UNIVERSITY OF CAMBRIDGE
Computer Laboratory

Data Centric Networking (R202)

paper

Efficient Content Location Using Interest-Based Locality in Peer-to-Peer Systems

authors: K. Sripanidkulchai et. al. (CMU)

MPhil in ACS

reviewer/presenter: S. Trajanovski (*st508*)

Motivation

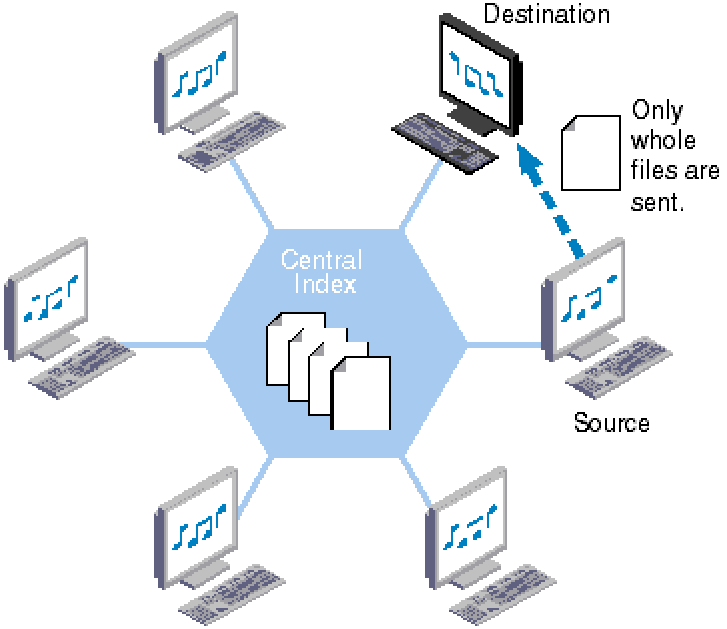
File seeking in P2P systems

- Challenges
  - file duplication
  - search algorithm
- Difference approaches
  - Centralized system (Napster)
  - Flooding (Gnutella)
- Both have weaknesses

Motivation

Centralized system (Napster)

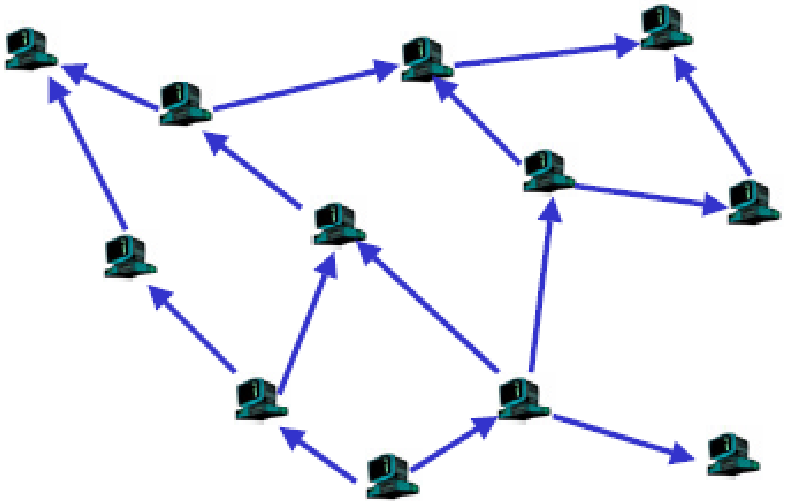
- Central Server
  - o one central node
  - o not in p2p sense
- Performance
  - o memory O(n)
  - o searching O(1)
- Resilience/Robustness
  - o just attack central node/server



Motivation

Massive flooding (Gnutella)

- Sending to the neighbours and so on ...
  - o first discovery
- Performance
  - o no indexing
  - o searching O(N)
- Features
  - ✓ robust
  - x scalable



Motivation/Proposal

How this could be improved?

- Starting point choice
  - o Gnutella
- Idea
  - o robust & simple
  - o improving scalability
  - o global solution
  - o main concept: **Interest - based locality**
  - o different from *popular/famous*

Proposal

Interest-based locality

- Building interest-based communities
  - usually exchange content
- Examples
  - networking (Van Jacobson, Crowcroft …)
  - mathematics (Tao, Perelman …)
  - politics (Obama, Merkel …)
- Counter examples
  - Golf or cricket players for *ME*

Proposal

The solution

- Architecture
  - o overlay on Gnutella network
  - o communities
- Entities
  - o shortcuts (additional links)
- Scenario
  - o 1st: try to find in the interest group
  - o 2nd: try in Gnutella

Proposal

The solution

- Shortcuts
  - keeping the limited list (up to 10)
  - priority links
- Shortcut list ranking scheme
  - content probability
  - path latency
  - available bandwidth
  - combination

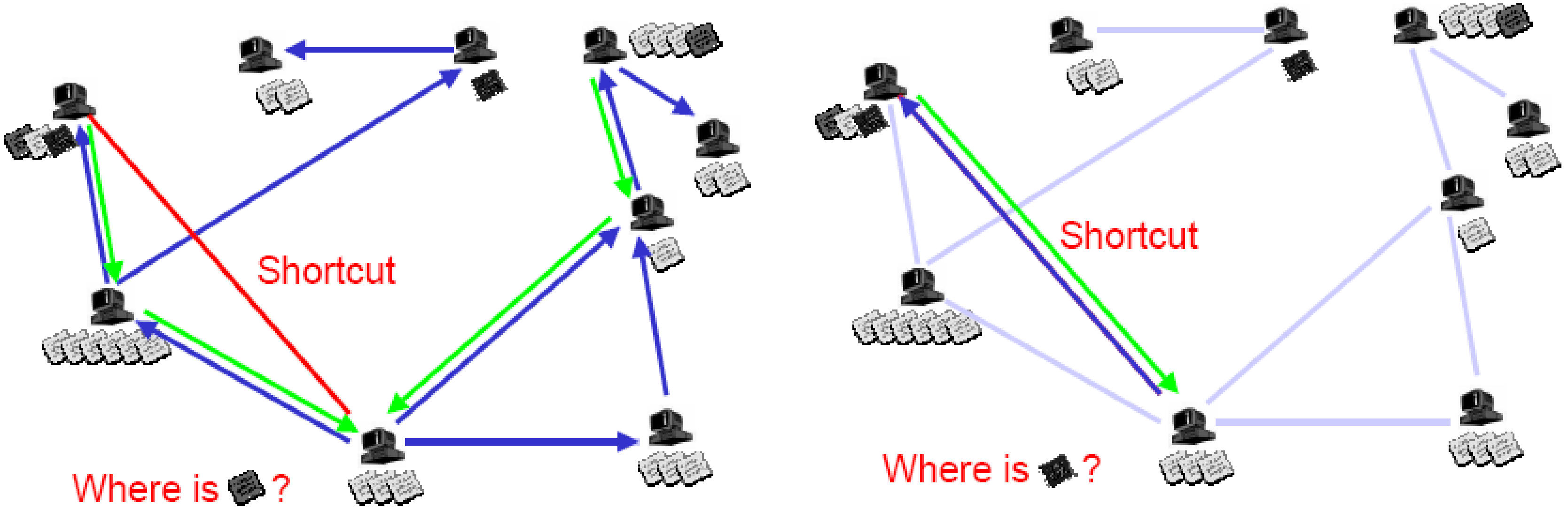
Proposal

The solution

- Node (peer) addition
  - initial flooding (Gnutella like)
  - forming the list (1 per time)
- Later scenario
  - refining the list dynamically
  - some peer introduced, another removed
- Applicable generic solution
  - other mechanisms (e.g. Kazaa)

Proposal

Usual scenario



(a) without (b) with

shortcut

Performance evaluation

Participants

- What is used?
  - o different data traces
  - o data from different sources
- How?
  - o methodology
- Why?
  - o Better understanding of the model
  - o proof for improvement

Performance evaluation

- Gnutella content location
    - TTL mechanism
    - avoid query duplication
- Performance pointers
    - success rate
    - load characteristics
    - query scope
    - minimum reply path lengths
    - additional states

Performance evaluation

Methodology

- Query workloads
  - o different data traces
  - o data from different sources
    - - Boeing
    - - Microsoft
    - - CMU web
    - - CMU Gnutella
    - - CMU Kazaa

Performance evaluation

Methodology

- Gnutella connectivity graph
  - o using Gnutella topology
  - o fitting to particular query workload
    - - one with similar number of nodes
    - - deleting nodes
    - - degree distribution
    - - max TTL = 7

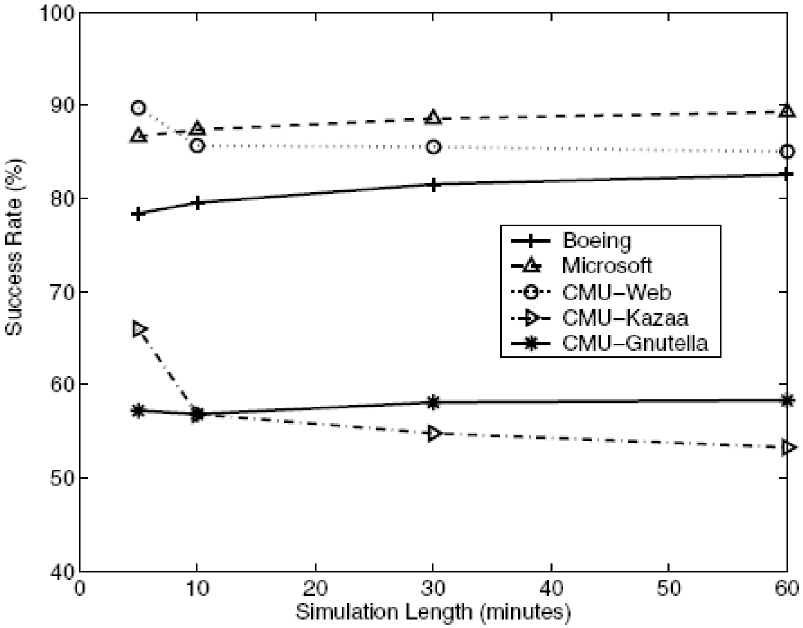
Performance evaluation

Storage and Replication models

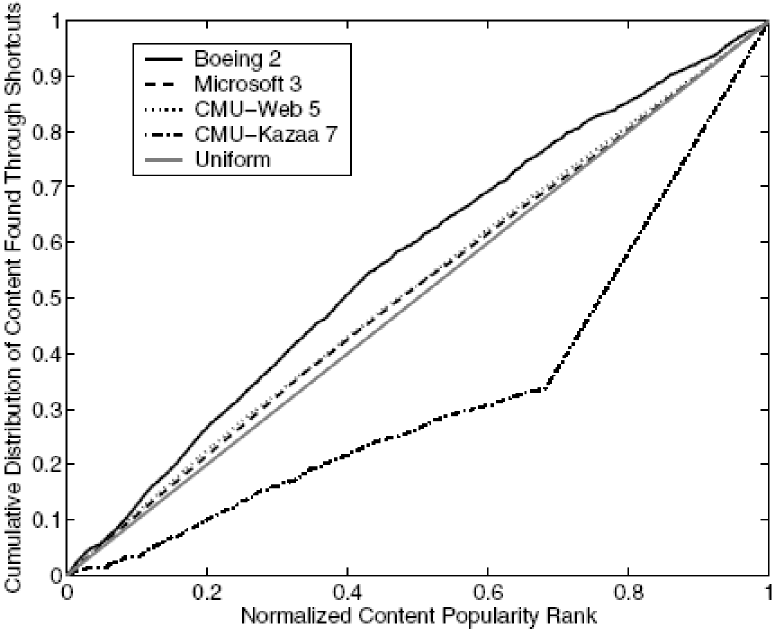
- Web traces
  - o all clients participate
  - o after downloading the file, peer has it
  - o no dynamic content
- CMU Kazaa and Gnutella traces
  - o clients and peers
  - o after downloading the file, peer has it
  - o no dynamic content

Experimental results

Shortcuts Gnutella vs. pure Gnutella



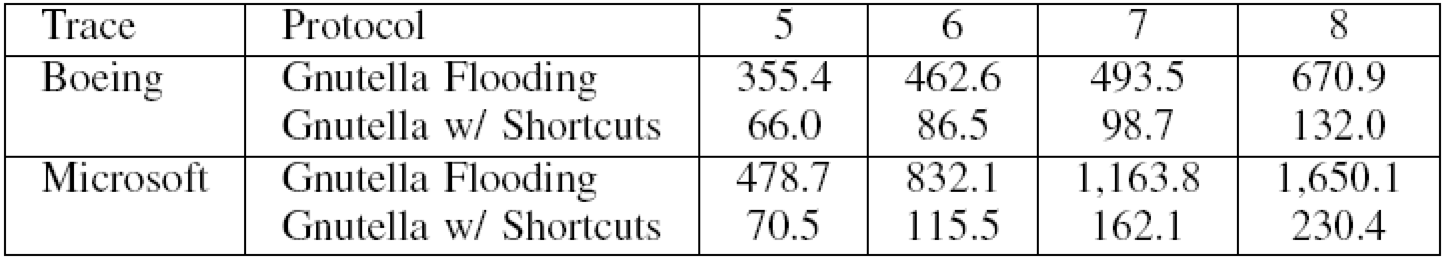
(a) success rate



(b) shortcuts target?

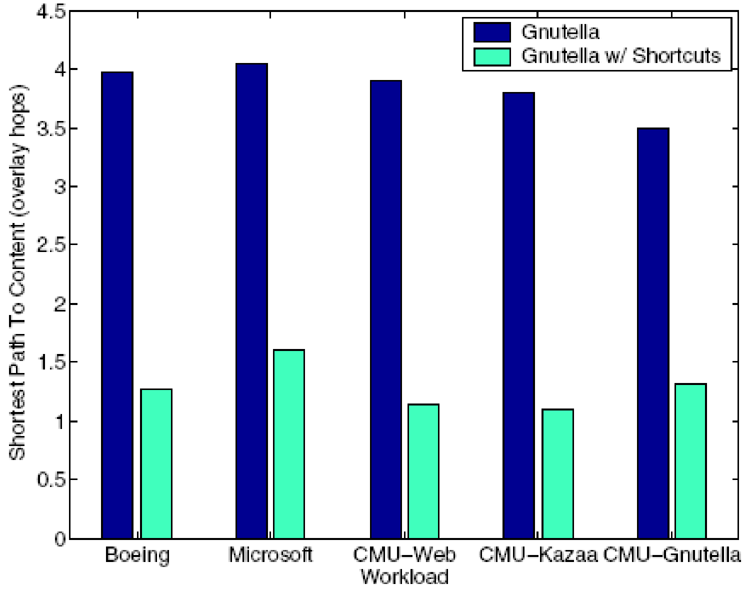
Experimental results

Shortcuts Gnutella vs. pure Gnutella



Trace	Protocol	5	6	7	8
Boeing	Gnutella Flooding	355.4	462.6	493.5	670.9
	Gnutella w/ Shortcuts	66.0	86.5	98.7	132.0
Microsoft	Gnutella Flooding	478.7	832.1	1,163.8	1,650.1
	Gnutella w/ Shortcuts	70.5	115.5	162.1	230.4

(a) load/packet



(b) shortest path/hops

Performance evaluation

Possible improvements/changes?

- change all (more shortcuts/time & unlimited list)
  - good performance (CMU Kazaa, Microsoft)
  - implementation difficulties
  - changes one property, maybe !?
- search in shortcuts' shortcuts
  - slightly improved performance (rate/loads)
  - increased shortest path

Additional evaluation

Understanding interest-based locality

- properties/structure
  - small-world behavior
- web pages vs. web objects (files)
  - fairly better than pure Gnutella
- objects from different publisher?
  - capture interests across multiple publishers

Related work

.. different from Gnutella

- query caching
- Ring searches
  - o minimize random walks
  - o effective for finding popular content
- Kazaa
  - o super-nodes
  - o possible Kazaa’s improvements (routing, loads)
- YouServ, BitTorrent, Squirrel

Conclusion/Summary

- Pros
  - o evaluated improvements of
    - - web contents (song, movies,..)
    - - p2p systems
  - o simple method (heuristic)
  - o increased scalability
- Cons
  - o possible congestion in shortcuts
  - o non semantic matching (similar files)

- Questions??

- Discussion ..