Jacob Uzzle

CSC 412

Prof. Ruoxi Zhang

11/28/2018

Caerostris: A Decentralized Network Model for Freedom-Minded People

## Introduction:

The Internet wants to be free. People using the internet want it to be free. People running websites and host servers want it to be free. The major corporations, governments, and moral busybodies of the world, however, have other ideas in mind. It seems as things currently stand, the idea of totally free, private, and unfiltered internet sits far out of reach. And realistically, having an unrestrained internet yet useful and friendly internet DOES sit out of reach. A really terrible, hectic, and generally psychotic internet, however, would be exactly what was necessary to create a nearly anonymous and unrestrained internet. None, as far as I can tell, have been easily made like this. To this extent, I propose the creation of a network that falls on the latter side of the internet coin, a model that prioritizes freedom of speech and democratic ownership over the means of communication over ease of access and practical usability.

I am trying to create a network model that:

* Circumvents centralized control,
* Allows users to remain reasonably anonymous,
* Disregards any need for IP addresses or servers (preferring direct UDP services instead),
* Circumvents centralized corporate censorship,
* Provides for unrestrained freedom of speech,
* Allows for mass communication with reasonable control on flow in order to prevent DDOS,
* Operates on a dissociated network of personal/private nodes/routers whose only knowledge is about the nearby nodes replicating most packets over the network,
* Uses an abstract idea of communication bands each with their own set of procedures and data specifications,
* And contains large, difficult-to-create website-like structures that constitute the access and content of the entirety of each of the abstract bands

The name of this network model for now I will refer to as ‘Caerostris’ after *Caerostris darwini*, the spider that makes the largest known web.

As one may expect, this sets up a unique number of challenges to security, maintaining privacy, eliminating spam, ensuring an anti-hierarchical system and providing freedom of speech. Balancing all of these presents a unique challenge, and all of these may not even altogether fit snugly in the same network model. Since Caerostris must use decentralized networking to circumvent authoritarian control over the entire network, and since the network model is explicitly designed to provide to users the ability to speak freely, the problems of spam, anonymity, and privacy will take a back seat to freedom of speech, and this will take a backseat to anti-hierarchical control.

## Definitions:

Caerostris- the network model in question

Relay node- a node in the network that resembles the router/packet switch of the normal internet system. A relay node has multiple connections to other relay nodes and user nodes and broadcasts duplicates of all messages receive, with many exceptions.

User node- a node in the network representative of a person or machine using the network. A user node is expected to connect to only 1 relay node and does not broadcast received messages.

Strand or Abstract Band- a partition of the network where each message sent along one strand stays on that strand.

## Relation to Other Models

I will begin by addressing certain other network models similar to this and their differences; namely, the peer-to-peer system which contains the most similarities to this model. In an abstract sense, Caerostris will use a similar idea to the ‘Guiding Principles’ outlined by the P2P Foundation website, in particular:

* Promoting the concept of individual and networked participation (Principle 1)
* Remaining an alternative to current socio-economic and cultural-political orders (Principle 2)
* Using the ideas of free and open-source software (Principle 5)
* Putting the reigns of society back into the hands of the people (Principle 7)

The issue with the rest of the principle, however, remains that Caerostris either uses the principles of peer networking and applies them to a narrower context, that of a physically, not abstractly, defined network model, and with a specified set of networking procedures.

For example, Principle 4 outlines the creation of a ‘new public domain’ for anyone to be able to use. Caerostris, while not a network model designed explicitly to promote the public domain, must have a ‘public domain’ idea in order to circumvent centralized corporate or business control. The network must cling fast to ideals of decentralization, which needs a free set of open software or even hardware specifications for each node and client access.

Principles 3, which discusses the creation of global markets, and 6, which discusses the reconnection to an older tradition of cooperation and egalitarianism, however, Caerostris will mostly ignore. This network model does not pursue the purpose of global markets, as it does not follow any ideas of buying or selling (except potentially the initial set of nodes). It also does not pursue any sort of egalitarian pursuit or any ancient traditions. The idea behind this model states that the individuals hosting it collectively and individually decide whether or not to participate. It also maintains that everyone be allowed a voice given access to the program. The network should not focus on giving everyone equal time or access except when they dedicate themselves to do so, and it does not have anything to do with global markets unless the users choose to make it so.

## Network Layers

The Caerostris network model will operate on every level, from the very top level with Applications to the very bottom layer on the physical connection level. In order to maintain the integrity of the network specifications, each one of these layers have to adhere to certain protocols. These network layer specifications must reflect the idea of decentralized control and anonymous and non-censorable communication.

### Layer 5: Application

Many modern networks use DNS and HTTP to access websites. Implementing HTTP and DNS in Caerostris, however, represents a unique problem—both often require a client/server system, which more or less violates the concept of decentralized control. HTTP, according to Fielding et al., requires a client/server system in order to function according to specification. It does, however, operate the process of simply transferring information in the form of text, which Caerostris will use (pp. 8-12).

SFTP, Secure File Transfer Protocol, probably follows the idea of decentralization better than IP, as SFTP simply transfers files from one device to another over a network. The SSH website says that FTP and SFTP simply act as connections over TCP/IP model, allowing each one to access to the other’s files, though simply transferring files from to the other works over this. Using this idea, though, remains quite insecure and allows too much interface between the user and the relay.

The question remains, however, how do we actually create the network? This is where the behavior of each of the nodes comes into play. Each node dedicated to relaying packets in Caerostris will broadcast all received packets to all other nodes with links to that node, excepting the node that just sent the packet. Each packet will also receive a random identifier marking it as a more-or-less unique packet, and each relay node will make sure not to re-broadcast packets with identifiers that they have received within the past certain number of packets.

Of course, this also negates the use of websites, as websites necessitate the use of a client/server system where a direct connection establishes a conversation between the IP addresses of the client and the server. DNS would work this way, but Caerostris does not implement such; thus Caerostris will have to implement something along the lines of a radio broadcast, something resembling a band system. Websites will have to be replaced with ‘abstract bands’ or ‘strands’ that each operate on their own set of rules and each communicate separate sorts of data. As well, strands may have ‘sub-strands’ that will operate according to a certain number of specifications depending on the creator implementing the sub-strand; however the number of sub-strands may be limited per strand as they will require data within the application layer payload to determine what sub-strand it belongs to. The purpose of all of this will be to separate information in case of too much chatter on one single strand.

### Layer 4: Transport

Due to the broadcasting nature of the Caerostris network and the lack of IP addressing and server/client modeling, the best model for the transport model appears to be a modified UDP and not other transport protocols such as TCP. UDP is an extremely light-weight protocol which only concerns sending data over certain ports—though it does presume IP addressing, it can function without it and only send to ports. According to Postel, it concerns sending data with minimum protocol requirements and sending data in fixed datagram sizes (pp. 1-2). A lack of server/client systems inside of its requirements also UDP ideal to broadcast information over each relay node. Each relay node within this specific model need only worry about demultiplexing the data received over each port from neighboring relay nodes and users, and multiplexing all of this data to nearby nodes over the ports dedicated connections.

When it comes to each individual strand, the data on one strand must be broadcast by a relay node over the same strand that it originally came from on the same port. User nodes will receive all data over the same set of ports. Ideally, in order to keep the idea of strands separate and fit with ‘out-of-band’ communication, relay nodes should communicate various strands over different ports as well as communicating strands over dedicated internet connections (as outlined in Layer 2 Data Link section). Each sub-strand, however, should communicate over the same port or dedicated connection as its super-strand and be translated or parsed apart by user applications.

As well, congestion control should not only receive management from UDP. If a relay node receives too many packets from a specific other relay node within a certain amount of time, it should restrict the amount of bandwidth in connection to the other relay node. If this continues, it may be prudent to simply terminate the connection to the offending relay node.

While this ‘shunning’ a node like this may not reflect values of ‘free speech’ per say, the only alternative would be to make the entire Caerostris system vulnerable to mass DDoS attacks. Since each relay node broadcasts copies of received packets to each other node, without such a protocol all it would take was one spammer to clog up the entire network. A DDoS attack of this sort would, in fact, restrict the freedom of the network to run, and would therefore act as an individual threat to the free speech of the whole. Thus, balancing this with the ideal of democratic ownership of the network, this can prevent spamming and encapsulate the idea of free speech for a larger group of people.

The use of individual and democratic control over flow control should reflect the idea of decentralized networking outlined in the driving idea behind Caerostris. Each operator of a relay node should have the personal discretion to decide the number of packets received from a certain relay/user node per second before cutting them off or restricting their bandwidth. Since there will be no centralized control anyways, nothing can stop individual relay node operators from changing or managing these rules. The idea behind Caerostris relies on community involvement; if no one wants to implement these rules, the point of Caerostris will be moot anyways, regardless of centralized control. So long as a majority of the operators involved believe in freedom of speech, the network as outlined will function.

### Layer 3: Network

Caerostris must use the IP in order to work. As Postel states, UDP has to assume IP in order to function (p. 1). The problem arrives, however, when IP addressing comes into play. Caerostris CANNOT in any way use IP addressing. While Caerostris fundamentally relies on its internet existence, the only two parts of the network that should know what an individual user/relay node has broadcasted are the sender of the packet and the direct receiver of the packet. Instead of IP to determine the uniqueness of a packet, each packet MUST receive an almost unique 64-bit identifier chosen at random, so each relay node knows not to re-broadcast a packet with the same identifier received within the past 1,000 or so packets. With this structure, the chances of a new packet receiving an identifier within the 1,000 inside of an individual relay node’s received packets is near zero. The IP address must be omitted within the UDP used, making it a modified form of UDP.

### Layer 2: Data Link

The data link should resemble any number of real-world connections. Data transmission could use anything from radio to Wi-Fi to Ethernet to fiber optics to communication via tom-tom drums.

Strands, however, make Caerostris need a separate method of distinguishing between each individual piece of the network. Strands use the idea of separation to cut out the noise for each individual listener, yet they also need to reflection parallel transmission to increase the speed of packet flow through the network. Given how each individual node must duplicate all packets received, improving speed must stay important or the service will not function effectively.

Ideally, each packet received from each strand should be transmitted along and received by a connection dedicated to that strand. For each strand that a relay node broadcasts on, it would be prudent to have an entirely different actual band to transmit that data along (e.g. a separate frequency, a different router to relay along a separate Ethernet cable, a separate Wi-Fi connection etc.). There can be multiple ways a user or relay node should be able to access a certain strand:

1. Sending a message to a relay node on which strand they will be broadcasting on or listening to, and the relay node will add their connection to a list of nodes to broadcast/receive that strand.
2. Directly connecting to a relay node ONLY along the dedicated network link/actual band that corresponds to the desired strand.
3. Receive all of the data from all strands over the same band or link

A is recommended for people who desire quick access and easy access over anonymous access. B is recommended for people with a desire for anonymous access, though connecting is more difficult as it may require changing what kind of connection the node even uses. C is recommended for people with only one possible connection to Caerostris and a limited budget at the dire cost of speed. A and C shall only be possible if nearby nodes use A’s proposed list of broadcast links and C’s proposed idea of shoving all of the data over the same link. B should be the method that most relay nodes use to communicate with other relay nodes. A should be the method that the casual client should use. C should be primarily used for user-to-user node communication, as the casual client usually initiated by the user would not have multiple simultaneous connections.

One must also consider that physical connections pose a problem. An Ethernet cable would require a direct connection between nodes of Caerostris, potentially infringing on people’s property rights. One could, if they were inclined, use the already-set internet structure, but this would require usage of IP addresses and the more centralized normal internet. If each were connected over direct wireless connections, however, this might solve the problem and avoid using existing internet architecture. According to the Wi-Fi Alliance, over 9 billion devices already use it so such a feat may not be problematic for overall efficiency. Such as it is, however, the connection should be left up to the users running the programs, though direct wireless connections may serve this better.

## Weaknesses:

The weaknesses of the program, one could say, run rife, but are more or less necessary to reflect the decentralization used by the Caerostris network. For example, a relay node could feasibly, depending on the connection style, tell which machine gave it what information. This would make it difficult to trust any single relay node and potentially restrict what people can and cannot say, since data collection could occur at any point in the network.

In addition, smashing together all information broadcast over the network would lead to a lot of chatter. This would make the network difficult to use and incredibly slow. Trolls using relay nodes could also pose quite a problem; by broadcasting the same message over and over again, they could use a number of separate relay nodes to DDoS the entirety of the network. This, however, would require mass mobilization in order to attack several of the relay nodes at once.

The problem of censorship can also occur on a small scale within this network. If an individual relay node does not like a message it has received, it can drop it if the owner programs it in such a way. If enough of these relay nodes get together and begin doing the same thing, entire swathes of this internet model can be censored. This will, however, rely entirely on the majority of the network complying with a different set of specifications of their own accord. As well, if an individual wants to make it more difficult for themselves to be censored or hacked by an individual relay node, they should eagerly host their own relay node. If enough people do NOT want freedom of speech, they shall not get it, thus keeping this in line with the idea of decentralized control over the network trumping a lack of censorship.

## Conclusion:

Caerostris will operate under the specifications to prioritize freedom of speech and anonymity above interference and ease of access and decentralized control over everything else. Every part of this network has to represent the core values outlined in the beginning of this document, that of decentralized control with an emphasis on sharing information without powerful middlemen controlling the flow. From the top layer, which requires bare-bones reconstruction of the received packets and strands, to the data link layer which requires navigating the tricky idea of strands, each part will remain a challenge with its own unique advantages and flaws. For Caerostris, it will remain to be seen whether or not broader circles will find a use for it or if it will merely stay useful to fringe elements.

Bibliography

Fielding, R, Irvine, UC and Gettys, J. et al. “Hypertext Transfer Protocol”. RFC, June 1999.

Postel, J. “User Datagram Protocol”. 28 August 1980.

P2P Foundation Core Group. “P2P Guiding Principles”. P2P Foundation,

<https://p2pfoundation.net/infrastructure/our-guiding-principles>

SSH. “SFTP – SSH SECURE FILE TRANSFER PROTOCOL”. SSH.com, Nov. 1 2018.

<https://www.ssh.com/ssh/sftp/>

Wi-Fi Alliance, “Discover Wi-Fi”. Wi-Fi Alliance, 2018. <https://www.wi-fi.org/discover-wi-fi>