

FINAL PROJECT

CORY COOK

CONTENTS

1. Introduction	1
2. Method	1
2.1. Requirements	2
2.2. Abstraction	2
2.3. Design: Addressing Scheme	2
2.4. Design: Performance Testing	3
2.5. Node Device Specifications	3
2.6. Implementation Description	4
3. Implementation	4
4. Conclusion	6

1. INTRODUCTION

The internet is a tool that is used by a third of the world's population. It allows us to connect and exchange information, provide services, and exchange ideas. The only problem is that in order to connect all of these devices and servers we need to lay out miles of wire and provide hubs that connect all of these wires together. Since most of us don't want to lay out the wires needed nor do we want to be responsible for managing the routers that connect all of these devices we need to have companies do this for us. Companies are happy to do this for us, for a modest fee. This turns out to be a bad deal for the customer as we are usually very limited to which internet service providers are available in the area, and, as such, those internet service providers have a lot of control over pricing and the quality of the services offered. Wouldn't it be better if we could do without the Internet Service Providers (ISPs) all together? In order to do this, however, we would need to be able to rid ourselves of our dependence on wires, routers, and other externally managed network focal points.

The goal of my final project is to provide a (very) basic implementation of a node-based networking system: including an addressing scheme, latency test, and speed test. Once I have a working implementation I will determine if the design is feasible by comparing it to modern internet tests and speeds offered by ISPs.

2. METHOD

I'm going to lay out the requirements of the design and then I'm going to abstract the requirements to better focus my solution to a general case, rather than a complete system. I'm going to ignore environmental and economic requirements

and problems that would need to be considered as part of the design, and I'm going to make false assumptions regarding the ethical use of the design by its users. Including such things would exceed the scope of the project and complicate the design beyond the problem I'm trying to solve.

2.1. Requirements.

- (1) The system must allow connections to be established between any two devices on the network.
- (2) The system must allow data to be transferred from one device to another.
- (3) The system must not rely on any managed nodes or complicated systems that would require repair or maintenance.
- (4) The system must be extensible such that connections can be made over several different network types.
- (5) The system must be secure and data-safe.
- (6) The system must be able to forward data without receiving the entirety of the data (packets)

2.2. Abstraction. For the purposes of this project I'm only going to consider the requirements necessary to establish connections and transfer data (1, 2, and 3). I'm also going to make several assumptions about the system that would not be true in a physical setting:

- (1) Assume all nodes are active. Nodes will not deactivate throughout the duration of any transfers.
- (2) Assume all nodes are stationary. There will be no mobile nodes.
- (3) Assume that there is no data loss during transfer.
- (4) Assume that all nodes are technically equivalent. All nodes will have the same properties (range, power, transfer rate, response time, etc.)
- (5) Assume that the devices can make an unlimited number of concurrent connections.

Given these abstractions I can create a relatively simple and expandable model that should be easy to implement.

2.3. Design: Addressing Scheme. The addressing scheme will have to be one such that a device will be able to contact another device on the network. In a network that doesn't have a clearly defined hierarchy, such as the one we are trying to implement, using a standard IP addressing scheme would not give the device enough information to know where to send its data. I'm sure that there are many solutions available to resolving addresses in a node-based network such as this. A couple of design solutions for addressing that I was able to come up with would work but each has its inefficiencies.

The first solution is to use a Global Positioning Satellite (GPS) Service to resolve addresses based on physical location. This would require the use of an external service that the device would have no control over. This GPS system would need to be actively maintained and serviced, which means that someone has to get paid. However, GPS services are currently being provided by the government so we are paying for it whether we like it or not. This does have the added benefit of being able to use the GPS coordinates to send data directly. Implementation of this would be relatively simple. Each device would poll a GPS system for its physical

coordinates and build its address based on that. When the device needs to forward data it will forward the data to the device in the direction of the target system.

The second solution is to create a hierarchy by combining devices that are near each other. So a set of devices would represent a network group which would be contained within a larger group of groups, and so on until all devices are encompassed. Each device in the group would have addressing information for all devices within that group. This would be more difficult to implement as addressing would not be based on any kind of technology that already exists, and each device would need to have a lot more information and do a lot more calculation to resolve addressing for all requests. However, it does have the added benefit of not having to rely on any external system that would require regular maintenance. Implementation of this scheme would be more difficult than the first, especially for optimization. Addressing would be based on levels, each level would contain a set number of devices or groups within it and assign an address for that group at that level. This could result in some very long addresses. When the device needs to forward data it will forward it outwards to all devices or groups until there is a match at that level.

For the purposes of this project I am going to implement the first solution, as it is much simpler. I am going to assume that the GPS system is a free service that is already in place. Each device on the network will be able to resolve its own address. I'm going to assume that the GPS has a high enough accuracy to give each device its own unique address. In a real-world situation there would be devices that fall within the same GPS coordinates and the system would have to account for that by adding the ability to select a specific device within an address or by expanding the address itself to fit a device-specific element. Either way it involves added complexity in device selection that, while being very closely related to this project, can be abstracted to get to (what I believe to be) the true goal of this project. That is to compare theoretical performance of this system to the high-maintenance systems currently being used. So I'm going to abstract this by assuming that each coordinate will have, at most, a single device. I'm going to abstract this further by not using true GPS coordinates and instead define the area of my test and randomly place objects within that area. Using true GPS would create very large addresses with very little meaning in the actual testing.

2.4. Design: Performance Testing. The performance of these systems can be determined through analysis of the specifications of the nodes, the relative distance between the client and the host, and the distance between each of the nodes. In order to analyze the performance between the design and currently used systems we must first be able to determine the best method for transferring data in the system that we attempting to design. Once I have my emulated devices I am going to simulate data transfer as a function of the specification, analyze my data set to find the worst case scenario for my test area, and use the result to compare with current internet speed standards.

2.5. Node Device Specifications.

Data Transfer Rate: Gigabit

Maximum Range: 40 meters

Signal Degredation Rate: $\sqrt{-\frac{x-40}{40}}$

The Data Transfer Rate is the rate of data transfer within the optimum range. The maximum range is the maximum distance in which a connection can be established.

These specifications were selected to ensure that the entire area of the next address over would be within optimum range for an addressing range of 8m regardless of the current position of the object within its own address range.

2.6. Implementation Description. Each node is going to be an object with an address and a physical position. If the current address is (a,b) and the target address is (c,d) then the next address down the line will be determined by the following function (one-dimension):

$$(1) \quad a = c? a : a + \frac{c - a}{|c - a|}$$

We need to determine the physical distance between the nodes such that we can determine the signal strength and throughput. The throughput of the system is determined by the slowest data transfer on the line so if the bandwidth of the current node is less than that of the system throughput then set the system throughput as the current. Add any delay or latency incurred at the current node. Repeat until we reach the destination.

This should give us a latency and a throughput for a connection which is all we need to determine its performance. I will repeat this method for many random connections and record the data. Once I have the data I will determine a function that fits the given data and use optimization technique to find the worst case scenario for the system. Once this is determined I will compare the performance to the performance of the average internet users' connection and determine (roughly) if having a massive mesh networking system would be feasible.

3. IMPLEMENTATION

```

1  grid=1000 // Sets the size of the test area
2  ps=8 // Sets the plot size (size per unit area)
3  datarate=1000
4  A = rand(grid,grid,2)
5  A = A*ps
6  // This is the signal degradation based on distance
7  function b=sig(t)
8      b=sqrt(-(t-40)/40)
9  endfunction
10 // Caluclates the distance between nodes
11 function d=dist(curr,next)
12     d=sqrt((curr(1)-next(1))^2+(curr(2)-next(2))^2)
13 endfunction
14 // Gets the coordinates of the node
15 function coords=cor(x)
16     coords=[x(1)+A(x(1),x(2),1)/ps x(2)+A(x(1),x(2),2)/ps]
17 endfunction
18 // Gets the next node address
19 function out=np(u, v)
20     if u==v then
21         out=u
22     else
23         out=u+((v-u)/abs(v-u))
24     end
25 endfunction
26 function [latency, throughput]=set_val(start_pos,end_pos,speed)
27     latency=0

```

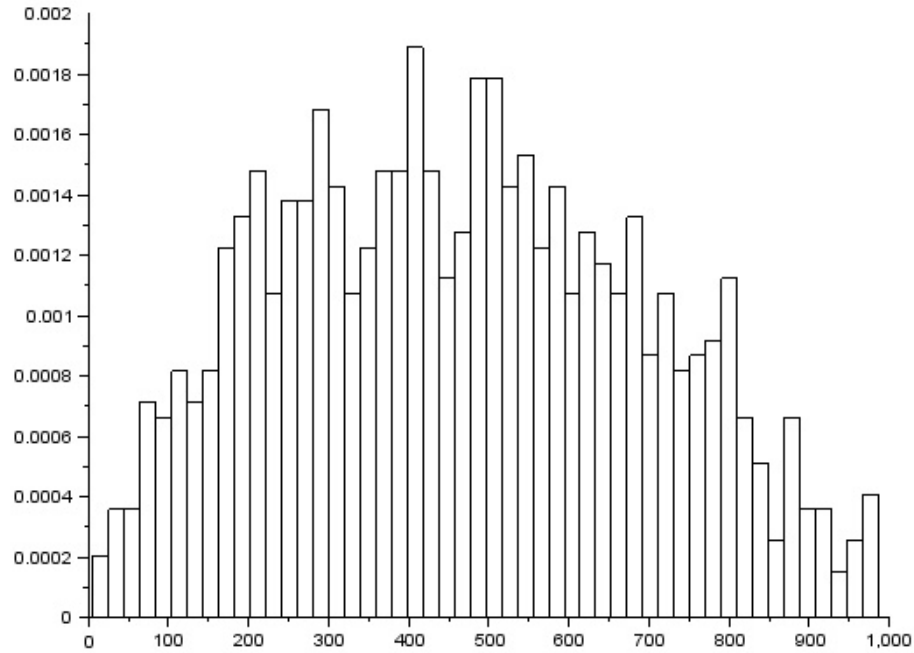


FIGURE 1. Latency Histogram Distribution vs. Time (ms)

```

28     throughput=speed
29     current_pos = start_pos
30     while or(current_pos~=end_pos)
31         next_pos(1,1)=np(current_pos(1), end_pos(1))
32         next_pos(1,2)=np(current_pos(2), end_pos(2))
33         latency=latency+1
34         val=sig(dist(cor(current_pos), cor(next_pos)))*datarate
35         if val<throughput then
36             throughput=val
37         end
38         current_pos = next_pos
39     end
40 endfunction
41 for i=1:1000
42     start_pos=[ceil(grid*rand()) ceil(grid*rand())]
43     end_pos=[ceil(grid*rand()) ceil(grid*rand())]
44     startp(i,:)=start_pos
45     endp(i,:)=end_pos
46     [lat(i), band(i)] = set_val(start_pos,end_pos,datarate)
47 end

```

LISTING 1. Mesh Network Implementation

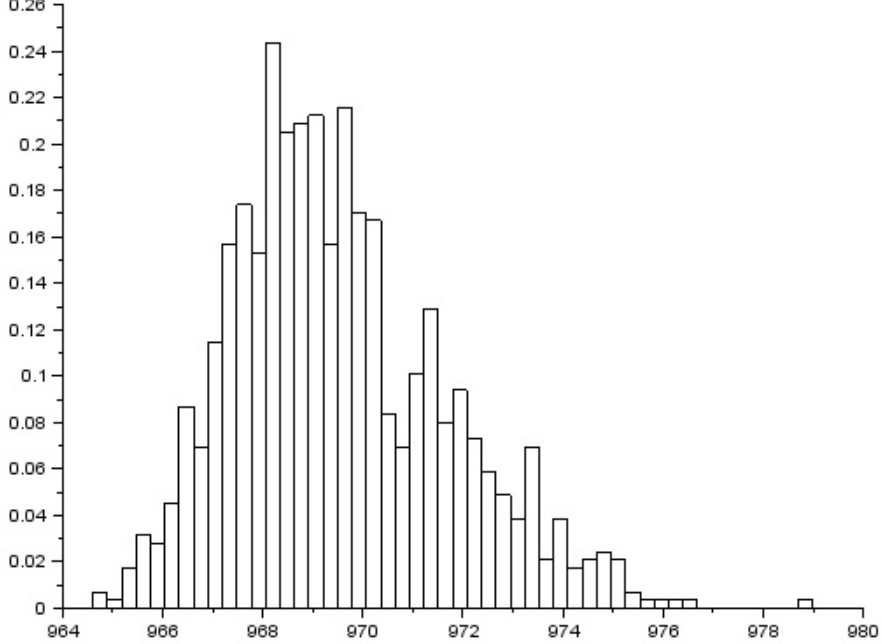


FIGURE 2. Bandwidth Histogram @ Gigabit Distribution vs Speed (Mbps)

4. CONCLUSION

The data shows that given the test area of $64km^2$ the current implementation of the wireless mesh network would be infeasible due to high latency. However, there is a large improvement in bandwidth over the average high-speed internet available via ISP, which currently averages about 30 Mbps.

This is a very abstract model of a simple implementation of a mesh network. A complete mesh network would be able to determine the best node to forward the data to based on the situation. It would also be able to send data to multiple nodes to ensure delivery or improve performance.

The charts show that Bandwidth and Latency are not directly correlated. This means that an effect in the latency does not guarantee an effect in the bandwidth and the only real factors are the distance between the nodes and the number of nodes transversed. Theoretically, although not supported by my charts, bandwidth and latency are correlated because selecting nodes that are further away would reduce the number of nodes necessary and, therefore, improve latency. On the other hand, choosing nodes that are closer would close the distance between nodes and, therefore, improve the bandwidth between the nodes. So distance between nodes affects both latency and bandwidth.

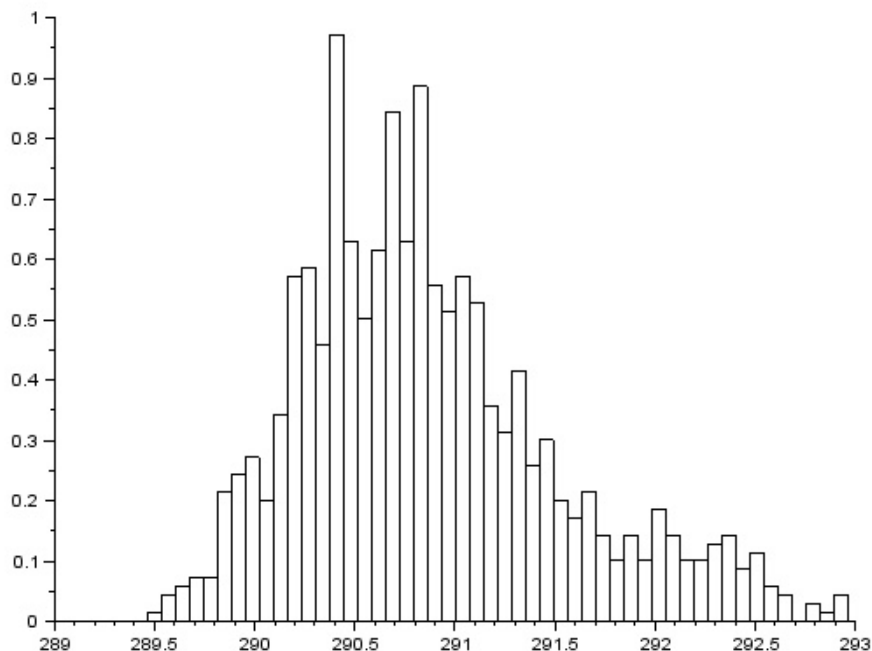


FIGURE 3. Bandwidth Histogram @ 300 Mbps Distribution vs Speed (Mbps)

I would like to move in the direction of a self-managed networking system similar to the one modelled here. This would allow us to move away from the shackles of ISPs and Data Carriers and implement network capability into nearly all devices. Cars would be able to talk to each other freely to prevent accidents and improve road safety, data would be available to you from almost any device at nearly any location, lost devices and people could easily be tracked and found, the benefits are limitless and all without having to pay hundreds of dollars for the privilege to access them.

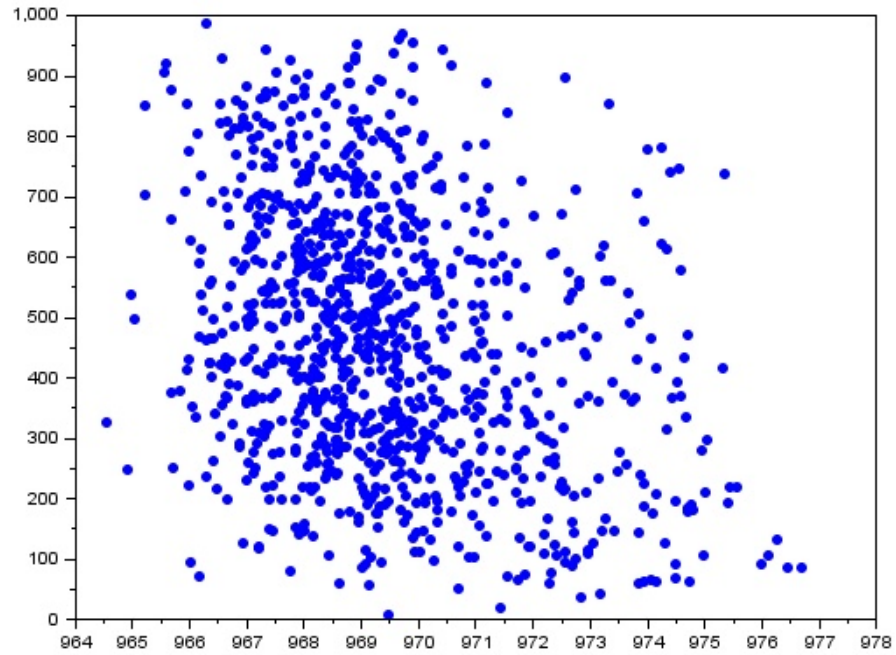


FIGURE 4. Bandwidth(Mbps) vs. Latency(ms)