Adaptive Data Dissemination and Content-Driven Routing in Peer-to-Peer Systems

MSc Thesis Defense

Thomas Repantis

Department of Computer Science and Engineering

University of California, Riverside

trep@cs.ucr.edu

http://www.cs.ucr.edu/~trep

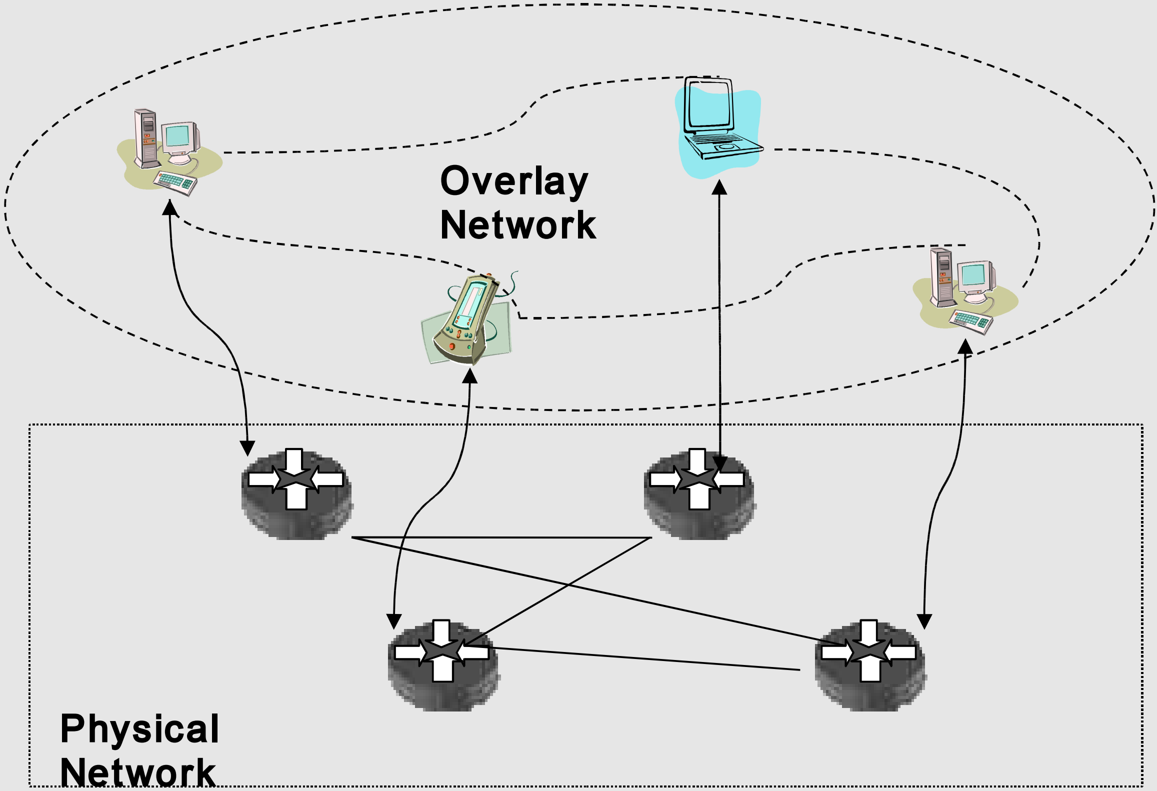
Outline

1. Introduction and Background
2. System Description
   – Bloom Filters
   – Content-Driven Routing
   – Adaptive Data Dissemination Strategies
     • Immediate Local
     • Adaptive Local
     • Adaptive Local Remote
3. Experimental Evaluation
4. Conclusions and Future Work

Peer-to-Peer Systems

Large-scale distributed systems that enable the sharing of computer resources between autonomous nodes

- Consist of nodes acting as both clients and servers
- Allow the nodes to form an overlay over the physical network
- Nodes can be geographically distributed, dynamic, and heterogeneous



Peer-to-Peer Success Stories

Advantages of peer-to-peer systems:

- **Resource aggregation** (cost effective)
- **Increased autonomy** (self-organizing network)
- **Improved scalability** (no central coordinator)
- **Improved reliability** (no single point of failure)

P2P has been successfully used for:

- **File Sharing** (e.g. Gnutella, Chord, Oceanstore)
- **Distributed Computing** (e.g. Seti@, Folding@)
- **Web Caching** (e.g. SQUIRREL based on Pastry)
- **Online Gaming** (e.g. Sony Playstation)
- **Spam Detection** (e.g. SpamNet)

Challenge

How to efficiently locate data or services in a large-scale, unstructured P2P network

- No central manager can have an accurate global view of the system' s contents
- The environment is dynamic, as peers join, leave, fail without a priori notification
- The peers are heterogeneous and restricted in their processing and communication capabilities
- The amount of traffic generated by queries can be overwhelming

Current Solutions

- Structured overlays (e.g. OceanStore, Chord, CAN, Pastry)
- Hybrid overlays (e.g. Gnutella 0.6 UltraPeers)
- Probabilistic query forwarding (e.g. Random Walkers, Routing Indices, ISM)
- Searching and data dissemination through gossipping (e.g. Planet-P, P-Grid, CUP)
- Content summarization (e.g. SummaryCache, Breadth and Depth Bloom Filters)

Problem Statement

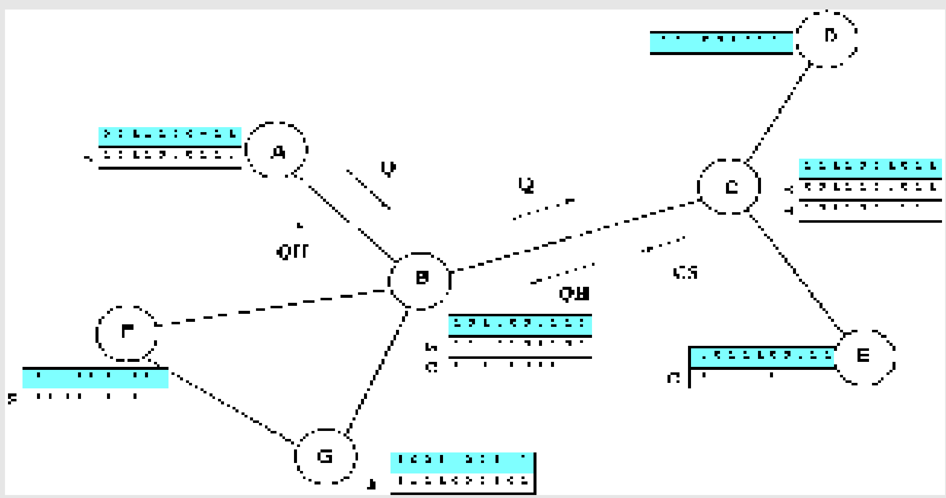
- How to build a system to locate content in a fast, efficient, and decentralized manner in a large-scale, unstructured P2P network
- Our approach:
  - Adaptive data dissemination and content-driven routing to intelligently route queries in the P2P network
  - The goal is to reach the users with the relevant content while keeping the number of propagated messages small

Contributions

- We employ a content-driven routing mechanism for locating objects, utilizing content synopses of the peers data
- We propose adaptive data dissemination algorithms for propagating the content synopses, based on the requests and replies generated by the peers
- We simulate large-scale networks and show the efficiency and scalability of our approach

Peer-to-Peer Overlay

- Each node has a unique global id and is responsible for keeping a small number of connections to other peers
  - The network is unstructured and self-organizing; peers join the overlay by establishing a connection with any peer in the overlay; no explicit membership is required
  - Number of connections depends on available resources
  - Each node maintains synopses of the content of its peers and uses them to route queries

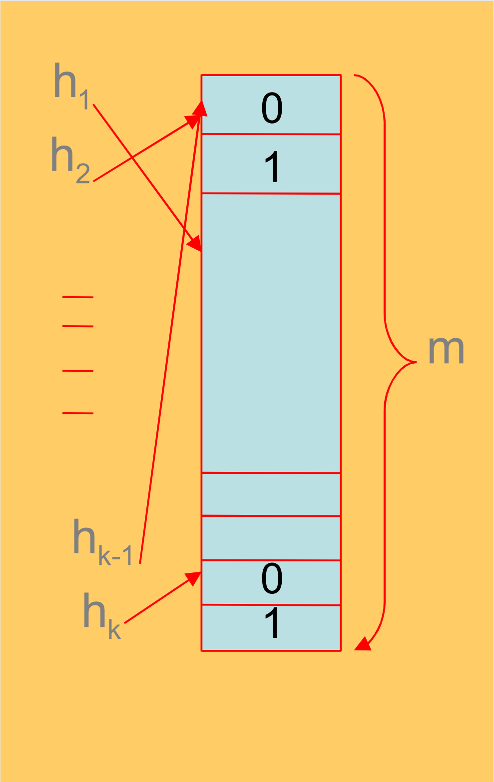


System Model

- Each peer p has n objects $S_p = a_1, a_2 .... a_n$
- A Bloom Filter BF_p of length m is used to represent the set of objects S_p
  - We assume k hash functions $h_1(s), h_2(s), .. h_k(s)$
  - Reduce memory and bandwidth demands – depending on the Bloom filter size
  - Tradeoff between accuracy and resource usage
- Each peer stores 2 types of Bloom Filters: a local filter for objects available locally at the peer and remote filters for objects stored in remote peers

Bloom Filters

- An efficient, lossy way of describing sets of data
- A Bloom Filter data structure represents the set of data by using an array of bits; each bit takes a value of one or zero
- To check if an object is in the set
  - Object is hashed
  - Corresponding bits of the filter are checked
    - If at least one of the bits is not set, object not in the set
    - If all bits are set, object may be in the set (false positive)

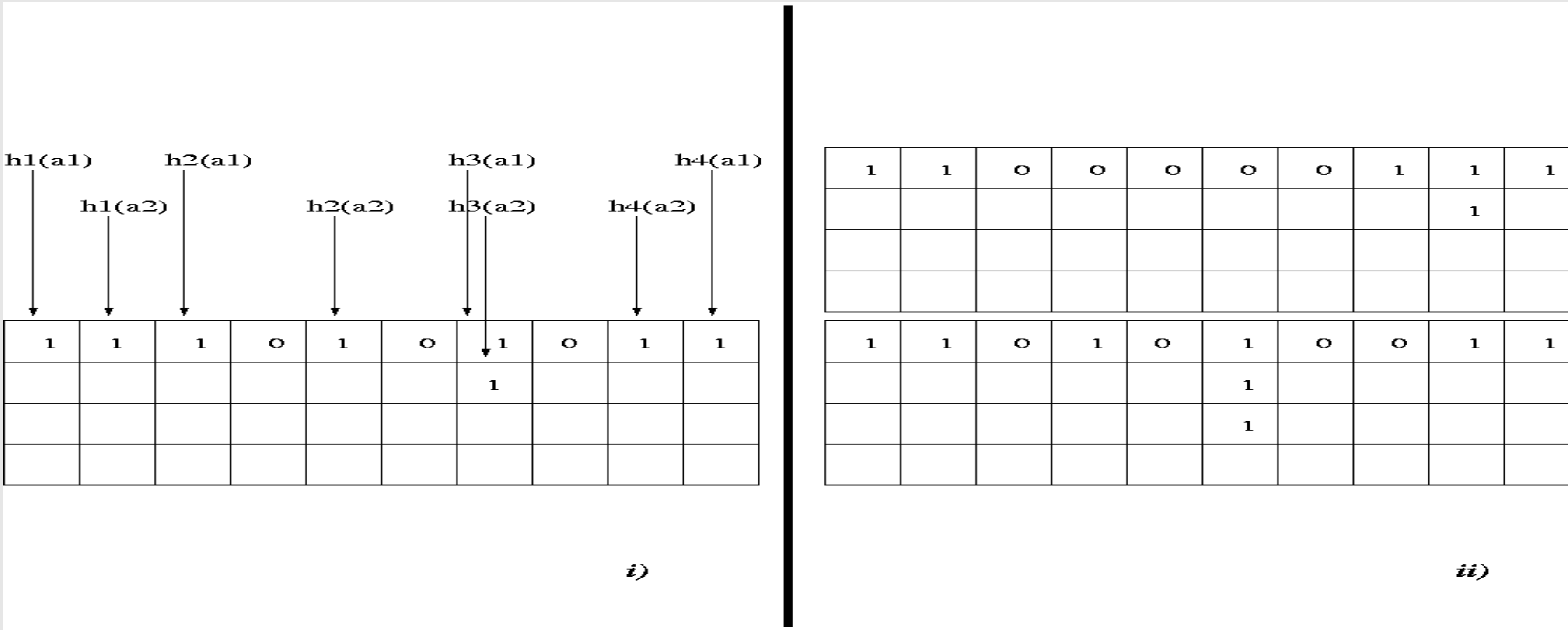


$$p_{err} = \left(1 - p_0\right)^k = \left(1 - \left(1 - \frac{1}{m}\right)^{kn}\right)^k \approx \left(1 - e^{-\frac{kn}{m}}\right)^k$$

$$(1/2)^k = (0.6185)^{\frac{m}{n}}$$

n: number of objects
m: Bloom Filter width
k: number of hash functions

Counting and Multi-level Bloom Filters



- Counting Bloom filters: Counters keep track of the number of objects corresponding to each bit of the filter
- Multi-level Bloom filters: The filter of each level is appended to that of the previous level

Content-Driven Routing

- Each peer stores its own content synopsis and also the content synopses of other peers
- When a peer receives a query, it performs a local search
- If results are not found, then it searches the contents of its peers
- The query message is propagated to those peers whose synopses indicate that they contain the requested object
- These are the peers with the highest probability of actually containing the object
- If no synopsis returns a hit, the query is forwarded to a random set of immediate peers

Adaptive Dissemination Strategies

Nodes employ adaptive data dissemination strategies to propagate content synopses to other peers

- IL: propagates its local synopsis to all peers one hop away.
- AL: propagates its local synopsis to selected immediate and remote peers.
- ALR: propagates both its local and stored remote synopses to selected immediate and remote peers.

Disseminate Local Content Synopsis to Immediate Peers (IL)

- Each peer sends its local content synopsis to all its immediate peers
- The queries are routed based on the content synopses of the immediate peers
- If no matches are found, the protocol resorts to randomly choosing peers
- Strategy is simple, but of limited use; only a small number of content synopses are examined

Disseminate Local Content Synopsis to Peers Selected Adaptively (AL)

- Each peer sends its local content synopsis to a selection of peers
  - The recipients of content synopses are among immediate and remote peers whose queries have been answered in the past
- The queries are routed based on the local content synopses of peers
- Local content synopses not only of immediate peers, but also of peers one has previously interacted with
- This strategy aims to make content synopses available to peers that have high probability of using them in the future

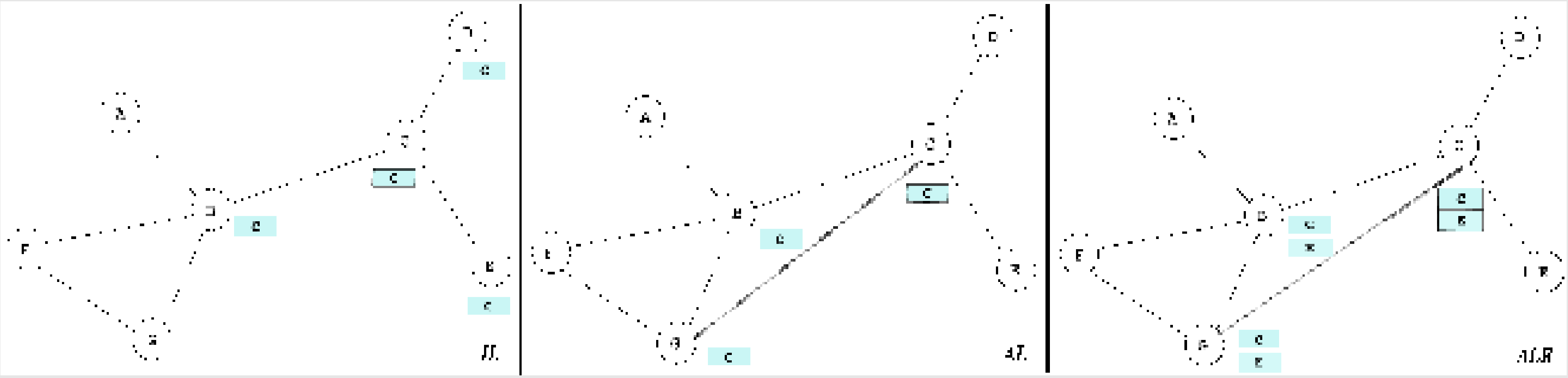
Disseminate Local and Remote Content Synopses to Peers Selected Adaptively (ALR)

- Each peer sends its local content synopsis together with remote content synopses it has stored to a selection of peers
- The queries are routed based on both local and remote content synopses
- This strategy enables the peers to examine more content synopses before routing queries
- The amount of information transferred between nodes is higher than the previous strategies

Adaptive Synopses Dissemination Parameters

- The AL and ALR dissemination strategies take into account the following parameters when disseminating the synopses:
  - The number of queries a node has received by a peer, and their frequency (indication of peers that route a lot of traffic)
  - The number of replies a node has provided a peer with, and their frequency (indication of the popularity of the stored objects among specific peers)
  - The number of connections a node maintains (indication of whether a peer is a hub or a leaf)

Synopses Dissemination Strategies



- IL: C propagates only its local synopsis to all peers one hop away.
- AL: C propagates its local synopsis to selected immediate and remote peers.
- ALR: C propagates both its local and stored remote synopses to selected immediate and remote peers.

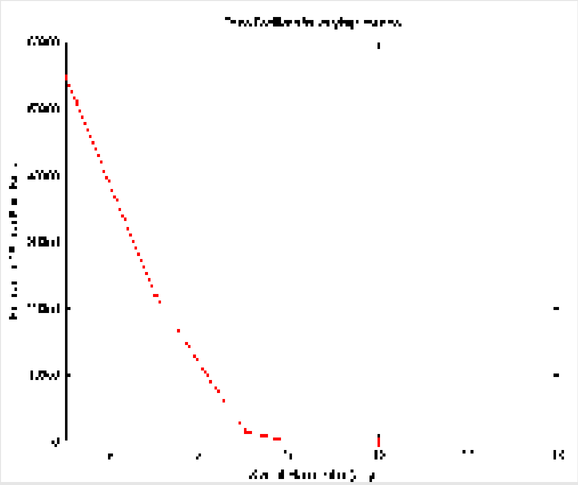
Implications of Dynamic Behavior

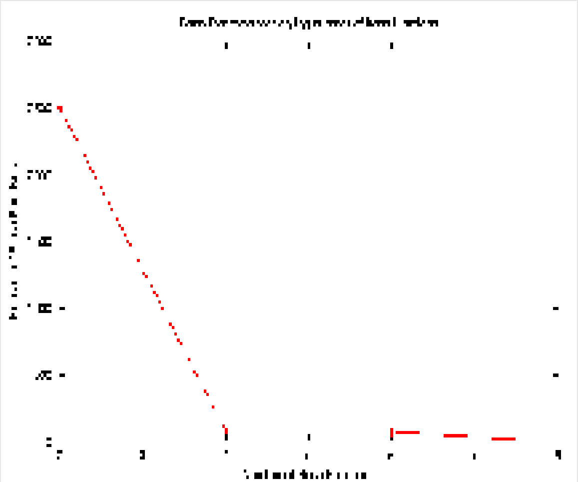
- In a dynamic P2P system, nodes may disconnect or join independently
- Content synopses must be updated when:
  - The content changes (objects inserted, deleted or modified)
  - Connections are established or dropped
- Synopses may be pushed or pulled
  - Push synopsis to other peers as soon as a peer is connected will result in replies with their content synopses
  - Pull synopses from other peers only when a query needs to be answered or routed (useful in highly dynamic environments)

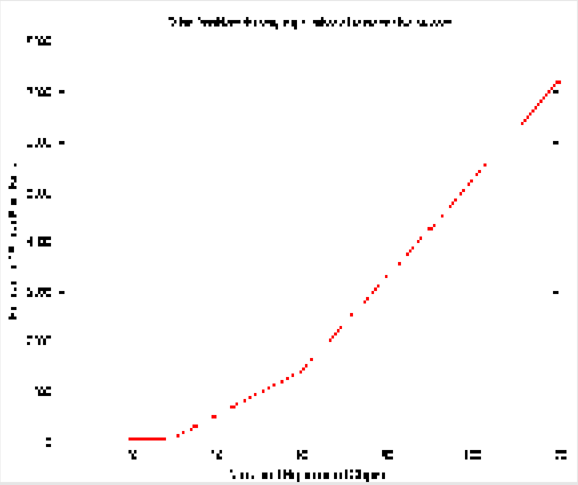
Simulation Setup

- We implemented our protocols on top of Gnutella using the Neurogrid simulator.
- 3000 nodes, 2000 object types, 90000 objects in total replicated on the nodes
- Random network topology, uniform distribution of objects over nodes
- 10bits Bloom filter size, 4 hash (SHA-1) functions, 4 bits counter for each position
- We compared IL, AL, ALR, and BFS (Breadth-First-Search)

Bloom Filter Parameters



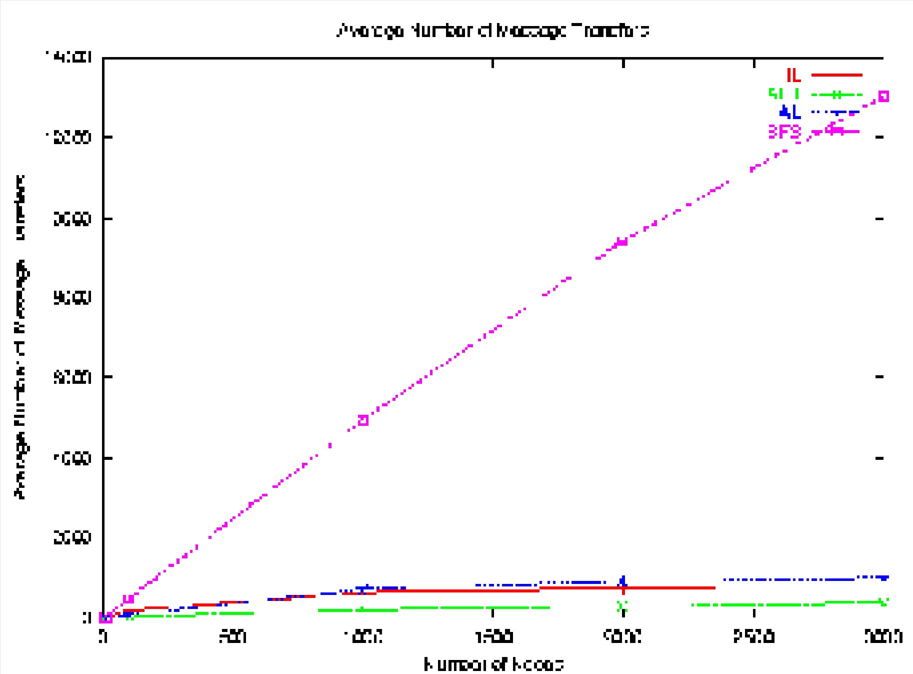




- We decided to use Bloom filters 10bits long, with 4 hash functions, and 30 objects per node for our content synopses.

Average Message Transfers during Search

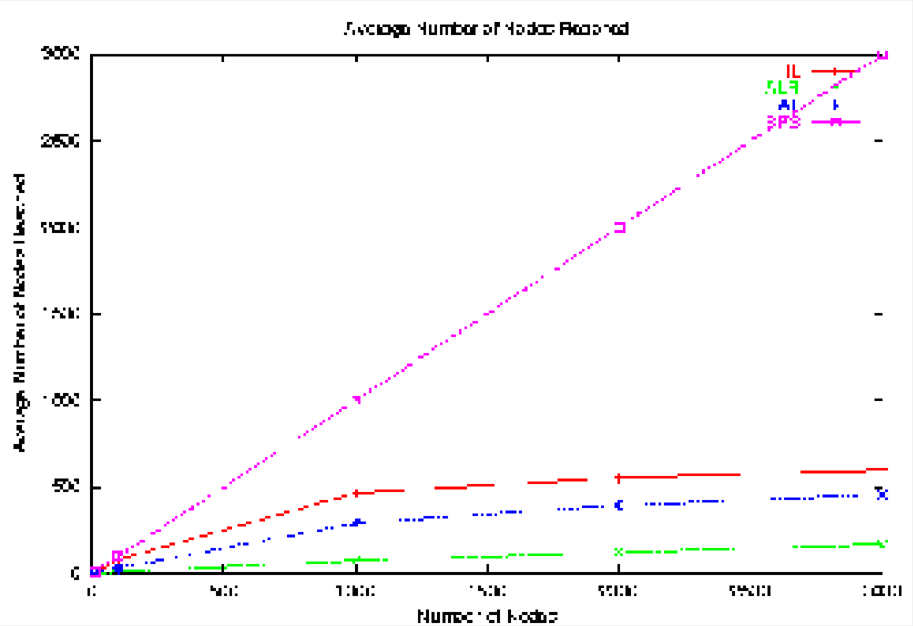
- Content-driven routing drastically decreases the number of query messages transferred during a search.
- In BFS, as the number of nodes increases, the number of message transfers grows dramatically.



- ALR achieves the minimum number of message transfers (97% less than BFS).

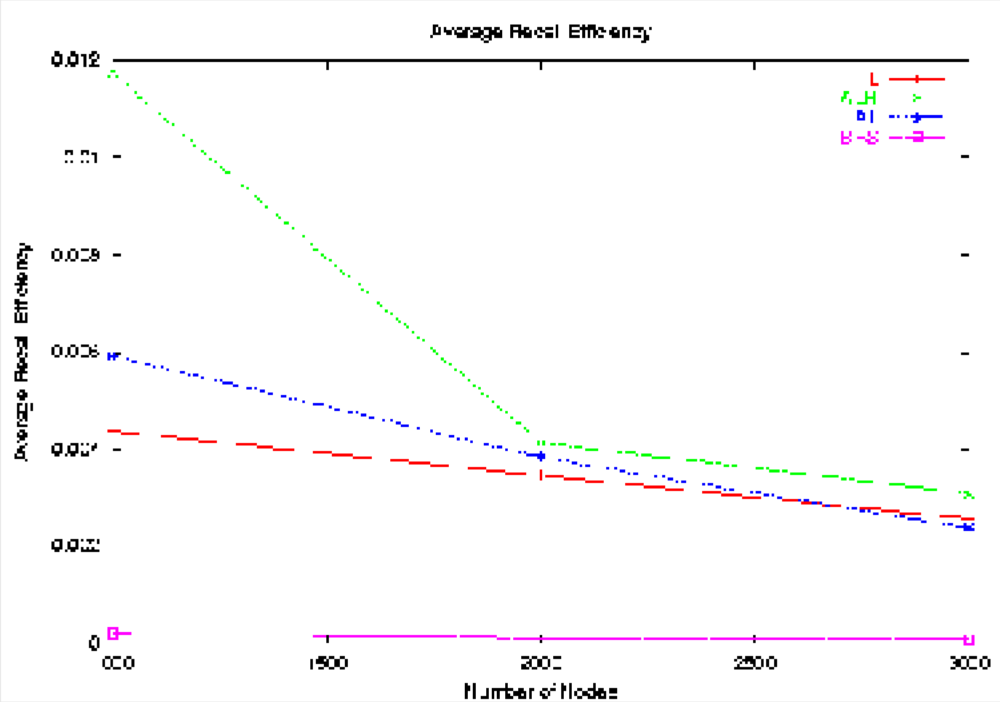
Average Number of Nodes Reached

- Content-driven routing techniques are able to provide query hits by contacting more than one order of magnitude less peers.
- The adaptive AL and ALR techniques guide queries more efficiently.



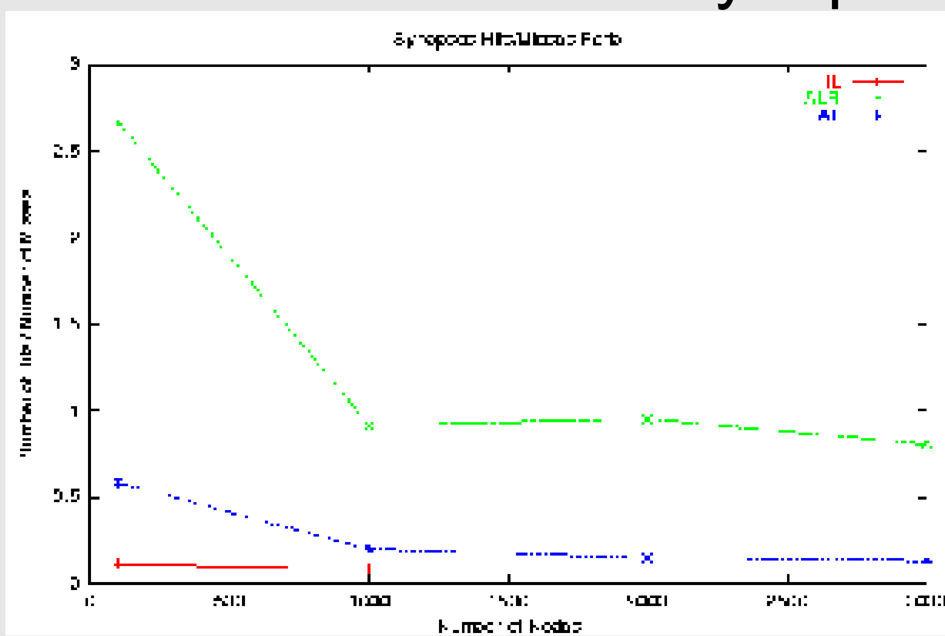
Average Recall Efficiency during Search

- Although the flooding of BFS is able to discover a lot of matches, the cost of query messages transferred results in low recall efficiency.
- ALR results in the highest recall efficiency.
- As the number of nodes grows, the proportion of the total matches discovered by the content-driven routing mechanisms decreases, since the queries are guided.



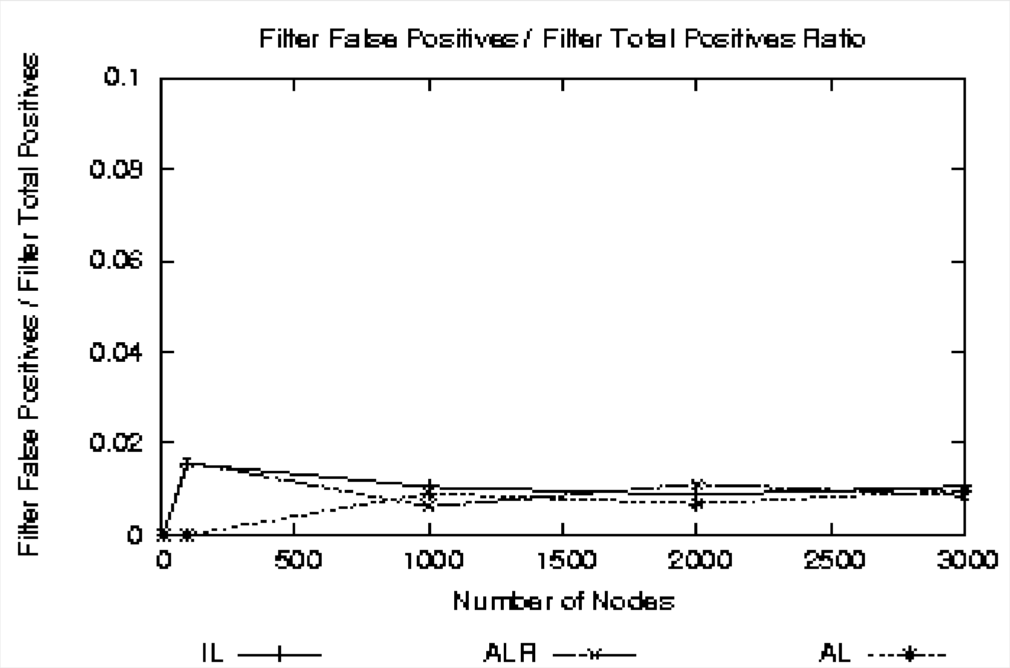
Content Synopses Hits over Misses

- Adaptively placing content synopses (AL and ALR) improves their usefulness (90% ALR-IL).
- ALR is able to route approximately half of the queries based on the content synopses.



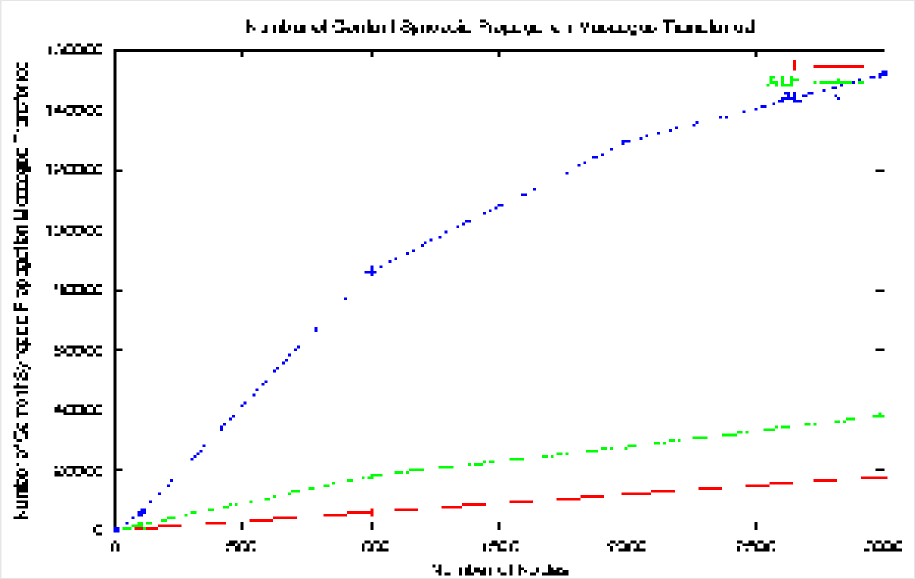
False Positives over Total Positives

- Content-driven routing is extremely accurate.
- Only a very small percentage (around 1%) of the total queries is falsely routed.



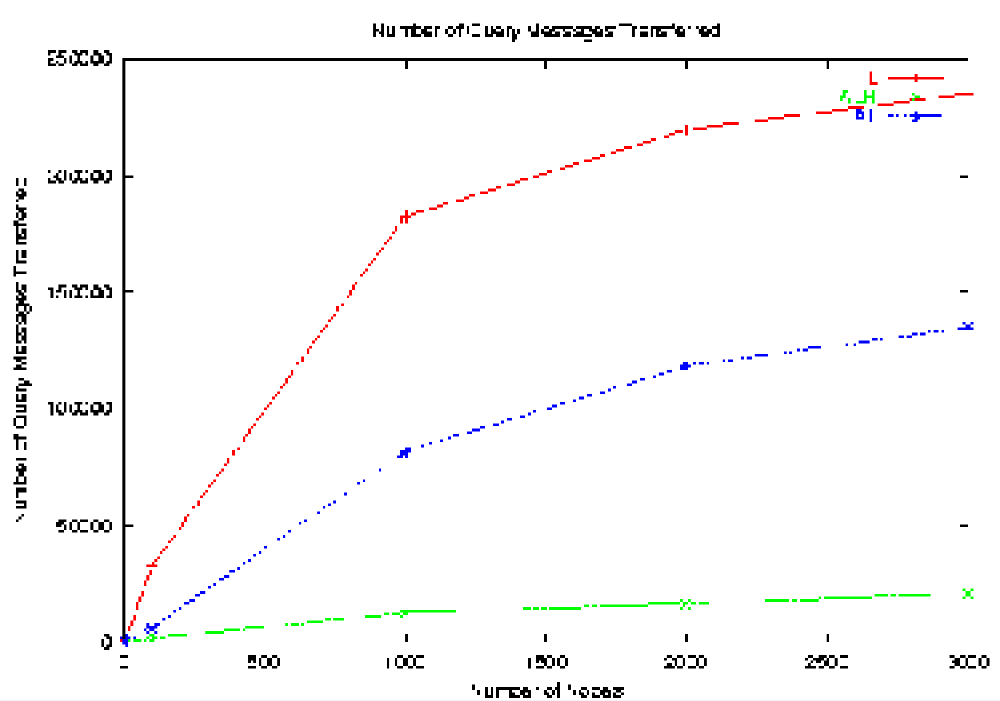
Total Content Synopses Messages

- By disseminating content synopses messages only to immediate peers IL keeps the overhead small.
- AL has to disseminate a lot of content synopses to be useful.
- ALR manages to route queries effectively and yet keep the overhead relatively small, even as the number of nodes increases.
- Combining several content synopses in one message (ALR) reduces their dissemination overhead.

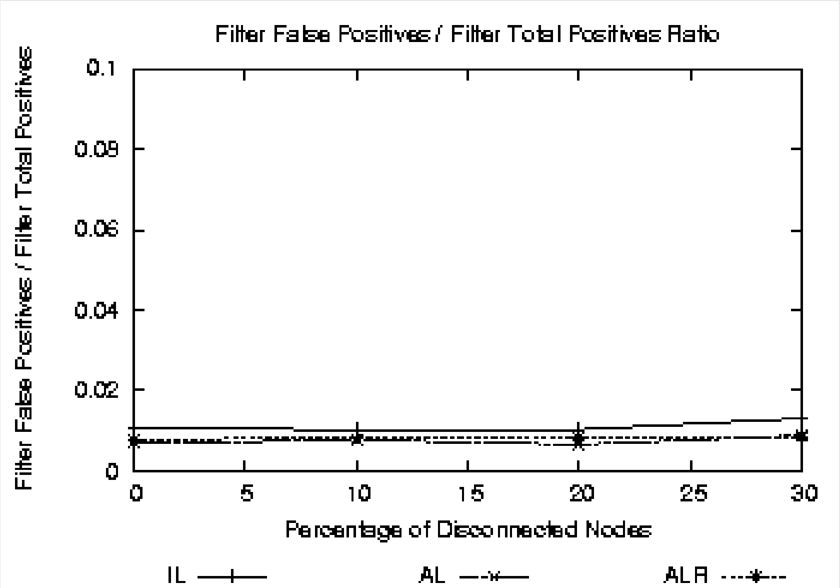


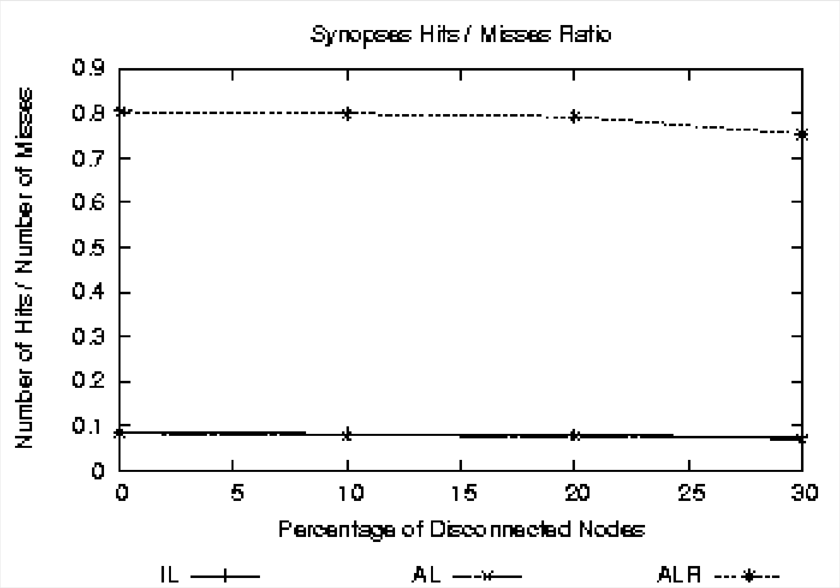
Total Query Messages

- ALR decreases the number of query messages transferred by half an order of magnitude compared to AL and by one order of magnitude compared to IL.
- ALR reduces query messages by utilizing a lot of content synopses and placing them intelligently in the network.



Performance in a Highly-Dynamic Environment



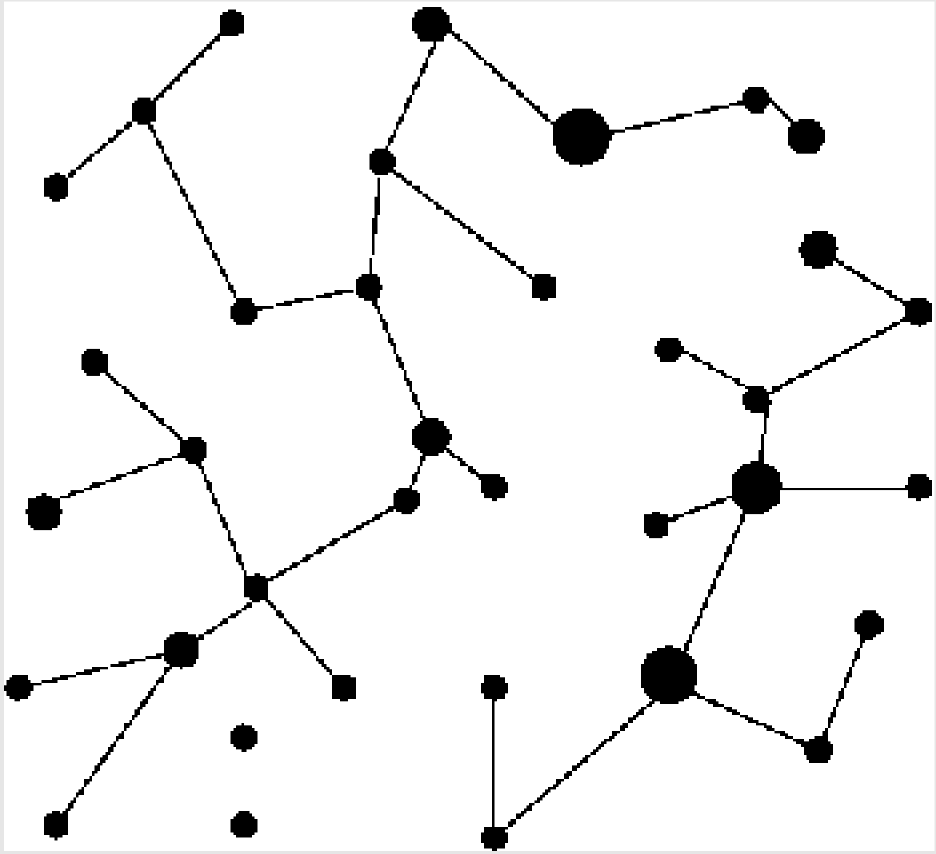


- Content-driven routing remains accurate even when peers disconnect. The immediate peers realize the disconnection and update their summaries.
- Peer disconnections do not considerably affect the success of the synopses in query routing.

Conclusions and Future Work

- We have presented:
  - Content-driven routing, so that queries are forwarded where they will most probably be answered
  - Adaptive data dissemination, so that content synopses are placed where they most probably be needed
  - Our mechanisms are scalable and make efficient usage of resources
- We plan to investigate:
  - More schemes for adaptive data dissemination
  - The effect of different network environments on our mechanisms (e.g. mobile ad hoc networks)
  - The behavior of our protocols in real WAN deployment

Thank you!



“Adaptive Data Dissemination and Content-Driven Routing in Peer-to-Peer Systems”

MSc Thesis Defense

Thomas Repantis
Department of Computer Science and Engineering
University of California, Riverside

trep@cs.ucr.edu
http://www.cs.ucr.edu/~trep