Overlay (and P2P) Networks

Part II

Samu Varjonen

Ashwin Rao

Outline for this lecture

- Internet Indirection Infrastructure
- Content Delivery Networks
  - Akamai
  - Coral CDN
  - Other P2P CDNs

Structured Overlay Networks

Recap of Chord

- Distributed lookup protocol
  - Given a key, map the key to a node
- Assign a unique m-bit key (identifier) to a node
  - Consistent hashing (of IP address) generates keys
- Identifiers (nodes) ordered in a circle module as 2^m
  - Every node has predecessor and successor
- A key k is assigned to a node whose identifier is equal to or follows k the identifier space.
  - k stored on successor(k)
- Routing table (at most m entries on each node)
  - i^{th} entry → first node succeeds node by at least 2^{i-1}

Chord Properties

- Each node responsible for ≈ K/N keys
  - K → total #keys, N → total #nodes
  - When a node joins or leaves the network only O(K/N) keys will be relocated
    - Relocation is local to the node
- Lookups take O(log N) messages
- $O(\log^2 N)$ messages required to re-establish routing invariants after join/leave
  - Each node's successor is correctly maintained
  - For every key (k), the node responsible for k is successor(k)

Packet's Perspective of Internet Services

- **Unicast**: One fixed source to one fixed destination
- **Broadcast:** One source to all destinations
- **Multicast**: One fixed source to multiple destinations who are part of a group
- **Anycast**: One source to exactly one destination who is a member of a group

Internet Services using Unicast, Broadcast, Multicast, and Anycast are built over the **point-to-point abstraction**

Can we use another abstraction?

Rendezvous-based Communication

- Source sends packets to a *logical identifier*. Receivers express interest in packets sent to an identifier
- Packet is a pair **(id, data)**
  - id → host/object/session/...

- Receivers use triggers **(id, addr)** to express interest
  - Forward packet with identifier **(id)** to receiver with IP address **(addr)**
- Packet sent to receivers if the
  - the interest ($\mathbf{id}_T$) from receiver is a longest prefix match
  - the match is longer than matching threshold k (k < m)

Abstraction decouples the act of sending from the act of receiving

API to Implement Indirection

- SendPacket (p)
- InsertTrigger(t)
- RemoveTrigger(t)

API Implemented in an i3 Overlay Network

- Overlay Consists of i3 Servers
  - Store Triggers
  - Forward packets using IP between i3 nodes and end-hosts
    - *Packets are not stored at the Servers*
  - Implemented using Chord (or any other DHT)

Benefits of i3

- Support for mobility
  - On moving to new address (addr'), receiver sends new trigger (id, addr')
  - Receivers periodically refresh triggers
- Multicast
  - Source is agnostic to the set of receivers
  - Receiver agnostic to the set of sources
  - Trigger chains can be used to minimize triggers
- Anycast
  - Id contains a common prefix component and a **suffix**
- Anonymity
- Service Composition
  - Stacked identifiers

Limitations of Web Proxies
(Caching)

- Inability of cache all objects/content
  - Dynamic Data
  - Encrypted data
- Server Side Analytics
  - Hit Metering, User Demographics, etc.
- Scalability
  - Inability to support **flash crowds**
- …

Content Delivery Networks

- Role
  - Redirect content requests to an 'optimal site'
  - Cache and Serve content from 'optimal site'
  - Export logs and other information to origin servers
- Redirection mechanism
  - DNS redirection
  - URL rewriting

Critical Issues in Deploying CDNs

- Servers Placement
  - Where to place the servers?
  - How many in each location?
- Content Selection
  - Which content to distribute in CDNs?
- Content Replication
  - Proactive push from origin server
  - Cooperative vs Uncooperative Pulls
- Pricing

George Pallis et al. **"Insight and perspectives for content delivery networks."** In Communicatoins of ACM 49, 1 (January 2006),

Server Placement Problem

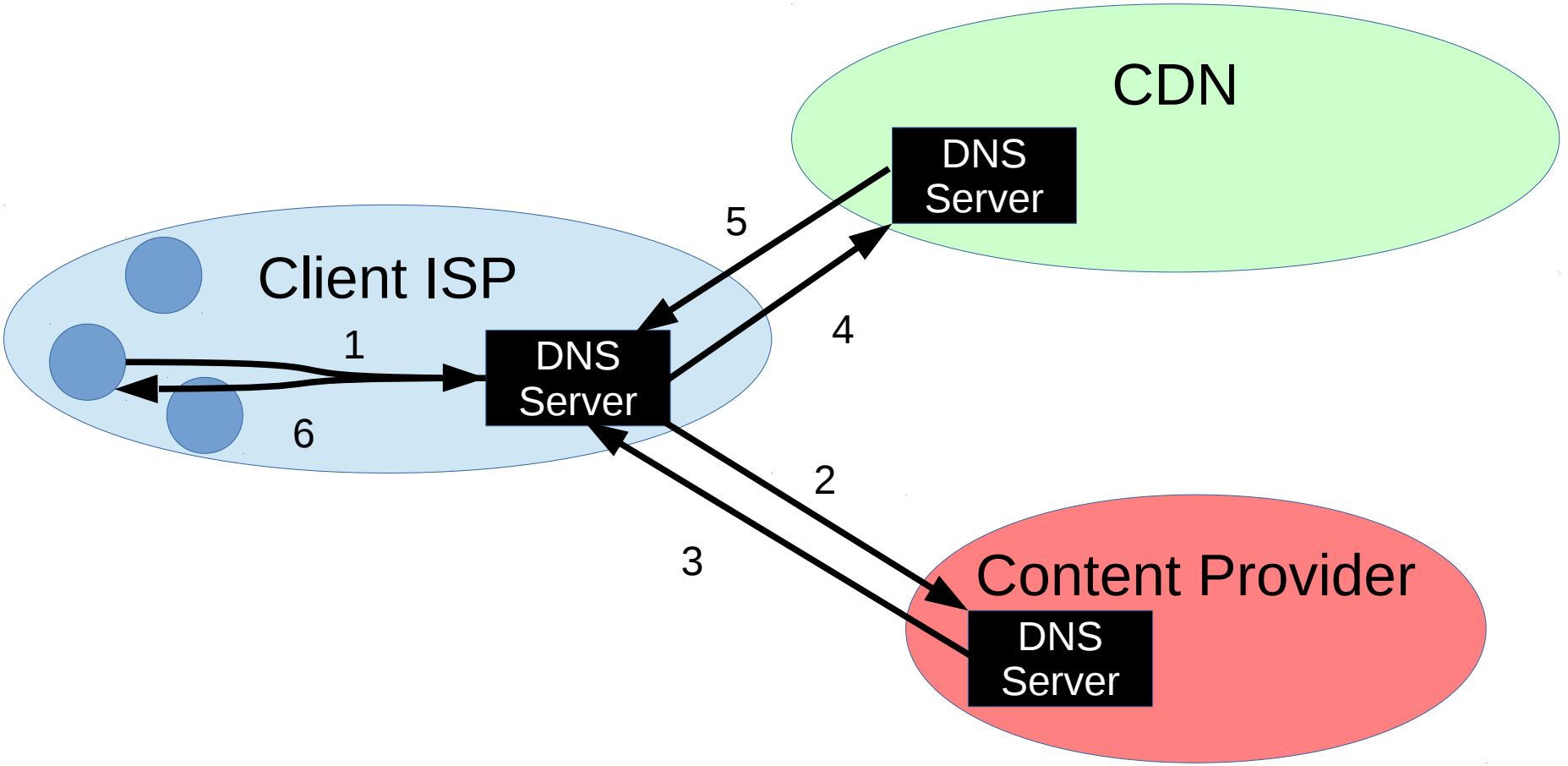
Given N possible locations at edge of the Internet, we are able to place K (K<N) surrogate servers, how to place them to minimize the total cost?

- Minimum K-median problem
  - Given N points we need to select K centers
  - Assign each input point j to a center 'closest' to it
  - Minimize the sum of distances between each j and its center
    - NP-Hard

Redirection Techniques

- Routing Strategy
  - Anycast
  - Load Balancing
- Application specific selection
  - HTTP redirection
- Naming based redirection
  - DNS

DNS Based Redirection

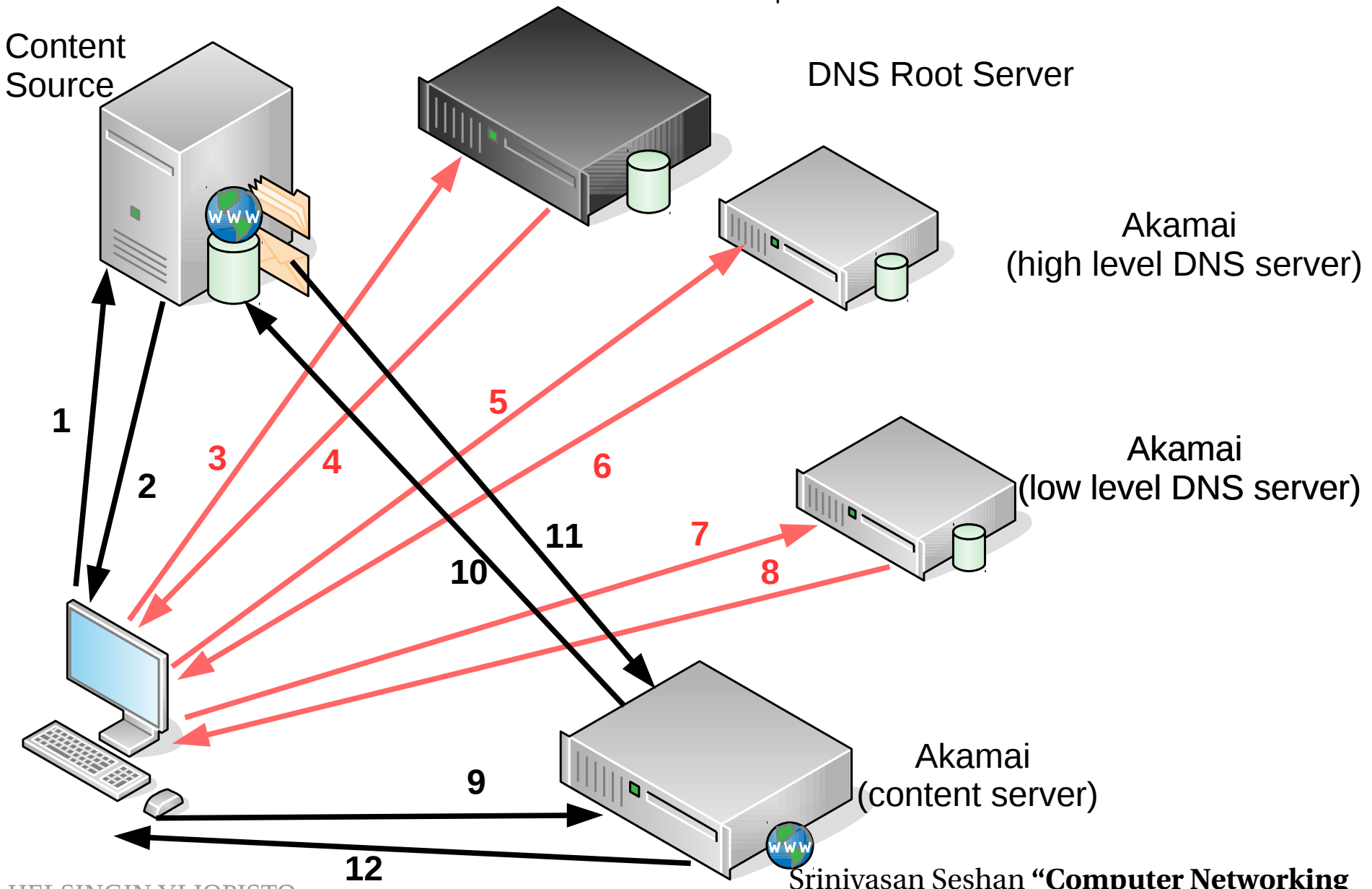


Akamai CDN
(overview)

- Client requests content from Original Server
  - URLs for content in CDN modified in the original response
- Client resolves <content>.<akamai host> name
- Server from the region (best server) chosen
- Client fetches content from akamai server

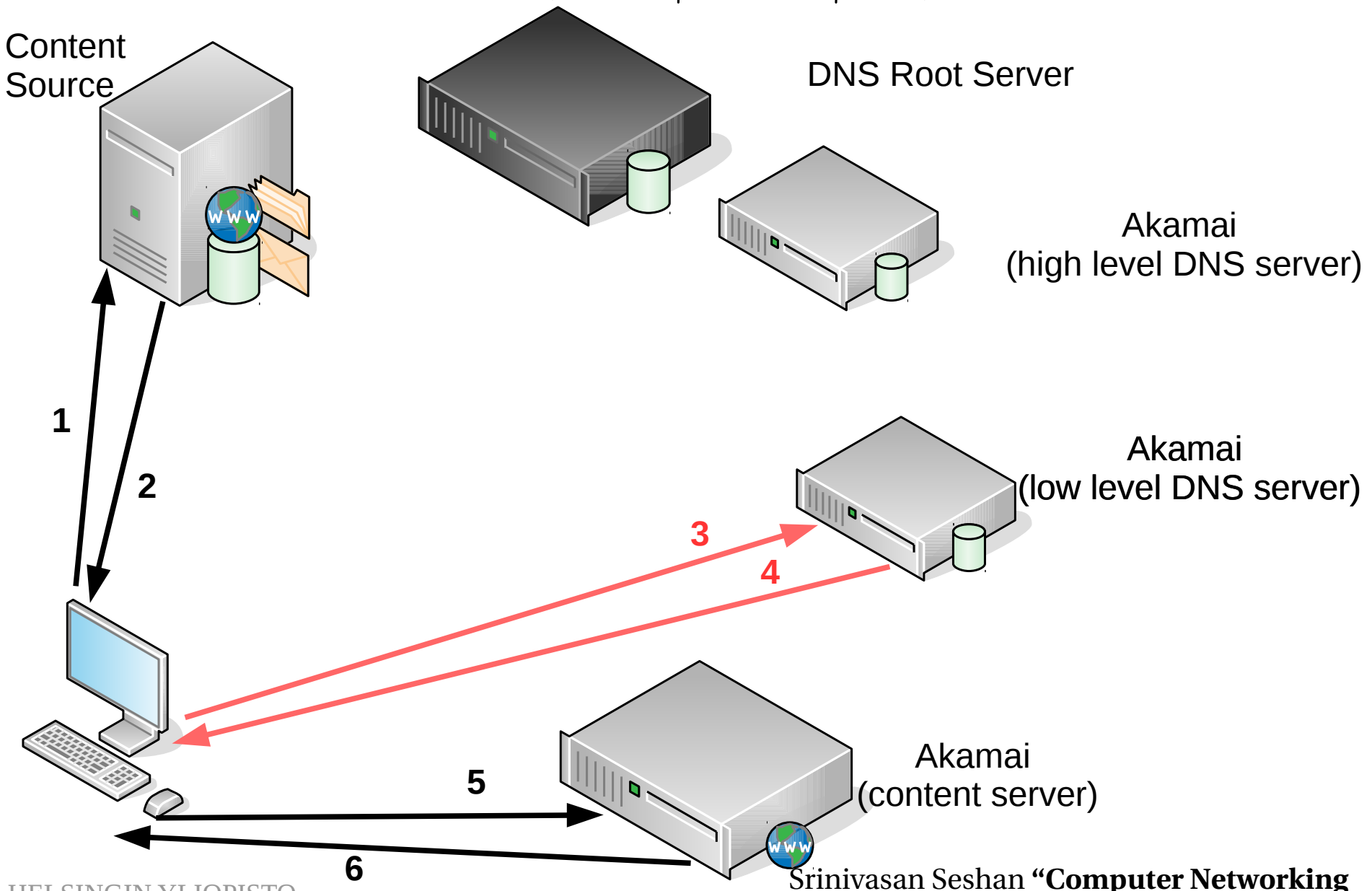
Akamai

(initial request)



Akamai

(subsequent request)

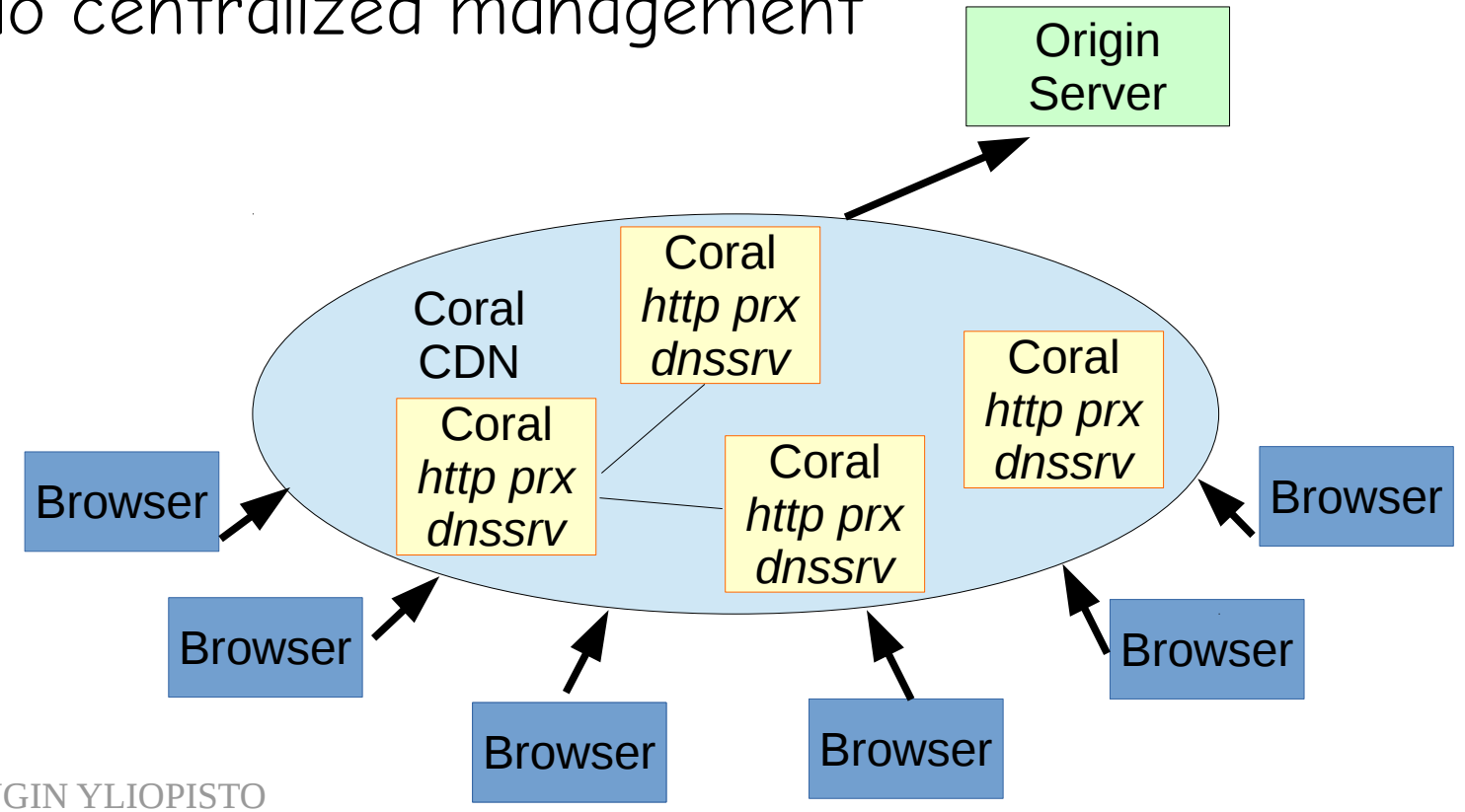


Coral Objectives

- Pool resources to dissipate Flash Crowds
- Work with unmodified clients
- Fetch content only once from Origin
- No centralized management

Coral Objectives

- Pool resources to dissipate Flash Crowds
- Work with unmodified clients
- Fetch content only once from Origin
- No centralized management



Using Coral

- Origin Server rewrites URLs
  - abc.com → abc.com.coralhost:coralport
  - Redirect clients to Coral server
- Coral CDN Components
  - DNS server
    - Given address of resolver used by the client, return the address of proxy near the client
  - HTTP proxy
    - Given the URL find nearest proxy that has content
    - Cache the content (DHT)
  - Distributed Sloppy Hash Table (DSHT)
    - No load-balancing & content locality support in Basic DHTs (Chord)

Coral System Overview

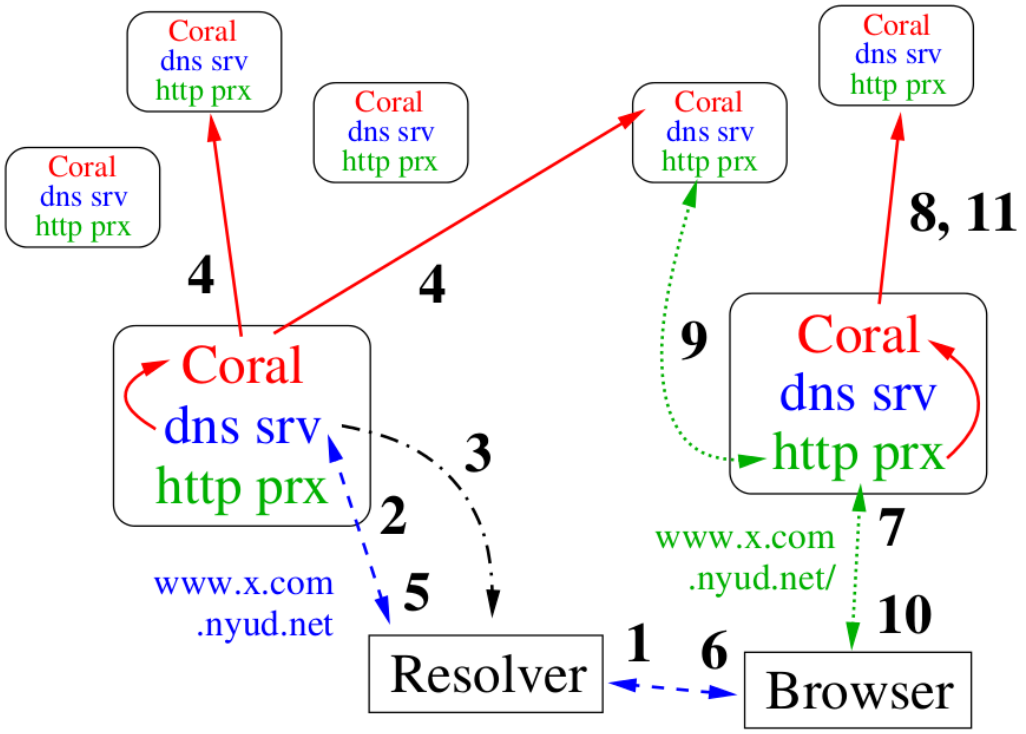


Figure 1: Using CoralCDN, the steps involved in resolving a Coralized URL and returning the corresponding file, per Section 2.2. Rounded boxes represent CoralCDN nodes running Coral, DNS, and HTTP servers. Solid arrows correspond to Coral RPCs, dashed arrows to DNS traffic, dotted-dashed arrows to network probes, and dotted arrows to HTTP traffic.

Freedman, Michael et al. **"Democratizing Content Publication with Coral."** In NSDI 2004.

Hierarchical Indexing

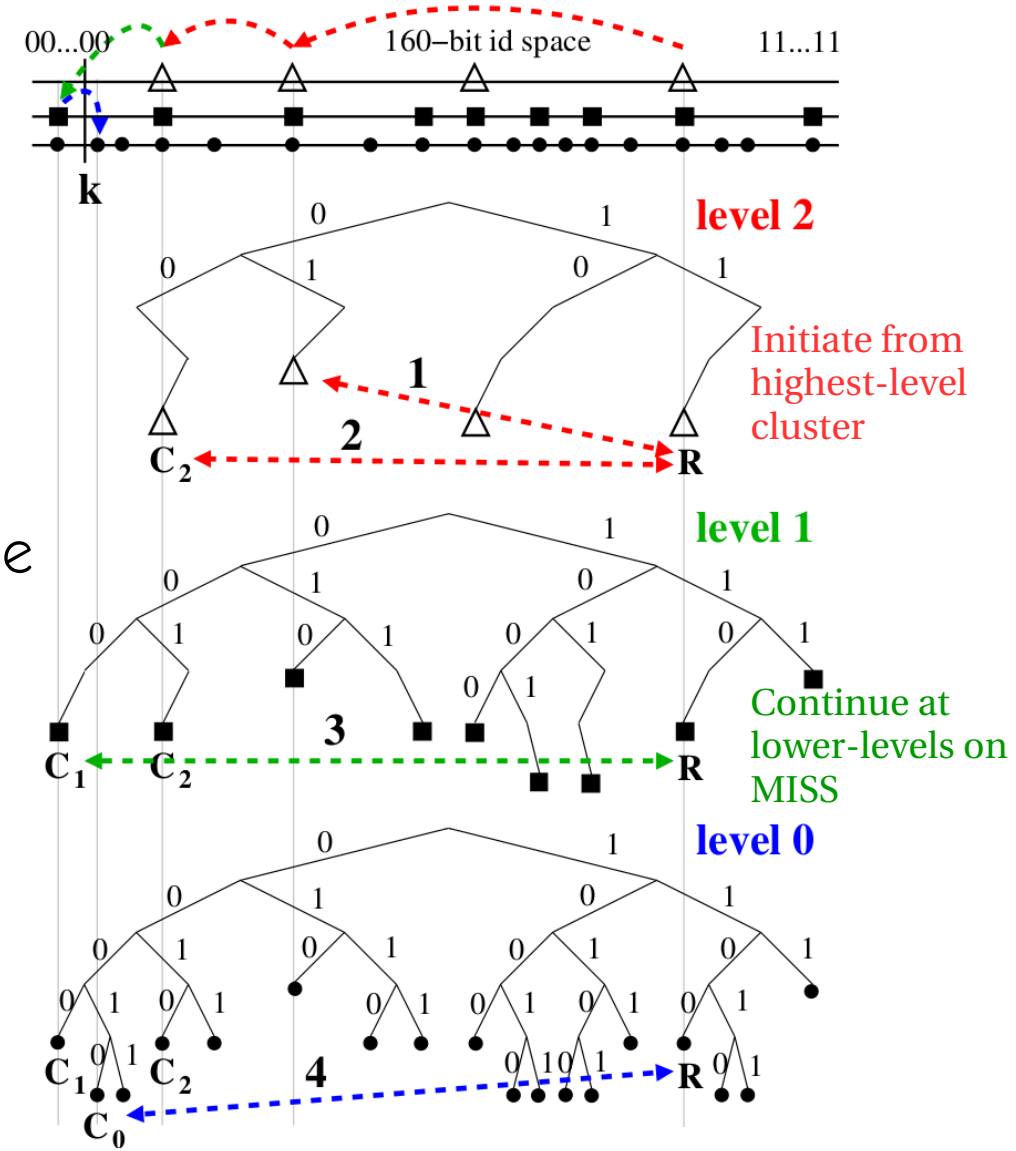
- Diameter, Clusters, Levels
  - Each Coral Node part of several DSHTs called clusters
  - Each cluster characterized by max RTT (diameter)
  - Fixed hierarchy of diameters called levels
  - Group of nodes can form a level-i cluster if the pair-wise RTT less than threshold for level-i
    - Paper uses 3 (levels): 20ms (2), 60ms (1), ∞ (0)

Hierarchical Indexing

- Diameter, Clusters, Levels
  - Each Coral Node part of several DSHTs called clusters
  - Each cluster characterized by max RTT (diameter)
  - Fixed hierarchy of diameters called levels
  - Group of nodes can form a level-i cluster if the pair-wise RTT less than threshold for level-i
    - Paper uses 3 (levels): 20ms (2), 60ms (1), ∞ (0)
- SHA-1 for Coral Keys and Node-Ids
- Bitwise XOR is distance (Kademlia)
  - Longer matching prefix numerically closer
  - Key stored at node having ID "close" to key

Routing and Sloppy Storage

- Routing
  - Routing table size logarithmic in total number of nodes
- Sloppy Storage
  - Cache key/value pairs at nodes whose IDs are close to the key being referenced
  - Reduces hot-spot congestion for popular content

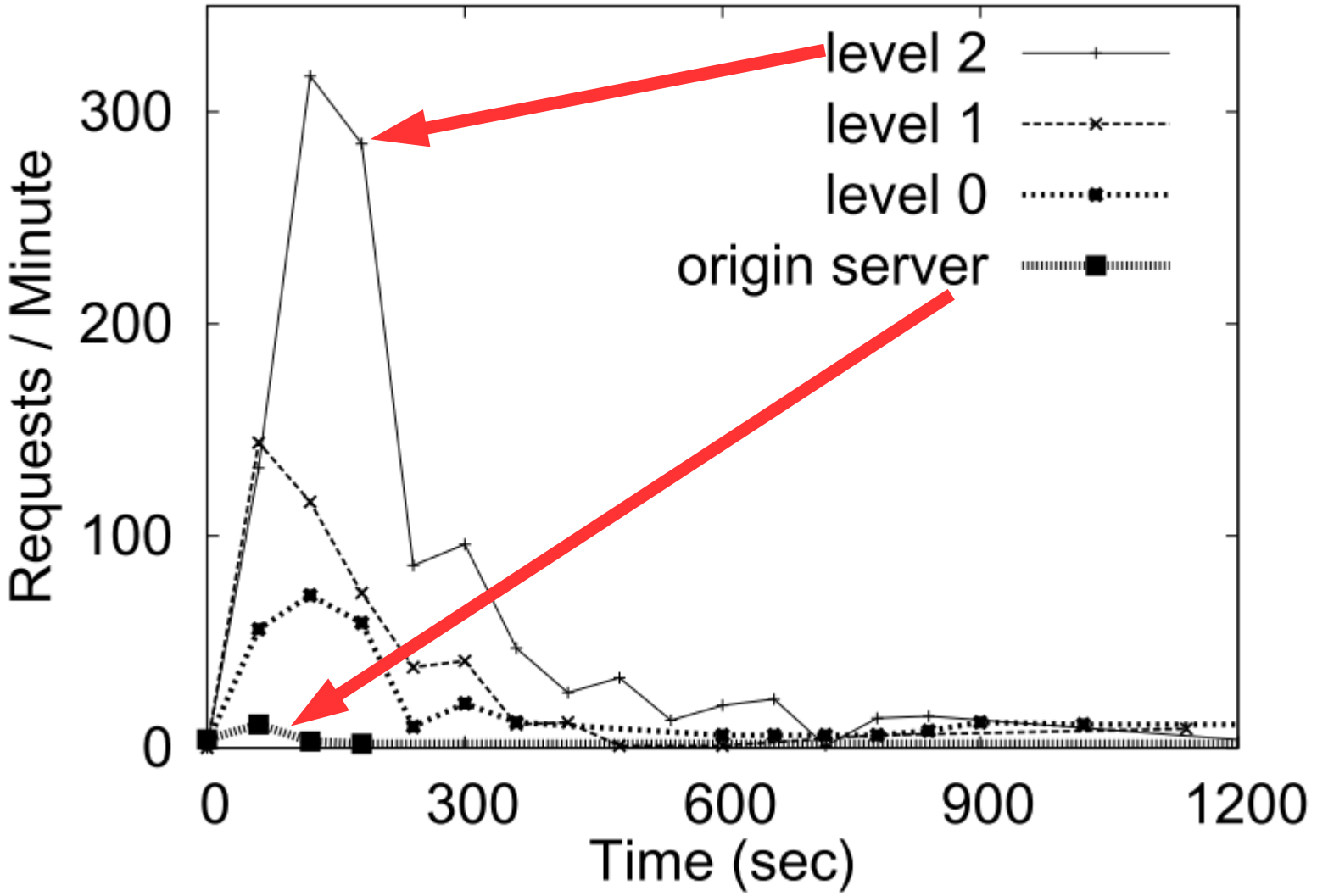


Freedman, Michael et al. **"Democratizing Content Publication with Coral."** In NSDI 2004.

Coral Implemented on PlanetLab

Global Research Network
As of Feb 2014, PlanetLab has 1181 nodes at 567 sites

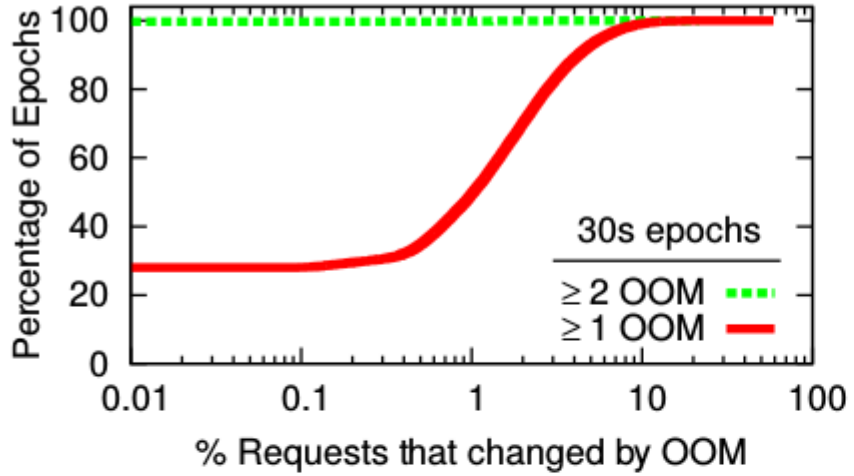
Reduction in Server Load

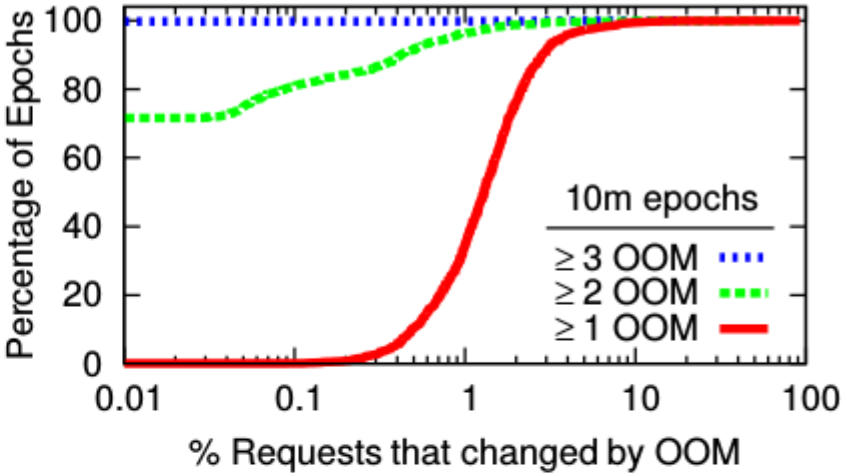


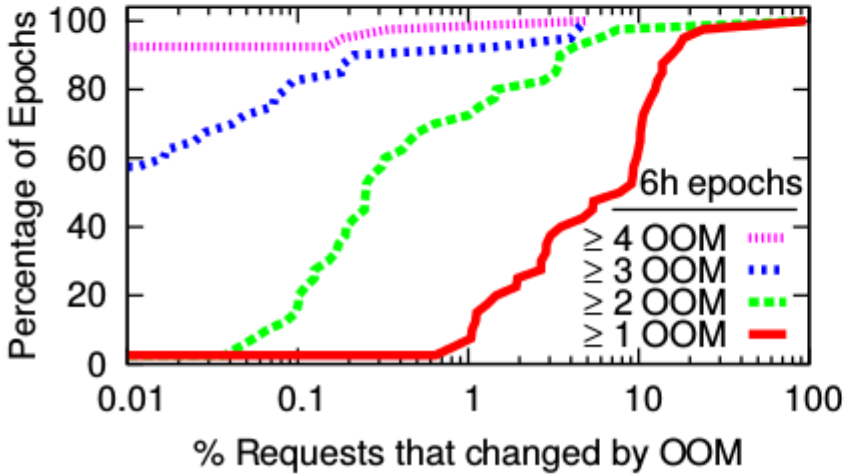
Freedman, Michael et al. **"Democratizing Content Publication with Coral."** In NSDI 2004.

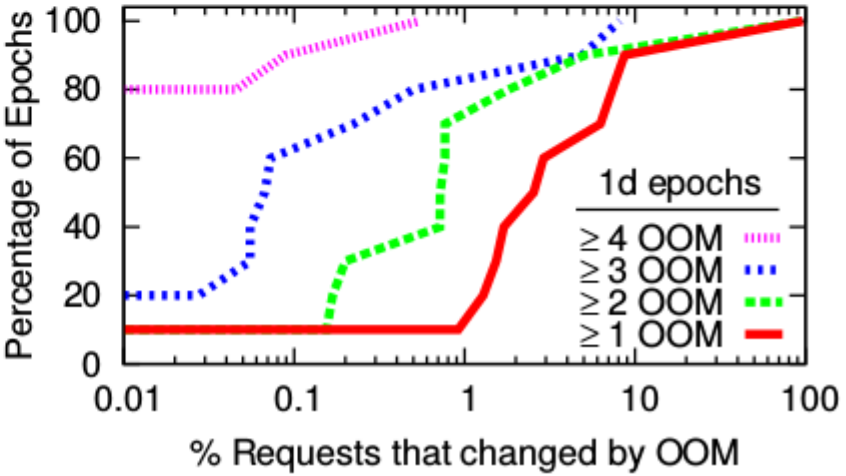
Dynamics of Flash Crowds

28% of 30s epochs have no domains with a ≥ 1 OOM rate increase









Freedman, Michael. **"Experiences with CoralCDN: A Five-Year Operational View."** In NSDI 2010.

Insights from 5 year Deployment

- A large majority of its traffic does not require any cooperative caching
  - Handling of flash crowds relies on cooperative caching
- Flash Crowds
  - Small fraction of CoralCDN's domains experience large rate increases within short time periods
  - Flash crowd domains' traffic accounts for a small fraction of the total requests
  - Request rate increases very rarely occur on the order of seconds
- Content delivery via untrusted nodes requires the HTTP protocol to support end-to-end signatures for content integrity

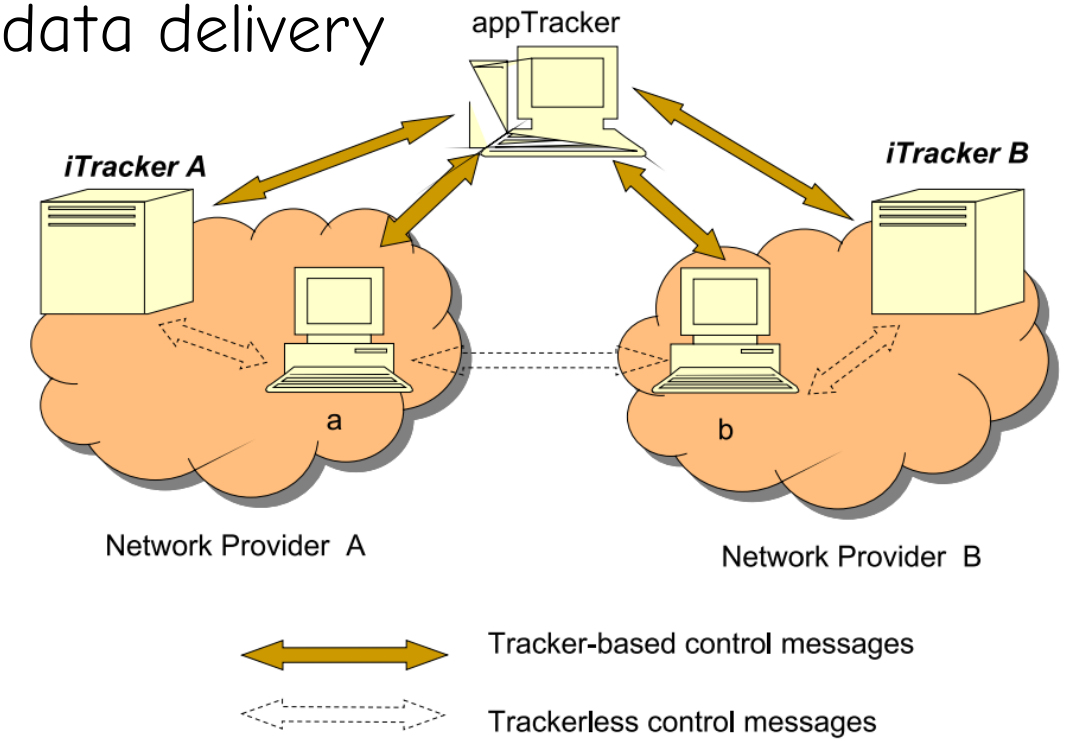
Freedman, Michael. **"Experiences with CoralCDN: A Five-Year Operational View."** In NSDI 2010.

P4P (Provider Portals for Applications)

- P2P applications may be oblivious to underlying network
  - Lot of inter-domain traffic (Karagiannis et al. 2005)
- Approaches to address this problem
  - ISP approaches
    - Block P2P, Rate-limit P2P, Cache content, etc.
  - P2P approaches
    - Locality (Ono Project)

itracker of P4P

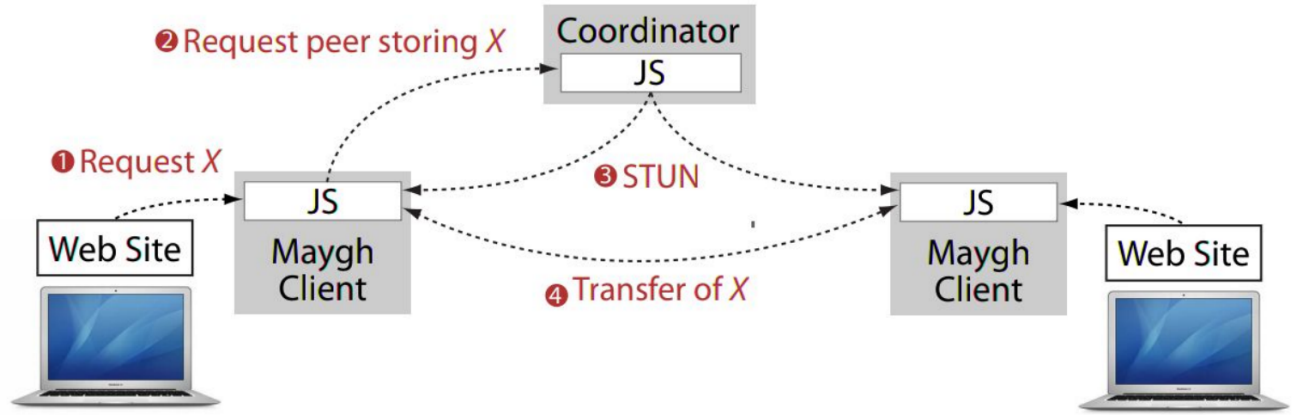
- Network provider runs an iTracker
- itracker used by ISPs to provides additional information regarding network topology
  - P2P networks may choose to utilize to optimize network data delivery



Haiyong Xie et al. "P4p: provider portal for applications." In SIGCOMM 2008

Maygh P2P CDN

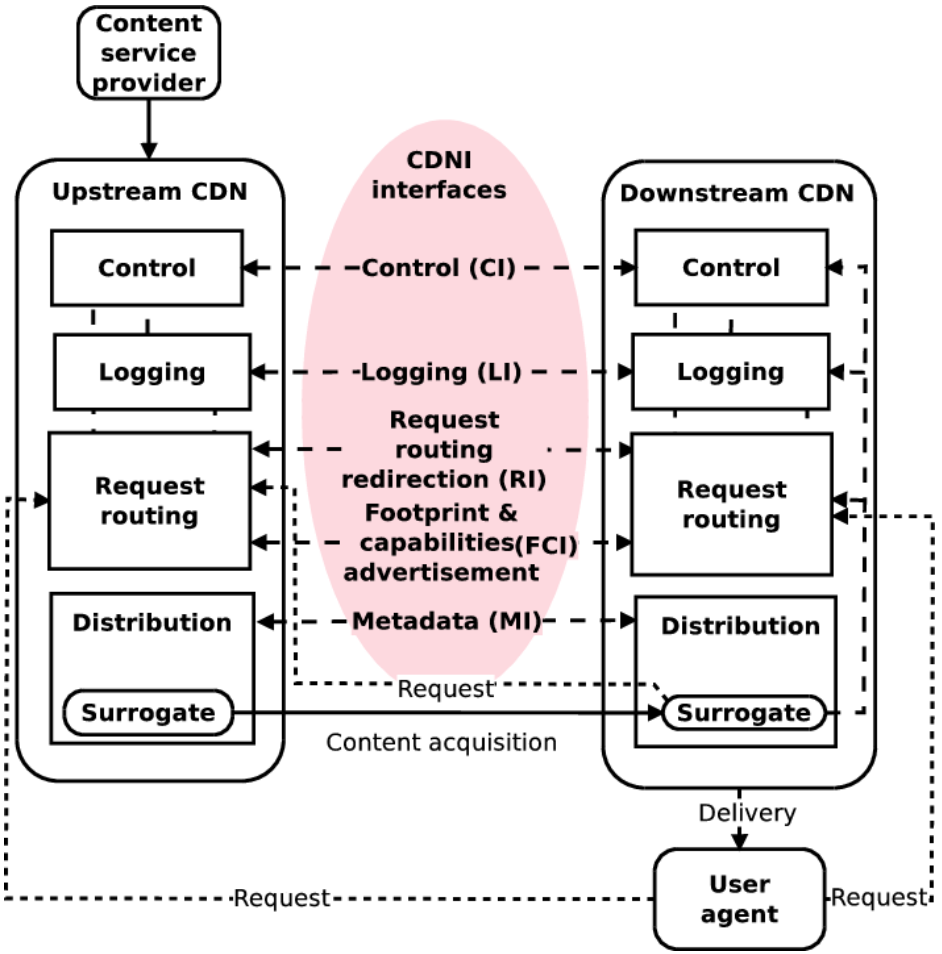
- P2P CDN on Browser
  - Leverage on WebSockets, WebRTC, WebStorage API



Liang Zhang et al. **"Maygh: building a CDN from client web browsers."** In EuroSys '13.

CDNI
(CDN Interconnection)

- Leverage collective CDN footprint
  - One CDN to reuse resources of another CDN provider
  - ISPs can deploy their own CDNs



Niven-Jenkins etal. **"Content distribution network interconnection (CDNI) problem statement."** RFC 6707 (2012).

Sources for these slides

- Sasu Tarkoma ”**Overlay and P2P Networks**”, 2015
- Karthik Lakshminarayanan **”Internet Indirection Infrastructure”**
- Srinivasan Seshan **“Computer Networking Caching, CDN, Consistent Hashing, P2P”**