Paper Review
Anycast-Aware Transport for Content Delivery Networks

The paper was origninally published at the World Wide Web Conference (WWW) in Madrid, Spain, April 2009. The paper's authors are: Zakaria Al-Qudah, Seungjoon Lee, Michael Rabinovich, Oliver Spatscheck, and Jacobus Van der Merwe.

The current problem is that Content Delivery Networks (CDNs) can have their TCP session connections interrupted by routing changes. Routing changes moves endpoints around disrupts TCP sessions.

A simple solution will be presented that will handle connection disruptions, along with having clients continue downloading though a routing change.

Anycast CDNs are explained as CDNs that all share the same IP address. Normal network routing protocols are unaware of the difference, and attempt to optimize the path. This automatically routs traffic at different nodes to the closest anycast CDN. Problems with them is that there is no load balancing and connections can be disrupted when routers re-router paths.

The first figure, Figure 1, shows what happens when the routers change the path the CDN servers. It shows a TCP connection going from a Client to ServerA, then a routing change moves the connection so it is Client to ServerB. Figure 2 clearly shows the problem with TCP behavior during this process.

One solution that is presented would be to use static bound servers. Redirections at application level (HTTP redirections) induce static bound servers, but removing any benefits of anycast CDN.

The solution is Anycast Aware CDNs.

On client side: “When a client detects a TCP connection failure during an ongoing download, the client issues an HTTP range request for the remaining portion, assuming that the failure is due to a redirection to a different server.” This can be implemented at application layer, as a browser extension or modification of an application that uses a CDN.

On server side: Lots of hanging TCP connections will form, and can last from 13-30 minutes. Solution: aggressive truncating of connections. After 1 attempt, kill connection completely.

Figure 3 shows the same Client/Server layout as Figure 2, but with the client modifications in place, and then with the client and the server modifications in place.

The next section goes into the implications of the new clients and servers. There will be more overhead due to the establishing of TCP connections (three way handshakes), but only problematic in high bandwidth, multiple reestablishing environments. There will be less chance for clients that have non routing change disruptions to reconnect to servers due to the aggressive TCP closing, but non routing change disruptions are already very insignificant amount of clients.

Experiments are conducted to evaluate the performance the proposed changes. Figure 5 shows the throughput of a 50 MB file download with various path characteristics and connection reset frequency.

The paper states that the marginal cost of restarting the TCP connection is better than migrating the old connection's already built up speed. It states if the server overestimates the bandwidth of a new client, then unnecessary loss will occur. While this is true, no testing is done to verify this.

Weaknesses in this paper include not explaining the current alternatives to CDN or Anycast CDN. The tests do not directly compare the performance of the Anycast Aware CDN to normal CDN with HTTP redirections.

The overall structure of the paper is very straight forward (Introduction → Background → Anycast Aware CDN → Performance Evaluation → Conclusion). The paper shows that Anycast CDNs can be used with routing changes with only minor modifications to the client and server TCP handling.