Tech Topic #1

January 20, 2010
Today’s Objectives

• CDNs
• A bit on patents
• CDN traffic stats… by Bryce
First There Was the Web

- The web basically got started in about 1992
- It started to take off in late ‘94 and into 1995
  - “take off”: lots of servers and readily available clients
  - The first killer for the Internet was born
- Had dramatic impact on traffic amount and mix
  - Combined with size and complexity of files (multimedia)
  - Coming last-mile broadband revolution
  - Led to rapid capacity increases (contributed to dot-com bust)
  - To a certain extent, network research was born
- Frame of mind is critical
A Major Challenge: Delay

• Lots of traffic led to congestion
  – Net result was increase in delay or timeouts

• Even increase in capacity contributed to some additional delay
  – More hops, more propagation delay, more chances for queuing

• How to solve the delay problem
  – More servers only solves some of the problem
  – Push content closer to users
Some Solutions

• Use caching
  – Browser, local, institutional
  – Need protocols to locate and pull content
  – Good example: Squid

• Other techniques
  – Local pre-fetching (multicast) and possibly filtering

• Use replication
  – “Mirror” content: a kind of static replication
  – Challenge was still to find mirror
  – Evolution moved from static connections to caches/mirrors to dynamically locating best content
Solution Evolution

• The challenge with mirrors is loading ALL content at EACH mirror site
  – Could be wasteful and unnecessary
  – Neat research question
  – Azer Bestavros at Boston U. looked at question of which documents to locate at a mirror/cache site

• Also, the more mirror sites, the more backend content delivery (prefetching)
  – Especially problematic when content was dynamic

• Area was hot and multiple people were looking at the same time
Solution Evolution

• Prior to Akamai was Digital Island
  – Farber 6,185,598 patent

• First Akamai Patent was 6,108,703

• Aspects of the solutions:
  – Use page splitting so original content owner knew of hit
  – Using caching instead of mirroring
  – Use combination of factors to determine closest cache site
  – Use DNS as the mechanism to communicate cache location
  – Also has a mechanism to check object freshness
Patent Components

• Filing date is key
  – Date of patent not so much
  – The priority date can be earlier than the filing date
    • Based on provisional patent date
    • Based on continuation patent
    • Based on conception and diligence in reduction to practice

• References cited is also key
  – What the patent examiner was supposed to consider
  – Some provided by patentee, some found by examiner search

• “Person of Ordinary Skill in the Art”

• Specification

• Claims
The Original Akamai Solution

• “Akamaize” your web page
  – Replace all embedded URLs with special URLs
  – http://<hash>.akamai.com/<stuff>/<original-hostname>/<original-path>/

• <hash> is communicated to akamai.com as part of DNS lookup
  – Provides some information about information being requested
  – Note: rest of URL does not go along with the DNS request
  – Also helps with confirming doc freshness

• Returns info on “close” server cluster
The Original Akamai Solution

• Still need to find IP address of server within the cluster
  – Remember, can either be recursive or iterative

• Where are servers located?
  – Network Access Points (NAPs): places where network providers peer
  – Evolved from just providing inter-connectivity to great place to provide other services
The Original Akamai Solution

• Still need to find IP address of server within the cluster
  – Remember, can either be recursive or iterative

• Where are servers located?
  – Network Access Points (NAPs): places where network providers peer
  – Evolved from just providing inter-connectivity to great place to provide other services

• Number of Akamai servers (not clusters)
  – 28000 as of November 2007
  – 48000 as of May 2009
  – 55000 as of November 2009
The Original Akamai Solution

• Still need to find IP address of server within the cluster
  – Remember, can either be recursive or iterative

• Local DNS server can choose a server from the cluster based any variety of factors
  – Disad is that mapping is stored in DNS cache
    • Could make “lifetime” short, but too short and DNS ignores timeout
  – Evolution of solution was to do TCP connection switching

• Still a challenge of how to do distribution within CDN
  – We’ll come back to this
Another Solution

• Another way of finding the local cluster: anycast
• Anycast is different than unicast, multicast, or broadcast
• Instead of <something>-to-<something>, it is more about finding the closest one of a set
  – The members of the set generally provide an identical “service”
  – “service” can be content or lookup (best example is DNS)
• Let’s talk about anycast routing
Challenges in Using Anycast

• Great for single “transactions”
  – Widely used in DNS today
  – Neat concept of, instead of using static DNS server mappings, use anycast routing

• Harder for TCP connections, but not impossible
  – If a TCP packet suddenly shows up at another server, the “other” server has none of the state (TCP control block or “TCB”) for the original TCP connection
  – See Al-Qudah article for one way of doing it
  – Are there others?
Some Other Thoughts

• CDNs: not just for web content
  – Can CDNs do live streaming?
  – Can CDNs distribute p2p content?
  – What can’t CDNs do?

• Follow the bouncing dollar
  – What is the revenue model?
  – Who pays?

• Moving content around the backend
  – Just caching? Push caching?
  – Use multicast?
  – Use P2P?
Some Other Thoughts

• How much traffic?

• …and now Bryce