An Analysis of Internet Content Delivery Systems (and more!)

Stefan Saroiu, Krishna Gummadi, Richard J.Dunn, Steven D. Gribble, and Henry M. Levy

Proceedings of the 5th Symposium on Operating Systems Design and Implementation (OSDI 2002)

Presented by Bryce Boe (CS290F W2010)

Paper Goals

- Measure how Internet traffic has changed from 1999
  - Highlight peer-to-peer traffic increases in up/down directions
  - Highlight shift (at UW) from client based traffic to server based
- Evaluate caching potential in peer-to-peer networks

Content Delivery Systems

- WWW
- Content Delivery Networks
- Peer to Peer Systems

World Wide Web

- Client / Server model
- Averaged small sized objects 5-10KB
- Zipf popularity distribution
  - Very small number of sites have ridiculously high popularity
  - Very large number of sites have ridiculously low popularity
- Support for caching
  - Fetch object from source when unavailable
  - HTTP Headers for give content providers some control

Content Delivery Networks

- Idea: move data close to end user
  - Accomplished through URL rewriting, DNS, or Anycast
- Grants more control to content providers
  - Expire/Invalidate objects
  - Pre-cache objects
  - Serve homepage primarily from CDN (reddit)
- Benefit from overlay networks
  - Work around unreliable middle mile
  - Fetch content from other CDN nodes rather than ISP

CDNs Continued

- First Mile (server to ISP) $$$ = 20x cap. growth/5yrs
- Middle Mile (ISP to ISP) no $$$ = little growth
  - Peering Wars
  - Physical Outages
  - BGP Attacks
- Last Mile (ISP to end user) $$$ = 50x growth/5yrs

T. Leighton, “Improving Performance on the Internet,” Communications of the ACM, Vol. 52, No. 2, February 2009

Peer-to-Peer Systems

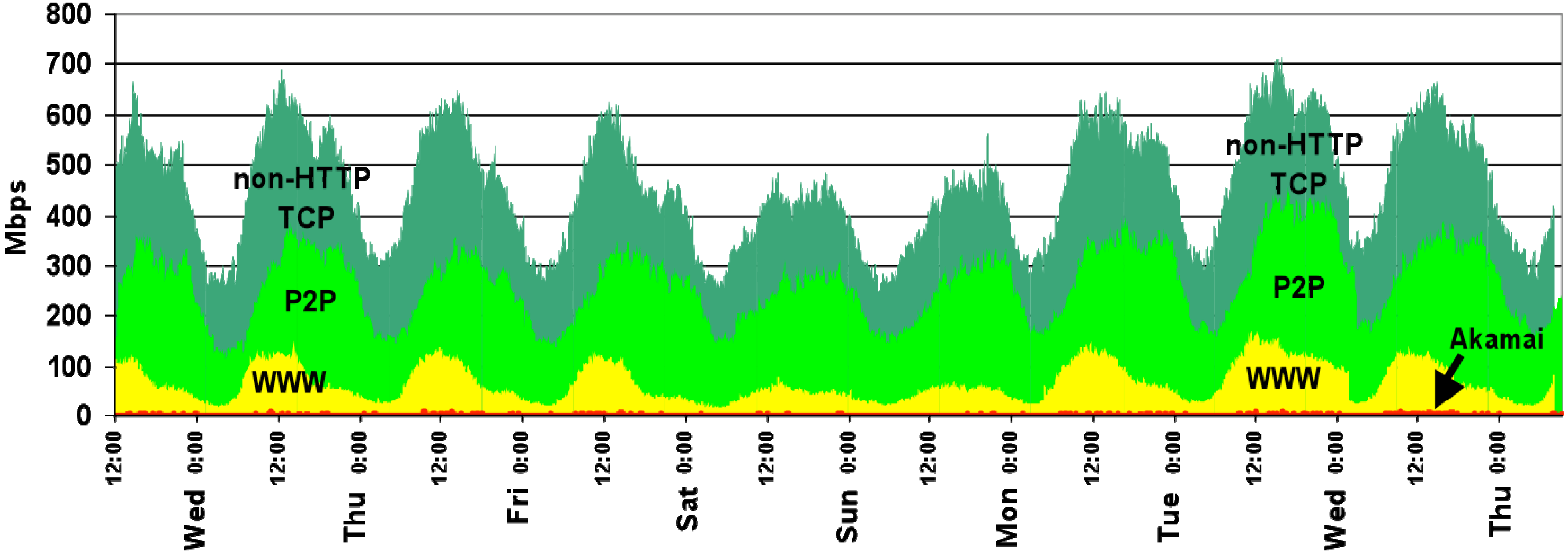
- Nodes behave as clients and servers
- Searching done through different means
  - Gnutella: query flooding across nodes within n-hops
  - Kazaa: Similar with addition of *supernodes which* contain indexes of all nearby nodes, and query floods across connected supernodes within n-hops
- Support for parallel fragment download
- *Designed* to be highly distributed

Passive Network Monitoring

- Monitor inbound and outbound connections on their 4 backbone connections
- Traffic Classifications
  - Akamai Traffic (akamai hosts)
  - HTTP Traffic (ports 80, 8080, 443)
  - Gnutella Traffic (ports 6346, 6347)
  - Kazaa Traffic (port 1214)
  - P2P (Gnutella + Kazaa)
  - Non-HTTP TCP (All TCP traffic – Akamai – HTTP – P2P)
- Limitations?

WHERE IS THE BANDWIDTH GOING?

Bandwidth Distribution

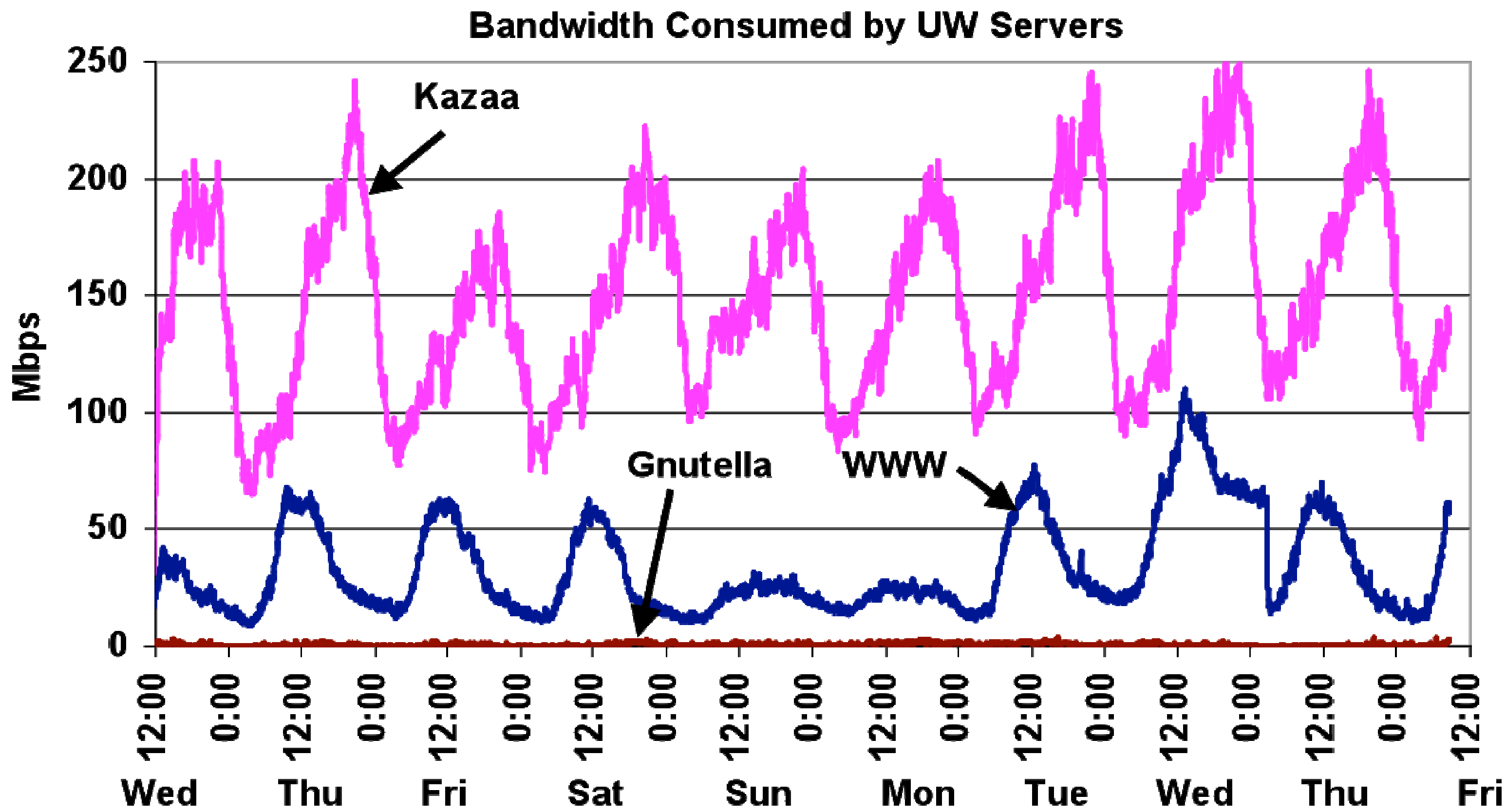


- Seemingly significant amount of non-HTTP TCP

WWW v. Kazaa Summary

	WWW		Kazaa	
	inbound	outbound	inbound	outbound
Bytes Xferred	1.51TB	3.02TB	1.78TB	13.6TB
Unique objects	72,818,997	3,412,647	111,437	166,442
Clients	39,285	1,231,308	4,644	611,005
Servers	403,087	1,463	281,026	3,888

UW Server Bandwidth



Today's Traffic

H. Schulze and K. Mochalski, "Ipoque Internet Study 2008/2009," Ipoque, 2009.

- Measurement study of 8 regions around the world (consider Germany trace)
- 14 days captured (v. 9 days)
- 560 terabytes (v. 20 terabytes)
- 100 thousand users (v. 60+ thousand users)
- 53% peer-to-peer traffic (v. 43%)
- 26% web traffic (v. 14%)

Changes from 2007

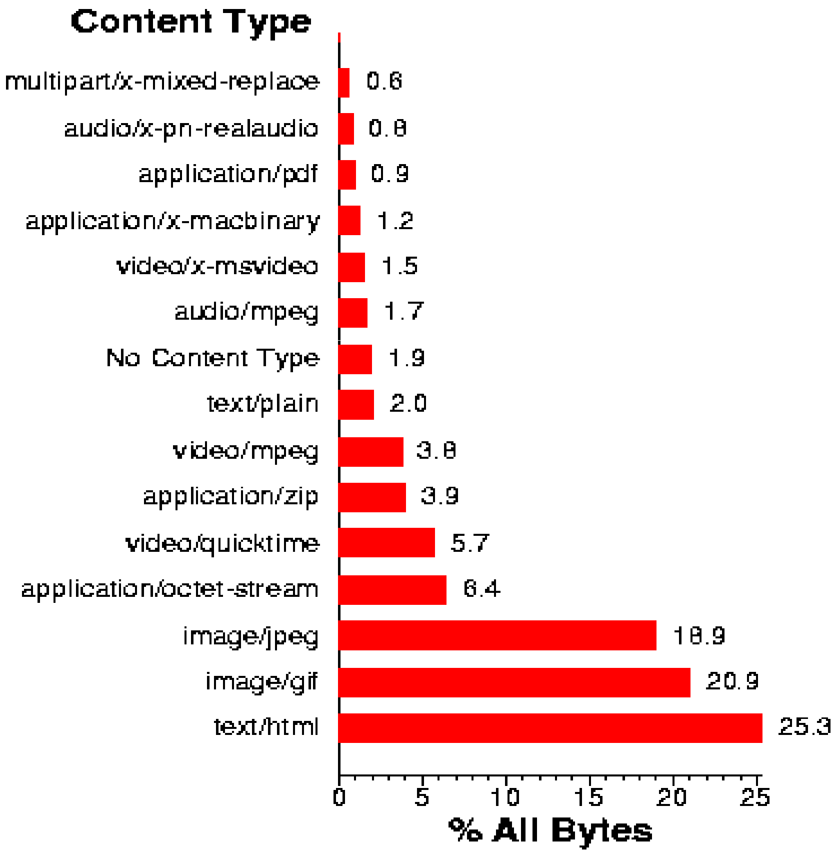
- P2P: 69.25% → 52.79%
- Web: 14.35% → 25.78%

Breakdown

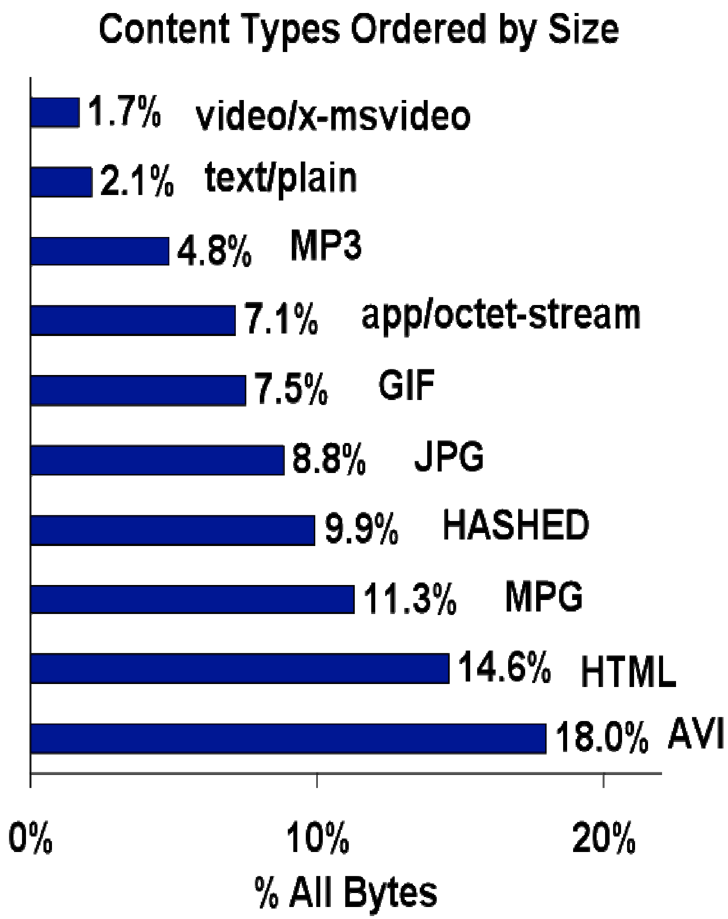
- P2P
  - 37% BitTorrent
  - 13% eDonkey
- Web:
  - 15% HTTP
  - 10% Filehosting (RapidShare, Megaupload)

VIEW 1: OBJECT VIEW

Bytes Transmitted: 1999 v. 2002

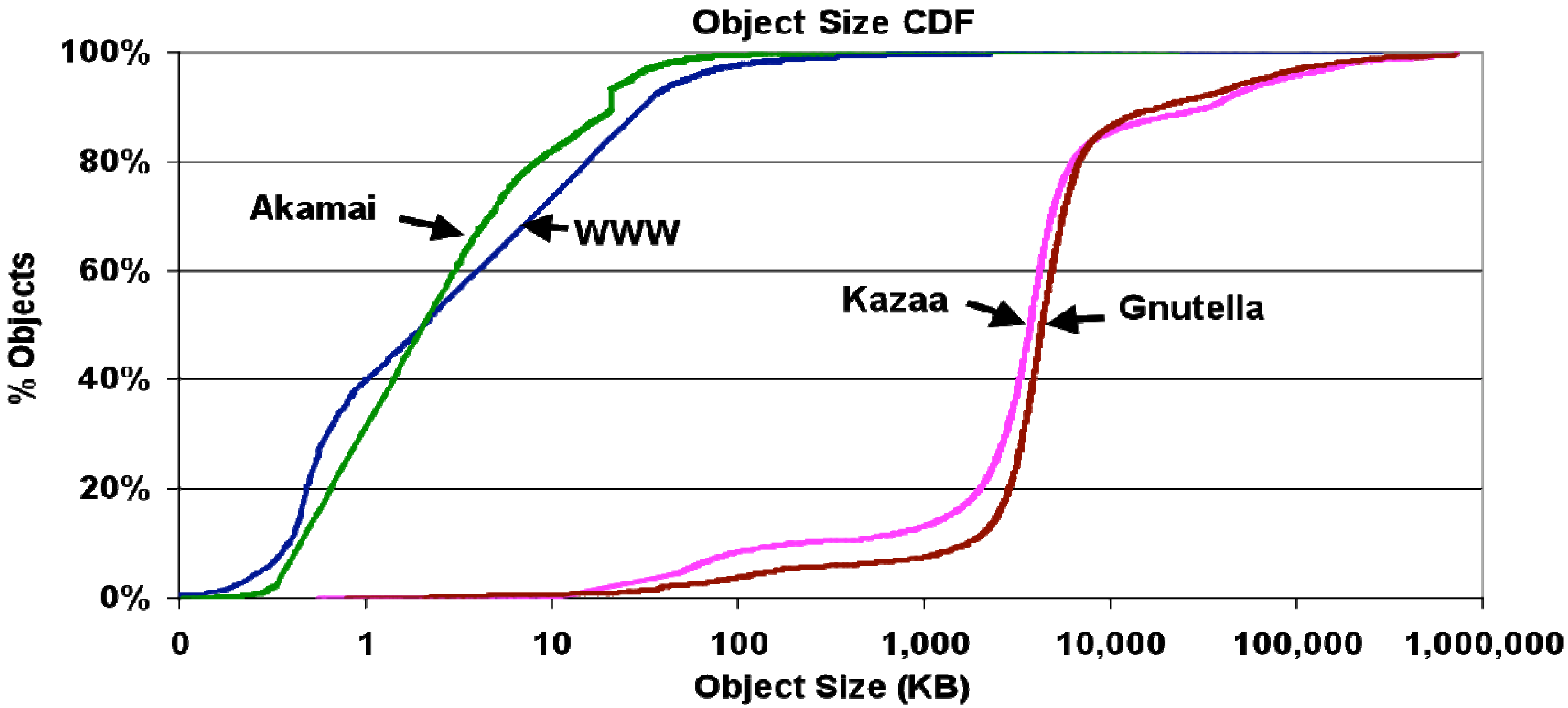


(b) By Bytes

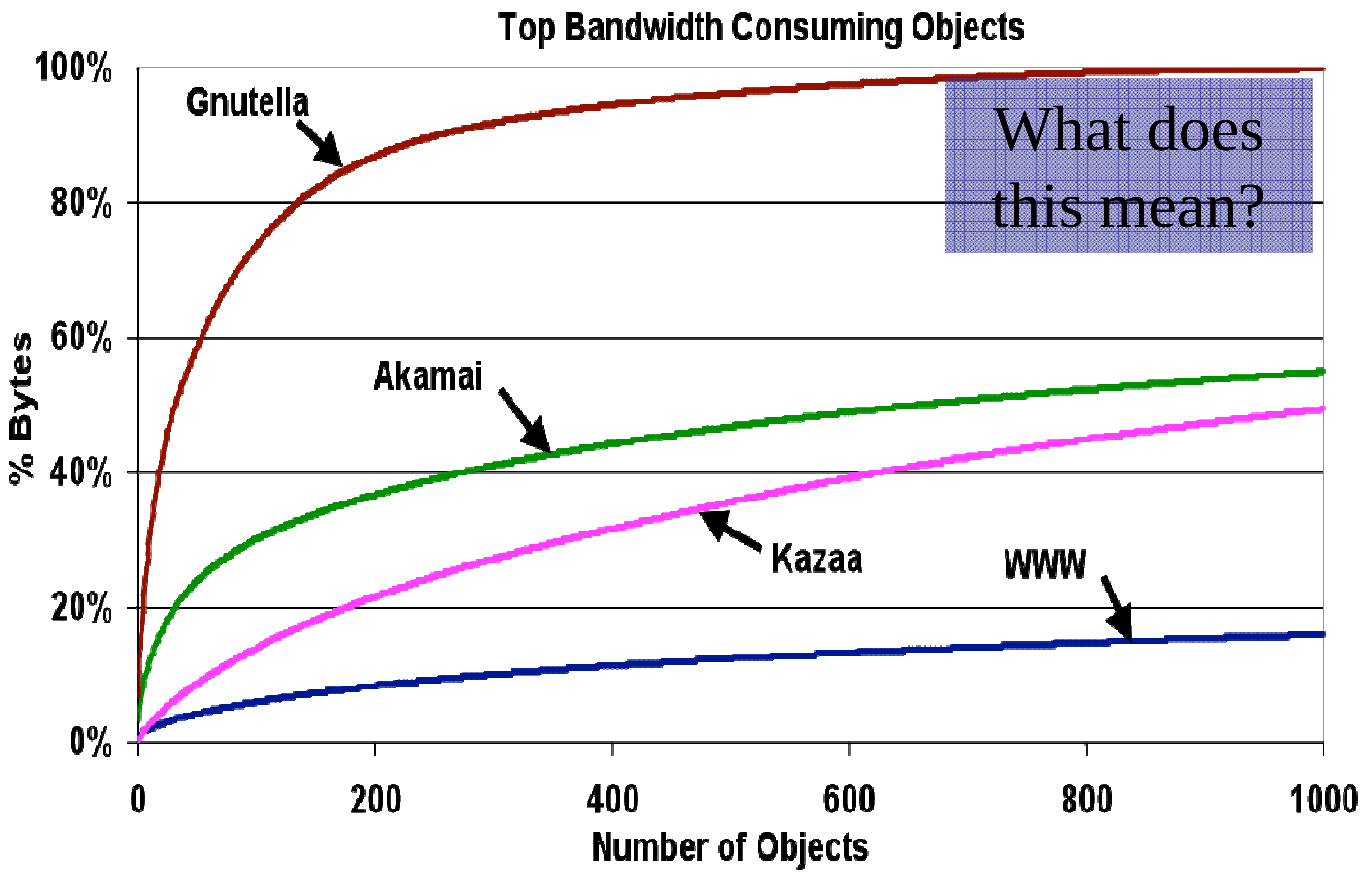


Object Sizes

UC Santa Barbara



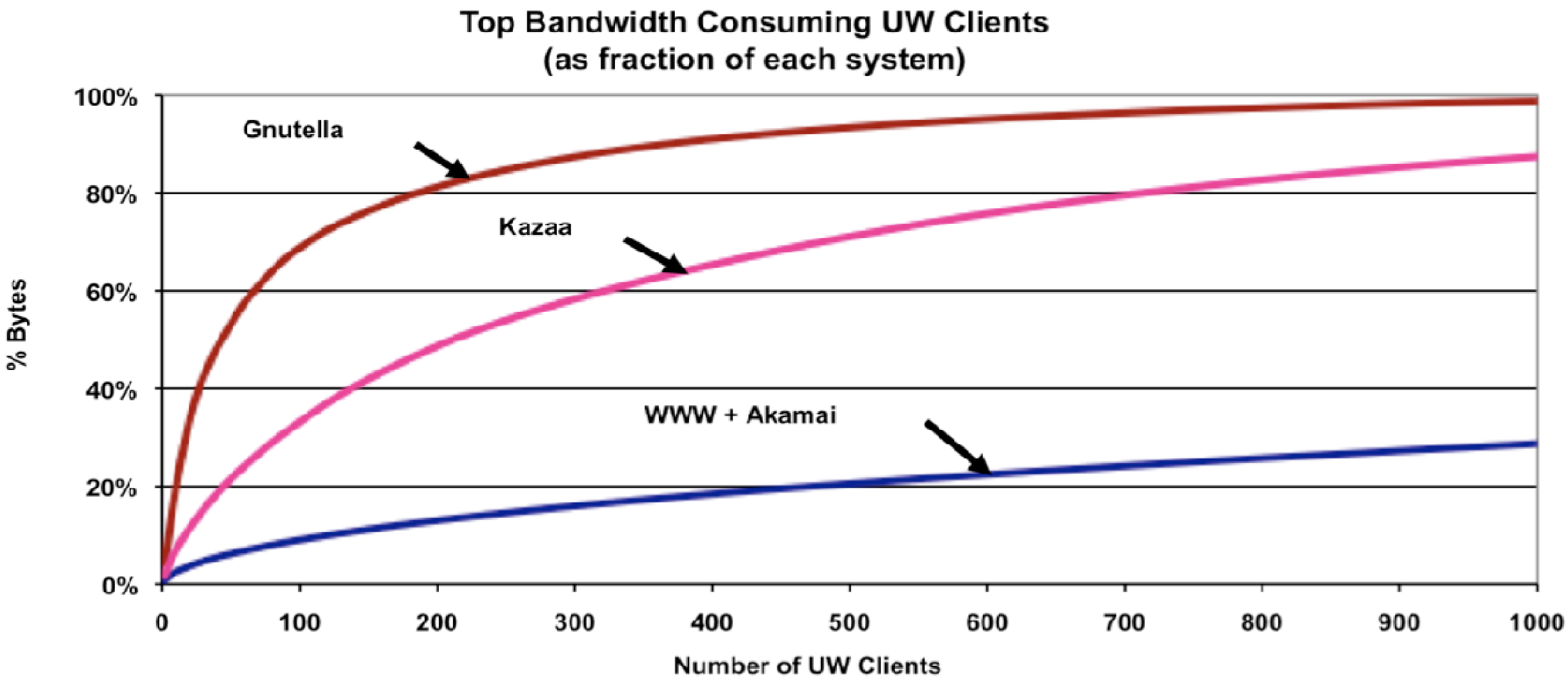
Top 1000 Object Popularity



VIEW 2: CLIENT VIEW

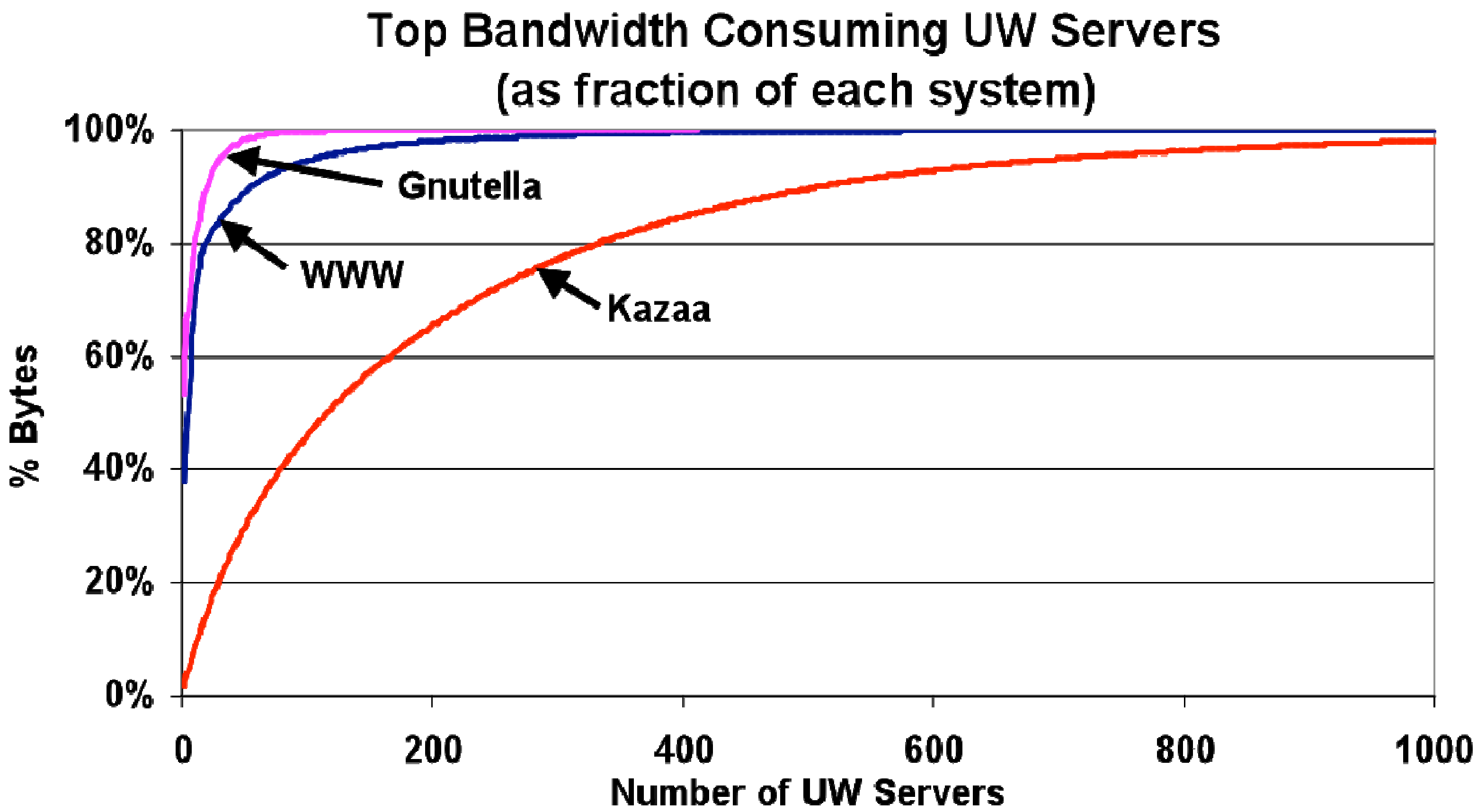
UW Client Allocation

UC Santa Barbara



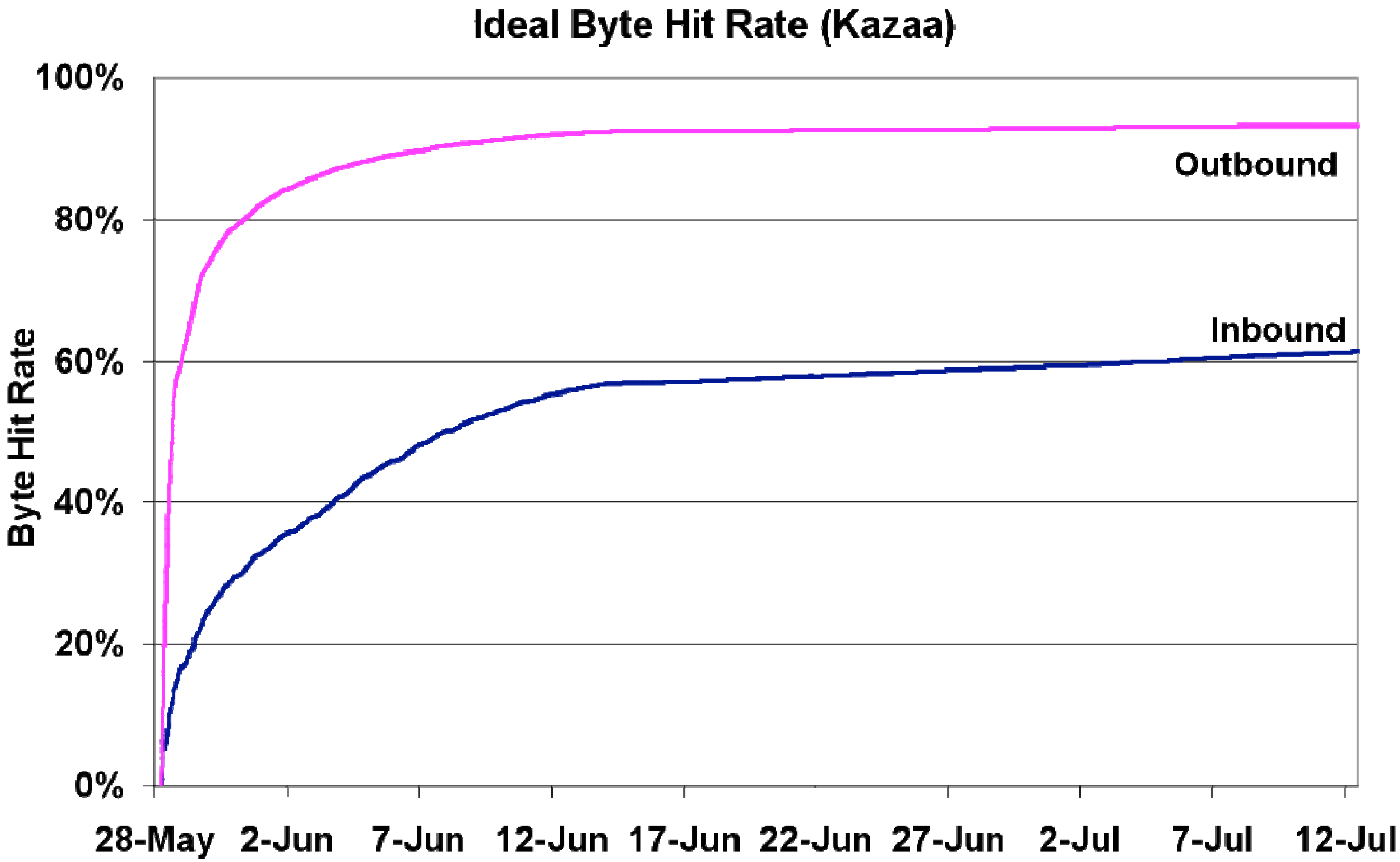
SERVER VIEW

UW Server Allocation



CACHING

Kazaa Caching



Criticisms

- UW traffic may not be representative
- What is all that unclassified traffic?
- Why are heavy P2P users called "worst offenders"?
- Does not include analysis of internal P2P traffic
  - Speculation that internal users receive much of their data already from internal users
- No suggestion on how to perform P2P caching
  - DPI and connection hijacking?
  - Protocol changes to support caches? Why not just prefer local network peers over remote peers (like BitTorrent now does)
- Why did they look at Gnutella?

FIN