GNUTELLA

A Summary Of The Protocol and it’s Purpose

By

Fr. Gayatri Tribhuvan

University of Freiburg
OUTLINE

- Introduction
- Structure
- Functioning Mechanisms
- Features
- Gnutella 2
- Conclusion
INTRODUCTION
What is Gnutella?

- Modern and efficient P2P network standard and architecture

- Provides services such as person to person communication, data location and transfer
History of Gnutella

- Originally conceived of by Justin Frankel, 21 year old founder of Nullsoft.
- There were *multiple* open source implementations at http://sourceforge.net/ including:
  - Jtella
  - Gnucleus
- Software released under the Lesser Gnu Public License (LGPL)
- The Gnutella protocol was widely analyzed
**GOALS**: Why is Gnutella needed?

- Open architecture
- Allows diversity
- Integrity.
- Good performance.
STRUCTURE
Scope of Gnutella

- Consists of two components:
  1. Gnutella standard
  2. Gnutella network
Gnutella network

- Easily recognizable component
- Built on a high performance P2P architecture
- On this a variety of applications can run and also be developed
- Examples of applications that can be built on this are, file sharing applications, communication tools etc.
- Any client that joins this network becomes a SERVENT.
Gnutella standard

- Set of applications for building requirements that run on the Gnutella network
- Specifies minimum compliance level required for an application to run on Gnutella.
- It recognizes if an application is Gnutella compatible or not.
- A standard is needed for Gnutella to allow diverse applications to interact with each other.
Some requirements what applications must have to be compatible with Gnutella: EXAMPLES

- Bidirectional TCP stream
- Routing maintenance
- Reverse connection response
- HTTP/1.1 for client server in P2P transactions
Gnutella protocol messages

- Broadcast Messages
  - Ping
  - Query

- Back-Propagated Messages
  - Pong
  - Query Hit

- Node-to-Node Messages
  - PUSH
Ping

- Discovers hosts on the network actively.
- Probes the networks for other Servents.
- Represented by descriptor header where: Payload Descriptor field is 0x00
Pong

- Reply to a Ping message

- Contains address of active Gnutella Servent

- More than One pong can be sent in reply to one Ping.
<table>
<thead>
<tr>
<th>Byte offset</th>
<th>Port</th>
<th>IP Address</th>
<th>Number of Files Shared</th>
<th>Number of Kilobytes Shared</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>5</td>
<td>6</td>
</tr>
</tbody>
</table>

**Port**  
The port number on which the responding host can accept incoming connections.

**IP Address**  
The IP address of the responding host.

*This field is in big-endian format.*

**Number of Files Shared**  
The number of files that the servent with the given IP address and port is sharing on the network.

**Number of Kilobytes Shared**  
The number of kilobytes of data that the servent with the given IP address and port is sharing on the network.
Query

- Primary mechanism for searching the distributed network

- Responds with query hit after searching its local data set.
- Minimum speed in Kbps that servents should respond to this message.

- Null terminated search string. Bound by the payload length field of descriptor header.
Query Hit

- Reply to Query.

- Must be generated only if the search criteria is accurately met.

- Has the same descriptor identifier as Query.
QueryHit (0x81)

<table>
<thead>
<tr>
<th>Byte offset</th>
<th>Number of Hits</th>
<th>Port</th>
<th>IP Address</th>
<th>Speed</th>
<th>Result Set</th>
<th>Servent Identifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>6</td>
<td>7</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>...</td>
</tr>
<tr>
<td>n</td>
<td>n + 16</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Result set

<table>
<thead>
<tr>
<th>Byte offset</th>
<th>File Index</th>
<th>File Size</th>
<th>File Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>3</td>
<td>4</td>
<td>7</td>
</tr>
</tbody>
</table>
Push

- Allows for data access from firewalled servent

- Servent sending Query hit is behind a firewall.
<table>
<thead>
<tr>
<th>Byte offset</th>
<th>Servent Identifier</th>
<th>File Index</th>
<th>IP Address</th>
<th>Port</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>15</td>
<td>16</td>
<td>19</td>
<td>20</td>
</tr>
</tbody>
</table>
FUNCTIONING MECHANISMS
How does a servent join Gnutella?

- Tries to establish connection with a fellow servent.

- TCP connection established once address of the other servent is obtained.
Following Gnutella connection request string is sent:
Gnutella Connect/ <protocol version String>
Servent accepting the connection responds:
Gnutella OK
## Descriptor Header

<table>
<thead>
<tr>
<th>Byte offset</th>
<th>Descriptor ID</th>
<th>Payload Descriptor</th>
<th>TTL</th>
<th>Hops</th>
<th>Payload Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>15</td>
<td>16</td>
<td>17</td>
<td>18</td>
<td>19</td>
</tr>
</tbody>
</table>
Payload Descriptor takes the following values:

0x00  Ping
0x01  Pong
0x40  Push
0x80  Query
0x81  Query hit
Gnutella search mechanism

Steps:
- Node 2 initiates search for file A
Gnutella Search Mechanism

Steps:
- Node 2 initiates search for file A
- Sends message to all neighbors
Steps:
- Node 2 initiates search for file A
- Sends message to all neighbors
- Neighbors forward message
Gnutella Search Mechanism

Steps:
- Node 2 initiates search for file A
- Sends message to all neighbors
- Neighbors forward message
- Nodes that have file A initiate a reply message
Gnutella Search Mechanism

Steps:
- Node 2 initiates search for file A
- Sends message to all neighbors
- Neighbors forward message
- Nodes that have file A initiate a reply message
- Query reply message is back-propagated
Gnutella Search Mechanism

Steps:
- Node 2 initiates search for file A
- Sends message to all neighbors
- Neighbors forward message
- Nodes that have file A initiate a reply message
- Query reply message is back-propagated
Steps:
• Node 2 initiates search for file A
• Sends message to all neighbors
• Neighbors forward message
• Nodes that have file A initiate a reply message
• Query reply message is back-propagated
• File download
• Note: if one client, X, is behind a firewall, Y can request that X push the file to Y
Descriptor Routing

- Necessary to control traffic in Gnutella network.
- Well behaved Gnutella servent routes descriptors in the following manner
Pong Descriptors are sent along the same route Ping Descriptors traveled.
- Query hits are sent along the same path as the query descriptors traveled.

- Push descriptors are sent along the same path of the incoming query hit descriptors.
Servent sends incoming ping and query messages to all directly connected servents, except the servent that initiated the message.

TTL decremented, and Hops field incremented, before descriptors are forwarded. If TTL is 0, descriptors are stopped being forwarded.
If the payload descriptor and descriptor ID is the same as received before, these descriptors should not be forwarded.
File Downloads

- Direct connection between source and target
- File is never transferred over the Gnutella network.
File download protocol is HTTP.

GET /get/<File Index>/<File Name>/ HTTP/1.0


Example

<table>
<thead>
<tr>
<th>File Index</th>
<th>2468</th>
</tr>
</thead>
<tbody>
<tr>
<td>File Size</td>
<td>4356789</td>
</tr>
<tr>
<td>File Name</td>
<td>Foobar.mp3</td>
</tr>
</tbody>
</table>

GET /get/2468/Foobar.mp3/ HTTP/1.0\r\n
Features of Gnutella

- **Scalability**
  When a node receives a ping/query message, it forwards it to the other nodes.

- **Existing mechanisms to reduce traffic**
  - TTL counter
  - Cache information about messages they received, so that they don't forward duplicated messages.

- **Anonymity**
  Gnutella provides for anonymity by masking the identity of the peer that generated a query.
Disadvantages

- Limited number of hosts being searched leads to a limited number of hits
- The search result is not accurate due to the search by filename
- An inefficient use of computer resource - spending too much resource on handling other peers' queries.
There was significant growth in the Gnutella network in 2001

- 5,000 nodes on February 2001
- 10,000 nodes on March 19, 2001
- 20,000 nodes on May 12, 2001
- 40,000 nodes on May 29, 2001

Statistics due to Matei Ripeanu, see

Growth Factors

- DSL and cable modem nodes grew substantially
- Multiple client implementations became available
Gnutella2

- Uses extendible XML packet format
- Uses SHA-1 hashes for file identification and secure integrity check of files.
- Uses a metadata system to ensure a quality search of files than merely searching based on file name.
How it Works

- Divides network into hubs and leaves
- Hubs are connected to each other and can have as many leaves as possible
- Searches only the required hubs mentioned in the list
- Doesn’t overload the network
Details of files at leaves are maintained by query routing tables in the hub, and forwarded to other hubs.

Reduces unnecessary flooding.
Gnutella 2 versus Gnutella

- Gnutella 2 relies on UDP
- Search Mechanism
  - Gnutella- Query flooding
  - Gnutella 2- Walk mechanism
- Searching node collects information only from the necessary hubs.
Conclusion

- Gnutella is a self-organizing, large-scale, P2P application that produces an overlay network on top of the Internet

- Since there is no central authority the open source community must commit to making any changes
THANK YOU