

# Broadcast Updates with Local Look-up Search (BULLS): A New Peer-to-Peer Protocol

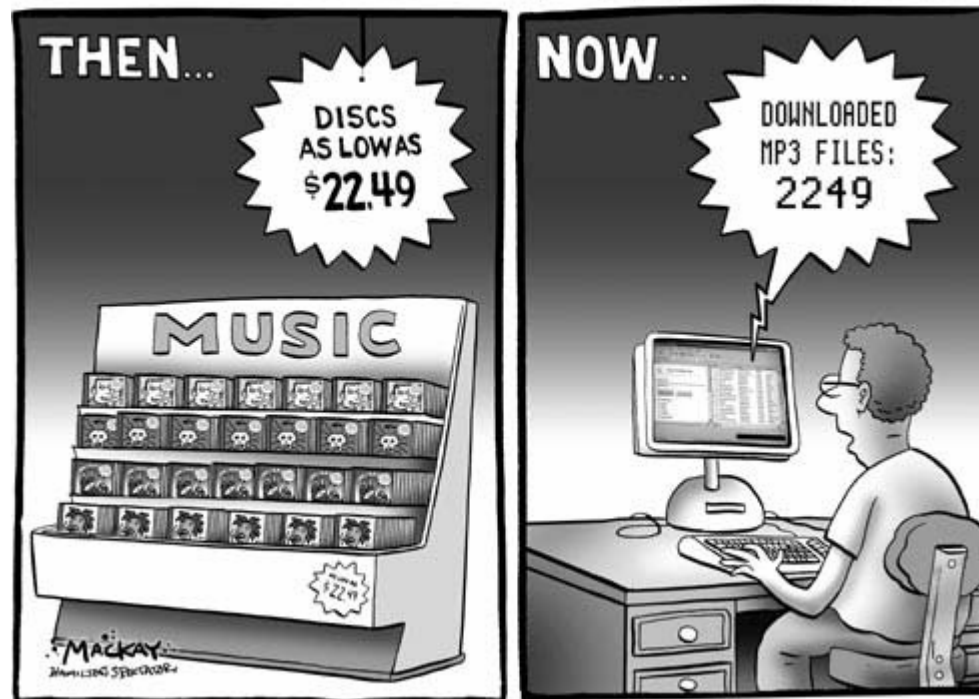
**Graciela Perera and Ken Christensen**  
Computer Science and Engineering  
University of South Florida  
Tampa, FL 33620  
{gpererao, christen}@csee.usf.edu



# Topics


❖ This talk is about a

**New Peer-to-Peer (P2P) protocol that enables all nodes to know the files shared by others**



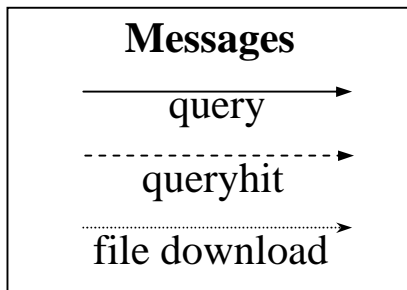
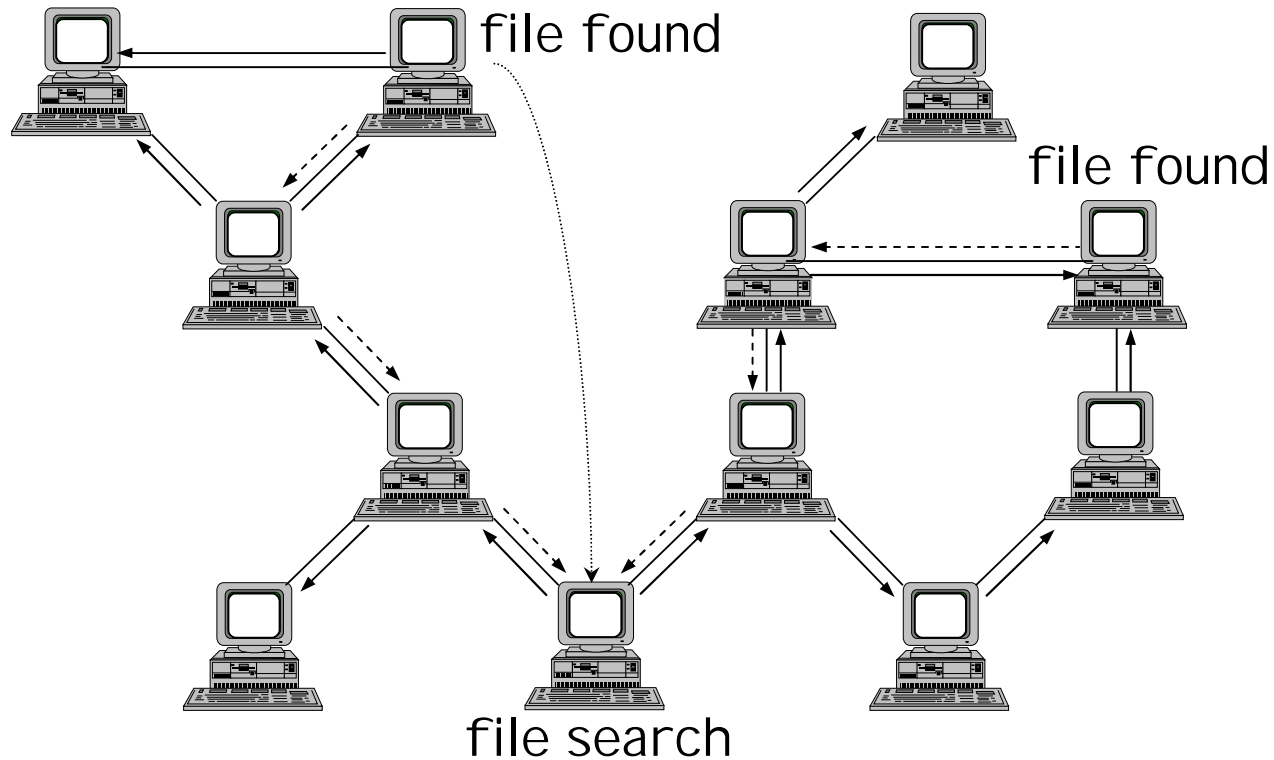
From: <http://cagle.msnbc.com>

# Topics continued

- ❖ Introduction
- ❖ Gnutella and BULLS protocol 
- ❖ Performance model
- ❖ Performance analysis
- ❖ Related Work
- ❖ Conclusions and future work

# Introduction

## ❖ Gnutella



## Introduction continued

- ❖ **Gnutella is decentralized, self-organizing and robust P2P**
- ❖ **Question: How to determine all the files shared in the network?**
  - Query for all files, overhead would be high
- ❖ **If this is possible (all nodes know what others share) then**
  - Power management applications
  - Ethical file sharing
  - Affinity group applications**can be developed**



Gnutella can not be used to enable these novel applications

## Introduction continued

- ❖ Our hypothesis...

**By broadcasting file updates nodes know the files shared by other nodes with an overhead equal or less than Gnutella.**

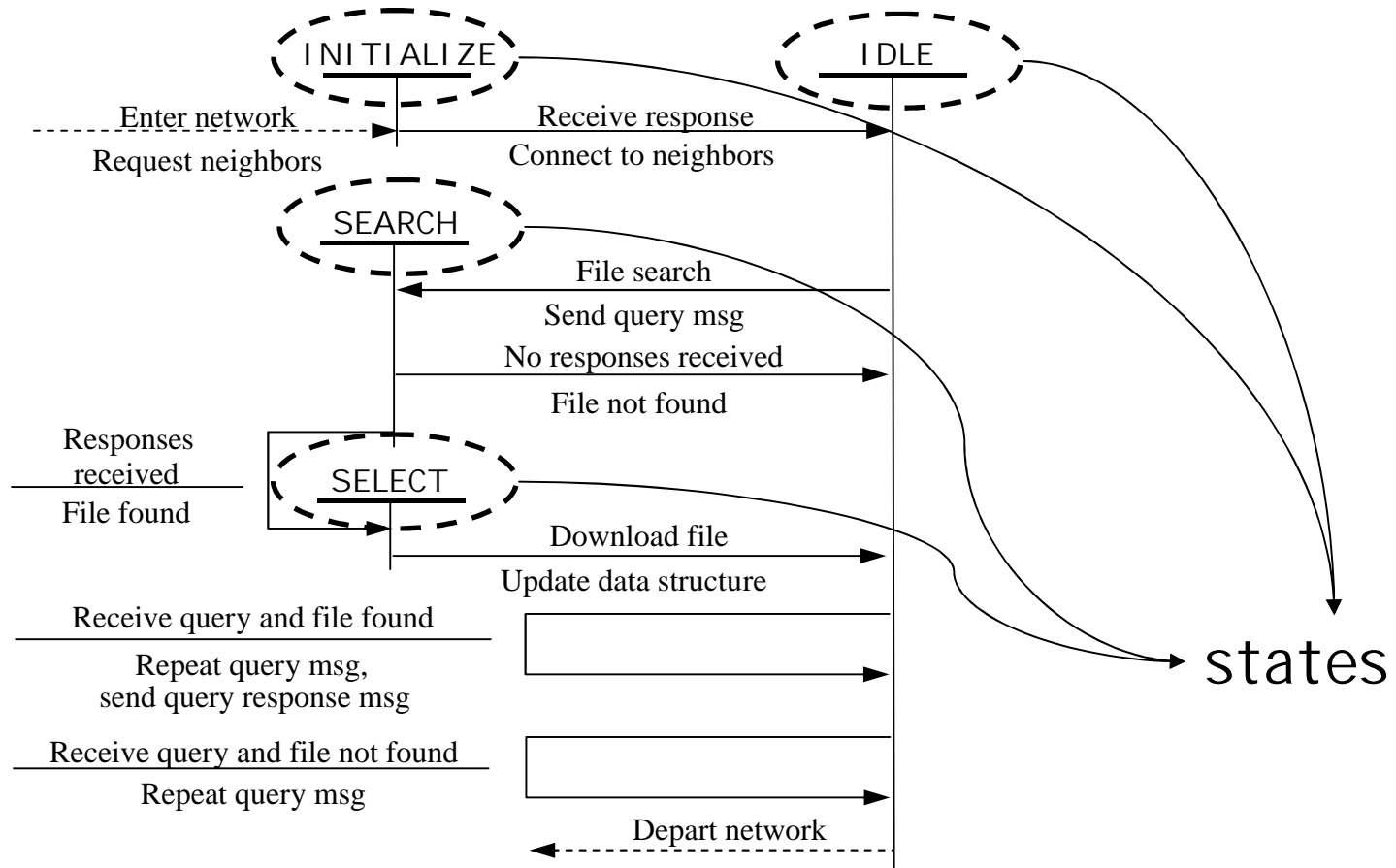
- ❖ **Overhead = number of messages generated to locate a file**
- ❖ **Reduce overhead by broadcasting file updates instead of queries**

# Gnutella and BULLS protocol

- ❖ **Gnutella and BULLS are described using Finite state machine (FSM)**
- ❖ **FSMs cover the behavior related to message exchange**
  - Gnutella query and queryhit messages
  - BULLS file updates messages
- ❖ **Low data rate operations are omitted from the FSMs**

# Gnutella and BULLS protocol continued

## ❖ FSM for Gnutella



# Gnutella and BULLS protocol continued

## ❖ Gnutella can be summarized by

- File search by query broadcast
- Download files directly from the multiple sources sharing the file

## ❖ BULLS

- Gnutella based broadcast updates instead of queries
- Updates correspond to downloading or adding/deleting a file
- Stores the updates in a global directory data structure
- The global directory contains the files shared by each node
  - *<IP address or node name> and <collection of shared file names>*

❖ **BULLS can be summarized by**

- Local look-up file search (no overhead traffic)
- Broadcast file update messages

# Performance model

## ❖ Three assumptions

1. Number of nodes  $N_{nodes}$  remains constant (stable state)
2. A single message from BULLS or Gnutella is equivalent to sending packet in the network
3. Each search is defined to be a one file search (searches are used to locate a single file)

# Performance model continued

## ❖ Independent variables are

$D$  = Node degree

$M$  = Number of files shared per node

$P$  = Probability of a node having a given file

$N_{filename}$  = Number of bytes required to store a filename

$N_{hops}$  = Number of hops (nodes) a query hit travels

$N_{nodename}$  = Number of bytes required to store a node name

$N_{nodes}$  = Number of nodes in the P2P network

$R_{search}$  = Rate of searches per node (messages/sec)

$R_{update}$  = Rate of file list updates per node (messages/sec)

$T_{stay}$  = Time a node stays in the P2P network (sec)

# Performance model continued

## ❖ Investigate protocol feasibility using a flow model for

- Storage in bytes required per node for BULLS ( $S_{bulls}$ )
- BULLS overhead messages rate per node ( $X_{bulls}$ )
- Gnutella overhead messages rate per node ( $X_{gnutella}$ )

## ❖ Reasonable values

- Storage in Mbtyes since hard drive size 100 Gbytes
- Overhead message rate in Kb/sec as broadband (Mb/sec) becomes popular Mb/sec

# Performance model continued

## ❖ Storage requirement for BULLS

$$S_{bulls} = N_{nodes} \left( N_{nodename} + MN_{filename} \right)$$

IP address and name of files stored for a node

Number of nodes



About 374 Mbytes for a typical network about 70,000 nodes with each node sharing 100 files

# Performance model continued

## ❖ Overhead traffic for BULLS

$$X_{bulls} = R_{updates} D(N_{nodes} - 1) + \underbrace{D(N_{nodes} / T_{stay}) (M + 1)}$$

Updates messages rate when  
of nodes enter/depart per node

Updates messages rate when  
files are added/deleted per node

# Performance model continued

## ❖ Overhead traffic for Gnutella

$$X_{gnutella} = R_{search} D(N_{nodes} - 1) + \underbrace{R_{search} N_{hops} P(N_{nodes} - 1)}$$

Queryhit messages rate as  
seen by each node

Query messages rate for a  
file search as seen by each  
node

## Performance model continued

- ❖ Trade-off in overhead traffic between Gnutella and BULLS is a function of
  - Nodes entering and departing the network
  - Number of files shared per node

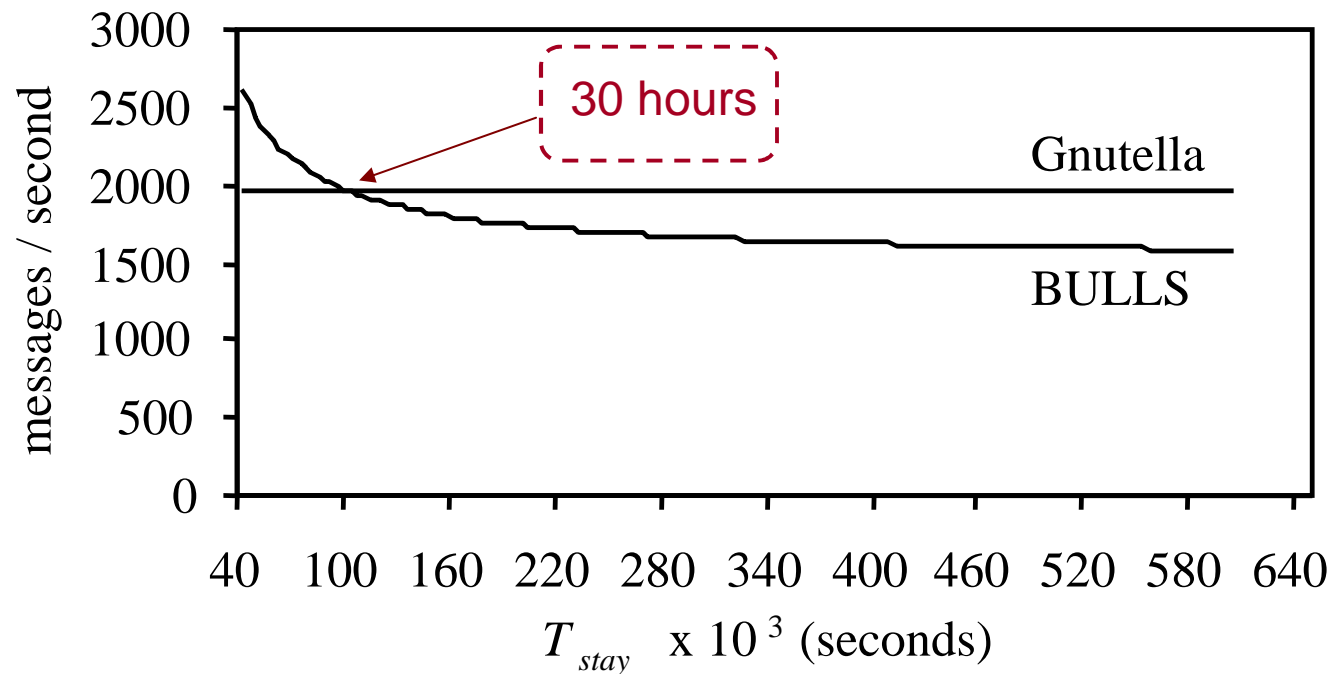
Gnutella                      BULLS

↓                                      ↓

$$R_{search} N_{hops} P(N_{nodes} - 1) > D(N_{nodes} / T_{stay})(M + 1)$$

# Performance analysis

- ❖ Impact of the time a node stays vs. the overhead traffic



# Related Work

- ❖ Flooding is suitable for a wide range of applications
  - Decentralized replica location mechanism for scientific data
  - Wide-area file system in which distributed users can share data
  - Link-state routing protocol OSPF
  
- ❖ These protocols cannot determine the entire set of files shared

## Related Work continued

- ❖ BULLS differs from the existing approaches in two aspects
  - All nodes are knowledgeable of what others share explicitly
  - A different perspective from most unstructured P2P protocols is explored
- ❖ **New applications for a different P2P paradigm remain a challenge.**

# Conclusions and future work

- ❖ **BULLS** is a new paradigm for P2P
- ❖ **BULLS** enables new applications such as Power management and Ethical file sharing.
- ❖ **BULLS** overhead is significantly less (**about 19% less**) to Gnutella
  - applications remain “always on” (about a week)

## Conclusions and future work continued

- ❖ Future work, improving **BULLS**
  - Broadcasting updates when file availability is affected
  - Merging multiple updates into one
- ❖ **BULLS** can enable P2P networks to be energy efficient and investigated as the set covering problem

# Acknowledgement

## ❖ The authors would like to thank...

- Chamara Gunaratne and Cesar Guerrero graduate students at the University of South Florida for their valuable comment