Load-Driven Neighborhood Reconfiguration of Gnutella Overlay

Enhanced Reconfigurable Gnutella Overlay (ERGO)

Evangelos Pournaras

Supervised by: Dr Nick Antonopoulos

Contributor: Georgios Exarchakos

UNIVERSITY OF SURREY

About me

➢ B.Sc. on Technology education & Digital Systems, University of Piraeus, Greece

➢ M.Sc. with Distinction in Internet Computing, University of Surrey, UK

Journal Publication

Special Issue of Elsevier Computer Communications on Self-Organization and Self-Management in Communications - Taking Vital Steps Towards Realizing Autonomic Networking

http://www.elsevier.com/locate/comcom

UNIVERSITY OF SURREY

Enhanced Reconfigurable Gnutella Overlay (ERGO)

Evangelos Pournaras, December 2008

AGENDA

- Problem Overview
- Objectives
- ERGO Concept
- Architecture
- Interaction Messages
- Model Description
- ERGO Maintenance
- Cost-effectiveness
- Simulator
- Simulation Environment
- Results
- Conclusions and Future Work

Problem Overview

➤ Why Gnutella???
  ➤ A challenging case-study environment
➤ Large-scale unstructured P2P network
➤ Does not scale well
➤ Flooding, large number of messages
➤ Many overloaded nodes
➤ Unexploited resources
➤ Tends to become star topology

Objectives

- Development of a generic load-balancing mechanism
- Alleviation of overloading areas by:
  - Exploiting underloaded ones
- Improve performance from:
  - client side (local view)
  - network side (global view)
- Succeed the above objectives with a minimum cost
- Self-optimization towards autonomic computing
  - *"Self-optimize: to tune and balance workloads to maximize the use of information technology resources"*, [IBM]
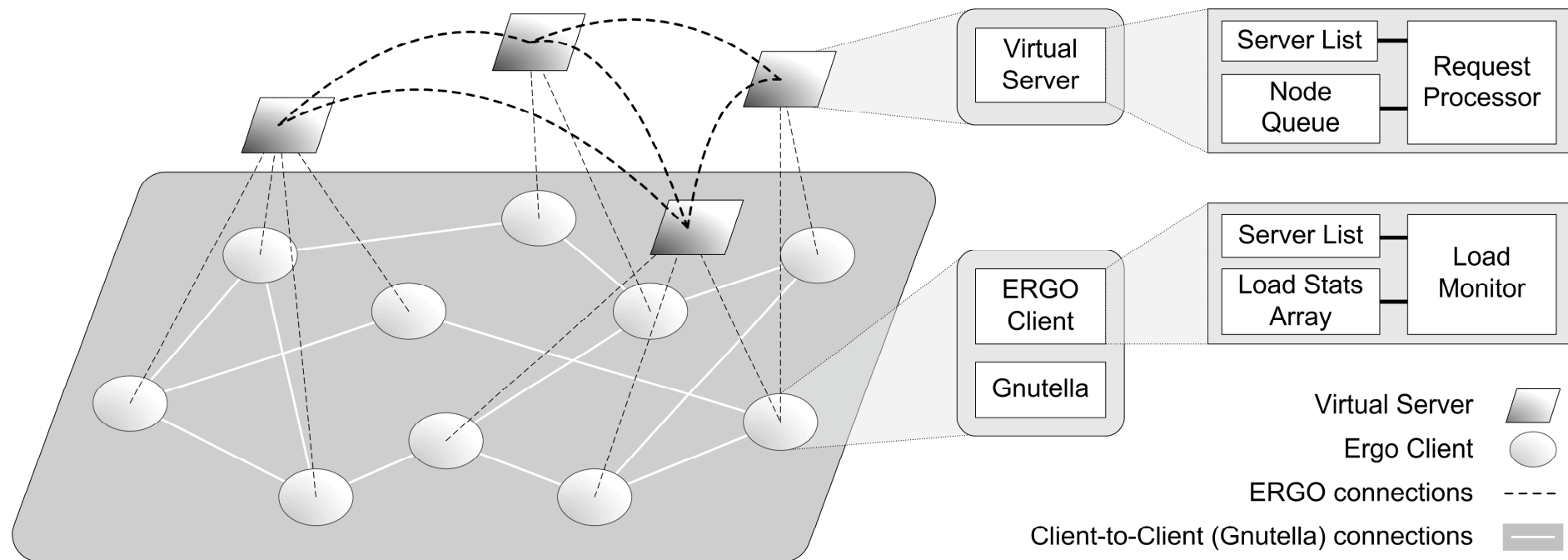
ERGO Concept

➢ Load-balancing nodes by **incoming queries**
➢ **Rewiring**: The **logical movement** of **incoming links** from overloaded nodes to underloaded
➢ Management of the rewiring by ERGO overlay
➢ **Virtual Servers**:
  ➢ Dedicated machines
  ➢ Clients
  ➢ Virtual Organizations
➢ Underloaded nodes **advertise** their selves to virtual servers and overloaded nodes **request help** from them

Architecture

Virtual Server

Server List

Node Queue

Request Processor

ERGO Client

Gnutella

Server List

Load Stats Array

Load Monitor

Virtual Server

Ergo Client

ERGO connections

Client-to-Client (Gnutella) connections

UNIVERSITY OF SURREY

Enhanced Reconfigurable Gnutella Overlay (ERGO)

Evangelos Pournaras, December 2008

Interaction Messages

Message	Description
advertisement	the advertisement of an underloaded node to a server.
request	the request generated by the overloaded node and forwarded to the overlay of servers containing the value of the excessive load to be served.
response	the response from the server to overloaded node with one underloaded node that can take over the excessive load.
rewire	command to the load originators to rewire their links from the overloaded nodes to the discovered underloaded one.
ack	used by both servers and nodes to acknowledge the other part that an action is complete or not.

Model Description

➢ Load Monitoring

load ────────────────────────────────── overloaded
     ──────────────── normally loaded
     ──── underloaded

capacity [_____|_____|_____]
              lower threshold   upper threshold
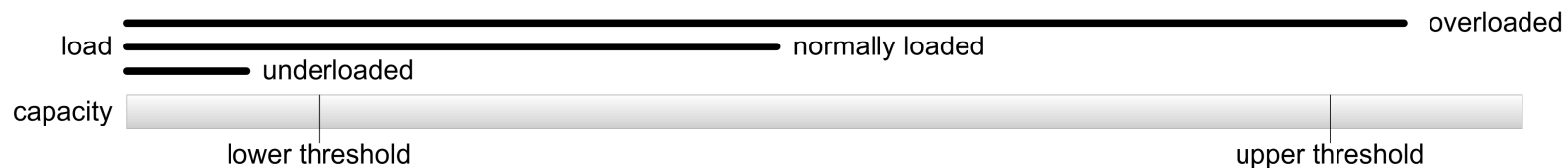
➢ Target Node Discovery
  1. Overloaded node must become balanced **AND**
  2. Underloaded node must remain balanced or underloaded
➢ Neighborhood Rewiring
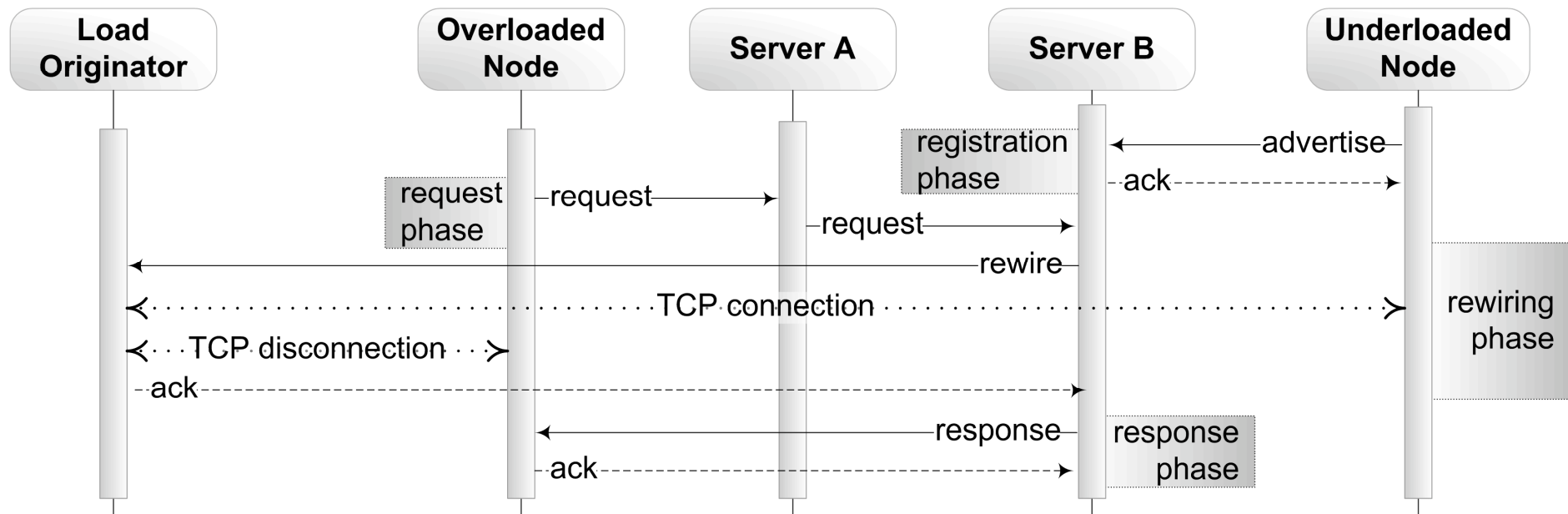 ➢ Virtual Server request rewiring to load originators and acknowledge to the overloaded node
➢ Failures and Extreme Conditions
  1. Exhaustive Failure
  2. No virtual servers
  3. Intermediate Failure

Model Description (cont.)

Enhanced Reconfigurable Gnutella Overlay (ERGO)

Evangelos Pournaras, December 2008

ERGO Maintenance

1. Through underlying searching mechanism
   - A new field in queries & query hits
   - In each visit the field is updated
     - Nodes read new available virtual servers
     - They fill the field if they are virtual servers
2. Through ERGO
   - Initialization: All nodes are virtual servers
   - Virtual Server deactivation - Request:
     - Refuse to help and keep the address of requestor
     - Requestor deletes node from Server List and tries another one
   - Virtual Server Activation:
     - Notification to all nodes refused to help

Cost Effectiveness

- Client
  - Advertisements
  - Requests
  - Load Statistics Array updates
- Server
  - In-degree
  - In-degree of overloaded nodes which belong to its incoming links
  - Convergence balance time

Simulator

- Built in Java from scratch
- Java Data Structure Language (JDSL)
- Gnutella simulation
- Keyword-based file distribution
- Downloading option
- GUI for user interaction
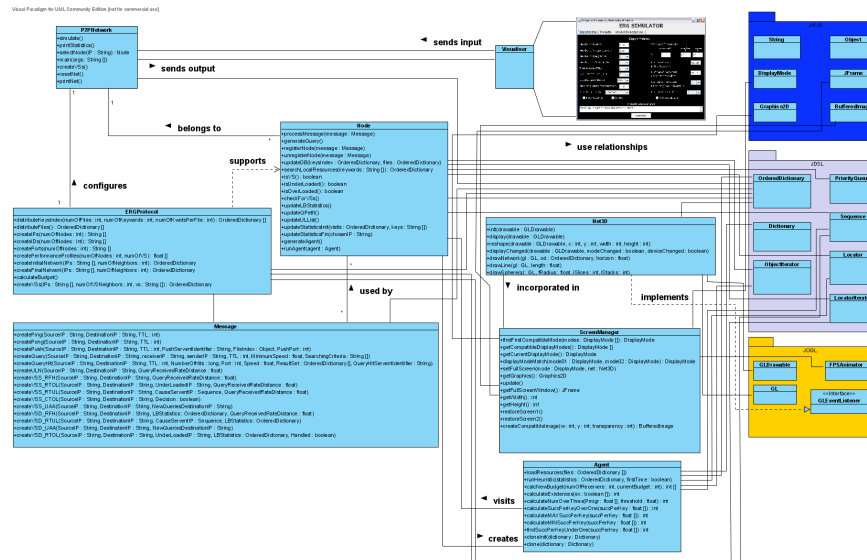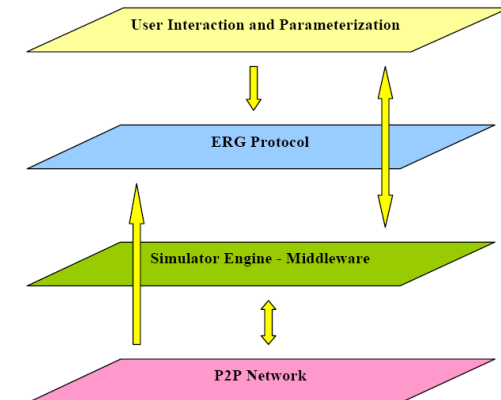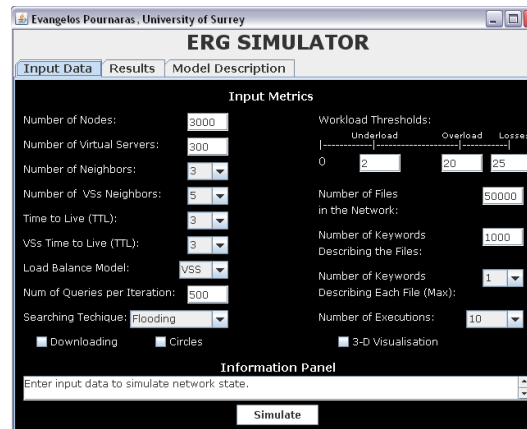
Simulation Environment

- 3000 nodes
- Varied servers ratio (0.1%, 1%, 5%, 10%, 50%, 100%)
- Gnutella out-degree=3, ERGO out-degree=3
- Gnutella TTL=4, ERGO TTL=4
- 50000 files, 1000 keywords, 1 keyword/file uniformly distributed
- 100 iterations
- Varried generated load per iteration (queries)
- Load thresholds: [10, 50, 60] and [20, 70, 90]

Results – Load Profiles of Nodes
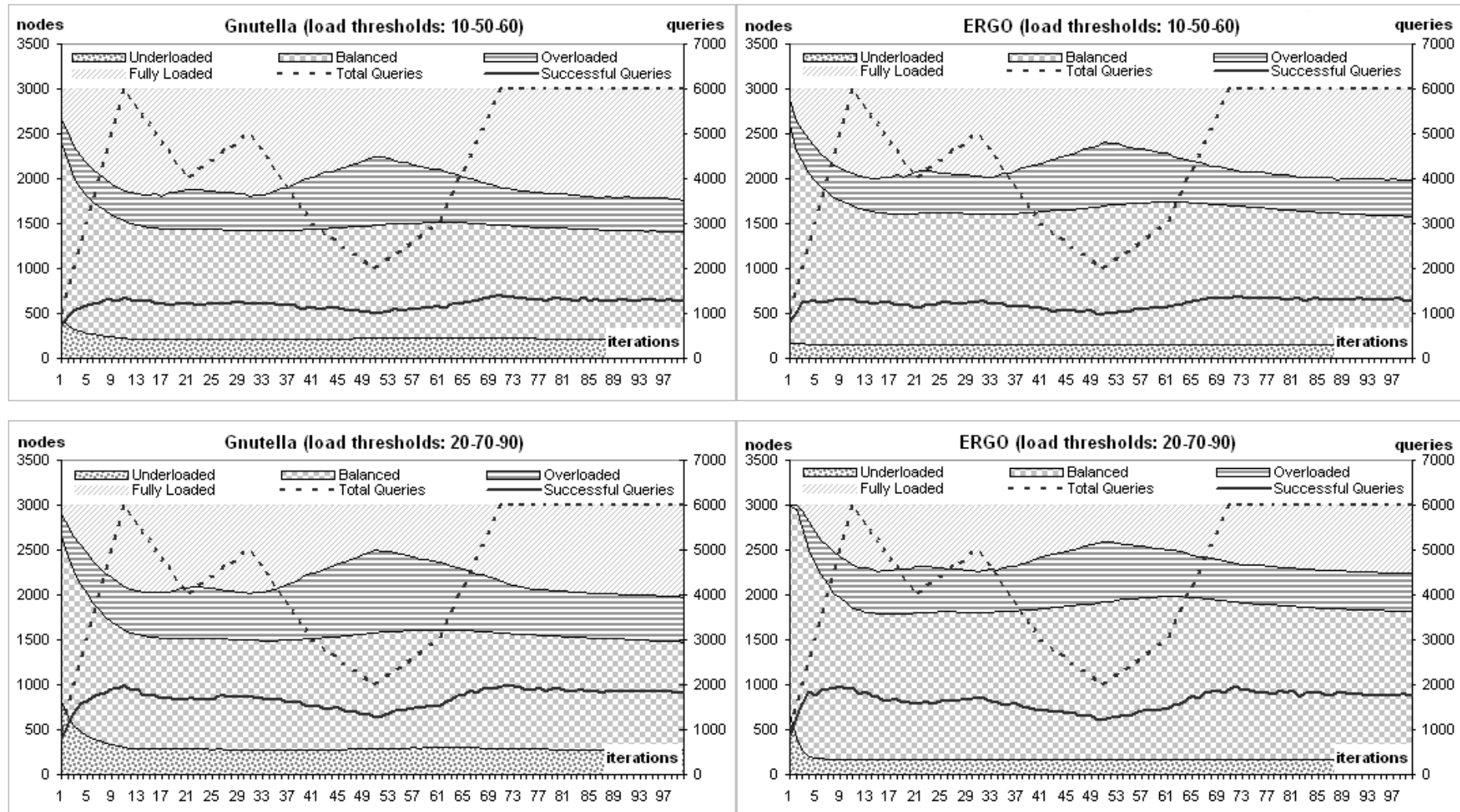
Results - System Availability

$$Avail_{Syst} = 1 - \frac{Overload_{Total}}{Load_{Total}}$$
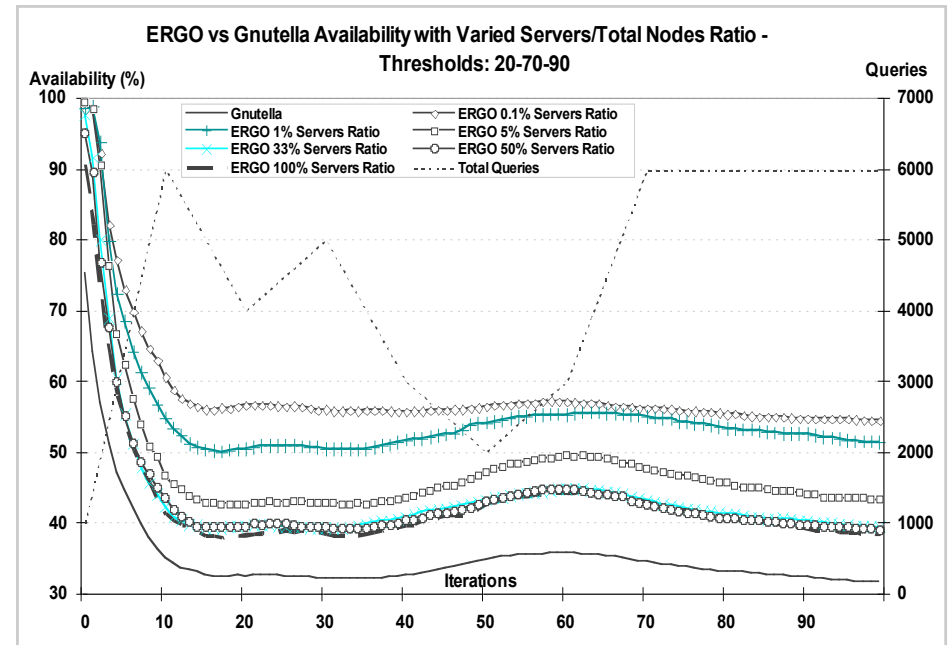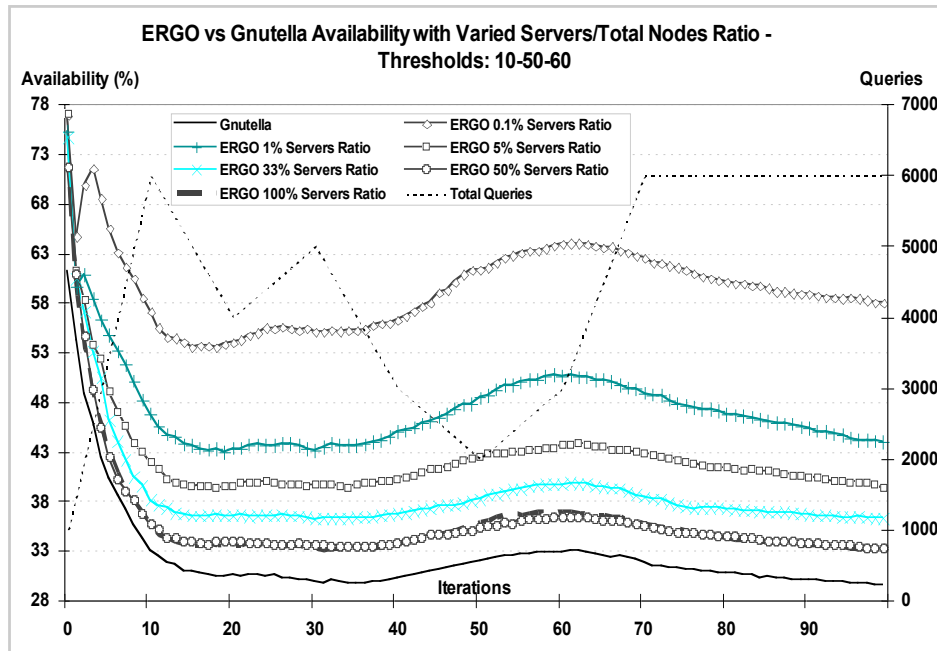
Results - Query Success
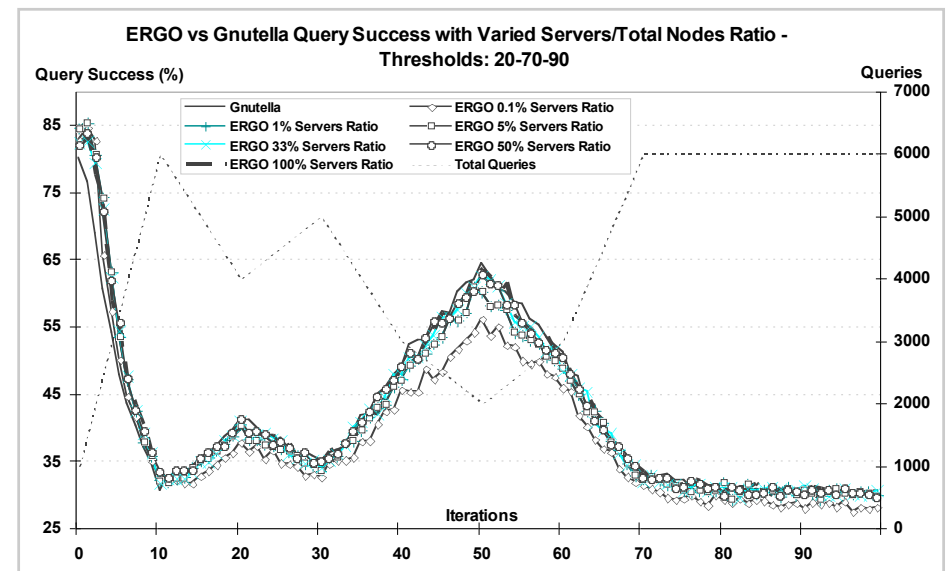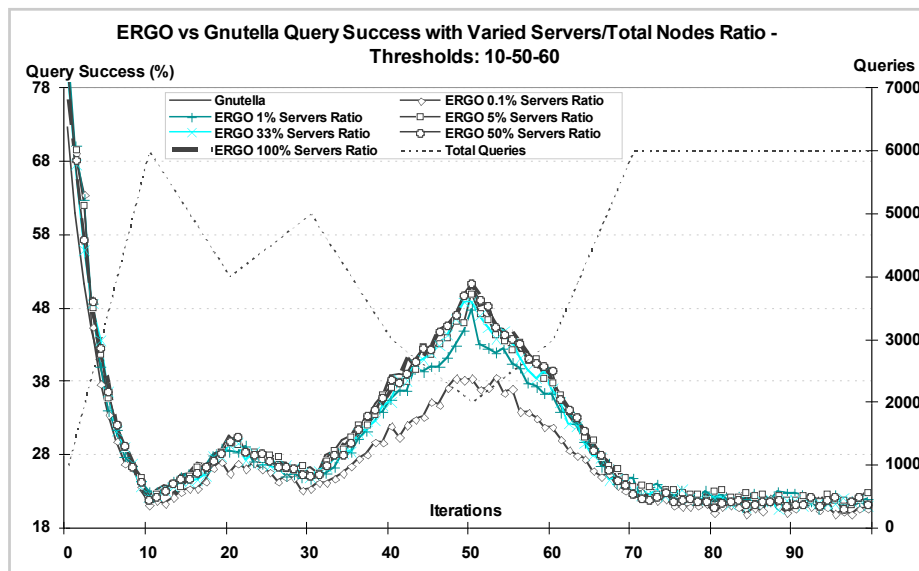
Results - Communication Cost
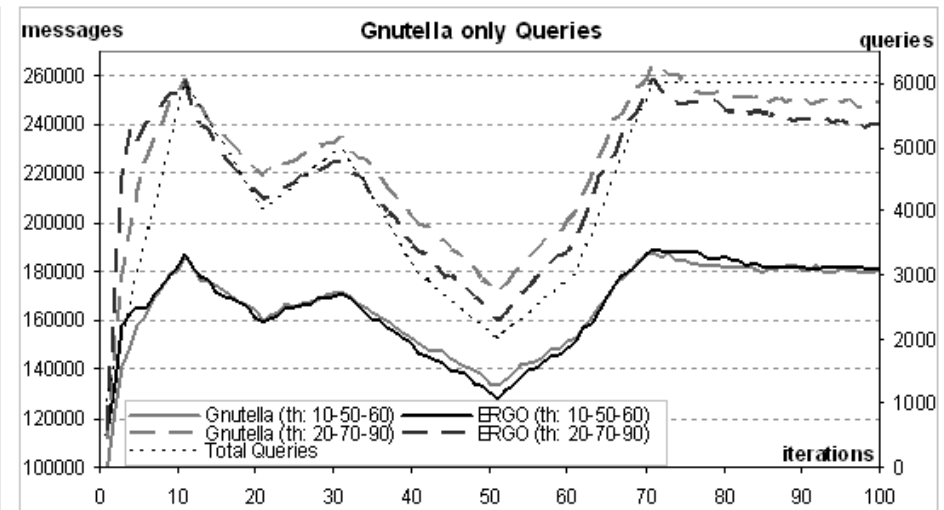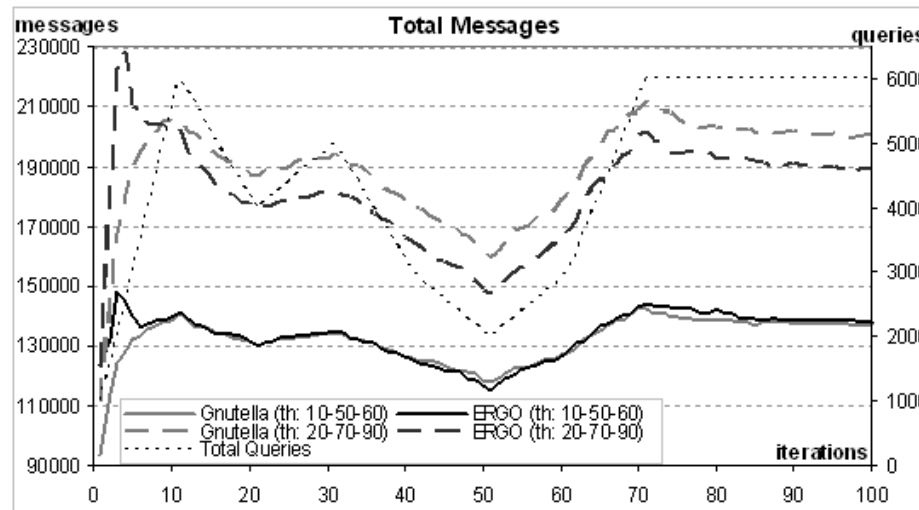
Conclusions and Future Work

- ➢ Increase of balanced nodes in the network
- ➢ Reduction of overloaded and underloaded nodes
- ➢ Increased system availability
- ➢ Retainability of query success efficiency of Gnutella
- ➢ Introduction a small communication cost
  - ➢ Even lower compared to Gnutella as thresholds increase

- ➢ More dynamic topologies simulations
- ➢ Layer maintenance implementation
- ➢ Variance of load thresholds
- ➢ Content and trust-based rewiring

Questions ?