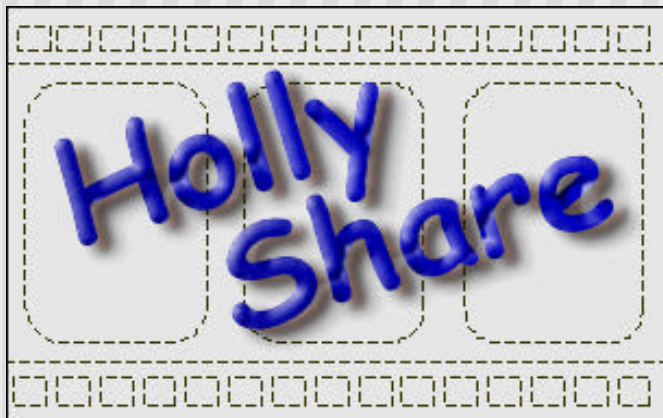


# Peer-to-Peer Application: HollyShare

---



Project team:

Bijit Hore

Ilya Issenin

Shannon Tauro

Sean McCarthy

Songmei Han

Presentation prepared by:

Shannon Tauro

# Project Objectives

---

- Limitations of existing free file-sharing tools:
  - Do not allow to share private files among a group of people
  - Only 'search' option, no way to show all the files from all the people together
  - No notification about new files
- Plan to use it personally in the future

# HollyShare: Unique Properties

---

- Users are known to each other
- Users join by invitation only
  - Secure communication between users (future work)
- Catalogue of available files

# Project Development

---

- Requirements Phase
- Architecture Development
- Distribution of Modules
- Development & Integration of Modules

# Project Development

---

- Requirements Phase
  - Approx. 5 initial meetings to establish
    - Project Idea
      - “Let’s Share Movies!!!”
    - Determined Requirements
      - develop a peer-to-peer application
      - share large multimedia files
      - small group of users (users are known to each other)
      - list all shared files
    - Specified Initial Systems to Survey

# Project Development

---

## ■ Architecture Development

### ■ Approx. 5 meetings to establish:

- From Requirements  $\implies$  what Modules are needed
  - Transfer Files: File Transfer Module
  - Connect/Disconnect/Remain: Network Connectivity Module
  - Support/Databases: Node Database(who is connected), Configuration Manager(user/system settings), File Database(manages list of users files)
  - User/System Interaction: GUI

# Project Development

---

- Distribution of Modules
  - Approx. 2 meetings to establish:
    - User/System Interaction: Shannon Tauro
    - Support/Databases: Sean McCarthy
    - Transfer Files: Bijit Hore
    - Connect/Disconnect/Remain: Ilya Issenin

# Project Development

---

- Development & Integration of Modules
  - Development took place on Windows NT workstations using C++ & Qt Library
  - Module Development: independently
  - Countless E-mails to:
    - notify of updated project files
    - request for services to be added in a module
  - Integration & debugging: managed by one member
    - Ilya Issenin

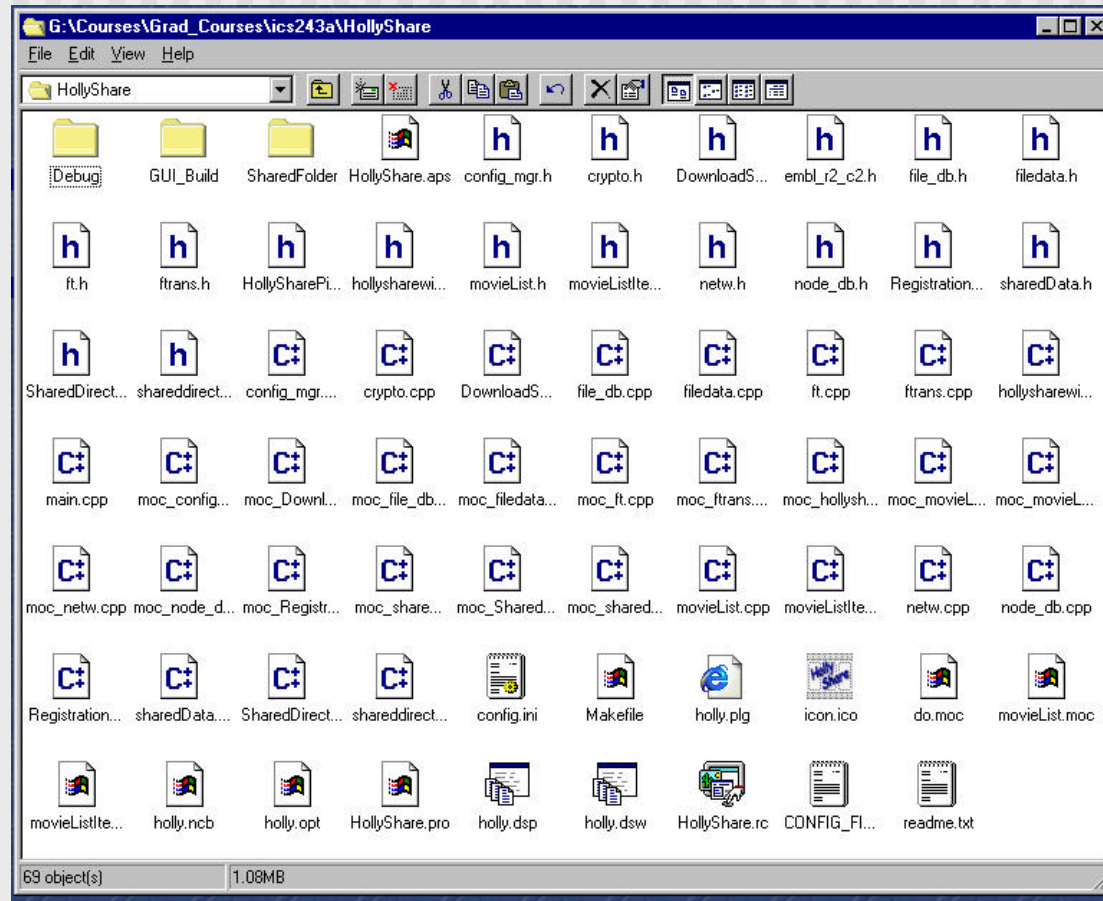
# Implementation

---

- The application is being designed in C++, using a development library called QT
  - Provides GUI functionality and basic network functions
  - Library is platform-independent
  - Test platform: Windows machines, connected to the campus internet

# File Listings (Implementation)

- 16 .h files
- 17 .cpp files
- >8000 lines of code



# Design Assumptions

---

- Providing a Catalog of all files in the system (each entry knows the owner) to all Nodes
- To keep the “members circle” somewhat exclusive, to enter a network of users, a first-time user must be given an IP address of another node on the system and the key for encryption (not implemented)

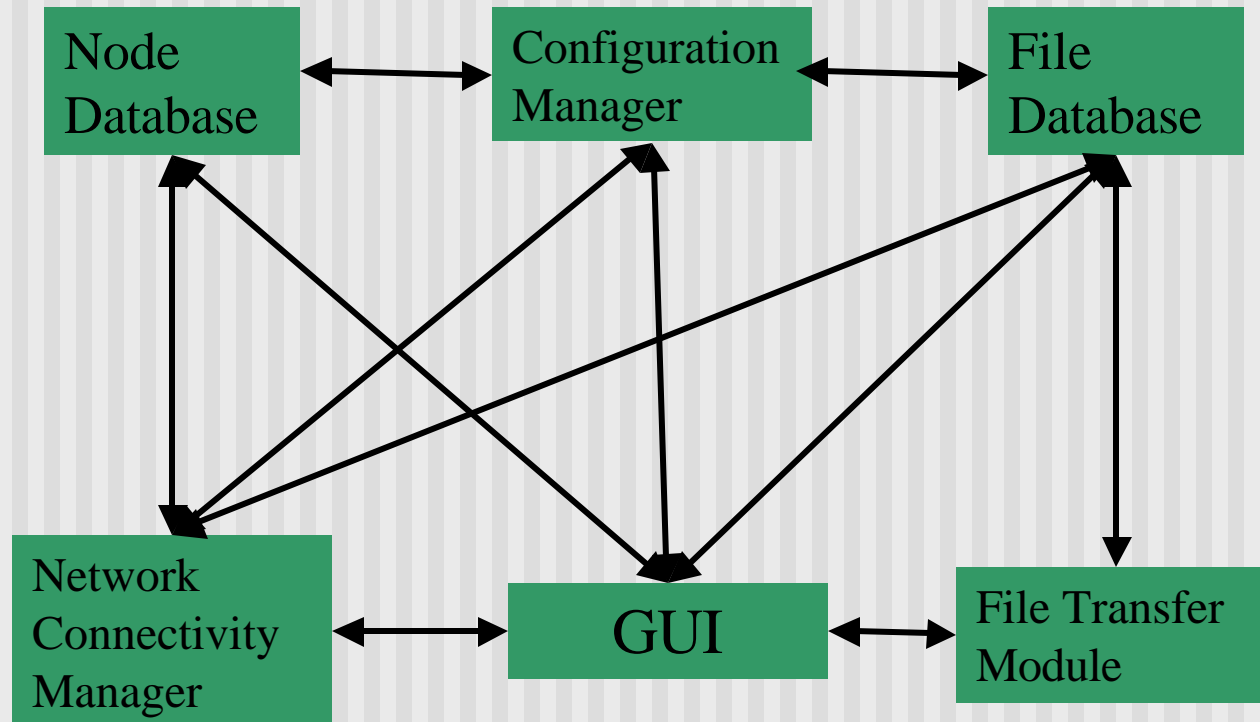
# Design Assumptions(cont.)

---

- The number of users and files are limited
- The files being shared are very large
- One or more users will have a static IP address

# Node Architecture

---



# Node architecture

---

- Node database contains information about all nodes in the system (status, address, etc.)
- File database contains information about all files available in the system (name, location, etc.)
- File transfer module controls connections for file transfers (transfers only)
- Network Connectivity Manager handles all node communication except file transfers (new connections, information propagation)
- GUI is the user interface for the program

# Division of Labor

---

- Shannon Tauro:
  - Graphical User Interface(GUI)
  - final presentation, chapter in final report
- Sean Mc Carthy:
  - Node Database, File Database, Configuration Manager
  - third presentation, 2 chapters in final report
- Ilya Issenin:
  - HollyShare Network protocol and reconnection algorithms
  - Network Connectivity Manager module, integration of all the modules, debugging
  - initial presentation, chapter in final report
- Bijit Hore:
  - File Transfer Module
  - second presentation, chapter in final report
- Songmei Han:
  - Survey presentation, survey part in final report

# HollyShare: Network topology

---

- All nodes are assumed to be identical (running identical programs)
- All nodes maintain the list of all IP addresses of the other authorized peer nodes (required for reconnecting – see later)
- Node connections form a tree (no loops)
- All nodes share information about all files in the system

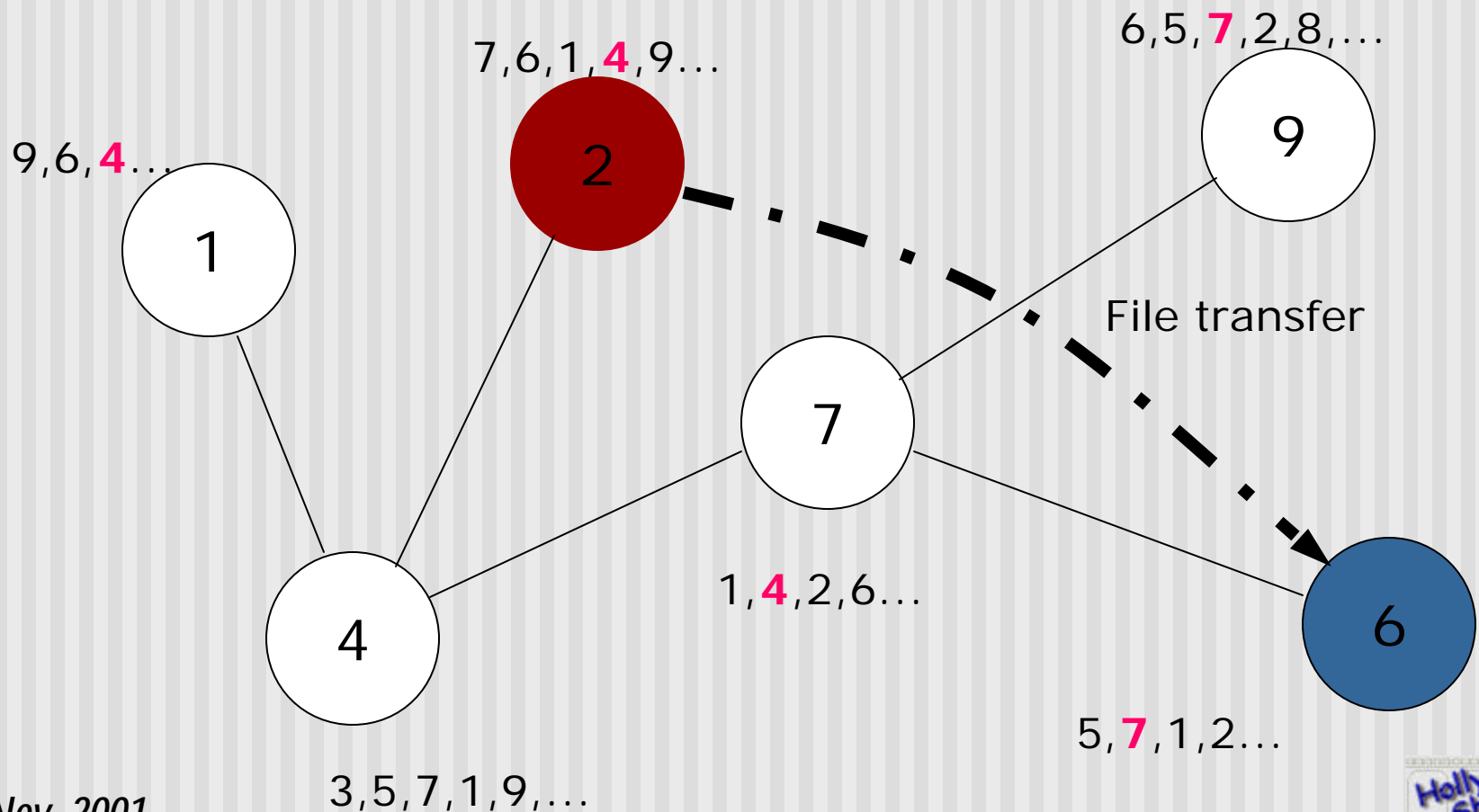
# HollyShare:

## Connecting to the network

---

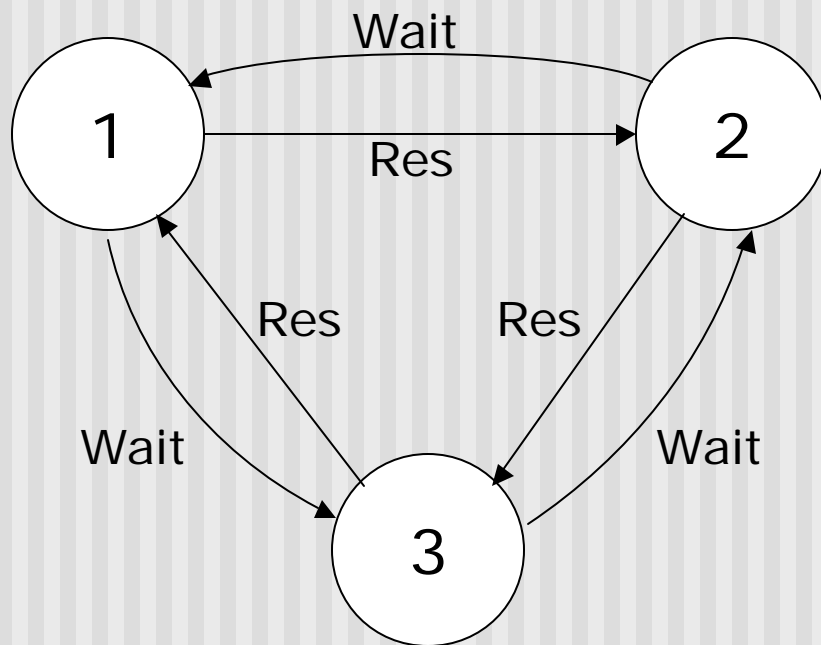
- First time after installing the program:
  - The program is provided (along with the key) with list of the IP addresses of nodes that may be connected to the network.
- Subsequent connections:
  - During each run, a **random** sequence of IP addresses is selected and the node shall attempt to make a connection to one of the nodes in that order.
  - This will ensure a **tree** of active nodes and a **small degree** for each node (load balancing).
- During reconnection process, all attempts to connect to this node will result in 'Wait' message being sent out

# Connecting to the network



# Connecting to the network

- Problem: what if three nodes try to connect to each other in a circle manner?



- Solution:
  - Timeout
  - If you have the highest (IP || port) number, do not try to reconnect the second time and go to 'on' mode
  - If not, try to do one more circle of reconnections

*For more details see Project Report*

# HollyShare: Reconnection Scheme

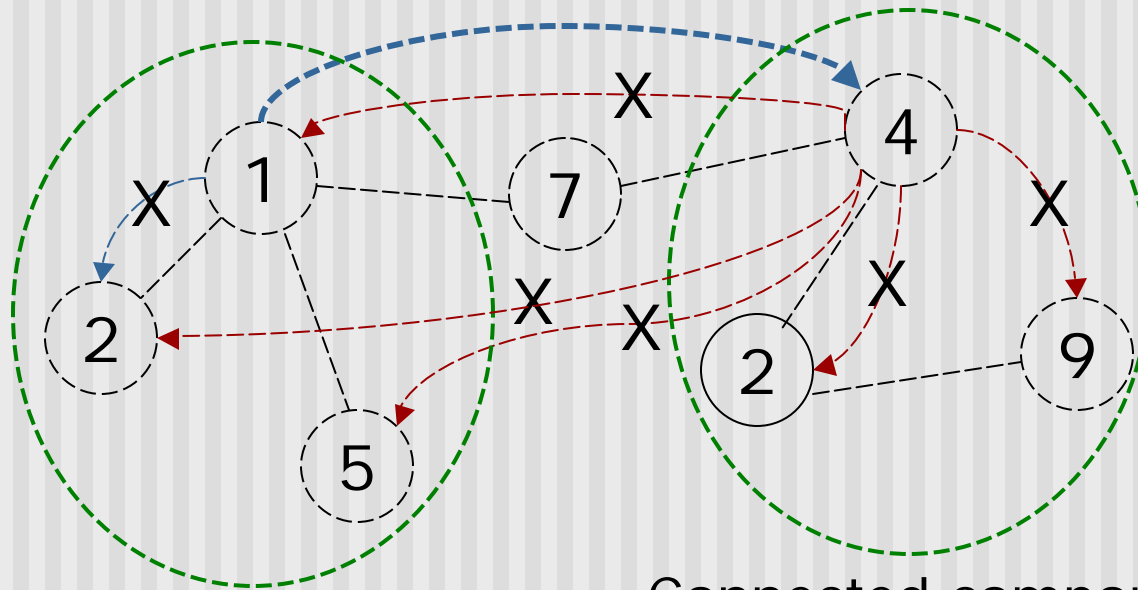
---

## ■ *Algorithm 1:*

- If any node disconnects, the network gets disconnected into several **connected components** (property of a tree)
- The neighbors of the disconnected node initiate the process of reconnection:
  - Each connected component determines the highest ID (IP || port) present in this group by going to 'selection' state and prohibiting connecting to this component until the highest ID is selected
  - Each neighbor tries to connect to one of the nodes with the ID, greater than one of its group
- Problem: what if several nodes go off during the 'selection' state?

# HollyShare: Reconnection Scheme

- **Example:**



Connected component 1

Group ID: 5

Connected component 2

Group ID: 9

# HollyShare: Reconnection Scheme

---

- **Algorithm 2 (implemented):**
  - If any node detects disconnection, it breaks all connections with the neighbors
  - After small random delay tries to connect to the network again as if the program was just started
  - All file transfer operations are not affected by the reconnection process

# Information discovery mechanism

---

- Whenever new node connects to the network, it sends a list of its shared files to the neighbors, they retransmit it to their neighbors, till each node gets the information.
- All nodes keep the list of all shared files while they are active.
- Notifications of additions or deletions of shared files are propagated from the nodes in a similar manner.
- When one of the nodes disconnects, the list of the files is automatically updated in each node.

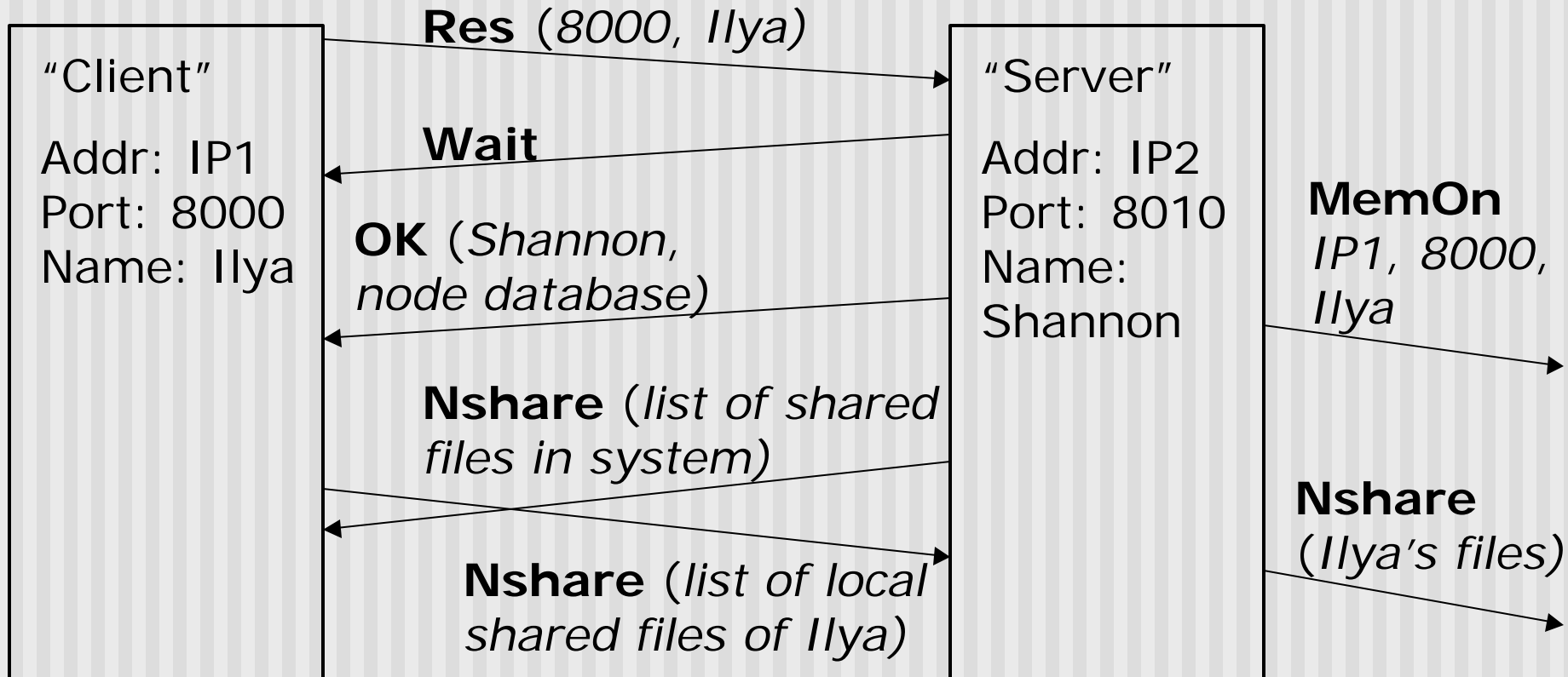
# HollyShare network protocol

---

- Network connectivity manager:
  - Res                      Reset – first message after establishing connection  
*my port, my name*
  - Wait                      Reply for Res message – wait for OK message
  - OK                        Reply for Res message – connection accepted  
*my name, node database*
  - Nshare                    New share – have some new files to share  
*file list*
  - Dshare                    Delete files from the shared list  
*file list*
  - MemOn                    Member connected  
*IP, port number, name*
  - Ping                      Request to send a Pong message
  - Pong                      “I’m alive” message

# HollyShare network protocol

- Typical scenario after establishing TCP connection:



# File Transfer Module

## (Algorithm Highlights)

---

- Download a file in parallel by opening connections to multiple hosts of the file (which are currently active)
- Tasks are assigned dynamically in chunks so that the fastest connections are used optimally.
- Client monitors the state of download to maximize speed and recovery in case of failure.

# File Transfer Module

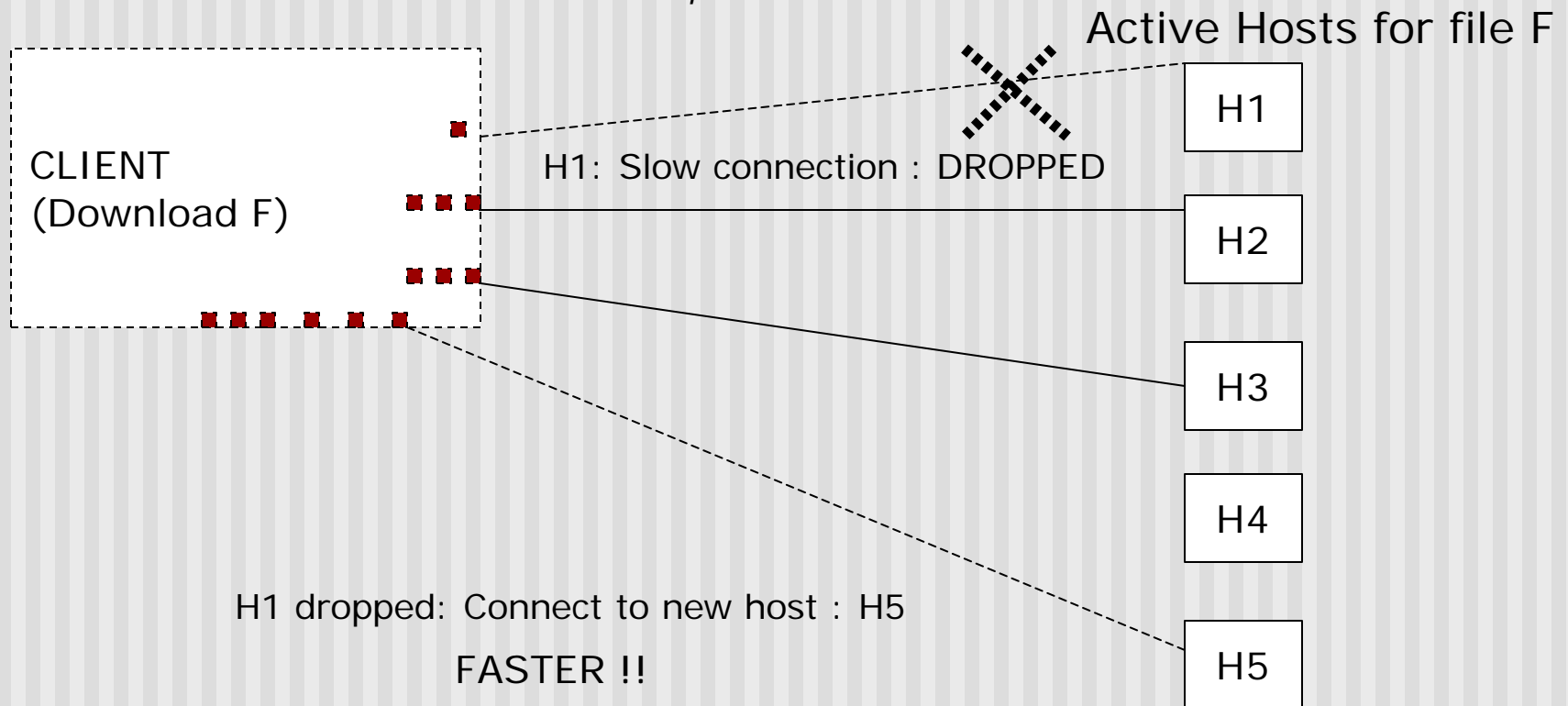
## (Algorithm Highlights)

---

- Recovery: client side is robust enough to tackle situations where connections might be dropped in the middle of a download, by:
  - Re-allocating failed or timed-out tasks to other stable connections.
  - Replacing slower connections by newer ones (new hosts of file) to maximize speed of parallel download.

# File Transfer Module

- Initial connection to H1, H2 & H3

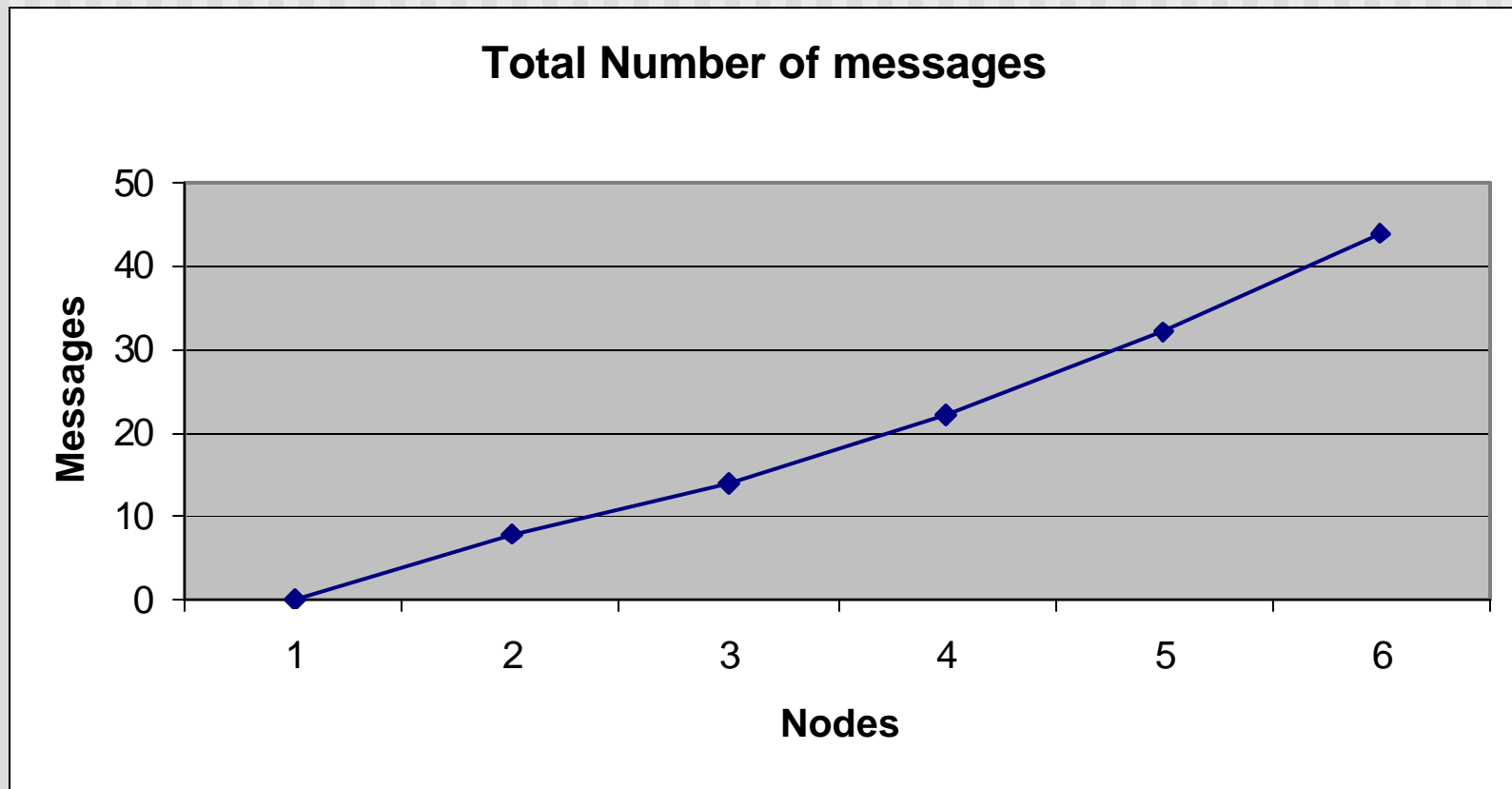


# Results & Analysis

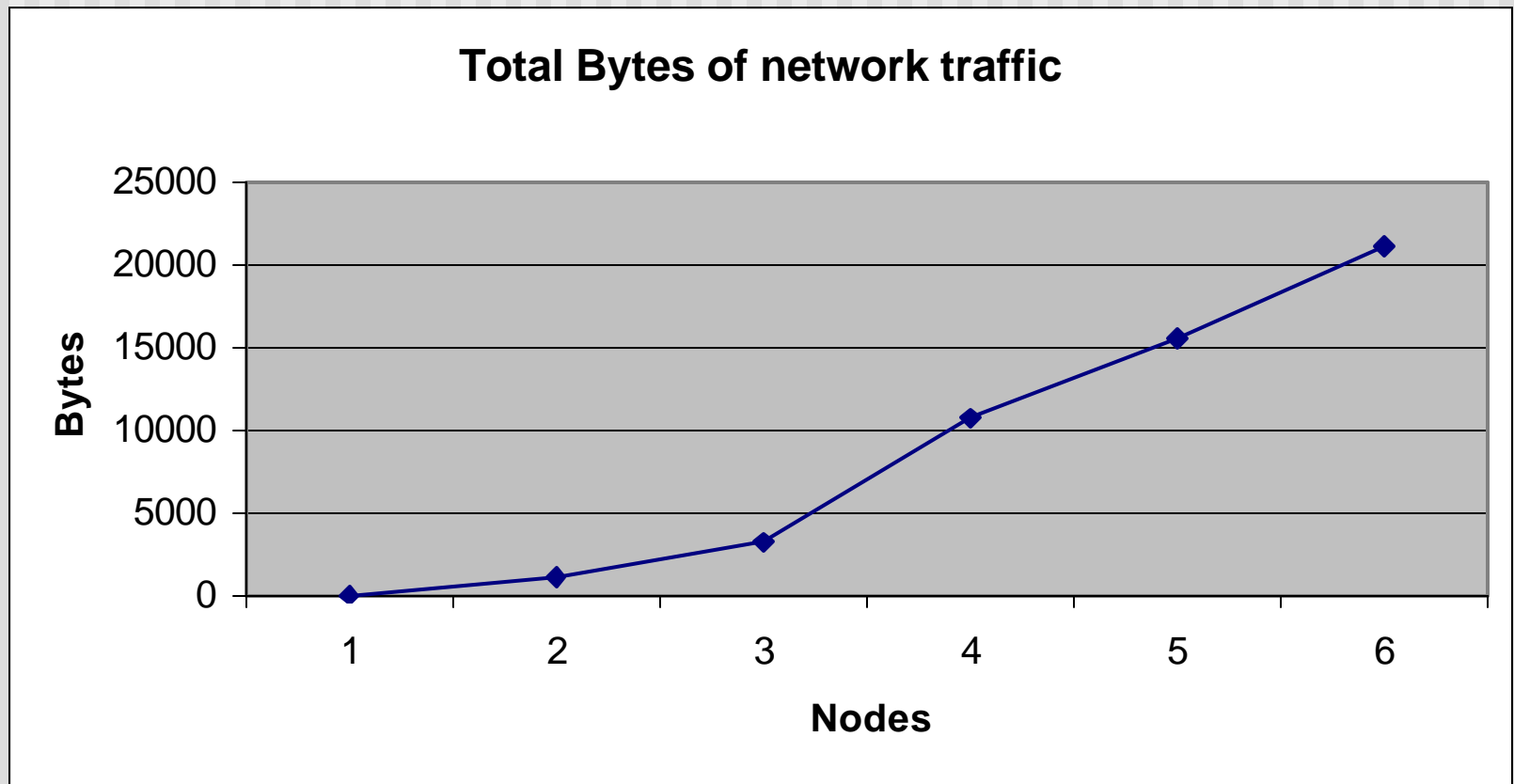
---

- Measured number of messages sent between nodes for
  - Connection / node database change propagation
    - Added nodes one at a time and waited until system stabilized (i.e. non-ping traffic ceased)
  - File database change propagation
    - Added 427 files at one node and measured propagation to all nodes
- Also measured bytes of traffic generated in these situations

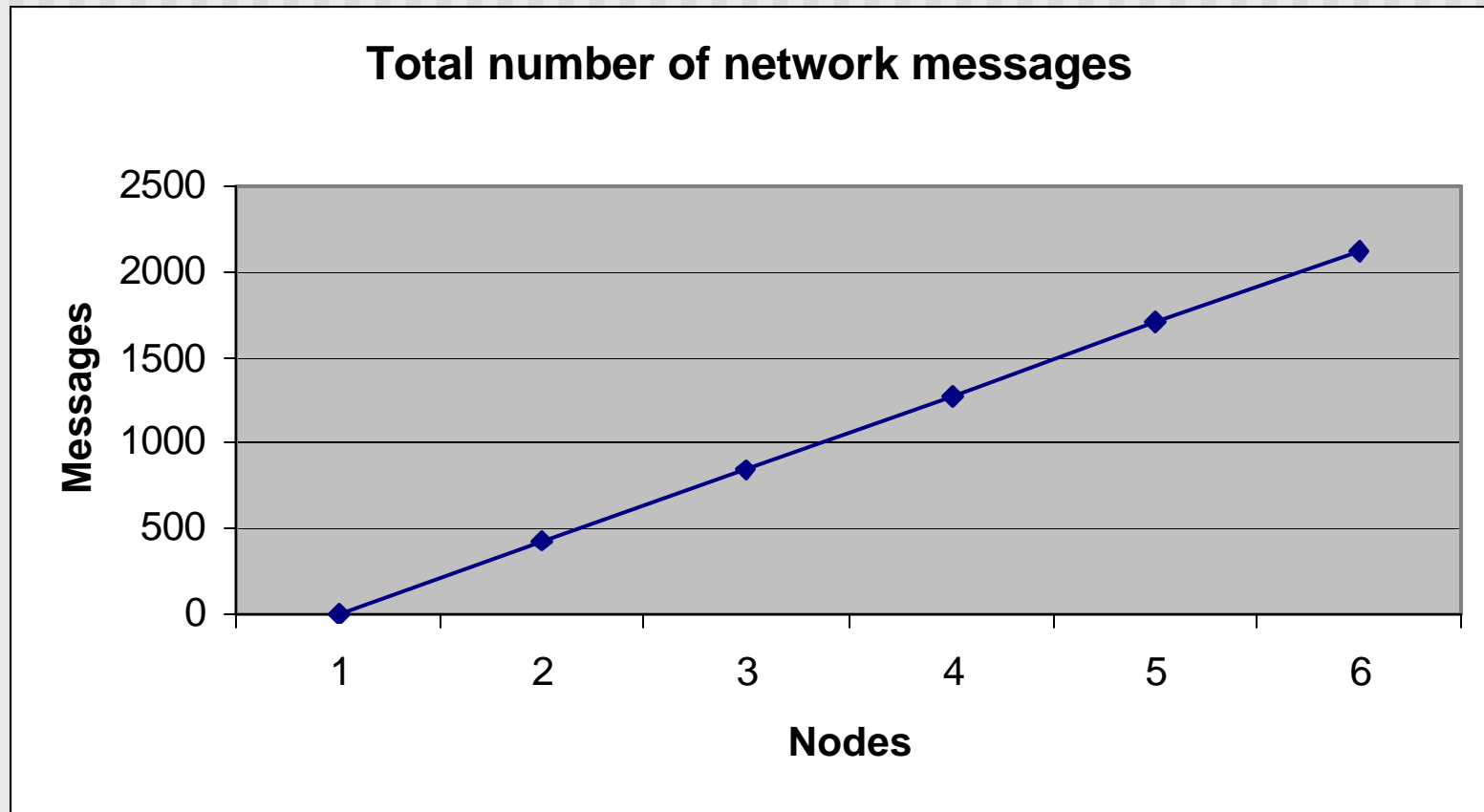
# Connection traffic (I)



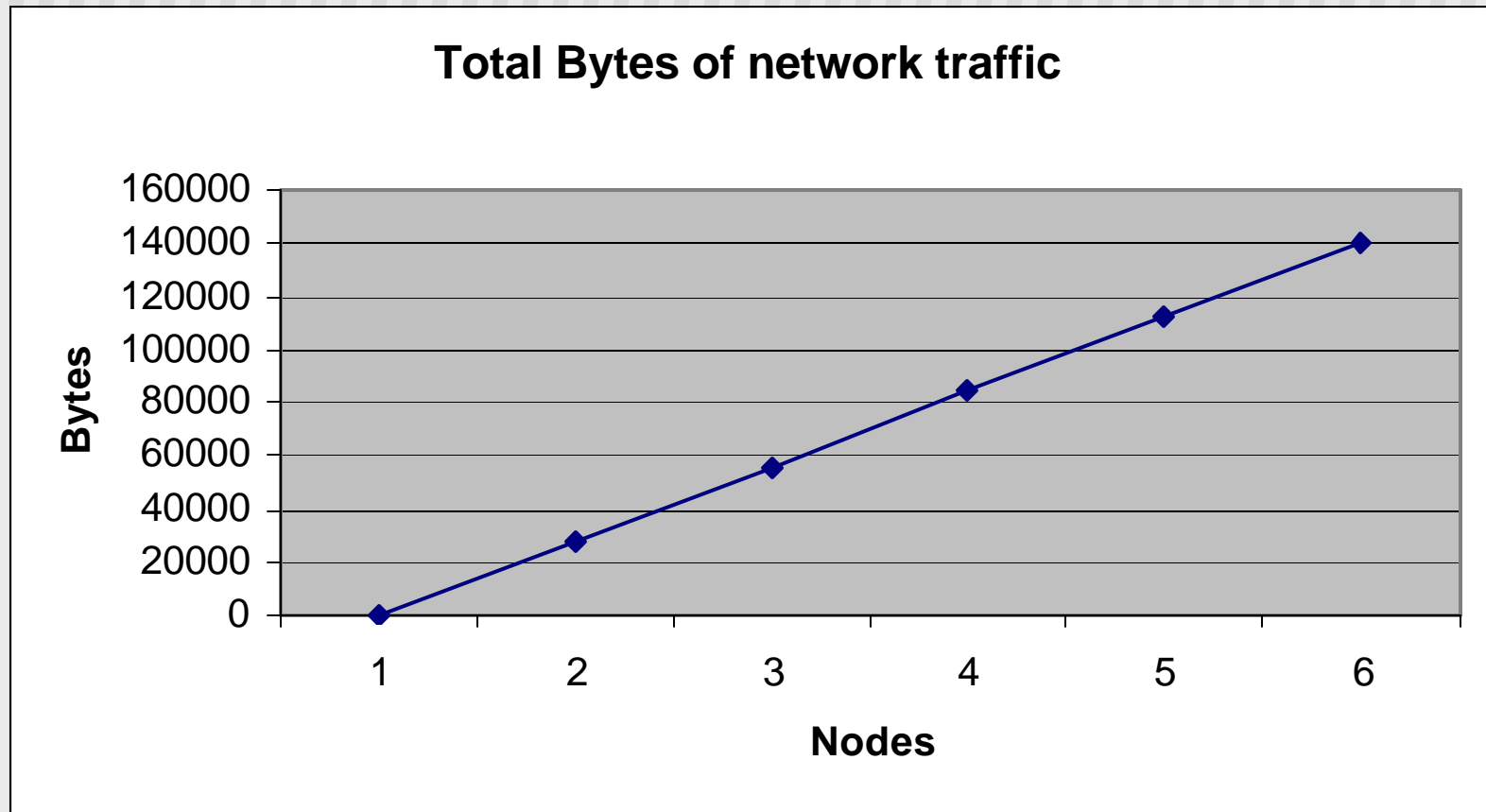
# Connection traffic (II)



# Propagation traffic (I)



# Propagation traffic (II)



# Analysis

---

- Message traffic for connection and database propagation is relatively small, assuming that the number of files shared by users is small
  - Based linearly on the number of nodes in the system and the number of files that are added/deleted

# Future Work

---

- Security: encryption, key management
- Review system (User Review & Movie Rating)
- "Send message" system
- Users may join more than one circle of users

# HollyShare Demonstration

