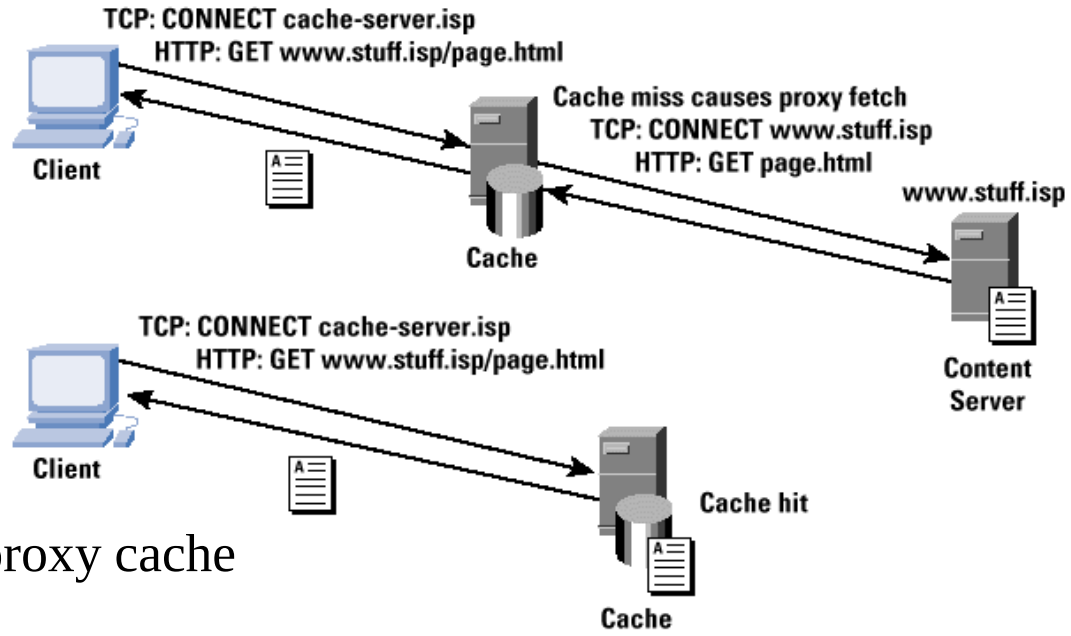
WEB CACHING

Mallika Ghurye, Aditi Sharma, Sruthi Shyamsunder

WEB CACHING

- Storage of web objects near the user to allow fast access, thus improving the user experience of the Web surfer.

- Types of caches
  - Browser cache
  - Proxy cache
  - Reverse (inverse) proxy cache



Summary Cache: A Scalable Wide-Area Web Cache Sharing Protocol - Li Fan et. al.

ADVANTAGES

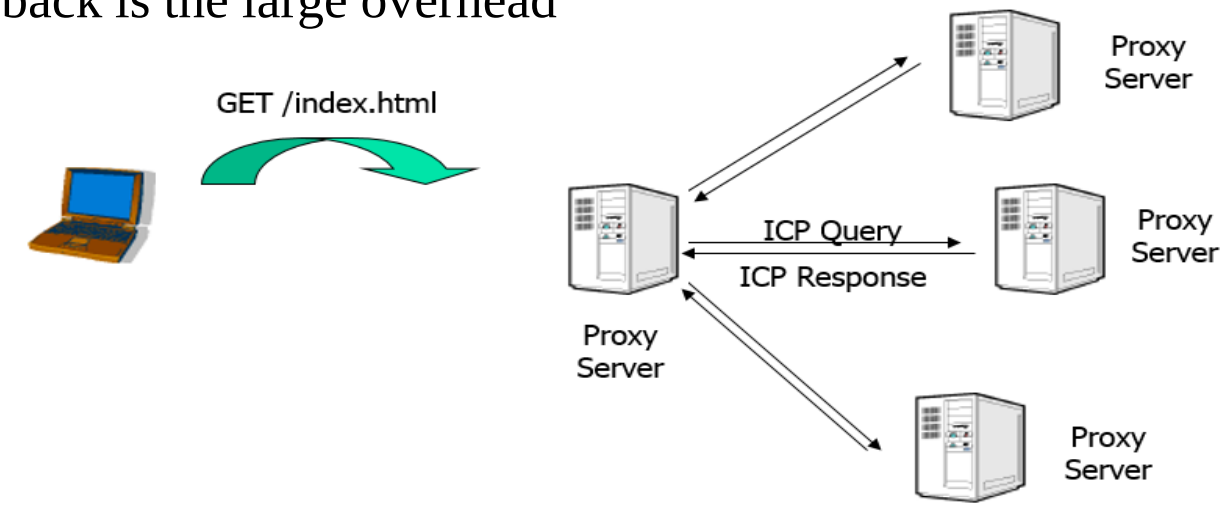
- Faster delivery of Web objects to the end user.
- Reduces bandwidth needs and cost. It benefits the user, the service provider and the website owner.
- Reduces load on the website servers

WEB CACHE SHARING

- As the content on the web grows, an important technique to reduce bandwidth consumption is web cache sharing.
- Improves the scalability of the web.
- Today many networks have hierarchies of proxy caches which interact to reduce the traffic on the internet.

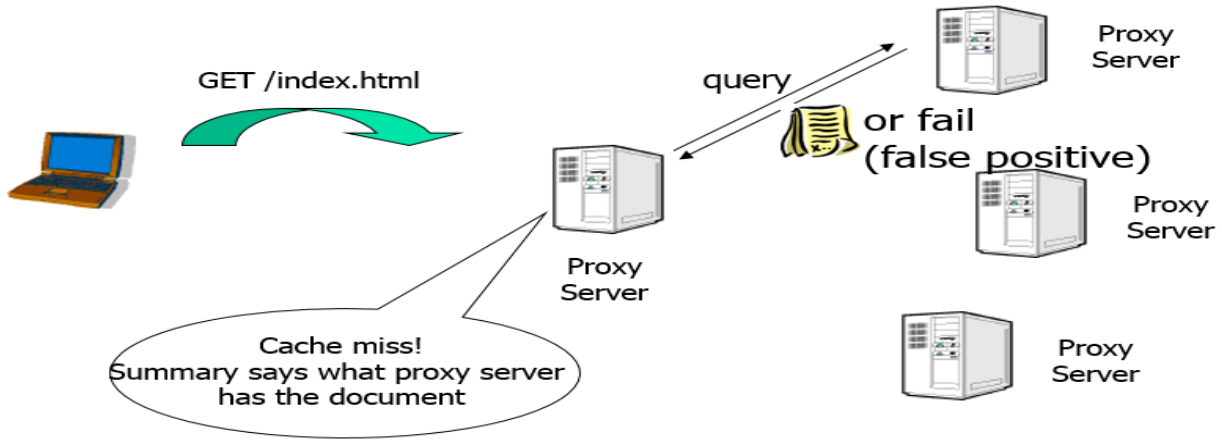
ICP (INTERNET CACHE PROTOCOL)

- Whenever a cache miss occurs, it sends a query message to all the neighboring cache
- As the number of proxies increases, it increases the total communication and CPU processing
- Drawback is the large overhead



SUMMARY CACHE

- Proxy keeps a compact summary of the cache directories of every other proxies
- When a cache miss occurs, checks all the summaries to see if it might be a cache hit in other proxies
- It then sends a query message only to those proxies whose summaries indicate a promising result



KINDS OF ERRORS

- False Miss:
  - summary does not reflect that the requested document is cached at some other proxy

  Effect: The hit ratio is reduced

- False Hit:
  - summary indicates that a document is cached at some proxy when it is actually not

  Effect: proxy will send a query message to the other proxy i.e wasted query message

FACTORS LIMITING THE SCALABILITY OF THE SUMMARY CACHE

- Network overhead (Interproxy traffic)
  - frequency of summary updates
  - number of false hits and remote hits

Solution: Instead of updating summaries at regular intervals , the update is delayed until a percentage of cached summaries is ‘new’ reaches a threshold

- Memory requirement
  - size of individual summaries
  - number of cooperating proxies

Solution: An ideal summary is small and having low false hit ratios. Summaries are stored in the main memory so that the lookups are faster.

CONTENT DISTRIBUTION NETWORKS

- Poor service quality by internet. Two reasons:
  - no central co-ordination
  - Increased load and content demand
- Solution: Distributes content from original server to replica servers close to the end users.
- Replica servers hold selective set of content and requests for that content set are sent.

CDN: Content Distribution Network∗ - Gang Peng et. al

ARCHITECTURE OF CDN

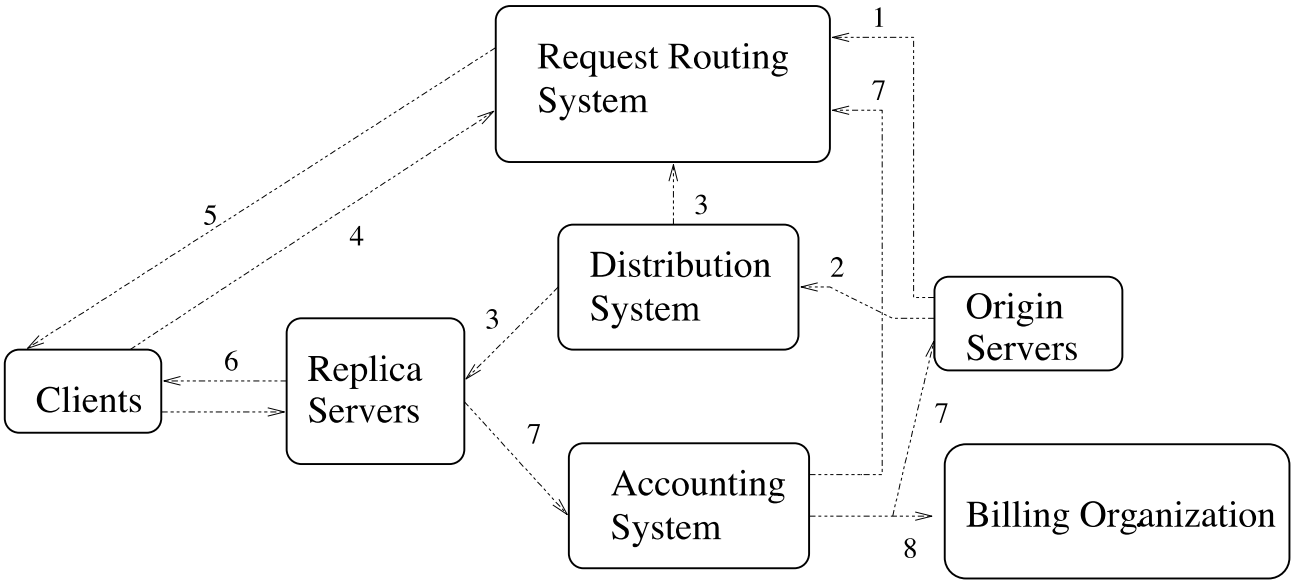


Figure 1: System Architecture Components of a CDN

*from 'CDN:Content Distribution Network' by Gang Peng

ARCHITECTURE OF CDN

- DISTRIBUTION SYSTEM: Distributes content from origin server to replica servers usually via tree or overlay network over the Internet.

- REPLICA PLACEMENT:
  - Where do I place the replica server?
  - Where do I place the replica object?.e.g. Web Page

- REQUEST ROUTING SYSTEM:
  - Sends the requests to replica servers which hold a copy of the requested content.
  - How do I choose a replica server? (distance/load based)
  - How do I route requests to it? (HTTP/DNS redirection, anycasting, etc)

AN EXAMPLE OF CDN-AKAMAI

- Placed replicas at data centres & PoPs of major internet providers.
- Akamizers: URLS to ARLS(Akamai Resource Locator)

- Object data-object freshness parameter

*

- Provider code- unique customer code
- Serial #-a group of akamized objects
- Type- interpretation of ARLs
- Akamai Domain- for Akamai DNS system lookup

*from ‘CDN:Content Distribution Network’ by Gang Peng

NEED FOR IMPROVED E2E PERFORMANCE

- Scope: Increase in internet content and data centers in the cloud had led to an increase in scale, cost and operation.

- Solution: Optimize overall response time using “proxy” front-end (FE) servers closer to users.

- How FE improve user-perceived performance?
  - Cache static portion of dynamic page at FE servers
  - FE can establish Persistent TCP via split TCP connections
    - Eliminates TCP slow-start between FE and BE
    - Reduces RTT between user and server.

Characterizing Roles of Front-end Servers in End-to-End Performance of Dynamic Content Distribution – Chen et. al.

ROLE OF FE SERVES IN E2E PERFORMANCE

- Purpose: Measurement-based comparative study of Google and Microsoft Bing Web Search services
- How?
  - PlanetLab nodes + in-house search query emulator
  - ~40,000 keywords with various combinations
  - Detailed TCPdump and Application Layer data collected
- Different conditions
  - First set – all measurement nodes launch search queries to their default FE servers every 10 seconds
  - Second set – one fixed FE server (Bing or Google respectively) at a time, gets queries from all nodes

DYNAMIC CONTENT DISTRIBUTION

- Content includes static and dynamic
  - Static portion: HTTP header, HTML header, CSS style files and the static menu bar.
  - Dynamic portion: keyword-dependent menu bar, search results and ads.
- Static portion is cached and directly delivered by FE servers.
- Dynamic portion is generated by BE data centers and them passed onto the FE servers for delivery.

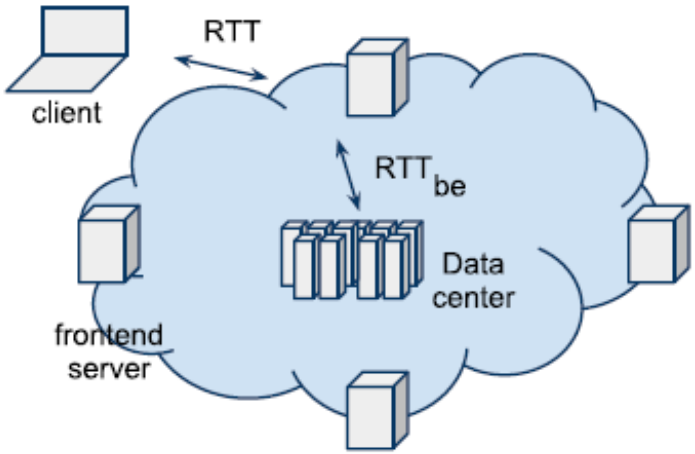


Figure 1: Content distribution infrastructure.

TCP HANDSHAKE AND PERFORMANCE

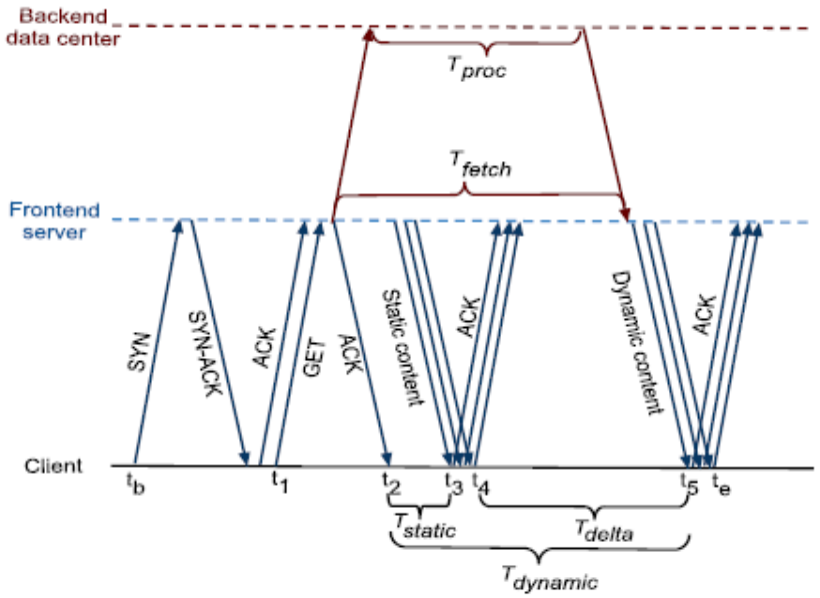


Figure 2: Modeling search query timeline.

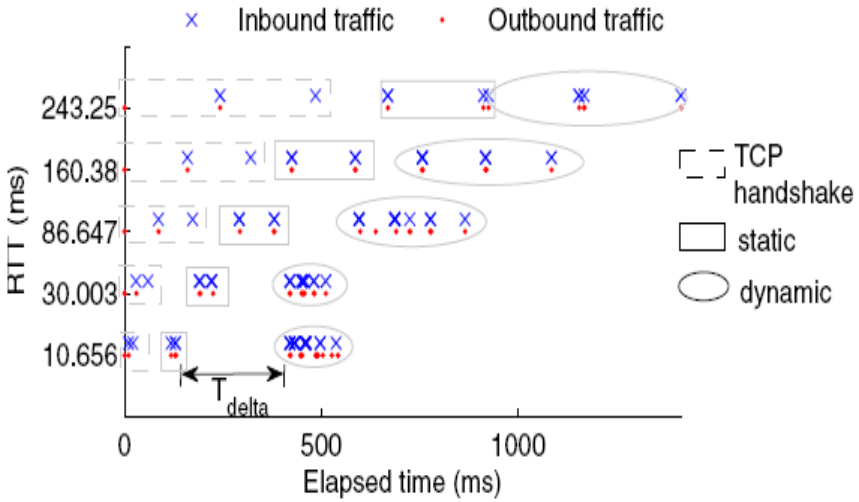
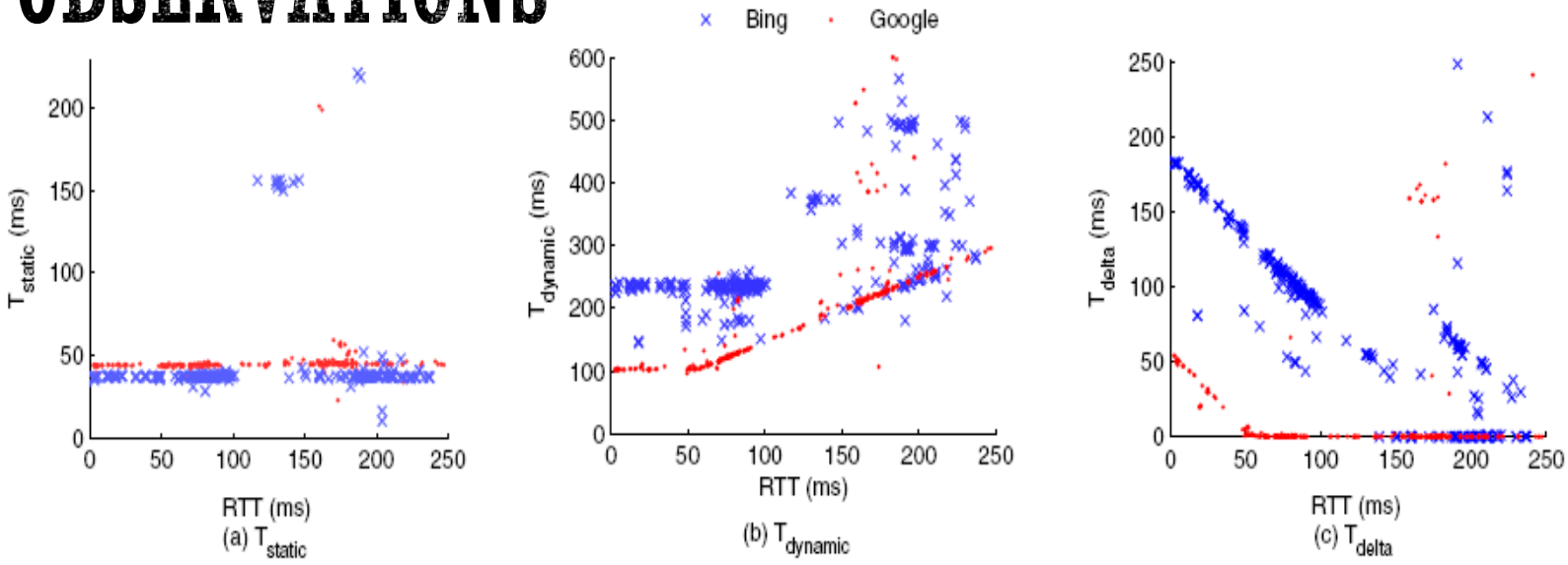


Figure 4: Inbound and outbound traffic events triggered by a single search query.

Several parameters:
t_b: start of TCP three-way handshake
t_1: HTTP GET request
t_2: receive packets from server
t_3/t_4: receive first/last static packet
t_5/t_6: receive first/last dynamic packet

As the RTT increases, the gap between the end of the second and the beginning of the third clusters decreases, and eventually the two are lumped together

OBSERVATIONS



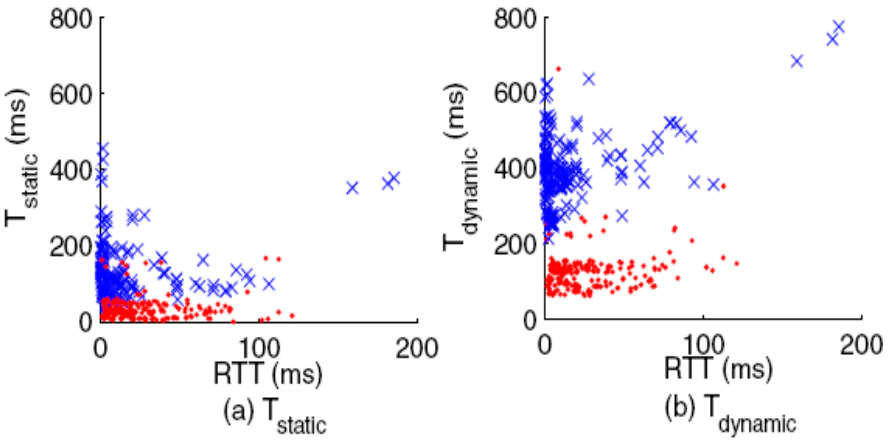


Figure 7: T_{static} and $T_{dynamic}$ for Planetlab nodes using default frontend servers.

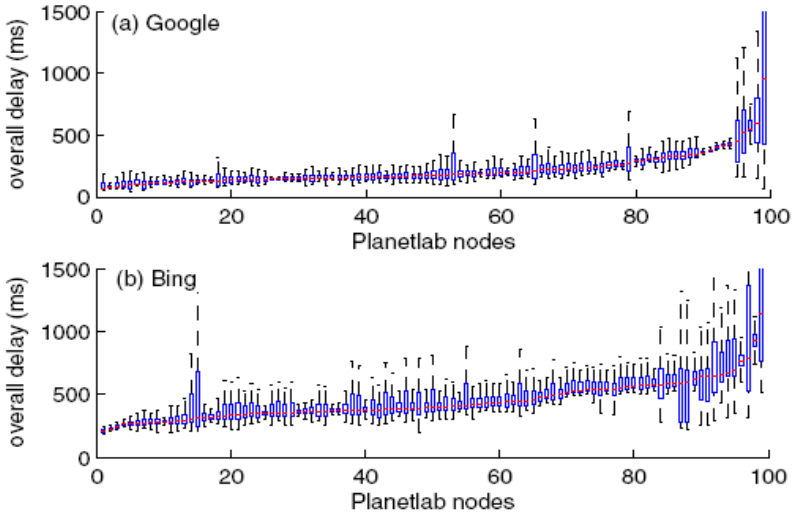


Figure 8: Overall delay performances.

FINAL POINTS OF PAPER

- RESULTS
  - T_{fetch} Google < T_{fetch} Bing and more stable
  - Bing FE servers closer to client but higher T_{static} and $T_{dynamic}$ compared to Google (possibly due to variable loads at Akamai FE server)
  - E2E performance determined by FE-BE fetch time i.e. T_{proc} and RTT_{be}
- SUMMARY
  - FE severs cache the static information of dynamic content but while proximity improves latency other key factors, such as processing times, loads at FE/BE data centers, and the quality of connections between them also play a critical role in determining the overall user-perceived performance.
  - Trade-off between placement of FE severs and the FE-BE fetch time. There is a threshold within which placing FE further closer to users is no longer helpful.
- DESIGN FLAWS
  - Interactive typing of search query was not taken into account
  - Most nodes used were close to Bing FE server hence unfairness/bias possible
  - No significant packet loss. With high loss rate, close FE servers would improve E2E performance.

CONCLUDING REMARKS

- WEB CACHE
  - Cache Sharing
  - Internet Cache Protocol (ICP)
  - Summary Cache

- CONTENT DELIVERY NETWORKS (CDN)
  - CDN Architecture
  - Specific CDN: Akamai
  - Use of Akamai with Bing& Google Example

ANY QUESTIONS?

- THANK YOU!