

Gnutella Dynamic Network Creation

**Anne Fouron &
Brian Fraser**

agfouron@sfu.ca,
bfraser@sfu.ca

RoadMap

- Gnutella
- Our Enhancement: Connectpeers
- Implementation in ns-2
- Results
- Conclusions et. al.



Life As We Currently Know it:

Gnutella

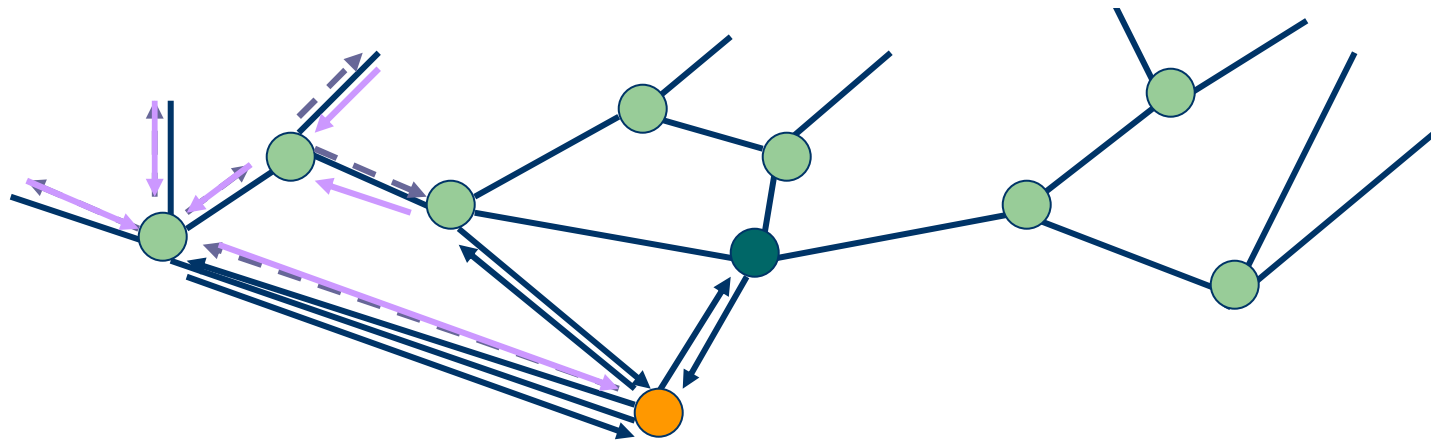


Gnutella: Overview

- peer-to-peer protocol
 - distributed, decentralized file sharing
 - every node (“servent”) is both a server and a client
 - no fixed topology
 - v. 0.6 of protocol is a two-layered hierarchy
 - supernodes (called ultrapeers)
 - client nodes (called shielded leafs)
 - Ultrapeers: faster, better networking and CPU power
 - act as proxies for client(leaf) nodes connected to them
 - shield leaf nodes from almost all ping and query traffic

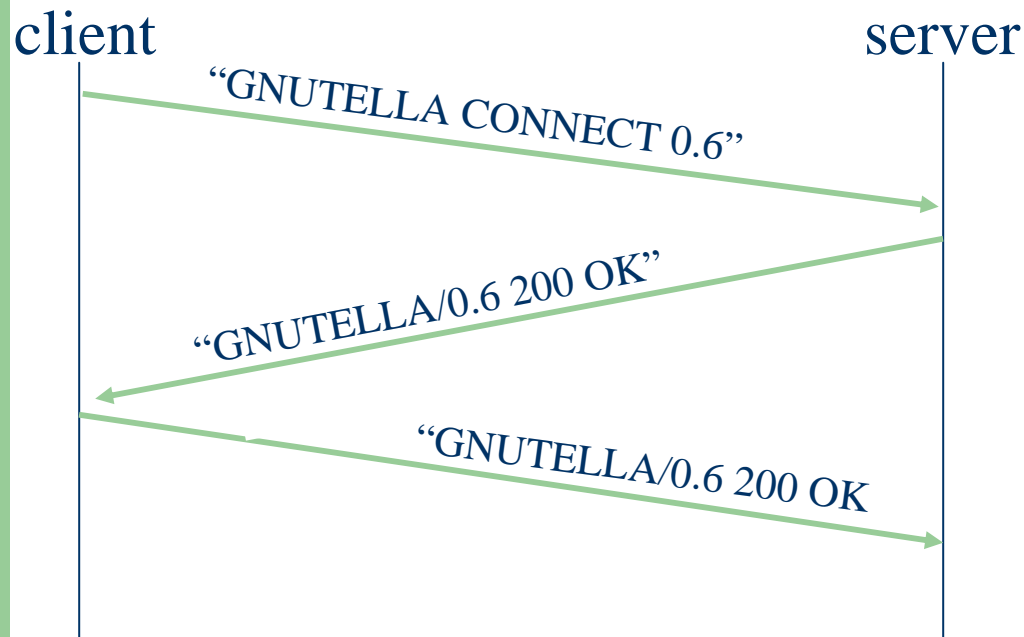
Gnutella: Connection Process

4. File downloads are done directly between two peers using GET/PUSH messages using HTTP protocol



Gnutella: 3 Way Connection Handshake

- Client evaluates capability headers to decide if it still wishes to connect to server
 - If yes, client sends a “GNUTELLA/0.6 200 OK” message



Capability headers:

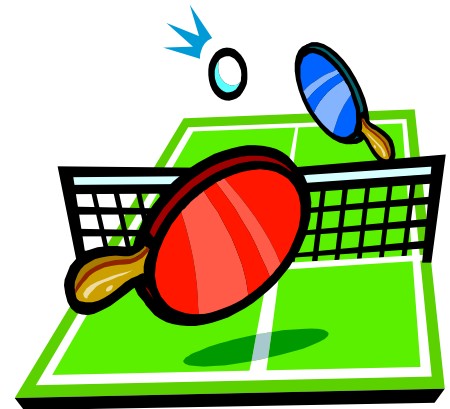
- gnutella client version
- version of query routing
- connection negotiation

X-Try headers:

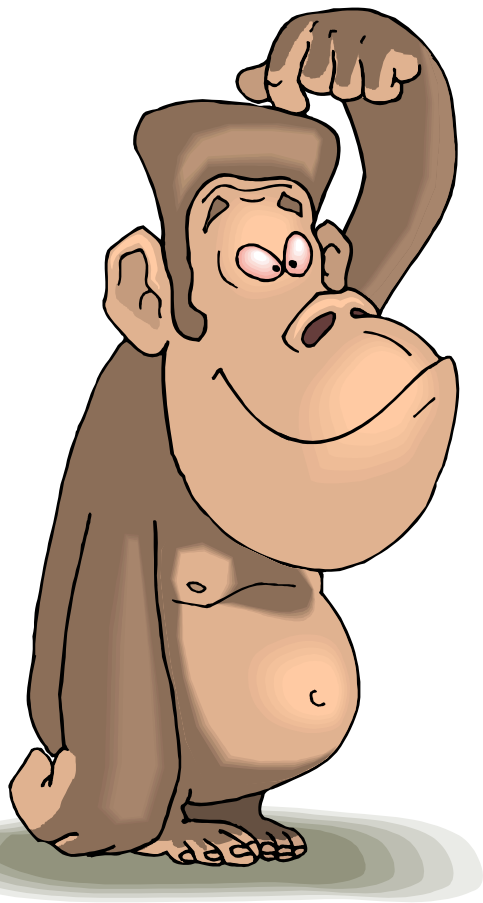
- list of nodes to connect to
- each has IP address of node and its listening port

Gnutella: Ping Pong Scheme

- After a servant joins network, sends out PING
 - PING has unique ID
- A servant that sees a PING
 - sends a PONG back to originator
 - forwards the PING to its neighbours
- On receive PONG
 - use routing table
 - routes PONG on connection that sent corresponding PING

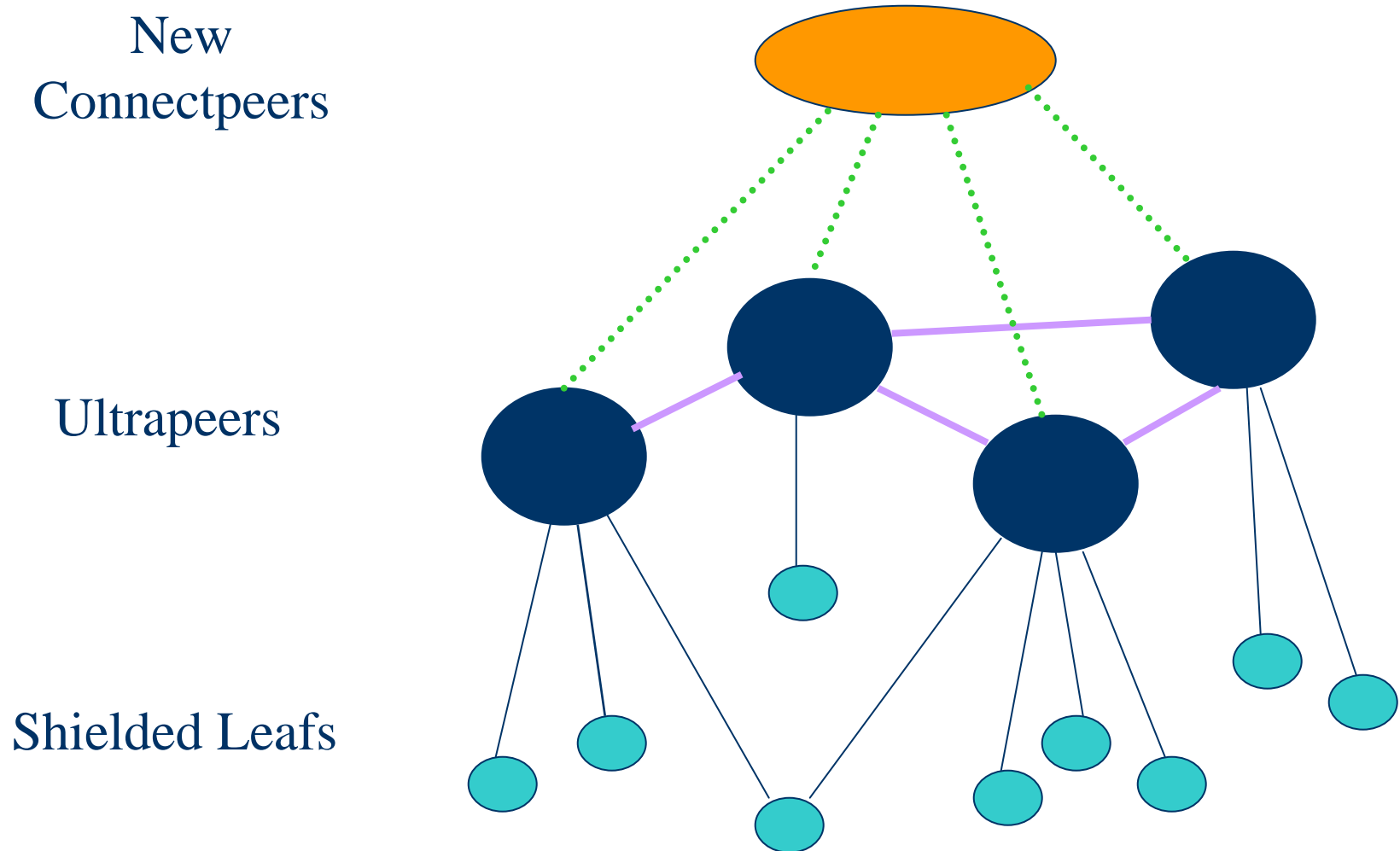


Our new idea!

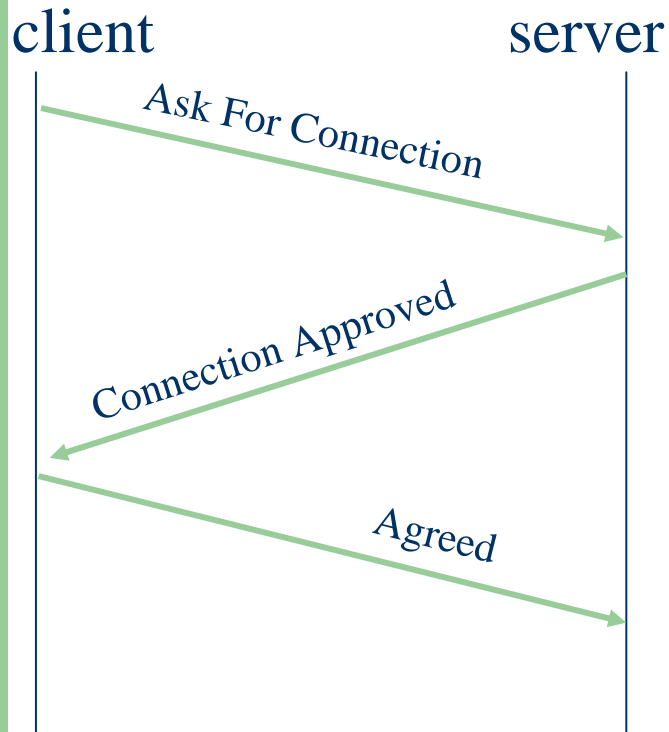


Connectpeer

Improved Gnutella: 3 layer architecture



Improved Gnutella: Updated Handshake



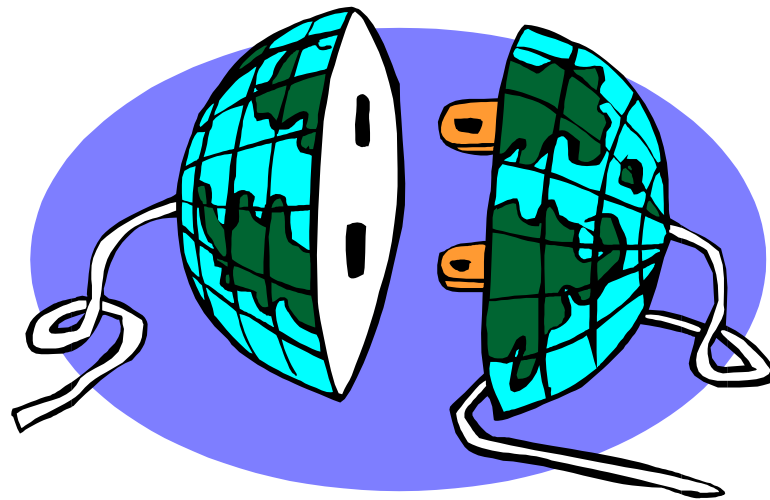
- Existing capability headers:
 - X-Ultrapeer
 - X-Ultrapeer-Needed
- New capability headers:
 - X-Connectpeer
 - X-Connectpeer-Needed

Project Goals

- Look at dynamic creation of network connections
 - Current standard (version 0.6 of protocol)
 - Our idea: Connectpeer
- Hope:
 - Connectpeers will accelerate connection process



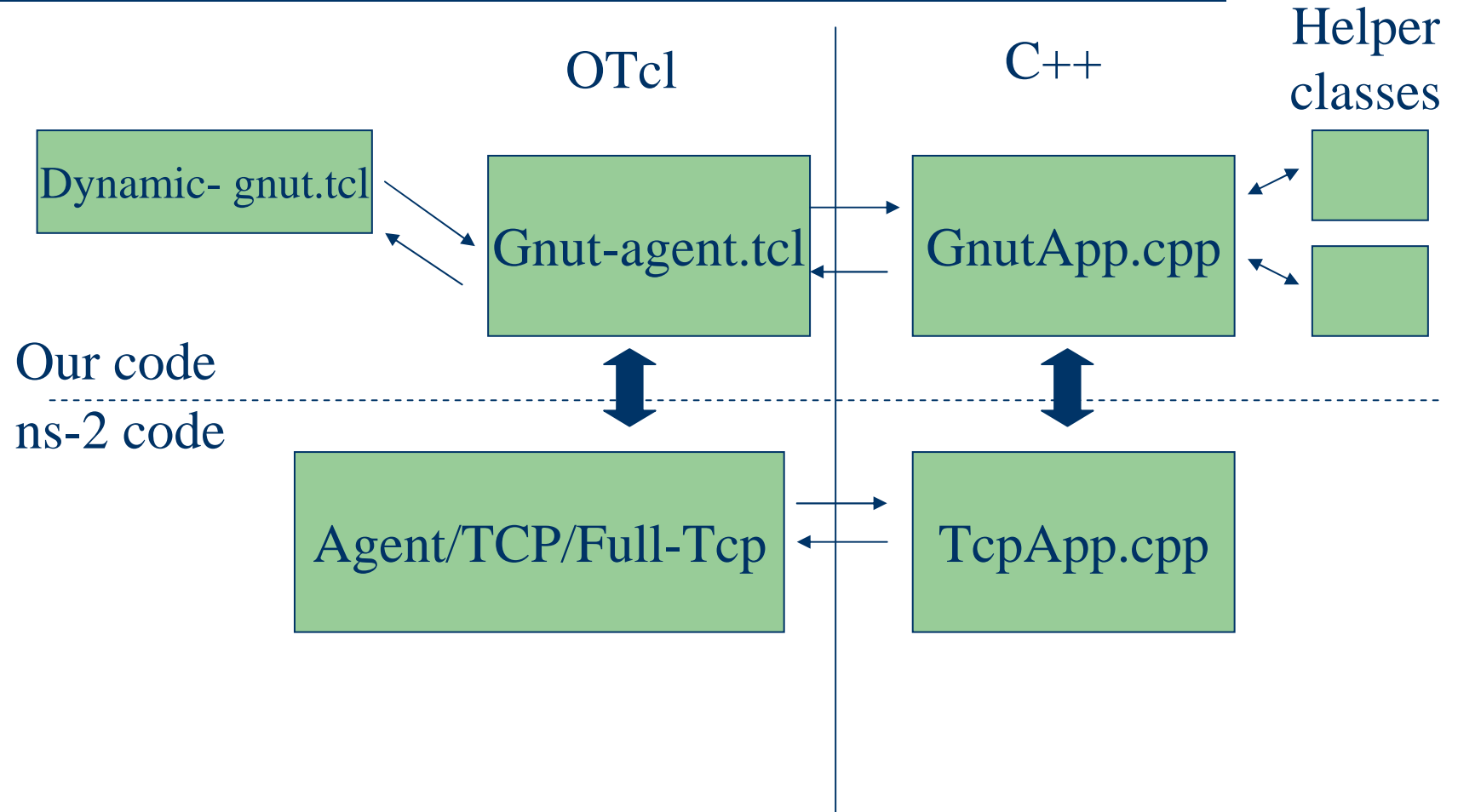
Implementation



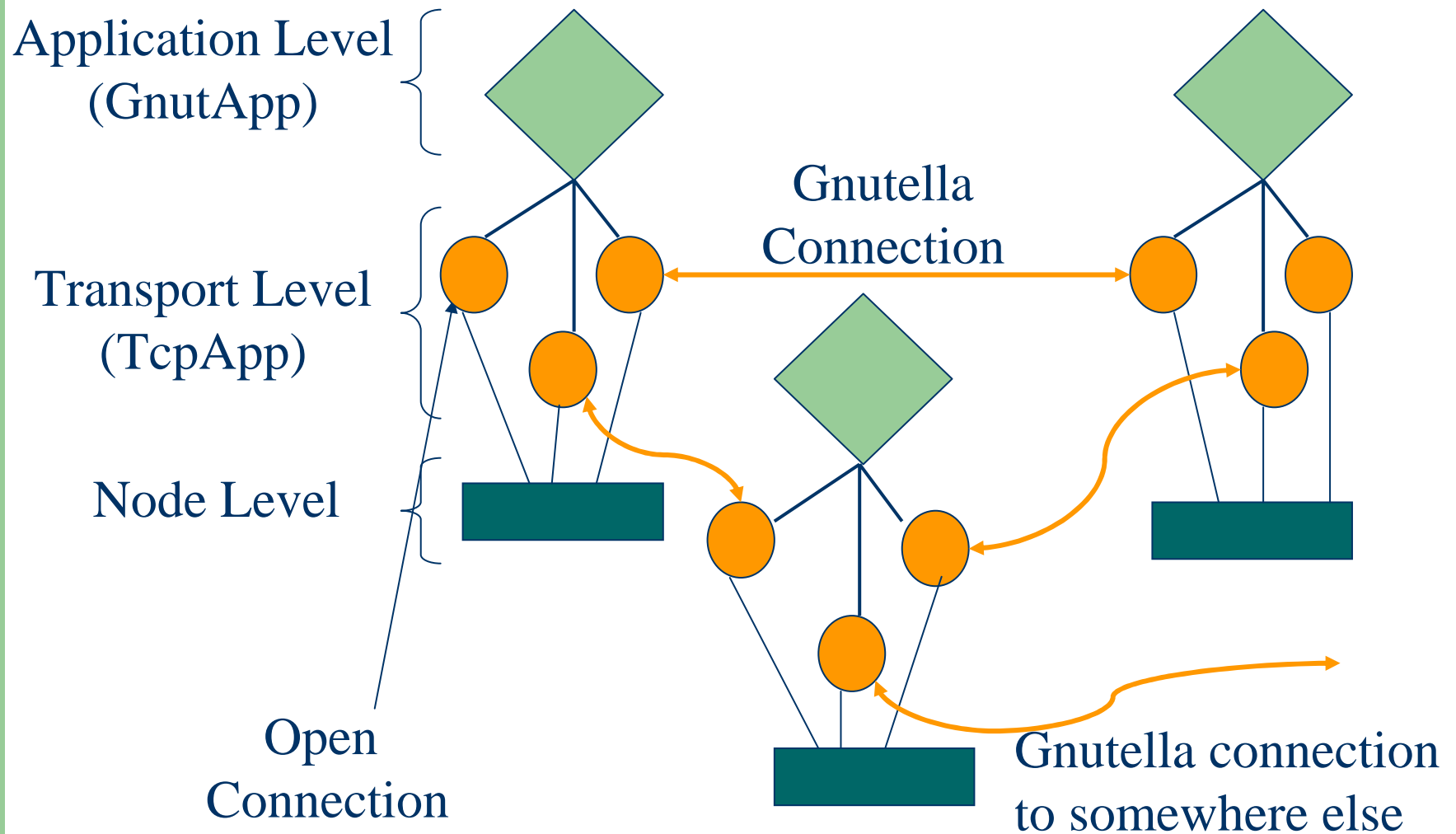
Interesting Requirements

- Dynamic topology
 - Connections and nodes
- Meaningful message data
 - Not just number of bytes
- Multiple connections per node
- Statistics

High Level Design



Implementation: GnutApp Architecture

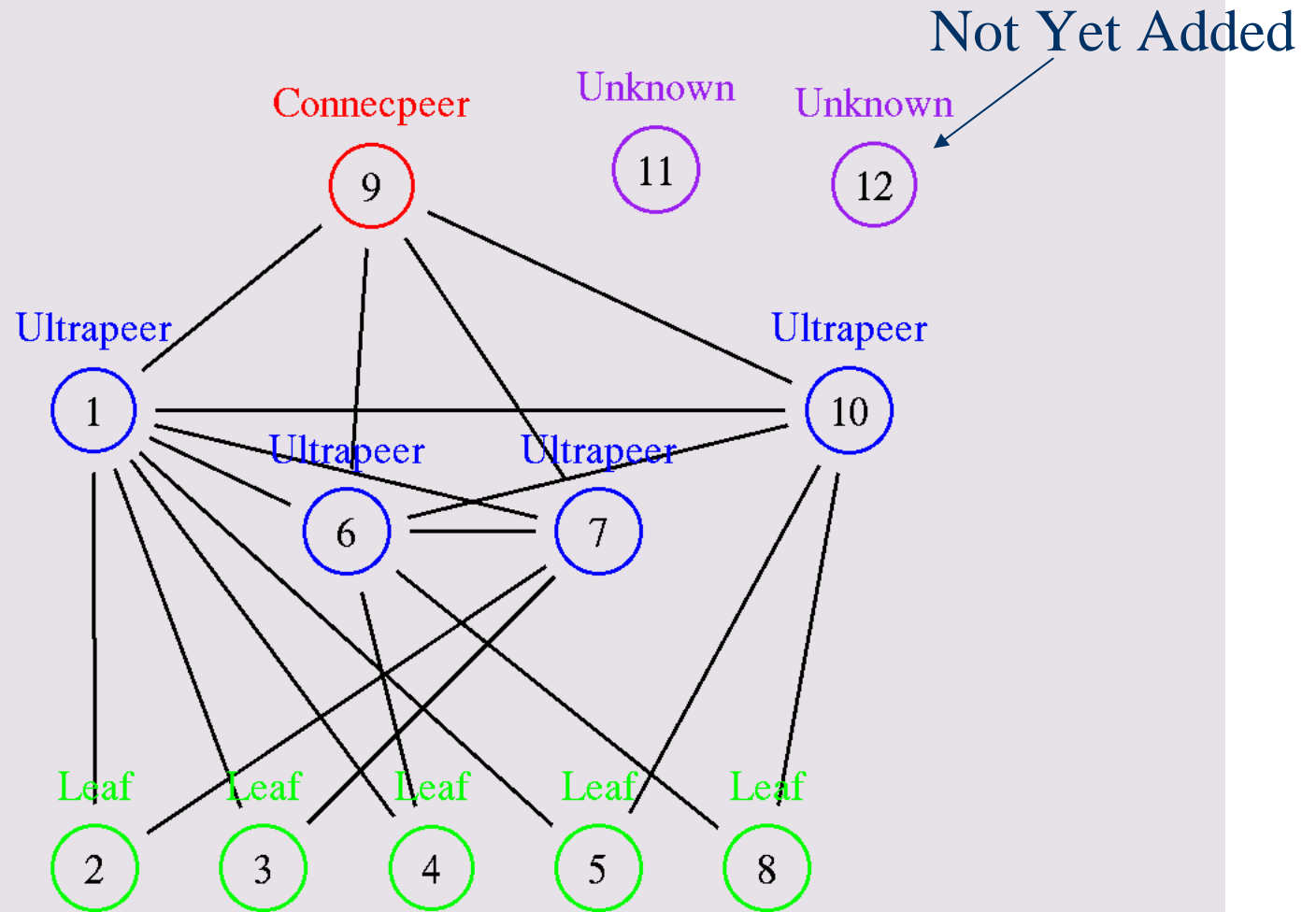


Implementation: Statistics & Results

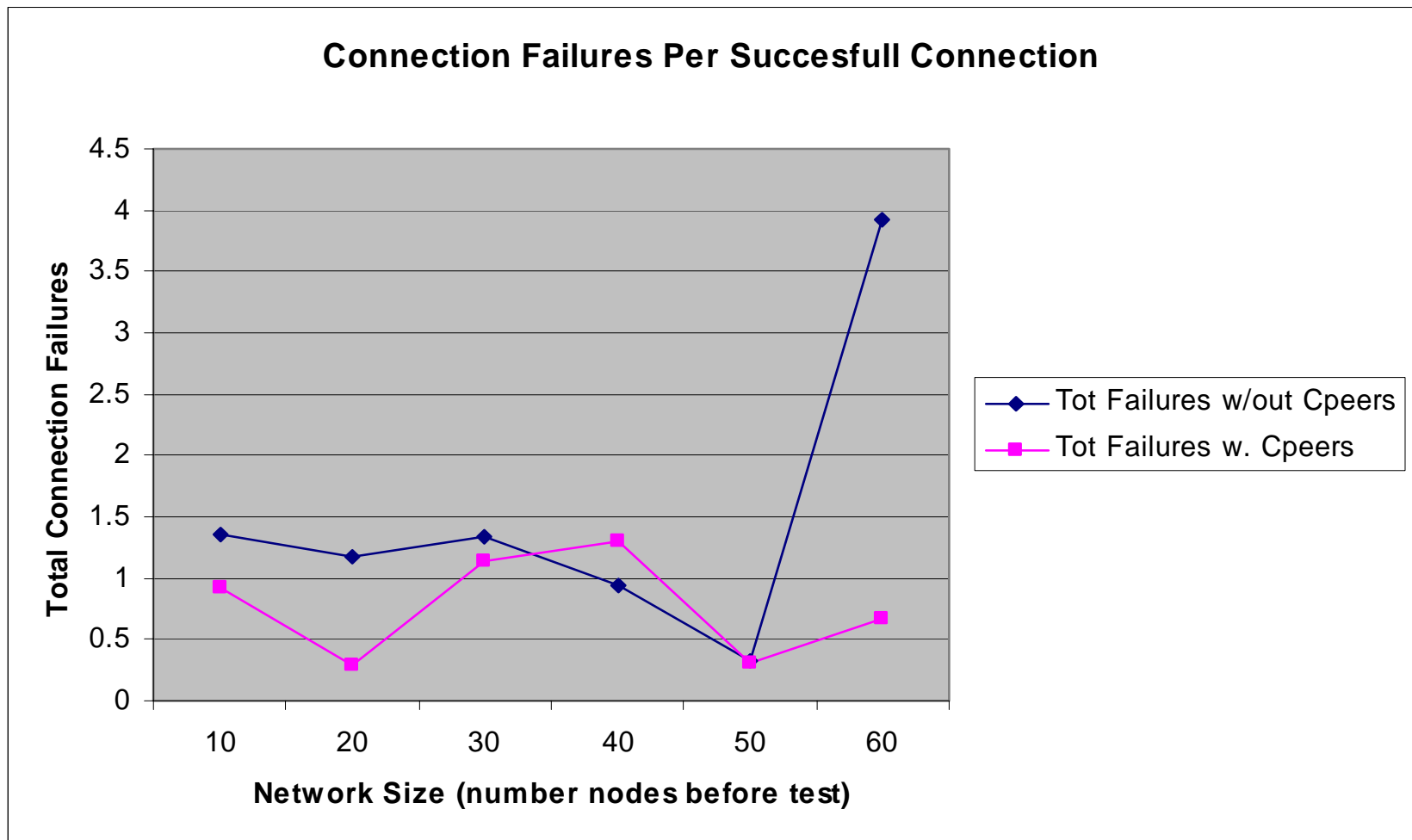
- # Connection attempts vs. # Good connections
 - With & without connectpeers
 - Different network sizes
- Expect
 - Shorter connection times
- NAM Visualization



Implementation: NAM Visualization



Results



Conclusions

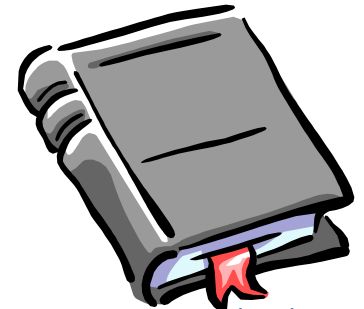


Conclusions

- Gnutella is optimized for network traffic, not new connections
- 3-Way handshake supports powerful connection negotiation
- Ns-2 is a powerful framework to simulate virtual networks
- Connectpeers:
 - Can be added to existing protocol framework
 - Results are inconclusive as to benefits

References

1. Bildson, G. “PROPOSAL: Handshaking Protocol”, The Gnutella Developer Forum (GDF), August 2001.
http://groups.yahoo.com/group/the_gdf/message/2010
2. Klingberg, T., Manfredi, R., “Gnutella 0.6”, Network Working Group, June 2002.
<http://rfc-gnutella.sourceforge.net/draft.txt>
3. Ripeanu, M. “Peer-to-Peer Architecture Case Study: Gnutella Network”, Proceedings of the First International Conference on Peer-to-Peer Computing (P2P’01), August 2001.
<http://people.cs.uchicago.edu/~matei/PAPERS/gnutella-rc.pdf>
4. Rohrs, C., “Query Routing for the Gnutella Network”, May 2002.
www.limewire.com/developer/query_routing/keyword%20routing.htm
5. Rohrs, C., Singla, A. “Ultraplayers: Another Step Towards Gnutella Scalability”, November 2002.
www.limewire.com/developer/Ultraplayers.html



2003/10/21

Questions and Demo

