

HYPER-LINKED COMMUNICATIONS

WebRTC enabled asynchronous collaboration

Wednesday 25th May, 2016

Henrique Rocha

Instituto Superior Técnico

Universidade de Lisboa

henrique.rocha@tecnico.ulisboa.pt

Advisor: Ricardo Pereira

Co-Advisor: Paulo Chainho



OVERVIEW

1. Introduction
2. Related Work
3. Architecture
4. Implementation
5. Evaluation
6. Conclusions

INTRODUCTION

INTRODUCTION

1. Introduction

1.1 Context

1.2 Problem Statement

1.3 Thesis Goals

2. Related Work

3. Architecture

4. Implementation

5. Evaluation

6. Conclusions

Written communication could never replace face to face communication.

“No computer in our lifetimes will ever rival a human voice’s capacity to conveying rich and complex social and emotional meaning”

— Geddes, Martin

Today, we can achieve more.

PROBLEM STATEMENT

Real-time communication applications can make a difference on business, education and health sectors.

An application that provides a collaborative environment and a way to remember our past communications would be a strong tool.

PROBLEM STATEMENT: USE CASE



THESIS GOALS

Allow multi party conference calls.

Record and playback interactive video.

Create a collaborative environment

Use only standard technologies like JavaScript, WebRTC, HTML5 and CSS3.

RELATED WORK

RELATED WORK

1. Introduction

2. Related Work

2.1 Early days of the Internet

2.2 Real-Time communications

2.3 Hypermedia

2.4 Collaboration & Time manipulation

3. Architecture

4. Implementation

5. Evaluation

6. Conclusions

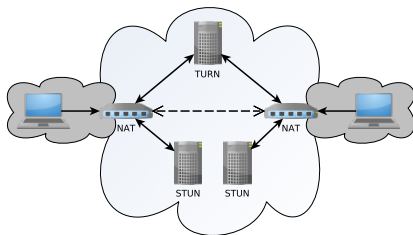
EARLY DAYS OF THE INTERNET

- IPv4 Address Exhaustion

- Network Address Translation

- Client-Server model

- STUN + TURN = ICE



WebRTC (Web Real-Time Communications)

- Access to camera, microphone and screen*
- Peer to Peer file and stream sharing
- Standardized protocols
- No plug-ins required



* requires installing a plug-in yet.

REAL-TIME COMMUNICATIONS



Skype

Audio/Video/Text

File Sharing

Proprietary



Hangouts

Audio/Video/Text

Collaborative Tools

WebRTC*



Jitsi

Audio/Video/Text

Collaborative Tools

WebRTC



Kurento

Audio/Video

Stream Recording

WebRTC

* requires installing a plug-in on non chrome web browsers.

HYPERMEDIA: MORE THAN WORDS, MORE THAN IMAGES

- **Concepts:** HyperText & HyperMedia & HyperCommunications
- **Implementations:** HyperCafe & HyperHitchcock



EXTENDING COLLABORATION TOOLS WITH TIME MANIPULATION

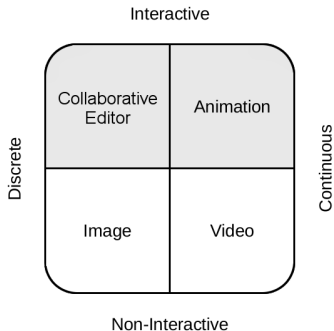


Figure: Media Types

EXTENDING COLLABORATION TOOLS WITH TIME MANIPULATION

Table: Comparison between Operational Transformation libraries

Library	Own Server	Own Storage	Operations
ShareJS	✓	✓	text+objects
TogetherJS	✓	✗	text+objects
Goodow	✓	✓	text+objects
Etherpad Lite	✓	✓	extendable
OT.js	✗	✗	text

ARCHITECTURE

RELATED WORK

1. Introduction

2. Related Work

3. Architecture

3.1 Modules

3.2 Implementation Proposal

4. Implementation

5. Evaluation

6. Conclusions

MODULES

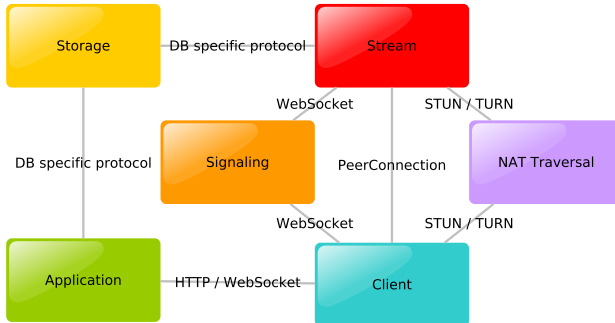


Figure: System Modules

SYSTEM INFRASTRUCTURE

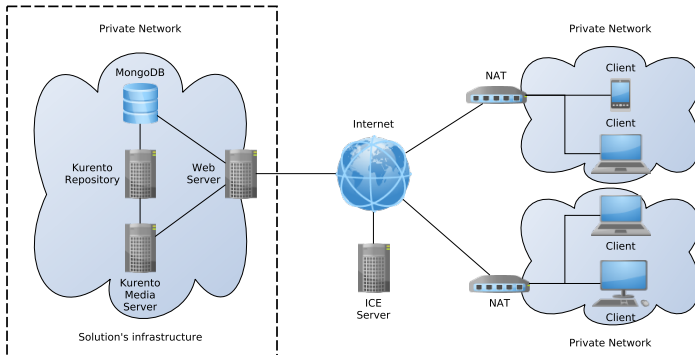


Figure: System Infrastructure

- **Signaling Server & Web Server:** Play Framework
- **Stream Server:** Kurento Media Server
- **Database:** MongoDB

Table: Application Architecture

Application						
jQuery	HTML5	CSS3 (Bootstrap)	Signaling	ot.js		adapter.js
HTTP	User Interface		WebSocket		WebRTC	

IMPLEMENTATION

RELATED WORK

1. Introduction

2. Related Work

3. Architecture

4. Implementation

4.1 Signaling Protocol

4.2 Stream Recording

4.3 Hyper-Content

4.4 Time manipulation

4.5 Stream composition

4.6 Chat & Collaborative environment

5. Evaluation

6. Conclusions

SIGNALING PROTOCOL

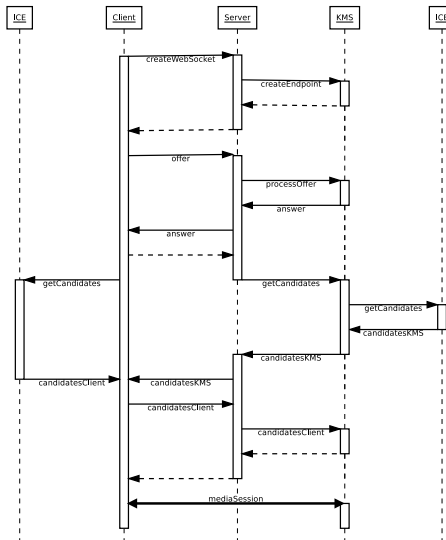
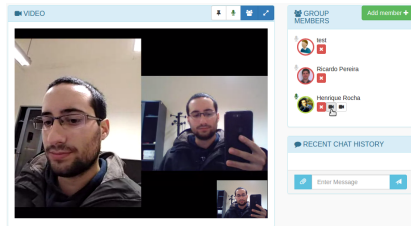
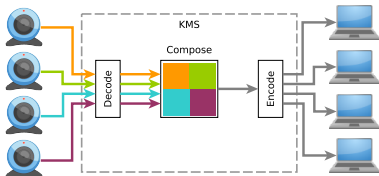


Figure: Signaling Protocol

- Client-side recording.
- Server-side recording to file system.
- Server-side recording to database (Kurento Repository).



STREAM COMPOSITION




HYPER-CONTENT

- Create & Search content
- Scheduler
- QR codes
- Security concerns




CONTENT EDITOR

Begin: (timeline )
29/03/2016 04:56:40 





End: (timeline )
29/03/2016 04:56:41

Content:
Hello World!

Is Caption: ☒

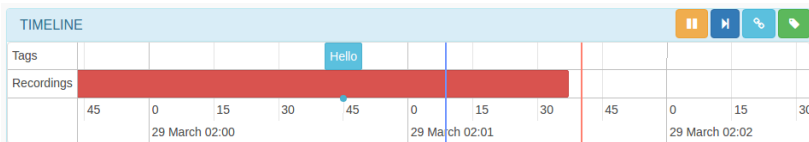
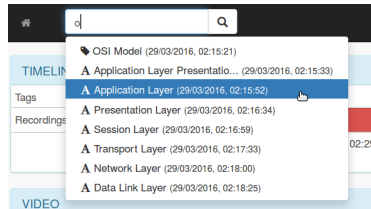
Select Time

March 2016						
Su	Mo	Tu	We	Th	Fr	Sa
28	29	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31	1	2
3	4	5	6	7	8	9

TIME MANIPULATION

- Playback recordings
- Create & Search annotations
- Time Hyper-links



- Instant text messaging
 - WebSockets
- File sharing
 - HTTP file upload
 - stored in the database
- Collaborative text editor (OT.js)
 - retain
 - insert
 - delete

EVALUATION

1. Introduction
2. Related Work
3. Architecture
4. Implementation
5. Evaluation
 - 5.1 Performance Tests
 - 5.2 User Interface Tests
6. Conclusions

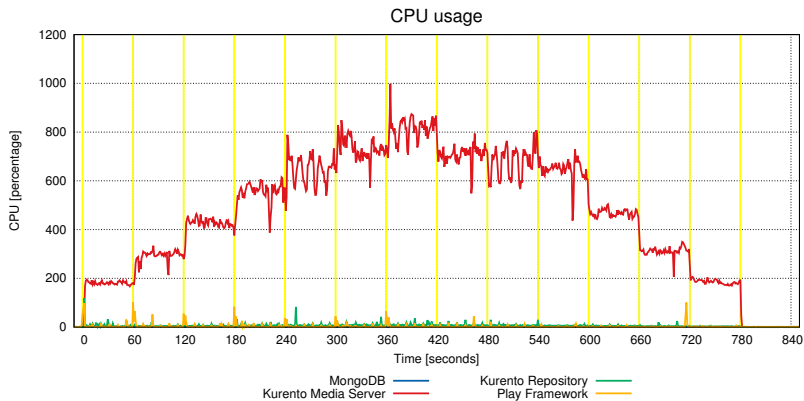


Figure: CPU usage at server

CPU (AVERAGE)

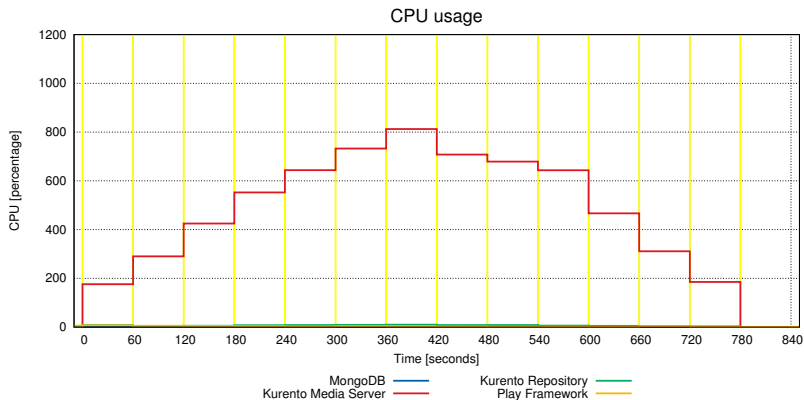


Figure: CPU usage at server (average per interval)

MEMORY

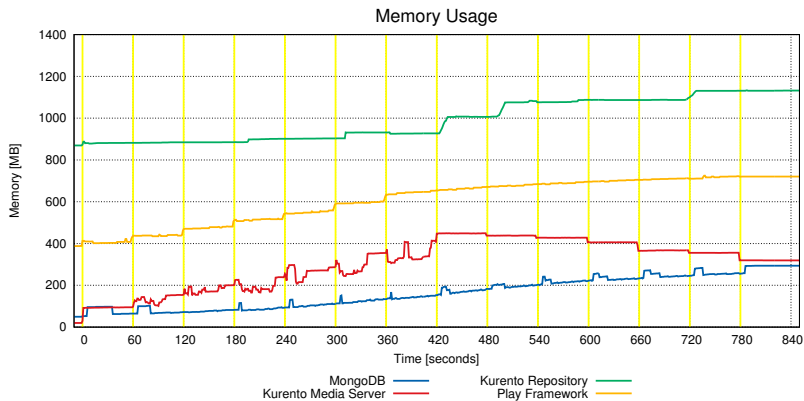


Figure: Memory usage at server

NETWORK USAGE

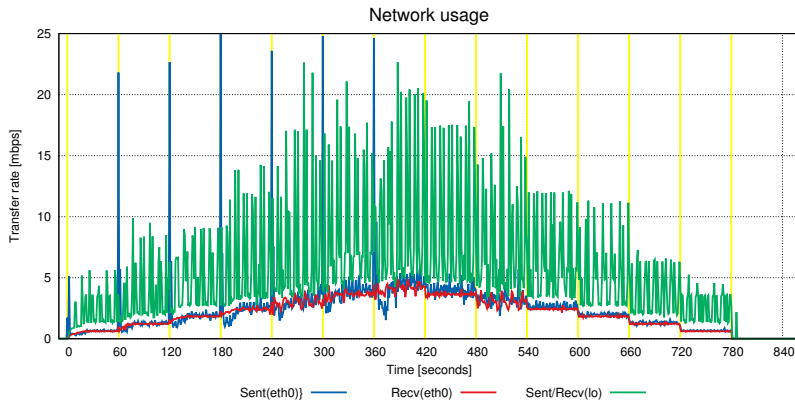


Figure: Network usage at server

NETWORK USAGE (AVERAGE)

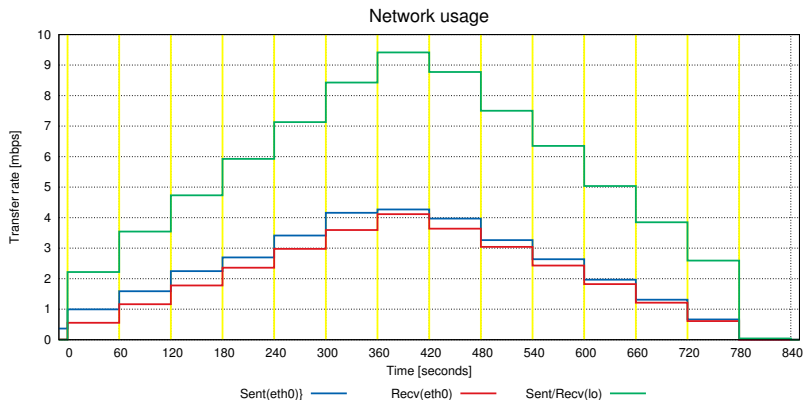


Figure: Network usage at server (average per interval)

NETWORK USAGE

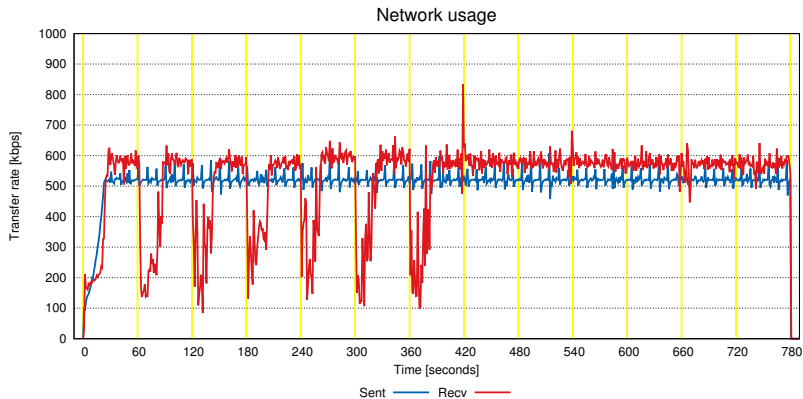


Figure: Network usage at client

USER INTERFACE TESTS

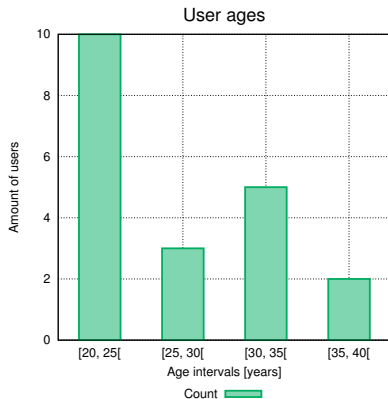


Figure: Users age

FIVE TASKS

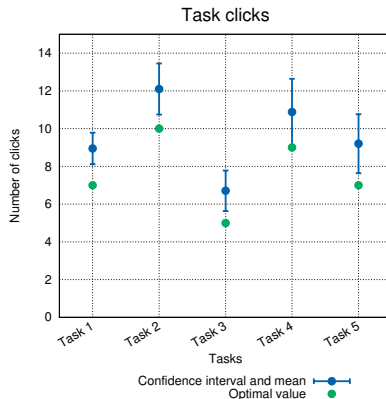
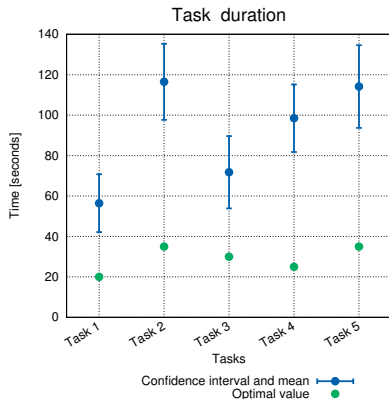


Figure: Tasks metrics

- Difficulty per task.
- Errors per task.

OVERALL EVALUATION

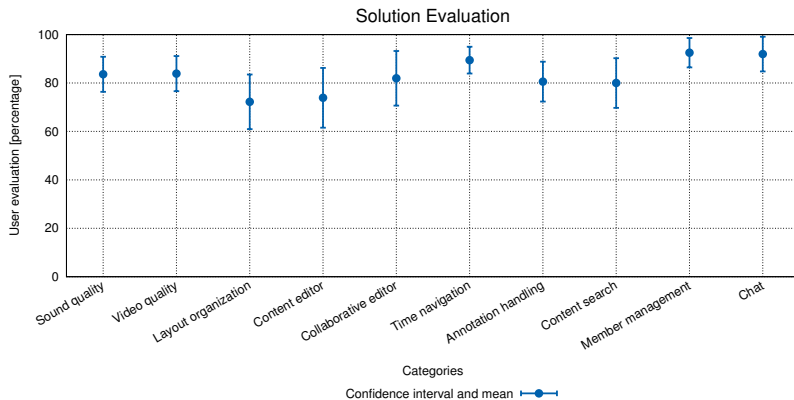


Figure: Overall evaluation

CONCLUSIONS

RELATED WORK

1. Introduction
2. Related Work
3. Architecture
4. Implementation
5. Evaluation
6. Conclusions

CONCLUSIONS

- New usage scenarios for communication and collaboration applications.
- Enrich communications using hypermedia concepts. Record, playback and collaboration features.
- Prototype implementation and testing.

Questions?