CS 580 Client-Server Programming
Spring Semester, 2004
Doc 20 Gnutella

Contents

Reference .................................................................................................................. 1
Gnutella .................................................................................................................. 2
  Requests and Responses ...................................................................................... 5
  Header ................................................................................................................. 6
  Ping 0x00 ........................................................................................................... 8
  Pong 0x01 .......................................................................................................... 9
  Query 0x80 ......................................................................................................... 10
  QueryHit (0x81) ............................................................................................... 11
  Push (0x40) ....................................................................................................... 14
  Some Routing .................................................................................................... 15
  File Downloads ................................................................................................. 16

Reference

The Gnutella Protocol Specification v0.4, Document Revision 1.2,
http://www9.limewire.com/developer/gnutella_protocol_0.4.pdf

Copyright ©, All rights reserved. 2004 SDSU & Roger Whitney, 5500 Campanile Drive, San Diego, CA 92182-7700 USA. OpenContent (http://www.opencontent.org/opl.shtml) license defines the copyright on this document.
Gnutella

- Peer-to-peer
- Gnutella program is both a server and a client: servent
- No central server
- Protocol does not discuss how one knows about other servants
Basic Idea

Servent connects to 1 or more remote servents

Can
• Ping the network
• Send a request for a file to see who has it

To get a file from a servent
• Connect to the servent directly with http request
Basic Protocol

Connect to another servent with

    GNUTELLA CONNECT/<protocol version string>

Where <protocol version string> is 0.4

If the remote servent accepts the connection it must respond with

    GNUTELLA OK

Both servents then can then send messages
Requests and Responses

Ping – who is on the network

Pong – response to a ping

Query – search the network for data

QueryHit – response to query

Push – Used to allow servents work behind firewall

Each Request/Response starts with a header
Header

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

Descriptor ID

16 byte string
- Uniquely identifies Request/Response

Payload Descriptor

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00</td>
<td>Ping</td>
</tr>
<tr>
<td>0x01</td>
<td>Pong</td>
</tr>
<tr>
<td>0x40</td>
<td>Push</td>
</tr>
<tr>
<td>0x80</td>
<td>Query</td>
</tr>
<tr>
<td>0x81</td>
<td>QueryHit</td>
</tr>
</tbody>
</table>

TTL

Time to live

Number of times message will be forwarded by servents

Many servents will set TTL to 5 if is it larger

Each servent that gets the message reduces TTL by one before forwarding the message
Hops

Number of times message has been forwarded

Each servant that gets the message increase Hop by one before forwarding

Payload Length

Length of rest of message
Ping 0x00

No more content other than header
**Pong 0x01**

Sent only in response to a ping

Servent can cache pongs of other servents

<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>
<tr>
<td></td>
<td></td>
<td></td>
<td>9</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>13</td>
<td></td>
</tr>
</tbody>
</table>

**Port**

Port that responding servent can accept incoming connections

**IP Address**

IP Address of responding servent

This field uses big-endian format
Query 0x80

<table>
<thead>
<tr>
<th>Byte offset</th>
<th>Minimum Speed</th>
<th>Search Criteria</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>…</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Minimum Speed**

Minimum speed (of connection) in kb/second of servents that should respond to this message

**Search Criteria**

Nul (0x00) terminated search string

Length of string must be included in the payload length field
QueryHit (0x81)

Sent in response to a Query

Descriptor ID in header should contain same value as the Query

<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>10</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>

**Number of hits**

Number of hits in the result set

**Port**

Port number on which responding servent can accept incoming connections

**IP Address**

IP Address of responding servent

This field uses big-endian format

**Speed**

Speed of responding host’s connection in kb/second
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>
<tr>
<td>8</td>
<td></td>
<td></td>
<td>...</td>
</tr>
</tbody>
</table>

File Index
A number used by host to identify the file

FileSize
Size in bytes of the file

FileName
Double-nul (0x0000) terminated name of the file

Servent Identifier
A 16-byte string uniquely identifying the responding servent on the network.

“This is typically some function of the servent’s network address”
### Extended Query Hit

<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>Trailer</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>10</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>…</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>n</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>m</td>
<td>m+1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>m+1</td>
<td>m+17</td>
</tr>
</tbody>
</table>

### Trailer

<table>
<thead>
<tr>
<th>Byte offset</th>
<th>Vender Code</th>
<th>Open Data Size</th>
<th>Open Data</th>
<th>Private data</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>n</td>
</tr>
</tbody>
</table>

How do we know if the trailer exists?

How do we know the length of the private data?
## Push (0x40)

<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></td>
<td>15</td>
<td>16</td>
<td>19</td>
</tr>
<tr>
<td></td>
<td></td>
<td>20</td>
<td>23</td>
<td>24</td>
</tr>
<tr>
<td></td>
<td></td>
<td>25</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Servent Identifier

A 16-byte string uniquely identifying the servent on the network that should push the file.

### File Index

Index of the file to push.

### IP Address

IP Address of to which the file should be pushed.

This field uses big-endian format.

### Port

Port to which the file should be pushed.
Some Routing

Pong messages

Can only be send along path the carried the Ping

Servents should not forward a pong if they did not see the ping

QueryHit

Can only be send along path the carried the Query

Servents should not forward a query hit if they did not see the query

Push

Can only be send along path the carried the QueryHit

Servents should not forward a push if they did not see the query hit

Forwarding

Forward all Ping and Queries to all directly connected servents except to the one that sent it

Decrement TTL and increment Hops field

Don’t forward messages that you have seen before
File Downloads

In response to a QueryHit download the file by using http.

Request the file uses following format:

GET /get/<File Index>/<File Name>/ HTTP/1.0
Connection: Keep-Alive
Range: bytes=0-
User-Agent: Gnutella

Remote server responses with:

HTTP 200 OK
Server: Gnutella
Content-type: application/binary
Content-length: fileSize