Software Specification „Folder Sync“

## 3-Layer-Architecture

receive

send

with neighbor list

* send messages
* receive messages
* file transfer
* receive ping and react with pong for peer exploration
* {build-up connection}
* {terminate connection}
* refresh neighbor list (delete neighbors with expired timeouts)
* analyze messages from application and outreach them to the network
* periodically sending file list to neighbors
* sending request for file x from neighbor y
* sending file x to neighbor y

n2o

Queue

o2a

a2o

o2n

# Message specification:

Ping

incoming ping (n2o) := ("ping", pingID, ttl, hops, senderUsername, senderIP, senderPort)

outgoing ping (o2n) := ("ping", pingID, ttl, hops, ownUsername, ownIP, ownPort, targetIP, targetPort)

Pong

incoming pong (n2o) := ("pong", id, [(username1, ip1), (username2, ip2), ...])

outgoing pong (o2n) := ("pong", id, [(username1, ipP1), (username2, ip2), ...], targetIP, targetPort)

RefFL (refresh filelist)

incoming refFL (n2o) := ("refFL", fileList, senderUsername, senderIP, senderPort)

incoming refFL (o2a) := ("refFL", fileList, senderUsername, urgentFlag)

outgoing refFL (a2o) := ("refFL", fileList)

outgoing refFL (o2n) := ("refFL", fileList, ownUsername, ownIP, ownPort, targetIP, targetPort)

reqFile (request file)

incoming reqFile (n2o) := ("reqFile", fileName, fileHash, senderIP, senderPortUDP , senderPortTCP)

incoming reqFile (o2a) := ("reqFile", fileName, fileHash, senderUsername, senderPortTCP)

outgoing reqFile (a2o) := ("reqFile", fileName, fileHash, targetUsername)

outgoing reqFile (o2n) := ("reqFile", fileName, fileHash, ownIP, ownPort, targetIP, targetPort)

sendFile (permission to network layer to send the file)

downgoing sendFile (a2o) := ("sendFile ", filePath, targetUsername, targetPortTCP)

downgoing sendFile (o2n) := ("sendFile ", filePath, targetIP, targetPortUDP, targetPortTCP)