CSC2231 – Internet Systems and Services

Paper Review #5 – Globally Distributed Content Delivery
Name: **Alex Wun**
Date: **Oct. 4th, 05**

There is nothing particularly groundbreaking about the fundamental concept behind Akamai. On the surface, the Akamai solution resembles so many others: a single Internet services site suffers problems under load, so *distribute* the load among multiple sites. Akamai, in particular, distributes its surrogate servers at the Internet's edge following the same motivations behind P2P systems in general. In fact, it would not be inaccurate to describe Akamai as an enterprise-level P2P network for large corporations. What is interesting about Akamai is that it manages to deploy over 12000 servers across more than 1000 networks (clearly P2P scale) – but it successfully does as a *single* organization. Control of the servers (unlike is real P2P) is *not* delegated to other organizations.

Akamai's success can be attributed to its focus on quality assurance as a company. Obviously, a great deal of cutting-edge technology is employed in their solution: dynamic fault-tolerant DNS (also used for load balancing), real-time customer traffic monitoring and reporting, ability to handle dynamic content and streamed media in addition to regular static content. However, it is clear that Dilley et al. have the extensive industry experience needed to put together such a large-scale project. Their services are heavily client-focused (real-time network monitoring, services customized to client's applications) and service reliability is put as a top priority – shown by how they test new software/hardware with live real-world traffic. Unfortunately, even so, it is difficult to pinpoint (from just this paper) exactly how they are doing what they do so well.

The drawback to Akamai's design lies in its global distribution – in the sense that such architecture is inherently more complex to maintain and operate. The authors admit that a globally distributed system must be extraordinarily careful in preserving scalability and correctness (cache consistency, etc.). Though this is good news for the authors since it means that the requirements for entering this market are quite high and not for the inexperienced.

Particularly interesting is the fact that Akamai is capable of distributing (semi-caching) dynamic and streamed content. This has tremendous implications for massively multiplayer online games. Currently, the "worlds" of MMO games are actually heavily partitioned. Companies make players log-on to a *specific* game server as a form of load balancing. As such, a player is only realistically able to interact with a subset of the other players who are logged onto the same server. The architecture used by Akamai may be a step towards providing a truly MMO game environment that is continuous and "global" rather than partitioned.