

CS 580 Client-Server Programming
Spring Semester, 2009
Doc 16 Size & Gnutella
9 Apr 2009

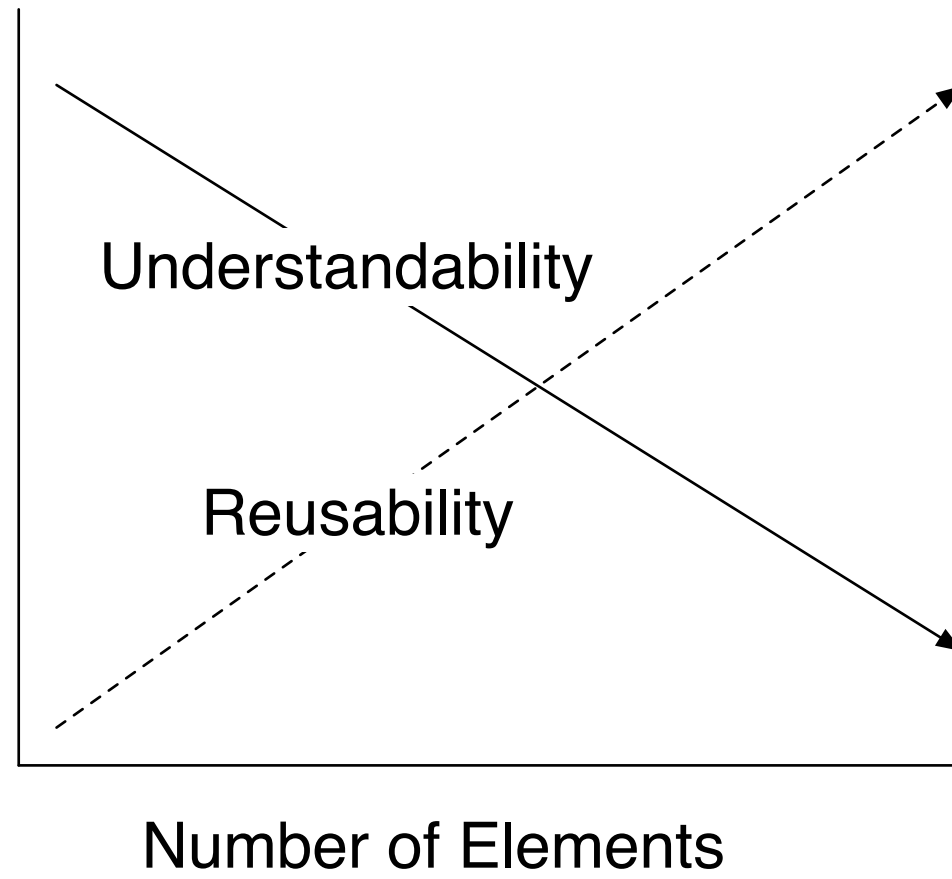
Copyright ©, All rights reserved. 2009 SDSU & Roger Whitney, 5500 Campanile Drive, San Diego, CA 92182-7700 USA. OpenContent (<http://www.opencontent.org/opl.shtml>) license defines the copyright on this document.

Reference

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

Refactoring in Large Software Projects, Roock, Lippert, Wiley, 2003

Trade-off



Rule of Thumb

If a programming element contains more than 30 subelements it is highly likely that there is a serious problem

Methods

30 (or less) lines of code on average

Class

30 (or less) methods on average

Package

30 or few classes

Subsystems

30 or fewer packages

Eclipse 2.1

Total

Line of source code	730,000
Packages	460
Classes	12,400
Methods	89,500

Averages

LOC per method	8.2
Methods per class	7.2
Classes per package	27.0

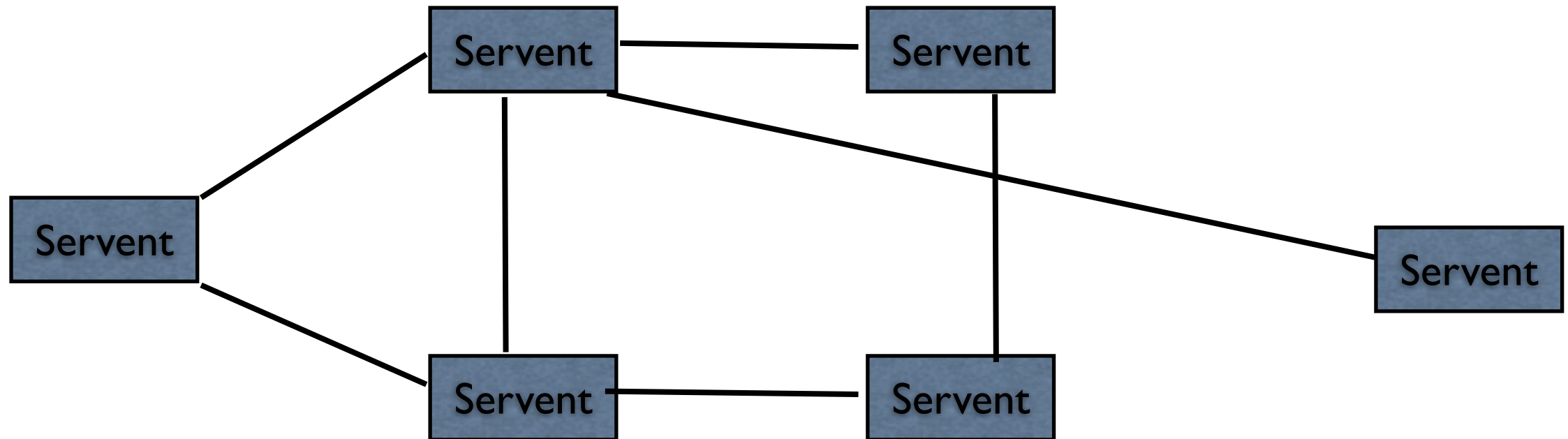
Gnutella

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 servents



Basic Operation

Servent connects to 1 or more remote servents

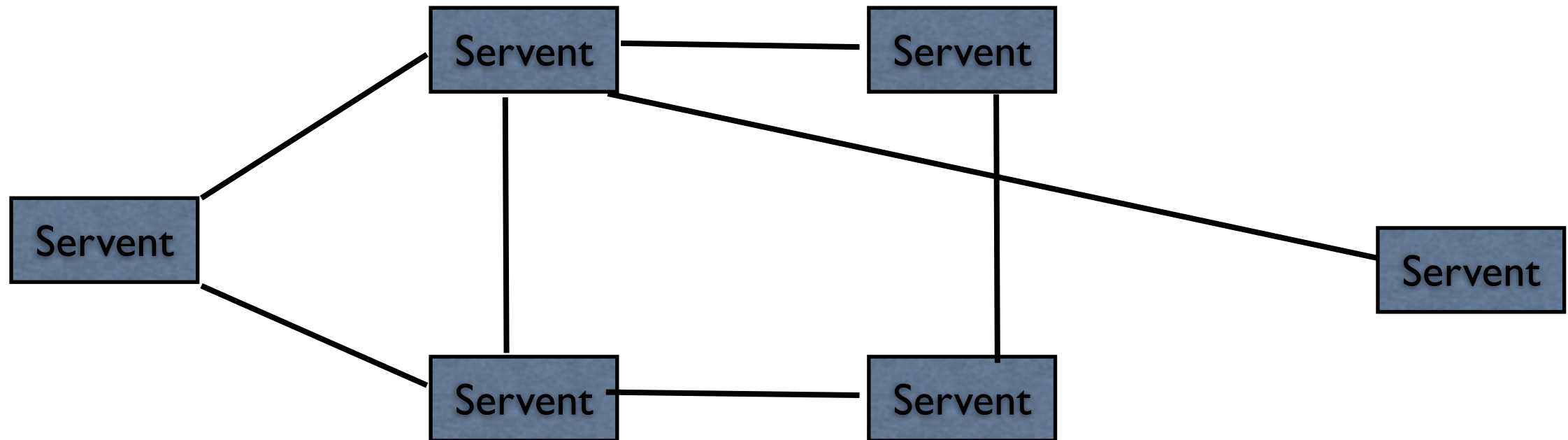
Can

- 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 server with

```
GNUTELLA CONNECT/<protocol version string>\n\n
```

Where <protocol version string> is 0.4

If the remote server accepts the connection it must respond with

```
GNUTELLA OK\n\n
```

Both servers 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 servers work behind firewall

Each Request/Response starts with a header

Header

	Descriptor ID		Payload Descriptor	TTL	Hops	Payload Length	
Byte offset	0	15	16	17	18	19	22

Descriptor ID

16 byte string

Uniquely identifies Request/Response

Payload Descriptor

Value	Meaning
0x00	Ping
0x01	Pong
0x40	Push
0x80	Query
0x81	QueryHit

TTL

Time to live

Number of times message will be forwarded by servants

Many servants will set TTL to 5 if it is larger

Each servant that gets the message reduces TTL by one before forwarding the message

Header

Hops

Number of times message has been forwarded

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

Payload Length

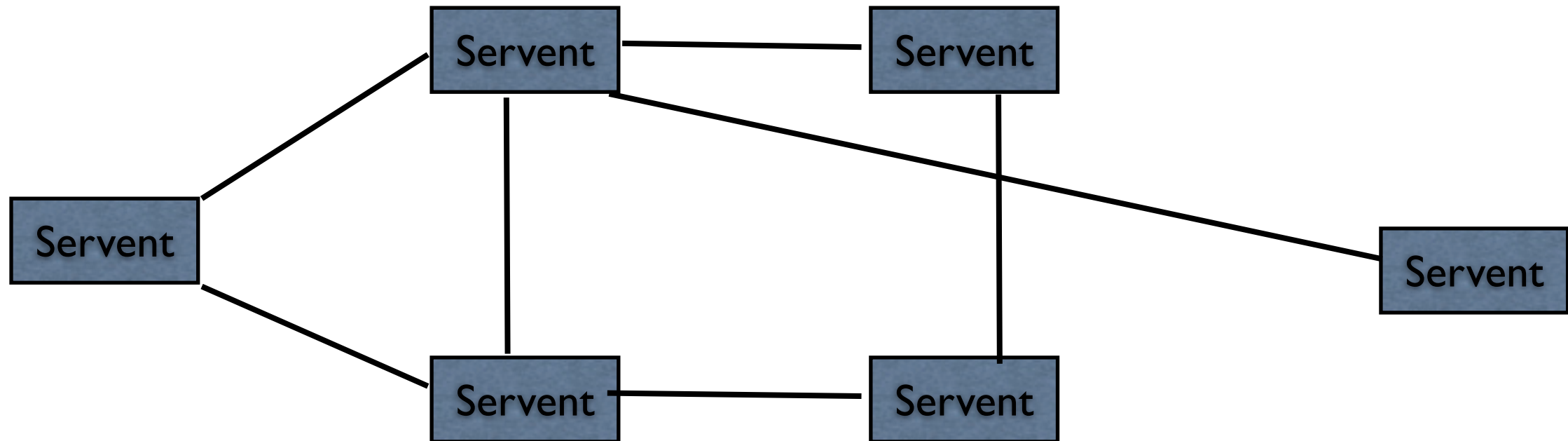
Length of rest of message

Ping 0x00

Header

	Descriptor ID		Payload Descriptor	TTL	Hops	Payload Length	
Byte offset	0	15	16	17	18	19	22

Descriptor 0x00



Pong 0x01

Sent only in response to a ping

Servent can cache pongs of other servents

Payload

	Port		IP Address		Number of files shared		Number of kilobytes shared	
Byte offset	0	1	2	5	6	9	10	13

Port that responding servent can accept incoming connections

IP Address of responding servent

This field uses big-endian format

Query 0x08

Payload

	Minimum Speed		Search Criteria	
Byte offset	0	1	2	...

Minimum Speed

Minimum speed (of connection) in kb/second of servers 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

Payload

	Number of hits	Port		IP Address		Speed		Result Set		Servent Identifier	
Byte offset	0	1	2	3	6	7	10	11	...	n	n+16

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

QueryHit 0x81

Payload

	Number of hits	Port		IP Address		Speed		Result Set		Servent Identifier	
Byte offset	0	1	2	3	6	7	10	11	...	n	n+16

Result Set

	File Index		File Size		File Name	
Byte offset	0	3	4	7	8	...

File Index

A number used by host to identify the file

File Size

Size in bytes of the file

File Name

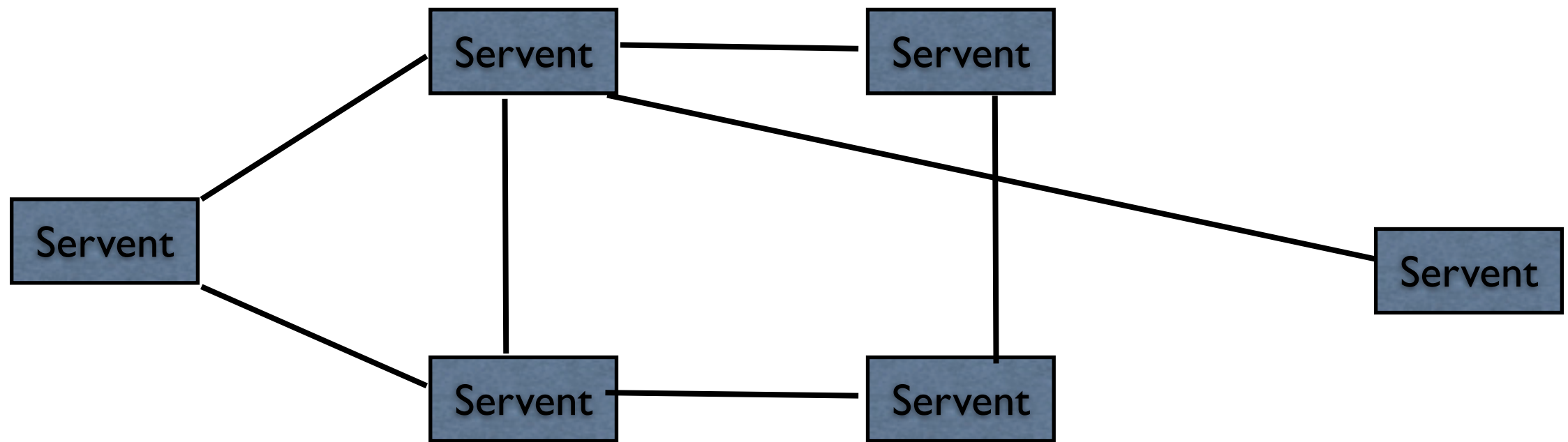
Double-nul (0x0000) terminated name of the file

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”

Query Example



Extended Query Hit

Payload

	Number of hits	Port		IP Address		Speed		Result Set		Trailer		Servent Identifier	
Byte offset	0	1	2	3	6	7	10	11	...	n	m	m+1	m+17

Trailer

	Vender Code		Open Data Size	Open Data	Private data	
Byte offset	0	3	4	5	6	n

How do we know if the trailer exists?

How do we know the length of the private data?

Push 0x40

	Servent Identifier		File Index		IP Address		Port	
Byte offset	0	15	16	19	20	23	24	25

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

Fowarding

Forward all Ping and Querys 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\r\n
Connection: Keep-Alive\r\n
Range: bytes=0-\r\n
User-Agent: Gnutella\r\n 3 \r\n
```

Remote server responses with:

```
HTTP 200 OK\r\n
Server: Gnutella\r\n
Content-type: application/binary\r\n
Content-length: fileSize\r\n
\r\n
```

File Example

