Content Delivery Networks

Shaxun Chen

April 21, 2009

Outline

- Introduction to CDN
- An Industry Example: Akamai
- A Research Example: CDN over Mobile Networks
- Conclusion

Outline

- Introduction to CDN
- An Industry Example: Akamai
- A Research Example: CDN over Mobile Networks
- Conclusion

What is Content Delivery Network

- A content delivery network (CDN):
  -- a system of computers (computing devices) networked together (across the Internet) that cooperate to deliver content to end users, in order to improve performance, scalability, and cost efficiency.

  - Distributed system
  - Transparent to end users

Motivation

- Request load overwhelms the web sites!
  - Web server processing ability
  - Bandwidth
  - Back-end transaction-processing infrastructure

1.5 billion visits per day

5.4 billion visits per day

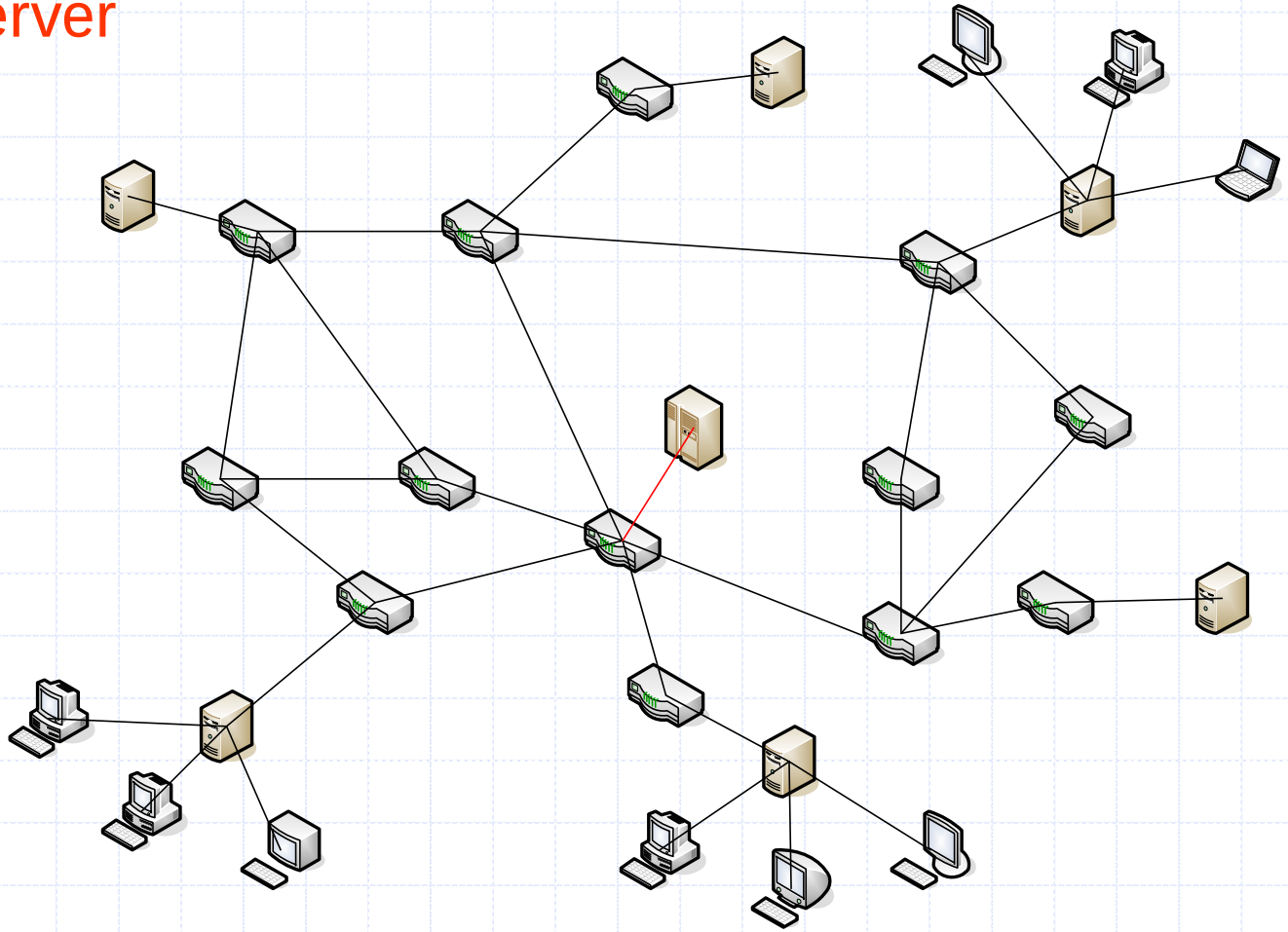
0.2 billion videos viewed every day

ALL YOUR VIDEO ARE BELONG TO US.

- If ..., customers will complain:
  - Speed
  - Availability

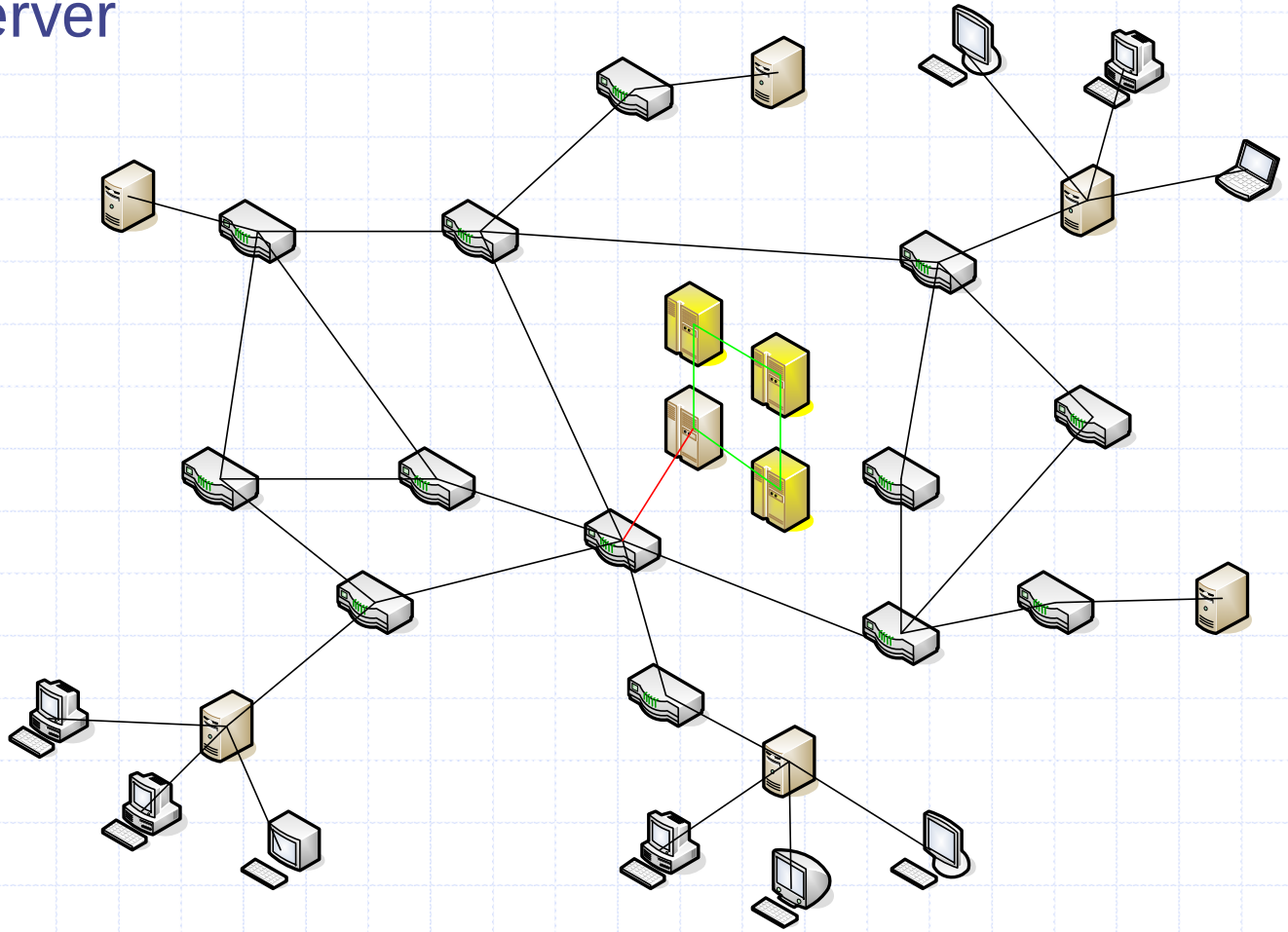
Approaches to Delivering Content

- Single Server
- Mirrors
- CDN



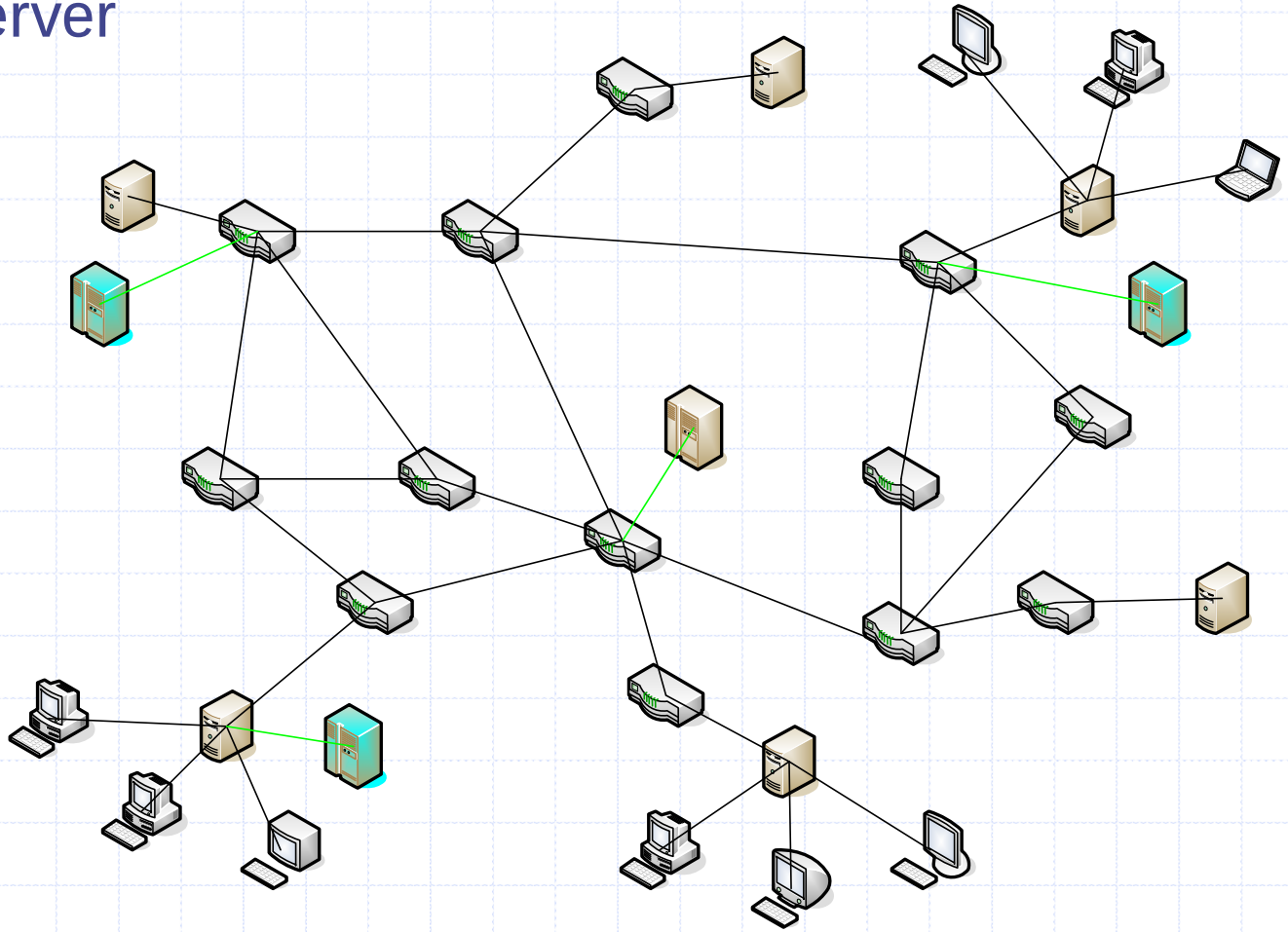
Approaches to Delivering Content

- Single Server
- Mirrors
- CDN



Approaches to Delivering Content

- Single Server
- Mirrors
- CDN



Concepts or Terms

- Servers
  - Origin server
  - Surrogate servers / Edge servers
- Roles
  - CDN providers: Akamai, Digital Island...
  - CDN customers: Yahoo, AOL, CNN...
  - Clients / end users

Advantages of CDN

- Reduce (backbone) bandwidth costs
- Improve end user performance
  - Shorter latency / response quickly
  - Less delay jitter
  - Higher bandwidth
- Increase global availability of content

CDN vs. Web Proxies

- First:
  - Web proxies store most frequently or most recently requested content; work in a passive way
  - What CDNs store is decide by CDN administrators
- Second:
  - Web proxies only handle static web pages
  - Dynamic content / secured content / streaming content (CDN handles them using ESI)
- Third:
  - Proxies work on a local basis
  - CDNs provide more availability

Best Surrogate Server

- From end-users' (clients) point of view, which surrogate server is the best?
  - Nearest (physical distance)
  - Speed (bandwidth)
  - Delay (round-trip time), delay jitter
  - Reliability (packet loss rate)
  - Data transmission cost
  - Server load
  - Combinations of the above

Placement of Surrogate Servers [1]

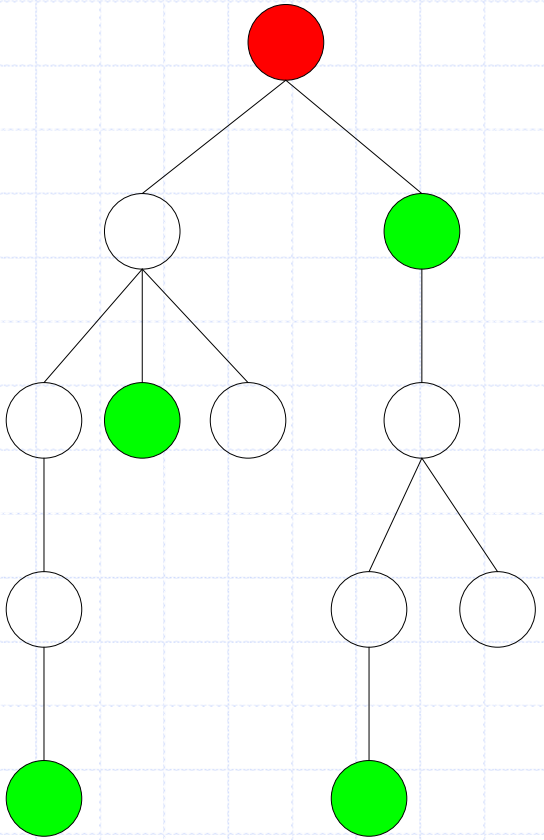
- We have *N* possible locations at edge of the Internet, and are able to afford *K* (*K*<*N*) surrogate servers, how to place them to minimize the total cost?
- Cost:
  - metrics
  - Refers to “choose best surrogate server”
- Formalization: (Minimum *K*-Median Problem)
  - Given *N* points, we must select *K* of these to be centers (surrogate servers), and then assign each input point *j* to the selected center that is closest to it. If location *j* is assigned to a center *i*, we incur a cost $d_j C_{ij}$. The goal is to select *K* centers so as to minimize the sum of the assignment costs.
    - NP-hard

---

[1] L. Qiu, et al., On the Placement of Web Server Replicas. Infocom 2001, Alaska, US.

Tree-based Algorithm [2]

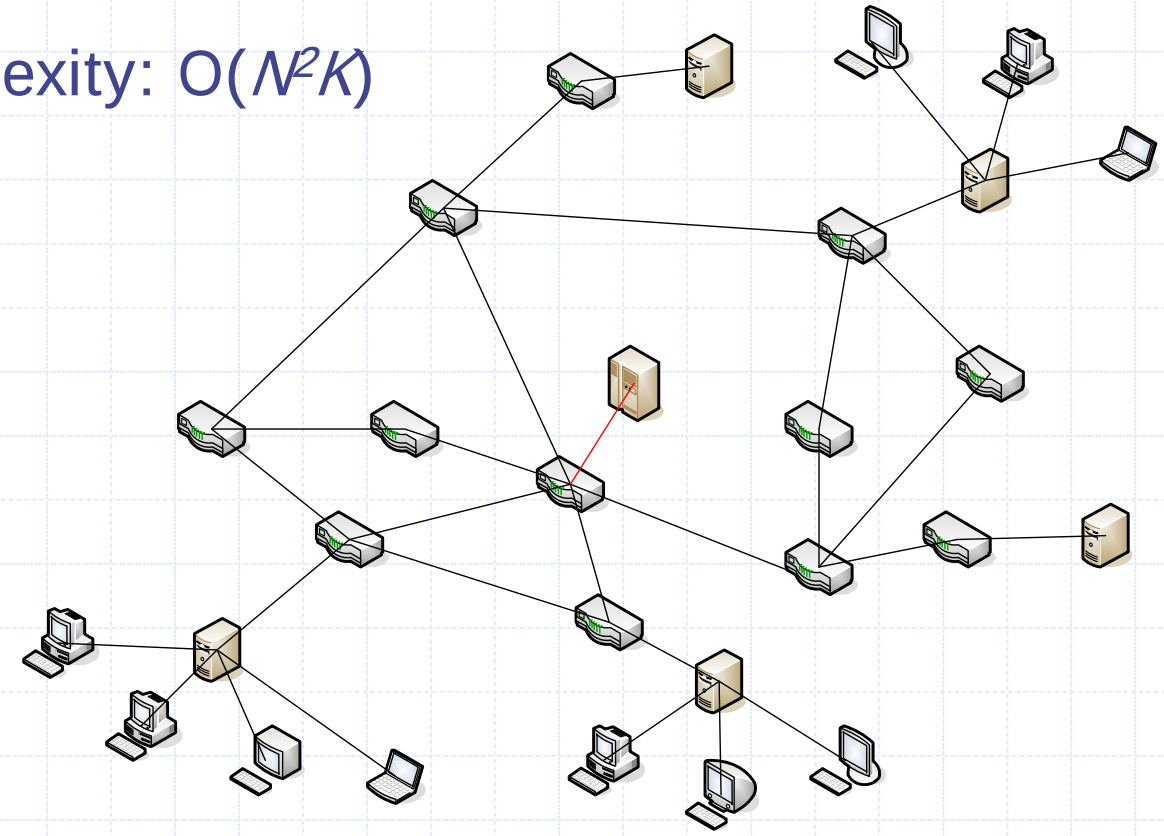
- Two assumptions
  - The network structure is a tree; the origin server is the root of the tree
  - Clients request from the closest surrogate server on their path toward the root
- Based on these two assumptions, we are able to get a optimum placement within $O(N^3K^2)$



[2] B. Li, et al., On the Optimal Placement of Web Proxies in the Internet. Infocom 1999, New York, US.

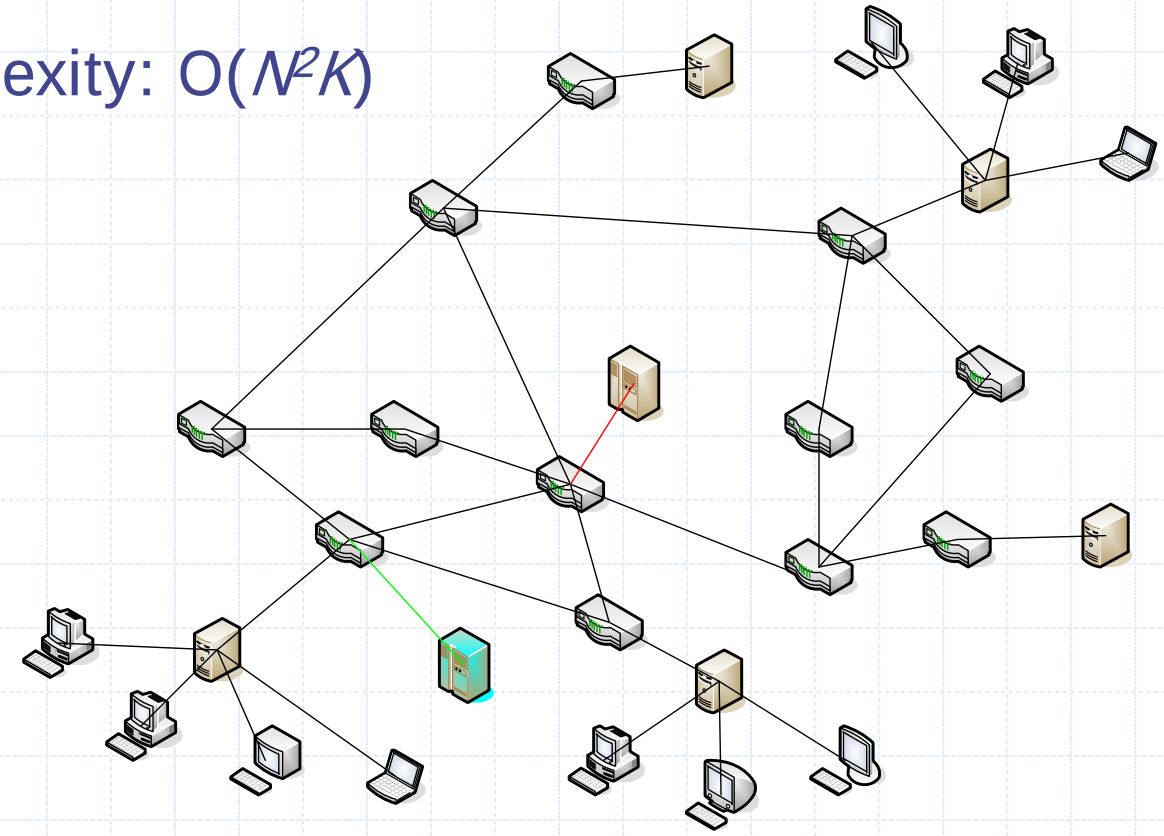
Greedy Algorithm

- Chooses the first center (with minimum cost) among *N* places; chooses next center among rest of *N*-1 places .....
- Time complexity: O(N^2K)



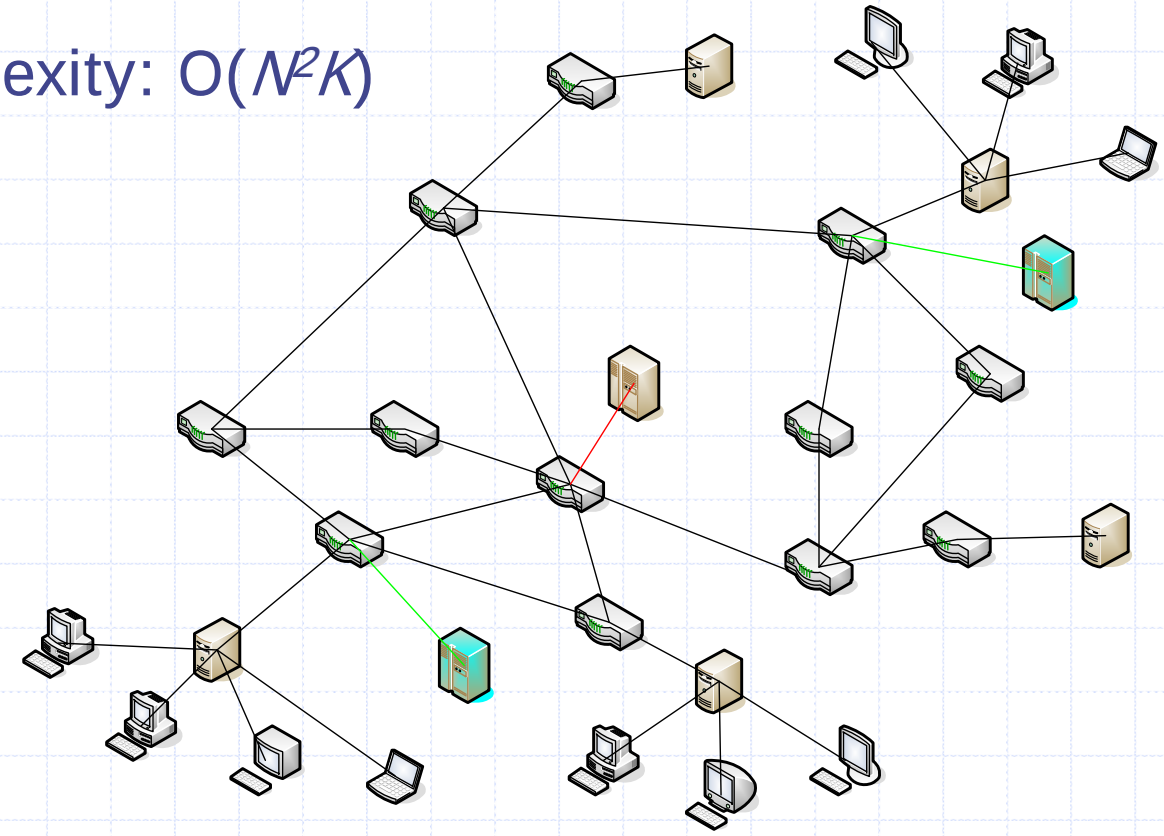
Greedy Algorithm

- Chooses the first center (with minimum cost) among *N* places; chooses next center among rest of *N*-1 places .....
- Time complexity: O(N^2K)

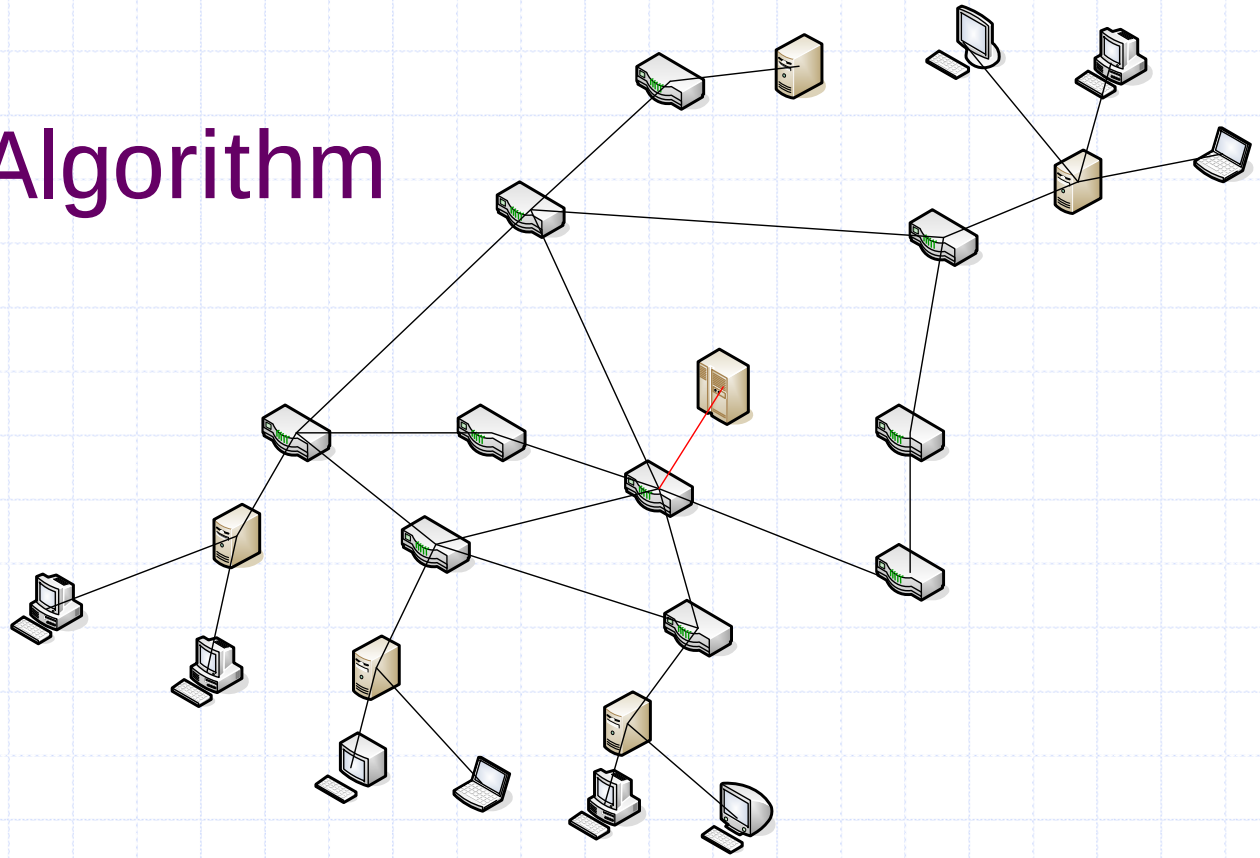


Greedy Algorithm

- Chooses the first center (with minimum cost) among *N* places; chooses next center among rest of *N*-1 places ……
- Time complexity: O(N^2K)

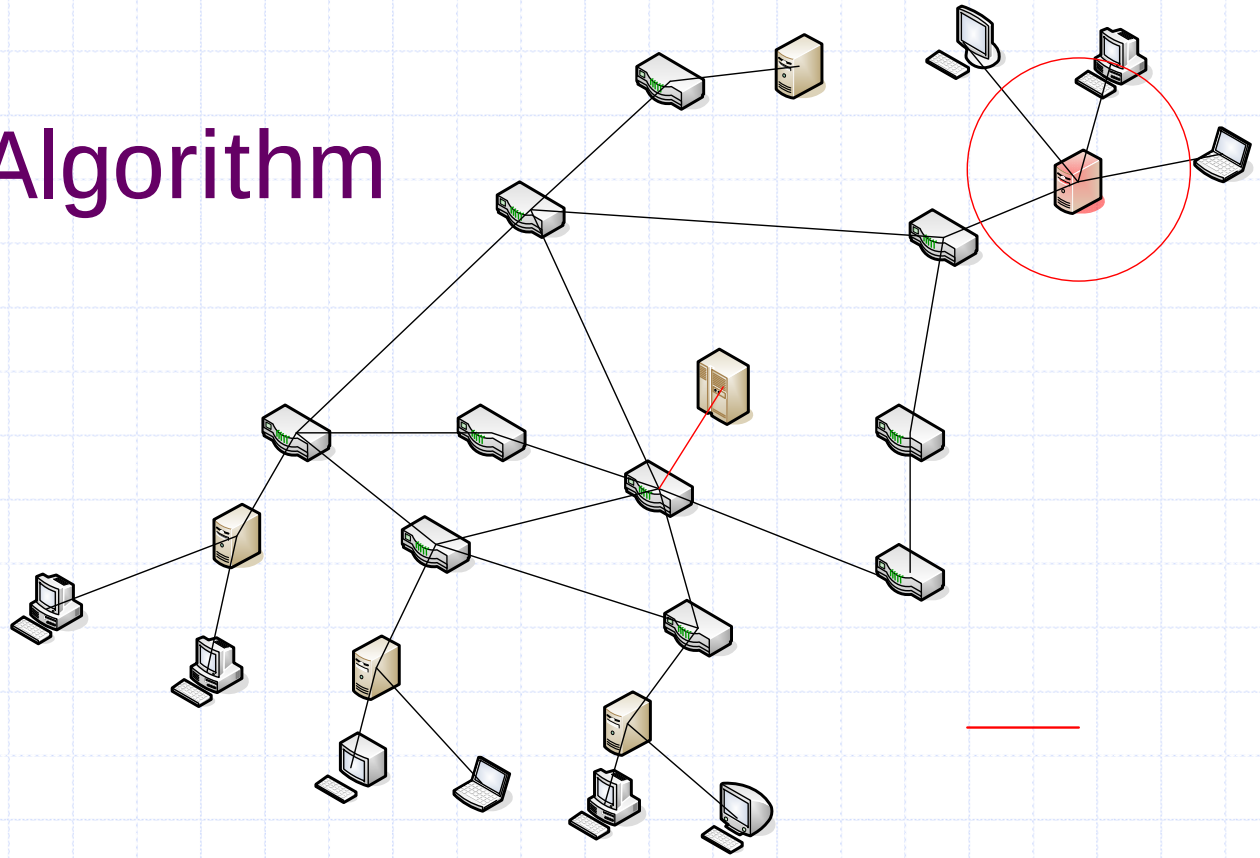


Hot Spot Algorithm



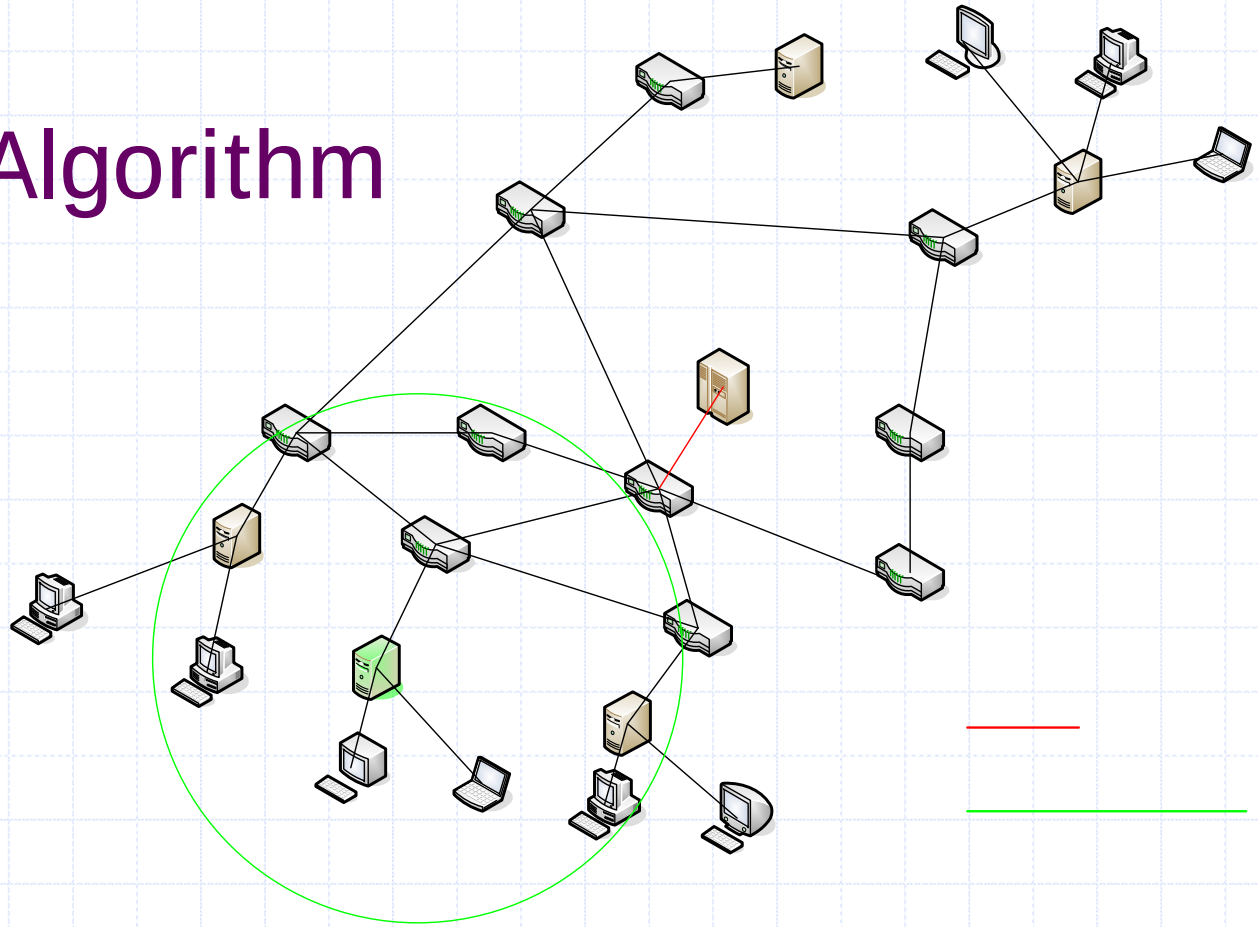
- Places surrogate servers near the clients generating the greatest load.
- Sorts the *N* potential sites according to the amount of traffic generated within their vicinity, and places the surrogate server at the top *K* sites.
- A's vicinity is the circle centered at A with some radius. We change the radius, repeat the algorithm and choose the best one as output
- Time complexity: $N^2 + N\log N + NK \approx O(N^2)$

Hot Spot Algorithm



- Places surrogate servers near the clients generating the greatest load.
- Sorts the *N* potential sites according to the amount of traffic generated within their vicinity, and places the surrogate server at the top *K* sites.
- A's vicinity is the circle centered at A with some radius. We change the radius, repeat the algorithm and choose the best one as output
- Time complexity: $N^2 + N\log N + NK \approx O(N^2)$

Hot Spot Algorithm

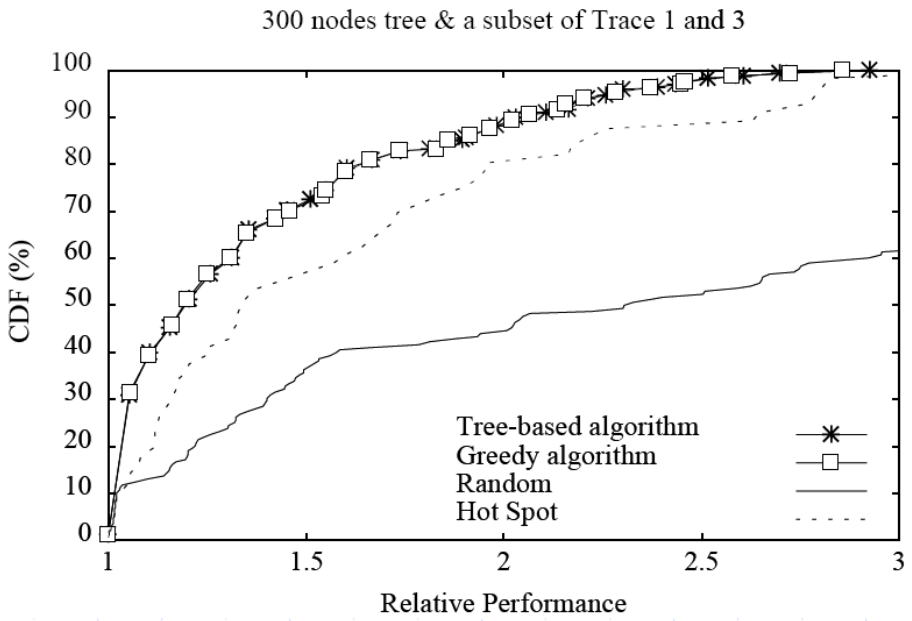


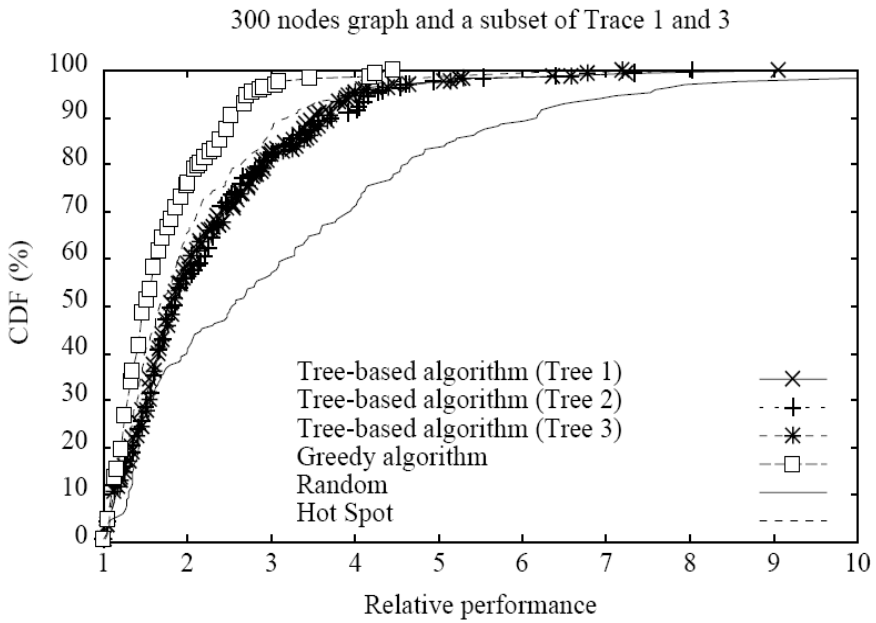
- Places surrogate servers near the clients generating the greatest load.
- Sorts the *N* potential sites according to the amount of traffic generated within their vicinity, and places the surrogate server at the top *K* sites.
- A's vicinity is the circle centered at A with some radius. We change the radius, repeat the algorithm and choose the best one as output
- Time complexity: $N^2 + N\log N + NK \approx O(N^2)$

Random Algorithm

- Chooses *K* sites from *N* places randomly; performs 10 times, and get the best result.
- Time complexity: O(*NK*)

Evaluation Results





- In a tree-structure network, greedy ≈ tree-based > hot spot > random
  - The paper says greedy is slightly better, why?
- In an arbitrary network, greedy > hotpot > tree > random

Server Placement in Real World

- Assumptions hold true?
  - Tree
  - Each client uses a single replica (surrogate server)
- Policy issues

How to Redirect

- URL rewriting
  - Origin server redirects clients to different surrogate servers by rewriting the page's URL links
    - Dynamic
    - Static
  - Bottleneck
- Domain Name System (DNS) redirection
  - DNS server direct requests to CDN

Select a CDN

- From a customer's point of view, which CDN to choose?
  - Cache hit ratio
  - Saved bandwidth
  - Surrogate server utilization
  - Reliability
  - ……

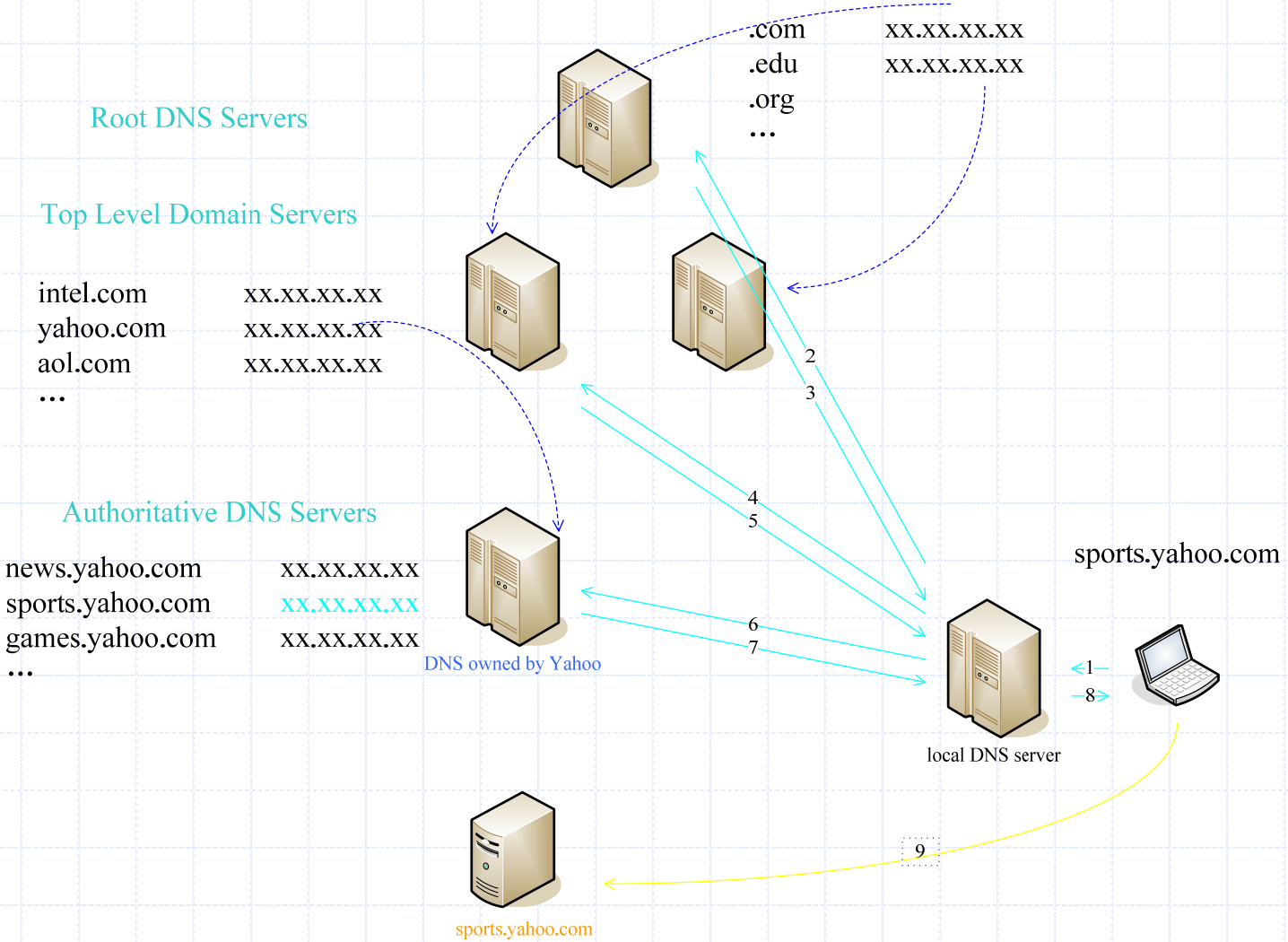
Outline

- Introduction to CDN
- An Industry Example: Akamai
- A Research Example: CDN over Mobile Networks
- Conclusion

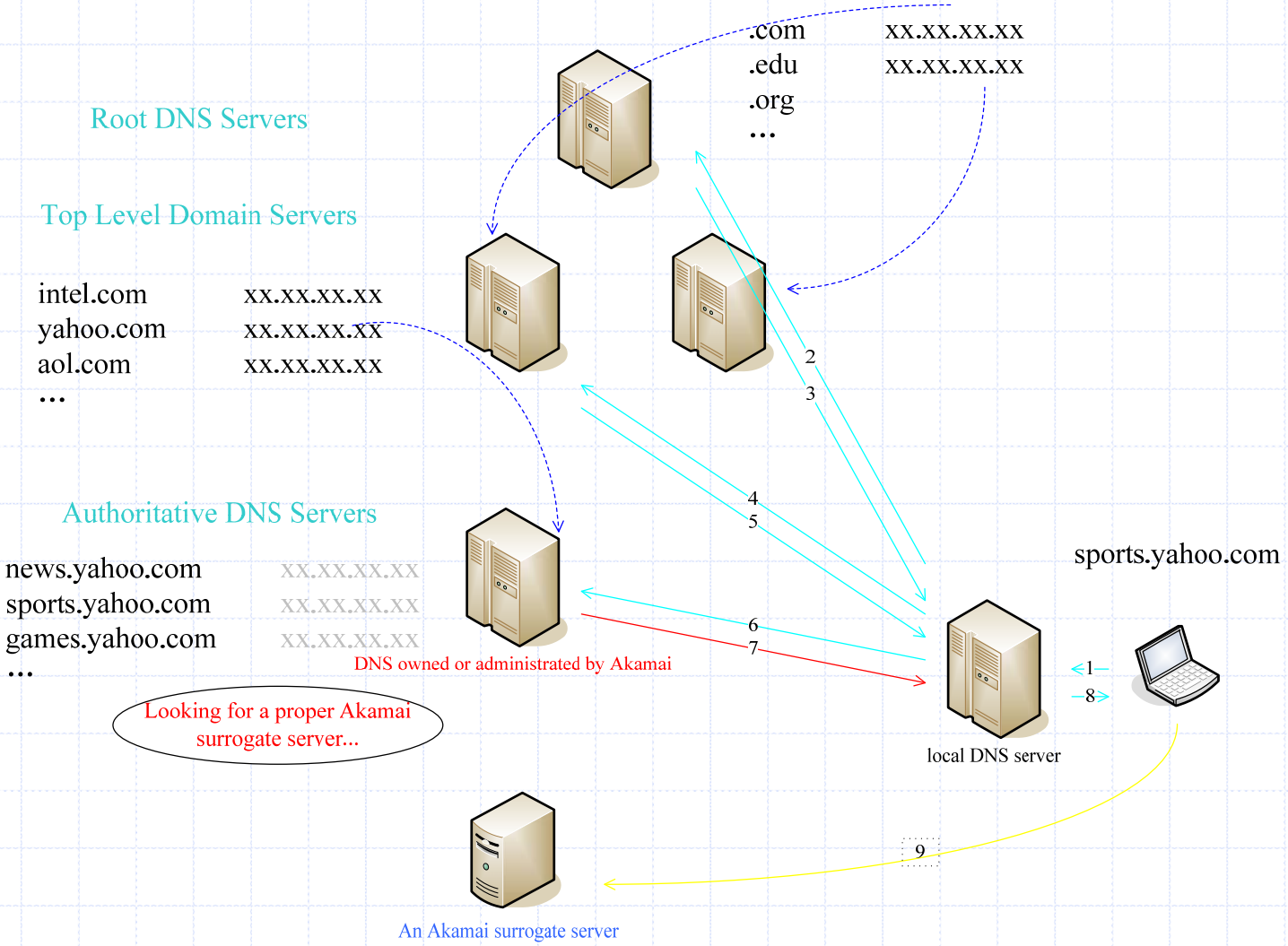
Overview of Akamai

- Launched in 1999
- Over 12,000 servers in 62 countries
- Serves Yahoo!, Apple, AOL.....
- How Akamai works
  - Domain Name System (DNS) redirection

Normal DNS Working Process



How DNS Redirection Works



How DNS Redirection Works (cont.)

- What Akamai does
  - Manage customers' DNS server
  - Pretend to be a normal DNS server
  - But more complex inside
    - Probing
    - Selecting
    - Load balancing
- Example: visit Yahoo via Akamai

Selecting Criteria

- Server must be able to satisfy the request
  - e.g. can handle streaming media?
  - Has the content?
- Server health
- Server load
- Network condition
  - Packet loss rate
  - Bandwidth
- Client location

Different Types of Content

- Static content
  - Lifetimes
  - Special features
- Dynamic content
  - ESI: break a dynamic page into fragments
  - Assemble dynamic pages at surrogate servers
- Streaming media
  - Deliver packets without significant delay or jitter
- Content unreachable
  - Split the TCP connection into two separate connections

Outline

- Introduction to CDN
- An Industry Example: Akamai
- A Research Example: CDN over Mobile Networks
- Conclusion

A Research Example: CDN over MANET [3]

- In a more general sense, CDN refers to
  - any overlay network built for the purpose of facilitating content delivery
- Background:
  - Pervasive computing / ubiquitous computing
    - Goal: make computers work in a more intelligent way
    - Decrease users' intended input
    - Representative applications, *Examples*

---

[3] S. Chen, et al., Application based Distance Measurement for Context Retrieval in Ubiquitous Computing. MobiQuitous 2007, Philadelphia, US.

Hardware Environment

- Sensors, computers, smart phones and PDAs
  - Both fixed and mobile nodes
  - Each node may serve as data producer and data consumer
- We want to improve the data retrieval performance, however, it's impossible to store replicas on every node
  - Limited storage
  - Limited energy
  - Communication costs

Context Clustering

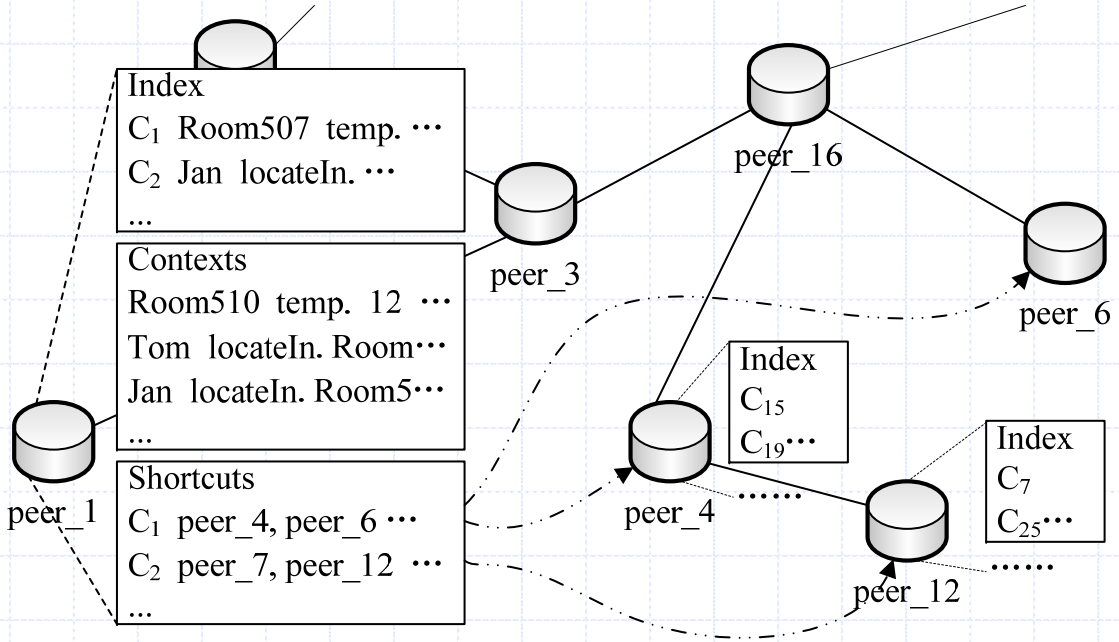
- Different types of data (contexts)
  - Light / sound / temperature / humidity / user's schedule / location...
- Logical distance between contexts
  - Which contexts are often used (queried) by applications simultaneously?

$$\cos(S,T) = \frac{\sum_{j=1}^{q} s_j t_j}{\sqrt{\sum_{j=1}^{q} s_j^2}\sqrt{\sum_{j=1}^{q} t_j^2}}$$

$$D(S,T) = \sin(S,T) = \sqrt{1-\cos^2(S,T)}$$

Replica Placement

- If the logical distance between two types of contexts is lower than a threshold, they cache each other, or shortcuts are built between them



Outline

- Introduction to CDN
- An Industry Example: Akamai
- A Research Example: CDN over Mobile Networks
- Conclusion

Conclusion

- CDN is a overlay network which aims at efficiently delivering content.
- Open issues
  - CDN on P2P networks / mobile networks
  - Authentication/ security issues
  - ...

Thank you