

Research Philosophy

Andrew Rosen

Executive Summary

I study, research, and develop highly fault-tolerant and robust systems. Fault-tolerant and robust systems are what hold a complex environment like the Internet together. The Internet is made up of millions of computers/ billions of devices, all going wrong all the time. The packets are dropped, the routers are poisoned, the hard drives keep crashing, some suit/poor sod tripped over a power cable, the software is full of bugs, we meant to add security but it was adopted before we got the chance, and nobody can connect to the printer.

It is, in a word, chaos. Complete chaos that works because smart people decided that failure was an assumption baked into every layer of every protocol. Networking only works because every layer, every part is fault tolerant.

Fault-tolerance is in many ways the most important part of what makes the Internet and networking in general work, and as companies get bigger and create larger and larger intranets and datacenters, as more people in the third world gain access to the Internet, adding billions of handheld devices that rival the power of last decade's consumer desktop, this problem is only going to get bigger.

I study fault-tolerant systems because they are absolutely critical to the Internet and networking, which have become so emeshed in our life.

My research primarily focuses on Distributed Hash Tables (DHTs), with applications towards peer-to-peer (P2P) networks, Big Data, and Distributed Computing. DHTs are primarily used in P2P applications, but other applications, such as botnets, use DHTs for their decentralization.

I believe that collaboration is vital to academic growth. I have worked with researchers in both Computer Science and other scientific fields, including Astronomy, Biology, and Psychology.

1 Past and Current Research

There are many different types of DHTs, but all DHTs share very important qualities. They are *scalable*, which means that each additional node in the network minimally impacts the cost of keeping the network organized. DHTs are also highly *fault-tolerant*. Unlike many other systems, DHTs assume that nodes will be continuously entering and leaving the network. Because of the way DHTs are organized, they can handle large scale failures, such as a power outage affecting an entire city. The last quality of DHTs is that they are *load-balancing*. This means the data stored in a DHT is evenly distributed among nodes in the network.

ChordReduce [?] was a proof of concept for using distributed hash tables for distributed computing. It used Chord [?], a well studied DHT, and exploited its various features to perform MapReduce [?] tasks, a very popular method for framing distributed computing problems.

Unlike other distributed computing frameworks, we made no assumptions as to the context in which ChordReduce would be used. ChordReduce could be used in either a large datacenter or in completely heterogeneous, peer-to-peer context. In a heterogeneous network, we assume all the nodes represent different pieces of hardware, with differing levels of computational power and reliability. As a result, we needed to rigorously test the fault-tolerance of the system and made an exciting discovery while doing so.

To test fault tolerance, each node essentially flipped a coin weighted in its favor, and when it lost the flip, left the network and reentered as a new node. This simulated a P2P concept called *churn*, turbulence in the network caused by the continuous entering and leaving the network. Churn is normally a chaotic influence on the network, but we found that at high levels of churn, nodes advantageously redistributed work in such a way that tests with high levels of churn performed better than tests with low levels of churn.

ChordReduce established three important ramifications. First, we experimentally demonstrated that DHTs are capable of being used as a framework for distributed computing. Second, DHT based distributed computing can have nodes enter the network at any time, even when a job is running, and be immediately put to work. Finally, we discovered that while churn is normally a disruptive force, at high levels it can be helpful to the network.

All of these qualities are highly desirable in distributed computing, so it was a natural step for us to build distributed computing framework based on DHTs. We have shown in previous work that DHTs can be used to distribute work among nodes the same way data is distributed [?].

I find great value in abstracting techniques and platforms even further, as that makes it possible to see novel applications for various technologies. This can be seen in my work on Distributed Hash Tables, where I showed that DHTs can be mapped to Voronoi Tessellations and Delaunay Triangulation.

We used this insight to create UrDHT [?], a generalized framework for building distributed hash tables. UrDHT essentially acts as a “fill-in-the-blanks” for creating distributed hash tables. The novelty of UrDHT is that it makes it easy for scientists in any field to construct a DHT based application. For computer scientists, UrDHT provides a means of rigorously comparing different architectures.

A large part of why I do research is that I want to help people. When I started doing undergraduate research, I worked in the Sonification Lab at Georgia Tech. There, I helped PhD students implement their research in creating non-traditional user-interfaces geared towards the visually impaired community.

2 Future research