Applications to Network Measurement

Outline

- Web measurement motivation
- Challenges of web measurement
- Web measurement tools
- Current web measurements
  - Web properties
  - Web traffic data gathering and analysis
  - Web performance
  - Web applications

Motivation

- Web is the single most popular Internet application. Measurement can be very useful.

Table 7.4 Web properties of interest to measure

Class	Measured property	Why measured
High-level characterization	Fraction of traffic Number of entities	Overall importance Statistical, sampling
Location	Presence of Web entities	Performance/Customer location
Configuration	Software/hardware configuration	Performance/handling load
User workload models	Access pattern	Modeling Web phenomena
Traffic Properties	Caching, flash crowds	Performance
Application demands	Impact on network	Protocol improvement
Performance	Web components performance	Maintaining site popularity

Challenges to measurement

- Hidden Data
  - Much of the traffic is intra-net and inaccessible.
  - Access to remote server data, even old logs is often unavailable.
  - From the server end, information about the clients (e.g. connection bandwidth) is obscured.
- Hidden layers
  - Measuring the in flight packets is much harder than measuring the server response time, so the protocol and network layers are harder to measure.
- Hidden entities
  - The web involves proxies, HTTP and TCP redirectors

Tools:Sampling and DNS

- Sampling traffic (e.g. netflow) can help determine the fraction of HTTP traffic.
- Examine DNS records. Well know sites are more likely to be looked up often.

Tools: Server logs

- From a web server perspective, you can examine the server logs.
- However, there are some challenges here:
  - Web crawlers
  - Clients hidden behind proxies

Tools: Surveys

- Estimating the number of web servers can be done via surveys.
- Users can download a tool bar and rank sites.

Tools: Locating servers

- We might assume that the servers for a site would be in a fixed geographical location.
- However:
  - Servers can be mirrored in different locations
  - Several businesses can use the same server farm to increase utilization.

Tools: Web crawling

Tools: Web performance

- Approaches:
  - Measuring a particular web site’s latency and availability form a number of client perspectives.
  - Examining different latency components such as DNS, TCP or HTTP differences, and CDNs
  - Global measurements of the web to examine protocol compliance, ensure reduction of outages and look at the dark site of the web.
- A variety of companies offer such services:
  - Keynote, Akamai, etc.

Tools: Role of Network aware clustering

- We can cluster groups of IP addresses using BGP routing table snapshots and longest prefix matching.
- This clustering allows for better analysis of server logs.

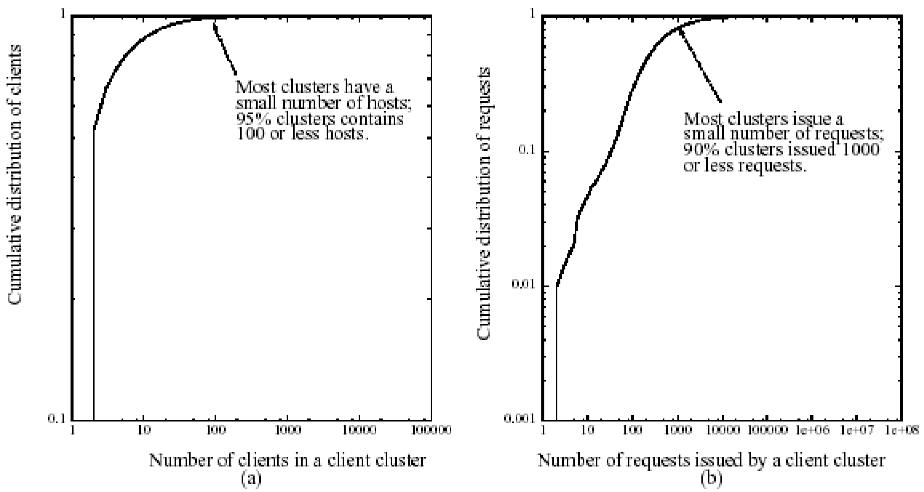
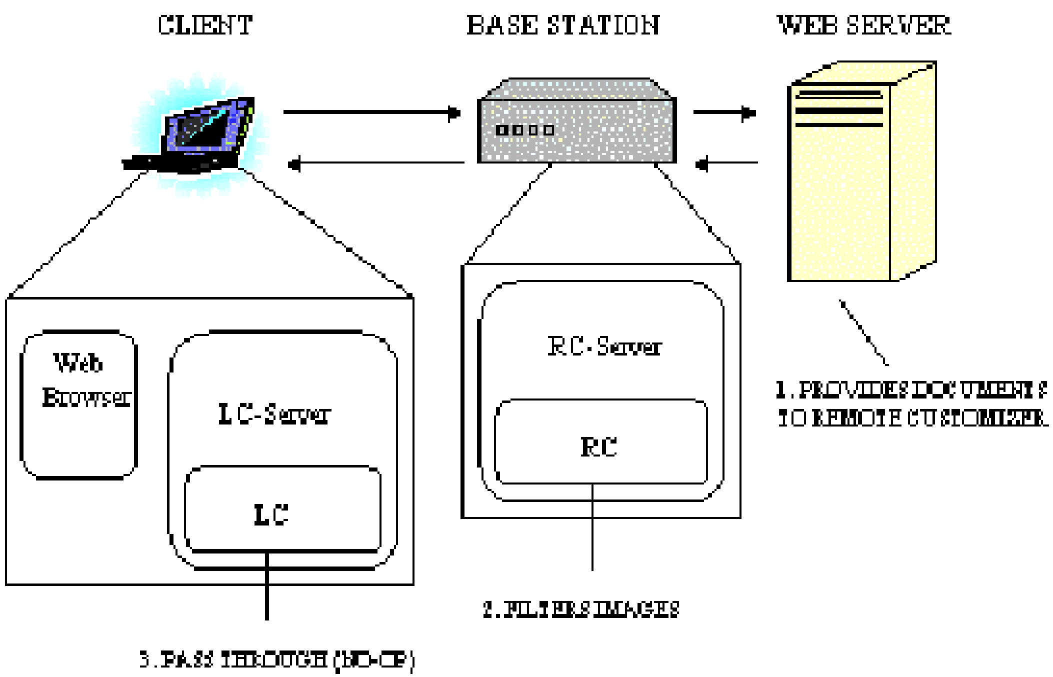


Figure 3: The cumulative distribution of clients and requests in a client cluster for the Nagano server log (y axis is in log scale): (a) is the cumulative distribution of number of clients in client clusters; (b) is the cumulative distribution of number of requests issued from within client clusters.

Balachander Krishnamurthy and Jia Wang. On Network-Aware Clustering of Web Clients. *In Proceedings of ACM Sigcomm,* August 2000.

Tools: Handling mobile clients

Figure 2. An Image-Filtering Customizer.



Jesse Steinberg and Joseph Pasquale. A Web Middleware Architecture for Dynamic Customization of Content for Wireless Clients. In *Proceedings of the World Wide Web Conference*, May 2002.

Tools: Handling mobile clients

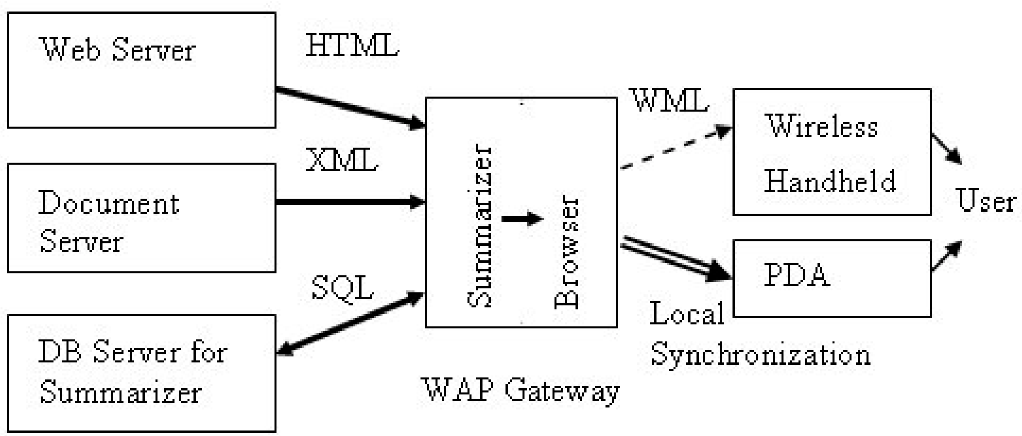


Figure 3. Document Browsing with Summarizer on WAP

Christopher C. Yang and Fu Lee Wang. Fractal Summarization for Mobile Devices to Access Large Documents on the Web. In Proceedings of the World Wide Web Conference, May 2003.

Tools: Handling mobile clients

- Mobile web use (e.g. PDA's and cell phones) continues to grow.
- Similar methods:
  - Server logs of mobile content providers
  - Lab experiments (e.g emulate mobile devices, induce packet loss)
  - Wide-area experiments

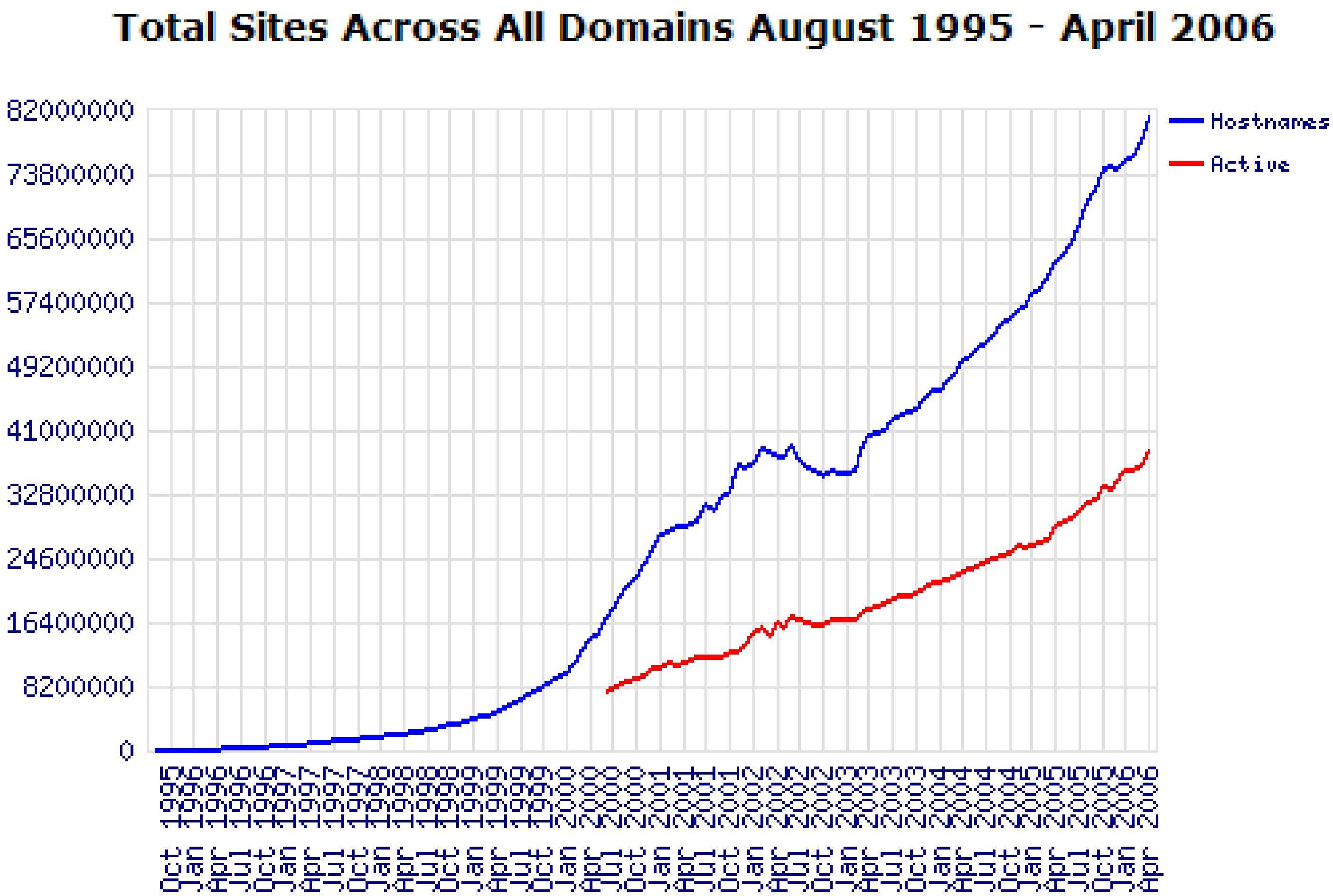
State of the Art

- Four main parts of Web Measurement:
  - High level characterization (properties)
  - Traffic gathering and analysis
  - Performance issues (CDNs, client connectivity, compliance)
  - Applications (searching, flash crowds, blogs)

Web properties: high level

- The number of Web sites numbers in the tens of millions. Popular search engines index billions of web pages, and exclude private Intranets.
- There has been a shift from Web, to P2P and now to games in the traffic patterns of the Internet.
- Monthly surveys by sites like Netcraft have shown around a million new sites a month.
- Estimates in the fall of 2004 showed 60 million web sites, the vast majority have little or no traffic compared to the top few hundred.

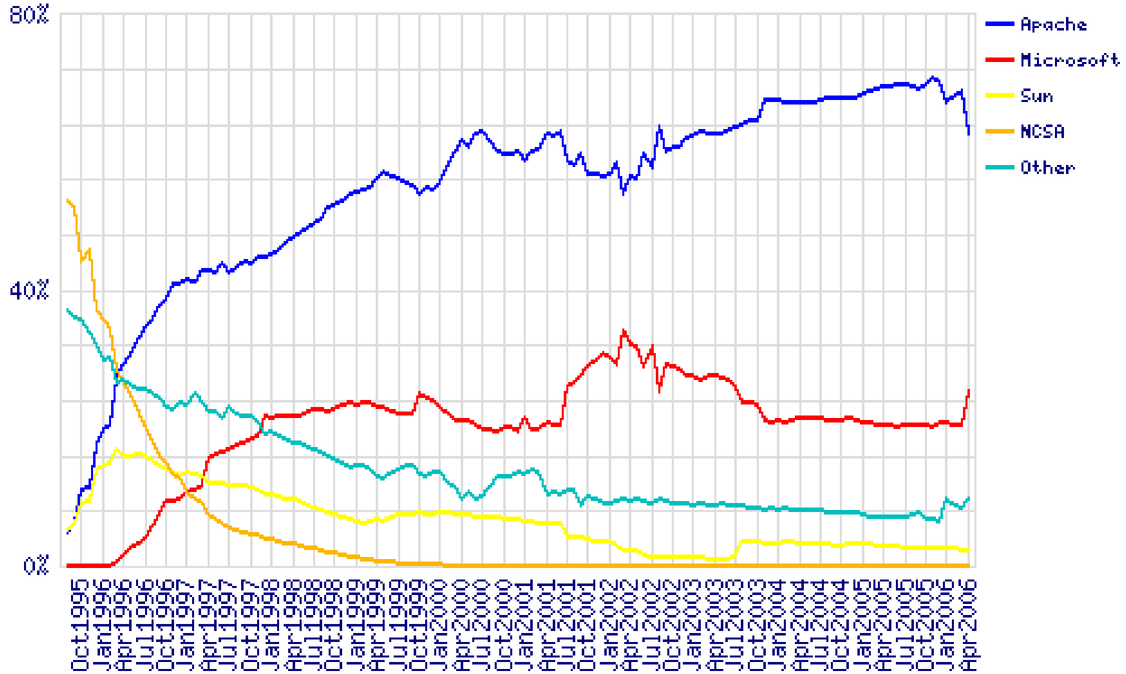
Web Properties: High level



Netcraft survey. (news.netcraft.com)

Web Properties: High Level

Market Share for Top Servers Across All Domains August 1995 - April 2006



Top Developers

Developer	March 2006	Percent	April 2006	Percent	Change
Apache	53287298	68.70	50588433	62.72	-5.98
Microsoft	15912427	20.51	20343656	25.22	4.71
Sun	1881587	2.43	1907503	2.36	-0.07
Zeus	574607	0.74	563381	0.70	-0.04

Netcraft survey. (news.netcraft.com)

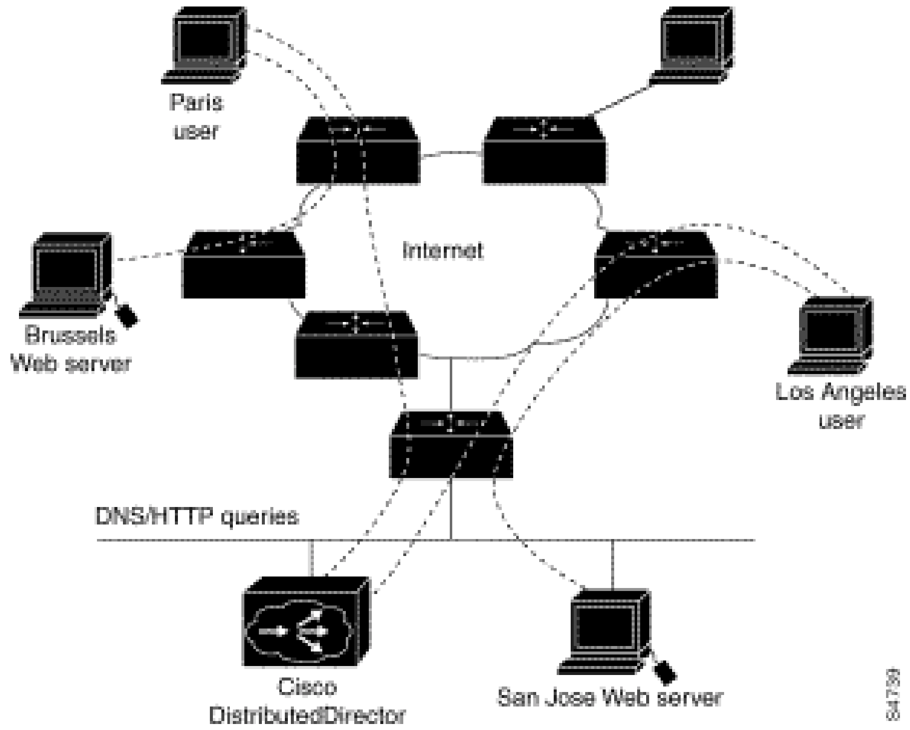
Web properties: Location

- Steadily number of users are in Asian countries such as China and India.
- The fraction of web content from the US and Europe is falling.

Web properties: Configuration

- Popular sites use a variety of techniques to improve server performance:
  - Distribute servers geographically (e.g. 3 world cup servers in the U.S., 1 in France)
  - Use a reverse proxy to cache common requests.

Figure 10-10: Cisco DistributedDirector



http://www.alliancedatacom.com/manufacturers/cisco-systems/content_delivery/distributed_director.asp

Web properties: User workload Models

- We measure user workload by looking at:
  - the duration of HTTP connections
  - request and response sizes,
  - unique number of IP addresses contacting a given Web site
  - number of distinct sites accessed by a client population, number
  - frequency of accesses of individual resources at a given Web site
  - distribution of request methods and response codes

Web properties: Traffic perspective

- Redirector devices at the edge of an ISP network can serve web pages from a cache
- These traditional caches are still sold.
- Reduction in cache hit rates have prompted companies (e.g. NetScaler, Redline) to integrate caching with other services.

Web Traffic: Software Aid

- In order to study the web traffic, a large number of geographically separate measurements need to be repeatedly done.
- httperf:
  - Sends HTTP requests and processes responses
  - Simulates workload
  - Gathers statistics

Web Traffic: Software Aid (2)

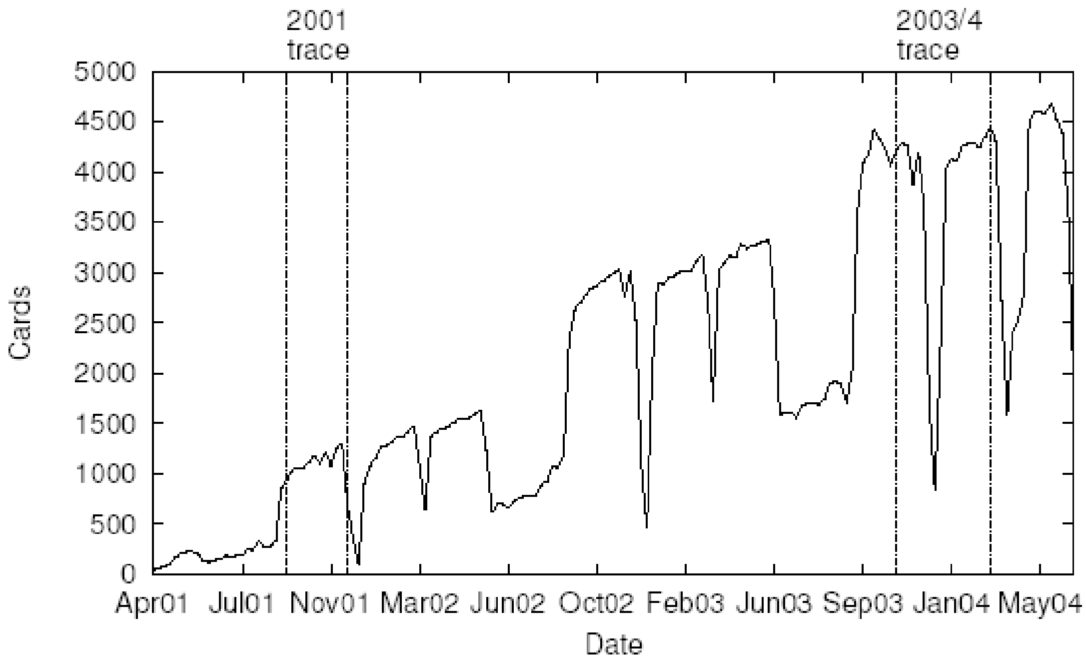
- wget
  - Fetches a large number of pages located at a root node.
  - Can fetch all the pages up to a certain “level” according to links
- Mercator (a personalized crawler)
  - Uses a seed page and then does breadth-first search on the links to find pages.

Web Traffic: Software Aid (3)

- Detailed study in 2000 of 33 million requests from over 50,000 wireless and PDA users.
  - Top 1% of notifications responsible for 60% of content.
  - Notification messages had Zipf-like distribution
  - For popularity: 0.5% of URLs were accessed 90% of the time.
- In another study:
  - Threefold increase in average daily traffic per wireless card between Fall 2003 and Winter 2004

Web Traffic: Wireless Users

Number of active cards per week at a Dartmouth.



Tristan Henderson, David Kotz, and Ilya Abyzov. The Changing Usage of a Mature Campus-wide Wireless Network. In Proceedings of ACM Mobicom, September 2004.

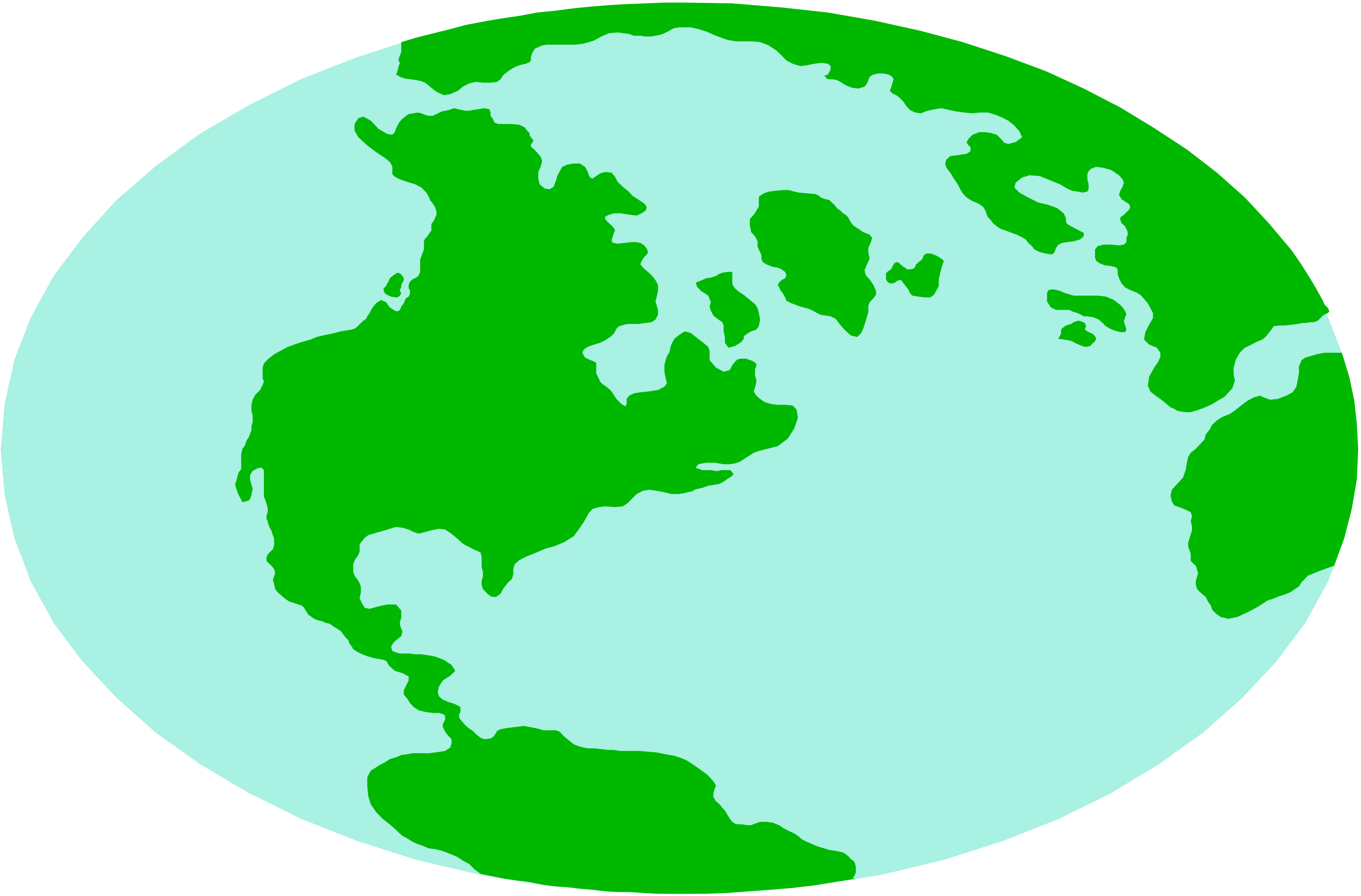
Web Performance: Intro

- User-perceived latency is a key factor because it affects the popularity of a site.
- In one study that passively gathered HTTP data for one day found that beyond a certain delay, user cancellations of the page increased sharply.

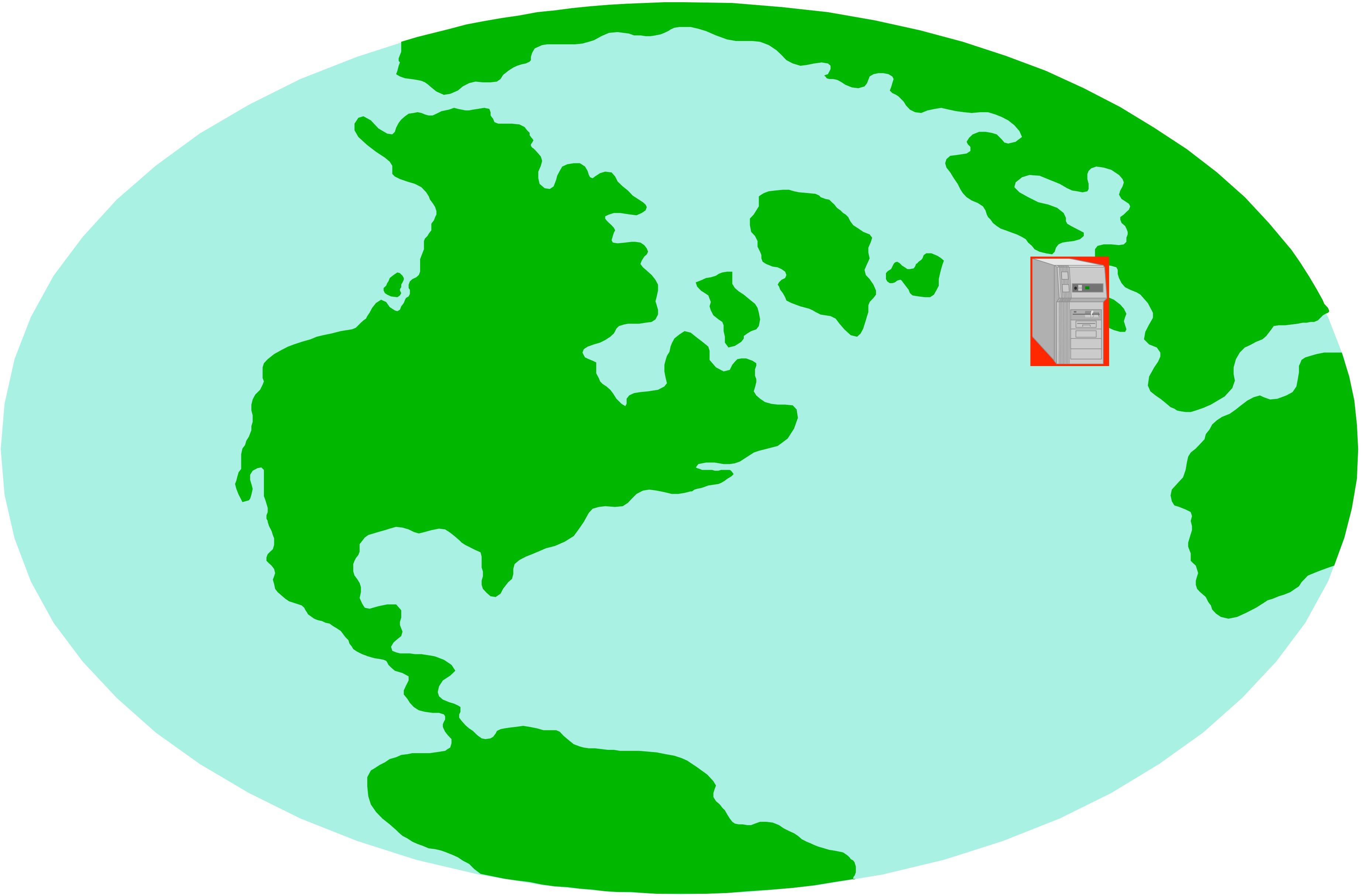
Web Performance: CDN’s

- Content distribution networks (CDNs) combine the workload of several sites into a single provider.
- The CDNs can be mirrored to be located near clients. DNS can be used to redirect clients to mirror sites.
- CDNs were initially thought to provide a large reduction in latency, but this has not always been borne out by experiments.

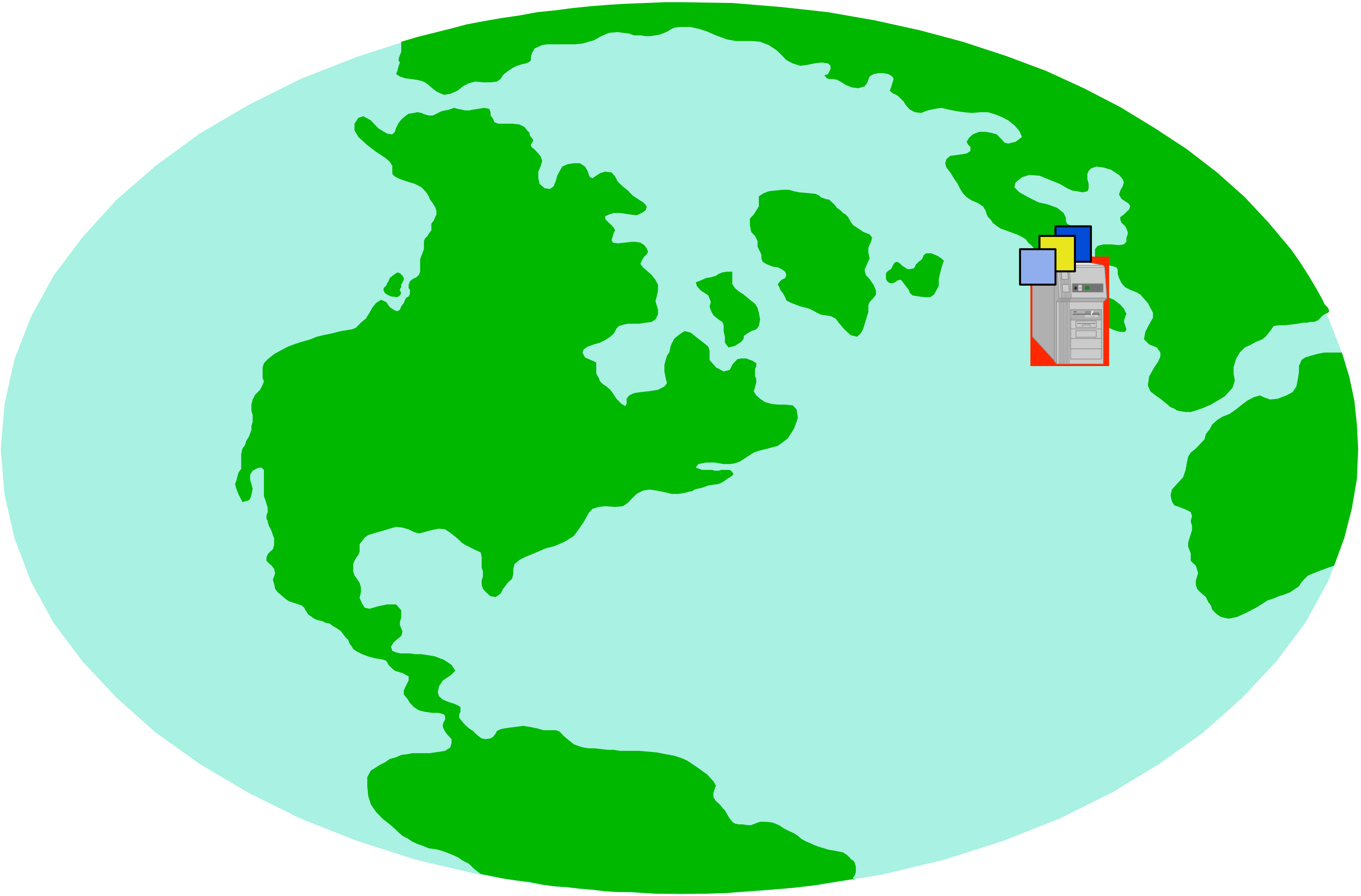
How CDN Works



How CDN Works

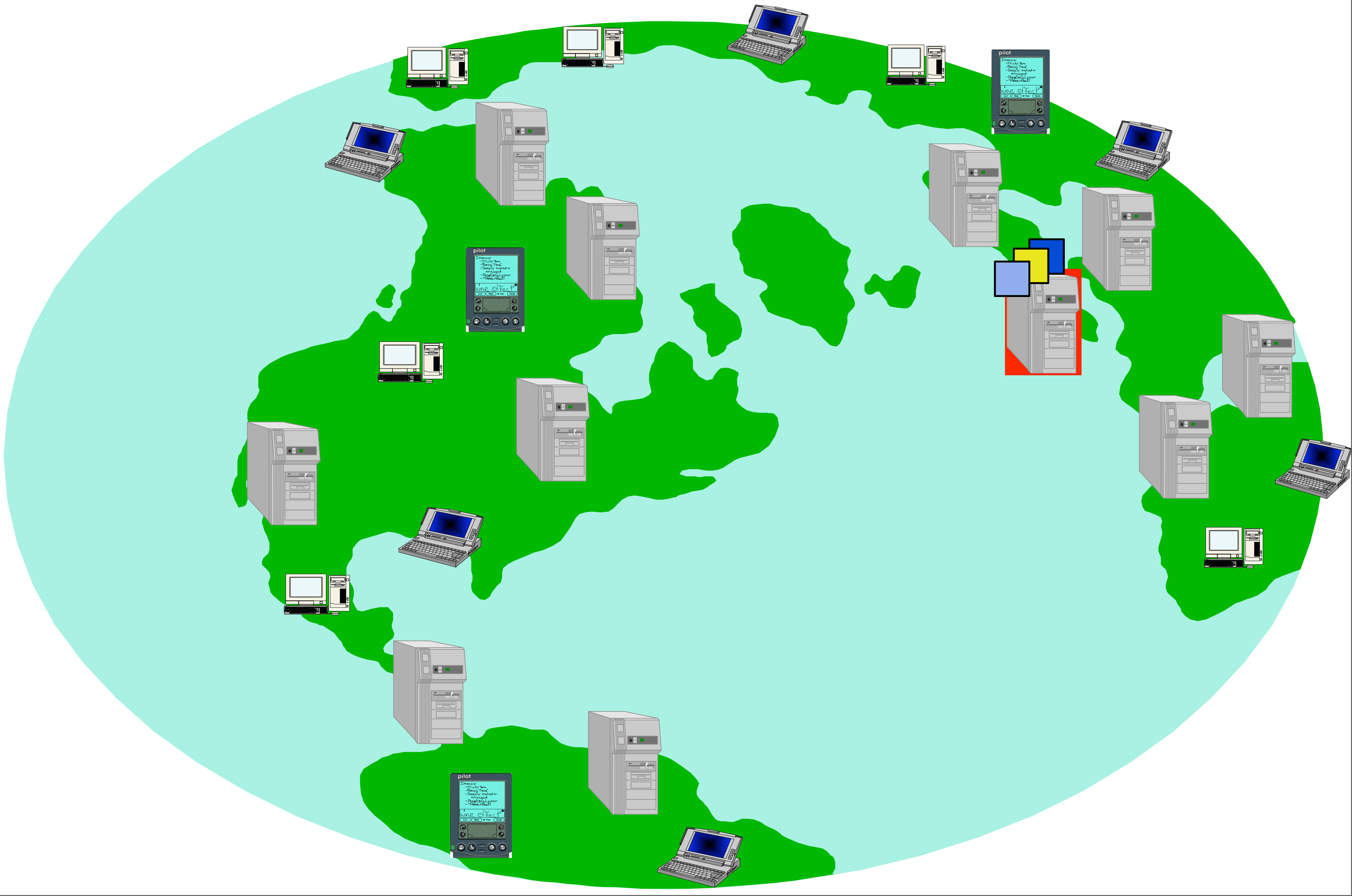


How CDN Works

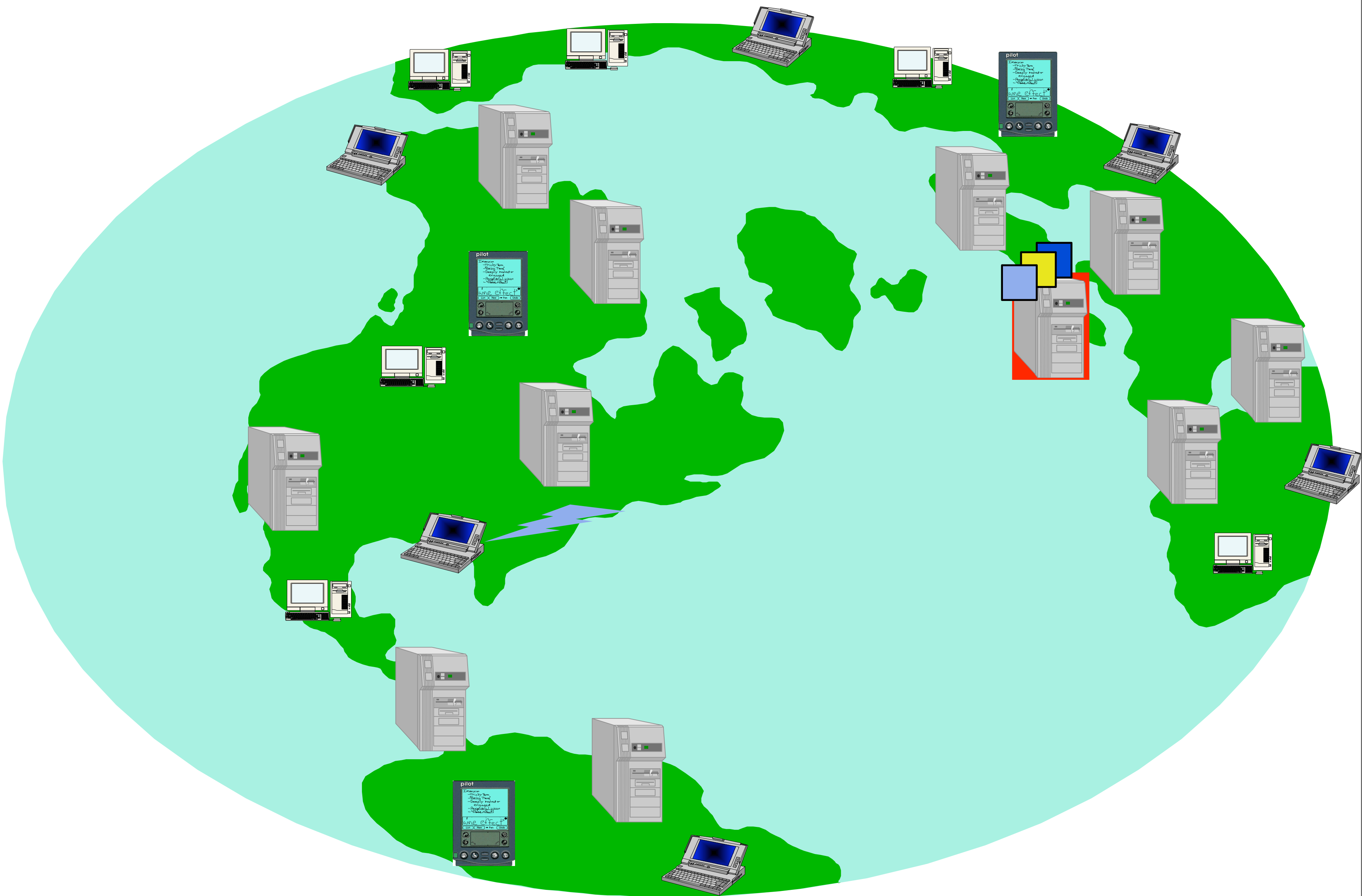


How CDN Works

How CDN Works



How CDN Works



How CDN Works

How CDN Works

How CDN Works

Web Performance: CDNs

TABLE IX

PARALLEL-1.0 PERFORMANCE (SEC.) FOR SERVER AT NEW AND FIXED IP ADDRESSES (JAN. 2001)

CDN (DNS TTL in sec.)	New Download Time			New Completion Time			Fixed IP Download Time		
	Mean	Med.	90%	Mean	Med.	90%	Mean	Med.	90%
Adero (10)	1.15	1.02	1.73	5.40	1.39	9.60	1.09	0.51	1.60
Akamai (20)	1.06	0.34	3.01	1.15	0.39	3.05	1.00	0.41	3.00
Clearway (N/A)	1.19	0.84	2.94	1.19	0.84	2.94	1.16	0.76	3.07
Digisle (20)	1.19	0.47	1.83	1.31	0.52	2.30	1.21	0.43	1.70
Fasttide (230)	1.58	0.96	3.37	2.10	1.19	4.72	1.46	0.91	3.25
Speedera (120)	0.57	0.20	1.18	0.72	0.26	1.53	0.53	0.18	1.01

Balachander Krishnamurthy, Craig Wills, and Yin Zhang. On the use and performance of content distribution networks. In Proceedings of the ACM SIGCOMM Internet Measurement Workshop, San Francisco, November 2001.

Web Performance: CDNs

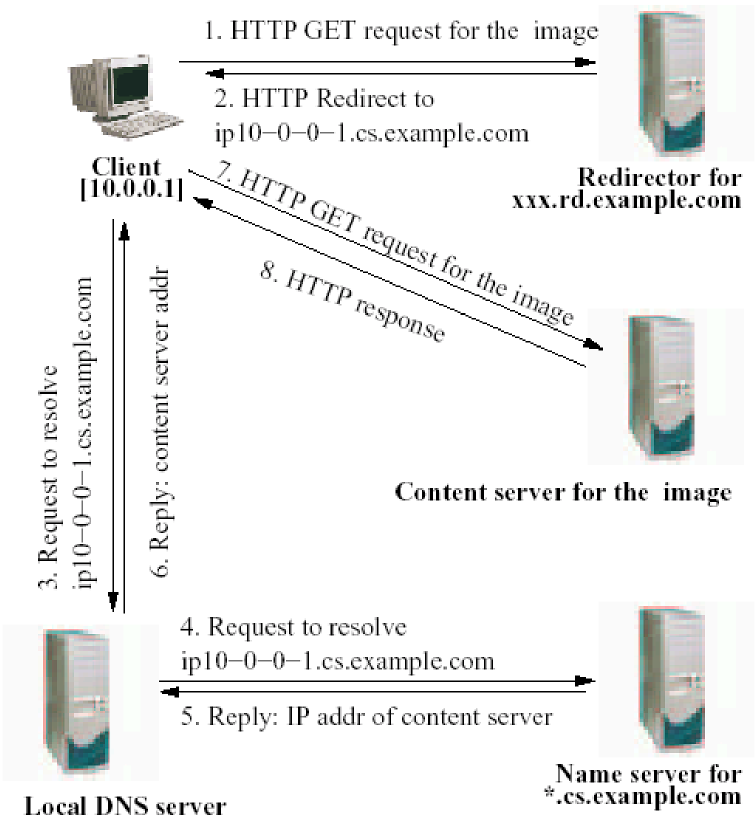


Figure 1: Embedded image request sequence

Zhuoqing Morley Mao, Charles D. Cranor, Fred Douglis, Michael Rabinovich, Oliver Spatscheck, and Jia Wang. A precise and efcient evaluation of the proximity between web clients and their local DNS servers. In Proceedings of the USENIX Technical Conference, Monterey, CA, June 2002.

Table 5: Percentage of client-LDNS associations sharing the same cluster classified according to the types of domains visited by the clients

Metrics	Client IPs			HTTP requests		
	educational	commercial	combined	educational	commercial	combined
AS cluster	70%	63%	64%	83%	68%	69%
Network cluster	28%	16%	16%	44%	23%	24%

Web performance: Client connectivity

- It is not practical to dynamically query a client's connectivity type, however such data can be stored on a server.
- We can measure the inter-arrival time of requests. Clients with higher bandwidth connections are more likely to request pages sooner.
- If we assume that client connectivity will be stationary (as one experiment showed), then we can adapt the server response based on the client connectivity

Web performance: Client connectivity

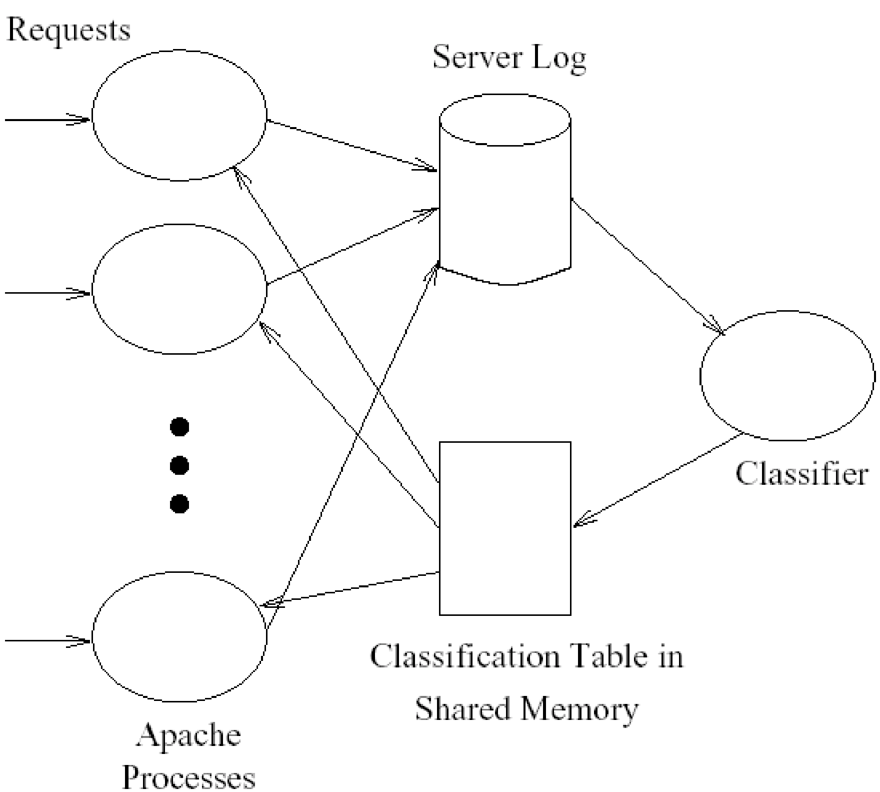


Figure 1: System Architecture

Server Action conclusions:

- Compression - consistently good results for poorer but not well-connected clients.

- Reducing the quality of objects only yielded benefits for a modem client.

- Bundling was effective when there was good connectivity or poor connectivity with large latency.

- Persistent connections with serialized requests did not show significant improvement

- Pipelining was only significant for client with high throughput or RTT.

Balachander Krishnamurthy, Craig E. Wills, Yin Zhang, and Kashi Vishwanath. Design, Implementation, and Evaluation of a Client Characterization Driven Web Server. In Proceedings of the World Wide Web Conference, May 2003.

Web performance: protocol compliance

- A 16-month study used the httperf tool to test for HTTP protocol compliance.
- The popular Apache server was most compliant, then Microsoft’s IIS.

Web Applications: Peer-to-Peer Networks

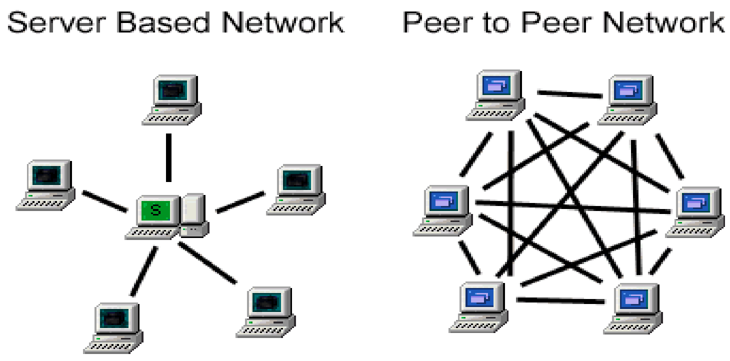
P2P : Overview

Network built and sustained by resources of each participant

Peers act as both client and server

Centralized/decentralized models

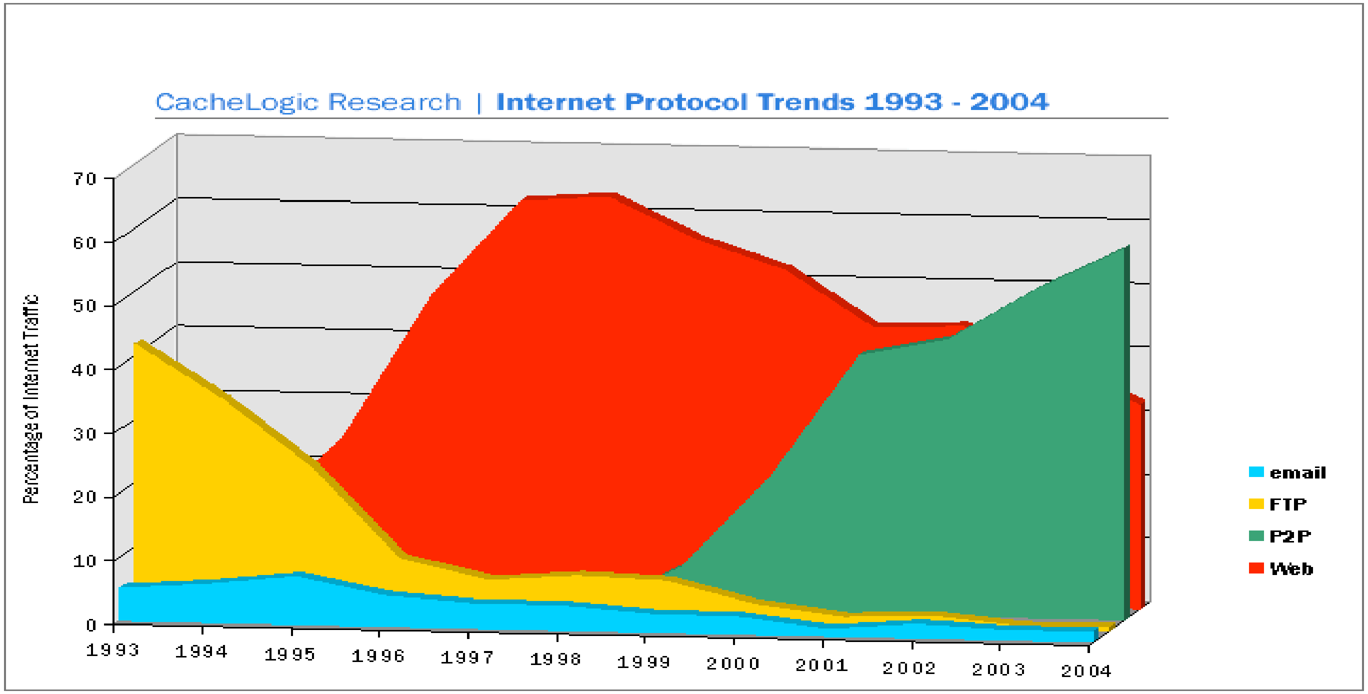
Issues: volatility, scalability, legality



P2P : Motivation

P2P networks generate more traffic than any other internet application

2/3 of all bandwidth on some backbones



P2P : Motivation

Wide variety of protocols and client implementations; heterogeneous nodes

Encrypted protocols, hidden layers

Difficult to characterize; node, path instability

Indexing, searching

Legal ambiguity, international law

P2P : Network Properties

Proportion of total internet traffic; *growth patterns*

Protocol split; *content trends*

Location of entities; *grouping/performance*

Access methods; *search efficiency*

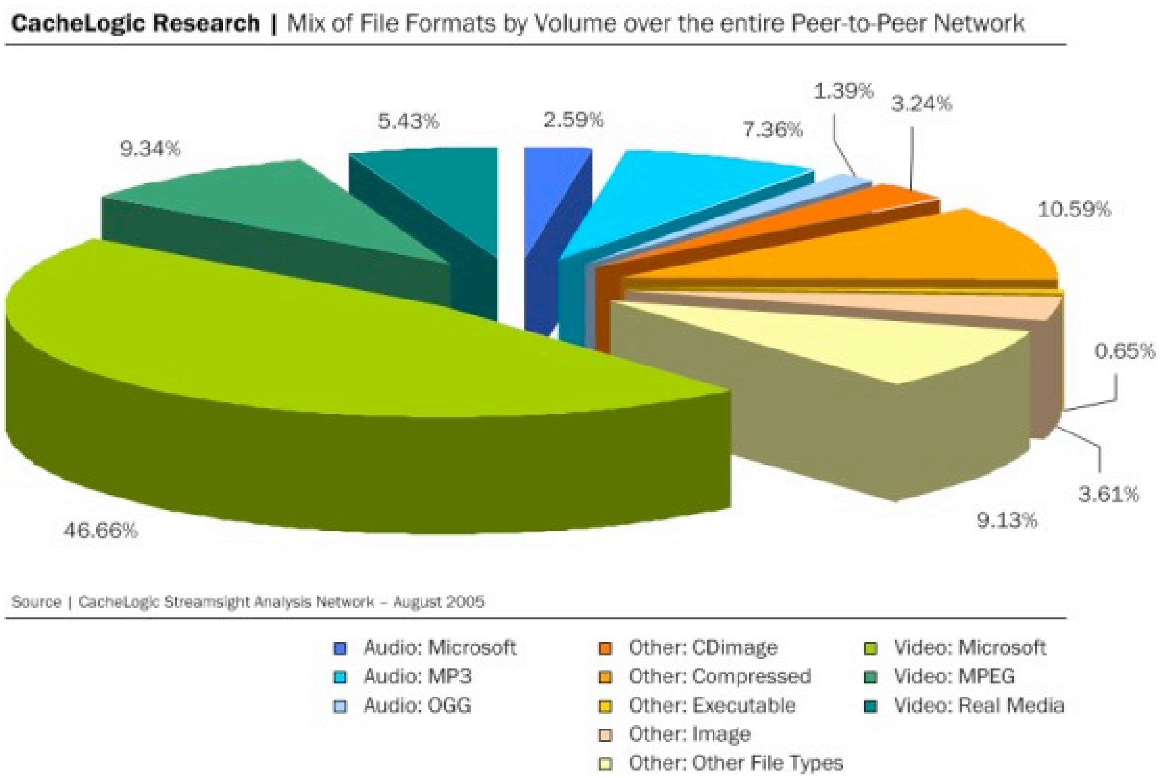
Response latency; *performance*

Freeriding/leeching; *network health*

Node availability; *performance*

P2P : Network Properties

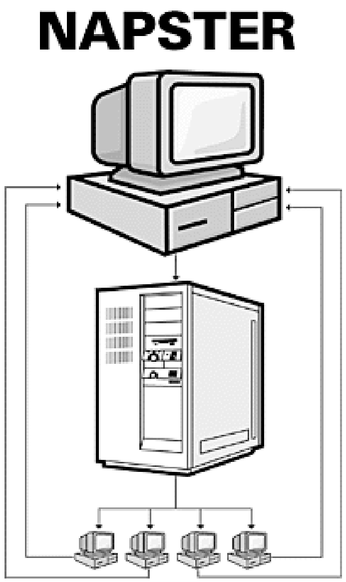
CacheLogic P2P file format analysis (2005)
Streamsight used for Layer-7 Deep Packet Inspection

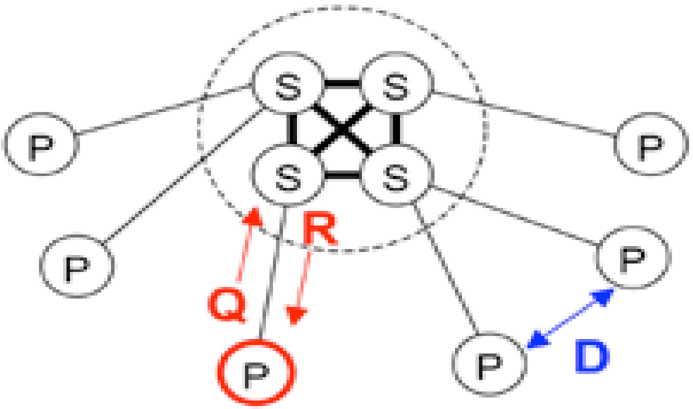


P2P : Protocols

Napster

- Pseudo-P2P, centralized index
- Tailored for MP3 data
- Brought P2P into mainstream, set legal precedence



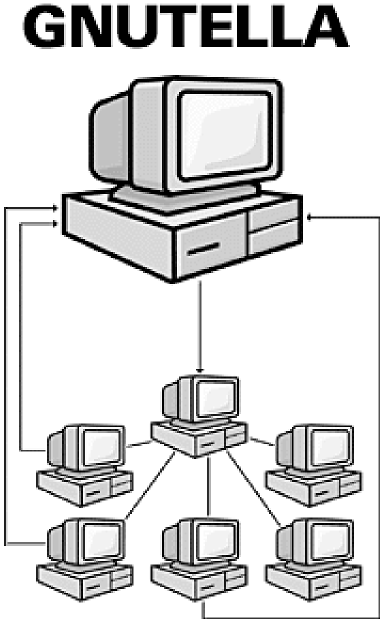


P2P : Protocols

Gnutella (Bearshare, Limewire)

- De-centralized algorithm
- Distributed searching; peers forward queries
- UDP queries, TCP transfers

Issues: Scalability, indexing



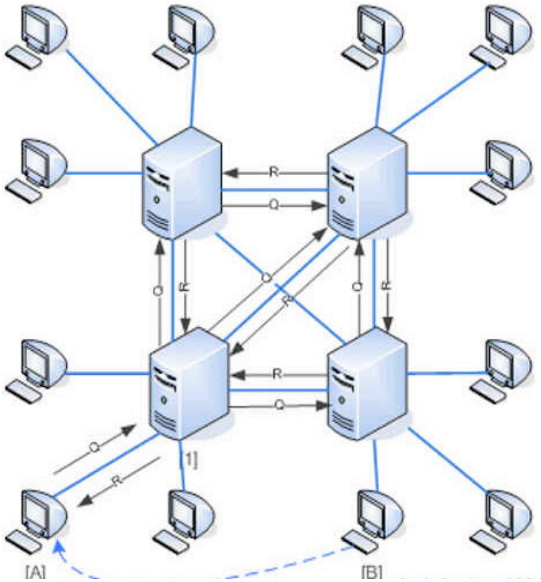
P2P : Protocols

- Kademlia (Overnet, eDonkey)
  - De-centralized algorithm
  - Distributed Hash Table for node communication
  - Uses XOR of node keys as distance metric
  - Improves search performance, reduces broadcast traffic

P2P : Protocols

Fasttrack (Kazaa)

- Uses supernodes to improve scalability, establish hierarchy
- Uptime, bandwidth
- Closed-source



- Uses HTTP to carry out download
- Encrypted protocol; queuing, QoS

P2P : Protocols

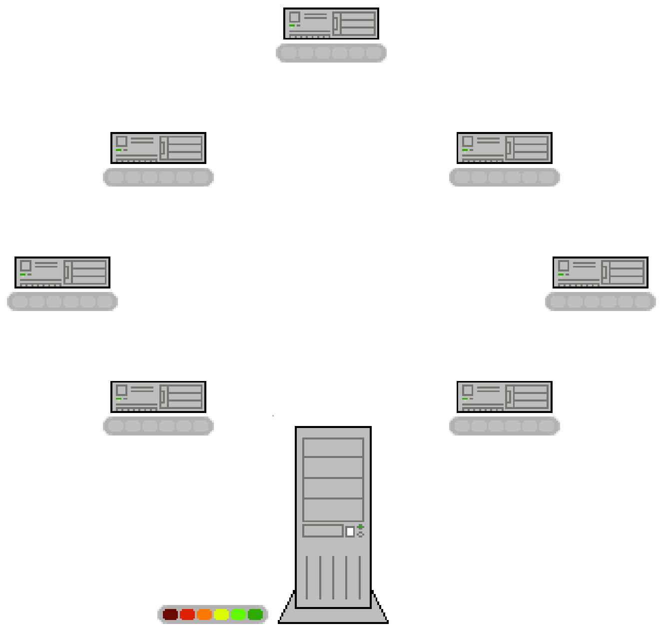
Bittorrent

- Simultaneous upload/download
- Decentralized network, external traffic coordination; trackers
- DHT
- Web-based indexes, search

- Eliminates choke points
- Encourages altruism at protocol level

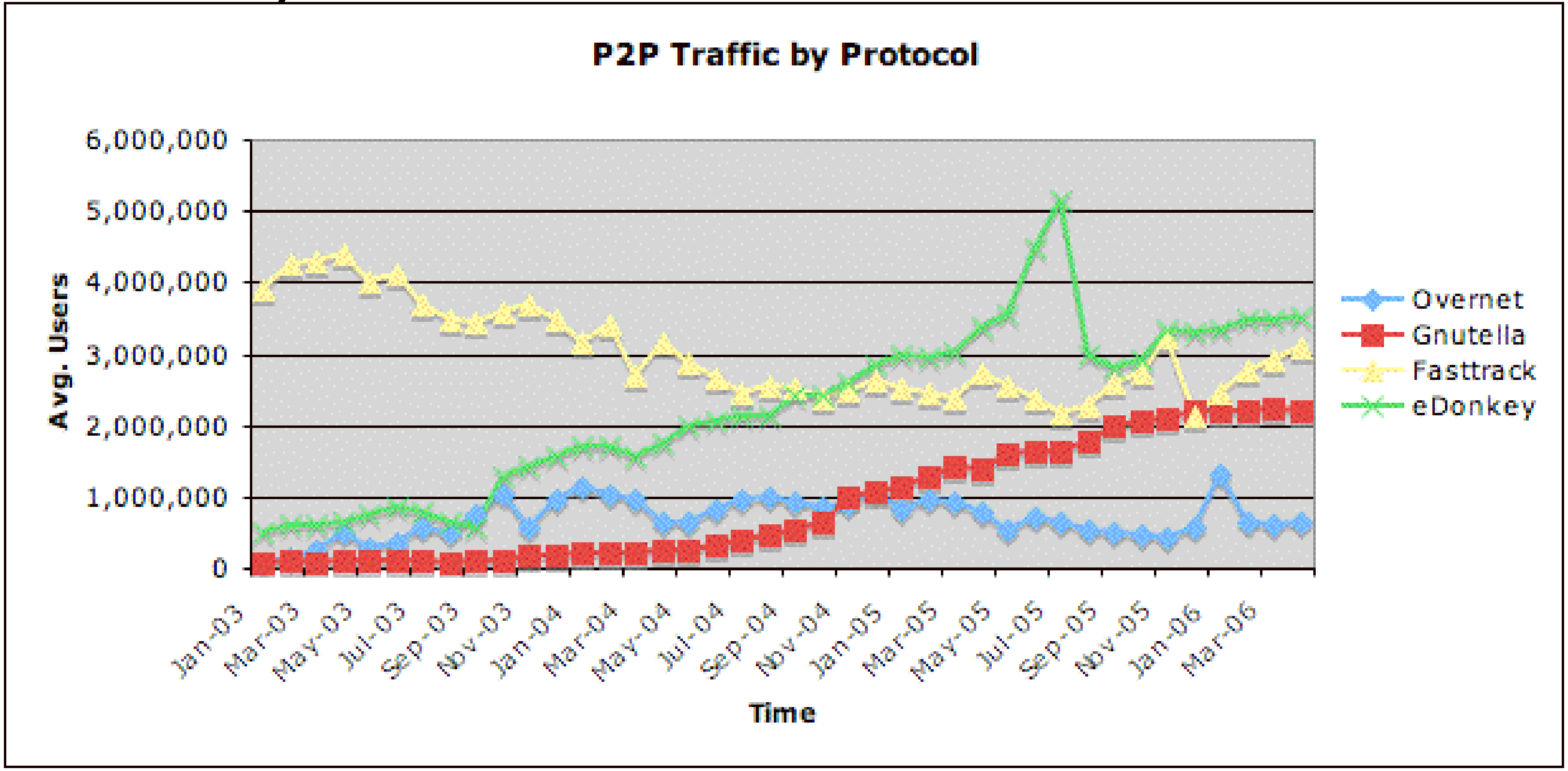
P2P : Protocols

Bittorrent - file propagation



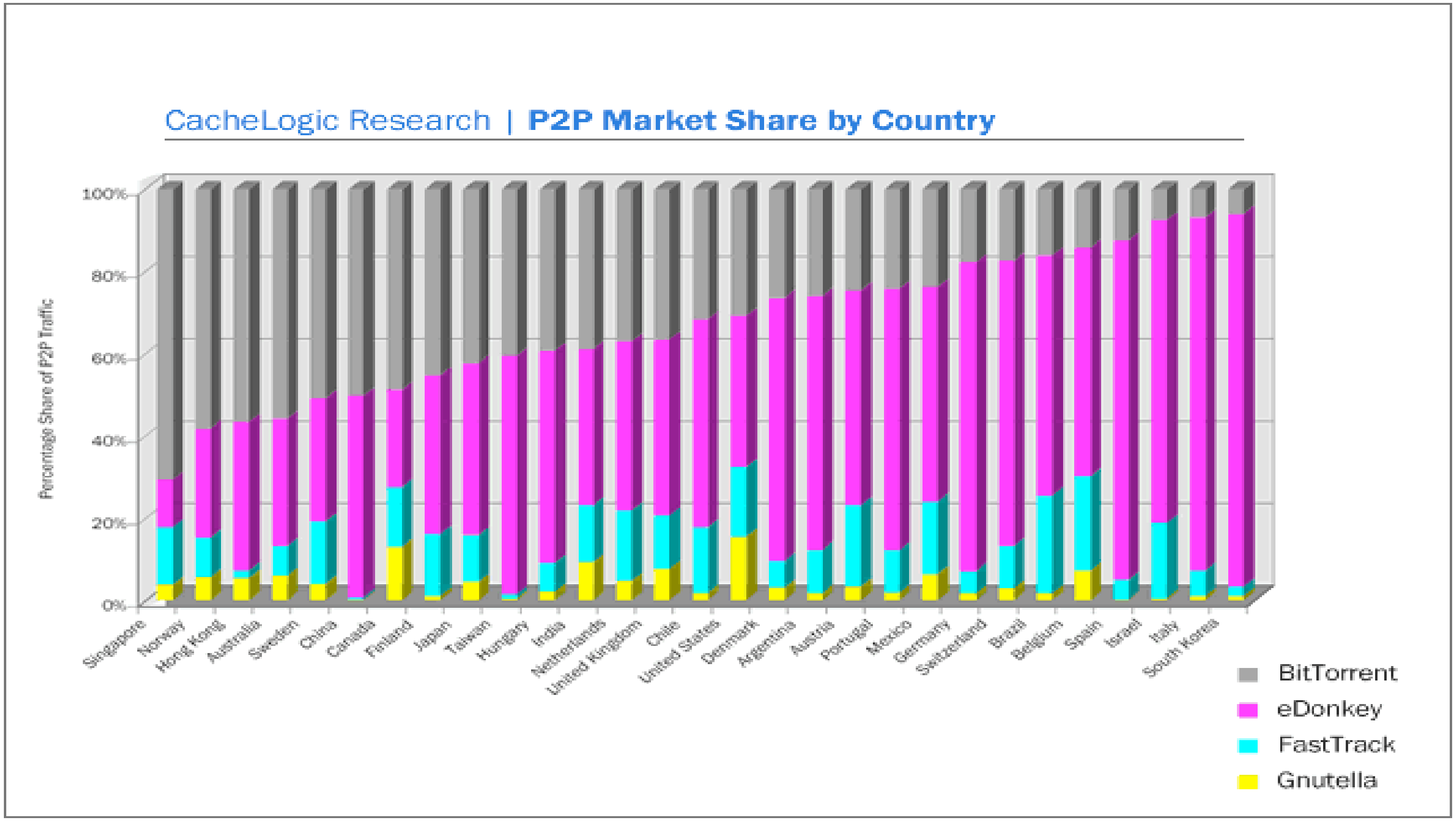
P2P : Protocol Trends

Trends in P2P Protocols (2003 - 2006)



P2P : Protocol Trends

Worldwide market share of major P2P technologies (2005)



P2P : Challenges

Lack of peer availability

Unknown path, URL

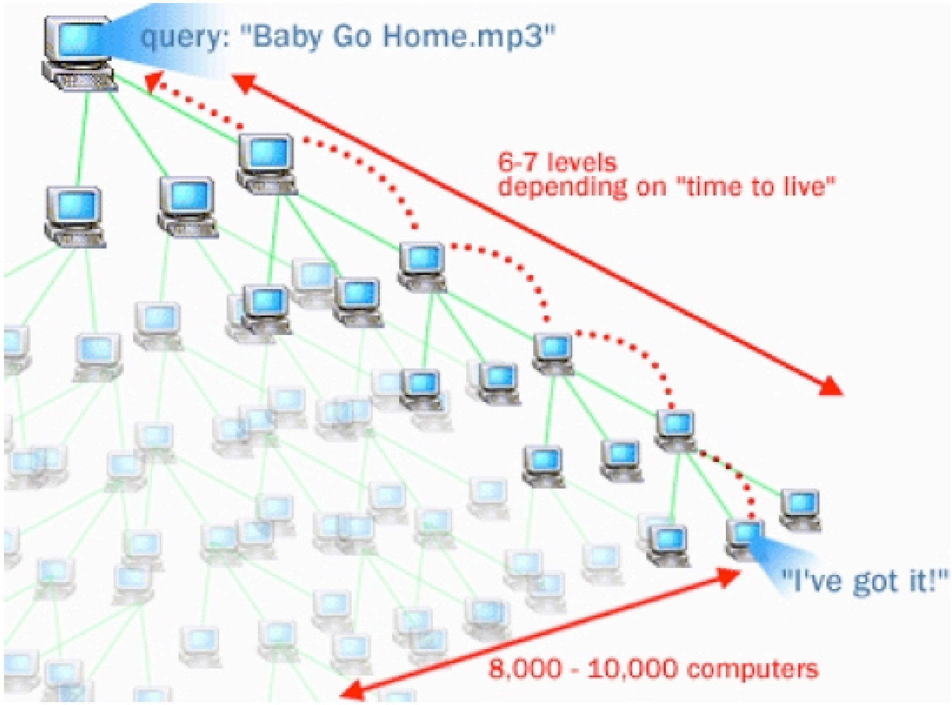
Measuring latency

Encrypted/hidden protocol

ISP/middleware blocks

P2P : Challenges

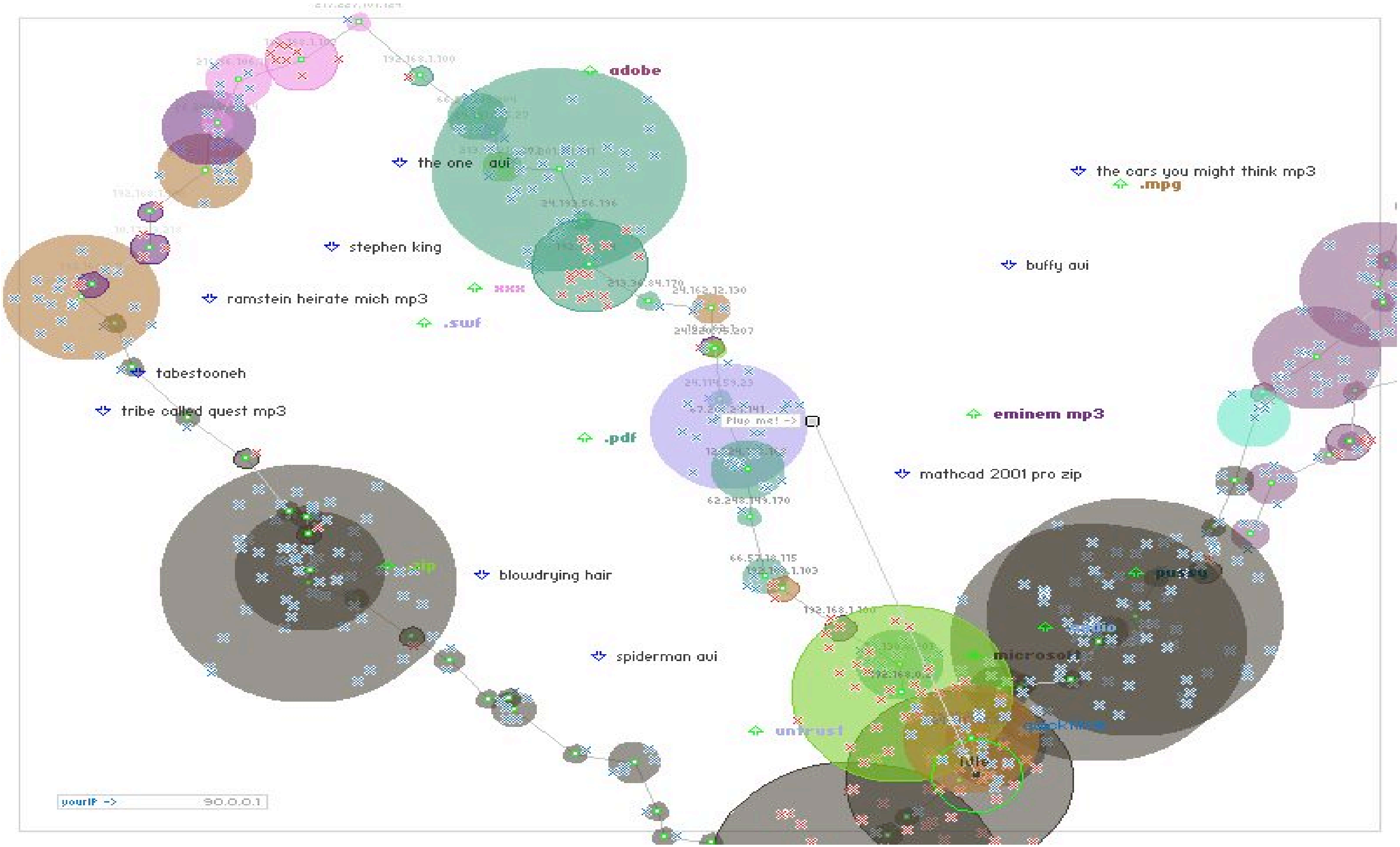
- Hidden Layers
  - Query diameter
  - Query translation/ parsing; response could be subset of query
  - Node selection



P2P : Measurement Tools

- Characterization - Active
  - P2P crawlers
    - Map network topology
    - Identify vulnerable nodes
    - Joins network, establish connections with nodes, record all available network properties (routing, query forwarding, node info)

P2P : Visualizing Gnutella



P2P : Visualizing Gnutella

- Minitasking - Visual Gnutella client
- Legend:
  - Bubble size ~ = Node library size (# of MB)
  - Transparency ~ = Node distance (# of hops
- Displays query movement/ propagation

P2P : Measurement Tools

- Passive measurement
  - Router-level information; examine netflow records
  - Locate “heavy-hitters”; Find distribution of cumulative requests and responses for each IP
  - Graph-based examination; each node has a degree (# of neighbor nodes) and a weight (volume of data exchange between nodes)

P2P : Architecture Examination

- Difficulty: Heterogeneous nodes, scalability
- Node hierarchy
  - nodes with the highest uptime and bandwidth becoming ‘supernodes’
  - cache valuable routing information
- Capacity awareness
  - Maintain state information; routing cache, edge latency, etc…
- Towards a more robust search algorithm…

P2P : Network-specific tools

Decoy prevention

checksum clearinghouse

Freeriding/leeching

protocol-level solutions to P2P fairness

P2P : State of the art

- High-level characterization
  - Experiment #1: Napster, Gnutella, Spring 2001
  - Java-based crawlers, 4-8 day data collection window
  - Distribution of bottleneck bandwidths, degree of cooperation, freeriding phenomenon
  - Findings:
    - Extremely heterogeneous; degree of sharing
    - Top 7% of nodes offer more files than remaining 93% combined

P2P : State of the art

- High-level characterization
  - Experiment #1: Napster, Gnutella, Spring 2001
  - Napster measurements:
    - Latency and Lifetime; send TCP SYN packets to nodes (RST = inactive)
    - Bandwidth approximation; measure peer’s bottleneck bandwidth
  - Findings:
    - 30% of Napster clients advertise false bandwidth

P2P : State of the art

Alternative Architectures

- Experiment #2: Gnutella, Summer 2001
- Used modified client to join network in multiple locations
- Logged all routing messages

- Proposed a network-aware cluster of clients that are topologically closer
- Clusters select delegates, act as directory server
- Found nearly half of queries across clusters are repeated and are candidates for caching
- Simulation showed much higher fraction of successful queries in a cluster-based structure
- Number of queries grow linearly, unlike Gnutella's flooding

P2P : State of the art

Experiment #3: ISP/Router data

- Used netflow records, 3 weeks
- Filtered for specific ports
- Found that signaling traffic is negligible next to data flow; 1% of IP addresses contributed 25% of signaling traffic.

P2P : Peer Selection

Challenge: Quickly locate better connected peers

Lightweight, active probes;

- ping (RTT)
- nettimer (bottleneck bandwidth)
- Trace + live measurement

P2P : Other uses

P2P-based Web search engine

Flash crowd; streaming video, combine with multicast tree

P2P support for networked games

P2P : State of the Art

eDonkey

- Tfcpdump-based study, August 2003
- 3.5 million TCP connections, 2.5 million hosts (12 days)
- 300 GB transer, averaged 2.5 MB download stream, 17 Kb for signalling traffic

Bittorrent

- Tracker log study, several months, 2003
- 180,000 clients, 2 GB Linux distro
- Flash crowd simulation, 5 days

- Longer client duration; 6 hours on average
- Nodes prioritize least-replicated chunks
- Average download rate: 500 kb/s

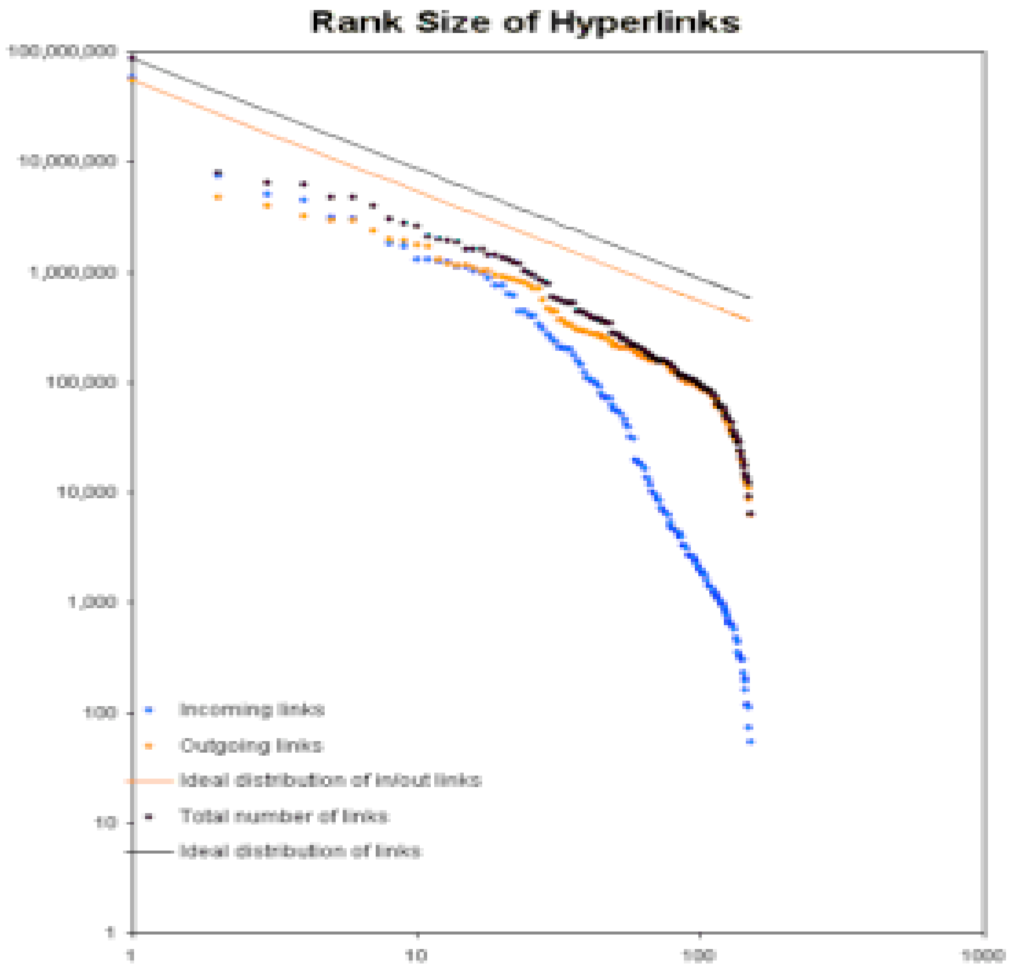
Web Applications

Searching

- Many popular search engines, key details on crawlers not widely published
- Research crawlers only gather fraction of Web

- 1999 Web study
  - Examined 200 million pages/1.5 billion links
  - Found that not all pages could be reached starting anywhere
  - Central core of web
    - Two parts either pointing to it or pointed to by it
    - Last part of web completely disconnected from core

– in/out degree distribution found to follow power law



- Web pages with large in-degree
  - Considered more important
  - Higher rank for search engines
- 90% web pages found reachable from each other
- Probability of reaching a random page from another is 25%
- Removal of hub will not always remove connectedness

- Method of crawling can distort results
  - Dynamic pages not included in study
  - Avoidance of loops requires parametric constraint on depth of crawl in site
- False links used to distort rank
- Crawler can be gamed

Frequency of page changes

- 1999 study showed wide-variance among content types
  - Images change infrequently
  - Periodicity in text changes
  - 15% changed between each access
  - Later studies showed frequency of changes increased access rates
- Crawlers used information to decide frequency of revisiting pages

Mercator 2002 Study

- 150 Million pages over 10 weeks crawled repeatedly
  - Half the pages successfully fetched in all crawls
  - Only .1% of documents saved
  - “Shingling” used to eliminate identical pages
    - Works for English language, no evidence for Asian languages

- Over half of pages from .com domain
- .edu pages half the size of avg. page
- .edu pages remain accessible longer
- 1/3 pages changed during crawls
- Longer documents changed more often than shorter ones

Impact of search engines

- Popular web pages get more popular through search engines
  - Rank increases higher
- Less popular pages drop further in ranking
- New high quality content has difficulty becoming visible

Dead links

- Study showed over 50% of pages dead links in some cases
- Crawlers must avoid dead links to complete crawls faster

Flash crowds vs. Attacks

- The avg # of requests per client remain the same
  - Proxies or spiders can high significantly higher rates
- Number of BGP clusters in flash event did not increase
  - Most clients belong to previous clusters

- Attacks (Code-Red worm)
  - Increase in requests per client
  - Client clusters varied from previous clusters
    - Only 0.5% to 15% clusters seen before

Blogs

- “Weblogs”
- Personal journal kept online
- Rapid growth in popularity

- Popular blogs provide warning for flash crowds
  - Links on sites such as slashdot.org indicate rising popularity
- Blogs typically have large in-degree
- Blogs must be updated frequently to maintain popularity

Characterization of Blogistan

- Early studies showed 1-4 million blogs
  - Found by crawling collection of 'seed' pages
  - New URLs found to have fewer references than older URLs
  - 12,000 unique IP addresses found
  - ~80% of blogs run on Apache
  - Avg. number of bytes added in changes low

- Rate of change for blogs different from traditional web pages
- Nature and count of links different
- Strong interaction found between blogs
  - Topic will cause rise in inter-references
  - Community built around topic, dies with the topic

Internet Measurement of Applications: *Games*

Usman Jafarey

Why?

- One of the fastest growing areas of the Internet
- Initially games with low real-time requirements (card games, etc)
- More recently non-sequential gaming has become popular

Properties

- Wide-variety of networked games
  - First Person Shooters (FPS)
    - Most popular type of online gaming
    - High real time requirements
  - Real Time Strategy (RTS)
  - Massive Multiplayer Online Role Playing Games (MMORPGs)

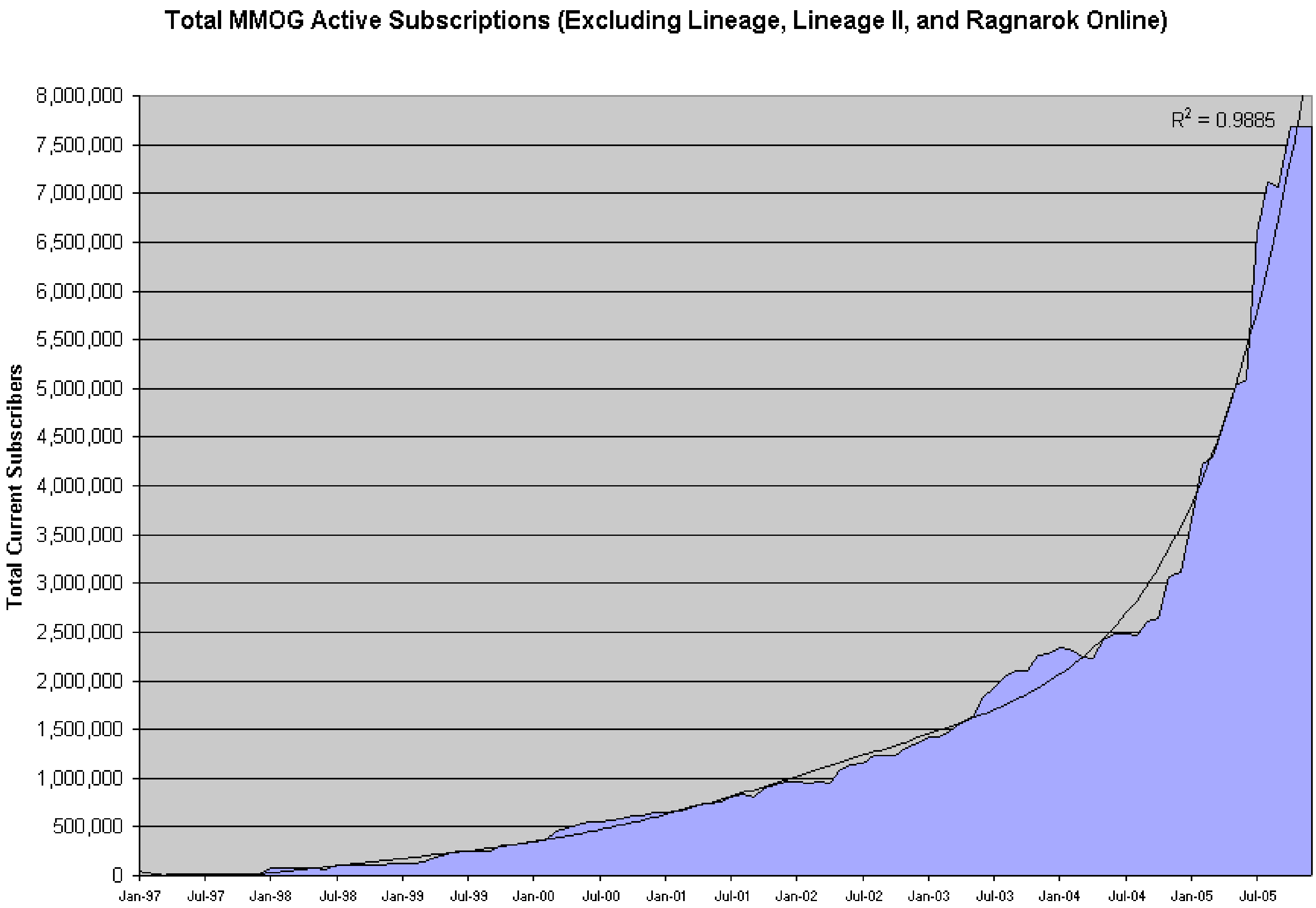
Motivation

- On-line games are big business
  - 60% of all Americans play video games (IDSA report, 2003)
  - MMO games
    - 4,000,000 World of Warcraft subscribers paying monthly fees
  - FPS games
    - 100,000 Counter-strike players at any given time
  - RTS games
    - >8 million Warcraft game copies sold
    - 200,000 Warcraft 3 games played online / day
- Hosting games very costly (30% of revenue)

Properties (cont.)

- Variety of platforms
  - PC
  - Playstation
  - Xbox
  - Nintendo

Growth in MMORPG subscriptions



Measurement properties

Measurement Property	Why Measured	Where Measured
Fraction of Internet Traffic	Growth Patterns, popularity	Across Internet
Game genre	Difference in architecture	Across Internet
Scalability	Provisioning, performance	Varies with game genre
Real-time requirements	Game viability/latency limitations	Game server
Manner of access	Mobility constraints	Client and server locations
Session duration	State maintenance	Server

Table 7.6

Server responsibilities

- Authentication
- Updating positions
- Maintaining scores/information about players and teams
- Managing forming of teams

Architecture

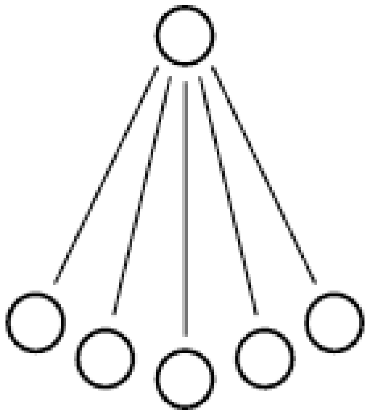
- Three types
  - Centralized
  - Decentralized
  - Hybrid of the above two

Centralized architecture

- All interaction requests sent through a central server

- All clients not required to know movements of all other clients at any given instant

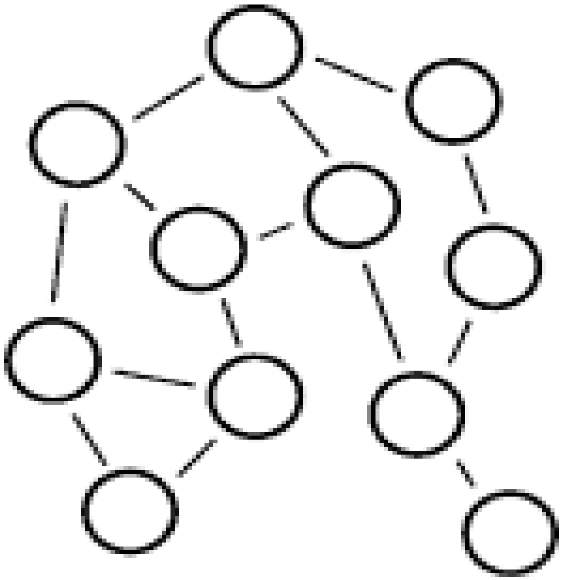
- Server decides what each client needs to know



- Server requirements:
  - High processing capability
  - High reliability
  - Low latency/packet loss between clients and server
- Used to prevent cheating amongst clients
- Most commonly used architecture today

Decentralized architecture

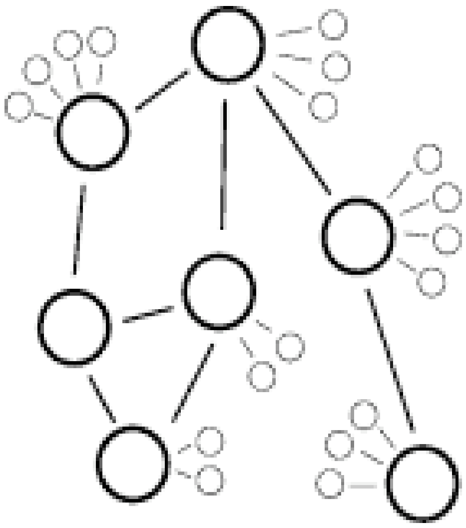
- Clients interact with each other directly
- Proposed decentralized architectures:
  - MiMaze
  - Mercury
  - P2P-Support
  - Zoned Federations



- Partial decentralization
  - partitioning players and associated responsibility into regions
- Complete decentralization
  - Any peer in P2P network can carry out authentication requirements to eliminate cheating

Hybrid architecture

- One example: Mirrored server
  - Each game has several distributed servers
  - Clients only communicate with one of these



Scalability

- Number of users that can simultaneously participate in a networked game
- Typical numbers
  - <10 for RTS
  - 10-30 FPS
  - Thousands in MMOGs
- Increased users cause increased delays

Real-time requirements

- Often the limiting factor in viability of a game
- Varying requirements for latency and packet loss
- Even within a single networked game, different objects may require different real-time standards
  - e.g., high accuracy sniper rifle vs. machine gun

Wired/Mobile environment

- Physical location of client can be used
  - Require accurate client location abilities
  - Active Bat, Cricket (indoor location systems)
  - *Human Pacman*
- Most games require wired environment for lower latency/packet loss

Single session vs. Multi-session

- Single session
  - User connects, plays, then exits game
  - more common among older games
- Multi-session gaming
  - User logs in, plays, stalls session until next game
  - Increases necessity for network performance in certain cases
    - Character value can drop with network performance (for example, Diablo II 'hardcore' mode)

Challenges

- High interactivity, low-tolerance compared to Web/DNS
- Harder to simulate user traffic via programs

Hidden data

- Skill levels of users
  - impacts importance of latency/packet loss/etc.
  - No uniform way to measure impact of network problems
- Information about game server rarely public, difficult to reverse engineer
- Downloading of new content can effect performance

- Games typically involve authentication, setting up parameters, playing, and quitting
  - One or more steps may be avoided through suspension of state at the and of a session
- Authentication generally done via TCP handshake
- Game actions usually sent over UDP or TCP
- Game updates sent over TCP
- Less complex than short session applications (e.g., Web)

- Quality of game effected by
  - Network
  - Client
  - Server
  - input/output devices
  - Delays cause different users to react differently
  - Delays on server end factored into measuring delays from player's view
- Team games add more complexity to measurements
- Time of game effects impact of adverse network conditions
- Location of player changes effect of network problems

Measurement tools

- Ping used to measure latency, latency radius (number of active players within latency threshold)
- Geographic mapping tools used to locate game servers
- RTT measured at time of special events such as a player dying

- Measured passively at server
  - Average bandwidth
  - packet interarrival time
  - packet count and size
  - number of attempted/successful connections
  - unique clients
- Non-traditional measurement tools tailored to individual games
  - Servers chosen based on network latency, number of players
  - GameSpy tool used to report number of players associated with game server

State of the Art

- Architecture
- Traffic characterization
- Synthesizing game traffic
- Mobile environment

MiMaze

- Decentralized server research
  - IP Multicast used for player moves
  - Latency limited to 100ms
  - Cheating prevented popularity of architecture

- Improvements for decentralization
  - Proxies to offset work on part of central server
  - Peer-to-Peer systems
    - Centralized arbiter only required during state inconsistencies
    - Account information stored centrally
    - Scales to number of players
    - Players in a region affects performance
    - Multicast used for position updates
  - Distributed Hash Tables used to remove application layer multicast

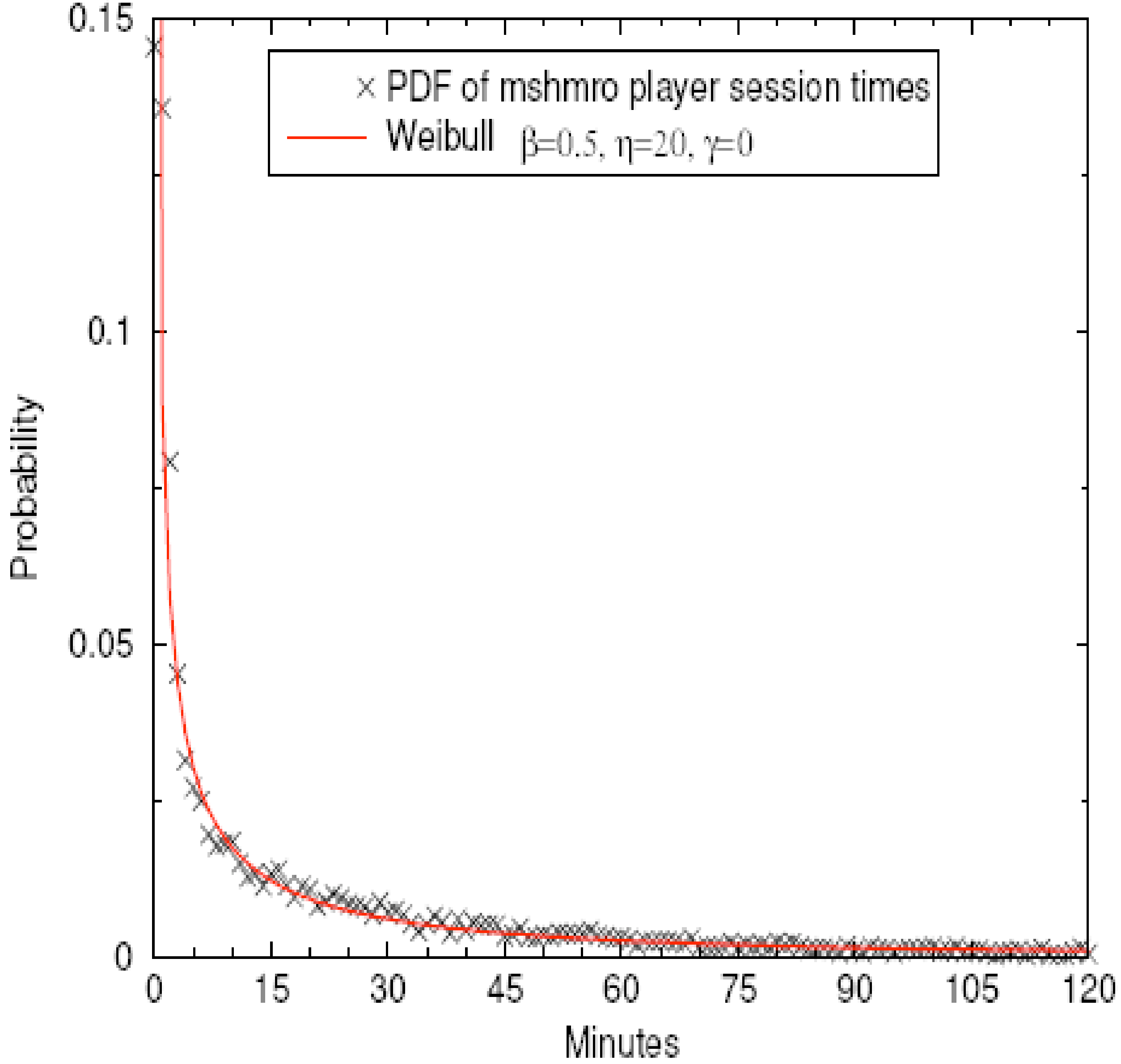
Characterization

- Quake World and Unreal Tournament
  - Both use UDP and listen on ports 27500 and 7777
  - Data gathered passively using DAG cards(packet capturing hardware)
  - Client packets found more numerous but smaller than server packets in Quake

- CounterStrike
  - Half a billion packets captured in 1 week from ~6000 players
  - Showed that updates must be predictable to compensate lag
  - Client/server packets maintained properties from Quake study
  - Regular traffic bursts found
  - Active clients sent relatively uniform load

- Player behaviour studied across a few thousand Half-Life and Quake servers
  - Time-of-day effects game traffic
  - Players joined games with higher numbers of players
  - Duration of player's session independent of number of players, relatively constant

- GameSpy used to study Counter-Strike
  - Contrary to most applications session times followed a Weibull distribution
  - Most players played for short durations
  - Study showed difficulty of generalizing network games


PDF of mshmro player session times
Weibull β=0.5, η=20, γ=0
Probability
Minutes
0.15
0.1
0.05
0
0
15
30
45
60
75
90
105
120

Quake 3 study

- Used server in California and London
  - Intentionally masked London server as California location
  - Found players chose servers closer to them geographically
  - Bottleneck last mile between user and ISP

- Unreal Tournament 2003 study
  - Emulating packet loss and latency according to live server data
  - Found no significant difference in ability to move due to packet loss (prediction compensation)
  - Even 100ms latency caused drop in perceived performance

Synthesizing game traffic

- Each game must be examined and synthesized separately
- Representative set of players must be found and data captured over a period of time
  - Skill of players will effect data
- Typical information gathered
  - number of packets
  - packet length
  - interarrival time
  - server response time

Mobile environments

- Few measurements so far
- Study on GAV game ported to PDA found that wireless environment could not support real time requirements of GAV

Traffic Characterization

- Fraction of Internet, individual popularity of games
- Sample traffic flowing to and from port numbers common to games

- Individual game characterization
  - Size, inter-arrival time of packets
  - Behavioural differences between clients and server
  - Large amount of games take place over proprietary networks – surveys used in these cases.

- One possible solution: allow game server to handle authentication/initiation while wireless terminals associated handle low-latency requirement operations

- Algorithms used by server to deal with traffic difficult to reverse engineer
- Arrival rate of broadcast packets depends on server/user-generated traffic
- Fortunately, usually no intermediaries between client and server

Negative network effects

- Latency
  - Delay in accessing game server
  - Load on game server
  - Load on network