

**SEAMLESS TRANSFERRING OF “WORK IN
PROGRESS” BETWEEN DEVICES IN
A PEER-TO-PEER ENVIRONMENT**

Dahanayaka Liyanage Tharika Chathumadi

(189310v)

Degree of Master of Science in Computer Science

Department of Computer Science and Engineering

University of Moratuwa

Sri Lanka

June 2021

**SEAMLESS TRANSFERRING OF “WORK IN
PROGRESS” BETWEEN DEVICES IN
A PEER-TO-PEER ENVIRONMENT**

Dahanayaka Liyanage Tharika Chathumadi

(189310v)

Thesis submitted in partial fulfillment of the requirements for the degree Master
of Science in Computer Science

Department of Computer Science and Engineering

University of Moratuwa

Sri Lanka

June 2021

DECLARATION

I declare that this is my own work and this thesis does not incorporate without acknowledgement any material previously submitted for a Degree or Diploma in any other University or institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Also, I hereby grant to University of Moratuwa the non-exclusive right to reproduce and distribute my thesis, in whole or in part in print, electronic or other medium. I retain the right to use this content in whole or part in future works (such as articles or books).

Signature: ***UOM Verified Signature***

Date: 14/06/2021

The above candidate has carried out research for the Masters thesis under my supervision.

Name of the supervisor: Dr. Kutila Gunasekera

Signature of the supervisor:

Date: 14/06/2021

Abstract

With the current technical improvements, there is a huge involvement of computational devices in human day today life which leads to a single person using multiple computing devices such as desktop computers, laptops and smartphones. A user often uses the same set of applications on each device and switches among devices according to the availability, convenience and requirement rather than carrying a single device everywhere. When the user wants to switch between devices and continue his work, he has to transfer the “work in progress” between devices. The most basic way for transferring this work is to pause the current task, transfer the data to the target device as a file, locate it on the target device, open it using a suitable application and then manually restore the previous work state. There are many technologies and applications that support transferring or synchronizing data and “work in progress” between devices. However, they depend on the availability of an intermediate service provider to facilitate such transfer. This research designs a framework which focuses on reducing the human involvement in transferring “work in progress” between two devices and also doing it without involving a third party. For this we transfer both application state and the application data. The state information represents the current state of the application which is needed to resume the “work in progress” at the target device end without user interaction. A proof of concept desktop application, VLC WIP Migrator (VLC Work In Progress Migrator), which transfers audio/video content between two Windows laptop computers is also described. The current play time of the audio/video file is considered as the state information and socket communication is used to transfer the files between devices. The time to resume the audio/video playing via VLC WIP Migrator is compared with the time taken to transfer the same audio/video file using “Nearby sharing”, a built-in application in Windows 10. The test results show that transferring via VLC WIP Migrator is both more convenient and quicker.

Keywords

Seamless, file transfer, work in progress

ACKNOWLEDGEMENT

I would like to take this opportunity to express my appreciation towards all those who contributed for making this research a success. I am truly grateful for the valuable guidance and assistance provided by my supervisor, Dr. Kutila Gunasekera. Without his thorough experience and constant advice, the work presented here would be incomplete. Finally, I would like to express my heartfelt thanks towards my family and friends for their support and encouragement throughout many days and nights dedicated for my research work.

TABLE OF CONTENTS

Declaration	i
Abstract	ii
Acknowledgement	iii
Table of contents	iv
List of Figures	vi
List of Tables	vii
1. Introduction	1
1.1. Research Problem and Research Objectives	3
1.1.1. Research problem	6
1.1.2. Research objectives	6
1.2. Thesis Outline	6
2. Literature	8
2.1 Seamless Transferring	8
2.2 Seamless Transferring Approaches	9
2.3 Synchronization	12
2.4 File Sharing Solutions	14
2.5 File Sharing in P2P Networks	15
3. Solution Framework Design	18
3.1 Design Considerations	19
3.1.1 Identify state	19
3.1.2 Preserve state	21
3.1.3 Transfer content and state	21
3.1.4 Restore content and state	22
3.1.5 Steps based on location	23
3.2 Architecture Diagram	24
3.2.1 Interaction between source device and target device	25
4. Implementation	27
4.1 Identify State	27
4.2 Preserve State	27

4.2.1 VLC Player http web interface	28
4.3 Transfer Content and State	32
4.4 Restore Content and State	32
4.5 Implementation of VLC WIP-Migrator	32
5. Evaluation	37
5.1 Transferring Delay	37
5.2 Reliability	37
5.3 Test Setup	38
5.4 Test Results	39
6. Conclusion	43
6.1 Limitations	44
6.2 Future Works	44
Reference List	45

LIST OF FIGURES

Figure 2.1 Roam system architecture	11
Figure 3.1 State of a document	20
Figure 3.2 State of a chess game	21
Figure 3.3 Architecture Diagram	24
Figure 3.4 Interaction between source and target	25
Figure 4.1 State information for audio or video file	28
Figure 4.2 Set password to web interface	29
Figure 4.3 Setting up http as the primary interface	30
Figure 4.4 Get current VLC instance	31
Figure 4.5 Get current state of VLC instance	31
Figure 4.6 Byte arrangement for data stream	32
Figure 4.7 Implementation of the design architecture	33
Figure 4.8 GUI for seamless transferrer	34
Figure 4.9 Create a WebClient and authorize the request	34
Figure 4.10 State object	34
Figure 4.11 Request to pause current audio/video	35
Figure 4.12 Building the byte stream to transfer	35
Figure 4.13 Restore application	36
Figure 5.1 File transferring using “Nearby sharing”	38
Figure 5.2 Time taken for file transferring in bar charts	41

LIST OF TABLES

Table 5.1 Time taken for transferring files using “Nearby sharing”	40
Table 5.2 Time taken for transferring files using VLC WIP migrator	40
Table 5.3 Maximum and minimum time taken for transferring	41