**Answer:** According to the Makefile included, the only thing not included for the submission is the .vagrant (the folder). So yes :) As long as you follow the submission instructions on the PDF file, you should be fine. (Let us know if the instructions don't work for you for some reason)

---

### [Ans] SSH password

**Details:** When trying to establish an ssh connection, I am prompted to enter ubuntu@127.0.0.1's password. The printed message in my console is:  
"The machine you're attempting to SSH into is configured to use password-based authentication. Vagrant can't script entering the password for you. If you're prompted for a password, please enter the same password you have configured in the Vagrantfile."  
I checked in the vagrantFile and couldn't find any password to enter.

**TA Answer:** Change config.vm.box line to config.vm.box = ubuntu/trusty64 in the vagrant file. (Verified by Aaron Chung) You'll need to set in your vagrant file username = vagrant, password = vagrant

---

### [Ans] Files not on virtual machine...?

I followed all of the steps to set up the machine, had to uncomment the xenial64 thing on the Vagrantfile. I'm working on a Windows 10 machine... Now when I vagrant ssh the files aren't in the vagrant machine. Are they supposed to be in there? What've I done wrong?

**TA Answer:** This actually pretty common (and I have to follow these steps all the time). If when you type "Ls" (all lowercase... just want to emphasize it's an "L") and it's empty, cd ../.. (two folders up). When you do "Ls", you should get a vagrant folder along with a bunch of other blue folders. You then cd into the vagrant folder and viola! All your files are there :)

---

[Ans] The meaning of two test commands of mininet

**Details:** Hi, can you give a detailed explanation about the meaning of “mininet> server1 /vagrant/server 5678 /vagrant/ &” and “mininet> client /vagrant/client 192.168.2.2 5678 /vagrant/test.file &”?

**TA Answer:**

The way we plan to test your code is by checking if the client can send a test file (in our case: test.file in the repo) to the server. This operation of sending a file will be managed by mininet at high level and all that you need to worry about is if your router recognizes the ARP and IP packets being sent from one node in the test network to another node and responds accordingly.

Thus, roughly speaking:

- The command "server1 /vagrant/server 5678 /vagrant/ &" is telling the server1 node to listen on port 5678
- The command "client /vagrant/client 192.168.2.2 5678 /vagrant/test.file &" is telling the client node to send the test.file to the server node at 192.168.2.2:5678

---

[Note] Forwarding & Checksum Debugging Test

**Details:** Per request, to more partially debug your code, we’ve created another intermediate debugging test! It’s similar to the good debugging test for ARP reply but will test part of your forwarding and your checksum. (I would do screenshots but the headers are too long and will take up too much space...)

Command for both tests on Mininet is “client ping server1”

**Forwarding Test**

After adding print\_hdrs() to where you are creating an ARP Request and to where you’re popping packets off the queue for ARP replies, when you run the ping, in your router SSH you should see the following flow during the FIRST time you ping: (client’s arp table is empty)

ARP Request (from eth3):

Ether\_hdr: CC:CC:CC -> FF:FF:FF

ARP\_hdr: 10.0.1.100 -> 10.0.1.1

ARP Reply:

Ether\_hdr: e3:e3:e3 -> CC:CC:CC

ARP\_hdr: 10.0.1.1 -> 10.0.1.100

IP Packet (from eth3):

Ether\_hdr: CC:CC:CC -> e3:e3:e3



```
IP_hdr: 10.0.1.100 -> 192.168.2.2
ARP Request:
  Ether_hdr: e1:e1:e1 -> FF:FF:FF
  ARP_hdr: 192.168.2.1 -> 192.168.2.2
ARP Reply (from eth1):
  Ether_hdr: s1:s1:s1 -> e1:e1:e1
  ARP_hdr: 192.168.2.2 -> 192.168.2.1
IP Packet (originating from router):
  Ether_hdr: e1:e1:e1 -> s1:s1:s1
  IP_hdr: 10.0.1.100 -> 192.168.2.2
```

If it's the second or subsequent time you're running ping and you've restarted your router (router's arp table is empty), you'll only see the last 4 of the above.

**NOTE:** I'm using the first 3 bytes of the MAC since it seems mininet's MAC generation isn't so random (first 3 bytes always turns out to be different from each other for me). The above MAC addresses are abstracted where CC:CC:CC represents client MAC, e1:e1:e1 is router's eth1 MAC, e3:e3:e3 is router's eth3 MAC, and s1:s1:s1 is server1's MAC.

### Checksum Test

If after the last IP packet of the above, you see an IP packet from eth1, your checksum is good :) If not, you'll need to debug it.

```
IP Packet (from eth1):
  Ether_hdr: s1:s1:s1 -> e1:e1:e1
  ARP_hdr: 192.168.2.2 -> 10.0.1.100
```

---

[Ans] How can I open a file in Sublime from the virtual machine?

**Details:** I really don't want to have to use emacs or vim. I downloaded Sublime on the VM but when I try to run "subl \*file name\*" I get a multitude of errors like this

```
Unable to load gtk_widget_modify_bg from libgtk-x11-2.0.so
Unable to load gtk_window_set_decorated from libgtk-x11-2.0.so
Unable to load gtk_window_set_skip_taskbar_hint from libgtk-x11-2.0.so
Unable to load gtk_window_set_skip_pager_hint from libgtk-x11-2.0.so
```

**Student Answer:** I believe you don't need to open sublime in the virtual machine. You can think of the virtual machine as being "on top" of the code on your computer. In the virtual machine the code is in the folder "vagrant", but on your computer the same code is in the folder cs118\_fall17\_project\_1. What the virtual machine does is make the code compiling/ testing/ mininet stuff platform independent.

ie. on computer: my code is located in cs118\_fall17\_project\_1

On the VM my code is located on the /vagrant folder

Any changes you make to the code on the computer will appear on the virtual machine. You can think of vagrant as having made a “link” to the code on your computer. So you can open sublime/ some code editor on the code on your computer (not the VM). Also I’m guessing you get those errors because in the Vagrant file the vm is set not to show.

**TA Answer:** I endorse the student answer above with the slight added emphasis on please please please do not install ANYTHING into your instance of the VM. You edit the files on your own OS via Visual studio, sublime, notepad++, or any other C++ editor.

---

[Ans] “:No such file or directory found”

**Details:** When I try running `sudo ./run.py` in the ssh thing, it gives me “:No such file or directory found”. It gives the same output if I run just `./run.py`, as well. I am using putty to ssh because I am on windows. Does anybody know how to go forward? Thanks. why

**Student Followup:** What do you get when you type “ls”?

Asker update: All the files in the cs118 project1 directory. Run.py is there, too.

Student followup, continued

Try typing “`sudo python ./run.py`” it is possible that the path variable or something wasn’t configured correctly from the vagrant file when it is run on windows (I ran into this problem also [ `/usr/bin/env: `python\r': No such file or directory`] and the step I listed is how I fixed it:). Let me know if it works.

*Asker update*

I don’t know...I fixed the previous problem by doing: dos2unix on the run.py file to change the carriage returns from windows to unix-style. Thats what messed it up before. I get this error when i run it now:

```
vagrant@precise64:/vagrant$ sudo ./run.py
```

```
Traceback (most recent call last):
```

```
File "./run.py", line 7, in <module>
```

```
    import mininet.net
```

```
ImportError: No module named mininet.net
```

```
|
```

**TA Update:** Asker updated vm box to Trusty64 and it seems to work :)

---

[Ans] Do we have to deal with Big/Little Endian with the packets?

**Details:** I remember the professor mentioning this in lecture but I can’t remember exactly what his response was. For example, will we potentially see an ethernet header type as 0608 and have to swap it to 0806?

**Student Answer:** Yes, I think so. For more specifics, check the `print_hdr_ip`, etc... methods. In those functions you see the function `ntohs` used (which is used to change the ordering).

**TA Answer:** We aren't having you deal with Big/Little Endian directly but primarily when comparing with the constants we're giving you, you will need to convert. You use `htons` or `ntohs`, depending on your preference. You'll need to see the question [\[Ans\] When to use ntohs and htons](#) for this :)

---

[Ans] Best way to create Buffer packet?

**Details:** After I create and fill the header structs with the appropriate values, what is the best way to convert all of this to a Buffer to send using `sendPacket`?

**TA Answer:** Not too sure if you're looking for the "best" way or a way that works. A way that works is hinted by `arp-cache.cpp`. But as for the "best" way, it depends what metric you're basing best on (e.g. memory used, variables, lines of code)

---

[Ans] What should the router do when it receives an ARP request which is not sent to the router?

**Details:** when I do `server1 ping server2`, the `server1` will send an ARP request to the router with the destination addr as `0xFFFFFFFF`. Should the router forwards this ARP request to `server2` or send back a reply? If the router just sends back a reply how could `server1` ping `server2` at the end?

**TA Answer:** It's a pretty lengthy explanation but I'll try my best with the below.

ARP Requests work as the following:

1. Client who wishes to ping/send a packet to `server1` will figure out its best next hop (the router's IP address on the interface connected to the client)
2. From the next hop IP address and subnet mask, client realizes next hop is in the same subnet. Client then sends an ARP request for the router
  - a. The ARP request carries in the target ip the Router IP address (the next hop IP address)
  - b. The destination MAC in the ethernet frame is set to broadcast
3. The router receives the ARP request, sees that the destination IP address is for it, and generates an ARP reply to the client

From the above, it's not possible for an ARP request to be generated for `server2` from `server1` since `server1` knows that `server2` is NOT on the same subnet in your project topology. Instead, `server1` will ask for router's MAC with the ARP. When `server1` gets the ARP reply, it will then send the packet to the router and hope the router will forward the packet to `server2`.

The router will then generate ARP request for server2 and hope server2 replies. Server2 and server1 will then be able to smoothly send packets back and forth using their cached entries (no need to send further ARP request and replies).

Ping is at a higher level than IP. So it is apart from ARP. ARP does not carry any payload after its fields (as in there aren't any headers after the ARP header).

---

[Ans] Making sure ethernet's destination matches the interface or the packet is broadcast...?

**Details:** So this is looking at the destination field of the ethernet header. I'd like to make sure it's either a broadcast packet or its destination is the interface it landed at. But testing if `ehdr->ether_dhost == BroadcastEtherAddr` is not giving me the right boolean. I'm printing the values out before comparing and they should be equal but it's giving me false. What am I doing wrong here?

... nvm I got it ... comparing pointers when i need to be comparing values good lordt demote me to cs32 right now...

**TA Comment:** no problems :) I ran into the same issues to and debugging those pointers were a pain :/

---

[Ans] SIMPLE QUESTION... i think

**Details:** I think I have a pretty good understanding of how ARP and IPV4 are supposed to work, but I am not sure if the way my code is working is correct right now. What I do is I keep looking for an ARP packet (and dropping the rest that aren't ARP or IPV4) until I get one. Then I check to see that it is an ARP request. It is. Then, I make a new packet with the updated things through the `sendPacket` function.

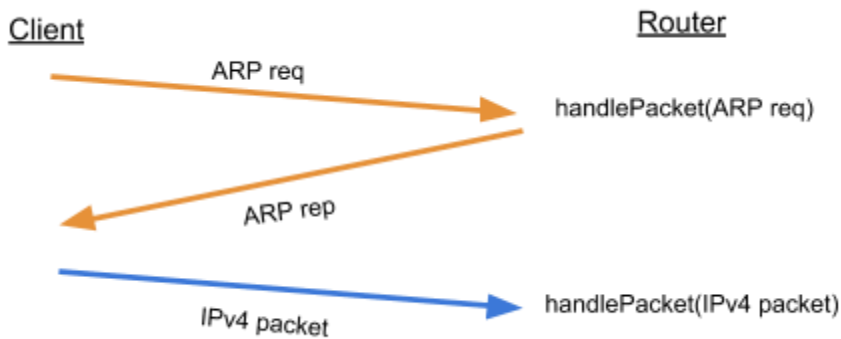
Here is my problem/question/request for clarification: The very next packet that shows up on my terminal is a packet that is recognized as being the IPV4 packet. I haven't implemented the 'ARP Reply' part of my code yet, so maybe that's why that isn't showing up right after sending the new packet from 'ARP Request'...? I want to know if I am doing it the right way, and when I implement 'ARP Reply' properly, will I get the correct sequence of packet flow automatically?

**TA Update:** I'm confused by what you mean by ARP reply part... when you get an IPV4 packet, it must be that the client has your router's MAC address or your router replied an ARP request. So I don't see how you got an IPv4 packet if you don't have ARP reply??? Can you clarify this point?

**Student Update:** Ok, so I implemented ARP Reply, too, and I send a cerr statement stating what kind of packet is being sent each time, and it goes directly from ARP Request to IP packet. I see no ARP Reply packet. I do not think I am doing this right... Or there is an error in the way I transmitted packets or something. Or I am not understanding this properly at all.

### TA Update 2:

The flow of the handlePacket calls is the following:



### [Ans] ARP Cache--Where and When

**Details:** According to the spec's pseudocode, it doesn't really tell us to cache the 'ARP Request' in the ARP cache but rather only in the 'ARP Reply', but I thought I remember seeing a video on YouTube that says you gotta cache the ARP Request as well as the Reply. Which way is right? (Sorry if it's a bad question. Super late when I asked this, and gotta finish this soon to do other work.)

**TA Answer:** Good question :) In Arp-cache, there is a queueRequest(). It will return an ARPRequest pointer if there was such a request that was made prior. However, do not confuse this with the datagram Arp Request. ARPRequest (the object) does not contain the packet you actually send (as in it doesn't contain an Arp Request message). It holds the queue of PendingPackets to be forwarded when an ARP Reply returns.

You technically "cache" both Request and Reply but that word is a bit overloaded. When processing the Reply, you store the IP-MAC info the reply reveals. This adds to the arp-cache thus why the YouTube video said you cache the reply. The "cache" of the Request is that when the router sends an ARP request, it remembers that it sent this request in the cache as well. As you can see, both add to the arp-cache but in different ways (in fact different tables/components of the same arp-cache). I guess this is why the video said it caches both but :/ I think that makes it more confusing...

## [Ans] Syntax for insertArpEntry

**Details:** I keep getting an error when I use insertArpEntry. "insertArpEntry was not declared in this scope"

**Student Answer:** Try looking at the private member variables of the SimpleRouter class! Look in the header/hpp file! :) If you look there you will find the answer to your question (variables for the arp cache and the routing table!!!).

---

## [Ans] How do we get the MAC address of the interface to send back in an ARP reply?

**Details:** I'm trying to get the router to build up an ARP reply. I don't know where to get the arp\_sha field. Also, the dhost is the router's IP address and shost is the requester's IP address right? Multiple questions in one post sorry. Here's them numbered:

1. Where do I get the arp\_sha field? (Given pretty much everything except the interface's MAC address)
2. The dhost is the router's IP address?
3. The shost is the requester's IP address right?

**Student Answer:** 1) theres a function called findIfaceByName() or something similar that can be used to get the MAC for the arp\_sha field. So the MAC address is the just pointer value of the interface returned by that function? The function returns a pointer to an interface class. I believe the member variables of that class are in the spec and one of them is the MAC. Ok so that makes Buffer addr the MAC address... So many data structures...

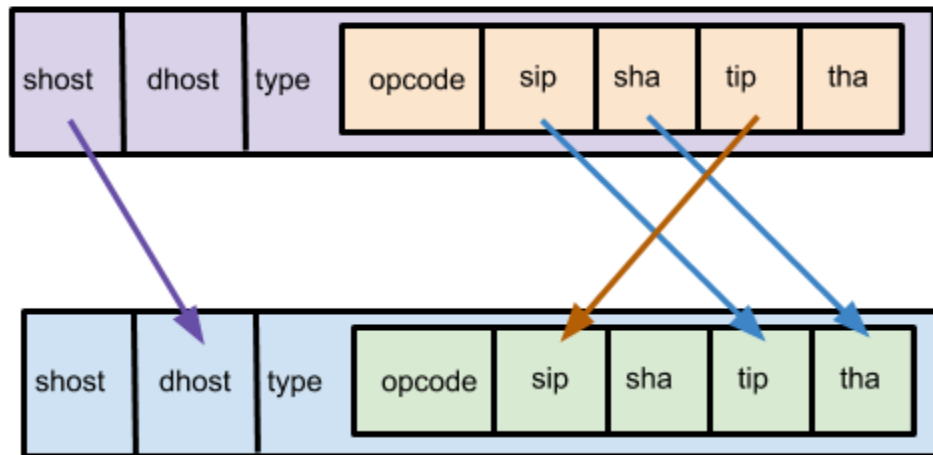
Still hoping for clarification on 2 and 3.

**TA Answer:**

1. There are multiple ways to get at it :) As the student answer above says there's a function called findIfaceByName() and a way to use that way ;) (can't reveal it). There's also another way of using findIfaceByIP(). Remember, you "may" have to check that the target ip of the ARP request is for the router on that face. These functions return a struct called Interface which contains addr (the MAC address field) as a buffer => addr.data().
2. I think for the field dhost you mean the field in the Ethernet header correct? (that says ether\_dhost) For the ARP reply, that is the ARP request's ethernet frame source host. It is the destination the ARP reply frame is intended for. Arp\_tip and arp\_sip are the fields in ARP that have ip addresses.
3. Ether\_shost is the router's MAC address of that interface.

(See drawing below for a picture clarification for 2 & 3)

### ARP request



### ARP reply

---

[Ans] Decrementing TTL and recalculating checksum when sending packets that you've just gotten an ARP reply for.

**Details:** This Buffer vector is SO CONFUSING to me. When the router gets an ARP reply, it wants to send out the packets to the MAC address it just learned. There's a list of pending packets and each pending packet has a Buffer that describes that packet. How do I get the TTL and checksum?

LMAO I answered my question as was typing it out. I realized that that packet (which is a Buffer) is just an IP packet, which has fields for TTL and header checksum. You can delete this is you want, but I kinda thought it would be helpful for people at a similar point.

**TA Answer:** Reinterpret\_cast. You can also move the logic of TTL and checksum before queueing the packet but it's up to you :) Reinterpret cast doesn't work on const though so a new packet has to be build if queueing packets up here.

---

## [Ans] About shared\_ptr

**Details:** I don't quite understand what shared\_ptrs work. If you look at functions in arp-cache.hpp, lookup(), it tells to "free the returned structure" if not NULL. Meanwhile, queueRequest() tells NOT to free the returned structure. I am confused on what this freeing means (since I couldn't simply delete the structure), and when should I /should I not free the structures. As of now, my router seems to work completely fine (it can send ping without loss, and can transfer the test file) without doing anything to the structure.

**TA answer:** As long as it works then it's fine :) Don't worry too much about it.

---

## [Ans] Router ARP Request

**Details:** If the router gets an IP packet and the MAC address isn't in the ARP Cache, then it needs to send an ARP request to get that MAC address. What is the ether\_host here? Where does that interface send the ARP Request? To the other interfaces?

**Student Answer:** I think that is what you do. Send an ARP request to the interfaces that you currently don't know the mac address of.

**TA Answer:** The ARP request is sent to the broadcast MAC address (FF:FF:FF:FF:FF:FF) since the router's interface only knows the IP address and does not know what MAC address to use to send out the ethernet frame.

---

## [Ans] findFacebyMac(packet)

**Details:** When I call this function on packet, it is returning nullptr even though I can see by printing the ethernet header that the MAC address in the packet is one of the interfaces. What am I doing wrong? For context, I'm doing this check when I evaluate the ethernet header to make sure the ether\_dhost is either broadcast or the interface whose name is inlface.

**Student Answer:** I believe findFacebyMac takes in the actual mac address (which is represented using a buffer) and returns the Interface which contains that mac address (an Interface consists of [1] interface name [2] mac address and [3] ip address).

```
const Interface* SimpleRouter::findlfaceByMac(const Buffer& mac) {}
```

What you might want to do here is convert the packet to appropriate structs through reinterpret\_cast or some other casting and using the size of the headers as an offset. Then you can access the mac address and use findFacebyMac .

**TA Answer:** The above answer is correct :) Also, you said in the context you provided that you are trying to check if the ether\_dhost is broadcast. If the MAC addr is broadcast, it will not match



any interface's MAC anyways, as the helper function you are using checks for exact MAC match and does not handle broadcast address as a special case.

---

[Ans] `cksum(packet.data(), packet.size())`

**Details:** What arguments should we be sending into `cksum`? The ones above don't work properly, but I can't imagine what else it would be. I've also tried `cksum(iphdr, iphdr->ip_len)` to no avail. Syntax nightmare.

**Student Answer:** You should pass the packet with the `ip_sum` set to 0 (which means you could either (1) store `ip_sum` somewhere, then overwrite it with 0 and pass to `checksum` or (2) make a copy of the ip header and put it somewhere, change one of the copy's `ip_sum` to 0 and pass that one to `checksum`. [Here is more information from a previous answer from TAs](#)

**TA Answer:** Student answer says the right thing :)

---

[Dup]: Timeout error while connecting. No idea what is going on. I have deleted everything and redid it but still same results.

```
vagrant@vagrant-ubuntu-trusty-64:~$ cd /vagrant
vagrant@vagrant-ubuntu-trusty-64:/vagrant$ ./router
-- 11/22/17 05:02:52.135 ./router: Network: trying to establish tcp connection to 127.0.0.1:8888
-- 11/22/17 05:02:52.136 ./router: Network: failed to establish tcp connection
    local address = 127.0.0.1:34436
    remote address = 127.0.0.1:8888
    Network.cpp:2169: Ice::ConnectionRefusedException:
    connection refused: Connection refused
```

**Student Answer/followup to above question:** Did you run `mininet` first in a `ssh` session, and then run the `router` second in a different `ssh` session? (so you would have 2 `ssh` sessions running in parallel). Also please follow the question format because it makes it easier for everyone to jump around from question to question.

**[update]** I am running two separate `ssh` however it still gives me the same error.

**TA suggestion:** Try following all the steps in this answer and see if it helps:

Can you try changing `vm.box` by changing the `vagrantfile` to

```
config.vm.box = "ubuntu/xenial64"
```

Also try [disabling windows firewall](#) as there is a chance that it might be blocking the connection.

**TA Update:** Marking this as a duplicate of: [\[Pending...\] ./router Connection Refused](#)

Please check this post for latest updates on fixes

---

## [Ans] Question about PING

**Details:** I'm not exactly sure how PING works. I think i have the project implemented correctly but some clarification needed. When i do 'client ping server1' i see 0% packet loss. However there are approx. 30 duplicate packets received. Is this a problem with my code? Also the command does not terminate until i hit ctrl-C. Is this normal? Related question: how do i know the file transfer command given in the spec works?

**Student partial Answer (other people can add on):** I believe that behavior from ping is correct. [ping](#) is used to test if you can reach a certain network, so it sends the duplicate packets on purpose so that the program can measure more clearly how fast they are transmitted, etc.

**TA Answer:** You should not have duplicate packets. If you have duplicate, that means you're making copies of the IP packet (and not forwarding the one received). But yes, you will need to stop the ping command with ctrl-C (ping keeps going until you tell it to stop)  
As for how can you tell if the file transfer worked, you'll need to take a look at your router ssh connection and verify your debug methods. A more "visible" output is 1.txt file being generated I believe... (personally I just verified my packets were generated and came in in the right order with the right field settings)

---

## [Ans] 95% packet loss... I'm at a loss

**Details:** So I implemented most/all of the code, to the point that I can see ARP replies, and IP packets sending, and even \*some\* packet transfers successfully. However, That is the problem... when I run "client ping server1", I get on average about 95% packet loss. About every 1 in 30 of my packets get through, but the others are lost. Idk what the problem could be. Anyone have any ideas? I thought it might be because I only send 1 packet

**Other Student follow up:** Did you resolve this? I have the same problem :(

**Other Student Update:** Jk I resolved it myself. Pretty sure no one is as dumb as I am, but I was using sendpacket() with the MAC address instead of updating the packet and using the packet as the argument.

**TA Guess:** I think it may be something with your ARP cache update? Or the creation of the ARP Request for server1? The first packet must've generated an ARP request and successfully sent 1 packet out but for the subsequent packets, it might've not been forwarded or the checksum was no longer correct such that the router or server drops the packet. ./

I'm thinking that why it's 1 every 30 is probably due to the ARP entry timing out. That means your ARP cache check isn't executing correctly for when the ARP entry DOES exist.

---

[Ans] IP address for Lookup

I seem to be getting the following values when I print my routing table.

```
00000000000000000000000000000000 00000000000000000000000000000000 0110010000000010000000000001010
0.0.0.0 0.0.0.0 0.0.0.0
00000010000000101010100011000000 00000000111111111111111111111111 00000010000000101010100011000000
192.168.2.2 192.168.2.2 192.168.2.2
00001010000000110100000010101100 00000000000000001111111111111111 00001010000000110100000010101100
172.64.3.10 172.64.3.10 172.64.3.10
```

Is there a reason that in binary format the addresses are from right to left.

For example 255.255.0.0 is 000000000000000000001111111111111111 instead of 1111111111111111110000000000000000. This would completely change my routing algorithm. I am guessing that since these are in network byte order ?

**Student Response:** Are you printing the routing table, or just calling the routing table function? If you call just the function, it should work fine. Maybe you are trying to print it?

**Response:** I am trying to print the table to see how the IP addresses are stored so that I can implement my lookup function.

**TA Answer:** Network byte order is Big Endian i.e. 192.168.2.2 should be stored in the same order of bytes - 192 followed by 168 followed by 2 followed by 2 (obviously all in binary :D). However, I am surprised to see that the order is Little Endian for both IP in the header and routing table entries. I shall try to find the reasoning behind this discrepancy.

As for performing bit by bit comparison, we need to start from MSB which in our case is at the rightmost position (LSB), hence reversing the order sounds like the correct thing to do in this case. I shall confirm and get back asap. If you just plan on ANDing the mask with the IP in the routing table and see if it matches with the destination IP, everything should still work fine as at this moment we can see that everything is in little endian format.

[Ans] When receive IPv4 packet, and if MAC address is not found in cache, after queueing the data, do we also need to make an ARP request from scratch and send it?

**Details:** The spec said “the router should queue the received packet and send ARP request to discover the IP-MAC mapping”, just to clarify, we do need to make an ARP request packet from scratch right?

**TA Answer:** Yes, you need to build packet (eth headers and ARP headers)

**Followup:** I had initially implemented it by building the packet from scratch but I found that the packet is also sent automatically after i add it to the queue. Are we supposed to send a duplicate?

**TA follow-up:** You said “packet is also sent automatically”, could you please explain which packet is sent automatically and where in the code? After I add the Arp request to the queue, the ARP request packet (from router to broadcast) is sent without me building an ARP request from scratch. I believe “Other Student” is right in which function does this. I was just wondering if we also have to build an arp request from scratch or should we let the periodic function handle it.

**Other Student:** It could be due to this function: [periodicCheckArpRequestsAndCacheEntries](#) Which is called by “[ticker](#)” once every second (I think) which is [bound to a thread](#) in the constructor of arp cache.

**Other Other Student:** so we just put it in the queue and we’re good?

**Other TA Answer (Just to summarize and hopefully answer this better):** When you queue the request, you are calling arp-cache’s queueRequest() function. This will output an ARP Request OBJECT (not the packet) which you will need to update to log that you have sent an ARP request for this IP-MAC mapping.

If you don’t do this, periodicCheckArpRequestsAndCacheEntries() may act strangely (like creating an ARP request packet when you just made one). I can’t tell you what to specifically update on the ARP Request Object because that’s part of reading code/comments but you will need to update it.

When you call queueRequest() it will not send an ARP request packet => you will need to create the ARP request packet with the correct ethernet header and arp request header. queueRequest() will add the packet to a list of PendingPackets for you to later process when you receive an ARP reply.

---

[Ans] Do we need to deal with when TTL is  $\leq 0$ ?

**Details:** above. And if so, when specifically do we need to do so?

**TA Answer:** TTL (Time to live) is used to eliminate packets that have been floating on the internet for a long time (i.e. if a packet is incorrectly being routed, it will indefinitely keep going places if not handled well. Many such packets can cause too much unwanted traffic. Thus, on every hop, each node in the internet reduces the TTL field by 1 - for this project, you need to check TTL right when you start looking at a packet and if it is  $\leq 0$ , you drop the packet.

---

[Ans] How do we know an ARP request is sent to the router as opposed to another device?

**Details:** Just getting started on the ARP request portion of the project. As I understand if the router receives an ARP request for its own address it'll return it, otherwise it'll forward its own ARP request to get the address of the "other" device. How do we get the address of the router? Maybe it's just late but I can't seem to find where to get this info. (aka how do we know what our own address is?)

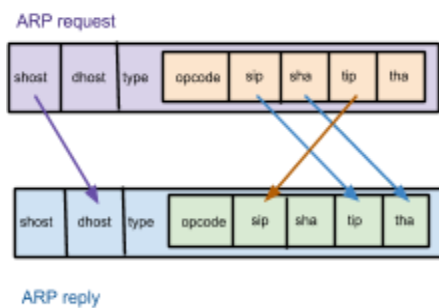
**TA Answer:** A node/router sends out ARP requests only if it wants to send packets to an IP but does not have the MAC for the next hop. It does not send ARP requests if it does not find answers for other node's ARP request; in fact a node does not even look at requests that do not have the node's IP address in the destination field.

Here is how addresses for **ARP request** look:

- Eth src : MAC address of the requestor (own MAC)
- Eth dest : Broadcast MAC (FF:FF:FF:FF:FF:FF)
- IP src : IP address of the requestor (own IP)
- IP dest : IP address of the node for which we want the MAC address

Here is how addresses for **ARP response** look:

- Eth src : MAC address requested
- Eth dest : MAC address of the requestor
- IP src : IP address of the node replying
- IP dest : IP address of the node requestor



[Ans] Lookup bits or bytes.

Do we have to do a bit level lookup or a byte level lookup.

I.e. do we have to find the longest prefix from the bits like 01100111

Or simply use ipToString and find it using 192.168.1.1 etc.

**TA Answer:** You have to search for longest match by bit level comparison.

---

[Ans] Should we handle ICMP headers?

**Details:** In `utils.cpp/print_hdrs`, the code handles ICMP headers separately. Should we also check if there's a ICMP header whenever we receive an IPv4 packet by checking `ip_hdr.ip_p`?

**TA Answer:** We previously thought of having ICMP as a part of this project, but now **you do not need to handle ICMP headers in this project.**

Just treat the ICMP packets as if they were IP packets :)

---

[Ans] How do we get all of the IP addresses of the router?

**Details:** In the pseudocode for `handlePacket()`, if `eth_type` is IPv4 packet, and "if packet is destined to the router (to one of the IP addresses of the router) - it should be discarded in this project". I'm confused how should we get all the IP addresses of the router? I know we can use `Interface.ip` to get the ip address for the interface from which our router received that packet, but what about the router's ip addresses on other two interfaces?

**TA Answer:** You can use `findInterfaceByName()` and pass the name of each interface in the topology to get the interface structs, then get the ip field in this struct.

---

[Ans] How exactly do we use vagrant?

**Details:** The instructions on how to use vagrant in the spec are really vague and I'm not sure what I'm supposed to be doing. Do I install vagrant onto my computer or onto the VirtualBox? Vagrant just seems to get stuck when I use vagrant up. What is the vagrant init setting that has to be used?

**Student Answer:** You install virtual box on your computer, and then you also install vagrant on your computer. Then you do vagrant up, etc. in the directory your project files are located in Which Operating System are you using?

- I know for windows 7 you have to make sure to [install the most recent version of powershell](#) for vagrant up to work (if you get stuck installing this update you might have to go to "services.msc" and stop the process "Windows Update"...then installing the fupdate should work)
- Other people can add tips for other operating systems here if they encountered problems and fixed them

**TA Answer: I have already received few emails asking the same question. Here is the answer to the FAQ:**

**Steps to setup vagrant and run/test your router code:**

1. Download VirtualBox, default install
2. Download Vagrant, default install. Reboot.
3. Open cmd in Windows, git clone to a folder.
4. Change config.vm.box line to config.vm.box = "ubuntu/xenial64" in the vagrantfile present in the repo. (Don't need vagrant init I suppose)
5. *vagrant up* (if you are asked to enter a password, you'll need to set in your vagrant file username = vagrant, password = vagrant and go ahead accordingly).
6. *vagrant ssh* (do this in two separate cmd prompts)
7. *ls* to check if you see a folder named "vagrant". If not, check in two directories above this one (cd ../../vagrant)
8. Run the mininet program in one cmd using the command *sudo python ./run.py*
9. Compile and run the router in the other cmd prompt using the commands *make* and *./router*
10. If you have completed your router related code as required by the project, you can now test your code.

The way we plan to test your code is by checking if the client can send a test file (in our case: test.file in the repo) to the server. This operation of sending a file will be managed by mininet at high level and all that you need to worry about is if your router recognizes the ARP and IP packets being sent from one node in the test network to another node and responds accordingly.

- The command "server1 /vagrant/server 5678 /vagrant/ &" is telling the server1 node to listen on port 5678
- The command "client /vagrant/client 192.168.2.2 5678 /vagrant/test.file &" is telling the client node to send the test.file to the server node at [192.168.2.2:5678](#)

Thus, if the code is running fine, after the commands in step 2, you should be seeing a file named "1.file" in your repo.

---

## [Ans] Routing table destination vs gateway?

**Details:** The destination and gateway items seem to be the same. Is it correct that, when using longest prefix match algorithm, we should compare the ip packet destination with the "destination" (and, of course, mask) column in the routing table, and use the corresponding "gateway" when deciding what's the next hop? For example, if the ip-packet's destination is different from the target's subnet's gateway, when the router check the ARP cache, do we look

for the MAC address of the gateway/next hop or the final destination? Since the spec says “MAC address mapped to the destination IP address for next-hop” which is kind of confusing.

**TA Answer:** We look for the MAC address of the gateway/next hop **and not the** final destination. Reason: destination MAC is of no use here, MAC is used to identify the immediate one hop neighbor to send to while sending out a frame on link-layer.

**Other TA Answer:** In this project, the gateway and destination are the same. In real world routing tables, they are not. Destination could be a completely different IP Subnet but the only way we can get to that subnet is via the gateway router indicated. Such entries that fit this format are those updated by internal BGP (one BGP router can reach a specific part of the world while the other cannot)

---

### [Ans] Calculate longest prefix

**Details:** Do we have to calculate the length of the longest matching prefix, or is it sufficient to compare the value of masks? Since to me longer prefix match is equivalent to a larger value of mask...

**TA Answer:** You need to mask both the routing table entry and the destination IP address by the mask of the routing table entry and compare if they both match. You compare all such matching entries on the basis of the length of their masks. The entry that matches and has the longest mask length is the best next hop to take as it takes us closest to the target network in a single hop from the current node.

---

### [Ans] *make* fails

**Details:** When I run make, I get the error:

```
mkdir -p build
```

```
slice2cpp -I/usr/share/lce/slice --output-dir=build --header-ext=hpp core/pox.ice
```

```
make: slice2cpp: Command not found
```

```
Makefile:14: recipe for target 'build/pox.cpp' failed
```

```
make: *** [build/pox.cpp] Error 127
```

I haven't finished implementing the project. Do I need to finish before I can see how the testing works?

**TA answer:** This error is likely because you have run the make command without ssh into the VM (you need to install Vagrant, VirtualBox, run vagrant up, run ssh vagrant, go to vagrant folder in the VM after ssh and then finally try to make). Let me know if that is not the solution that works for you.



Student note: I already performed those steps, I'm entering the command for:  
ubuntu@ubuntu-xenial:~\$

**TA followup:** Resolved after student cloned the repo fresh. Was probably because the student had an old copy of the repo before Seungbae added the "core" folder to the repo.

Student note: Might also have been due to the fact I didn't clone into ~/cs118-proj1

---

## [Ans] Queueing and sending ARP request

**Details:** queueRequest returns a pointer to a list of PendingPacket. Do I send ARP request for all packets in the list? Also do I have to check timeSent or nTimesSent before sending request?

**TA answer:** We send out ARP request when we do not know the MAC for the next hop. Hence, in the meanwhile, we have to hold on to the IP packet for which we use the queueRequest() function. This function returns the pointer to a queue of PendingPackets that are waiting on an ARP request to be answered as you correctly said. Now **we send out an ARP request for every packet with unknown MAC for next hop that we encounter, but we also update the timeSent field to current time** (I used std::chrono::steady\_clock::now(); to get current time) so that the function periodicCheckArpRequestsAndCacheEntries() has a track of the time for when an ARP request was sent out last for the queue of Packets waiting on that ARP request.

**Student Follow-up:** The comment for packet inside PendingPacket struct says it is "a raw Ethernet frame, presumably with the dest MAC empty" so I have to send ARP request for all packet in the list after all?

**TA answer:** The router will receive packets one at a time. Now the packets received can as well be all be for the same destination. If the router has found out the appropriate IP address to forward the packet, it now tries to add ethernet header over the IP packet to send it to the next hop. This is when the router looks at the IP to MAC mapping and discovers that it does not know the MAC for the next hop. It now sends out ARP requests. **Thus, to answer your question, yes you have to send out ARP request for every such packet you receive in the handlePacket() function.**

However, this is not the only point of time when ARP requests are sent out. Let us assume a scenario where we get only one packet which is pending and for which we have sent out the ARP request once when we received this packet. Now what if our one and only ARP request was lost/unanswered for some reason? Solution: keep sending ARP requests after reasonable fixed interval. periodicCheckArpRequestsAndCacheEntries() has code that already looks at the timeSent field and sends out new ARP requests if it has been long enough. You don't need to worry about that part. All you need to worry about is sending ARP requests when you get a

packet in handlePacket function (provided the next hop Mac is unknown) and updating the timeSent field for the queue of PendingPackets that are waiting on the same ARP request to be answered.

---

[Ans] How to get the name of each interface in the topology?

**Details:** When checking if an IP packet is destined to the router, we need to compare the packet's destination with the ip addresses of the router on each interface, which is obtained by using findInterfaceByName(name of each interface in the topology). In one previous post, the TA says "do not hardcode the IP addresses or name of each interface." So how should we get the name of each interface in the topology? I was thinking about iterating each entry in the routing table, but the RoutingTable class doesn't provide a public method allowing us to do so. Should we read from the file IP\_CONFIG or RTABLE?

**Student Answer:** findInterfaceByIp() might be useful

---

[Ans] insertArpEntry() returns NULL

**Details:** I did queueRequest() when forwarding with no arp entry found but when handling ARP reply, insertArpEntry() return NULL ArpRequest pointer.

**TA Answer:** insertArpEntry() adds an Arp Entry to m\_cacheEntries, then looks up if this entry is an answer to an already pending request. If it is found, returns a pointer to the ArpRequest. Otherwise, returns nullptr. If you get a NULL pointer, it means that there are no pending Arp requests for the entry you just entered in the cache. You need to worry only if in your step you are sure you should have gotten a list of pending entries for the Arp reply you added to the cache.

---

[Ans] After queueing the packets, how are we accessing the queued packets to be forwarded?

**Details:** I may just be missing something, but is there a function that points use to the queued packets for a certain ip address?

**TA Answer:** You would want to queue pending packets for an IP if the MAC is unknown. You would only want to access this queue once you have an answer (ARP response) for the ARP

requests you sent out for the queue. When you receive an ARP response, you also make a call to insertArpEntry(). This is the function that will help you.

insertArpEntry() adds an Arp Entry to m\_cacheEntries, then looks up if this entry is an answer to an already pending request. If it is found, returns a pointer to the ArpRequest. Otherwise, returns nullptr. If you get a NULL pointer, it means that there are no pending Arp requests for the entry you just entered in the cache.

---

[Ans] Can we add methods to utils.hpp & utils.cpp?

**Details:** Can we add debug functions (like printing stuff) to utils? In general, are we allowed to modify / add code for debug purpose?

**TA Answer:** Yes, you can. No points will be deducted :)

---

[Ans] Mininet not working

**Details:**

I am getting this error when i try running ./run.py

\*\*\* Adding links:

Traceback (most recent call last):

File "./run.py", line 97, in <module>

main()

File "./run.py", line 79, in main

net = mininet.net.Mininet(topo=topo, controller=mininet.node.RemoteController,  
ipBase=IPBASE )

File "/usr/lib/python2.7/dist-packages/mininet/net.py", line 172, in \_\_init\_\_

self.build()

File "/usr/lib/python2.7/dist-packages/mininet/net.py", line 442, in build

self.buildFromTopo( self.topo )

File "/usr/lib/python2.7/dist-packages/mininet/net.py", line 429, in buildFromTopo

self.addLink( \*\*params )

File "/usr/lib/python2.7/dist-packages/mininet/net.py", line 364, in addLink

link = cls( node1, node2, \*\*options )

File "/usr/lib/python2.7/dist-packages/mininet/link.py", line 424, in \_\_init\_\_

node1, node2, deleteIntfs=False )

File "/usr/lib/python2.7/dist-packages/mininet/link.py", line 468, in makeIntfPair

deleteIntfs=deleteIntfs )

File "/usr/lib/python2.7/dist-packages/mininet/util.py", line 194, in makeIntfPair

( intf1, intf2, cmdOutput ) )

Exception: Error creating interface pair (client-eth0,sw0-eth3): RTNETLINK answers: File exists

The last thing i did was to try the test commands in the specs.

**TA Answer:** `sudo python run.py`

**Response:** I have tried with sudo and I still get the same error.

**Student Answer:** I had problems with the same thing. Last night I was kicked off the connection so I assume mininet didn't close properly (deleted the file that was supposed to be deleted for creating that interface pair). I was looking online and running `sudo mn -c` would clean up the initialized files that were supposed to be deleted at exit. So I ran `sudo mn -c` and ran `sudo ./run.py` and it worked for me.

---

[Ans] Why does my arp response test not work?

**Details:** I run the ping like the note says to do, but when I do `client arp 10.0.1.1`, the hw address is incomplete. I only use `insertArpEntry` if the opcode is 2, since that would be a reply. However, using print statements, I discovered that my program never makes it into the `opcode == 2` body. In other words, the router never gets a packet with `opcode = 2`. Is this a problem with the way I'm running the test, or a problem with my code? I know my IP and Mac addresses are correct since I print them out, but my code never gets into the body of the (`opcode == 2`) statement. Any help is much appreciated!

**TA follow up:** What '`client arp 10.0.1.1`' does is that it sends an ARP request to your router from the already present client in the mininet topology. Thus, it is obvious that you are getting an ARP request at the router (and not an ARP reply). Your router code is the one who will build and send out the ARP reply for this ARP request. Hence, your code never goes into the `opcode==2` condition.

Also, make sure you use `ntohs()` to convert the opcode from network byte order to host byte order before comparing.

**Student follow up:** Right, but the spec says we should only update the arp cache when we receive an ARP reply. If we never receive the ARP reply, why should `client arp 10.0.1.1` return a filled in ARP table?

From the spec:

"Your implementation should not save IP-MAC mapping based on any other messages, only from ARP replies!"

Wait, I have definitely been using the wrong interface for my ARP reply. It should be the interface that we received on, correct? Gonna answer my own question here, yes its that interface.

**TA Answer:** client arp 10.0.1.1 returns the CLIENT's Arp table (not the router's). So if it did receive a response from your router and it was correct, the client table will be non-empty :)

---

[Ans] How do we use reinterpret\_cast to obtain the header structures?

I'm having trouble extracting headers from the packet Buffer. Since Buffer is just a vector<unsigned char>, I create a subvector out of each section in Buffer that I believe should be a header, then call reinterpret\_cast to cast vector into struct. For example: ethernet\_hdr:

```
ethernet_hdr e_hdr = reinterpret_cast<ethernet_hdr>(&e_vector);  
//e_vector is a subvector containing the first 14 elements of packet.
```

I keep getting:

Error: invalid cast type from vector<unsigned char> to simple\_router:ethernet\_hdr.

What am I doing wrong?

**TA Answer:** You could simply use memcpy to copy contents from the buffer to the appropriate header. Use sizeof(header\_struct) to get the length of data to be copied.

**Student Answer:** This could be wrong, but what worked for me is: Instead of &e\_vector, you should do e\_vector.data(). Similarly, change the ethernet\_hdr to ethernet\_hdr\*.

---

[Ans] Send timeout

**Details:** "client /vagrant/client 192.168.2.2 5678 /vagrant/test.file" outputs:

"error: send timeout

send: connection refused"

What could be wrong here? ("client ping server1/2" does send packets with 0% loss for me)

**TA Update:** Probably that port is already taken???? (Not too sure how but that's the only thing I can think of). Try closing your mininet and router ssh connections and retry without running ping. (Ping uses a different port than the file transfer)

**Student Update1:** I tried the suggestion but still got the same error.

>> ./ you ran the line before that one? (the server 5678)

**Student Update2:** Actually I didn't when I test the finished version of my code. Silly me!  
Thanks!

---

[Ans] “High level design of your implementation“

**Details:** What is expected in the README file for “high level design of your implementation”? Since we basically write the code based on the given pseudocode, can we just rewrite what the spec says? Also how long should this part be?

**TA Answer:** The README is meant to emphasize any problems you encountered and show the logic/approach you took. I guess yours could be close to the pseudocode but it shouldn't copy it exactly. Also recall that the pseudocode did not include info on how you approached longest prefix match.

---

[Upd -- Awaiting student update] Forwarding and checksum results are weird

**Details:** I ran the client ping server1 test from the start (no cache entries) and got the exact results that were expected. sender/receiver IP addresses and Mac addresses were all correct. However, when I should have gotten a packet from the server to eth1, I instead got a packet from the client to eth3. I also report 100% lost packets. Does this mean my code is incorrect? If it is, do you have any idea what part might be wrong? I've been debugging for hours but it's hard to debug when the results are all correct.

**TA Guess:** Ping keeps sending ping packets all the time (it doesn't wait for the 1st ping packet to succeed to send). Since yours says 100% lost packets and you're sure that a eth1 to server1 packet was sent, I think most likely your checksum is incorrectly calculated. Servers will only respond the ping if the checksum is right.

**Student Response:** Interesting. The checksum is found by using the cksum function on the header with ip\_sum set to zero, correct? Because that is what I'm doing.

>> cksum is on just the IP header with the ip\_sum set to 0 first. You then put the result into the ip\_sum field if you're forwarding the packet down.

---

[Upd--Awaiting Student Response] Getting all 0s for hardware address of server?

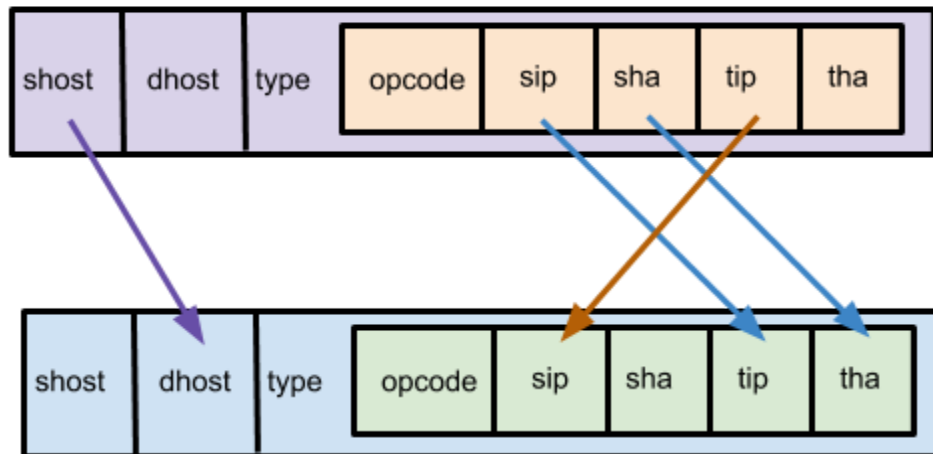
**Details:** I tested my ARP request handling code as suggested by the TA “A good debugging test for ARP response”. All fields except the hardware address work as shown. The response I get is this:

```
mininet> client arp 10.0.1.1
Address      Hwtype  Hwaddress      Flags Mask      Iface
10.0.1.1     ether   00:00:00:00:00:00  C               client-eth0
mininet>
```

Is this by design?

For the .arp\_sha and .arp\_tha in the response arp header, I used the values .arp\_tha and arp\_sha from the received arp packet respectively. Basically this:

### ARP request



### ARP reply

What am I doing wrong?

**Student answer:** The arp\_tha that you received is all zeroes, since the arp request doesn't know the mac address. Try printing out headers anywhere in your code and you'll see this

**TA Answer:** based on the above... it seems your router responded to the client with a broadcast address...

[Ans] vagrant can't find run.py

**Details:** Entering the command "sudo ./run.py" after cd'ing into /vagrant gives the error "usr/bin/env: 'python\r' No such file or directory. Making worked just fine.

**Student answer:** This is probably because vagrant is being run on a Windows machine. run dos2unix run.py before running sudo ./run.py.

---

[Ans] What exactly is a 'mask'

**Details:** This is probably covered somewhere, but I have no idea what the mask is in the routing table, and what it has to do with the longest matching prefix.

**Student:** & the entry's mask field with its dest field when finding the longest matching prefix. The mask defines the subnet in which the nodes sit.

**TA Answer:** The mask determines how many bytes of the IP address you have been assigned make up your subnet's address

---

[Ans] Test file range

**Details:** My code worked fine with the test.file but when I tried with large files (~7MB), it didn't work (got 1KB output files). Will my code be tested/graded on transferring large files?

**TA Answer:** We'll be testing with something close to the test.file. (don't hardcode based on that it is test.file)

---

[Ans] Mininet Error

**Details:** Ok so my client ping server1 is working in that the packet gets from client to server1. But then when the server sends an IP packet back, it throws an error and mininet prints the following: "From 10.0.1.100 icmp\_seq=10 Destination Host Unreachable" Thoughts?

**TA Answer:** Please do not use clear() for the PendingPackets queue. Instead, delete the ARP request object (with removeRequest method in arp-cache) when done popping the PendingPackets~

---

[Ans] When to use which interface when sending packets.?

**Details:** As of right now, I'm sending back every packet on the same interface I receive it from. I've checked my other code, and I think it's all correct, but I'm still getting 100% packet loss (I'm definitely receiving the packets, just unable to respond to them). Can someone tell me which interface we're supposed to send out packets to. There are 3 places in handlePacket() where I



think we send out packets: ARP request (arp\_op == 0001), IP packets if we find a MAC translation in our ARP cache, and n times in ARP Reply (arp\_op ==0002)  
For the first two cases, I send back on the same interface it came from (the iface variable in the skeleton code), and in the third case, I use the item in list<PendingPackets> packets's iface variable.  
I could be wrong about this. Please advise.

**TA Answer:** It should be that when you get an ARP Request, you send a reply back the way it came. When you send an ARP Request or forward an IP packet, you send it down to the next hop.

---

[Ans] How to check the file you transfer

**Details:** How do you check the contents of the file you send using mininet? Where do the output files go?

**TA Answer:** If the file transfer succeeded, you should get a #.txt (e.g. 1.txt from the first run and so forth). You can open it up with a normal text editor but it's a boring file. :)

---

[Ans] 100% packet loss but right flow of addresses

**Details:** Right now, when testing using client ping server1, I get the right flow for the addresses following the forwarding test. However, I get 100% packet loss and the packets just keep getting sent again. What am I doing wrong?

**Student Answer:** If you don't see messages coming back from server1 at the router, it means that the packets aren't getting to it/it is dropping it. When I had this issue it was bc I wasn't recomputing the checksum after changing the TTL when forwarding.

---

[Ans] Compiler warnings in arp\_cache.cpp

**Details:** Do we have to worry about the warnings in this file about unused variables? I'm scared to delete them because I don't know if it'll cause everything to explode.

**TA Answer:** Do ignore the unused variable warnings. :)

---

## [Ans] Included in tar.gz

**Details:** Are there supposed to be a bunch of git objects out of a git/ directory included in the tarball? Is the ipch/ directory supposed to be included as well? I just want to be sure cause it would be a shame to lose credit for this.

**TA Answer:** if the directories were already existing from the git repo (like the .git folder) then they are safe. If they are extra directories that you made (excluding core if you made the core folder yourself), they aren't allowed in the folder.

**Another student:** How about the .vagrant folder?

**TA Update:** I believe the Makefile excludes the .vagrant folder when tarball is called so that should not be included in the tar.gz

---

## [Ans] Running Natively on Ubuntu

**Details:** How do we run the emulation process without the use of VM? I use Ubuntu has my standard OS, but every time I try to run ./run.py I get an error for not having mininet. Solutions above say we need a vagrant folder for this, but my understanding is that vagrant is for VM...

**TA Answer:** We will never be running your code natively. We are running your code in our working VM :) If you want to run the code natively, there are many packages you will need to download

---

## [Ans] ARP packet sanity check

**Details:** When the router receives an ARP packet, should we check hardware\_type == 1, protocol\_type == 0x0800, HW\_addr\_len == 6, and Prot addr len == 4? And if not, discard the packet?

**TA Answer:** Primarily check that the packet is for the router at that interface. Drop otherwise. You can check the other fields if you like but it's not required

---

## [Ans] Submission files

**Details:** There's one previous post where TA/student says we should only include files that exist in the original repository (no temp or newly added files). So after I run make clean and make tarball, I compare the files I have in the .tar.gz file with the original git repo, and the only

difference is that my tarball includes one additional file called "ubuntu-xenial-16.04-cloudimg-console.log". It is not excluded by Makefile, so should we delete it manually, or it's fine to leave it there?

**TA Answer:** It's fine to leave it there. It's not generated by you (but the VM xenial) and the main point is that your code is not referring to any "illegal" files (e.g. output text files or C++ classes)

---

[Ans] ERROR: 'Invalid port number, try another'

**Details:** I get this error after trying to copy the file multiple times when I do the server1 /vagrant/server 5678 thing. , although the file still copies. Does that mean my code is working or not?

**TA Answer:** you're making the server listen to the same port multiple times => the server makes the port busy/taken. Though you're using the same port for the same application (the file transfer), the server thinks you're trying to grab the port for a different application.

---

[Ans] Lookup routing table endianness

**Details:** When I look up the next hop in my routing table, let's say the next hop has IP 192.168.2.2, the Routing Table returns it in reverse order, i.e. 2.2.168.192. Calling ntohs or htons on it changes the address to 0.0.192.168. How can I reverse the IP address so I don't get a core dump when I call findInterfaceByIp with the result of the routing table lookup?

**Self-answer:** Psych, I'm an idiot all of the ip addresses are like this and the Routing Table Entry has an interface name member field.

---

[Ans] IP Packet ARP Request no ethernet dest

**Details:** When we send an ARP Request after receiving an IP packet, the spec says we need to queue our request in the arp\_cache. Since we don't have the destination MAC at this point, should we queue an incomplete packet, and deal with this in our ARP Reply handling? The spec is a little unclear on this.

**Follow Up:** Nvm, the packet will be resent and forwarded successfully after the ARP reply is cached so we don't need to mess with the ethernet header. Duh.

---

## [Ans] Sending Packets After Receiving ARP Reply

**Details:** After we receive the ARP reply, is the code to send out packets already implemented, or do we have to implement it ourselves?

**TA Answer:** You'll need to implement it yourselves. Have fun ;)

---