March 8, 2021

Reading Summary: On the performance and fairness of BitTorrent-like data swarming systems with NAT devices.
Alexander McRae, V00890516

1 Please describe the problem(s) in your own words. Is the problem important at the time of paper publication, and how about now? Why?

The problem this paper addresses is that although current usage on the internet could benefit from seamless mobility, current internet architecture does not support it. Furthermore, previous attempts at providing mobility either cover a subset of situations or require large changes to the current internet architecture.

2 Please describe the main idea(s) in your own words. How is the idea different from the existing work at the time of paper publication? How does the idea impact the follow-on work till now?

The main idea of the paper is to provide a userspace wrapper around TCP connections, then using a new packet format keep separate sequence numbers from TCP. Whenever a peer changes network locations they can reconnect 1 of 3 ways, wait for a timeout and use the global name service to reconnect, (client) reconnect to the server assuming the same address, (server) tell the peer you are moving and where. This handles the 3 peer mobility situations, connection time, individual, and simultaneous.

Further, the paper allows peers to create multiple flowpaths per connection using ConnID to locally identify the connection and PathID to identify a path between two network interfaces, one connection can have many paths. This allows for multihomed situations and limited reconnection time when reconnecting as more paths could be used to transfer data. Due to the TCP independent sequence numbers and userspace IO buffers the connection does not care which path the packets take and handles out of order transmission well.

This is different from work at the time of publication as it does not require any operating system changes and handles all forms of peer mobility not just a subset. Further work has built off of it and we have seen new protocols such as QUIC use many of the ideas from the paper.

3 Please list at least three most important things in this paper. Why do you think they were important at the time of paper publication? How about now?

1. It accurately corrects and identifies some of the mistakes made in the past such as conflating IP with ID and the failures of the attempts to fix it. This library and paper doesn't get much use in current day but has provided key insights into performance and new approaches to tackling the problem. The same can be said at the time of the paper.
2. The paper clearly acknowledges the usage and prevalence of middleboxs such at NAT and tackles them head on. This matters today more than ever as NAT is found everywhere. At the time of publication NAT was being widely used.
3. The code is both readable and usable (github), including 3rd party services (Auspice). This is really important as it strengthens the paper's ability to explain and reason about real world performance. It also has allowed many new papers to come from it and projects from the code.

4 Please list at least three things you think may need further improvement in this paper. Has the improvement appeared in the follow-on work already?

1. Although the paper was generally readable, it skimmed over a couple of things which caused me to refer to the code. For example *connectionless control sockets* are an

important part of the implementation, however it is only mentioned a handful of times with no dedicated section clearly defining its role. No follow on work has occurred to remedy this.

2. Both the implementation and protocol are both extremely complex, this may lead to buggy implementations. In order to lower the complexity the protocol should remove features and push the responsibility to a higher level for some things like the proxy. No further work does this.
3. Not TCP compatible. Although the underlying sockets are TCP, the protocol does not stay compatible with TCP as it creates its own packet format. This means that it cannot interface with standard internet architecture at the time and fails to solve one of the problems it identified. No work has been done to allow this with the same set of features.

5 Do you have some ideas of your own on this problem? Can you do something better or differently? How can you show that?

I believe the protocol can be greatly simplified and improved upon by making it compatible with existing protocols. One of the main issues with the protocol is it still requires a large change to internet architecture to work, one of the problems it identified in the paper. I believe and am pursuing this term that a subset of the features can be completed while maintaining compatibility with both TCP and UDP.

Link: https://web.cs.umass.edu/publication/details.php?id=2326