Anycast CDNs Revisted

Hussein A. Alzoubi
Case Western Reserve University

Seungjoon Lee
AT&T Research Labs

Michael Rabinovich
Case Western Reserve University

Oliver Spatscheck
AT&T Research Labs

Jacobus Van Der Merwe
AT&T Research Labs

Outline

- Introduction
- Motivation
- Approach
- Algorithms and Experiments
- Results
- Conclusions

Introduction

- Content Delivery Networks (CDNs) are used as means to efficiently and cost effectively distribute media content on behalf of owners
- A CDN consists of nodes distributed across the internet to act as proxies or caches on behalf of content owners
- The challenge is to redirect users (eyeballs) to the "best" node
  - Proximity
  - Prevent servers overload
- Most commercial CDNs uses DNS based redirection. However,
  - Local DNS (LDNS) proximity might not apply for the actual eyeball
  - DNS was not designed for very dynamic changes in the mapping
  - Even with short TTLs, LDNS and some browser caching beyond TTL is still an issue
  - To find proximity between any two IPs in the Internet is still difficult

Introduction … Continued

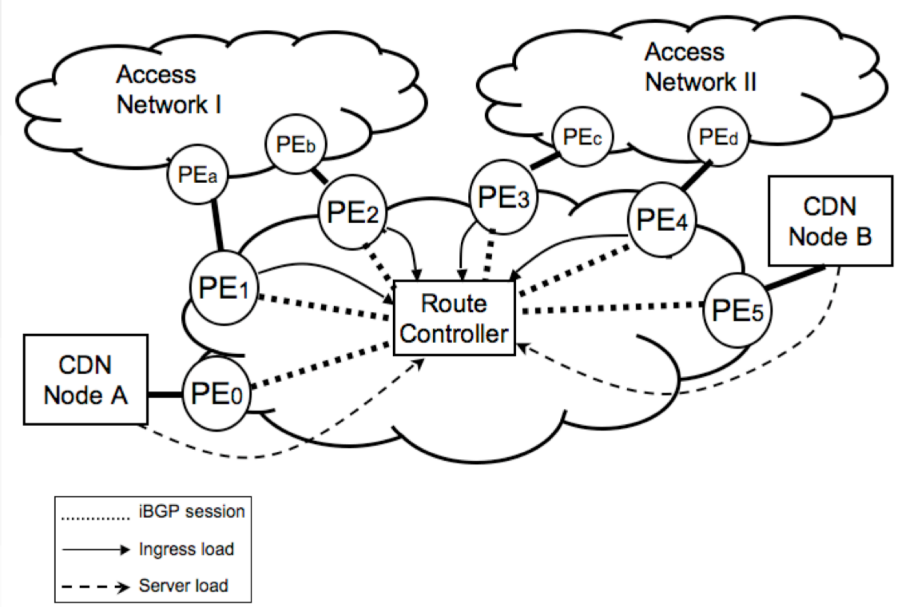
- IP Anycast refers to the ability to allow the same IP address to be assigned to multiple endpoints.
- Thus, IP Anycast packets are always routed “optimally” to the most proximate endpoint from a network prospective.
- IP Anycast was deemed unsuitable for CDNs due to:
  - Any routing changes for Anycast traffic may cause (TCP) session reset
  - IP Anycast is unaware and cannot react to network conditions
  - IP Anycast is unaware of any CDN node (server) load, thus can not react to overloads

Motivation

- For CDNs deployed in a large footprint AS, Route control mechanisms allow route selection to be uploaded to routers
- Recent work promoted that Anycast can be appropriate to be deployed to facilitate proximal routing in CDNs
- We present our work on a load aware IP Anycast CDN.

Load Aware IP Anycast CDN Architecture

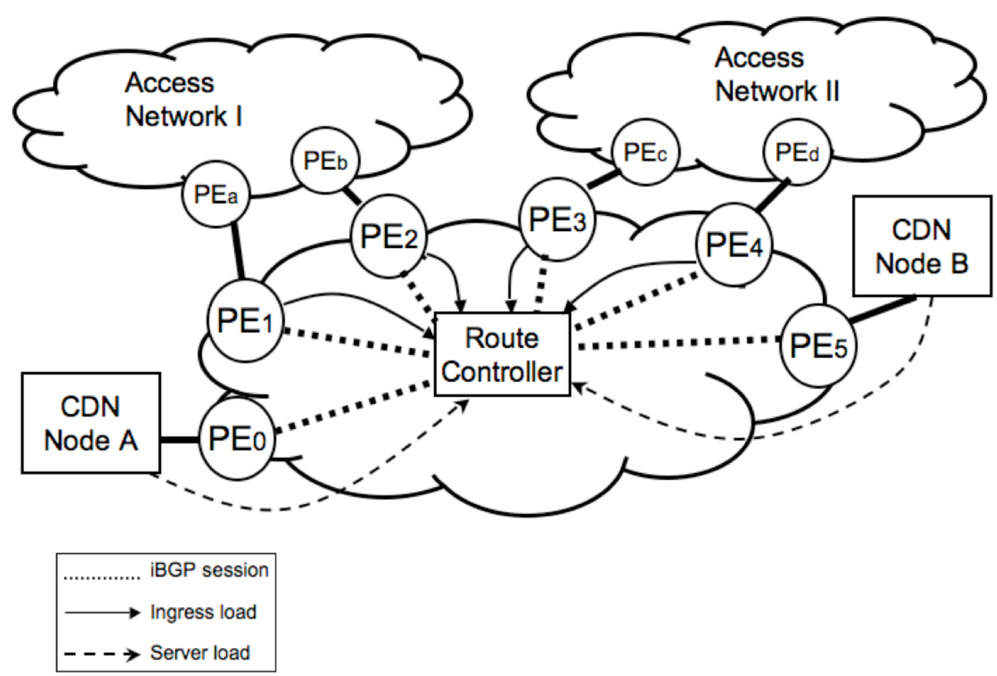
- CDN nodes are distributed in a single (AS)
- Servers {A,B} advertise Anycast IP via BGP through Egress PEs {PE0, PE5}
- Egress PEs advertise Anycast IP to Rout Controller
- Route Controller advertise this IP to the all ingress PEs {PE1 to PE4}which advertise it via eBGP to all peering routers {PEa to PEd}



Load Aware IP Anycast CDN Architecture

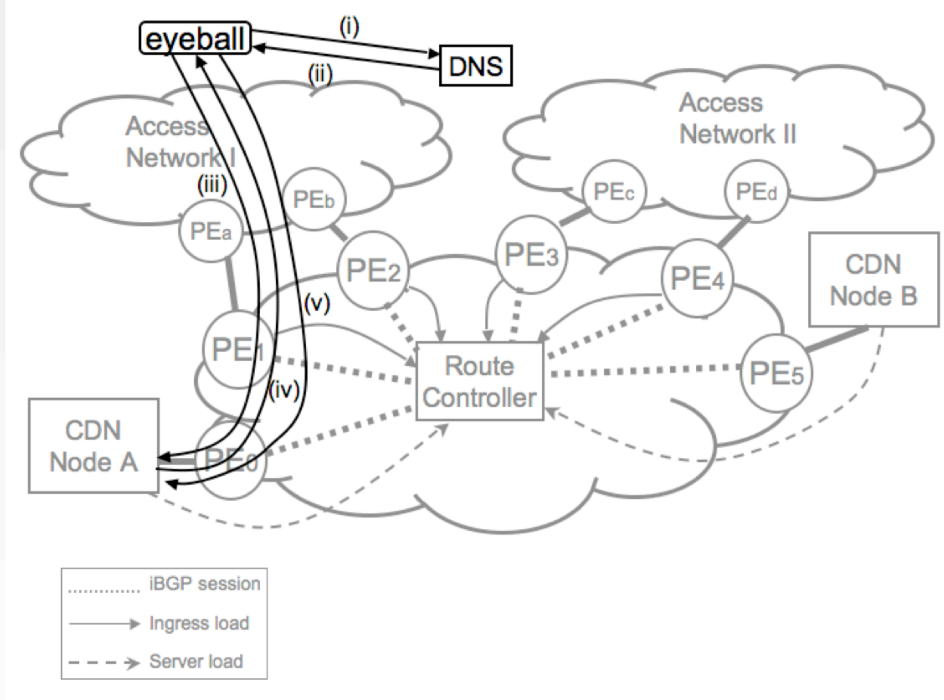
- Route Controller uses server loads from CDN nodes and ingress PEs Loads to decide which ingress PE to be mapped to which server.

- Requests follows the reverse path arriving at the most proximal server



Long Lived Sessions

- We use application level redirection to avoid session reset for large file sessions
- After an eyeball request content and get redirected to the proximal server it tries to establish connection $\{i, ii, iii\}$
- CDN responds with application level redirection containing unicast IP(iv)
- This unicast IP is associated only with this node (v)



Problem formulation

- Our system has *m* servers. Each server *i* can serve up to *Si* request per unit time.
- Requests arrive from *n* ingress PEs each PE *j* contributes r_j amount of requests per unit time
- Cost matrix *Cij* for serving PE *j* at server *i*
- *Shmoys and Tardos* presented an approximation algorithm, Given a total cost C, their algorithm decides whether there exists a solution of overall cost C and indivdual server load < Si + Max r_j

$$\begin{aligned} \text{minimize} \quad & \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} \\ \text{subject to} \quad & \sum_{i=1}^{m} x_{ij} = 1, \ \forall j \\ & \sum r_j x_{ij} \le S_i, \ \forall i \\ & x_{ij} \in \{0,1\}, \ \forall i,j \end{aligned}$$

x_{ij} =1 iff server *i* servers PE *j*

Minimizing Cost Algorithm

- Run the optimization algorithm
- Identify most overloaded server i
- Identify PEs set F mapped to i sorted in ascending order according to their load such that off-loading F will decreases i's load below Si
- Starting with large PEs in F, Off-load PEs to a minimum cost server j with enough residual capacity as long as i's load is > Si.
- Repeat for all over-loaded servers
- If we couldn't find server j with enough residual capacity, find server t where t's load after off-loading is less than current i's load

Minimizing Connection Disruption Algorithm

- This algorithm attempts to re-assign PEs *only* when we need to off-load an over loaded server
- We set priority such that PEs prefer to stay assigned to their current server
- Find a set of PEs :
  - Sufficient to reduce server load below capacity
  - Minimize disruption penalty due the reassigning

Experiments: Methodology

- A production CDN servers log were obtained for a weekday in July 2007
- Two sets were obtained: Large File download, and small web objects
- Each log file contains detailed information about http requests such as: client IP, Server IP, request URL, Request Size, …etc
- Cost matrix reflects air miles between server I and ingress PE j multiplied by load r_j
- Ingress PE load corresponds to the number of connections arriving at PE j at a particular time
- We used CSIM to create our trace driven simulations

Metrics

- Number of Ongoing Connections
- Service Data Rate
- Average Mile a request traverse
- And Number of Connection Disruptions

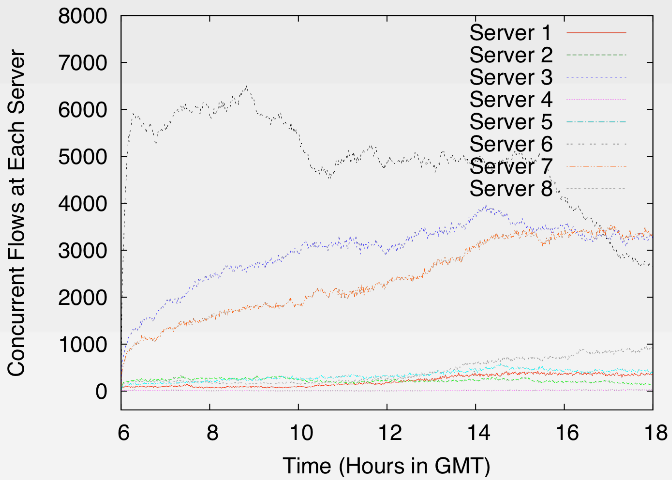
Schemes of Experiments

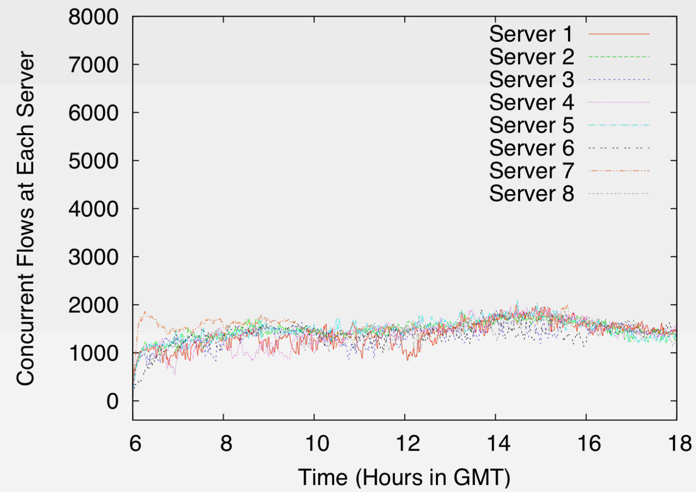
- Simple Anycast (SAC): native anycast which represents an idealized proximity routing. Fixed mapping all the time
- Simple Load Balancing (SLB): minimize the difference in servers load all the time. Recalculated every 2 minutes
- Advanced Load Balancing – Always (ALB-A): follows the minimum cost algorithm. Recalculated every 2 minutes
- ALB – On-Overload (ALB-O): follows minimum disruption algorithm. Recalculated (for a subset of PEs) every 2 minutes

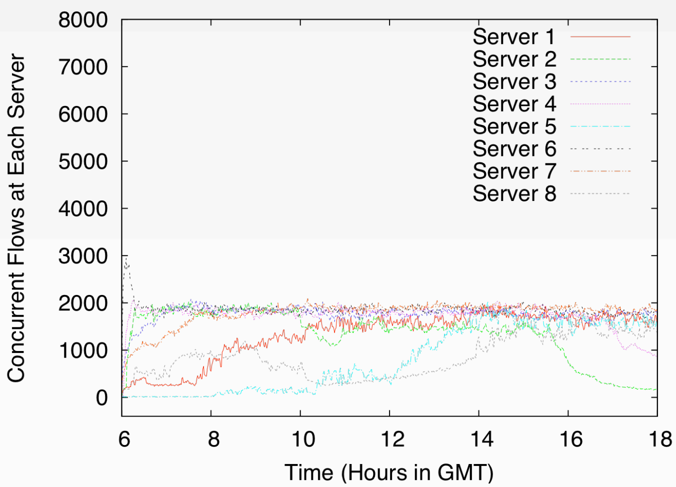
Ongoing Connections

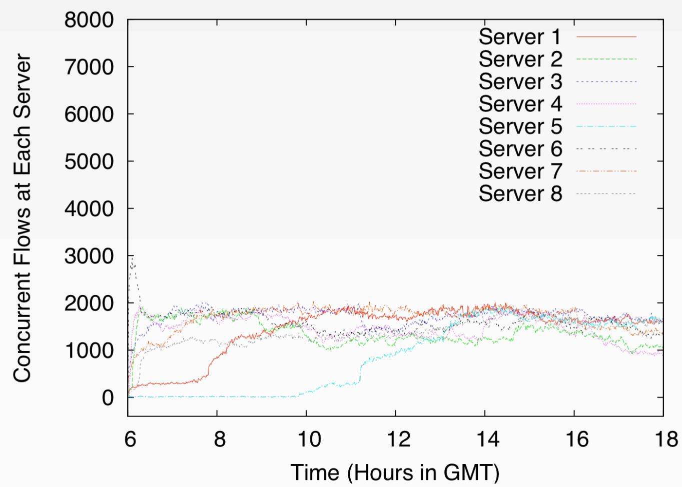
(Large File Downloads)

Case Western Reserve University EST. 1826

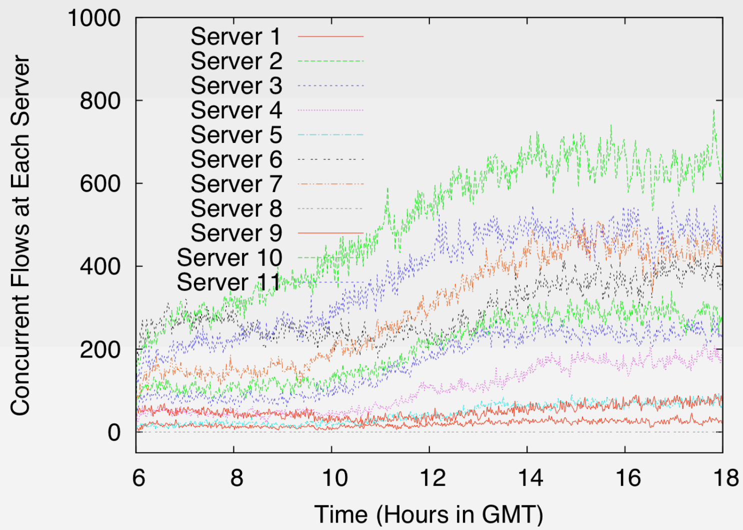


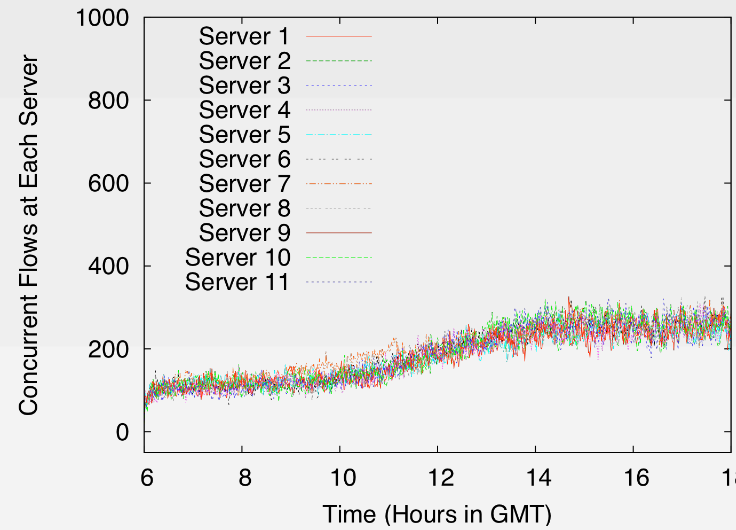


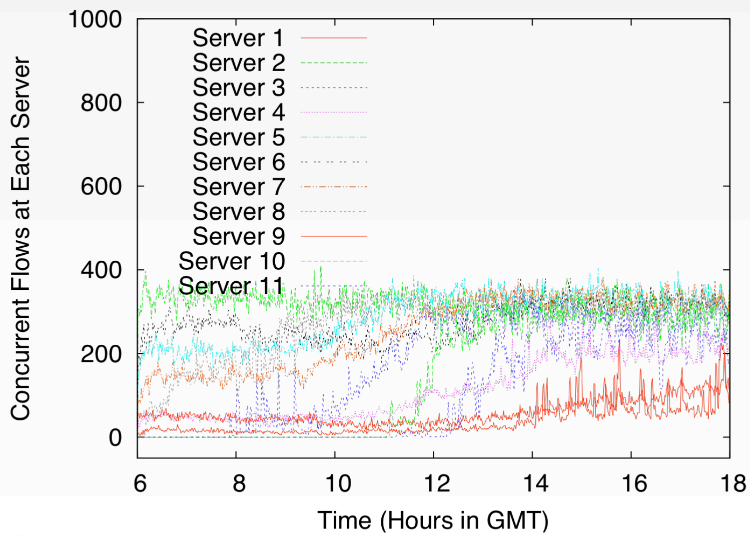


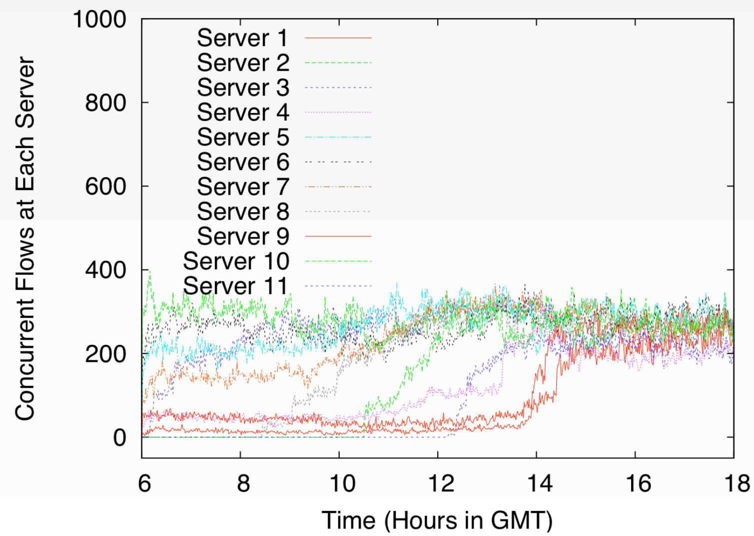


Ongoing Connections
(Small Web Objects)

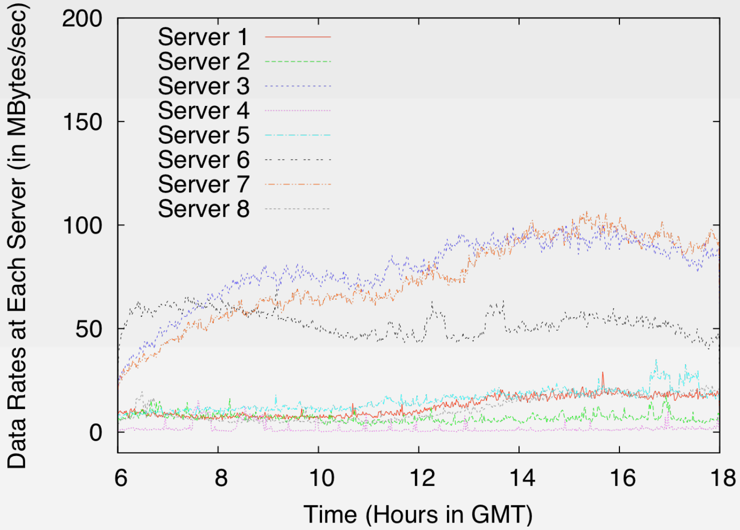


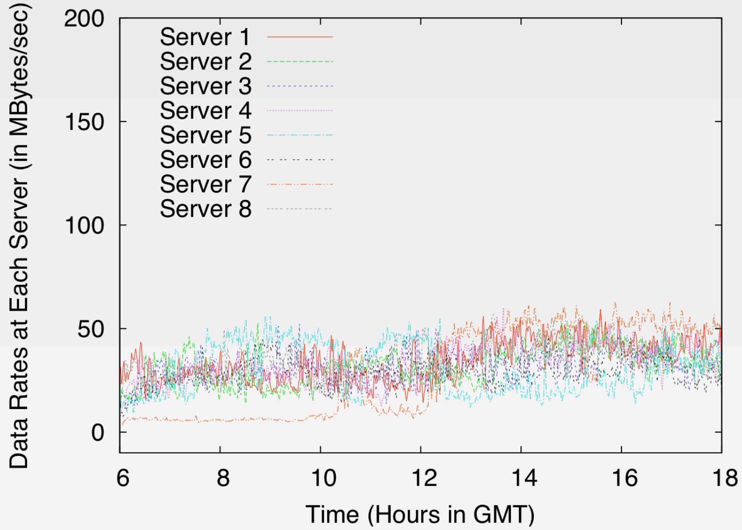


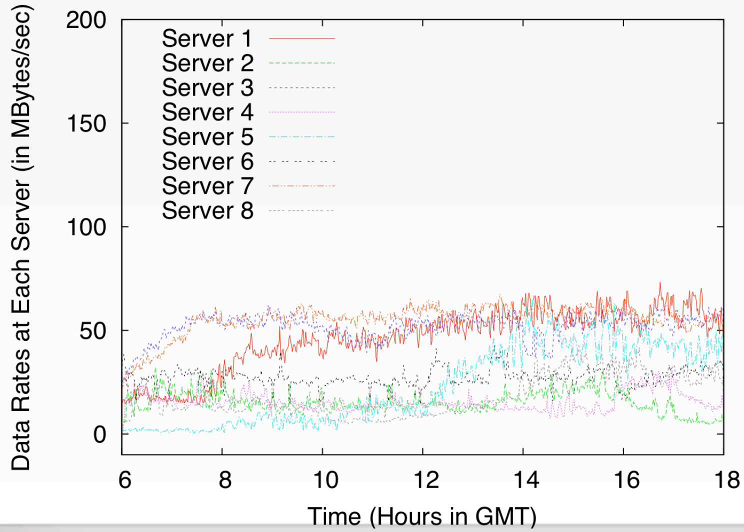


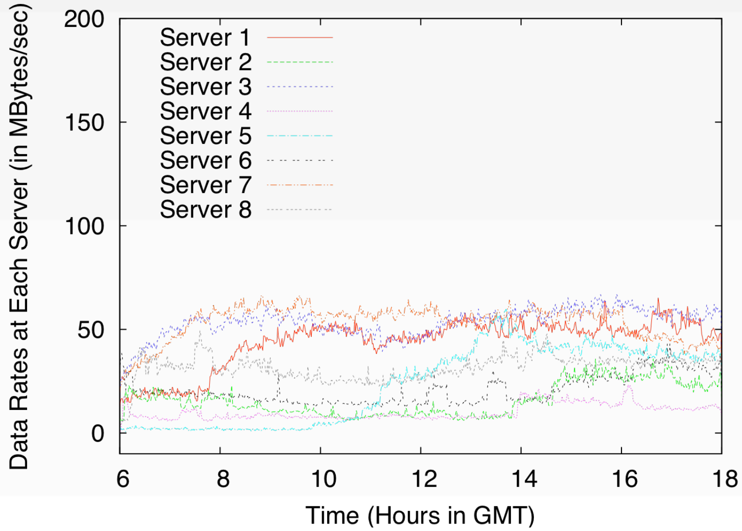


Service Data Rate
(Large File Downloads)



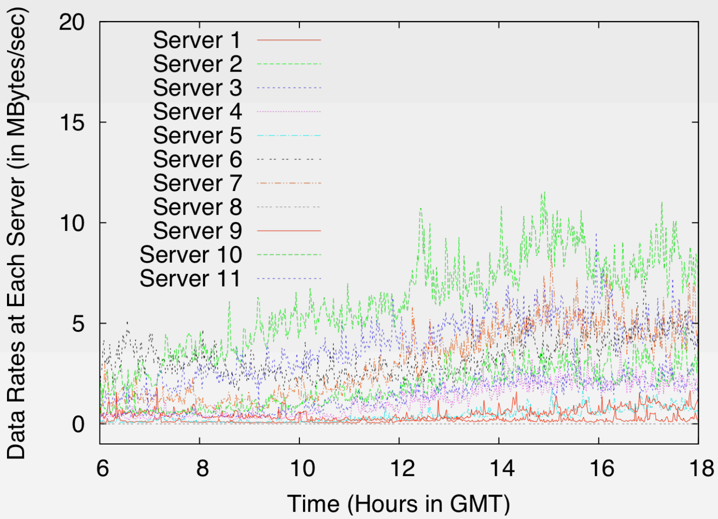


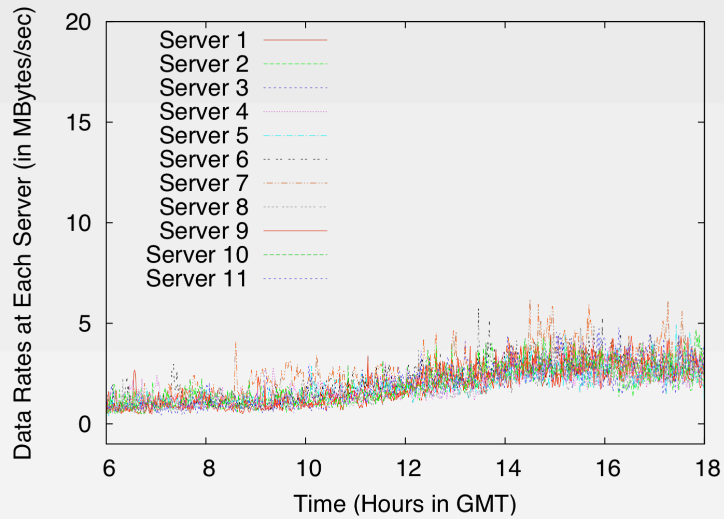


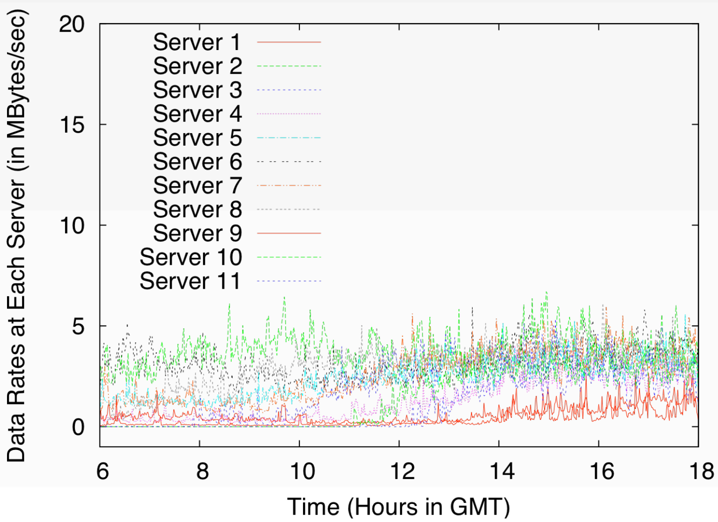


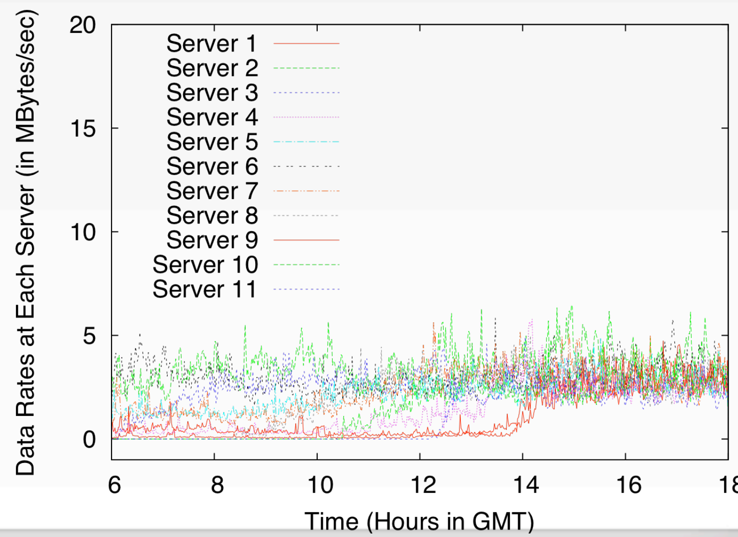
Service Data Rate

(Small Web Objects)

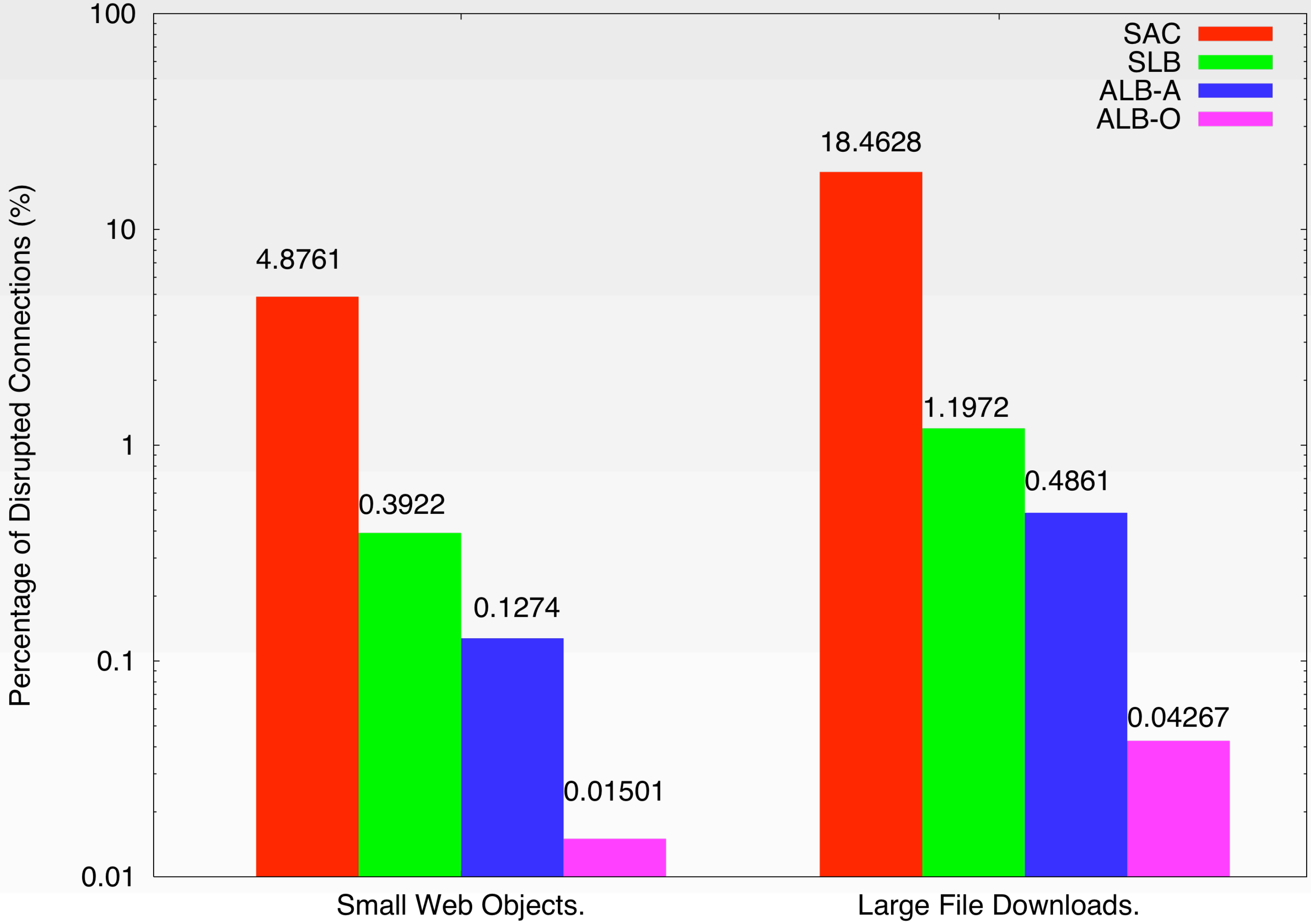




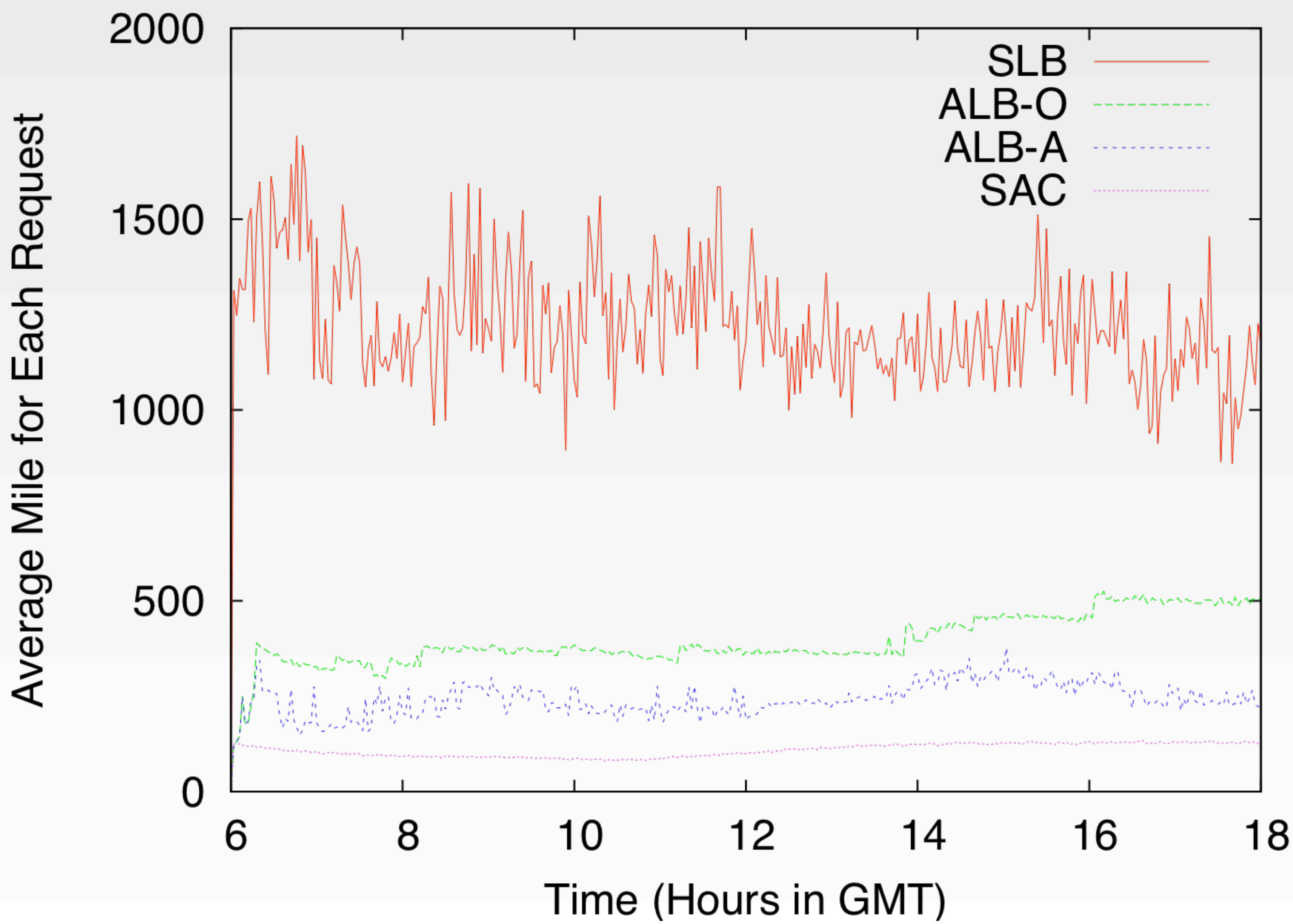




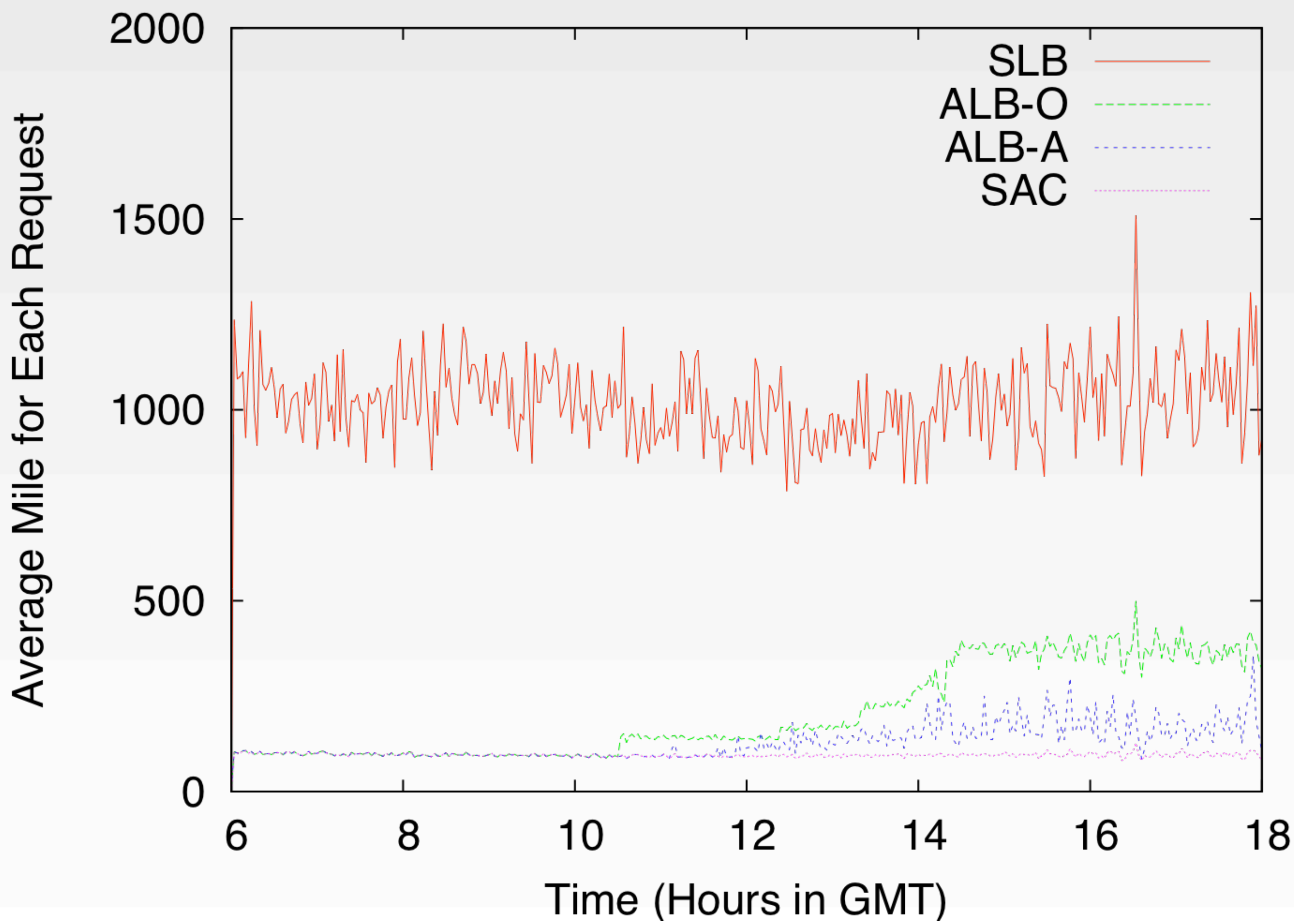
Connections Disruptions



Average Miles Per Request
(Large File Downloads)



Average Miles Per Request
(Small Web Objects)



Conclusions

- We Presented a load aware IP Anycast CDN architecture
- We described algorithms to utilize IP Anycast redirection property to reflect effective proximity and cost redirection
- We evaluated our algorithm using trace data from operational CDN and showed that perform as well as native anycast in terms of proximity and managed to keep server load within capacity

Comments & Questions

?!