1. Check the DNS resolver’s configuration file /etc/resolv.conf. Change the “nameserver” to be “10.0.0.1”.

2. nslookup and dig: Try the following commands. Read the utilities’ manuals to get more information.
   a. Get the IP address of the given domain name
      \texttt{nslookup google.com}
   b. Specify a server type (NS=name server). It returns the local DNS servers in MIT.
      \texttt{nslookup -type=NS mit.edu}
   c. Query a specific DNS server (8.8.8.8)
      \texttt{nslookup umb.edu 8.8.8.8}
   d. Again, get the IP address (go through all the sections of the output)
      \texttt{dig umb.edu}
   e. Specify server type (MX=mail exchanges)
      \texttt{dig umb.edu MX +short}
   f. Specify the DNS server
      \texttt{dig umb.edu @8.8.8.8}
   g. Inverse query
      \texttt{dig -x xxx.xxx.xxx.xxx +short}
   h. No recursive option: Try to dig some domain names that haven’t been cached yet.
      \texttt{dig +norecurs domain}
      It should return no “answer section”.
   i. Check TTL in the answer section. Google (e.g., gmail.com) usually sets it to 300 seconds. (Do you see 300 there? If not, why?)
      \texttt{dig gmail.com}
   j. Test DNS cache and negative cache. Try to \texttt{dig} some probably-existing but rarely-visited domain names. Check the query time in the output. Then repeat the same command and check the query time again. (Can you see the effect of DNS caching?) Similarly, try to \texttt{dig} some probably-non-existing domain names twice and compare the query time.
3. Use **ifconfig** (short for **interface configuration**) to get basic network information
   a. Open a terminal and type in **ifconfig**, all the available network interfaces will be listed. Normally, the Ethernet card is named as eth0. If multiple Ethernet cards are installed, you will see eth0, eth1, ...
      You can list the information of a particular network interface by specifying the interface’s name as the argument of **ifconfig**
      
      ```
      $ ifconfig eth0
      ```
   
   b. Find out the IP address and MAC address of the VM.

4. Use **wireshark** to catch packets
   a. Run **wireshark** as root (2 options)
      i. You can open a terminal and type in
         ```
         $ sudo wireshark
         ```
      ii. Or you can create a launcher as follows:
         1. Right click on the desktop, select “Create Launcher…”
         2. Type in “Wireshark” in the field of “Name”
         3. Type in “gksudo wireshark” in the field of “Command”
         4. Click “OK”
   
   b. Start monitoring on eth0
   c. Switch to the host machine
   d. Open a terminal and ping the VM with option `-n`
      ```
      $ ping xxx.xxx.xxx.xxx
      ```
   e. Switch to the VM and stop monitoring

5. Select an ICMP Echo Request message sent by the host machine, and expand the Internet Protocol part of the packet in the packet details window. Identify the each field in the IP header, especially ‘total length’, ‘protocol’, ‘source address’, and ‘destination address’
6. Understand the packet size: check the following numbers and explain how they are related to each other.
   a. In the first line of the packet details window, “74 bytes on wire”
   b. In the IP header, “Total Length: 60”
   c. Expand Internet Control Message Protocol, “Data (32 bytes)”

7. Start wireshark monitoring and switch to the host machine. Send large ping packets by using -l option
   
   `$ping xxx.xxx.xxx.xxx -l 4000$

8. Switch back to the VM and stop wireshark. Check the Echo request messages:
   a. Are they fragmented? What information in the IP header indicates that the datagram been fragmented?
   b. How many fragments were created? How to determine the first fragment and the last fragment?
   c. How the specified 4000 bytes are distributed into the fragments?

9. Start wireshark monitoring, open a terminal and run
   
   `ping google.com -c 5`

10. Stop wireshark.

11. Get routing information: Use the following two commands to get the kernel routing table and ARP mapping.
    a. `route -n`
    b. `arp -n`
    What are the next hop’s IP and MAC addresses? Check the packets captured by wireshark, are there ARP query/response? Look at the outgoing echo request packet, what’s the destination’s MAC address? Understand the roles of the MAC and IP addresses.