NSHG: A Network Simulation Hacking Game

# Analysis

## 1.1, 1.2 Introduction (What is networking and why is it important)

The fundamentals of communication and networking is a massively important topic in the computer science A-level, with the growing connectivity of the world through the internet of things it is growing ever important that any computer scientists worth their weight will need a strong understanding of what these concepts are and how they ate implemented in the real world.

The topic in the A-Level is split up into 4 topics which are each split into further sub topics, these main and sub topics are as follows.

1. Communication
   1. Communication methods
      1. Serial/parallel
      2. Synchronous/Asynchronous
      3. Start and Stop bits
   2. Communication basics
      1. Baud rate, Bit rate, Bandwidth, Latency, Protocol
      2. Differentiate baud rate and bit rate
      3. Understand the relationship between bit rate, baud rate and bandwidth.
2. Networking
   1. Network Topology
      1. Difference between and how star and bus topologies work
   2. Types of networking between hosts
      1. Explain when peer to peer may be used
      2. Explain when client-server may be used
   3. Wireless networking
      1. Explain the purpose of Wi-Fi
      2. Be familiar with the components required for wireless networking
      3. Be familiar with how wireless networks are secured
      4. Explain the wireless protocol Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) with and without Request to Send/ Clear to Send (RTS/CTS).
      5. Be familiar with the purpose of Service Set Identifiers (SSID)
3. The Internet
   1. The Internet and how it works
      1. Understand the structure of the internet
      2. Understand the role of packet switching and routers
      3. Know the main components of a packet
      4. Define Router and gateway and consider where they are used
      5. Explain how routing is achieved across the internet
      6. Describe the term uniform resource locator in the context of internetworking
      7. Explain the terms fully qualified domain name (FQDM), Domain name and IP address
      8. Describe how domain names are organised
      9. Understand the purpose and function of the domain name service and its reliance on the domain name server (DNS) system
      10. Explain the service provided by internet registries and why they are needed
   2. Internet Security
      1. Understand how a firewall works (packet filtering, proxy server, stateful inspection)
      2. Explain symmetric and asymmetric encryption and key exchange
      3. Explain how digital certificates and digital signatures are obtained and used
      4. Discuss worms, Trojans, and viruses , and the vulnerability’s they exploit
      5. Discuss how improved code quality, monitoring, and projection can be used to address worms, Trojans and viruses.
4. The TCP/IP protocol stack
   1. TCP/IP
      1. Describe the role of the four layers of the TCP/IP stack (application, transport, network and link)
      2. Describe the role of sockets in the TCP/IP stack
      3. Be familiar with the role of MAC addresses
      4. Explain what well known ports and client ports are used for and the differences between them
   2. Standard application layer protocols
      1. Be familiar with the following protocols
         1. FTP
         2. HTTP
         3. HTTPS
         4. POP3
         5. SMTP
         6. SSH
      2. Be familiar with FTP client software and an FTP server, with regard to transferring files using anonymous and non-anonymous access
      3. Be familiar with how SSH is used for remote management
      4. Know how an SSH client is used to make a TCP connection to a remote port for the purpose of sending commands to this port using application level protocols such as GET for HTTP, SMTP commands for sending email and POP3 for retrieving email
      5. Be familiar with using SSH to log in securely to a remote computer and execute commands
      6. Explain the role of an email server in retrieving and sending email
      7. Explain the role of a web server in serving up web pages in text form
      8. Understand the role of a web browser in retrieving web pages and web page resources and rendering these accordingly
   3. IP address structure
      1. Know that an IP address is split into a network identifier part and a host identifier part
   4. Subnet masking
      1. Know that networks can be divided into subnets and know how a subnet mask is used to identify the network identifier part of the IP address
   5. IP standards
      1. Know that there are currently two standards of IP address, v4 and v6.
      2. Know why v6 was introduced
   6. Public and private IP addresses
      1. Distinguish between routable and non-routable IP addresses.
   7. DHCP
      1. Understand the purpose and function of the DHCP system
   8. Network Address Translation (NAT)
      1. Explain the basic concept of NAT and why it is used
   9. Port Forwarding
      1. Explain the basic concept of port forwarding and why it is used
   10. Client server model
       1. Be familiar with the client server model
       2. Be familiar with the Web socket protocol and know why it is used and where it is used.
       3. Be familiar with the principles of Web CRUD Applications and REST:
          1. CRUD is an acronym for:
             1. C – Create
             2. R – Retrieve
             3. U – Update
             4. D – Delete.
          2. REST enables CRUD to be mapped to database functions (SQL) as follows:
             1. GET → SELECT
             2. POST → INSERT
             3. DELETE → DELETE
             4. PUT → UPDATE.
       4. Compare JSON (Java script object notation) with XML.
   11. Thin vs Thick client computing
       1. Compare and contrast thin-client computing with thick-client computing.

This is a vast specification and in order to answer the questions fully the student must understand all of these points and be able to apply them and evaluate aspects of them. For example discuss how the TCP/IP stack works, what information is needed my each layer and what information is passed between them.

To begin my project I am going to interview my end users, whom are teachers and students, to see some of the problems they have with teaching networking concepts, engaging students with the material and the understanding of the abstract concepts covered in networking and the internet.

The interview will be semi-structured having a list of required questions, written before the interview, and asking more questions based of the end users answers. This will allow for more in-depth information to be gathered reducing the risk of a possible re-design when showing the end user the “alpha” version of the project.

During the interview I will be taking notes

In the interviews I will discuss the problems and then possible solutions to those problems. I will focus on managing the expectations of the end user as to not have them expect more than I can produce in the given time.

## Current systems

The subject of networking is usually taught in 4 weeks with 5 hours a week. The lessons 1 hour long with 1 block of 3 lessons and 1 block of 2 lessons each week.

The class size currently is 7 people thought this varies from year to year (with a maximum of 15)

### What’s the problem?

## Interview Summary

After having an interview with the teacher he has given me a list of lessons/concepts that are covered in networking and the internet when teaching the subject:

* The OSI Model
* Encryption
* Routing
* Packet Switching
* TCP/IP protocol stack
* DNS
* Gateways
* Fire Walls
* Proxy Severs
* Smtp. Imap, pop, ftp

## The Current system

The subject of networking is usually taught in 4 weeks with 5 hours a week. The lessons 1 hour long with 1 block of 3 lessons and 1 block of 2 lessons each week.

The class size currently is 7 people thought this varies from year to year (with a maximum of 15)

The lessons consist of a power point with several task sheets to provoke questions from the students and assess their ability while going through the lesson. These tasks are fast and easy to complete.

At the end of the subject there is a 2 hour subject test which allows the teacher to make sure that the teaching has been successful and that the students have taken in the information this can then be processed to find what parts of the subject were misunderstood and what was received well which is used for revision/ recaps for a re-test if it is deemed necessary by the teacher.

## 1.4 Success Criteria

|  |  |
| --- | --- |
| Success Criteria | Operationalised |
| The solution must teach all of the specification points fully. | Have a check list of all teaching/information points that need to be taught & check they are all covered. Further check that they someone who hasn’t completed the specification before can use the program, with the test in front of them, and answer all of the questions fully. |
| Allow the students to revise different areas in the specification |  |
| The solution must be engaging to students to make them want to learn about networking. | Test with the students to make sure that they enjoy the solution and engage with it. Found through observation. |
| The solution must have an “End of unit test” in order for the teacher to be sure all the students fully understand the topic | Have the end of unit test implemented as well as the one in the system to compare results to make sure the system is working, this ensures reliability of the new test. |
| The end of unit test would be self-marked | Have a teacher and the system mark the test to make sure it is marked correctly (interrater reliability) |
| Results should be output as a file making it easy to analyse | Have the results of each of the students be output as a CSV value (or another file type that can be imported into google docs/excel) |

## Limitations of my system

With this system being used in a school environment there are many technical limitations to my solution. Firstly the school network has a large firewall that will blocks lots of network traffic to make sure the school network says secure. This means that any data will have to be hosted within the school system so it can be accessed while in school. However this has the problem that the data would only be available while in school. A possible (though unlikely) solution to this is allowing an exception through the school firewall to allow the data to be accessed from outside the system.

Another limitation is that I will not be able to store information about students as otherwise I wouldn’t be able to be able to ensure its security (as I am not qualified). My solution to this is not storing any personal information about the student meaning it cannot be stolen from me. This does mean that I can’t do any data analysis on the students and instead have that handled by the current data handling system, which has been recently updated.

*With the time given it will be difficult to implement all specification points fully so I will most likely have the result of this project be a polished first quarter of the networking & the internet specification*

## Possible Solutions

I brainstormed some ideas that I thought could solve the problems my end user was having from there I will show those ideas to the teacher and the students and see what idea they will find most engaging and will be most representative of what my end user had in mind

### 1 Revision Game

My first idea was to create a revision game where students would answer questions and be told if their answers were right or wrong. When the student got a question wrong it would be added to a Challenge bank to be used later. The challenge bank would allow students to continue with questions then revisit what they found hard to see if they had learned what they got wrong after a period of time, this is to make sure it goes into their long term memory instead of short term memory.

When a student gets a question correct they are given points

## Chosen solution (and why)

“I have the attention span of a fish so the ‘hacking’ game would keep me occupied and engaged for longer”

Solution 1 doesn’t fully address the solution

## Research

### The OSI model

The Open Systems Interconnection model (OSI model) is a way to standardise the communication between systems it is broken down into 7 layers;

1. The physical layer
2. Data Link
3. Network
4. Transport
5. Session
6. Presentation
7. Application

I will go over each of the layers in detail now

#### Layer 1: Physical

This defines the physical, electrical connections between devices e.g electrical cable or optical fibre. This includes: voltage levels, timing of voltage changes, physical data rates, maximum transmission distances, and physical connectors.

#### Layer 2: Data Link

This layer describes the connections between nodes and doesn’t deal with the physical layers. This layer has basic error correction that will occur on the physical layer. This layer doesn’t deal with connecting multiple nodes together (networking) it is only concerned with the connection between two nodes and terminating it.

This layer can sometimes be divided into two sub layers

##### The Medium access control (Mac) Layer

This is the sub layer responsible for how devices gain access to data and permission to transfer

##### The Logic Link Control (LLC) layer

This layer controls the encapsulation of protocols and error correction

#### Layer 3: Network

#### Layer 4: Transport

#### Layer 5: Session

#### Layer 6: Presentation

#### Layer 7: Application

### Headers

How does data travel across the internet?

I purchased the book The Networking Bible in which I found a reference to something called an RFC so I decided to investigate, the rfc it pointed to was rfc 791 which is the official standard for how the IP Header works and how IP addresses are structured

#### IP

#### TCP

#### UDP

### DHCP

Now I have the Headers sorted I need to think about what happens when a system first connects to a network. The first thing a system does when it connects to a network is obtain an IP address and other information through DHCP This means I will have to implement a version of the DHCP protocol which uses UDP. The basic premise behind DHCP is 4 packets, 1 a broadcast from a DHCP client onto their local network to discover a DHCP server if there is one, 2 A DHCP server Responds multicast with an offer to lease an IP address, 3 the client responds (unicast) with a request to the DHCP server requesting the offered IP Address, 4 The DHCP server responds with a DHCPAck packet, which contains confirmation of the IP address, the subnet mask, Default gateway and server[[1]](#footnote-1) [[2]](#footnote-2) [[3]](#footnote-3)

I will use the RFC1 to form the layout of my packets however with the scope of this project and to reduce the processing overhead I will have to simplify it and it will not be fully featured, only the barebones will be there

0 1 2 3

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

| op (1) | htype (1) | hlen (1) | hops (1) |

+---------------+---------------+---------------+---------------+

| xid (4) |

+-------------------------------+-------------------------------+

| secs (2) | flags (2) |

+-------------------------------+-------------------------------+

| ciaddr (4) |

+---------------------------------------------------------------+

| yiaddr (4) |

+---------------------------------------------------------------+

| siaddr (4) |

+---------------------------------------------------------------+

| giaddr (4) |

+---------------------------------------------------------------+

| |

| chaddr (16) |

| |

| |

+---------------------------------------------------------------+

| |

| sname (64) |

+---------------------------------------------------------------+

| |

| file (128) |

+---------------------------------------------------------------+

| |

| options (variable) |

+---------------------------------------------------------------+

The DCP Packet will be structured as seen above.

FIELD OCTETS DESCRIPTION

----- ------ -----------

op 1 Message op code / message type.

1 = BOOTREQUEST, 2 = BOOTREPLY

htype 1 Hardware address type, see ARP section in "Assigned

Numbers" RFC; e.g., '1' = 10mb ethernet.

hlen 1 Hardware address length (e.g. '6' for 10mb

ethernet).

hops 1 Client sets to zero, optionally used by relay agents

when booting via a relay agent.

xid 4 Transaction ID, a random number chosen by the

client, used by the client and server to associate

messages and responses between a client and a

server.

secs 2 Filled in by client, seconds elapsed since client

began address acquisition or renewal process.

flags 2 Flags (see figure 2).

ciaddr 4 Client IP address; only filled in if client is in

BOUND, RENEW or REBINDING state and can respond

to ARP requests.

yiaddr 4 'your' (client) IP address.

siaddr 4 IP address of next server to use in bootstrap;

returned in DHCPOFFER, DHCPACK by server.

giaddr 4 Relay agent IP address, used in booting via a

relay agent.

chaddr 16 Client hardware address.

sname 64 Optional server host name, null terminated string.

file 128 Boot file name, null terminated string; "generic"

name or null in DHCPDISCOVER, fully qualified

directory-path name in DHCPOFFER.

options var Optional parameters field. See the options

documents for a list of defined options.

#### DHCP State Transition Diagram

-------- -------

| | +-------------------------->| |<-------------------+

| INIT- | | +-------------------->| INIT | |

| REBOOT |DHCPNAK/ +---------->| |<---+ |

| |Restart| | ------- | |

-------- | DHCPNAK/ | | |

| Discard offer | -/Send DHCPDISCOVER |

-/Send DHCPREQUEST | | |

| | | DHCPACK v | |

----------- | (not accept.)/ ----------- | |

| | | Send DHCPDECLINE | | |

| REBOOTING | | | | SELECTING |<----+ |

| | | / | | |DHCPOFFER/ |

----------- | / ----------- | |Collect |

| | / | | | replies |

DHCPACK/ | / +----------------+ +-------+ |

Record lease, set| | v Select offer/ |

timers T1, T2 ------------ send DHCPREQUEST | |

| +----->| | DHCPNAK, Lease expired/ |

| | | REQUESTING | Halt network |

DHCPOFFER/ | | | |

Discard ------------ | |

| | | | ----------- |

| +--------+ DHCPACK/ | | |

| Record lease, set -----| REBINDING | |

| timers T1, T2 / | | |

| | DHCPACK/ ----------- |

| v Record lease, set ^ |

+----------------> ------- /timers T1,T2 | |

+----->| |<---+ | |

| | BOUND |<---+ | |

DHCPOFFER, DHCPACK, | | | T2 expires/ DHCPNAK/

DHCPNAK/Discard ------- | Broadcast Halt network

| | | | DHCPREQUEST |

+-------+ | DHCPACK/ | |

T1 expires/ Record lease, set | |

Send DHCPREQUEST timers T1, T2 | |

to leasing server | | |

| ---------- | |

| | |------------+ |

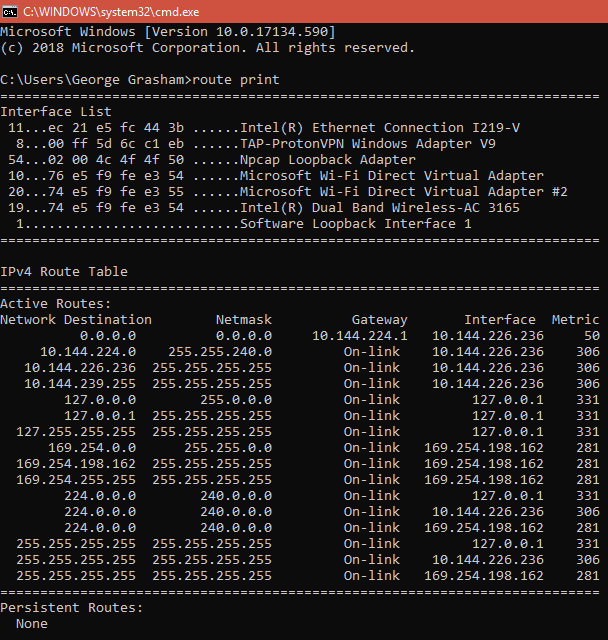
+->| RENEWING | |

| |----------------------------+

----------

### Routing Tables

I will now have to look at how a system decides what adapter to send a packet to for it to forward on. After a brief google search I found that Routing tables are responsible for looking at a packets IP header and deciding what physical connection to send it down. You can access your own routing table in windows by using the “route print” command, I ran this command to see what I could understand from the beginning and what I needed to research more about. The result of my route print command is included below



From here I can see a few familiar things, 1 at the top there is an interface list. This seems to be a list of network adapters that the system has. For me I have a few that stand out, The Inter(R) Ethernet Connection I219-V which is my Lan Ethernet adapter so If I plugged an Ethernet cable into my laptop that adapter would be used, The Intel(R) Dual Band Wireless-AC 3165 which is my current adapter as I am connected to my current network wirelessly. There is also the software loopback which is the loopback adapter for my system, for example if I wanted to ping myself I would use the loopback adapter so I don’t need to be connected to a network.

Next the IPv4 Route Table. This is the routeing table for my system, this confuses me a little but here are my first ideas about each of the columns before research. The destination is the destination address that a packet would have to be addressed to be forwarded to a specific adapter, the Net mask also play a role in this to as the net mask would be applied in the packet that needs to be routed to see if it fits the destination address once it is also had the net mask applied to it. For example for my system if I sent a packet with the address 127.6.1.1 it would be accepted by the 127.0.0.0 entry as if you apply the mask 255.0.0.0 to 127.6.1.1 it yields 127.0.0.0 meaning that is the route it would take. The first entry in these addresses has the 0.0.0.0 address and the 0.0.0.0 mask meaning that any address would be mapped to this entry successfully. This leads me to believe when a system is using the table it doesn’t use a top down approach and instead likely has something to do with the last column “Metric”. The next column is the Gateway. I think this is the address that the packet needs to be forwarded to in order to reach its destination. As of right now I am unsure about on-link but I would like to note that each entry with on-link is a local address. I believe that Interface is the specific adapter that the packet needs to be routed down to reach its destination. The reason it uses the IP address of the interface instead of the mac address is unclear to me but could have something to do with setting the source address of the packet. I will now research to see what I was correct about

From my research I know that what I thought was basically correct, when sending a packet, the routing table is used to know what interface to use. The system will look up the table in a decrementing fashion looking at the highest metric first and getting smaller, the destination and netmask are used to see if the sending destination address is matched by the rule, if the address is a match then the packet is sent on the appropriate interface. The Gateway specifies that the destination isn’t on the local network and needs to be forwarded to that gateway for NAT to take place and then be forwarded from there onto the larger network.[[4]](#footnote-4),[[5]](#footnote-5)

My next question was how Microsoft populates this table

### Routing Protocols

After looking at many routing protocols and how they are used I found that classless routing algorithms are very complex (Especially for what I need them for) but all follow a similar protocol. A hello message sent regularly to make sure that the router is till up, when a router stops responding the routing tables of the surrounding routers are changed. My implementation will follow a similar premise with the hello packets to discover the network and then when a router goes down a message will be sent saying that it is down, updating the network maps of the routers.

The route the router will forward the packet down is found using Dijkstra’s shortest path algorithm to create a forwarding table.

# Design

1. <https://tools.ietf.org/html/rfc2131> [↑](#footnote-ref-1)
2. <https://www.youtube.com/watch?v=S43CFcpOZSI> [↑](#footnote-ref-2)
3. <https://en.wikipedia.org/wiki/Dynamic_Host_Configuration_Protocol#Operation> [↑](#footnote-ref-3)
4. http://techgenix.com/making-sense-windows-routing-tables/ [↑](#footnote-ref-4)
5. http://library.mobrien.com/Manuals/MPRM\_Group/images/rightframe\_files/routing.htm [↑](#footnote-ref-5)