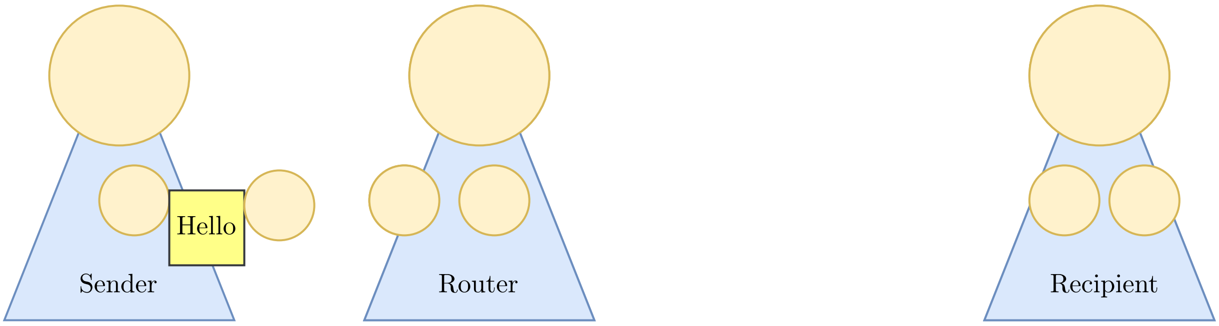
Simulating The Internet*

**With post-it notes and masking tape*

s1532620



The goal of this lesson is to give pupils hands on experience with designing the basic protocols behind data transmission and reliable communication over the internet. The pupils will be asked to send data to each other over a human driven computer network, this data will start small at simple text messages and will move on to sending pictures.

Demographic: This class is aimed at CfE Third Level pupils (S1 Computing Class)

Participants: 5 is recommended but the activity can be easily altered to accommodate for less or more pupils.

Learning Outcomes

This lesson covers the following Curriculum for Excellence learning outcomes:

- **TCH 3-13a** (Recognises and describes information systems with communicating processes which occur in the world around me)
- **TCH 3-14b** (Demonstrate an understanding of how computers communicate and share information over networks including the concepts of sender,receiver, address and packets)

Lesson Preparation (15-20 minutes)

What You Need

- Post-it notes or any similar sized pieces of paper, you will need quite a lot.
- At least 4 printed out pictures of some cute animals, no larger than half a page of A4. (Perhaps in anticipation of the lesson, ask pupils to print out a picture of an animal they like or a pet)
- Scissors, blu-tac or glue, and masking tape.

Setup and Explanation of Roles

Set up desks in the layout described in figure 1. Each node will need its own desk and use masking tape to form the lines between the desks. Make sure there are visible labels on each desk saying which vegetable it is. There are 2 roles in the lesson, Users (named after vegetables) and the Router or "Root-er" if you prefer. In terms of tools, the users each need post-it notes, pencil and a picture. The router doesn't need anything. The users job is to send packets, they will send messages and responses to the other users in the network. The "root-ers" job is to forward the packets. They will receive packets from each of the users and need to forward them to the correct recipient. They will use labels on the desks to see where each packet needs to go.

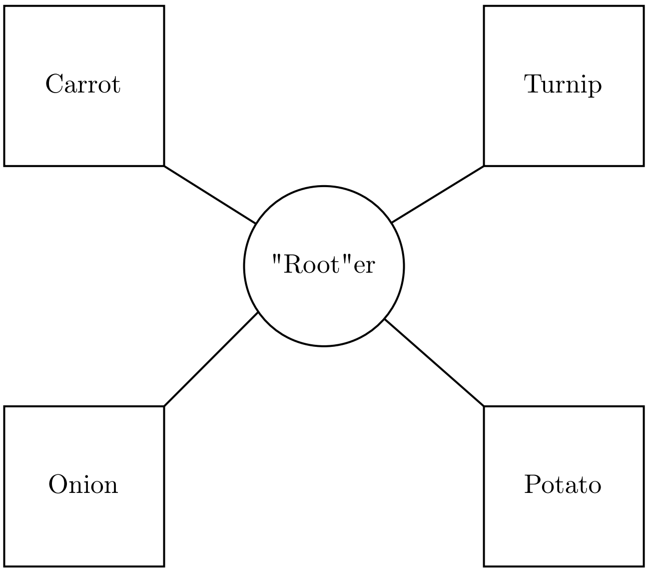


Figure 1: Network Layout for Activity

The Activity (30-40 minutes)

Rules

Before conducting the lesson explain the following rules:

1. **You are not allowed to talk to each other or use hand gestures, only use the packets to communicate.**
2. **Every piece of information you communicate must fit on the packet(post-it note).**
3. **You can only follow the lines on the floor.**
4. **You can only transport one packet at a time**
5. **You must only use the computers names and not your classmates real names**

Now on to the main chunk of the lesson:

1. Start by telling each of the users to send a message to someone else on the network, who sends what to who doesn't matter at this stage.
2. The pupils should quickly realise that, since they cant talk or point, they can't tell the router where to send the message. Turn to the class and ask if anyone has any ideas on how to solve this problem. If they need prompting relate our network to the postal service. What do you need to send a letter to someone?
3. After the packets have been sent you will encounter another problem, now that the message has successfully been delivered, how does the person respond? Turn to the class to ask for solutions. If the pupils are having trouble, prompt them by talking about sending letters (sender address and return address). The solution is to have an additional label that tells the recipient the "return address". The part of the packet that contains this basic information is called the "header".
4. The next part of the activity is moving on to transporting larger data. Give one of the users a new task: ask "Carrot" for a picture of one of the animals you printed. The pupils need to work out a way to send these larger things but still keep the size of the packets the same. Turn to the class and ask for ideas. There are several solutions, involving cutting or folding. The main point is, it has to fit on the packet alongside the header.
5. Once the picture has been sent, you need to work out how do you put it back together, especially since the packets will probably not arrive in the order they should have in. The images should be complex enough that the students cant just put them together. Prompt them by talking about "ordering" the packets. The solution is to label the packets with numbers that tell you the order in which the packets should be assembled, this will help the recipients put them all together.
6. The last thing is to put it all together, get the rest of the users to ask someone else for the picture, and make sure they don't all ask the same person. With all the necessary protocols in place the users should be able to send the request to the right person, get the image back and correctly reassemble the image, revealing a cute animal.

After the lesson is done you can talk about the theory behind the lesson, which is detailed in the next section.

Further Discussion

The intent of this section is to explain the concepts behind the lesson. This can be used as simply a primer for the educator or as the basis for a follow up lesson that takes more of a technical approach to explaining the concepts.

Packets

In the lesson we use packets as the main unit of data transmission. This is exactly how it works in the real world. Data units have different names depending on which part of the networking stack we are concerned with, but when we are talking about the Internet Protocol (IP) it is called a packet. If learners are having difficulty, the packet can easily be related to food packets as they basically serve the same purpose, they are containers for the thing you're actually interested in.

Protocols

What we have done throughout this lesson is build a very basic version of TCP/IP. IP is the internet protocol, it's the message format that defines how (almost) all messages are sent over the internet. An example of what the header of a real IP packet looks like is shown below:

<table>
<tr><th>Offsets</th><th>Octet</th><th colspan="8">0</th><th colspan="8">1</th><th colspan="8">2</th><th colspan="8">3</th></tr>
<tr><th>Octet</th><th>Bit</th><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th><th>10</th><th>11</th><th>12</th><th>13</th><th>14</th><th>15</th><th>16</th><th>17</th><th>18</th><th>19</th><th>20</th><th>21</th><th>22</th><th>23</th><th>24</th><th>25</th><th>26</th><th>27</th><th>28</th><th>29</th><th>30</th><th>31</th></tr>
<tr><th>0</th><th>0</th><td colspan="4">Version</td><td colspan="4">IHL</td><td colspan="6">DSCP</td><td colspan="2">ECN</td><td colspan="16">Total Length</td></tr>
<tr><th>4</th><th>32</th><td colspan="16">Identification</td><td colspan="3">Flags</td><td colspan="13">Fragment Offset</td></tr>
<tr><th>8</th><th>64</th><td colspan="8">Time To Live</td><td colspan="8">Protocol</td><td colspan="16">Header Checksum</td></tr>
<tr><th>12</th><th>96</th><td colspan="32">Source IP Address</td></tr>
<tr><th>16</th><th>128</th><td colspan="32">Destination IP Address</td></tr>
<tr><th>20</th><th>160</th><td colspan="32" rowspan="4">Options (if IHL > 5)</td></tr>
<tr><th>24</th><th>192</th></tr>
<tr><th>28</th><th>224</th></tr>
<tr><th>32</th><th>256</th></tr>
</table>

Figure 2: An IP Header. Source: https://en.wikipedia.org/wiki/IPv4

The important parts are highlighted. The IP packet has a "source" and "destination" address just like the packets we designed. The lesson will help users understand the fundamental idea of strict formats that define communication, all network protocols are built on this premise. Keep in mindS that this part is just the header, the actual data would come after this header in the data stream.

The Transmission Control Protocol (TCP) is the other fundamental part of the internet. Its used in conjunction with IP so ensure that communications are reliable and are able to get to where they need to be. Below is what the header of a real TCP packet looks like:

<table>
<tr><th>Offsets</th><th>Octet</th><th colspan="8">0</th><th colspan="8">1</th><th colspan="8">2</th><th colspan="8">3</th></tr>
<tr><th>Octet</th><th>Bit</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><th>0</th><th>0</th><td colspan="16">Source port</td><td colspan="16">Destination port</td></tr>
<tr><th>4</th><th>32</th><td colspan="32">Sequence number</td></tr>
<tr><th>8</th><th>64</th><td colspan="32">Acknowledgment number (if ACK set)</td></tr>
<tr><th>12</th><th>96</th><td colspan="4">Data offset</td><td colspan="3">Reserved
0 0 0</td><td>N
S</td><td>C
W
R</td><td>E
C
E</td><td>U
R
G</td><td>A
C
K</td><td>P
S
H</td><td>R
S
T</td><td>S
Y
N</td><td>F
I
N</td><td colspan="16">Window Size</td></tr>
<tr><th>16</th><th>128</th><td colspan="16">Checksum</td><td colspan="16">Urgent pointer (if URG set)</td></tr>
<tr><th>20
...</th><th>160
...</th><td colspan="32">Options (if data offset > 5. Padded at the end with "0" bytes if necessary.)
...</td></tr>
</table>

Figure 3: A TCP Header. Source: https://en.wikipedia.org/wiki/Transmission_Control_Protocol

The important part this time is the sequence number, it tells you much much data you've already gotten and therefore, where this new data should be located in the sequence. We have achieved a similar effect with our protocol, labelling each packet with a gradually increasing number (1,2,3,4. . . ..) shows us the order the packets should be in so we can rearrange them if if necessary to get the intended result.

Network Congestion

Network congestion is not explicitly covered by the lesson but will be demonstrated in some capacity. Use this section to help answer potential student questions on the subject. The main point is that they will experience the congestion first hand when they start to deal with larger volumes of data.

Networks, like roads, get congested. When there is lots of traffic over a network the data flow can grind to a halt, increasing latency and lowering internet speeds. In this lesson: the pupil acting as the router needs to physically deliver each packet one at a time, this means that the users will have to wait until the router is ready to deal with their requests. This somewhat emulates how it really works. Real routers can only deal with one packet at a time, they're just really good at sharing their time so it looks like its happening simultaneously. This only starts to become a problem when large volumes of packets are being received by a system, but our small scale example still emulates this.