



Tulicreme

A decentralized timeline service

T2G16

David Ferreira - up202102686

Eduardo Correia - up201806433

João Cardoso - up201806531

Ricardo Fontão - up201806317

Context

- The goal of this project was to build a decentralized timeline service
- Peer-to-peer communication systems were designed to retrieve (infrequently modified) files. In the case of a social network, timelines are frequently updated
- Inspired by “Social Networks in Peer-to-Peer Systems”



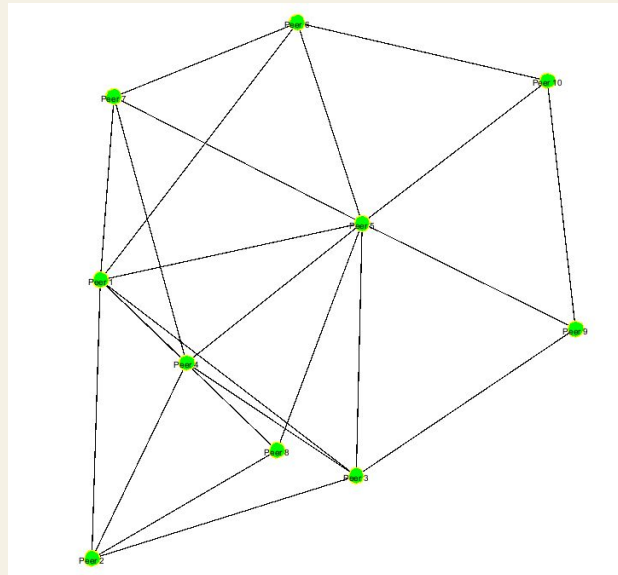
Used Technologies

- Kotlin
- Java Sockets (for message exchanging)
- Swing/FlatLaf (for the GUI)



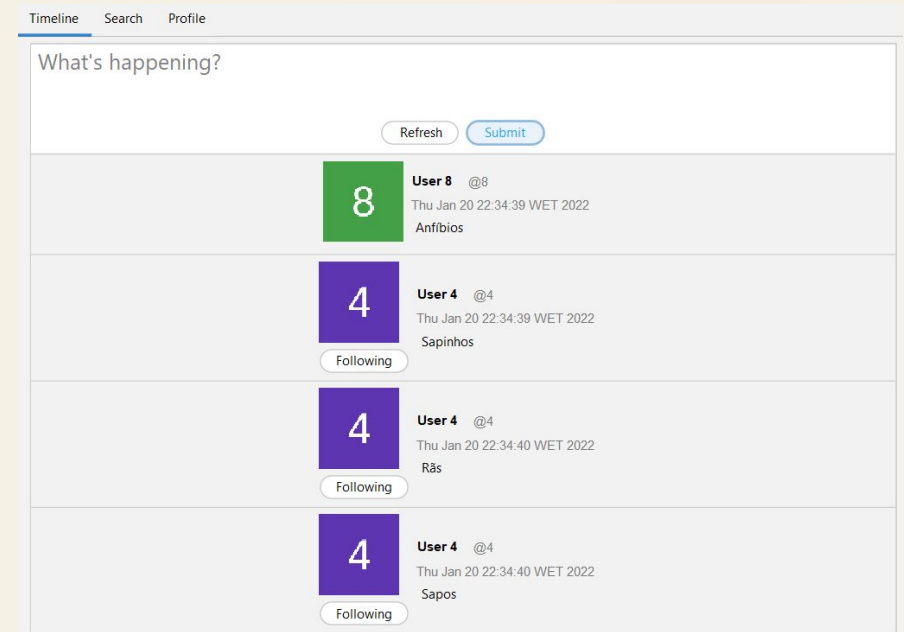
Protocol

- Our protocol is inspired by *Gnutella* with some modifications targeting social features
- Gnutella is a peer-to-peer network protocol that allows users (peers) to exchange files between each other
- Peers communicate by flooding messages through the network



Features

- Follow/unfollow users
 - Peers occasionally query for their followers to get new content
- Post content
- Check timeline
- Content discovery
 - Can search for specific users' posts or by words in those posts



Messages

- **Ping/Pong** messages for finding new neighbours and checking if the current ones are still alive
- **Query/QueryHit** messages for knowing who has certain posts
- **Discover** messages for content discovery
- **Get/Send** messages for requesting/sending posts
- **RequestConnect/ConnectTo** messages to communicate with the host cache and request possible neighbours
- **AddNeighbour/RemoveNeighbour** are used to connect to new neighbours

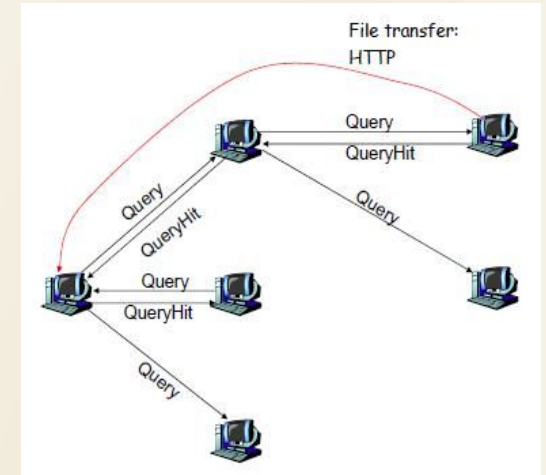


Image source:
Wikipedia



Friends

- How they are created:
 - When a peer receives a **QueryHit** message from another peer from a certain topic, it adds that peer as a friend and increments that friend's topic count
 - More messages received from a friend on a topic → their friendship is stronger
 - A relationship's strength is given by the following formula, where R is the number of messages received from friend A and topic X and Q is the total number of messages from A :

$$S_{AX} = \frac{R_{AX}}{Q_A}$$



Friends

- How they are used:
 - In a regular *Gnutella* implementation, whenever a query message is sent, it is propagated to every neighbour (flooding)
 - With the friend system, prioritization is given to sending queries to n friends
 - If the peer has a friend with relation > 0 , the peer sends queries to the n best friends
 - But if there aren't enough friends, it also sends to random neighbours



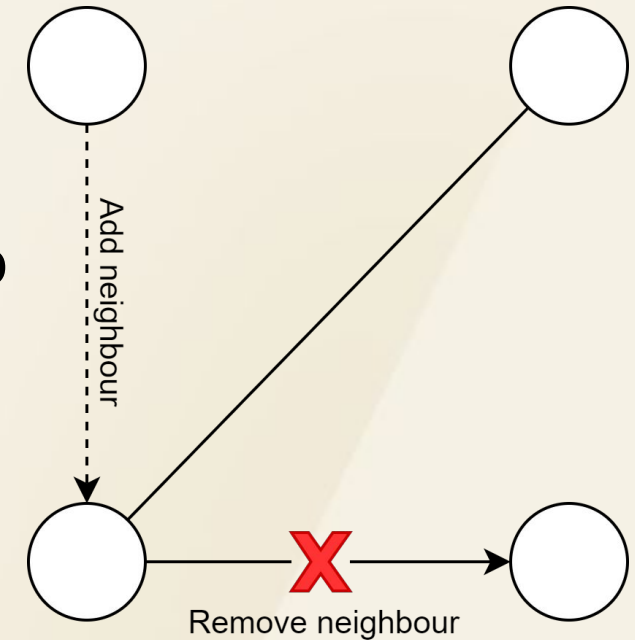
Host cache

- Single server
- Every peer connecting must request for peers already on the network
- Single point of failure
- Periodically pings a random set of peers and updates internal state accordingly



Fault tolerance

- Whenever a peer sends a message to enter the network to another peer, if that peer is full, it forcibly removes another peer and notifies them of such
 - This prevents a situation where every peer is connected to a maximum number of peers and new peers cannot enter
- If a peer ends up without neighbours for any reason, it will try to rejoin the network via the Host Cache



Fault tolerance

- Each peer has a minimum acceptable and maximum amount of neighbours
 - Whenever it reaches below the former, it will send connection requests until it reaches the latter
 - Otherwise, it only sends pings to check if the neighbours are alive (time to live = 1)
 - This minimizes the number of forcible removals of peers
- Peer state is saved to disk
 - A file is created for each user



Content discovery

- Using Bloom Filters
 - Each node keeps a bloom filter of all the words in its posts
 - Simple content search through keywords
 - **Advantages:**
 - Space efficient
 - Fast element search
 - **Disadvantage:**
 - Error probability of false positives
 - Not critical to the nature of the use



Future work

- Add *ActivityPub* API protocol on top of our system
- Explore authentication mechanisms
- Implement some kind of super-peer approach
- Posts with media content that isn't just text
- User profile



References

- Upadrashta, Y., Vassilev, J., & Grassmann, W. (2005). Social Networks in Peer-to-Peer Systems. Proceedings of the 38th Annual Hawaii International Conference on System Sciences.
- Gnutella 0.6 RFC Draft
<http://rfc-gnutella.sourceforge.net/src/rfc-0_6-draft.html>



Thank you

Any questions?

