Analysis of peer-to-peer systems: workload characterization and effects on traffic cacheability

Mauro Andreolini
University of Rome “Tor Vergata”

Riccardo Lancellotti
University of Modena and Reggio Emilia

Philip S. Yu
IBM T.J. Watson research center

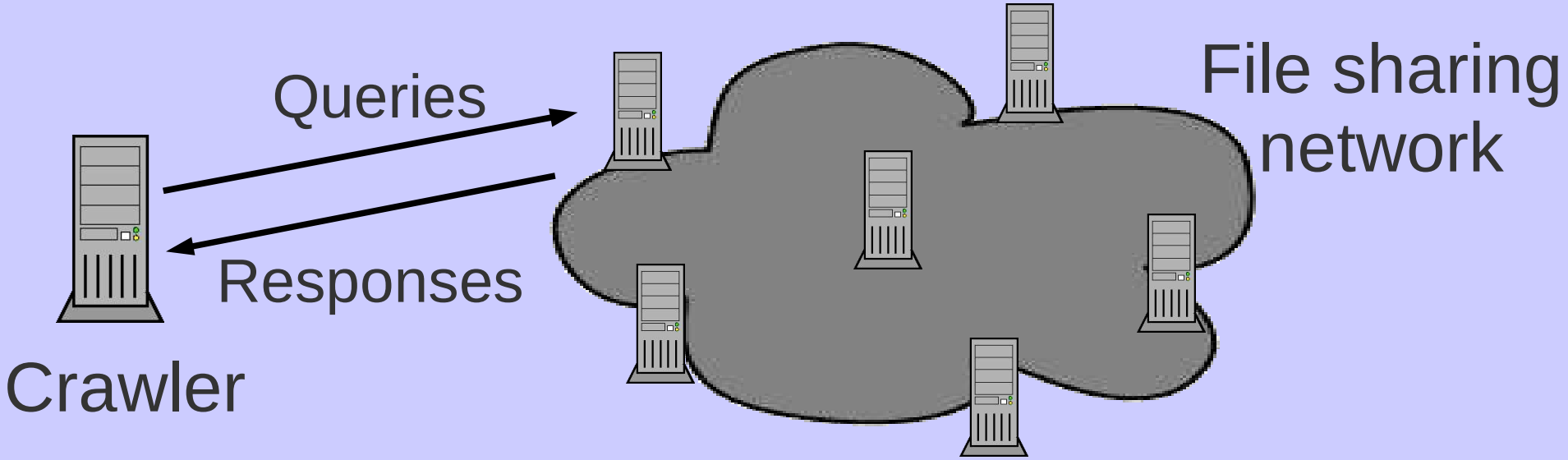
File sharing

- Killer application of peer-to-peer systems
  - More than 10^5 peers involved
  - More than 30% of Internet traffic is related to file sharing
- Not yet widely studied
- Our contribution:
  - Workload overview
  - Analytical models of some workload characteristics
  - Analysis of factors reducing cacheability

Experimental methodology

- Traffic interception
  - Analyzes actual file-sharing traffic
  - Needs representative traffic to analyze (e.g., backbone links)
- Crawling
  - Crawler sends queries and analyzes responses
  - Needs known protocols: Gnutella network
  - Does not need high traffic links
  - Different definition of some workload characteristics respect to packet Interception (e.g., resource popularity)

Overview of experiments

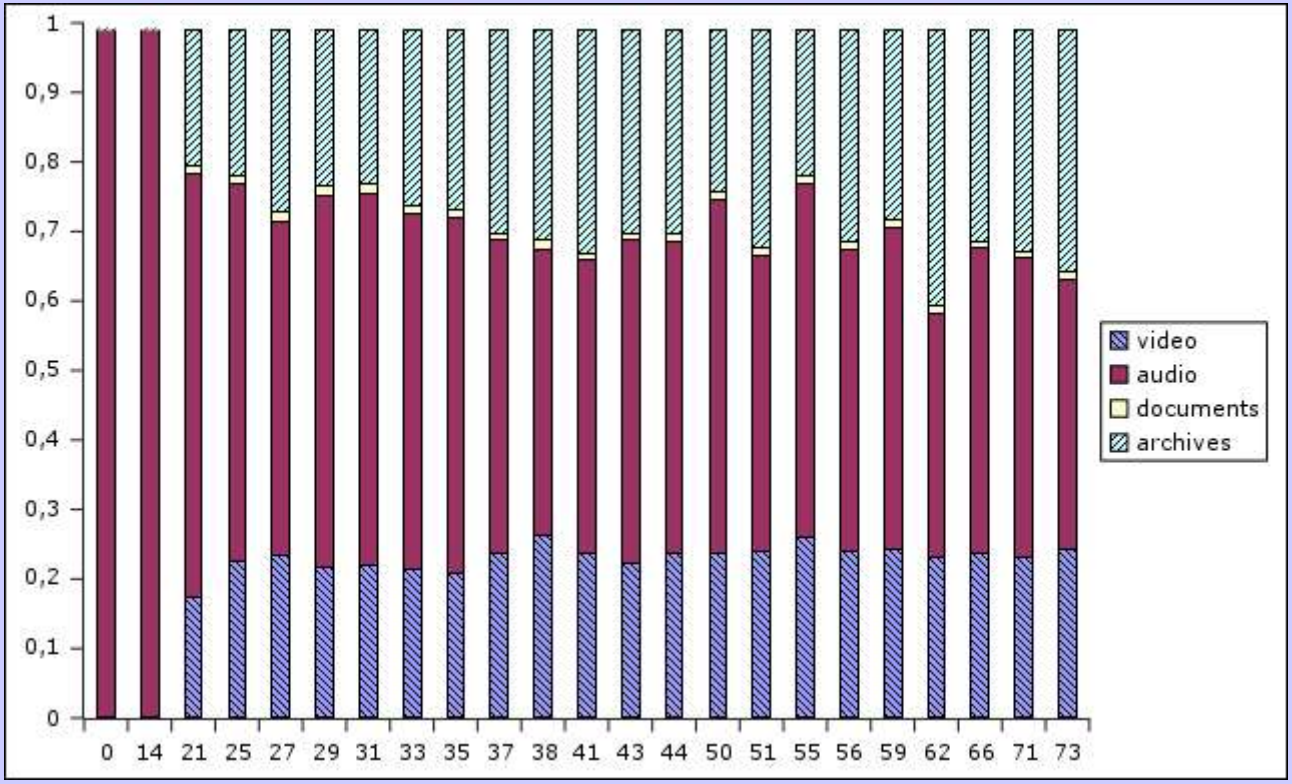


- Crawling for nearly three months (Aug-Oct 2003)
- Average of 78,900 nodes for each crawler run, with peaks >100,000 nodes
- Up to 1,500,000 resources per run
- File sharing ***is*** a killer application for P2P

Working set composition

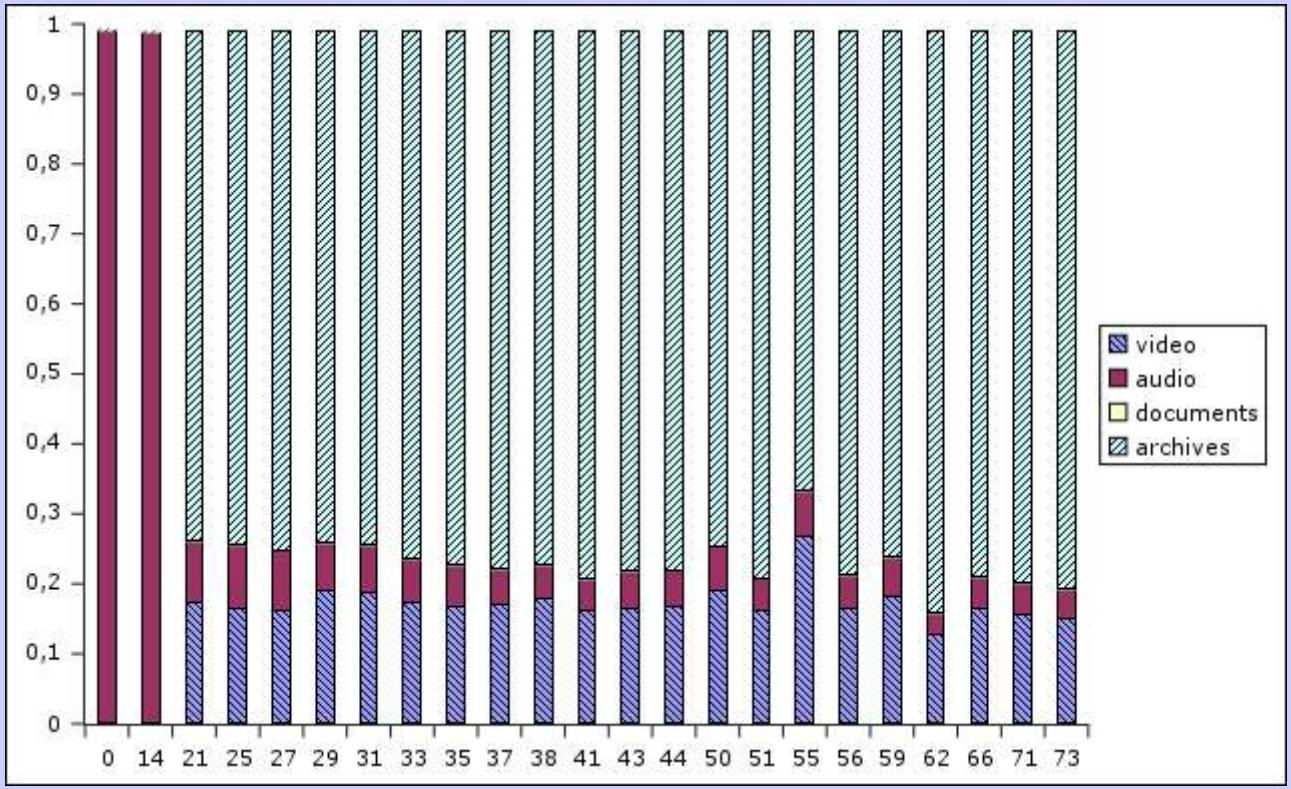
- 4 sets of resources
  - Video, Audio, Documents, Archives
  - Type identification based on filename extension
  - Sample downloads shows that extension is reliable to identify file type
- Results stable over time
- For each type we consider
  - shared resources
  - shared bytes

Working set composition by type



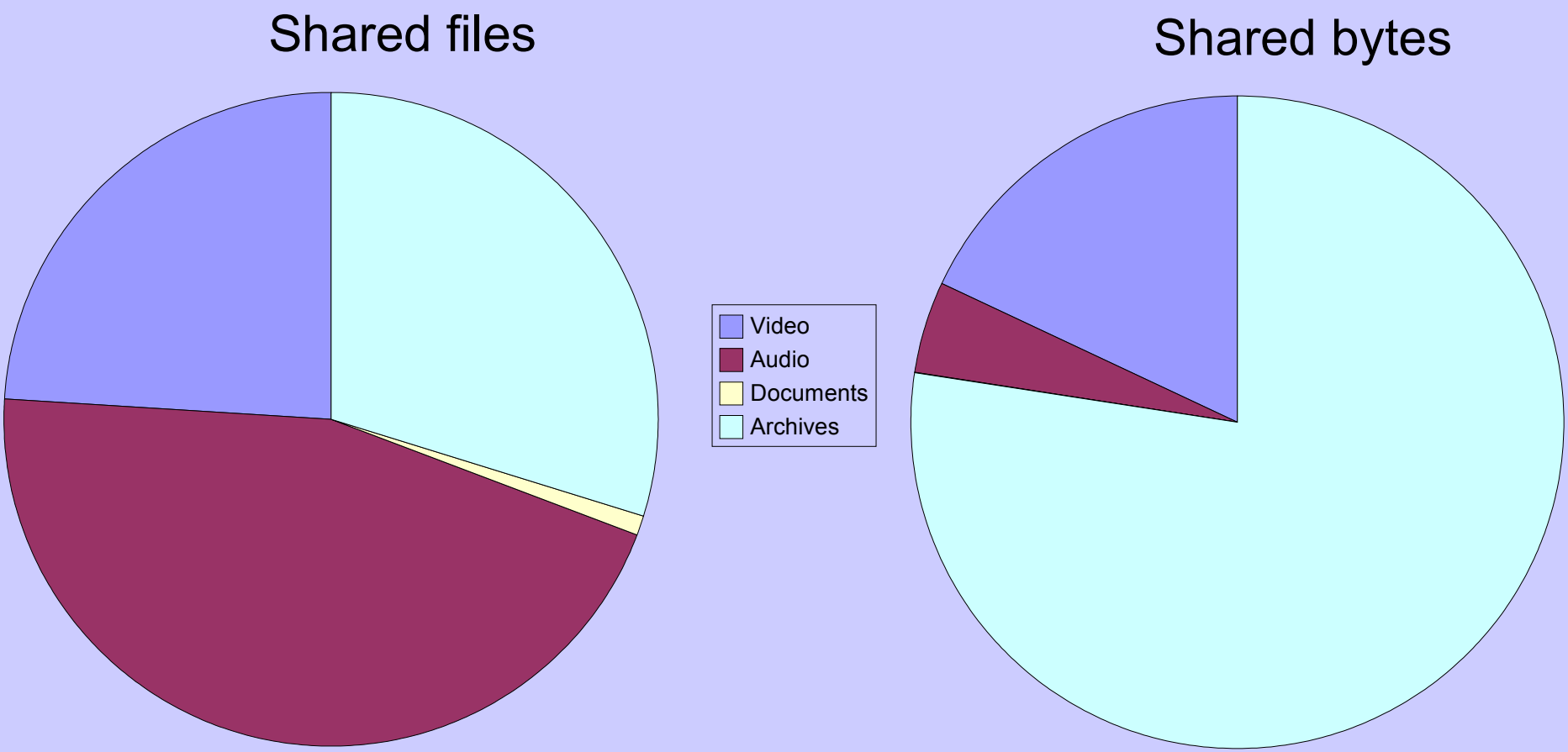
Audio clips accounts for the best part of shared files

Working set composition by type



Archives accounts for the best part of shared bytes

Working set composition by type



Our result confirms the observations of Leibowitz et al. ***(obtained through traffic interception)***

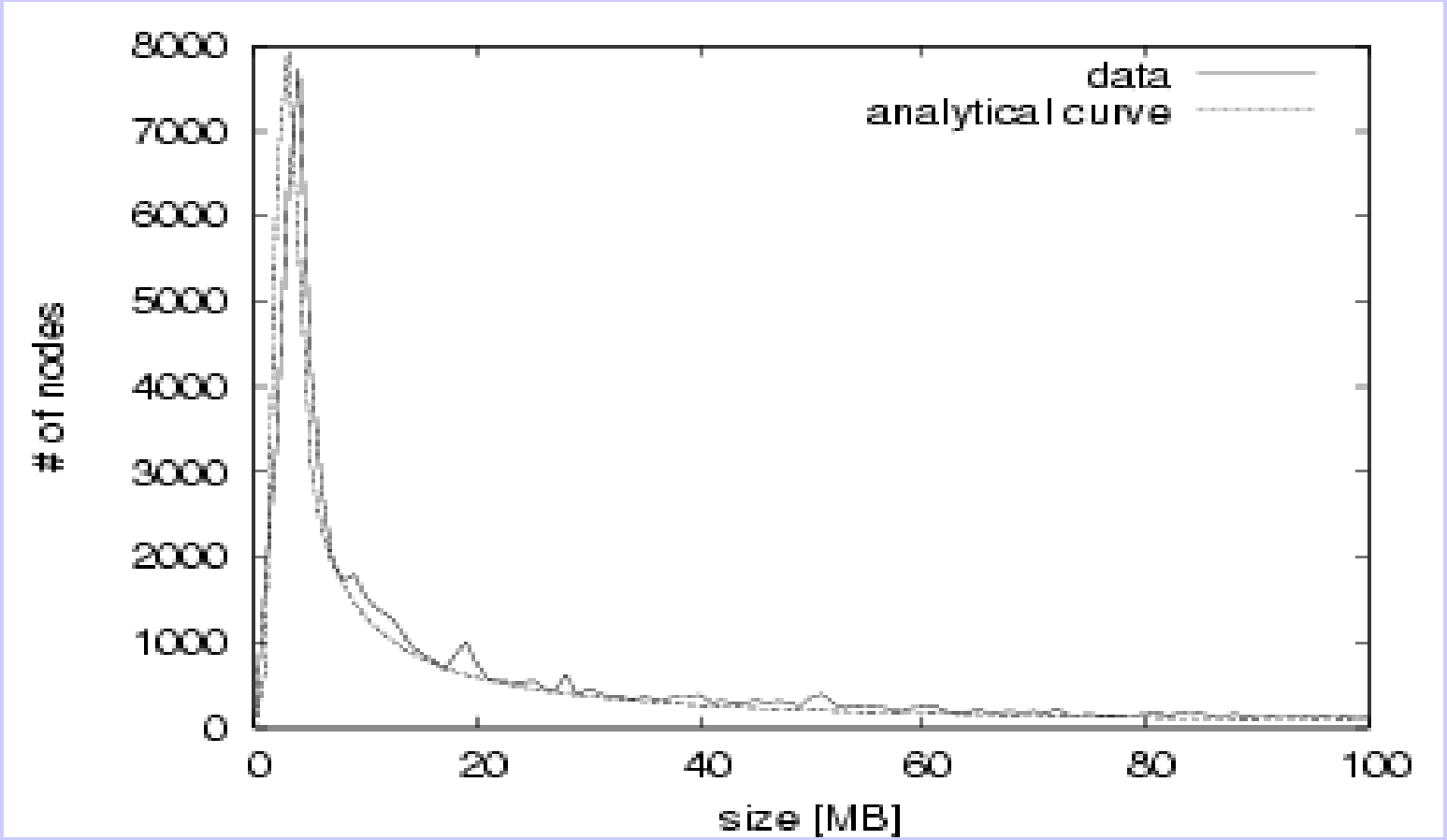
Analytical models

- Resource size according to type
  - Video and archives:
    - Heavy tailed size distribution
    - Lognormal body
    - Pareto tail
  - Audio and documents
    - Lognormal size distribution
    - ***non*** heavy tailed
- Volume shared by each node
  - Lognormal body, Pareto tail

Analytical models

Video	
Distribution	Lognormal if $x < 6$ MB, Pareto otherwise
Lognormal param.	$\sigma^2 = 1.23, \mu = 1.55$
Pareto param.	$a = 6.0, b = 0.12$
Audio	
Distribution	Lognormal
Lognormal param.	$\sigma^2 = 0.12, \mu = 1.42$
Document	
Distribution	Lognormal
Lognormal param.	$\sigma^2 = 2.38, \mu = 1.23$
Archive	
Distribution	Lognormal if $x < 10$ MB, Pareto otherwise
Lognormal param.	$\sigma^2 = 0.31, \mu = 1.00$
Pareto param.	$a = 5.98, b = 0.1$

Analytical models

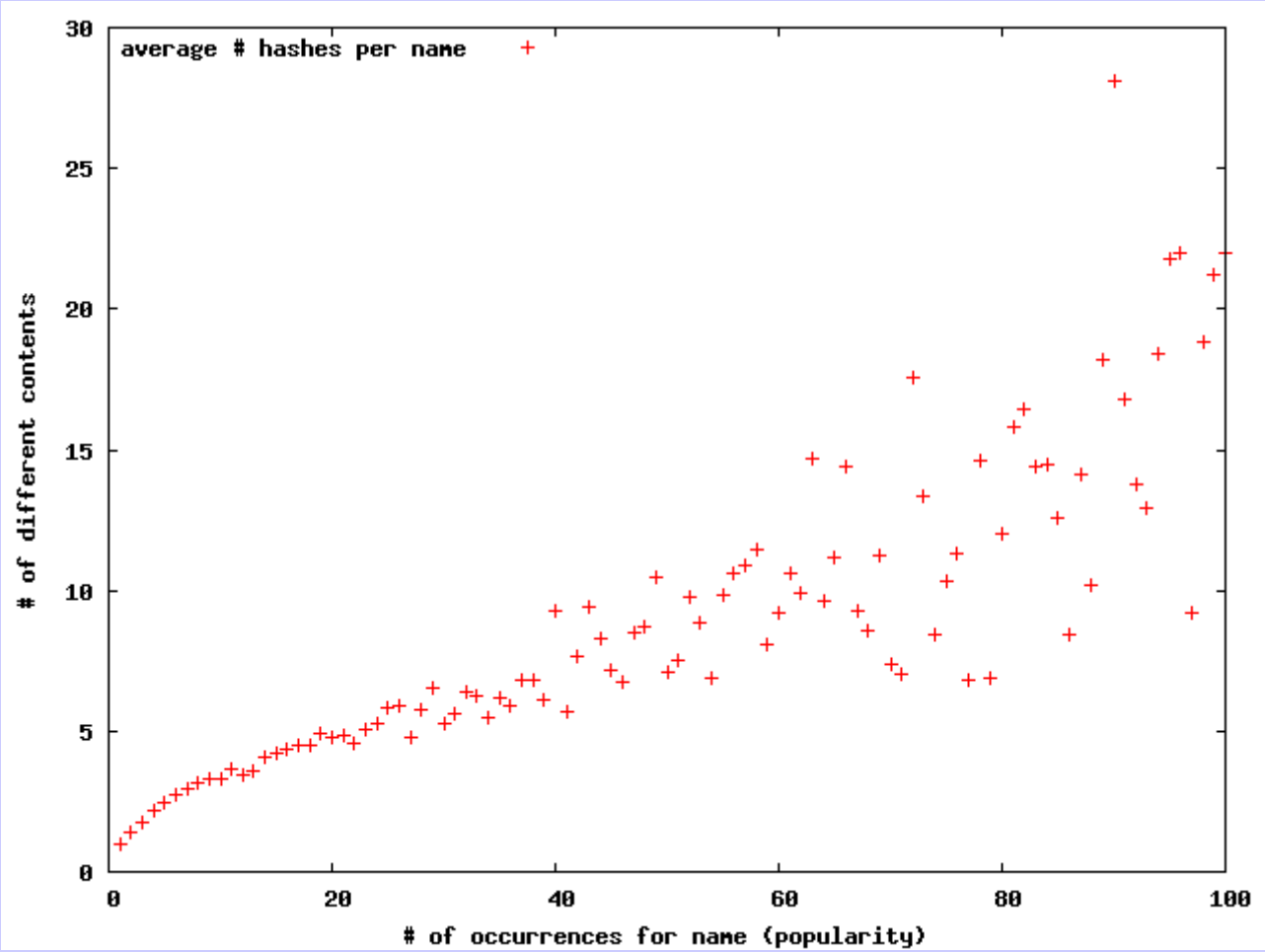


Volume of resources shared by each node

File sharing traffic cacheability

- Common belief:
  - ***"File sharing download is based on HTTP, hence we can use off-the-shelf Web caches"***
  - Not completely true
- Cache hit rate estimation should take into account two differences with Web traffic
  - ***Resource identifiers:***
    - File name
    - Hash code
  - ***Firewalled nodes with unroutable IP addresses***

Filename vs. Content hash

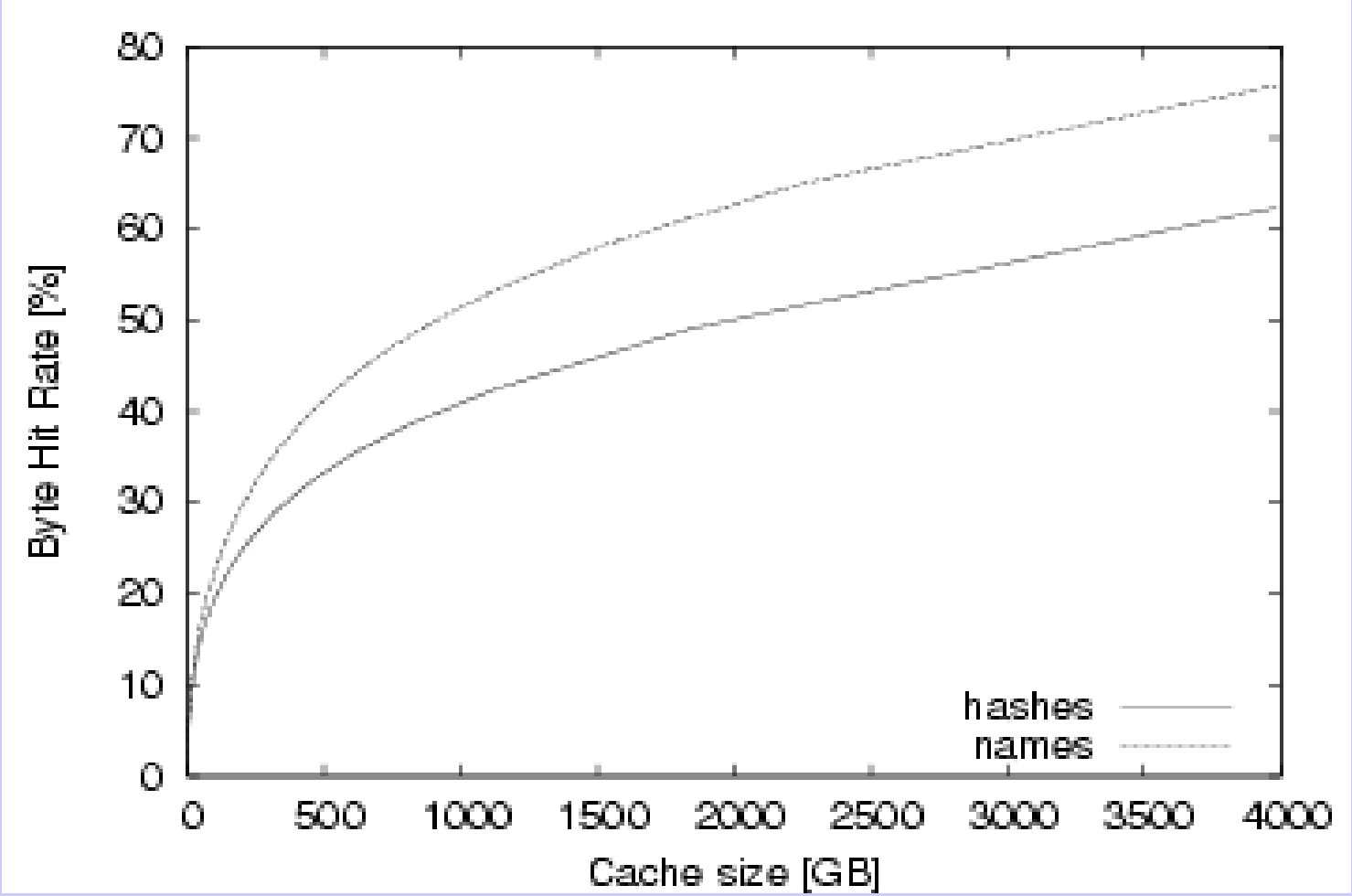


For popular resources the filename is not a suitable identifier: multiple files share the same name

Filename vs. Hash: Impact on cacheability

- Previous studies based on traffic interception used filenames as a resource ID
- Use of name as resource ID
  - Over-estimation of Zipf alpha parameter (popularity seems more skewed)
  - Under-estimation of working set size (with hashes we have a greater number of distinct resources)
- Cache hit rate seems higher

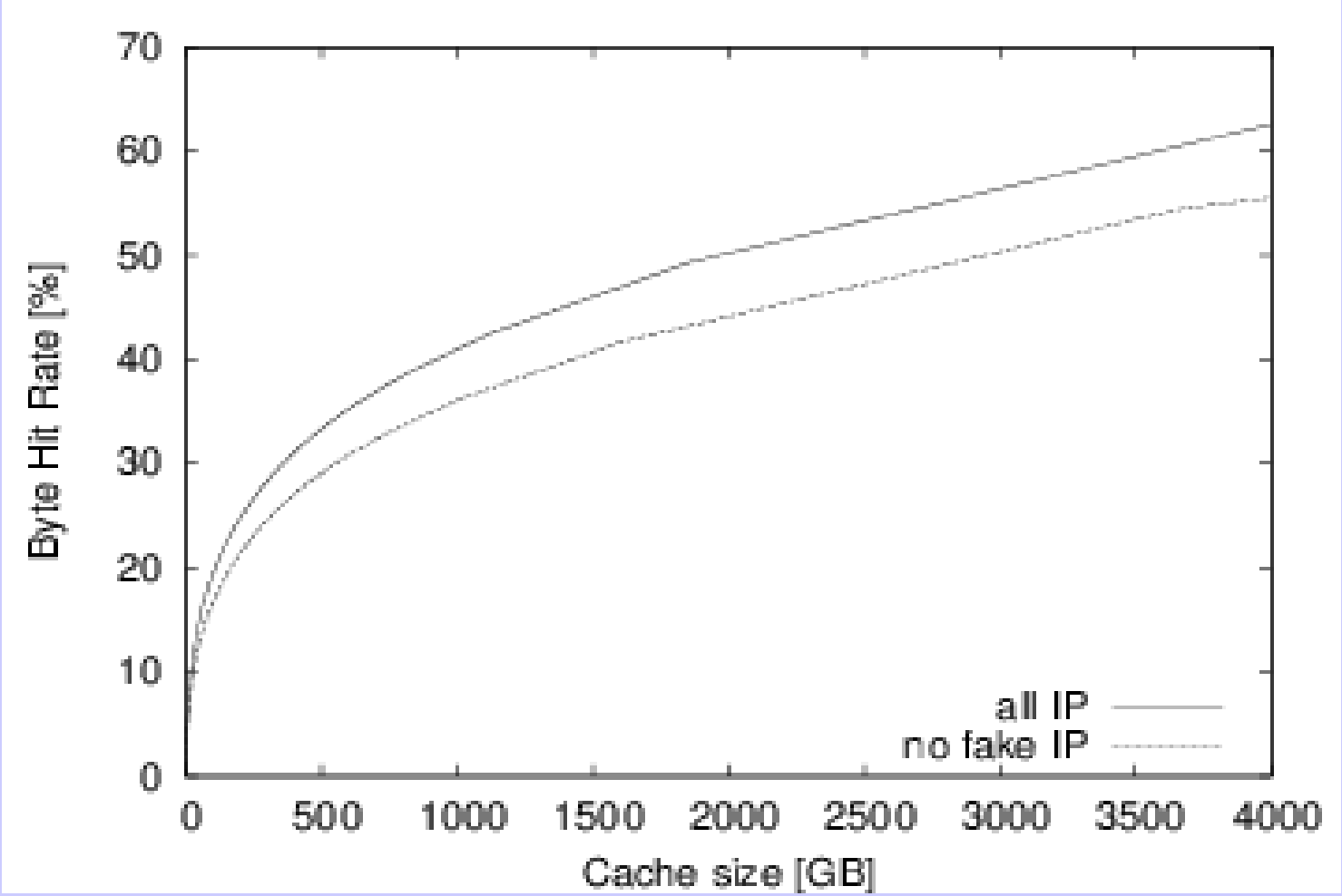
Filename vs. Hash: Reduction of cache hit rate



Non-routable IP addresses: Impact on cacheability

- Previous studies did not take non-routable IP addresses into account
- 10% nodes behind a firewall
- Download from these nodes needs a push-based mechanism which is not compatible with Web caching
- ***Resource on these nodes are not cacheable***
- Cache hit rate seems higher

non-routable IPs: Reduction of cache hit rate



Conclusion on cacheability

- File sharing traffic is cacheable
- Web caches ***need to be modified*** to take insto account file-sharing characteristics
  - Cache ***must*** consider also content hash (have to interact also with the query mechanism)
  - Cache ***must*** deal with push-based downloads

Open issues

- Comparison of data obtained through different methods
  - Crawling
  - Traffic analysis
- Study of time-related patterns at different ime scales:
  - Daily patterns
  - Weekly patterns
  - Yearly patterns

Analysis of peer-to-peer systems: workload characterization and effects on traffic cacheability

Mauro Andreolini
University of Rome “Tor Vergata”

Riccardo Lancellotti
University of Modena and Reggio Emilia

Philip S. Yu
IBM T.J. Watson research center