Methodology specification

The project consists of two separately treated parts. Part one is the development of a Distributed Hash Table (DHT) and part two the development of a Reliable User Datagramm Protocol (RUDP).

Development in this case, includes the creation of a standalone version that will be tested on its own. With the standalone version of the DHT implementation, the measurement of accurate results will be possible. The RUDP implementation will be compared to the Transmission Control Protocol (TCP). The comparisons will show advantages as well as disadvantages of the implementations performances.

After the implementation and testing is done, each part will be integrated into the existing MediaSense framework.

The Eclipse IDE will be used as development environment and all project related content will be shared over a Git repository.

DHT

The DHT will be a distributed lookup service for the MediaSense framework and fulfill the following requirements:

- work in a cycle structure (chord[1])
- perform a lookup in O(log(n)) steps (n = number of nodes)
- automatically recover from a leaving node
- automatically detect and recover from a disappearing node
- automatically integrate a new node

Measurement

To measure the functionality, reliability, and performance, a management software handling a configurable amount of DHT nodes and passing messages among them will be created. Using this way, the network layer is not used as it does not play a role in the functionality of the DHT, because it is assumed that the network performs absolute reliable and without delay.

The management software will have the following functions:

- connect DHT nodes to each other
- transmit messages between DHT nodes
- log each transmitted DHT message with the following data for later analysis:
    - Sending node
    - Receiving node
    - Message transmission time
    - Message type
    - DHT Node information (state, finger table entries, …)
- Simulate entering, leaving and disappearing nodes

The analysis based on the logged information of the management software will be the following:

- number of transmitted keep alive messages

[1] http://en.wikipedia.org/wiki/Chord_%28peer-to-peer%29

- number of messages / transition time after a node entered the DHT
- number of messages / transition time after a node left the DHT
- number of messages / transition time after a node disappeared from the DHT
- number of messages to perform the lookup of a node
- compare all result to the theoretical amount of needed messages (if possible)

RUDP

RUDP should be the implementation of a reliable packet-based data transmission protocol that has the following features:

- detection of bit errors in transmitted packages and automatic resending
- detection of lost packets and automatic resending
- avoiding of any connection establishment and connection shutdown
- packet oriented and not connection oriented protocol
- avoid reception of packet duplicates
- maintain the order of transmitted packets
- (detection of a destination that is / became unavailable) ?

Measurements

To ensure that the aforementioned features are implemented correctly and to perform performance measurements a helper tool should be created that measures the throughput during data transmission. The helper tool is going to be a piece of software that acts as bridge for data transmission between two clients exchanging data with each other.

To test the error recover capabilities the transmission bridge should also be able to damage, drop, and delay passing packets in a controllable way to simulate unreliable connections. Statistics should be generated for comparison with other protocols.

In detail the measurements should be the following:

Description	**Parameters for transmission bridge**	**Result**
Is data transmission reliable using RUDP?	packet delay: n ms packet loss: i % packet reordering: j% packet bit errors: k%	success: yes / no
Transmission time of data with fixed size using RUDP	packet delay: n ms packet loss: i % packet reordering: j% packet bit errors: k%yes /no	transmission time in ms
Transmission time of data with fixed size using TCP	packet delay: n ms packet loss: i % packet reordering: j% packet bit errors: k%yes /no	transmission time in ms
Average round trip time for a RUDP packet	packet delay: n ms packet loss: i % packet reordering: j% packet bit errors: k%yes /no	RTT in ms
Average round trip time for a TCP packet	packet delay: n ms packet loss: i % packet reordering: j% packet bit errors: k%yes /no	RTT in ms

Time plan:

Week 1	Discovering of DHT and RUDP
Week 2	Develop the method
Week 3-5	Implementation and testing of a standalone DHT
Week 5-6	Integration of the DHT to the MediaSense platform
Week 7-8	Implementation and testing of a standalone RUDP
Week 9	Integration of the RUDP to the MediaSense platform
Week 10	Finish the report / present the result