peer-to-peer and agent-based computing

Gnutella

Plan of lecture

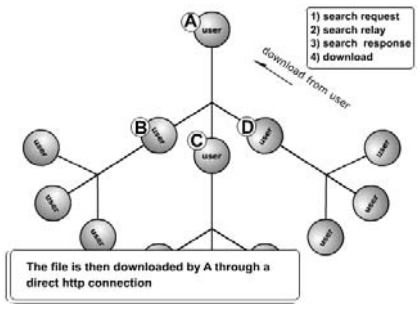
- Gnutella Background
  - Gnutella, the Name…
  - History of Gnutella
  - What is Gnutella?
- Gnutella in operation
  - Gnutella jargon
  - Gnutella descriptor
  - Gnutella scenario (algorithm)
  - Searching Gnutella
  - Joining a Gnutella network
- Gnutella protocol
  - Gnutella descriptors
  - Gnutella descriptor header & payload types
- Gnutella clients

Gnutella

- Infrastructure for file sharing among peers
  - A peer-to-peer information network
- Peers can run in any hardware/software
  - Windows, Linux/Unix, MacOs,…
  - Provided a client tool is available for that…
- Different client tools for peers:
  - Morpheus
  - LimeWire
  - …

Gnutella (Cont'd)

Source: www.limewire.com



Gnutella (Cont'd)

To join in as a peer:
1. Download and install a Gnutella client
2. Launch it – you are now a peer…
3. Search and download files, etc.

- The files you searched and downloaded will become available in your shared folder
- Other peers can see you as a provider for the files you searched (information) and downloaded (actual file)

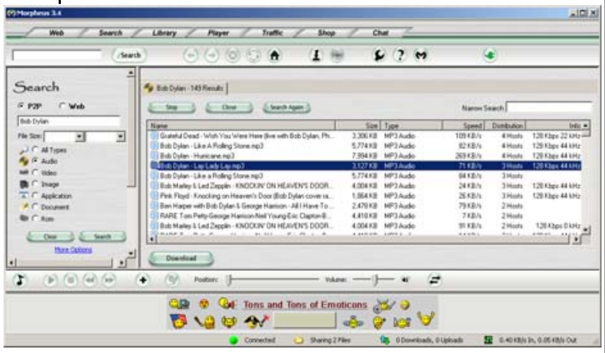
Gnutella (Cont'd)

Morpheus

- The full gamut (not just mp3's)
- Uses metadata (XML) to describe contents of file; easier to find things
- Largely decentralized, speed of query engine rivals that of centralized systems (a la Napster)
- "No more" incomplete downloads:
  - SmartStream: Fail-over system that attempts to locate another peer sharing same requested file, and automatically resume download where it left off at failed host.
- Improved download performance and faster searches (faststream)

Gnutella (Cont'd)

Morpheus



Bob Dylan - 149 Results

Name	Size	Type	Speed	Distribution	Info
Grateful Dead - Wish You Were Here (live with Bob Dylan, Ph...	3,306 KB	MP3 Audio	109 KB/s	4 Hosts	128 Kbps 22 kHz
Bob Dylan - Like A Rolling Stone.mp3	5,774 KB	MP3 Audio	82 KB/s	4 Hosts	128 Kbps 44 kHz
Bob Dylan - Hurricane.mp3	7,994 KB	MP3 Audio	269 KB/s	4 Hosts	128 Kbps 44 kHz
Bob Dylan - Lay Lady Lay.mp3	3,127 KB	MP3 Audio	71 KB/s	3 Hosts	128 Kbps 44 kHz
Bob Dylan - Like a Rolling Stone.mp3	5,774 KB	MP3 Audio	84 KB/s	3 Hosts	
Bob Marley & Led Zepplin - KNOCKIN' ON HEAVEN'S DOOR...	4,004 KB	MP3 Audio	24 KB/s	3 Hosts	128 Kbps 44 kHz
Pink Floyd - Knocking on Heaven's Door (Bob Dylan cover ra...	1,864 KB	MP3 Audio	26 KB/s	3 Hosts	128 Kbps 44 kHz
Ben Harper with Bob Dylan & George Harrison - All I Have To ...	2,478 KB	MP3 Audio	79 KB/s	2 Hosts	
RARE Tom Petty George Harrison Neil Young Eric Clapton B...	4,410 KB	MP3 Audio	7 KB/s	2 Hosts	
Bob Marley & Led Zepplin - KNOCKIN' ON HEAVEN'S DOOR...	4,004 KB	MP3 Audio	91 KB/s	2 Hosts	128 Kbps 0 kHz

Connected Sharing 2 Files 0 Downloads, 0 Uploads 0.40 KB/s In, 0.05 KB/s Out

Gnutella (Cont'd)

- Easy to use, but aimed at human peers
  - Easy to find and download MP3 and videos
  - Bandwidth depends on specific peer
- Not for deploying within a DIS solution
  - so we shan't be saying much about it…
- The organisation of files and peers AND the search mechanism are very efficient and scale up gracefully.

Gnutella, the name

The 'Animal' GNU: Either of two large African antelopes *(Connochaetes gnou* or *C. taurinus)* having a drooping mane and beard, a long tufted tail, and curved horns in both sexes. Also called **wildebeest**.

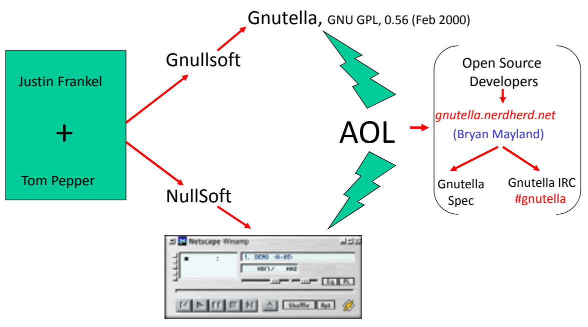
GNU: Recursive Acronym GNU's Not Unix ….

Gnutella =

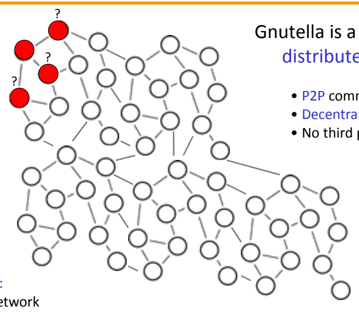
+

Nutella: a hazelnut chocolate spread produced by the Italian confectioner *Ferrero* ….

The history of Gnutella



What is Gnutella?



Gnutella is a protocol for distributed search

- P2P communities
- Decentralised model
- No third party lookup

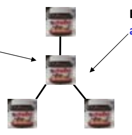
Two stages :

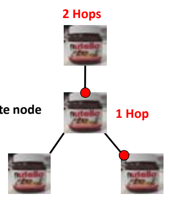
1. Join Network
2. Use Network
    1. Discover other peers
    2. Search other peers

Some jargon

Servent: **A Gnutella node.**

Each servent is both a client and a server

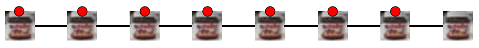




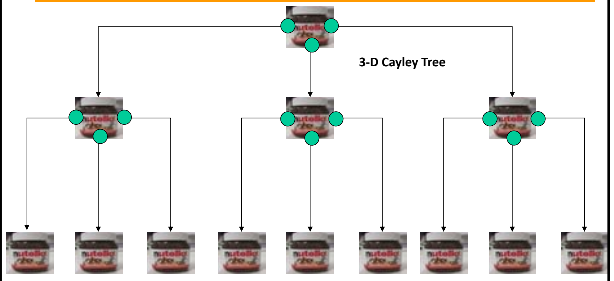
Hops: **a hop is a pass through an intermediate node**

Horizon: **how many hops a packet can go before it dies**

- Default setting is 7 in Gnutella
- Also known as “time-to-live” (TTL)



Search via broadcasting



• Searching in Gnutella involves broadcasting a Query message to all connected peers.
• Each connected peer will relay it to their connected peers (say 3) and so on.
• Typically, this search will run 7 hops.
• If the number of connected peers c=3 and the hops TTL=7 then the total number of peers searched (in a fully connected network) will be:

S = c + c2 +c3 + ….. ch = 3 + 9 + 27 + 81 + 243 + 729 + 2187 = 3279 Nodes

Gnutella descriptors

- Descriptors: messages passed around the network
- 5 types:
  1. Ping: used to actively discover hosts on the network.
     - A servent receiving a Ping descriptor is expected to respond with one or more Pong descriptors.
  2. Pong: the response to a Ping
     - Each Pong packet contains a Globally Unique Identifier (GUID) plus address of servent and information regarding the amount of data it is making available to the network
  3. Query: the primary mechanism for searching the distributed network.
     - A servent receiving a Query descriptor will respond with a QueryHit if a match is found against its local data set.
  4. QueryHit: the response to a Query: contains IP address, GUID and search results
  5. Push: allows downloading from firewalled servents

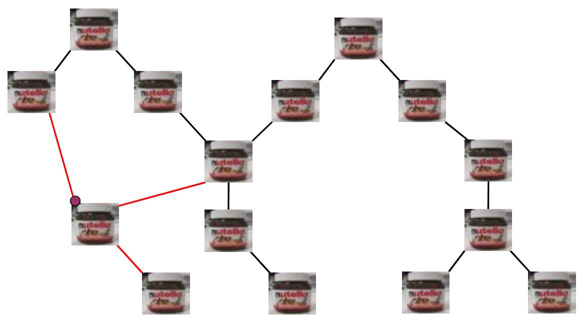
Gnutella scenario

Step 0: Join the network
Step 1: Determining who is on the network
- "Ping" packet is used to announce your presence on the network.
- Other peers respond with a "Pong" packet.
- Also forwards your Ping to other connected peers
- A Pong packet also contains:
  – an IP address
  – port number
  – amount of data that peers are sharing
  – Pong packets come back via same route

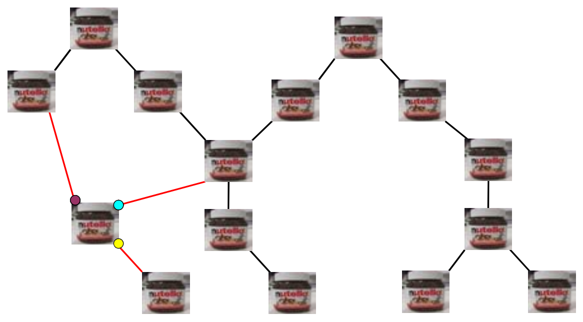
Step 2: Searching
- Gnutella “Query” ask other peers if they have the file you desire (and have an acceptably fast network connection).
  – Example, "Do you have any content that matches the string ‘Homer’ ”?
- Peers check to see if they have matches & respond (if they have any matches) & send packet to connected peers
- Continues for TTL

Step 3: Downloading
- Peers respond with a “QueryHit” (contains contact info)
- File transfers use direct connection using HTTP protocol’s GET method
- When there is a firewall a "Push" packet is used – reroutes via Push path

Searching from one node



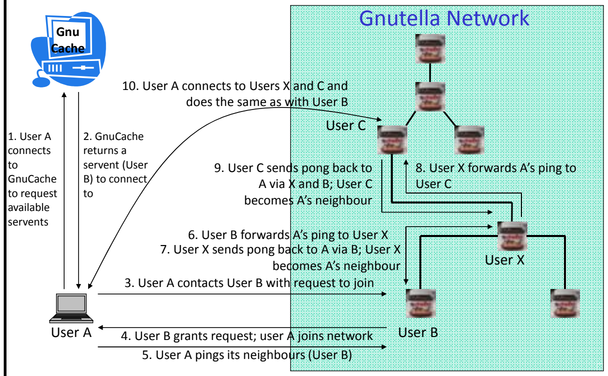
Searching from all nodes



Discovering peers

- Early days – "out of bounds" methods:
  - IRC (Internet Relay Chat) and asked users for hosts to connect to
  - Web pages – users checked a handful of web pages to see what hosts were available
  - Users typed hosts into the Gnutella software until one worked…
- Host Caches:
  - e.g. GnuCache was used to cache Gnutella hosts and was included in Gnut software for Unix
- Dynamically:
  - by watching PING and PONG messages noting the addresses of peers initiating queries

Discovery of peers via host caches



Gnutella Network
Gnu Cache
10. User A connects to Users X and C and does the same as with User B
1. User A connects to GnuCache to request available servents
2. GnuCache returns a servent (User B) to connect to
User C
9. User C sends pong back to A via X and B; User C becomes A's neighbour
8. User X forwards A's ping to User C
6. User B forwards A's ping to User X
7. User X sends pong back to A via B; User X becomes A's neighbour
User X
3. User A contacts User B with request to join
User A
User B
4. User B grants request; user A joins network
5. User A pings its neighbours (User B)

Gnutella descriptors

Descriptor Header	Descriptor Payload
0 … 22	23 … Variable, 0…Max

Types:

- Ping: to actively discover hosts on the network
- Pong: the response to a Ping
  - Includes the GUID address of a connected servent and information regarding the amount of data it is making available to the network)
- Query: search mechanism
- QueryHit: the response to a Query (containing GUID and file info)
- Push: mechanism for firewalled servents

Gnutella descriptor header

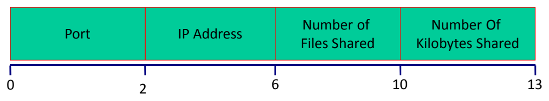
Descriptor ID	Payload Descriptor	TTL	Hops	Payload Length
0 … 16	16 … 17	17 … 18	18 … 19	19 … 22

- Descriptor ID
  - Unique identifier for the descriptor on the network (16-byte string)
- Payload Descriptor
  - Ping: **0x00**; Pong: **0x01**; Push: **0x40**; Query: **0x80**; QueryHit: **0x81**
- TTL: "Time To Live" or "Horizon"
  - Each servent decrements the TTL before passing it on
  - When TTL = 0, message is no longer forwarded
- Hops:
  - Counts the number of hops the descriptor has travelled
  - When TTL expires, the no. of hops is what TTL was at the beginning
- Payload Length:
  - Next descriptor header is located exactly "Payload" length bytes from end descriptor header

Payload 1: Ping descriptor

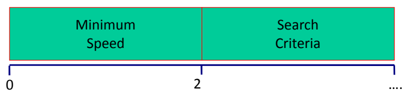
- Ping descriptors:
  - No associated payload, that is, zero length
- Ping represented by Descriptor Header whose:
  - `Payload_Length` field is `0x00000000`
  - `Payload_Descriptor` field is `0x00`

Payload 2: Pong descriptor



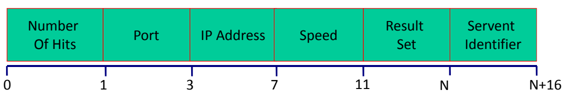
- Port:
  - Port responding host can accept incoming connections
- IP Address:
  - IP address of the responding host (big-endian)
- Number of Files Shared:
  - No. of files responding host is sharing on the network
- Number of Kilobytes Shared:
  - KBs of data responding host is sharing on the network

Payload 3: Query

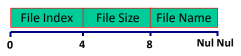


- Minimum speed:
  - Minimum speed (in kb/second) of servents that should respond to this message
  - Servents receiving a Query descriptor with a minimum speed field of *n* kb/s should only respond with a QueryHit if it is able to communicate at a speed ≥ *n* kb/s
- Search Criteria:
  - A search string ending on null (i.e. **0x00**)
  - Maximum length bound by Payload_Length field of the descriptor header
  - E.g. “myFavouriteSong.mp3”

Payload 4: QueryHit



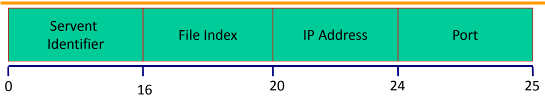
- **Number of Hits** in the result set
- **Port** which the responding host can accept incoming connections
- **IP Address** of the responding host (big-endian)
- **Speed** (kb/second) of the responding host
- **Result Set** of Number_of_Hits responses to corresponding Query with the following



- **File Index:** ID of file matching the corresponding query - assigned by the responding host
- **File Size:** size (bytes) of this file
- **File Name:** name of the file (double-nul (i.e. 0x0000) terminated)

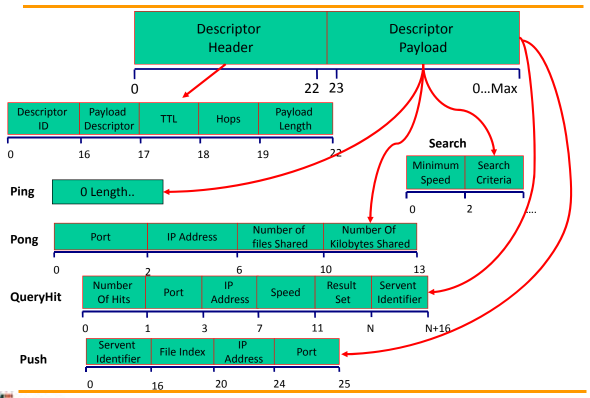
- **Servent Identifier:** servent network ID (16-byte string), typically function of servent's network address; instrumental in the operation of the **Push Descriptor**

Payload 5: Push



- Servent Identifier:
  - Target servent network ID (16-byte string) requested to push file (with given index File_Index)
- File Index:
  - ID of the file to be pushed from the target servent
- IP Address:
  - IP address of target host which file should be pushed (big-endian format)
- Port:
  - Port on target host which file should be pushed

Header and Payloads



Some Gnutella “clients”

Windows	Linux/Unix	MacIntosh
BearShare Gnucleus Morpheus Shareaza Swapper XoloX LimeWire Phex	Gnewtellium Gtk-Gnutella Mutella Qtella LimeWire Phex	LimeWire Phex

Some Gnutella “clients” (Cont’d)

BearShare (http://www.bearshare.com) (July 23, 2001)
- “… an exciting new Windows file sharing program from Free Peers, Inc. that lets you, your friends, and everyone in the world share files! Built on Gnutella technology, BearShare provides a simple, easy to use interface combined with a powerful connection and search engine that puts thousands of different files in easy reach!” [BLOCKED in ABDN]

Gnotella (http://www.gnotella.com) (July 23, 2001)
- “… clone of Gnutella, a distributed real time search and file sharing program. Gnotella is for the Win32 environment, and offers extra benefits such as multiple searches, improved filtering/spam protection, bandwidth monitoring, enhanced statistics, upload throttling, and skinning, as well as more.” [BLOCKED in ABDN]

Gnucleus (http://gnucleus.sourceforge.net/) (July 23, 2001)
- “An open Gnutella client for an open network. Made for windows utilizing MFC (works in WINE). Constantly evolving, easy enough for the first time user and advanced enough to satisfy the experts.”

LimeWire (http://www.limewire.com) (July 23, 2001) – Java
- “… a multi-platform Gnutella client with nice features like auto-connect, browse host, multiple search, upload throttling, connection quality control, library management and sophisticated filtering. It is built for the both the novice and power user”

Some Gnutella “clients” (Cont’d)

Phex (http://www.phex.org) – Java
- “Phex is entirely based on William W. Wong’s Furi. As Furi has not been updated for over one year I decided to continue it´s development. But in case Wong is currently working on a new version of Furi i decided to rename my branch of the client to Phex. FURI is a Gnutella protocol-compatible Java program that can participate in the Gnutella network. It is a full version program with a easy to use GUI interface that can perform most of the tasks of a Gnutella servant.”

Toadnode (http://www.toadnode.com)
- Toadnode described itself as “an extensible platform for peer-to-peer (P2P) networks. Its core functionality revolves around the ability to find, retrieve and distribute data between users across multiple networks. Toadnode pairs this ability to search, with an application layer to accommodate plug-ins that fully exploits and leverages the data that is distributed.”

Gnutelliem (http://newtella.com/linux)
- “Gnewtellium is the Linux/Unix port of Newtella. Newtella is the new way to share music over the internet. It combines a focus on music, like Napster, with a decentralized network of users, and is based on the gnutella protocol. The software is designed to retrieve and exchange only MP3 files. As such, it prevents the unrestricted duplication of viruses and self-executing trojan horses. It also prevents illicit uses (such as child pornography) of the gnutella network.” [NO LONGER ACTIVE]

Some Gnutella “clients” (Cont’d)

Gnut (http://www.gnutelliums.com/linux_unix/gnut/)
- “Gnut is a command-line client which implements the gnutella protocol. It supports all features available in the original Nullsoft client, as well as many others. Bandwidth limiting, sorting of results, regular expression searching, are among the list. It will compile and run on a wide range of POSIX compliant (and not so compliant) systems including: SunOS, Linux, FreeBSD, HP-UX, and Win32.” [NO LONGER ACTIVE]

Hagelslag (http://tiefighter.et.tudelft.nl/hagelslag)
- “Hagelslag is an implementation of GNUtella. The main goals for this implementation are flexibility, stability and performance. The development of Hagelslag was primarily aimed at i386 machines running Linux, as of version 0.8, FreeBSD is supported as well.” [NO LONGER ACTIVE]

Qtella (http://www.qtella.net/)
- “Qtella is a new Gnutella client for Linux written in C++ using the Qt libraries. It should be no problem to use Qtella on any platforms where Qt with thread support (library qt-mt must exists) is installed.”

Mactella (http://www.cxc.com/)
- “Mactella is the Mac version of Gnutella, an open-source file-sharing network that allows you to exchange an assortment of file formats with other users. It can operate on any port and has no centralized server. This program is capable of transferring any type of file that users put online, including ZIP, MPEG, ASF, MOV, QT, HQX, EXE, JAR, and SIT.” [NO LONGER ACTIVE]

Summary

- Gnutella Background
  - The name, its history, what is it?
- Gnutella in Operation
  - Organising a Gnutella Network
  - Searching Gnutella for peers and files
  - Peers are discovered by IRC, GnuCache and message monitoring
- Gnutella Protocol
  - Gnutella Descriptors consists of a header and a payload
  - 5 types of payload: Ping, Pong, Query, QueryHit, Push
- Things to Know
  - Gnutella scenario – joining, discovering and searching Gnutella networks
  - Difference between Gnutella and Napster

Reading list

- Chapter of textbook
- Wikipedia entry:
  http://en.wikipedia.org/wiki/Gnutella
- Gnutella for Users (GnuFU):
  http://gnufu.net
- O’Reilly
  http://www.oreillynet.com/topics/p2p/gnutella