

# Project Plan for Degree Projects

## Department of Computer Science

### General Information

Title:	Reel-time communication using a peer-to-peer network in the web
External company:	Name of the company (if you do your degree project at an external company)

### Persons involved

Student 1:	Henry Pap	hp222fq@student.lnu.se
Student 2:		

Supervisor:	Mauro Caporuscio <i>mauro.caporuscio@lnu.se</i>
External supervisor:	Your supervisor at the company (if you do your degree project at an external company)

## Background

Real-time communication (RTC) [1] is a term describing live communication between two or more mediums with no or very low response time. In the context of networking this means distributed clients (geographically) can communicate with each other on a low latency. Today we either use client-server or peer-to-peer architectures that both come with their own pros and cons. Peer-to-peer (P2P) systems in the accepted definition by Theotokis et al. [2] are pure and loose P2P systems, where pure refers to total distribution meaning no use of central servers and loose being something in between of pure P2P and client-server without relying too much on centralized servers. Web technology offers three main methods for communication (native to the web-browsers):

1. HTTP
2. WebSockets
3. webRTC

But only WebSockets and webRTC is considered fast enough for RTC, WebSockets uses what is called a push and receive service meaning that a server also can send messages to clients. But only the "newly" API webRTC supports peer-to-peer communication as it resolves finding the direct communication-link between two peers. Using the web browser comes with drawbacks, it can not handle too much peers and thus the recommended number of connected peers to one client is 6. To overcome this issue a specific P2P topology is required, by smart routing and use of super-nodes (client that acts like servers) a virtual network will then be formed.

## Problem formulation

Peer-to-peer RTC networks is a challenging problem on itself that has been solved multiple times, but doing this entirely in the web is not very common. This thesis will focus on multiplayer game networking using the native browser-built in webRTC API. The webRTC API has limits of connections since it is running directly in the browser. The focus for this thesis will thus be to extend the small limit webRTC has while still keeping a low latency as this is an very important aspect of real-time multiplayer games. The peer-to-peer network must also consider synchronous playback that is everyone in the network should get the same feed of data at an considerable same time (overall same latency over the whole network). Dropout of nodes is another important issue that peer-to-peer brings as the dropout peer could be a potential gateway for other peers in the network. To sum the problem formulation up, in order for the webRTC API to be a potential solution for online real-time communication for games an topology must be carefully picked and finalized.

and extend the webRTC peer-connection limit by using a peer-to-peer topology in order to achieve scalability. Real time multiplayer game bring the problem of having low latency since it would be pointless without it thus the topology must also cover this part.

## Motivation

As mentioned in the problem formulation, peer-to-peer real time communication using the web is not something that has been solved on a higher level (i.e. having scalability and low latency). From a economical view point this helps the game creator to not spend money on a expensive server that handles multiple connections. From a scientific point of view, as more and more application is integrated into the web browsers due to its broad availability to almost all platforms we need better ways for real-time communication using only the web browser as our toolbox as this solution can easily be extended to not only real-time multiplayer games.

## Objectives

<b>01</b>	Selecting some peer-to-peer network topologies that aims for low latency and scalability.
<b>02</b>	Defining requirements in order to have something to compare against (low latency, high scalability etc).
<b>03</b>	Compare the selected peer-to-peer network topology in order to lower the amount of topologies.
<b>04</b>	Experimenting with the selected topologies and their corresponding settings to lower the list even more.
<b>05</b>	Setting up a test/simulation in order to test webRTC topology.
<b>06</b>	Implement and experiment using the test and webRTC to pick the final topology.
<b>07</b>	Finalize & optimize the final implementation.

## Method

To select the different network topologies a systematic literature review (SLR) must first take place to gather the background knowledge. Same applies for background knowledge of the webRTC API. Selecting the topologies requires a comparative method between the many topologies as a result of the SLR in order to lower the list.

Running experiments on the topologies with their own settings results of a small topology list that will be implemented using webRTC techniques. Further experiments and implementation will then be conducted thus resulting in a final implementation using a specific optimized topology. The final implementation will be verified and validated to make sure it runs and meet the defined requirements.

## Time plan

Date	Milestone
2019-02-15	Systematic Literature Review for background knowledge (SLR) + selecting potential topologies
2019-02-19	Degree project plan finished
2019-03-08	Reading and gather the information generated from the SLR
2019-03-17	Writing the peer-to-peer background part based on the selected topologies
2019-03-20	Exploring the webRTC API by both reading the specs and testing with simple implementations
2019-03-22	Finishing the background with webRTC
2019-03-23	Defining the requirements for the topology
2019-03-25	Writing the motivation and aim (re-using this document with some retouching)
2019-03-27	Starting with the (method?) and result
2019-04-05	Experiment with the different topologies to achieve low latency and scalability
2019-04-15	Implement the most suitable topology into web environment using the webRTC API
2019-04-20	Verifying and validating the implementation
2019-04-22	Discussing the results based on the evaluation
2019-05-01	Finalizing the thesis

## References

- [1] What is real-time communications (rtc)? - definition from techopedia. Accessed 19 Feb. 2019. [Online]. Available: <https://www.techopedia.com/definition/24426/real-time-communications-rtc>
- [2] S. Androutsellis-Theotokis and D. Spinellis, “A survey of peer-to-peer content distribution technologies,” *ACM Comput. Surv.*, vol. 36, no. 4, pp. 335–371, Dec. 2004. [Online]. Available: <http://doi.acm.org/10.1145/1041680.1041681>