uOttawa

L'Université canadienne
Canada's university

FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTDOCTORALES

uOttawa

L'Université canadienne
Canada's university

FACULTY OF GRADUATE AND POSTDOCTORAL STUDIES

Mohammod Shamim Hossain
AUTEUR DE LA THÈSE / AUTHOR OF THESIS

Ph.D. (Electrical Engineering)
GRADE / DEGREE

School of Information Technology and Engineering
FACULTÉ, ÉCOLE, DÉPARTEMENT / FACULTY, SCHOOL, DEPARTMENT

Towards a Biological-inspired Framework for Multimedia Service Management
TITRE DE LA THÈSE / TITLE OF THESIS

A. El Saddick
DIRECTEUR (DIRECTRICE) DE LA THÈSE / THESIS SUPERVISOR

CO-DIRECTEUR (CO-DIRECTRICE) DE LA THÈSE / THESIS CO-SUPERVISOR

EXAMINATEURS (EXAMINATRICES) DE LA THÈSE / THESIS EXAMINERS

A. Boukerche

D. Petriu

H. Saliah-Hassane

J. Zhao

Gary W. Slater
Le Doyen de la Faculté des études supérieures et postdoctorales / Dean of the Faculty of Graduate and Postdoctoral Studies

Towards a Biologically-Inspired Framework for Multimedia Service Management

Mohammod Shamim Hossain

Thesis submitted to the
Faculty of Graduate and Postdoctoral Studies
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Electrical and Computer Engineering

Under the auspices of the Ottawa-Carleton Institute for Electrical and Computer Engineering

University of Ottawa
School of Information Technology and Engineering
Ottawa, Ontario, Canada
August 2009

Your file Votre référence
ISBN: 978-0-494-61393-1
Our file Notre référence
ISBN: 978-0-494-61393-1

Abstract

The advent of service-oriented architecture (SOA), internet and ubiquitous delivery technology has resulted in multimedia services (e.g. repurposing, streaming and conferencing services) being accessible at any time, from any device, through any network. However, there are still some problems related to heterogeneity, scalability and QoS demand of the management of such multimedia services. Some of the existing solutions are centralized, which evolve scalability problems in terms of the number of concurrent requests for the target service composition. Other solutions are distributed, which depend on the use of traditional algorithms (e.g. Dijkstra, Bellman Ford). Such distributed solutions also use replicated services, which can also result in scalability problems for large networks.

In order to mitigate the above problems, this thesis proposes a framework for multimedia service management that is based on a biologically-inspired approach. It utilizes an ant-colony-based selection algorithm for collecting the QoS requirements from the individual repurposing service in order to select the most suitable one for the desired composition process, which ensures higher scalability and efficient load balancing. It also develops a QoS-aware service selection algorithm for a multimedia repurposing service. The proposed framework's performance is validated through both simulation and prototype implementation.

Acknowledgment

This thesis would not have been possible without the mercy of Almighty, Omnipotent and Omniscient, to whom I express my gratitude.

First of all, I would like to thank Professor Abdulmotaleb El Saddik, my supervisor, for providing me with the opportunity to pursue a PhD as a member of the Multimedia Communication Research Laboratory (MCRLab), SITE, at the University of Ottawa. His tremendous support, constant guidance, constructive feedback, periodical extensive discussions and invaluable understanding enabled me to complete this thesis. I hope that he will support me morally and technically for the rest of my life.

I would like to thank the respectable members of my thesis committee, Prof. Jiying Zhao, and Prof. Dorina Petriu, for their helpful discussions concerning my work. Their insightful comments during my Ph.D. proposal have greatly helped in improving my work. I am also obliged to them for carefully reading my thesis and providing helpful feedback. Also, I would like to extend special thanks to the OCICS coordinator, Prof. Iluju Kiringa, for his continuous support.

I am grateful for the support and assistance from some of my friends, especially Anwar Hossain, Atif Alamry and Dewan Tanvir Ahmed. Thanks also to Co-op students Bogdan Soloman and Yohannes Tadesse for helping out with some of my research implementations.

No thankful words are sufficient to express my gratitude towards my parents and in-laws for their love, encouragement and prayers (doas). Thanks to my father-in-law, Ottawa University EE alumni Prof. Dr. Quamrul Ahsan, for his encouragement that I continue with higher studies. Last but not least, I extend my deep gratitude to my father, Prof. S.M. Habibullah, whose hard work, constant prayers, guidance, and unconditional affection made all of this possible.

I would especially like to thank my wife for her patience and unending support during the most challenging periods of my PhD research. Thanks to my children who were born during my studies, who miss me a lot during my study.

Table of Contents

List of Figures

List of Tables

List of Acronyms

Acronym	Definition
J2SE	Java 2 Platform Standard Edition
JMF	Java Media Framework
JNI	Java Native Interface
J2ME	Java 2 Micro Edition,
LAN	Local Area Network (LAN),
PC	Personal Computer
PDA	Personal Digital Assistant
QoE	Quality of Experience
QoS	Quality of Service
QoP	Quality of Perception
RTCP	Real-time Control Protocol
RTSP	Real-time Streaming Protocol
SAP	Session Announcement Protocol
SDP	Service Discovery Protocol
TCP	Transport Control Protocol
SCTP	Stream Control Transmission Protocol
SIP	Session Initiation Protocol
UDP	User Datagram Protocol
XML	Extensible Markup Language
Codec	Compressor- decompressor or coder-decoder
ACO	Ant Colony Optimization
BioReSS	Biologically Inspired Service Selection Algorithm
BISON	Biology-Inspired techniques for Self-Organization in dynamic Networks
UDDI	Universal Description, Discovery and Integration registry
WSDL	Web Service Description Language
XML	Extensive Mark-up Language
SOAP	Simple Object Access Protocol
PSNR	Peak Signal to-Noise-Ratio
SOA	Service Oriented Architecture
SOC	Service Oriented Computing
WSES	Web Service Emergent System
NEI	Neuro-Endocrine- Immune
CIF	Common Intermediate Format
QCIF	Quarter Common Intermediate Format
SQCIF	Sub-QCIF
GUI	Graphical User Interfaces
AMIGOS	Advanced Multimedia in Group Organized Services

NP-hard	Nondeterministic Polynomial-time Hard
NP-Complete	Nondeterministic Polynomial-time Complete
GPRS	General Packet Radio Service
GSM	Global System for Mobile communications
UMTS	Universal Mobile Telecommunications System
Mbps	Megabit per second
SP2	Service Pack2

List of Symbols and Notations

Symbol	Meaning
θ	resolution (e.g. CIF, QCIF) of the multimedia (e.g. video) stream
ϑ	resolution of the stored video (e.g. H.263, H.264, MPEG-4)
β	bandwidth of the stored video stream
ς	area factor
β_a	available bandwidth for the multimedia stream
β_γ	bandwidth required for the multimedia (video) stream in order to get maximum user satisfaction
f_{max}	the maximum frame rate for the multimedia stream
f_a	available frame rate for the multimedia stream
λ	average arrival rate of the service request
T	response time in seconds.
N	number of concurrent service requests
$\emptyset_i$	pheromone value $\sum_{n\in N} \emptyset_n = 1$, Where, N represent neighbours
P_i	probability of selecting next repurposing service node
δ_e	selection variable to indicate whether service is selected or not
s_i	a component satisfaction function
s	a sender or server
d	a destination or receiver
MS	multimedia streaming service,
RS_n	repurposing services
SC_n	service composition plan
$P_R(s,d)$	repurposing path from a server (s) node to a receiver node (d)
R_N	set of repurposing services considered by the system
R_v	set of repurposing services that have already been considered
R_i	set of connecting (or neighbouring) repurposing services

E	set of links between repurposing service nodes
V	set of proxy nodes
R_{v_i}	set of a set of n_i repurposing services for each node v_i
Q_{in}	input format for repurposing services
Q_{out}	output format for repurposing services
$e \in P$	each link on the path
$q^e_{\beta_\gamma}$	resource required (QoS attributes for bandwidth) for the service link
Q_{β_a}	resource available (QoS constraints for bandwidth) for each service link
q^e_D	resource required (QoS attributes for delay) for each service link
Q_D	resource available (QoS constraints for delay) for each service link
S_e	score for every repurposing link
f^e_i	user satisfaction function of every repurposing link
Υ	reinforce variable (whose value is 0.3)
α	$\alpha \epsilon [0.2, 0.5]$ relative importance of the heuristic correction with regard to pheromone values in the pheromone routing table

Chapter 1. Introduction

This chapter presents research motivation with an application scenario, followed by the research objectives and solution overview. The thesis contributions are highlighted. The chapter concludes with a presentation of the organization of the thesis.

1.1. Motivation

The unprecedented growth of ubiquitous communication infrastructures, emerging multimedia networking systems, multimedia coding standards, and pervasive handheld devices facilitates the access to multimedia services at any time, and from anywhere. However, managing these multimedia services is a challenging task due to the heterogeneity of the services, device capabilities, network constraints and client's QoS demands. In order to mitigate the heterogeneity and to deliver content to the resource-constrained handheld devices, there is a growing need for multimedia content repurposing services. The content repurposing [35], [105] service is a process of converting or transforming a multimedia content (e.g. video) from one form to another. However, a single software solution or a single repurposing service [59], [61] is not enough to accommodate all repurposing needs of the clients. A multimedia content may require multiple conversions to customize the content for the target user, which can be performed by using multiple repurposing services. In this case, one or more simple repurposing services are composed to customize the user's desired content. To illustrate, we consider a live game streaming scenario as depicted in Fig. 1-1.

In this scenario, a streaming server (source) delivers the video content in a single format which is viewed by many users (clients). The users are located at distributed sites. Each of them wants to render the streamed content in their respective devices, which has different rendering capabilities in terms of processing power, screen size and bandwidth requirements. The system customizes or repurposes the content for the user based on their varying requirements. For example, a user is capable of viewing a low bit rate H.264 vid-

eo, where original video is in M-JPEG format. In order to serve this user, the system repurposes the high bit rate M-JPEG content by composing two separate repurposing services, which are MJPEG to H.263 and H.263 to H.264.

The described scenario may also be considered as an example of GOOGLE or CBC, where the same video is customized (or repurposed) in different steps according to user's capability, as well as network connections. The end result is then sent to ubiquitous users that range from PDA to desktop.

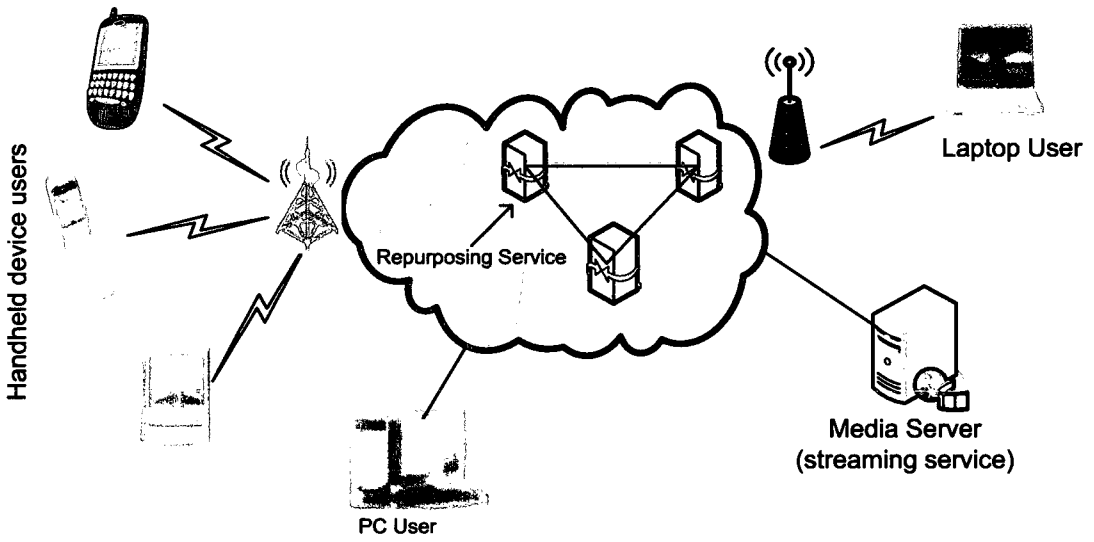

Figure 1-1. Example scenario that make use of service composition in a multimedia management framework

The scenario illustrated in Fig. 1-1 demonstrates the need for a service management framework, which will facilitate the complex repurposing task by selecting and composing required number of simple services. There exist clear beneficiaries of such a framework such as video conferencing, adaptive video streaming for the next generation network (NGN), autonomic service management for wireless sensor networks and Ambient Intelligence (AmI) applications.

1.2. Research Objectives and Solution Overview

Challenges and Objectives

There exists an increasing demand for scalable multimedia service management in large scale distributed multimedia systems, especially when a large number of users make simultaneous composite service requests. One of the issues in this case is to dynamically compose services from a number of simple services. Once composed, another issue is to maintain the composite services in a highly distributed and constantly-changing computing environment. In order to customize the composite services according to users' need, different services must be selected in a way such that it ensures scalability and load balancing. Thus, the challenge is to select services for the target composition based on a number of media-related QoS parameters, while ensuring scalability.

The objective of this thesis is to solve the service management issues for the multimedia services that are accessed by various resource-limited devices connected through heterogeneous networks. In particular, this research will focus on selecting the appropriate repurposing services for target composition tasks by using a bio-inspired approach to ensure that the composed service paths efficiently and effectively function in a dynamic environment.

Solution Overview

To address the above challenge, traditional selection algorithms that are based on Bellmanford [22] and Dijkstra [30] are initially used. The effectiveness of these algorithms is constrained by their limitations, which include complexity for runtime decisions, as well as cost effectiveness for large networks. Some traditional algorithms are centralized, which results in scalability problems in terms of the number of concurrent requests for the target service composition. Some are also distributed, which result in a lack of scalability for larger networks. Specifically, highly dynamic traffic and heterogeneous network conditions significantly degrade the performance of a traditional algorithm and make it subject to error or loss. This causes high latency and repurposing delays. Moreover, the traditional methods do not have enough flexibility to satisfy new service demands, end user's preferences and QoS for a multimedia transmission in a highly dynamic heterogeneous network environment.

A solution to cope with the above-mentioned constraints of traditional algorithms is to draw inspiration from biological systems. Such systems can effectively manage large numbers of unreliable units, which can be referred to as nodes or multimedia services in network environments. Biological systems are robust to failure or loss [29]. It has been a significant research challenge to characterize the ideas that enable biological systems to function, and then apply those ideas to multimedia systems in a heterogeneous network environment. Moreover, biologically inspired systems are robust to error and are able to function in highly dynamic heterogeneous network environments [29]. It additionally provides higher scalability, as well as better load balancing.

The thesis proposes an ant-based [52], [55] selection algorithm to solve the repurposing service selection problem in a scenario, where the number of repurposing services are distributed in several proxies over the networks. This algorithm is based on the biological foraging concept [33] and provides an efficient utilization of network resources in a continuously changing network environment. The idea behind the algorithm is that after receiving the request from the sender, the server generates a number of ants at fixed intervals. Those ants then move through intermediary repurposing service nodes, for the desired repurposing services. On returning to the source from the destination, ant follows a repurposing path traversed by forward ants, and updates QoS parameters as well as pheromone. After arriving at the sender, the best repurposing path or service is selected, based on the desired QoS.

The bio-inspired selection algorithm is one of the key components of the proposed framework. The proposed framework includes the following key phases:

Service Discovery and Collection of QoS and Other Profiles: Based on the service requirements, such as repurposing, streaming and selection, this phase discovers the available appropriate services. QoS parameters and other service profiles (e.g. device, network and user preferences) are then collected prior to the selection and composition. Based on the collected information, the service graph is then derived. The service profiles are discussed in Chapter 3.

Service Selection and Composition: This is the main component of the proposed bio-inspired management framework. As a multimedia service user expects to receive a constantly growing selection of dynamic functionalities, it is not possible for a single repur-

posing service to fulfill such a demand. In other words, there is no single complete software solution that can satisfy all repurposing needs to customize the content, because the service may be crushed or overloaded. Therefore, the use of only one repurposing service to customize the content is likely to be a difficult or impossible task.

Multiple repurposing services that are distributed in different networks are therefore required to select and compose, in order to receive the desired content for the user. Such services repurpose the requested media content in a chain fashion, such that the output format of one repurposing service is supplied to the input format of the next repurposing service, until the requested content satisfies both the outgoing target channel's and the end user's requirements. Both non bio-inspired as well as bio-inspired mechanism are used for the selection of service and their performance is compared. These are presented in chapter 4 and 5.

Service Maintenance: After service selection and composition, composite service maintenance is essential to ensuring the proper functioning and serviceability in terms of load management, scalability and self recovery throughout the service. Maintaining such serviceability characteristics in a highly distributed and heterogeneous computing environment is a crucial and essential task. In this type of environment, numerous services may dynamically join or leave the environment, and the service may be crushed or overloaded. Moreover, this environment may have diverse resources in terms of devices, network traffic, and topology. These service maintenance issues are discussed in Chapter 6.

The proposed framework is one of the few attempts at bringing the combined potential of service-oriented architecture [90] and the foraging behaviour of ant colony concepts [33] into multimedia service management for heterogeneity and scalability problems. As has previously been mentioned, the bio-inspired repurposing service selection algorithm [55] is the essence of this framework. The system is capable of accommodating both wired and wireless users by adapting to the constantly changing environment. The system also can effectively manage a large number of service nodes. The proposed framework is validated through implementation, as well as simulation, in terms of scalability, load distribution and some QoS parameters. The scalability is evaluated in terms of concurrent service requests.

1.3. Thesis Contributions

The major contributions of this thesis are:

- The design and development of an ant-based algorithm for multimedia service selection;
- The design and development of a framework that is focused on integrating the combined potentials of service-oriented concepts and biologically-inspired concepts for multimedia service management research;
- The design and development of QoS-aware service selection algorithms for multimedia repurposing services.

1.4. Publications

The following papers have been published as a result of this research:

Papers in Refereed Journals

- M. Shamim Hossain, and A. El Saddik, "QoS requirement in the Multimedia Transcoding Service Selection Process," IEEE Transaction on Instrumentation and Measurement, Vol. 58, No. 12, December 2009. (Accepted, to appear)
- M. Shamim Hossain, A. Alamri, and A. El Saddik, "A Biologically-Inspired Framework for Multimedia Service Management in Ubiquitous Environment", Wiley Concurrency and Computation: Practice and Experience, Vol. 21, No. 11, pp. 1450-1466, 2009.
- M. Shamim Hossain, and A. El Saddik, "A Biologically-inspired multimedia content repurposing system in heterogeneous Network Environments, ACM/Springer Multimedia Systems Journal, Vol. 14, No. 3, pp. 135-143, 2008.
- M. Shamim Hossain, and A. El Saddik, "Scalability Measurement for Multimedia Repurposing System", International Journal on Advanced Media and Communication, Vol. 2, No. 3, pp. 265-285, 2008.

Papers in Refereed Conferences

- M. Shamim Hossain, A. Alamri and Abdulmotaleb El Saddik, "QoS-Aware Service Selection for Multimedia Transcoding," IEEE-I^{2}MTC 2008, May 12-15, Victoria, BC, Canada.
- M. Shamim Hossain, A. Alamri and Abdulmotaleb El Saddik, "A Framework for QoS-aware multimedia service selection for Wireless Clients," ACM-WMuNeP'07, 22-26 Oct. 2007, Chania, Creta Island, Greece.
- M. Shamim Hossain, and A. El Saddik, "Multimedia content repurposing for heterogeneous Wireless clients," IEEE-ICSPC 2007, Nov. 24-27, 2007, Dubai, UAE.
- M. Shamim Hossain, M. A. Hossain, and A. El Saddik "Multimedia Content Repurposing in Ambient Intelligent Environment," IEEE-AIMS 2007 (with ICDE), April 20, 2007, Istanbul, Turkey.
- M. Shamim Hossain, and A. El Saddik, "Proxy-based Visual Content Repurposing using Selection Algorithm," IEEE-ICN 2007, April 22-28, 2007, Martinique, French Caribbean.
- M. Shamim Hossain, and A. El Saddik, Multimedia Content Repurposing in Heterogeneous Network Environments," In Proc. IEEE-ICECE 2006, Dhaka, Bangladesh, Dec. 19-21, 2006.
- M. A. Hossain, M. Shamim Hossain and A. El Saddik, "Ambient Intelligence Multimedia Management Framework," In Proc. IEEE AICS 2006, Sept. 7-8, 2006, Sheffield, UK.
- M. Shamim Hossain, and A. El Saddik "Scalability Analysis for Personalized Multimedia Repurposing System", In Proceedings of the IEEE Instrumentation and Measurement Technology Conference (IEEE-IMTC-2006), Sorrento, Italia, 24-27 April 2006.
- M. Shamim Hossain, M. A. Rahman and A. El Saddik, "A Framework for Repurposing Multimedia Content", In Proc IEEE-CCECE-2004, Niagara Falls, Ontario, Canada, May 2-5, 2004.

Chapters in Books

- M. Shamim Hossain and A. El Saddik, "Multimedia content repurposing in wireless environment," in Encyclopedia of Wireless and Mobile Communications, Borko Furht Ed., CRC Press, Taylor & Francis Group, 2007.
- A. El Saddik and M. Shamim Hossain, "Multimedia Streaming for wireless communication," in Encyclopedia of Wireless and Mobile Communications, Borko Furht Ed., CRC Press, Taylor & Francis Group, 2007.
- A. El Saddik and M. Shamim Hossain, "Multimedia Content Repurposing" in Encyclopedia of Multimedia, Borko Furht Ed., Springer Verlag Book Series, Feb. 2006.

1.5. Thesis Outline

This thesis is organized into seven chapters.

Chapter 1 introduces the motivating application scenario, which is followed by the research objectives and solution overview. The outline of contributions and the list of publications as an outcome of this research are also presented. The chapter is concluded with thesis organization.

Chapter 2 briefly describes the state of the art related to multimedia service management. It also presents the problem formulations of the thesis and notations that will be used throughout the thesis.

Chapter 3 discusses the proposed framework's architecture, requirements and functionality. The details of the participating components of the framework are also discussed.

Chapter 4 describes the non bio-inspired selection algorithm, along with some experimental results.

Chapter 5 presents the details of the bio-inspired selection algorithm and its key components. It also presents the comparison between the non bio-inspired and the bio-inspired algorithm.

Chapter 6 validates the proposed multimedia service management framework through simulation and prototype implementation. The behavioural characteristics with the increased number of composition requests are then presented. This chapter also discuss about composite service maintenance, such as scalability, and load management. Finally, it concludes with the discussion of qualitative analysis through a small scale usability study.

Chapter 7 draws conclusions and offers suggestions for future work.

Chapter 2. Background, Related Works and Problem Formulations

In this chapter, we firstly introduce some general background knowledge to multimedia repurposing service. Then, we present the state of the art research work and the formal formulations of the addressed problem. The proposed research addresses service management of multimedia services-based composite processes using bio-inspired mechanism. One of the purposes of the research is to leverage the potential of SOA for the multimedia service, by executing and managing the composite multimedia service through the use of a bio-inspired service selection approach. Thus, the proposed work is related to a variety of interrelated interdisciplinary research efforts. The related work is categorized into several groups, including a service management framework, SOA for multimedia services, the bio-inspired approach for service selection, and bio-inspired approaches for service management. Each of the above is described and compared with our proposed framework below.

2.1. Background

2.1.1 Multimedia Content Repurposing Service

Most of the multimedia content that is designed for desktop computers (PC) and high speed wired networks are not suitable for wireless devices with limited display capability, processing power and network bandwidth. A solution to the issue includes multimedia content repurposing service being applied to eliminate the mismatch between the rich multimedia content and the limited wireless terminal capabilities. The process of converting or transforming a multimedia stream from one form to another is called content repurposing. Sometimes, content repurposing is referred to as either transcoding or adaptation. This transformation can take place in different ways, some of which are as follows [35] :

- Conversion of different modes (e.g. Speech to text, voice to image, image to text)
- Conversion of video coding format (e.g. Mpeg-1 to MPEg-4)
- Conversion of coding parameter
  - Frame rate (e.g. 30fps 20 fps)
  - Bit rate (e.g. 6 Mbps TV broadcast to 56 kbs cell phone)
  - Spatial resolution (e.g. CIF to 4QCIF)
- Conversion of Spatio-temporal resolution (e.g. VGA to QVGA)
- Content Summarization (e.g. Viewing two hours of news in 15 minutes by highlighting the more pertinent information)
- Visualization of Content (e.g. key fame visualization and browsing for a video in order to understand that)
- Selection of best variation based on MPEG-21 [63] standards:
  - Network Capabilities (e.g. bandwidth, delay and error characteristics)
  - Terminal Capabilities (coding and decoding capabilities, device profiles, and I/O capabilities)
  - User Preferences (e.g. content preferences, presentation preferences, accessibility, mobility and destination)
  - Natural Environments (e.g. location, noise level etc.)

2.1.2 Repurposing Techniques

There are different ways the repurposing can be performed. The brief descriptions of some of those techniques are as follows:

Bit Rate Repurposing

The bit rate repurposing [75] is required in the case of multimedia transmission over heterogeneous wireless networks when the wireless channel capacities (bit rate) of the outgoing channel is less than those of the incoming channel. The bit rate repurposing is essential for those multimedia applications (e.g. multipoint video conference) in a wireless environment. More specifically, where an incoming multimedia stream needs to transmit to multiple wireless clients through a channel with varied capabilities, or where the channel characteristics through which the resulting multimedia stream transmitted is un-

known. Through bit rate repurposing, the bit rate of pre-encoded and/or live multimedia can be dynamically repurposed to the available bandwidth and variable communication conditions of wireless networks. For delivering multimedia to wireless clients in wireless environments, the following steps [74] are required: a) with the help of content repurposing, high bit rate multimedia content is transformed into low bit rate; b) repurposed bit rate is adjusted, based on the varying wireless channel bandwidth; c) the end to end delay is controlled within the application requirement.

Standard Repurposing

In addition to bit rate and frame rate repurposing, sometimes standard/syntax repurposing is required. This type of repurposing is also referred to as heterogeneous repurposing [4]. It is completed between various standards, such as converting from MPEG-2 to MPEG-4, H.263 to H.264/MPEG-4 and M-JPEG to H.264/H.263. According to [74], the heterogeneous repurposing algorithm consists of the following steps: a) frame header adjustments; b) multimedia data translation from one standard to another; and c) required bit stream stuffing for synchronizing. In literature, there have been some efforts [4], [76], [86] made with this repurposing approach. Liang et al. [76] presents a heterogeneous repurposing approach that can repurpose between MPEG-4 visual simple profile and H.263. Nguyen [86] presents efficient repurposing between H.263 [64] and H.264 [2] .

Temporal and Spatial Resolution Repurposing

In order to repurpose multimedia stream for a low bandwidth wireless link, a high repurposing ratio is required. However, if the incoming multimedia stream is repurposed with the full frame rate as the incoming stream, the high repurposing ratio may cause quality degradation. Temporal resolution repurposing is required for such wireless clients that have resource constraints towards displaying high quality multimedia. The temporal resolution or frame rate repurposing reduces the bit rate and makes the quality of the multimedia acceptable to the users. For example, some cell phones or PDA can only play multimedia applications at a very low frame rate (e.g. 10 or 15 fps), so high quality (30 fps) multimedia intended for PC are repurposed to lower frame rate for such hand held devices. One way to do the temporal resolution repurposing is to drop or skip frames. Spatial resolution repurposing is crucial for the resource-limited wireless clients that have

limited display capabilities, such as screen size, resolution and colors. For the spatial resolution repurposing frame size from the incoming multimedia is down sampled. The resolution repurposing has two advantages: a) lower resolution multimedia can easily be delivered to the display-limited wireless clients over wireless networks; and b) the user need not scroll down bars in order to play multimedia.

Error-resilient Repurposing

As known that the heterogeneity of the client networks makes the encoder difficult to repurpose the multimedia content to a wide variety of different channels conditions in error-prone wireless networks. In order to resolve this issue, an error-resilient repurposing approach is used. By using an error-resilient repurposing approach, the delivered multimedia quality in wireless environment can be improved, even in the existence of error while maintaining input bit rate. In this approach, the network node (such as a base station or wireless access point or proxy server) connected to wireless networks, performs error resilient multimedia repurposing by injecting error-resilient features into the multimedia stream. Error-resilience tools [116] include Reversible Variable Length Coding (RVLC), Multiple Description Coding (MDC), Reference Picture Selection, Multiple Reference Pictures, Adaptive Intra Refresh, Resynchronization marker and Data Portioning.

2.1.3 Repurposing Types based on Content Variation

According to content variation, multimedia content repurposing is divided into the following approaches:

Static Repurposing

In this approach, a server pre-processes and stores multiple versions of the multimedia content. A version is repurposed based on the network capabilities, client capabilities and user preferences (i.e. these factors determine how the content varies). When a client makes a request, the appropriate version is selected from among the existing versions of multimedia content, without any alteration. Most current websites use this approach to avoid extra processing. The advantage of this approach is that there is no processing required, because the desired version is already available. This allows for quick delivering

and optimal bandwidth utilization. On the other hand, there is the need for large storage capacities to house the different formats of the multimedia content. Also, every time new formats/versions are invented, content on the server needs to be converted to the new version, which renders the task of maintaining the content a costly one.

Dynamic Repurposing

In this approach, after reading multimedia content from the server, it is repurposed "on the fly" in order to match the client's capabilities. While this approach obviously requires low storage space because the server keeps a single version of the original multimedia content, it requires significant computing power. There are many operations involved other than repurposing, such as accepting and dynamically classifying user requests, providing personalization, holding and constantly updating user preference profiles [103]. This problem could be solved by content summarization, as the repurposing could be referred to as text summarization, format/version changes, reduction of image or audio quality, cutting down number of key frames and audio to text repurposing. Due to the heterogeneity and mobility of devices and their ubiquitous networks connections, dynamic content repurposing is critical.

Hybrid Repurposing

This approach was introduced by Shin and Koh [103] and selectively uses both the static and the dynamic repurposing approaches, in order to preserve bandwidth and storage space. The static approach is used for the frequently accessed multimedia content, while the dynamic approach is used when multimedia content is infrequently accessed. For example, a cellular phone accesses multimedia content frequently. In this situation, static repurposing is more appropriate than dynamic, as for each request, the server sends the appropriate version or format to match the client's capabilities. When the content to be accessed is large and infrequently required, network bandwidth requirements are kept at a low priority. Thus, the hybrid approach, in this case, selects the dynamic repurposing scheme.

2.1.4 Repurposing Types based on Architectures

Repurposing may occur at some point between the creation of the initial content and the final rendering on the client device. This can take place at the client's end, at the server's, or at an intermediate proxy between server and client. Based on the architecture, repurposing can be classified into: client-based, server-based, or proxy-based. Each of the mentioned architectures possesses strengths and weaknesses. The main issue is to consider the effectiveness of content repurposing and the efficiency of utilization of the client capabilities in terms of processing power, bandwidth, and storage capacity.

Client-based Repurposing Architecture

This is a raw form of repurposing that is mainly performed on the client devices, without affecting the communication protocol. This approach helps to avoid transporting large volumes of device and user metadata to the server or the intermediate proxy for repurposing. The repurposing approach is performed on the client device without changing its characteristics, by transferring the entire application from the server to the client. For example, multimedia content and its corresponding style sheets are delivered to the client. Then conversion is performed. Here, multimedia content may include text, images, audio, and video. Another way to client-based repurposing is to deliver the entire multimedia content to the client for the appropriate navigation and selection, based on the client capabilities.

Used with thin clients such as mobile computing and cellular phones, client-side architecture has significant limitations, including restrictions of network bandwidth, device memory, and processing power. However, there are some advantages such as incurring the repurposing overheads at the client site, rather than server end. This architecture is also advantageous when the server is unable to determine device capabilities from the request. Client-based repurposing architecture does not depend on the employed communication protocol. The approach helps to avoid the sending of large extents of device and user profile information to the server or intermediate proxies for repurposing. Opera Software is another example for client side architecture. It introduces small screen rendering (SSR) and medium screen rendering (MSR) [35]. SSR repurposes contents and generates a user-friendly version of the contents for small mobile browsers, like smart phone, therefore eliminating the need for horizontal scrolling. MSR identifies the Web

page's content and individually adapts these different elements to fit medium-sized screens that range from PDAs to low resolution TVs. Original fonts, colors, design and style are left virtually untouched.

Server-based Repurposing

In this architecture, repurposing is performed at the server by determining a client's capabilities in terms of storage, network bandwidth, processing power, screen size, media format (e.g. HTML, XML, WML etc), and user preferences (e.g. language, fonts etc.). Both static (off-line) and dynamic ("on the fly") repurposing are supported by this architecture. The Static approach generates pre-repurposed variants of the same content off-line, in order to match client and user capabilities. The dynamic approach repurposes content "on the fly" when a request is received. For example, due to the lack of proper translation engines, multimedia content with different languages is appropriately repurposed by the static approach; the dynamic approach is not useful here. Language preferences can be accepted via content negotiation, using the HTTP requester header field. The server can discover the appropriate content version by parsing the HTTP header as the case in using MPEG-21 digital item adaptation.

In the server-side architecture, authors can control the repurposed result under different client characteristics and user preferences with a lower cost. In reality, however, it cannot repurpose the content to complex dynamic context. This architecture may depend on the communication protocol used in the content delivery. For example, in Stateful protocols, such as RTP, the server requires the client's response to perform efficient repurposing. However, in a stateless protocol such as HTTP, it does not happen. In that case, repurposing depends on the client or server interaction. Existing Server-side repurposing solutions include IBM WebSphere Transcoding Publisher, BEA Weblogic and AvantGo. All of these solutions are suitable for wired and wireless clients. An example of server-side repurposing is what is used by some news providers such as CNN or Canadian Broadcasting Corporation (CBC). They provide several different media formats on the server representing the original media content. This media content can be selected by the client, based on its preferences and capabilities. For example, three different formats: QuickTime, Real Player, and Windows Media Player, are offered by CBC or CNN, employing three different network connections: Dial-up (56 kbps), GPRS (144 Kbps) and

DSL Basic (256 kbps). When a user wants to watch a streaming video from CBC or CNN, he or she should select the format supported by his hardware or software and the network connection that he or she is using.

The advantages of using server-side repurposing architecture are:

- Content creators have more control over how the content is presented to the clients.
- The server offers more processing power than the client devices, which is very important in the case of Video Repurposing.
- In the event of the existence of multimedia content in XML, XSLT is used to transform this content to another appropriate markup language. This will match the browser's capabilities/presentation, such as using Scalable Vector Graphics (SVG), and filtering HTML documents to WML.

The weaknesses of using server-based content repurposing architecture can be summed up with the following:

- Not all browsers support content negotiation, or the server may not have control over all browsers. Therefore, the server must make assumptions or use default parameters, based on the browser's ability to present the content.
- Heavy server-side applications may slow down the server.

Proxy-based Architecture

In this architecture, the proxy server is located on the network between server and client. The proxy, which makes request to the server on behalf of the client, receives content response from the server, analyzes and repurposes the requested content on the fly; and finally sends the repurposed content back to the client, based on its capabilities. The repurposing performed by the proxy is transparent to the clients and server. The proxy can cache the repurposed results for future use, which reduces the need for re-repurposing and maintaining multiple variants of the same content. This architecture supports hybrid repurposing and addresses the heterogeneity issues, and reduces the overall end to end response time. Existing proxy-based repurposing include IBM's Web Intermediaries (WBI) [61], UC Berkeley's Ninja project [87] etc.

Repurposing also can be defined as a combinatorial process [106] where multiple transcoders can be chained together, in order to obtain a larger repurposing sequence. For example, for the task of Repurposing MPEG-2 video to Quicktime, the video could first

be transcoded from MPEG-2 to MPEG-1 using one repurposing service, then the output could be passed on to another repurposing service, in order to obtain Quicktime. The result would be exactly the same as if Repurposing had been performed in one step. First of all, this approach ensures that even with a limited number of transcoders, more complicated repurposing can be done. Secondly, it allows for the computations to be distributed.

Proxy-based approach has a number of benefits [8], [50] , including improved opportunities for proxy caching [8], [79] improved client-perceived latency, leveraging the installed infrastructure, and properly scaling with the number of clients. It also provides a clear separation between content creation and content Repurposing. Therefore, the proxy based approach is a better solution for the successful distribution of multimedia content, based on the user and device profile. Due to the nature of the real-time rich multimedia content and its high complexity of repurposing, the proxy-based repurposing approach [8], [47] is the most suitable one. To reduce this complexity, a series of repurposing proxies can be chained together [78], [80]. Now the issue is to find the best sequence of repurposing paths between the sender and the receiver that will maximize the user satisfaction. This can be done through using a proper service selection algorithm, either traditional or bio-inspired. The focus of the thesis is to use bio-inspired service selection algorithm.

2.1.5 Protocols, Coding Standard involved in Multimedia Content Repurposing Service

Protocols

Real Time Protocol (RTP) [102] in combination with the Real Time Control Protocol (RTCP) is the key protocols for multimedia streaming over wireless networks. The common approach for multimedia streaming is to use RTP/UDP for the multimedia content and RTCP/TCP or RTCP/UDP for the control. For the media control the Real Time Streaming Protocol (RTSP) or the Session Initiation Protocol (SIP) is used. In order to initiate and establish multimedia streaming session, the Session Announcement Protocol (SAP) is used together with the SIP and/or RTSP protocols. The Session Description Protocol (SDP) is used to describe sessions including audio, video, bit rate and codec.

RTP and RTCP

Real Time Protocol (RTP) [102] is Real-Time Transport Protocol for transmitting real-time multimedia over IP networks. RTP typically runs on top of the UDP. RTP [102] comes with the Real Time Control Protocol (RTCP) [40]. RTP is responsible for transferring the multimedia content while; RTCP is responsible for the control information. RTP itself does not guarantee QoS or real-time delivery of multimedia, however, it does provide mechanism that supports streaming multimedia in real-time. RTP provides a mechanism for real-time, end-to-end multimedia delivery by adding time stamps, sequence numbering and type identification. RTCP provides QoS feedback information including the number of lost packets and delay and jitter. The multimedia streaming server can use the feedback to adjust its operation, such as to repurpose bit rates or frame rates.

RTSP

Real Time Streaming Protocol (RTSP) [101] is an application-level multimedia control protocol that is used in multimedia streaming to establish a streaming session. In other words, it acts as a 'remote control' for multimedia streaming. It supports VCR functionalities [7], such as play, start, pause, resume and record. RealPlayer is an example of an RTSP application which provides the mentioned VCR functionalities. It usually uses RTP to control the stream.

SDP and SAP

The Session Description Protocol (SDP) [48] is used to describe a multimedia session for the purpose of session announcement or session invitation. The SDP message contains the following information that describes the session: session name and purpose, session time, type of media (voice or video), media format (e.g. H.264, MPEG4, and GSM), transport protocol and port number, bandwidth requirements, and contact information. SDP does not have its own transport protocol; it is used with the SIP and/or RTSP to deliver the session information to destinations. However, for the multicast session, it is used with SAP. The Session Announcement Protocol (SAP) [49] is a protocol that users can use to periodically send sessions description like SDP. Through the Session Announcement Protocol (SAP) [49], users advertise the availability of multicast session to other users, while other users listen to this multicast address to be informed of which sessions

that are announced, followed by a description of each session. After getting all of the session-related information, SIP is used to initiate the session.

SIP

Session Initiation Protocol (SIP) [98] is a signalling protocol that is used to initiate sessions and to invite users to participate in multimedia sessions. SIP can establish sessions for audio/videoconferencing, multimedia streaming, and interactive gaming. It is also used for VoIP telephony. It supports user mobility.

TCP

Transmission Control Protocol (TCP) resides on top of the Internet Protocol (IP). TCP guarantees reliable communication on top of an un-reliable wireless communication networks via acknowledgements and retransmissions. It also provides flow and congestion control. For multimedia streaming, TCP/IP is used for controlling the multimedia stream, for example to request a stream or to play a stream. TCP is not suitable for delivering multimedia stream, because it does not give time integrity. In streaming multimedia over wireless communication networks, the TCP's congestion window may limit the transmission rate to below the multimedia bit rate [36].

SCTP

Stream Control Transmission Protocol (SCTP) [107] is a reliable transport layer protocol that operates analogously to TCP or UDP. It runs on top of IP. It can transport multiple independent message streams in parallel, while TCP transports a byte stream. SCTP is the packet format, where payload is transmitted in the form of message, whereas for TCP, payload is transported as a unified chunk of data. It ensures reliable packet delivery and is resistant to flooding and masquerade attacks. It offers the following services to its users [107]: a) acknowledged error-free non-duplicated transfer of user data, and b) data fragmentation to conform to discovered path MTU size. It may be mentioned that the acknowledgement and congestion avoidance function is responsible for packet retransmission if acknowledgement has not been received in due time. SCTP has a backup mechanism, which facilitates to recover from link failures applicable to wireless links.

UDP

Even though User Datagram Protocol (UDP) provides unreliable communication, it is used for delivering multimedia streams, because small packet loss in multimedia streams is acceptable. For multimedia streaming, UDP/IP is used for transmitting multimedia streams; it does not provide flow or congestion control [39].

Coding Standard

As resource bit rate, coding efficiency and error resilience features are important for the delivery of real-time multimedia in heterogeneous network environment (both wireless and wired); the following CODECs (e.g. H.263, H.264, MPEG-4 and M-JPEG) are used for wireless multimedia streaming:

H.263

H.263 [61] is a video coding for low bit rate communication. It is suitable for wireless multimedia streaming, due to the bandwidth constraint of the wireless communication environments. The Version 2: Interactive and Streaming Wireless profile of H.263+ as well as Version 3: Interactive and Streaming Wireless profile of H.263++ provides enhanced coding efficiency and enhanced error resilience for the delivery to wireless devices [35]. The 3GGP also selected H.263 as a mandatory codec for multimedia applications like streaming over 3G wireless networks. Baseline profile of H.263 also supports multimedia streaming over 3G wireless networks.

H.264

The H.264/the MPEG-4 Part 10 [44], [62], is a video CODEC that is suitable for high quality video streaming over wireless networks. More specifically, it exhibits better error concealment, temporal resolution and bandwidth utilization properties than any other codec (e.g. MPEG-2, H.263). It has a very wide range of multimedia applications over heterogeneous networks, from low bit-rate multimedia streaming to multimedia messaging services. H.264 has been adopted by the streaming standardization bodies like 3GPP (for GSM) organizations and is under final consideration with the 3GPP2 (for CDMA2000) organization. H.264 is used for multimedia applications (steaming) over 3G wireless, because H.264 considers required characteristics like limited bandwidth, transmission power, and compression efficiency for target wireless multimedia delivery. H.264/MPEG-4 uses the Real Time Transport Protocol (RTP) as a transport protocol for

the multimedia streams during streaming. H.264 enables to deliver high quality multimedia at a lower data rate than MPEG-2 and to play back on wireless devices.

MPEG-4

The use of MPEG-4 [62] for wireless streaming applications centers on the Simple Visual Profile and The Simple Scalable Visual Profile, due to the device capabilities (computational resources) and the data rates associated with wireless clients. Both of these profiles are appropriate for delivering multimedia over 2.5G and 3G wireless networks, which have highly variable throughput and error-prone characteristics. The Simple Visual Profile of MPEG-4 provides efficient, error resilient coding of rectangular video objects that are suitable for applications (e.g. wireless multimedia streaming, wireless multimedia messaging service, video conferencing, surveillance system) whereas low bit rate and low resolution are mandated by other conditions such as network bandwidth and device capabilities. The Simple Scalable Visual Profile of MPEG-4 adds support for coding of temporal and spatial scalable objects to the Simple Visual Profile. It is useful for wireless multimedia streaming because it provides services at more than one level of quality, due to bit-rate or decoder resource limitations. It can adapt to different client side environments and link bandwidths, the temporal and spatial scalability.

M-JPEG

Video that applies the JPEG compression format on all of its frames of video sequence is called Motion JPEG. JPEG (Joint Photographic Expert Group) [60] is a standard jointly developed by ISO/EIC JTC1/SC2/WG10 and ITU-TS for still image compression. In this video compression, it compresses each frame individually, without reference to any other frames in the sequence (i.e. it does not consider inter-frame redundancies). It has advantages over the other, including that the loss of frames does not affect other frames; it greatly improves the storage and transmission overhead. Furthermore, it has less encoding complexity and delay and it allows for easier editing. It is used by many proprietary formats e.g., AVI and QuickTime. M-JPEG is not a video coding standard or is even standardized, but is used in the high quality video capture process as a raw data format to be easily edited and compressed into another format. Therefore, many network-enabled

cameras provide M-JPEG streams so that clients connected to the browser can easily view the M-JPEG streams.

2.2. Related works

2.2.1 Service Management Framework and Project

Currently, there are numerous existing researches that are related to web service management [38], [88], [91] and web service composition [5], [124]; however, only a select few have addressed multimedia service management [13], [68]. Multimedia service management deals with multimedia service composition [85] and service maintenance or monitoring (e.g. load balancing, scalability, and fault-resilience). Multimedia service composition is a process where services are first integrated to create a new multimedia service and then spread the service over a heterogeneous network and distributed system infrastructures [85]. In other words, Multimedia service composition allows the composition of complex multimedia services that can be customized for the user in a ubiquitous environment. Furthermore, the composition process uses simple component services from heterogeneous environments without regard to the details of the underlying technologies.

In [70], Kalasapur, Kumar, and Shirazi, describe a dynamic multimedia service composition scheme for a pervasive computing environment, where SOA was envisioned in their future work. In [95]-[96], the SAHARA project deals with the fault-resilience and load balancing issues as service management for the service composition. The SpiderNet [46] project discusses the QoS-aware service composition framework, where the fault-resilience issue was addressed. In [117]-[118] Vukovic and Robinson describe a context-aware service composition framework, where Artificial Intelligence (AI) planning is used to control management issues like scalability and failure recovery. In [68], Jin and Nahrsted proposed a QoS-aware service management framework, where scalability was evaluated in terms of network size and the client application's population size.

Mao et al. [80] developed the Ninja project [87], an automated service composition system for heterogeneous devices and networks that deals with fault-recovery issues. The path creation component discusses a set of services for creating and executing a composite service. The discovery of a composite service entails that the path creation

component creates a service path by searching over a service graph, using a shortest path approach. It then creates a physical service path to locate the desired service instances, which are used to execute and monitor a composite service. Our preliminary work is more consistent with this work, where it uses non bio-inspired traditional algorithms (e.g. Dijkstra or bellman-ford based shortest path algorithm) for the path selection component.

Due to a) the rich semantics of multimedia content; b) the complexity and dynamic characteristics (e.g. synchronization and continuous flow of stream) of multimedia applications; and c) the QoS demand of the user as well as multimedia content itself, it is not easy to directly apply some of the existing solutions in web services composition to the multimedia domain [71], [84]. Our proposed work differs from the above works in that it focuses the use of a bio-inspired approach by leveraging SOA potential to mitigate the service management issues for the multimedia composite service.

2.2.2 SOA for Multimedia Service and Service Management

Service Oriented Architecture (SOA) is based on the realization of SOC (Service Oriented Computing). It is gaining significant attention in industry, science, and the academic world because of its advantages, which include scalability and the flexible composition of its services, which range from telecommunication to multimedia. SOA has been applied in different application domains, including mobile services [100], grid services for multimedia streaming in e-learning environments [6], optical network services [113], pervasive and ubiquitous environments [70], web applications [3], service composition [69] and service management [68], [91]. However, little is known about the use of SOA in the multimedia application domain, other than the work mentioned in [6], [121]. The challenges of SOA for multimedia service composition has been addressed in [85].

SOA can provide a flexible architecture that combines multimedia service composition processes by modularizing large multimedia applications into services. It allows for new services to be added into the system without stopping existing multimedia applications. It can solve numerous distributed multimedia computing challenges by enabling a user from any device, using any operating system, in any programming language, protocol, multimedia format and platform, to access a SOA service to create a new desired composite multimedia service that is customized for the user. In SOA, software resources

are regarded as a service, which provides a specific functionality in order to accomplish a composite task (business process). Services and functionality in SOA [90] are defined by a web services standard, such as the Web Services Description Language (WSDL) [19], Simple Object Access Protocol (SOAP) [45], and Universal Description, Discovery and Integration registry (UDDI) [20]. However, the above-mentioned underlying technologies often result in poor web service performance [124] and are overwhelmingly degraded if the web service is used for multimedia service. For example, communication between web services is established through XML-based SOAP messaging, which can negatively affect the performance of web services regarding different parameters related to transmission, processing, and message parsing [126] .

In [121], Wu et al. proposed a Global Multimedia Collaboration System based on grid computing, which a parallelization of applications and a systematic use of idle resources in the overall network environment. The authors attempted to integrate the SOA paradigm with grid computing, where XML based General Session Protocol (XGSP) and SOAP are used for creating and controlling VoIP conferencing services. Using a XGSP schema, a Global Multimedia Collaboration System (Global-MMCS) was developed, which can support and integrate services such as videoconferencing, instant messaging, and streaming. Audio streaming was extensively used, video transcoding was used to a limited extent, and scalability was tested using a text chat session.

In [23], two models of service composition in SOAs are presented: process-oriented composition and structural composition. The former model combines services using a workflow model to define service component and the Business Process Execution Language (BPEL) specification is used for this composition model. The latter model identifies the participating service components and the component connections that represent service component interaction channels. The SCA (Service Component Architecture) specification, which addresses the policy aspects of service composition, is used for the structural composition model. However, those models are not suitable for multimedia service composition, as BPEL [72] is static composition and is not flexible [124]. Any change to any component of a web service requires changes to the entire composition process.

In [6] Amoretti et al. presented an ongoing work for campus-wide service-based multimedia content access and distribution in an e-learning environment. The grid technologies were used to search and access the multimedia content (e.g multimedia resources related to courses) from several video servers through the use of services like multimedia discovery service, search service, video index service, bandwidth broker service and monitoring service.

In [68], Jin and Nahrsted proposed a QoS-aware service management framework where scalability was evaluated in terms of network size and the client application's population size. This approach used a Dijkstra-based algorithm for service path computation and service composition. The service management includes the collection of QoS (e.g. delay and bandwidth), as well as functional information, composition (path finding) and service maintenance (e.g. resource adaptation and failure recovery). In this work, the potential of the SOA model is envisioned to be used for content customization and for solving heterogeneity problems in large as well as open networks. Our preliminary work [56] is more consistent with Jin and Nahrsted's approach, where a Dijkstra based algorithm was used for service path computation, as well as for service composition. However, the approach with the Dijkstra-based solution may be inefficient for large networks and for a dynamically changing, ubiquitous environment where new multimedia services appear every day, and existing services disappear at any time.

In our proposed work [53], we bring SOA with a biologically inspired technique to a multimedia service composition, which allows for more scalable and robust multimedia delivery. We get the desired composite multimedia service through the composition of basic multimedia services, such as streaming services and a variety of repurposing services. We have used the biologically-inspired approach to collect the QoS requirements from individual repurposing services, in order to select the most suitable services for the target service composition. For the composite service maintenance step, we have demonstrated load management and robustness, instead of failure recovery, which has been stated in [68].

2.2.3 Bio-Inspired Approach for Service Selection

In previous service selection studies, most researchers [112], [125] have used traditional algorithms such as the Dijkstra, Bell-man ford, heuristic linear programming and dynamic programming algorithms, for service selection. Existing solutions [15], [42], [125] for the service selection problem based on the above mentioned algorithms have some limitations. Using linear programming and dynamic programming algorithms, we may experience a theoretically optimal solution; however, those algorithms are too complex for runtime decisions. Dijkstra-based [56], [122] algorithms are widely used; however, they may not be cost effective for large and dynamic networks. Consequently, we have to look at alternate efficient solutions for the multimedia repurposing service selection issue.

Some researchers [51], [56], [122] focus on multimedia service selection for service composition. In [122] Gu and Nahrstedt, use the Dijkstra-like algorithm for selecting service in their multimedia service composition research, where link delay and link availability are used as QoS parameters. In [14] Canfora et al. use the Genetic Algorithm for service selection, prior to service composition. They also compare their approach with the integer programming approach. However, genetic algorithm [15], [109] for service selection is also unsuitable for multimedia service selection, because of its centralized optimization character.

We believe that selecting multimedia services by using the nature-inspired algorithm is one of the new ways of service selection, as the nature-inspired algorithm has the ability to adapt to a constantly changing environment, like wireless networks and clients [11]. To overcome challenges related to the use of traditional as well as a genetic algorithm for service selection, we use an ant-based [25] QoS-aware selection approach for multimedia repurposing service selection [54]-[55], which is distributed and demonstrates robustness to network changes and service uncertainty.

To the best our knowledge, there is little work related to the application of Ant Colony Optimization (ACO) in a multimedia content repurposing service selection, other than some applications in image processing such as image coding methods based on color image quantization [58], and multimedia traffic through MANET through the use of reinforcement learning techniques [127]. Since ant-based methods found in the literature are not directly applicable to the repurposing service selection problem in the multimedia

domain, the remainder of this review section lies on data routing. The ant colony algorithm has been applied to the routing of data packets in different modern networks. Among those, AntNet for data routing [25], AntHocNet for MANET [27] and the biologically inspired routing for MANET [77] are worth mentioning. Some of the existing related works related to Ant-based QoS routing or selection and their respective QoS parameter are summarized below:

In [89], Oida and Sekido proposed ant based QoS routing or selection, where bandwidth and hop count were used as QoS constraints, while in [17], Carrillo et al. described an ant based QoS aware best path selection solution that was based on the delay. In [77], Liu et al. presented ant based lightweight QoS provisioning, such as transmission delay related to service classification traffic control. In [111], Tadrus and Bai explained an AntNet based algorithm for QoS routing, where bandwidth was considered as QoS constraints. Conversely, Subing and Zemin [108], proposed an algorithm that considered multiple metrics, such as loss rate and bandwidth as QoS parameters.

Our method is consistent with the approach proposed by AVANTEL project, Advanced Multimedia in Group Organized Services (AMIGOS) [120]. AMIGOS presents a swarm intelligence based algorithm, which enables the implementation of service lookup and access. The algorithm is implemented using simple ant-like agents, which are able to search for near optimal paths of resources in a large network environment. It uses Rubinstein's work on cross-entropy and combinatorial optimization [99], while our approach extends the idea from Di Carro and Dorigo's work on data routing [25]. Additionally, in their approaches [120], QoS is only addressed to a limited extent.

The above-mentioned algorithms consider mostly network level QoS. Our algorithm [54]-[55], however, considers both application level QoS such as frame rate and resolution, and network level QoS such as delay. In the proposed bio-inspired selection algorithm, an ant uses the QoS metric instead of trip times to select repurposing services or the best repurposing path, where QoS is considered in terms of the frame rate of the desired multimedia service. This required frame rate uses dynamic calculations based on bandwidth, desired format, and user preference.

2.2.4 Bio-Inspired Approach for Service Management

Recently, interdisciplinary research has been gaining attention in the research community. Specifically, inspiration from the metaphor of a bio-inspired concept towards a multimedia system [52], [55], distributed system [9], service composition [32], [82], service management [18], different communication network services and systems [12], [16], and network application adaptation [110]. The reason behind such choice is that: a) bio-inspired system has the natural capability of self-healing and robustness in dynamic and uncertain environments; b) it is naturally more distributed and heterogeneous than a ubiquitous network environment; and c) it shows the scalability as well as adaptability to the highly distributed, large scale, and dynamically changing ubiquitous network environment.

Y. Ding et al. [32] uses bio-entity inspired from the immune behaviour of a neuro-endocrine-immune (NEI) system for web service composition and management. The system is regarded as a web service emergent system (WSES), where service composition is developed through the emergence of web service. In WSES, in response to a web request, the bio-entities set up the new composite web service through a negotiation of the bio-entities. The system was evaluated in terms of energy consumption, response time and adaptability. While the system discusses scalability, it does not demonstrate any results or a model that is related to scalability, in terms of services, resources or networks.

Musunoori and Horn's [82] research describes an ant-based service partitioning for grid service configuration as a service composition. The approach proposed is actually concerned with a service-partitioning problem for a computational grid environment. The approach may consequently require unnecessary state information of the network, as each source establishes its own "grid" across the entire network. With the increase of the sources, overhead of the "grid" information also increases, which may cause a scalability problem. Our approach uses the SOA paradigm, as well as the ant based service selection approach to overcome the scalability issue.

Suzuki [110] describes and proposes a biologically inspired adaptation mechanism for the adaptation of network applications in the NetSphere architecture, iNet [73]. In this architecture, the biological system is inspired from a bee colony metaphor, where bee agents mimic functional services that are related to network applications, in order to

autonomously: (a) monitor their surrounding environmental conditions (e.g., traffic); and (b) accommodate their behaviour and customize it for the current environmental conditions. However, our work is different from [110] in selecting multimedia services and using it in a multimedia system. Specifically, our works is inspired from an ant colony metaphor, while the work in [110], is inspired from a bee colony metaphor [119].

In [18], Chiang et al. propose a bio-inspired framework for service management in a ubiquitous computing environment. An Ant Colony Optimization (ACO) meta-heuristics is adopted for the configuration of this network service component. As proof of their concept, they simulate the service configuration process (composition plan) for a multimedia messaging service, which is actually a simple e-mail application.

In [55] authors focus on a biologically-inspired ant colony's foraging behaviour in multimedia content repurposing, where authors make use of a biologically inspired service selection algorithm for selecting suitable repurposing services for heterogeneous network environments that are based on QoS. Our framework is somewhat consistent with [18] in that we use the ACO-based approach. However, our work is distinguished in numerous ways. First, we adopted an AntNet service selection strategy in the context of real time multimedia streaming for ubiquitous users. Second, in order to validate the above process, we implemented the proposed management framework for live multimedia streaming. Third, through the use of simulation, we also present some service management issues like load management and scalability. We also distinguish our proposed framework from the previous work [55] in that we use the SOA paradigm's potential and actual promises in order to fulfill the distributed ubiquitous access of multimedia service.

2.3. Problem Formulations

As described in the previous chapter, the main focus of the proposed multimedia service management framework is the repurposing service selection or path computation.

To find or select an optimal repurposing path $P_R(s,d)$ between the multimedia system sender and receiver, we consider a directed acyclic graph (DAG), $G = (V,E)$, where V represents a set of proxy nodes $V = \{v_1, v_2 \cdots v_x\}$, and E represents a set of links between service nodes $E = \{e_1, e_2, \cdots e_y\}$. Each node v_i provides a set of n_i repur-

posing services $R_{v_i} = \{r_1, r_2, \cdots r_{n_i}\}$. The cost associated with each link ($e \in E$) on each path of the repurposing service network is expressed as a positive weight, whose value is between 0 and 1. The cost or weight refers to the satisfaction of the user. The user's satisfaction metric (s_i) is used to quantify the application level QoS. The application level QoS has parameters like frame rate, resolution, and signal to noise ratio (SNR), which is represented by $q_i \in \{q_1, q_2, \cdots q_m\}$. As described in (1), s_i has a minimum acceptable (L) value of q_i below which the satisfaction is negligible and a maximum value (M) of q_i, above which the satisfaction s_i is approximated to 1.

$$s_i = \begin{cases} 0 & Unsatisfactory/negiligible & q < L \\ 1 & No\ change\ in\ satisfaction & q > M \\ (q-L)/(M-L) & Acceptable\ satisfaction & L \le q \le M \end{cases} \cdots\cdots (1)$$

To find an optimal repurposing path $P_R(s, d)$ from a server (s) node to a receiver node (d) via one or more intermediate repurposing service nodes that maximizes user satisfaction subject to QoS constraints, we used the following component satisfaction score (2) for each link for a service path:

$$S_e = \sum_{i=1}^{n} w_i s_i \cdots\cdots (2)$$

Where, each component satisfaction s_i is a function of application level QoS parameter, i.e. $s_i = f_i^e(q_i)$ and $\mathbf{w_i}$ is the weight for each QoS parameter set by a user.

For more than one multimedia application level QoS parameter, the overall user satisfaction [97] is determined as a combination function of the individual component satisfaction s_i that is presented in the following equation:

$$S_{comb} = f(s_1, s_2, \cdots s_m) = f\left(f_1^e(q_1), f_2^e(q_2) \cdots f_m^e(q_m)\right)$$
$$= \frac{m}{\sum_{i=1}^{m} 1/s_i}, 0 \le s_i \le 1 \cdots\cdots (3)$$

Where, each satisfaction s_1, s_2, $\cdots$ s_m component is based on previously mentioned application level QoS parameters. In (3) the overall satisfaction function is low if one individual satisfaction function component is low. For instance, if a multimedia stream has a very high resolution but plays back at 5 fps, then the combination satisfaction is low.

Now the problem is to find a service path $P_R(s,d)$ from a sender (s) to a receiver (d) via one or more intermediate repurposing service nodes which maximizes the total user's satisfaction (4) subject to QoS constraints (5) and (6): So the problem can be expressed as:

$$maximize \sum_{e \in P} S_e \cdots\cdots (4)$$

$$Subject\ to$$

$$\sum_{e \in P} q_D^e\, \delta_e \leq Q_D \cdots\cdots (5)$$

$$q_{\beta_\gamma}^e\, \delta_e \leq Q_{\beta_a} \cdots\cdots (6)$$

Where, q_D^e and $q_{\beta_\gamma}^e$ are the QoS attributes or requirements for delay and bandwidth, respectively; Q_D and Q_{β_a} are the QoS constraints or resource constraints with regard to delay and bandwidth respectively. δ_e represents the decision variable that indicates whether the service link is selected, as described in (7). According to the constraint (5), total cumulative delay (QoS attributes) over the selected repurposing path must be less than the overall constraints, Q_D. Based on the constraint (6), the required bandwidth in each link is less than the available bandwidth of that link.

$$\delta_e = \begin{cases} 1 & e \epsilon P \\ 0 & e \notin P \end{cases} \quad \begin{matrix} if\ the\ service\ is\ selected\ or\ the\ link\ on\ the\ path \\ otherwise \end{matrix} \quad \cdots\cdots (7)$$

Finding the optimal service path and server selection that is based on multiple constraints can be regarded as a restricted shortest path problem or a bottleneck Traveling Salesman Problem. These problems have already been proven as NP-hard [43].

In order to solve the problem we first attempted to apply traditional (non bio-inspired) selection algorithm. We then applied the ant colony metaphor for selecting the best repurposing service route or path. In order to select a repurposing service among proxies, we adopted AntNet algorithms, where the ant controller packet was responsible for sampling and generating the repurposing path among the repurposing services that ran on intermediate proxies.

Chapter 3. The SOA-Based Framework for Multimedia Service Management

In this chapter, we present the proposed service management framework. We first describe the information or profiles required for the framework. Based on this information (e.g. profile), the system will select devices and appropriate QoS parameters in order to maximize user satisfaction. This information is then used by the service selection subsystem. We then illustrate the system architecture of the framework and the details concerning the participating subsystems and services.

3.1. Information Requirements for Service Selection

Effective management of multimedia composite service is very much required in order to provide a reliable and efficient multimedia content delivery. The success of this multimedia management depends not only on the selection of suitable repurposing services, but also on the effective organization and/or information maintenance involved prior to the selection. This information facilitates the construction of the service graph and is eventually used to select an appropriate service path. The more information collected concerning users, services and their surrounding environment, the more customized content can be delivered in such a way that is acceptable (e.g. higher frame rate, better resolution, better quality) for the user. Such information may be captured and stored in different profiles, which include, but are not limited to: user environment profiles (e.g. user, terminal, and network profiles), content profiles and repurposing service profiles. As previously mentioned, these profiles play an important role in determining the context of the usage environment and, hence, assisting in the selection of an appropriate repurposing service that will ensure ubiquitous and universal multimedia distribution and delivery. These profiles are briefly discussed in the following:

3.1.1 User Environment Profile

The user environment profile describes the context surrounding the user. Context information can be characterized by the situation of an entity, where an entity can be a person, a place, or a computational or media object, which is considered relevant to the interaction between a user and an application [24]. These user environment profiles fall into the following categories, as described in the natural environment characteristics tool of MPEG-21 [63], [115] .

- User profile: The user profile describes the personal properties and preferences of a user, which includes audio-video qualities such as frame rate and resolution. It also describes the user's content preferences, presentation preferences, accessibility, mobility and destination. User characteristic tools of MPEG-21 describe such profiles in detail.
- Terminal profile: This profile describes the codec capabilities, device capabilities and available input/output characteristics. The terminals could be a server, client, or proxies throughout the repurposing path. In addition to general capabilities, the server has the information about the neighbouring proxies, which are connected to the server. Proxies also contain the information about neighbouring connected proxies, as well as the available repurposing services that run on each proxy (described in Appendix D). Based on these profiles, a server can identify whether the client is capable of decoding the content, or whether the content requires further conversion in order to be rendered by the requesting client. The terminal capability tools of MPEG-21 [63] describe the above terminal capabilities in detail.
- Network profile: The network profile is important for dynamically repurposing the multimedia content, based on varying network capabilities and conditions. This profile describes maximum network bandwidth, available bandwidth, delay, and error. The network characteristic tools of MPEG-21 [63] also describe these characteristics.

3.1.2 Repurposing Service Profile

This profile contains a copy of the list of inputs, Q_{in} (e.g., different formats, standards) that the repurposing service can accept, as well as a list of outputs, Q_{out} (e.g., formats, standards, etc.) that the repurposing service can create by repurposing one of the inputs.

Again, for simplification, it is assumed that any of the inputs can be repurposed into any of the outputs. Both inputs and outputs are described using the MPEG-7 [81] content descriptors. For example, the repurposing service can send MPEG-4 [62] with a resolution of 352 × 240, and a frame rate of 30 fps as input, and it can receive H.263 [64] with a resolution of 88 × 72 and a frame rate of 10 fps as output. Additionally, this profile may contain a link back to the description of the proxy from where the current repurposing service is running. The example of a repurposing service profile is shown in appendix E.

3.1.3 Content Profile

The content profile is a description of the content that a sender can deliver. It includes the metadata or description of different media types (such as audio, video) and their variations. Some of these metadata include information about transport protocol, coding standards (e.g. H.263 video, MPEG-4 video, M-JPEG) and other identifying parameters. This metadata description follows the MPEG-7 standard [81]. The system uses the content profile along with other profiles to repurpose the multimedia content for client devices.

3.2. System Architecture for the SOA-Based Proposed Multimedia Service Management Framework

In addition to the client, the proposed multimedia service management system consists of three main subsystems, which represent a variety of services that are depicted in Fig. 3-1. The services are comprised of the registry service, the streaming service and the repurposing service. Each service has a different job or functionality to provide to its requestor and can guarantee the flexibility, scalability and extendibility of the architecture to include more services, if required.

The client first queries the registry to find out a streaming service. Then, it sends a request to the streaming service with specific quality of service (QoS) parameters. The streaming service originates or creates the multimedia stream. The originated stream can either be captured in real-time by using a web cam, for example, or it can be an already stored content which the streaming server retrieves. The repurposing service transforms the media stream from one media format to another, according to the request. In addition to the incoming stream, the repurposing service should accept a file from the requester.

This file corresponds to a composite repurposing service that is comprised of multiple calls for primitive repurposing services, in order to produce the final content as a reply to the client request.

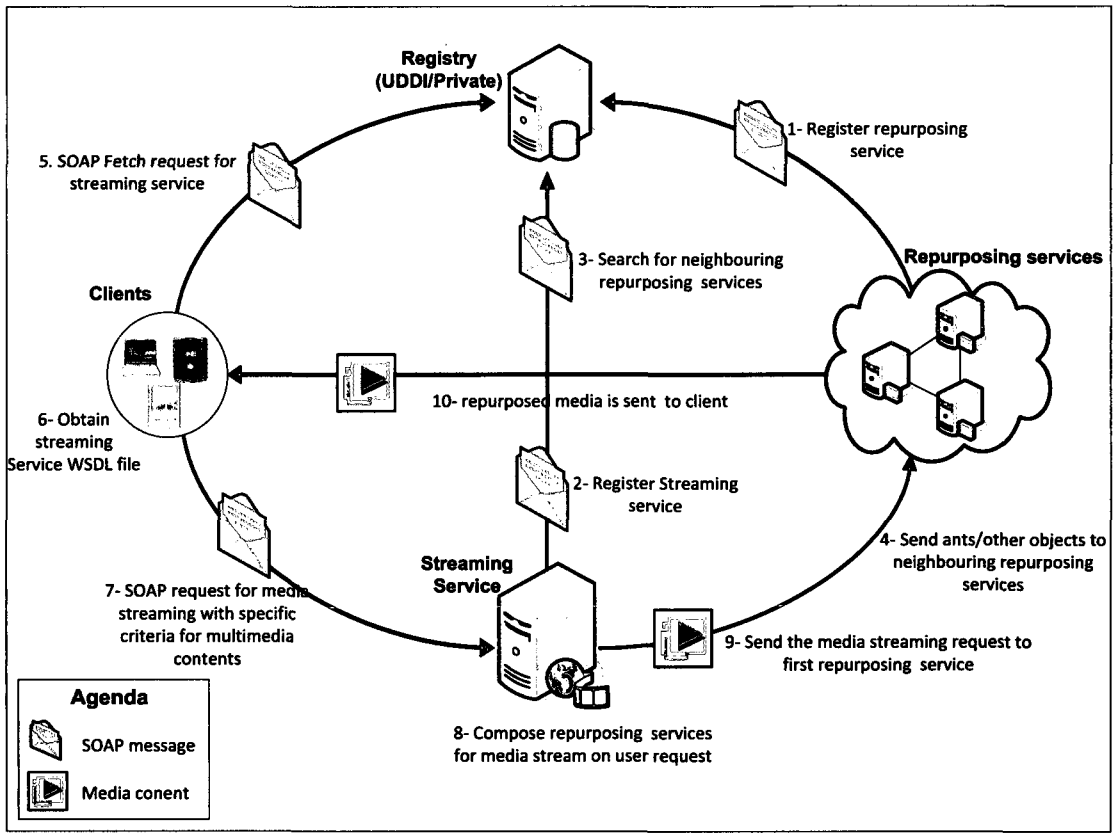

Figure 3-1 The SOA-based multimedia management system scenario

Each repurposing service forwards its output stream to the next subsequent service based on the composite service plan, until the final service, which forwards its output directly to the client. The framework relies on a registry, which is used for service query purposes. The registry is like a database that contains identification information about the different services in the architecture. After the creation of each service, the first step entails registering the service in the registry using the correct keywords and under the proper classification that clearly describes the service. Multiple repurposing services can reside in the network host node, which is usually referred to as a proxy server.

When the client requires a media stream, it first searches a streaming service from the registry. The fetch request returns the Web Service Description Language (WSDL) file [19], as described in appendices A and B. The streaming service that geographically

resides in the nearest point to the client is returned as a reply to the client fetch request. This is significant in order to minimize network resource consumption. Using the WSDL file, the client generates a request to the streaming service with Simple Object Access Protocol (SOAP) [45] and sends it to the service. The request consists of a call to the streaming service, with parameters such as the format of the stream and the client location.

Upon receiving a request, the streaming service uses already gathered information by the service selection subsystem to create a composite repurposing service that will satisfy the client request. In the case of bio-inspired selection, the service selection subsystem works independently within each streaming service to dispatch periodically bio-inspired agents (e.g. ant) that are based on the information available at the registry to collect information about repurposing services. The streaming service then utilizes this information through a service composition system to create a composite service as previously described.

When the composite repurposing service is created, the streaming service sends a repurposing request to the first repurposing service in the composite file, followed by the media stream itself. The repurposing service then transforms the stream to another form of media stream and forwards it to the next repurposing service, preceded by the composite service file. Before sending the composite service file, each repurposing file eliminates itself from the file to ensure the sequence of execution. When the last repurposing service in the XML file list is reached, the repurposing service directly sends its output to the client as the result of its request. In the following sections, we will describe each of the services in more details.

3.2.1 The Streaming Service

The streaming service is one of the two main web service types in this system that composes the SOA architecture. As shown in Fig. 3-2, the streaming service consists of three subsystems: the service selection subsystem, the composition subsystem and the execution subsystem.

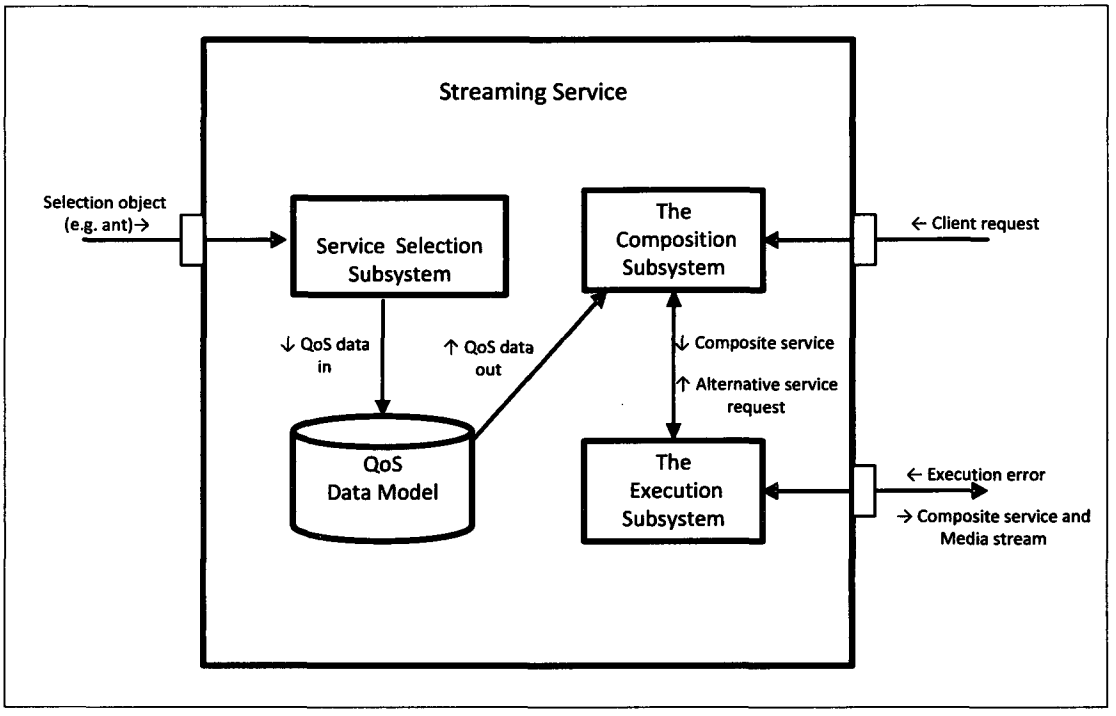

Figure 3-2. A decomposed view of the streaming service

The Service Selection Subsystem

The selection subsystem is an independent system that resides at each streaming service in this SOA-based architecture. The service selection subsystem may be based on either traditional service selection mechanisms discussed in Chapter 4 or bio-inspired service selection mechanisms discussed in Chapter 5. As our main focus is bio-inspired mechanism, we briefly discuss the bio-inspired service approach here.

The bio-inspired service selection subsystem is motivated from the foraging behaviour of an ant colony metaphor [25] of AntNet. Using the main registry of the architecture, this system discovers other repurposing services and periodically dispatches ants to these services. Through this subsystem, the streaming service is able to discover and collect QoS information about other repurposing services by dispatching ants, called forward ants, towards a targeted repurposing service. The collected QoS data is used to facilitate service discovery and selection, and is eventually used to optimize the composed process. In order to accomplish its task, the ant should know neighbour repurposing services, which can be achieved by inquiring the registry. When a targeted repurposing service fails to find other neighbour repurposing services, the forward ant dies and a back-

ward ant is created and sent back to toward the original streaming service. Upon receiving the backward ant at the original streaming service, the system strips the ant object and stores its collected data in a separate QoS data Model to be used by the composition subsystem.

The Composition Subsystem

The composition subsystem builds up a composition plan of multiple repurposing services to satisfy a client query. This subsystem accepts a client request of a media stream with a specific format and size. It then un-marshals the request and identifies the request's criteria that will be used to create the composition plan. The plan contains all the information needed to identify the repurposing services that will be used for the client.

The system then uses the available information in the QoS Data Model to determine possible composition plans that satisfy the client's request. In case more than one plan is found, the subsystem will select the plan with the highest QoS score. The composition plan is a file that plans the sequence of execution for the media repurposing services until the final repurposed media streams reach the client. It is comparable to the service configuration stated in [123]. Some of the examples of this composition plan are as follows:

$$SC_n: MS \rightarrow RS_1 \cdots RS_n \rightarrow Clients(\ e.g. PDA, CellPhone, Laptop)$$

For example,

$$SC_1: MS \rightarrow RS_1 \cdots RS_n \rightarrow CellPhone$$

$$SC_2: MS \rightarrow RS_1 \rightarrow RS_2 \rightarrow RS_3 \cdots RS_n \rightarrow Desktop$$

$$SC_3: MS \rightarrow RS_1 \rightarrow RS_2 \cdots RS_n \rightarrow PDA$$

$$SC_4: MS \rightarrow RS_3 \rightarrow RS_5 \rightarrow PDA.$$

Where MS represents a multimedia streaming service, RS represents a repurposing Service and SC represents a service composition plan. Once the composition plan is created and finalized, the subsystem hands the composite service to the execution subsystem.

The Execution Subsystem

The execution subsystem runs the composition service plan. Upon receiving the plan, this subsystem establishes and handles a communication link with the first repurposing ser-

vice in the plan. It then initiates communication by sending the first composite service to that service, followed by the media stream. Once the media stream ends, the subsystems close the connection channel with the remote service. During communication, the subsystem monitors and handles any exceptions or errors that occur in the execution. In the case of execution failure, the system requests an alternative composition service plan from the composition subsystem, which it attempts to execute.

3.2.2 The Repurposing Service

Fig. 3-3 shows an architectural view of the main subsystems that constitute the repurposing service: the dispatcher subsystem and the repurposing subsystem. Both subsystems are further explained in the following:

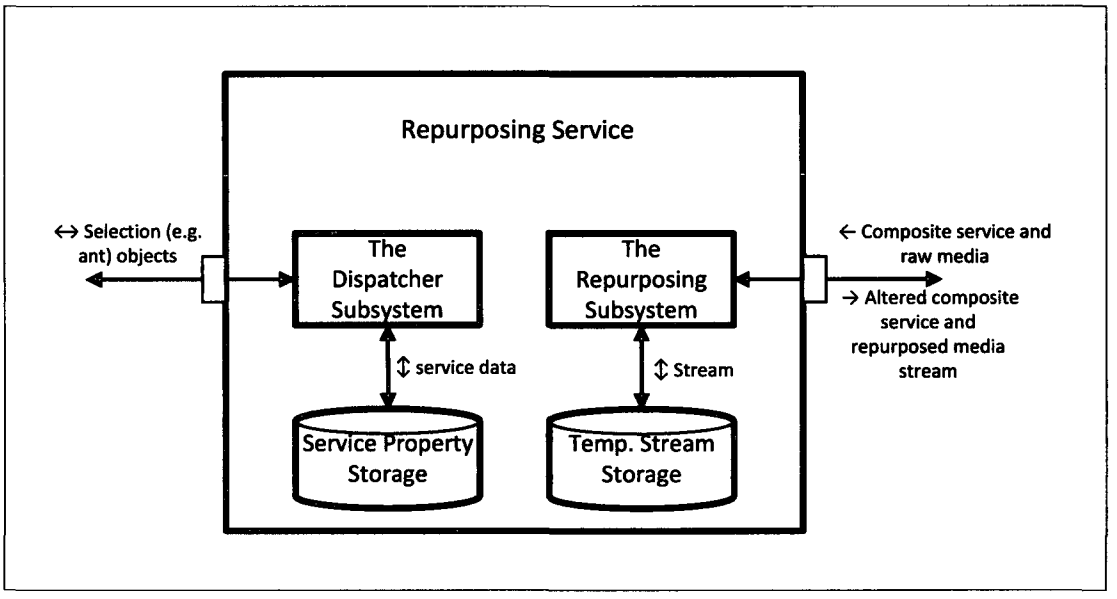

Figure 3-3. A decomposed view of the repurposing service

Dispatcher Subsystem

This subsystem is mainly used for the bio-inspired approach. The subsystem represents the interface for the repurposing service that predominantly handles the functionality of the ant objects. When an ant is dispatched by the streaming service, it is usually sent as a streamed object toward the targeted repurposing service. This subsystem is responsible for first accepting incoming ant objects and then forwarding them toward their final destinations. Simply, the subsystem accepts the incoming ant object, performs some testing

to recognize the ant's final destination node and then forces the ant to jump to the next node, if the current node is not the final destination. To determine its final destination, the ant should be aware of other neighbour repurposing services, which can be achieved by inquiring the registry. It additionally allows ant objects to access the repurposing service properties that are usually fixed and kept in permanent storage.

In addition to the above, this subsystem also acts as a load balancer, which is invisible to the users, as they do not need to know the service from which they can be served. This subsystem monitors the availability of appropriate services and dispatches the incoming multimedia composition request to the proxy server where repurposing services are running. It facilitates the selection of appropriate repurposing services and distributes the media server load based on the QoS collected by the service selection subsystem.

Repurposing Subsystem

The Repurposing subsystem is responsible for receiving a raw media stream and then converting it into another media stream format. This subsystem accepts a composite service plan, followed by a raw media stream. Upon receiving the request, it first updates the composite service by removing the entry in the plan corresponding to its service node. The repurposing subsystem then initiates a connection to the next repurposing service in the plan and sends the composite service first to that remote repurposing service. Once it starts receiving the raw media stream, the composite service initiates the conversion process and sends the converted stream immediately to the remote subsequent service using the previously established connection. In order to begin the conversion process, the subsystem utilizes temporary storage to store the converted stream before the sending process begins.

The details of repurposing subsystem are described in Fig. 3-4. As shown in Fig. 3-4, the ConfControlerGUI class is a controller class that triggers the execution of the subsystem. When an object of this class is constructed, it creates two other objects: the SDPHandler and the ConfAgentGUI. The ConfControlerGUI extracts the host IP address of its node.

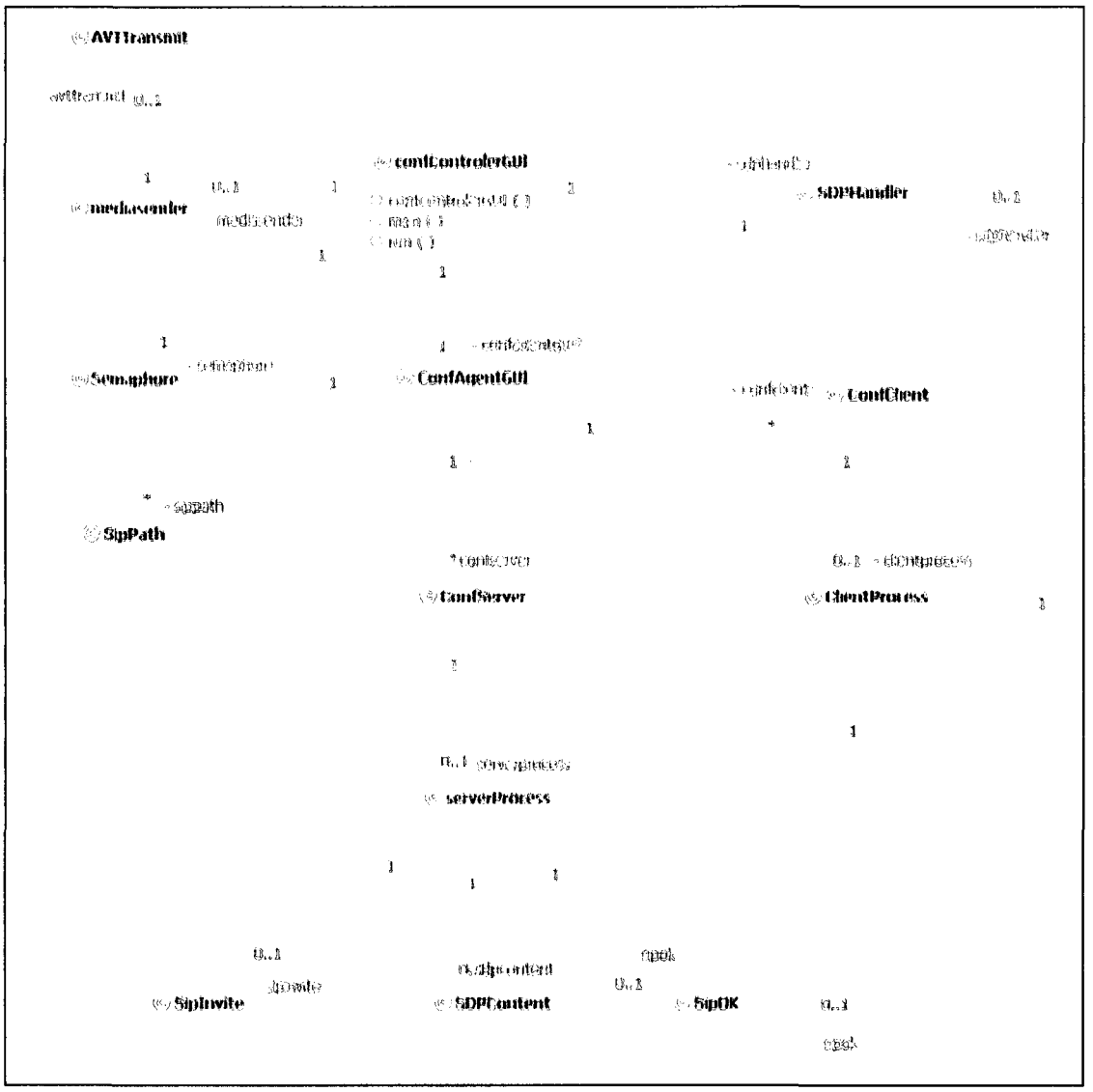

Figure 3-4. Subsystem view of the repurposing engine

When its thread of execution starts, it also starts the thread of execution for the ConfAgentGUI object and creates an object, the MediaSender. Once the repurposing path for the requested media has been created and the service is in the state of sending a stream, the ConfControlerGUI checks the next remote repurposing node to see whether it is ready for receiving the media. If the next remote repurposing node is ready, the ConfControlerGUI sends a message to the MediaSender object that contains addressing information for the source and destination nodes in the repurposed media stream. If the next remote repurposing node is not ready, it checks whether the service is receiving a

media message. After that, it constructs a new ServerProcess object to handle that request. The MediaSender object listens to streaming requests using a specific port, which is already configured. Once it receives a request, it simply extracts the IP address and the port number for the target-streaming node, and creates the AVTransmit object that repurposes and sends the stream (either live or pre-encoded) by using the RTP protocol to that target node.

When the ConfAgentGUI object is constructed, it initializes and starts the GUI of the repurposing service and handles its events. It additionally creates two objects, the ConfClient and the ConfServer, and begins their execution threads. It further creates and initiates the AntController, in the case of bio-inspired service selection. The AntController is used by the ServerProcess to identify the best node for the next jump toward the final destination node in the media stream.

The ConfServer mainly filters incoming requests from a remote node, because new calls are not accepted while the service is in the process of sending a media stream. If there is no a service in progress, then the ConfServer passes the request to a ServerProcess object for execution and returns the control to the ConfAgent. The ServerProcess sends the SIP invitation to the remote node by creating two other objects: the SDPContent and the SIPInvite. It then blocks the stream until it receives a SipOK response from the remote node. If the SIP session is established successfully, then it asks the ConfAgentGUI object to set the receiver of the stream. During the receiver setup, the Java Media Framework (JMF) is used to create a stream listener from the supplied addressing information. The JMF is then used to initiate an RTP streaming session for Video and Audio.

When a message arrives at the local SIP server socket, the ConfClient accepts it. Upon receiving the message, it dispatches a ClientProcess object to process the message. The ClientProcess interprets the call request. If it is a new call and the service is not sending or receiving any media, the ClientProcess establishes a new connection and sends a SipOK as a response, and then waits for the reply. Upon receiving the reply, the ClientProcess sets up a service stream receiver based on the reply information. If it is a new call and the service has already sent or received a stream from the caller node then it sends only a SipOK message to that remote node.

3.2.3 The Client Subsystem

The client is a viewer that enables the user to select and view a media stream of their choice, based on what is available in the registry. The end-user will only have access to the client and the client will only display what is available in the registry. More specifically, the client will only allow the user to select a media stream of a particular format and resolution, given the registry states that a particular media stream is available and that it can be viewed at a particular resolution in a particular format. The client is designed to be lightweight and can thus be installed on any machine to enable it to view a chosen stream

3.2.4 The Registry Subsystem

The registry is a permanent storage where the available media services, streaming services and available repurposing services register themselves in order to be discovered by other services. The streaming services and repurposing services are responsible for registering information about the video (e.g. QoS, format, and resolution) and they are able to output information pertaining to their particular "contact" information (i.e. their address). The registry maintains a reference to all the available media (e.g. video) streams, proxy nodes, and repurposing services.

3.3. Summary

This chapter describes the system architecture of the service management framework. We first briefly describe the information (e.g. profile) requirement for the framework. We then introduce the participating key services and subsystems in the proposed framework, including the streaming service, repurposing service, registry service, service selection subsystem, composition subsystem and execution subsystem. We also discuss their responsibilities, their required information (profiles) and their interactions.

Chapter 4. The Non Bio-inspired Service Selection Algorithm

This chapter proposes a solution that uses a series of repurposing services in a chain fashion to accomplish the composite repurposing task. To find the best repurposing service among these proxies, we implemented and compared the path selection algorithm. The selection algorithm finds the best repurposing rules to apply to the multimedia content, where the user's satisfaction is used as an optimization metric for the repurposing service selection. Finally, we present the performance comparison of the algorithm.

4.1. Optimization Metric Computation

This repurposing may take place either on the client-side, the server-side or on the proxy. Some client devices (e.g. PDA, cell phone) have some constraints regarding processing power and memory. Servers usually have the limitation of computational load and resource consumption of complex repurposing if the dynamic context of a user and its surrounding natural usage environment is considered. Thus, in our system, repurposing is performed in distributed resourceful proxies, in multiple steps, in order to reduce the computational load on the server. However, the distribution of repurposing tasks among the proxies requires special considerations such as selecting and composing the best repurposing services.

To this end, we used a repurposing service selection algorithm. The repurposing service selection algorithm is one of the decision makers that finds the best repurposing services to repurpose the content, in order to make it acceptable to the user. The repurposing service selection algorithm uses the satisfaction function as a metric to determine the best repurposing path. As described, we refer to this metric as s_i that has an aggregate value between 0 and 1, where 1 denotes the highest satisfaction of the user. The satisfaction metric is a function of m-tuple, which includes the user's preferences (audio quality, frame rate and resolution of video etc.). The function has already been defined in chapter

2. For simplicity, we will give a particular example of the calculation procedure for the satisfaction metric based on frame rate, which is used for the test results in section 4.5. In this particular case, the satisfaction metric is computed using (8) and (9).

$$\beta_\gamma = \varsigma \times \beta, Where\ \varsigma = \sqrt{\theta/_\vartheta} \cdots\cdots (8)$$

$$f = (\beta_a \times \beta_\gamma) \times f_{max} \cdots\cdots (9)$$

An example of how a satisfaction metric is calculated is listed below:

Example: The computation of the satisfaction metric, in terms of frame rate

As has been previously stated, the computation of a frame rate depends on the multimedia content resolution and the available bandwidth between neighbouring proxies. In order to calculate the user's satisfaction based on frame rate, we have selected a number of properties that can be used for each type of multimedia standard. For this reason, it is required to create a table that would store each coding standard with one frame resolution value and the bit rate (bandwidth) required to send 30 frames per second at that resolution. The selection algorithm computes the frame rate and satisfaction metric for video content using the bandwidth stored in the table and the available bandwidth between the two proxies. The bit rate in Table 1 represents the bandwidth necessary to obtain a frame rate of 30 frames per second.

Table 1. Stored frame resolution and required bit rate for different video content

Format	Width	Height	Bit Rate
Video/H.264	176	144	64
Video/H.263	352	288	64
Video/MPEG-2	320	240	4096

Let us compute the frame rate and user's satisfaction metric s_i of a MPEG-2 video stream with a height of 80 and width of 60 between two proxies with a connection that has a bandwidth of 512 kbps by using Table 1 and equation (9).

The multimedia repurposing system will first calculate the resolution of the stream:

$$Area\ \alpha = 80 \times 60 = 4800\ Pixel^2$$

The resolution of the stored values from the table for MPEG-2

$$Area\ \alpha = 320 \times 240 = 76800\ Pixel^2$$

Area factor

$$\varsigma = \sqrt{\frac{4800}{76800}} = 0.25$$

Using this value, the required bandwidth for 30 frames can be calculated using (8), which is

$\beta_\gamma = 0.25 \times 4096 = 1024\ kbps.$

As only 512 kbps are available, the frame rate is

$$f = (512 \times 1024) \times 30 = 15\ fps$$

Now the satisfaction metric can be calculated as follows:

$$s_i = {}^{15}/_{30} = 0.5$$

4.2. Repurposing Service Graph Creation and Simplification

Before applying service selection algorithm to application, service DAG (Directed acyclic graph) is created. The service is comprised by using user environment profiles (e.g. user, terminal, and network profiles), content profiles and repurposing service profiles, as discussed in chapter 3. We generate the service graph as follows: a) each node or vertex in the graph represents repurposing service. Repurposing service is defined in section 3.1.2. Each node has a number of input and output links. The input links specify possible input formats (Q_{in}) to the repurposing service and the output links indicate achievable output formats (Q_{out}) to the repurposing service, which is defined in the repurposing service profile in chapter 3. The input links for the sender are defined in the content profile, which includes all meta-data information (e.g. video type and format) of all the possible variants of the content. The input links of the receiver are defined in the device profile; b) The edges in the service graph represent the service network, which connects two nodes in such a way that the input link of one node (i.e.Q_{in}) matches the output link of another node (i.e. Q_{out}); c) Sender is connected to service node without an incoming link (i.e. sender should not have any incoming link). Receiver is connected to all service nodes

without an outgoing link (i.e. sink or receiver should not have any outgoing link); and d) The addition of the node constraints to its incoming link and the optimization metric (user's satisfaction) of every link, is calculated using (2).

To create the repurposing service graph, the system starts with services that the sender node (server) provides and connects the sender's output link with the neighbouring repurposing service nodes (vertices), such that neighbouring services have the same input format. This graph creation procedure continues until all repurposing service nodes have been considered and the client has been reached.

The simplification includes two steps: the entire path that does not lead to the receiver is removed and then the extra edges between the same two repurposing service nodes are removed by using the user satisfaction metric. The goal is to have any two repurposing service nodes be linked only by one direct link that will provide maximum satisfaction. This is required to ensure that the graph will be acyclic, with no infinite loop, while calculating the best route.

4.3. Non Bio-inspired Service Selection Algorithm

After computing the satisfaction metric, an acyclic repurposing graph is created and simplified to ensure that no infinite loop exists. After simplification, the next step is to find the series of repurposing services that provide user satisfaction to the receiver. This is done by conducting successive comparisons of edge/link costs, which, for this application, is the user satisfaction metric. Algorithms, which are used and compared to find the best service path, are the adapted versions of Dijkstra's shortest path algorithm and Bellman ford's algorithm. Because of the resource requirement as well as the QoS constraints, the existing shortest path selection algorithm cannot be directly applied to compute or select an optimal service path between two service nodes. We implemented, adopted and modified the Dijkstra-based path algorithm (named as RPS_1) to incorporate them into the multimedia repurposing service selection for optimizing the satisfaction metric. In the algorithm, the metric has been maximized instead of minimized. We then compared the RPS_1 algorithm with the Bellman-Ford [22] based selection algorithm RPS_2 algorithm as described in Appendix G.

In the algorithm described below, R_{V_i} is the set of repurposing services considered by the system and R_N is the set of candidate repurposing services. The set R_N is constructed with those repurposing services that are not in R_{V_i} but can be reached from R_{V_i} using one edge. Initially, the set R_{V_i} contains only the Media Sender node that holds the original media format, whereas the set R_N contains all other repurposing services in the graph that are connected to the Media Sender as well as the Receiver.

Algorithm 1: High level description of the non bio-inspired selection algorithm

Input : Media sender, repurposing services, user satisfaction
Output: Repurposing path

Let $R_{v_i} \in$ {Media sender}
Let $R_N \in$ downstream neighbour {Media sender}
Let R_i be the set of repurposing services
Let s_i^M be the user satisfaction of R_i
for *each* $R_i \in R_N$ **do**
    $s_M \leftarrow 0$
    for *each* $R_i \in R_N$ **do**
        if $s_i^M > s_M$ **then**
            $s_M \leftarrow s_i^M$
        end
    end
    Optimize(s_M, Q_{in}, Q_{out}, resource constraints)
    if $R_N = \emptyset$ **then** /* i.e. no more repurposing service to traverse */
        TERMINATE (FAILURE)
    end
    $MaxQoS \leftarrow 0$
    for *each* $R_i \in R_N$ **do**
        if $QoS(R_i) > MaxQoS$ **then**
            $MaxQoS \leftarrow QoS(R_i)$
            $R \leftarrow R_i$
        end
    end
    $R_N = R_N - \{R\}$
    $R_{v_i} = R_{v_i} \cup \{R\}$
    if $R ==$ *Receiver* **then**
        Print the repurposing path from the Media sender to Receiver
    end
    for *each* $R_j \in$ *downstream neighbour*$\{R\}$ **do**
        Optimize(s_M, Q_{in}, Q_{out}, resource constraints)
        $R_N = R_N \cup \{R_j\}$
    end
end
Print the repurposing path from the Media sender to Receiver

Figure 4-1. Non bio-inspired service selection algorithm

In each iteration, the algorithm selects the repurposing service R_i that generates the maximum user satisfaction(s_i^M) subject to the resource constraints. An example of resource constraints is described as bandwidth required ($q_{\beta_\gamma}^e \delta_e$) $\leq$ bandwidth available (Q_{β_a}), which is further explained in chapter 2. The user satisfaction is computed as an optimization function of the frame rate, frame size or the quality for the output format of R_i, which is subject to the QoS constraint of the available bandwidth between R_i and its ancestor repurposing services. R_i is then added to R_{v_i}. The R_N set is then updated with the connected neighbouring repurposing services, R_i. The algorithm stops when the R_N set is empty or when the Receiver node is added to R_{v_i}. The steps of the selection algorithm are listed in Fig. 4-1.

Let us illustrate the run time complexity of our algorithm. We assume, V_r and E_r are the nodes and the edges (links) respectively in a directed acyclic repurposing graph G(V_r, E_r). The satisfaction function is called once for each edge in the repurposing graph (line 7-31), for a total of $|E_r|$ calls. Lines 7-31 are repeated at most $|V_r|$ times, as (repurposing node) vertex $v_r \in V_r$ is added, at most, once to the set R_N Selecting the repurposing service with the maximum satisfaction takes O (V) time. Thus, the total run time complexity of our algorithm is O ($|V_r|^2 + |E_r|$).

4.4. System's Functionality with the Selection Algorithm

In the system with a non bio-inspired selection algorithm, the initial connection between the client and the server is established using the Session Initiation Protocol (SIP) [7]. As shown in Fig. 4-2, after getting the user or client request, the server sends a SIP "invite" message to the client containing the available ports and the IP address. The client replies with a 200 OK message, to which it adds its profile including the device's capabilities, the IP-address of the device and the supported codecs.

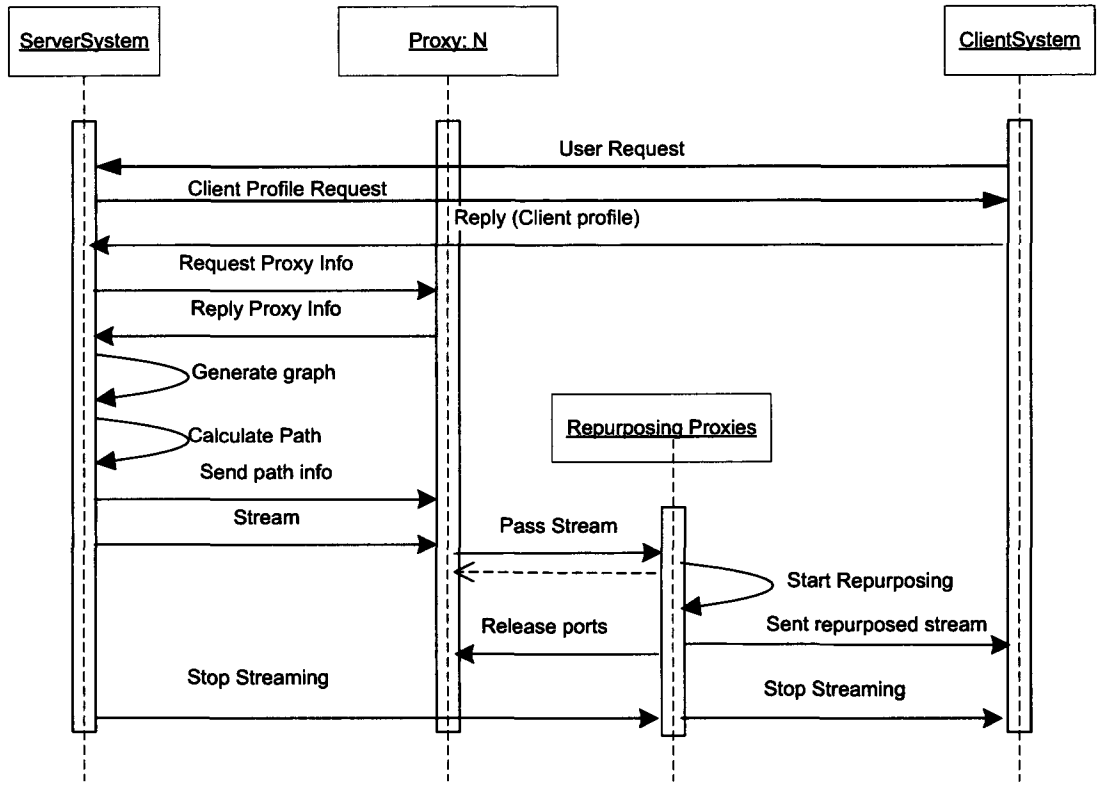

Figure 4-2. Overall repurposing system's functionality

Upon connection with the server, the server stores the client's profile. The server then connects to each of its neighbouring proxies. From each of the neighbouring proxies, the server requests the proxy's profile that includes one or more connected neighbouring proxies and the repurposing services that are running on each neighbouring proxy. Once all of the information from all of the proxies is received, the server generates a repurposing graph. It begins with the server-provided services and expands the graph until all the proxies and the required repurposing services have been considered and the client has been reached. The graph is then simplified in order to ensure that no infinite loop exists and that there is no backtracking. The graph is simplified in such a way that all paths lead to the receiver and there are no extra edges between the neighbouring repurposing services. After the graph simplification, the server discovers the repurposing path that provides the best quality (user satisfaction) to the receiver. The path discovery uses the proposed satisfaction metric in (2) and the traditional selection algorithm.

Once the server creates the best repurposing path, it starts streaming the captured and/or pre-encoded visual content, using the Real Time Protocol (RTP) [8]. The server

then sends the path information and the stream to the first proxy in the chain. Upon receiving both path and stream, the proxy creates one or more repurposing services. A stream can pass through multiple proxies and repurposing services and if required, can pass through the same proxy twice by using two different repurposing services. However, it cannot use the same proxy twice in a row in the repurposing chain. The proxy then sends the rest of the path and its own extracted information to the next proxy in the chain, such that the output formats of the previous repurposing service matches the input format of the next repurposing service in the chain. The last proxy in the chain simply forwards the results of its repurposed content to the client. Once the streaming through a repurposing service stops, the reception port of the proxy is freed and the repurposing object is destroyed.

4.5. Experimental Results

In this section, we have conducted simulation tests to study the performance of the algorithm in our framework. The implementation of the simulation prototype has been performed using Java (J2SE 1.5) and the Java Media Framework (JMF 2.1.1a) [66]. Session Initiation Protocol (SIP) [98] is used as the signaling. JMF can support several RTP [102] formats for video like H.261, H.263, MPEG-1 and M-JPEG [60]. Repurposing is performed through the use of JMF. RTCP is used in combination with RTP over UDP in the transport level. SIP over TCP is used to establish initial communication between the server and clients. The Real Time Streaming Protocol (RTSP) [101] and Session Description Protocol (SDP) [104] are used at the session level to manage streaming.

The tests are conducted 20 times by using 15 proxies and 50 different repurposing services. The time is measured to generate a repurposing graph and to find the path for a service. The tests are conducted for different proxies that range from 1 to 15. The measured findings are as follows:

- Only the number of repurposing services determines the time required to create a repurposing graph and to calculate the path. For a test with 20 repurposing services running on 8 and 9 proxies, the average graph creation time is 1525 ms and 1575 ms, respectively. Therefore, increasing the number of proxies does not significantly influence the time required to create a repurposing graph.

Table 2. Performance comparison using 15 proxies and 50 repurposing services

	RP-AL-1	RP-AL-2
Maximum user's satisfaction	0.66	0.53
Average graph creation time (msec)	2032.41	1993.75

- In another test, it was found that the RP-AL-1 is more satisfactory than that of the RP-AL-2, shown in Table 2; users were more satisfied with the results of algorithm one (RP-AL-1). In this test, it was found that even the number of proxies and repurposing services were increased to 15 proxies and 50 repurposing services respectively; the graph creation time is not increased significantly when compared with the previous test of 9 proxies with 20 repurposing services. This is due to the method overloading and initial run of the system. After the first run, it takes less time than the subsequent runs.
- The path finding time in each of the algorithms is almost 10 ms; however, in the work of Mao et al. [80], this time was approximately 500 ms.
- With the increase in repurposing services (nodes), the graph creation time also increases, which demonstrates a linear relationship [8], as shown in Fig. 4-3. This reveals that the modified algorithms are scalable.

To evaluate the feasibility and effectiveness of the service selection approach, we have ported the selection algorithm to a multimedia conferencing service application. The test was conducted in our university lab, where the proposed algorithm was implemented for selecting multimedia repurposing services. In this test, an Intel Pentium 4 3.6 GHz Windows XP Pro SP2 with 512 RAM PC connected to a webcam was used as the media server. Two Intel Pentium 4 3.6 GHz Windows XP Pro SP2 with 1GB RAM PC were used as proxies where transcoding services were running, and finally an Intel Pentium 4 3.5 GHz Windows XP Pro SP2 with 512 RAM PC and a PDA (blackberry) were used as clients

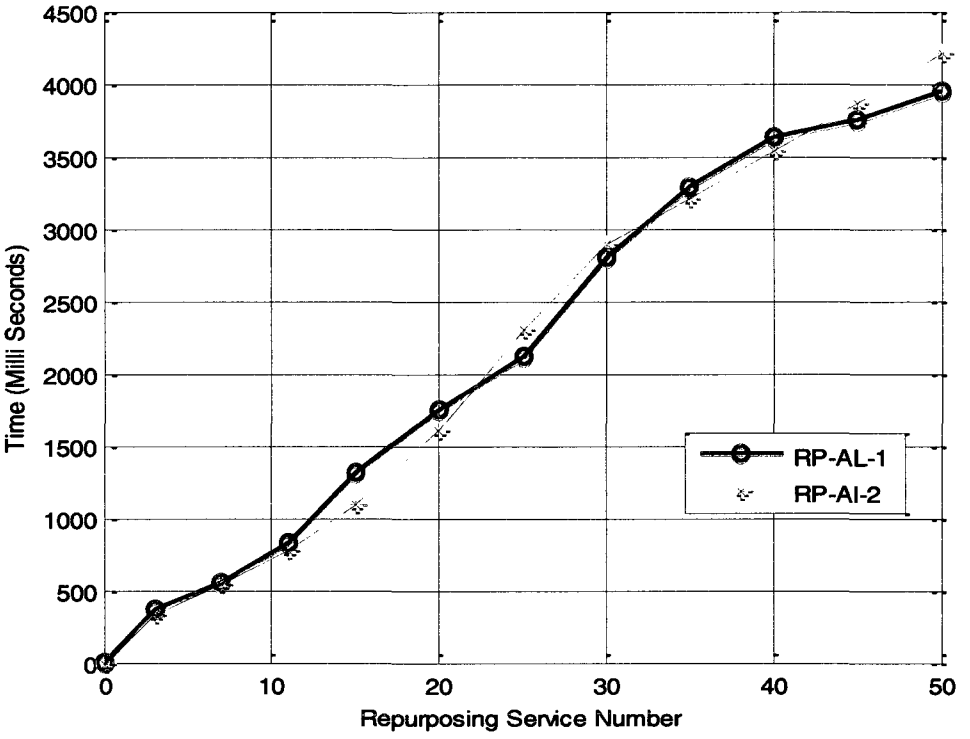

Figure 4-3. Linear relationship between repurposing services and graph creation time

The system periodically scans for available multimedia repurposing services and determines the best available service, in order to render the multimedia stream to different clients based on the QoS. During this test, one of the services is selected where an input video stream is repurposed from H.263 with 30 fps to H.264 at different frame rates (10, 15 and 30 fps). The different frame rate is intended for different wireless clients. As shown in Fig. 4-4, the average PSNR difference between the first two (the repurposed stream of 10 fps and 15fps) is 0.24 dB, while the difference between the first and the last (the repurposed stream of 10 fps and 30 fps) is around 0.85dB. The quantization parameter was the same for both the encoded stream and the repurposed stream.

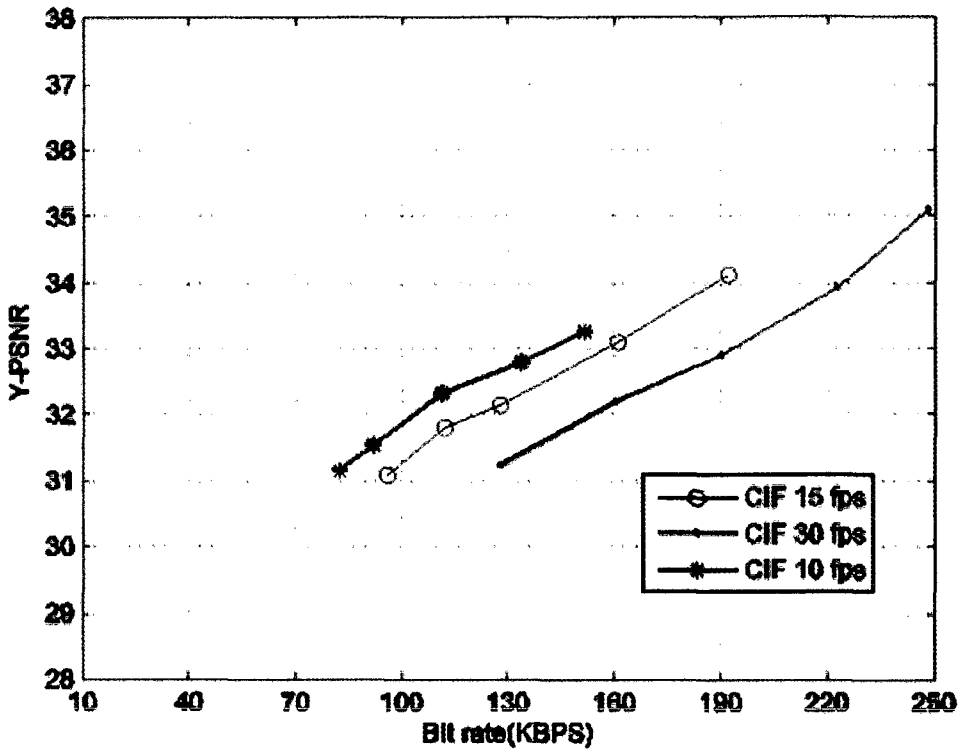

Figure 4-4. Repurposed Results: Different frame rates for different clients

4.6. Summary

This chapter presents the non bio-inspired (traditional) service selection algorithm to determine the best repurposing service path between the sender and the receiver (e.g. user/client). We use a series of repurposing proxies in a chain fashion between the server and multiple client devices. In order to find the appropriate chain of repurposing services that maximizes the user's satisfaction with the delivered content, we propose a service selection algorithm that is based on the user's satisfaction, with the quality of the repurposed content. The user's satisfaction is used as an optimization metric for the repurposing service selection and configuration process.

Chapter 5. The Bio-inspired Service Selection (BioReSS) algorithm

The repurposing service selection system with the traditional algorithms stated in the previous chapter depends on the global exchange of information among repurposing service nodes for path or service selection. However, due to highly dynamic traffic and heterogeneous network conditions, it becomes unfeasible. It cannot immediately respond to network changes or to repurposing service node failure. In order to overcome some of these challenges, systems with bio-inspired selection algorithms are gaining research attention. In this chapter, we will describe the proposed system with the Biologically Inspired Service Selection (BioReSS) algorithm. The BioReSS algorithm depends on the local exchange of information among repurposing service nodes, where each node autonomously sends ants through the network for finding the best repurposing path or service selection. In order to find the appropriate chain of repurposing services that satisfies the Quality of Experience (QoE) requirements, the proposed service selection algorithm uses the ant colony metaphor. During the communication session, the algorithm uses biological foraging behaviour inspired from ant agents to find optimal service paths between the media sender and the destination. Experimental results demonstrate that the proposed algorithm provides a significant performance gain over the traditional, state of the art selection algorithms and saves the average delay. Moreover, under an increased network load, the BioReSS has better performance than the traditional approach.

5.1. The Quality of Experience (QoE)

We apply the ant colony metaphor for selecting the best repurposing service route or path, which satisfies both the requirements of the users and resources (Q^E). This Q^E is used to evaluate a service path and optimize the selection algorithm. The QoE (Q^E) metric refers to the overall end-to-end service level performance from the user's perspective,

relative to their expectations and requirements about a multimedia service or application. The QoE (Quality of Experience) is more crucial to consider than the QoS in selecting multimedia services. The QoE has a direct impact on how well the multimedia service fulfills the user's needs, rather than the internal or implied impact of the QoS [65]. The QoS is a subset of the QoE and it may have an effect on the QoE. The QoE is a multi-constraint complex function that may include the user's satisfaction, response time, a multimedia quality parameter (e.g. frame rate) and a QoS-related parameter (e.g. delay and bandwidth). In order to select a repurposing service among proxies, we adopt Ant-Net[25] algorithms, where the ant controller packet is responsible for sampling and generating the repurposing path among the repurposing services that run on intermediate proxies. The model we used for assessing the Quality of experience (QoE) is as follows:

$$QoE\ (Q^E) = \rho \times QoS + (1 - \rho) \times S_e \cdots\cdots (10)$$

Where QoS ($q_D^e, q_{\beta_\gamma}^e etc.$) refers to the Quality of Service parameter (e.g. delay, bandwidth, jitter and so on), S_e refers to the user's satisfaction score level (qualitative measures), which is actually expressed in terms of the user's satisfaction metric as described in chapter 2, and ρ is the weighting factor, with a value between 0 and 1 for the importance of each of these measures.

The Quality of Service (QoS) is the summation of the values, given by two aspects, of all the quality of service attributes. The first aspect, ω is the weighting factor of attribute *i* QoS (q_i) parameter, such that the summation of all ω factors are to equal to 1. The second aspect, q_i is the value of attribute *i*. The formula is as follows:

$$QoS = \sum_{j=1}^{m} \omega_i q_i \cdots\cdots (11)$$

The user satisfaction (or user experience) is the summation of the values, also provided by two aspects, of all the user experience attributes. The first aspect, w is the weighting factor of attribute *i* with a value between 0 and 1. The summation of all w factors must be equal to 1. The second aspect s_i, is the value of attribute *i*. The formula is as follows:

$$S_e = \sum_{i=1}^{n} w_i s_i \cdots\cdots (12)$$

In a particular case, where the qualities of service measures were not considered, at that time ρ was given a value of 0. With a ρ of 0, (10) becomes (12) or (2). In that case, Quality of Experience (Q^E) and the user's satisfaction are the same.

5.2. Biologically-Inspired Service Selection (BioReSS) Algorithm

Biologically-inspired routing eventually solves the complex problem of routing through simple agents (ants) that are sent through networks to gather required routing information. This algorithm, inspired from a biological 'foraging' concept, which provides an efficient utilization of network resources in response to constant changes of networks. We first describe the AntNet algorithm in section 5.2.1. Then in Section 5.2.2, we discuss our proposed Biologically-Inspired Repurposing Service Selection (BioReSS) algorithm and the changes we made in order to adapt the AntNet into a repurposing service selection.

5.2.1 AntNet Algorithm

AntNet is based on the Ant Colony Optimization (ACO) meta-heuristics [6], which is inspired by the collective foraging behavior of ants that use a chemical substance called "pheromone" for indirect communication. These ant-deposited pheromone trails attract other ants to find the shortest path from the starting point to a destination. Based on a specific problem, an ant is given a starting point and moves through a series of intermediary neighbouring states to find the shortest path (optimal solution), by applying a stochastic decision policy. In AntNet [25], the data network is mapped on to a graph with N nodes, where *s* and *d* are the source and destination node respectively. Each node has a routing table that stores information about the outgoing links and their amount of pheromones. The pheromone routing table consists of a row for each destination of the network and a column for each neighbour node, for storing the pheromone values. The routing tables for each reachable destination are initialized with a uniform distribution of probability (equal value) i.e. the sum of each row must be equal to 1, which is shown in Fig. 5-1.

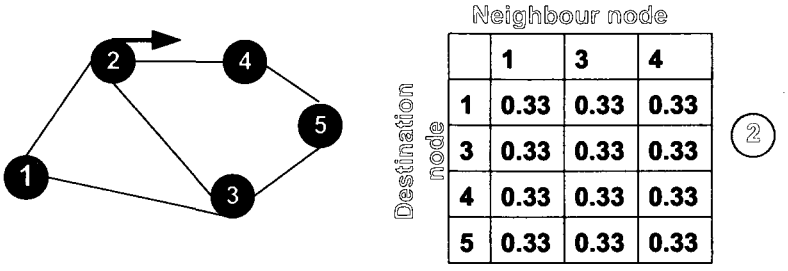

Figure 5-1. An example of pheromone routing table of node 2

As is illustrated in Fig. 5-2, there are forward and backward ants. At regular intervals, each node s launches a forward ant $F_{s\to d}$ to a randomly selected destination d stochastically. After that, a forward ant $F_{s\to d}$ applies the transition rule to choose the next node. This transition rule is based on the traffic load (e.g. link cost) and the amount of pheromones. While moving each ant $F_{s\to d}$ collects information about the state of the network, which is later used by backward ants $B_{s\to d}$ to update the routing tables along the followed path. In order to avoid the cycle, the forward ant $F_{s\to d}$ is forced to return to an already visited node. Upon successfully reaching the destination d, it generates a backward ant $B_{s\to d}$, transfers all of the information to the backward ant and then dies. The backward ant $B_{s\to d}$ returns to the source node s using the same path that was used by $F_{s\to d}$. The backward ant updates the corresponding routing table and the traffic (data) model of each visited node. On reaching the source node s, it dies.

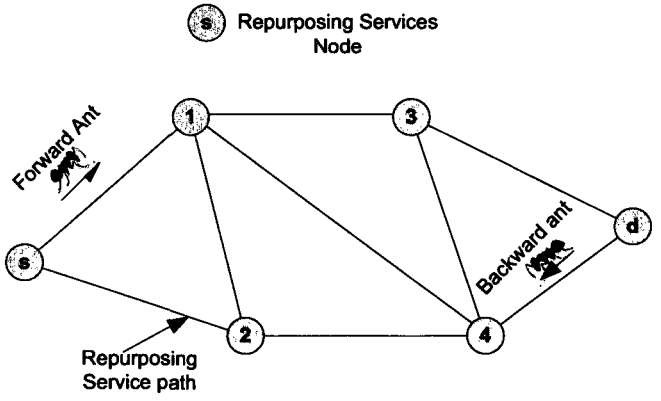

Figure 5-2. Repurposing service graph with a forward and backward ant

5.2.2 Content Repurposing by BioReSS Algorithm

In our proposed system with the Biologically-Inspired Repurposing Service Selection (BioReSS) algorithm, we use the similar transition rule, pheromone-updating rule and model data-updating rule, as described in [25]. The changes and other related features of BioReSS are summarized below:

In order to adapt AntNet-based algorithm for the optimization of repurposing service selection, we need to adjust its transition rule and pheromone updating rule. It may be mentioned that the pheromone updating rule is used for optimization. These rules are used to: a) choose or select the optimal service path according to the amount of pheromone on the service path; or b) adjust or update the amount of pheromone on the service path. We use a combination of QoE metrics with the pheromone for selecting an optimal repurposing service path or optimal (or one of the best) repurposing services.

An optimal repurposing path is referred to as a path that satisfies all QoS constraints. Among these paths, the repurposing path with the maximum positive weight is selected. The selection of the next repurposing service is based on the probability (P_i) [25] of selecting services according to transition rule (13). This selection decision is based on the local pheromone value ($\emptyset_i$) and QoE (Q_i^E) values. The pheromone values are initialized as $1/N$ towards a destination column.

$$P_i = \frac{\emptyset_i + \alpha \times Q_i^E}{1 + \alpha \times (N-1)}, \qquad 0.2 \leq \alpha \leq 0.5, \text{where } \alpha \text{ is empirically tested} \cdots\cdots (13)$$

If all neighbours have been selected, then the probabilities are equal for all of the neighbours. As shown in (14), Q_i^E is updated as the QoE value of selected service node (Q_δ), divided by the sum of all the QoE (Q_j^E) of the neighbouring repurposing service nodes, times the number of neighbours -1:

$$Q_i^E = \frac{Q_\delta}{\sum_{j=1}^{N} Q_j^E} \times (N-1) \cdots\cdots (14)$$

In the forward phase of the BioReSS algorithm, the ant stores the QoE, which is obtained from the service path or link. As described in (10), the QoE is calculated based on the network level parameter (e.g. bandwidth) and the user satisfaction or application level parameter (e.g. frame rate). This is required to send the video via that link. In this particular case, the normalized value for QoE (Q^E) is based on the frame rate, where the

maximum value of the frame rate is 30. This (Q^E) is also referred to the local heuristic variables (l_n) of each link, where $l_n = {}^{f}/_{30}$, $1 \leq f \leq 30$ and f is calculated using (8) and (9) in chapter 4.

Algorithm 2: High level steps for the proposed biologically-inspired service selection algorithm

Initialize:

1) Create a pheromone, routing and model table for all known neighbouring service nodes
2) Periodically create a forward ant and sends it towards a final destination

begin

  foreach *node* **do**

    foreach *ForwardAnt* **do**

      1) Add this node to the stack of visited nodes
      2) **if** *a possible loop detected* **then**
        Remove loop creating service nodes
      3) Store the QoE and other service information
      4) **if** *it is not the final destination* **then**
        Jump to the next service node and do following:
          a) Calculate the probability for selecting neigbouring service nodes using (13)
          b) Roll a random value to determine the service link
          c) Serialize the ant to the next service node
        else
        Create a backward ant to jump backward towards the sender

    foreach *BackwardAnt* **do**

      1) Pop the top of the stack to determine the next service node to jump
        if *the stack is empty* **then**
        die
      2) Update the pheromone value using (17)
      3) Update the QoE for the service node
      4) Update the data model (e.g. the mean,variance and best QoE)
      5) Update the routing table

end

Figure 5-3. Bio-inspired service selection algorithm

In the backward phases, the routing table is updated by changing the value of the (Q^E) that is obtained by sending the video/audio stream via the next node in the path towards a certain destination. In the BioReSS algorithm, it is the responsibility of each proxy node to decide the next node on the path and what stream it should be repurposed to, in order to obtain the best QoE at the receiver. Furthermore, in the BioReSS algo-

rithm, the graph of the network is generated as the ants return. Each node knows to start with only its 1 hop neighbours. At each node, the statistical parametric model or data model [25], such as the average QoE, the best QoE, and the variance of the QoE for each destination are updated. During the backward phases, as stated in the AntNet algorithm, the pheromone value is updated using (15).

$$\emptyset_i = \begin{cases} \emptyset_{i-1} + \Upsilon \times (1 - \emptyset_{i-1}), & \textit{Selected service node} \\ \emptyset_{i-1} + \Upsilon \times \emptyset_{i-1}, & \textit{Non} - \textit{Selected Node} \end{cases} \quad \cdots\cdots (15)$$

As backward ants return, the routing table is extended with nodes that can be reached via multiple hops. The backward ants generate and reinforce information (Υ) related to the optimal path. This information is stored in a local routing table on each proxy node. This information includes, among others, the format of the required media, QoE, the final destination, as well as the next proxy node in the chain – if any – where the output stream should be sent, in order for the repurposing process to be accomplished. The high level pseudo code of the proposed algorithm is given in Fig. 5-3.

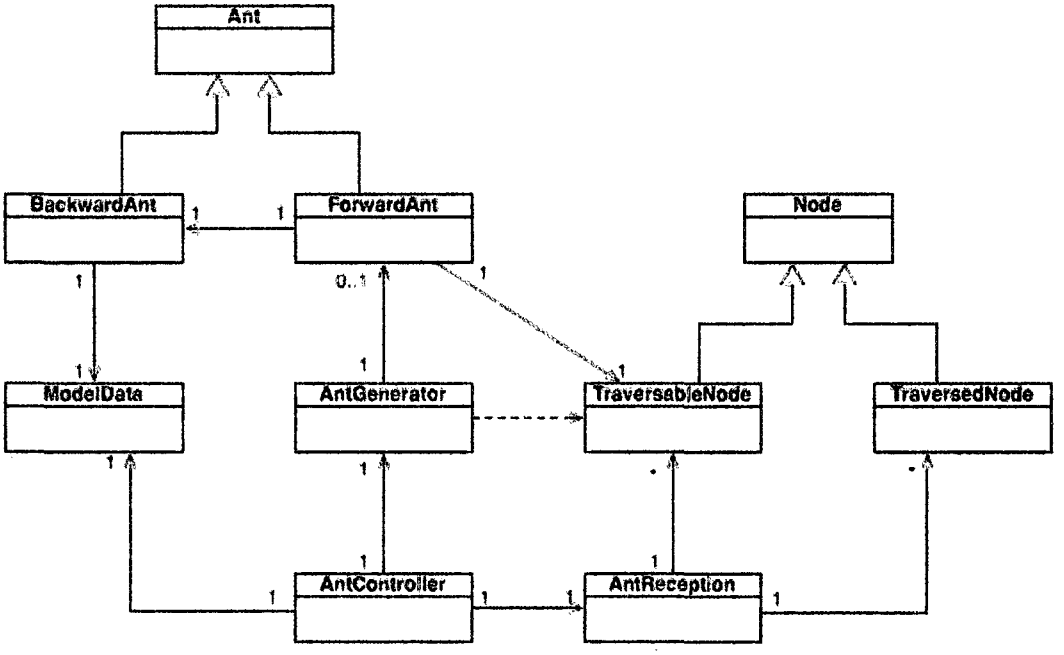

Figure 5-4. The class diagram for the BioReSS algorithm implementation

Moreover, for the case of service link or service failures with its neighbour, the probability (P_k) of selecting neighbouring service node (k) for the failed service node is dependent on the un-failed service nodes. In this case, the probabilities (P_k) for the failed neighbouring services are updated as [93] using (16).

$$P_i = (1 + \eth)P_i \cdots\cdots (16)$$

$$\eth = \frac{P_k}{1 - P_k} \cdots\cdots (17)$$

5.3. Implementation of the BioReSS Algorithm Key Components

Fig. 5-4 shows the class structure view for the proxy component that implements the Ant-based algorithm. In the following, we will briefly explain how these different classes interact with each other. First, each proxy in the system contains one single AntController object. It is responsible for initializing the ant algorithm by reading the XML configuration file, as presented in Fig. 5-5. It is also responsible for managing the generation and reception of Ant objects. This is achieved by running two separate threads: the AntGenerator and the AntReception, respectively. Moreover, the AntController initializes the proxy with different properties, such as the video repurposing type, and the number of frames per second. It stores them in the ModelData object for later use.

```
<?xml version="1.0" encoding="UTF-8"?>
<proxyNode xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
xsi:noNamespaceSchemaLocation='file:/c:/MCR/proxyNodeDefinition.xsd'>
<RepurposeDirectory>
      c:/RepurposeProxy/AntTranscProxy1/proxy1Dir.xml
</RepurposeDirectory>
<IP>137.122.89.112</IP>
<pathPort>20204</pathPort>
<antPort>30204</antPort>
<neighborList>
      <neighbor neighborDir="h:/MCR/senderProfile.xml"
      IP="137.122.91.89" dataPort="22322" pathPort="20202"
      antPort="23000" bandwidth="500000" type="in"/>
      <neighbor neighborDir="h:/MCR/AERIC.xml" IP="137.122.90.167"
      dataPort="22232" pathPort="20204" bandwidth="500000" type="out"/>
      <neighbor neighborDir="h:/MCR/receivProxy.xml" IP="137.122.91.223"
      dataPort="22240" pathPort="23000" antPort="23000"
      bandwidth="813700" type="out"/>
</neighborList>
</proxyNode>
```

Figure 5-5. Example of a simple ant proxy XML configuration file

The AntGenerator thread creates a ForwardAnt object towards the destination node. The ForwardAnt choose the next candidate *TraversableNode* object based on the proposed selection criteria and the available neighbours. The AntGenerator then forces the ForwardAnt object to jump to that TraversableNode. Upon arrival of the ForwardAnt at the TraversableNode, the AntReception reads the ForwardAnt object, stacks this node into the TraversedNode under the ForwardAnt and examines the ForwardAnt destination information against the node information, which is retrieved from the ModelData object. This scenario is repeated at each node along the path toward the destination node. If the node is the final destination of the ForwardAnt object, then the AntReception at that node forces the ForwardAnt object to create a BackwardAnt object and forces it to jump back toward the ForwardAnt object originator node before it dies. The BackwardAnt utilizes most of the ForwardAnt's collected information started from the source node. At each node along the path back to the source node, the AntReception repeats the same scenario with the BackwardAnt as it happens with the ForwardAnt object.

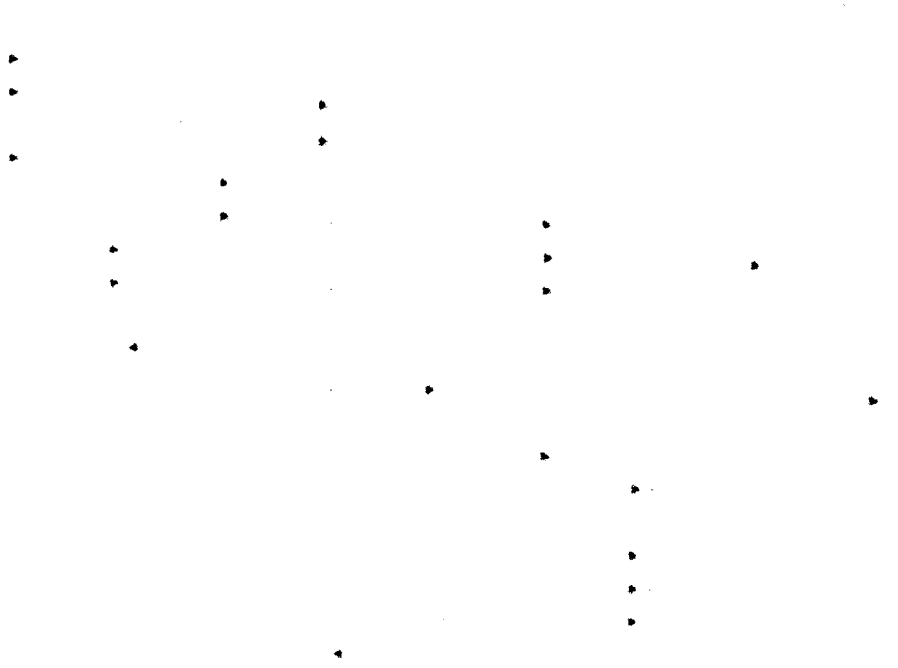

Figure 5-6. The Sequence diagram for the BioReSS algorithm

Finally, when the BackwardAnt arrives at its source node, it reports updated the statistical information and stores it in the ModelData object before it dies. The overall proposed service selection approach extended from AntNet is also described in the sequence diagram in Fig. 5-6.

5.4. Experimental Results and Comparisons

We compared the proposed BioReSS algorithm with the traditional service selection algorithm. Firstly, we implemented and adopted a Dijkstra-based service selection algorithm as a traditional service selection approach, which was presented in chapter 4. We then incorporated the BioReSS algorithm in a multimedia repurposing system to see how it performs at the service selection and/or the service path finding.

5.4.1 Comparison with the Traditional Service Selection Algorithm

In these experiments, we compared the throughput, routing overhead, and adaptability or dynamicity of the BioReSS and traditional selection algorithm.

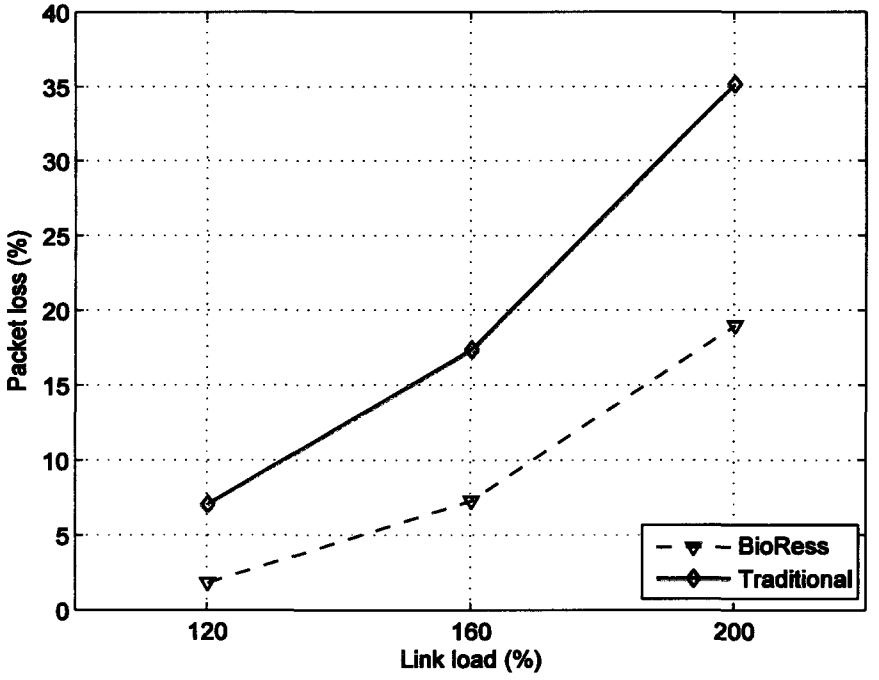

Figure 5-7. Throughput comparison of the traditional and BioReSS algorithm

Throughput Comparison

In this comparison, we measured the number of lost packets under an increased network load. In the BioReSS, the ants are sent via a socket through the use of Java Serialization, while the video/audio stream is sent via RTP/RTCP [98] through the use of Java Media Framework (JMF) [63]. Consequently, the ants are sent via high priority queues. As a result, fewer packets are dropped under the increased network load in the BioReSS. This causes the increase in routing overhead in our proposed BioReSS over the traditional algorithm. As plotted in Fig. 5-7, the BioReSS algorithm shows higher throughput than the traditional selection algorithm. This is because the proposed BioReSS algorithm forwards extra packets to the destination, while these extra packets do not exist in the traditional selection algorithm. In the BioReSS approach, the entire network is used efficiently, while in the traditional approach, it is not used efficiently when the network links experience an increased load. In such a case, some links are not used at all.

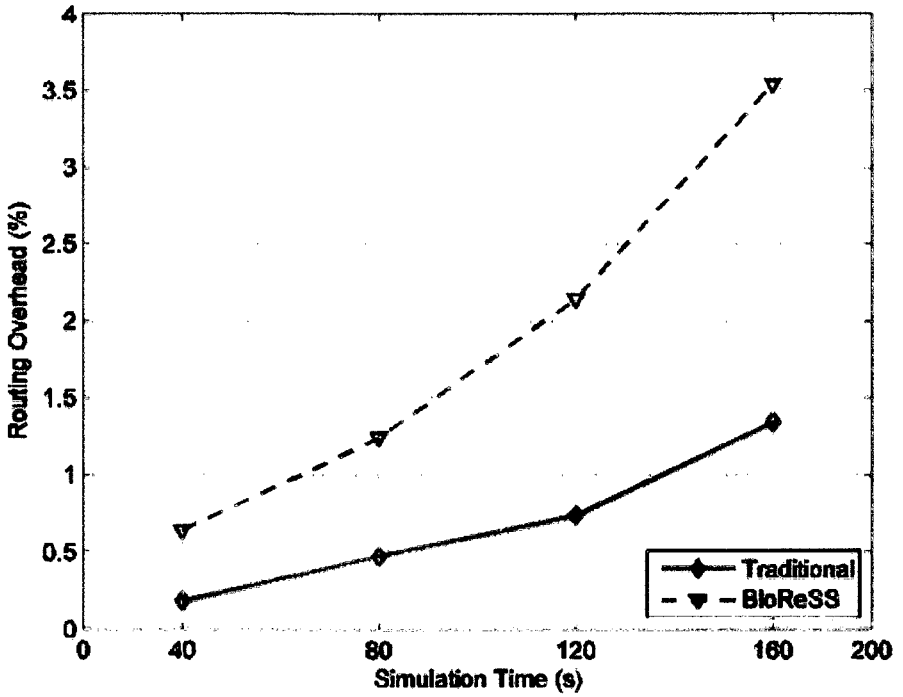

Figure 5-8. Routing overhead comparison of traditional and BioReSS algorithms

Routing Overhead Comparison

In this experiment, we compared the overhead generated by the routing packets of the BioReSS with that of the overhead for the traditional algorithm. For each algorithm, the overhead is measured as the ratio of the bandwidth occupied by routing packets to the total available bandwidth in the network. For this experiment, we use a network topology similar to that of Fig. 5-2, which has 6 nodes and 9 bi-directional links. The link bandwidth is around 1.0Mb/s and link delays range from 5 to 10ms. For the overhead comparison, the simulation is conducted in Java. The simulation runs for 200 s with the initial 45 s for initializing the routing tables for the network topology. We consider the packet size of 128 bytes and the mean packet inter-arrival time of 0.001 s. The transfer rate is therefore approximately 1.04 Mb/s (128×8×1000). All reported data in the graph is averaged five times.

Fig. 5-8 depicts the comparison of the overhead generated by the routing packets for both algorithms. As in the BioReSS, it is periodically required to send ant packets to monitor and collect the state of the repurposing paths and networks. The routing overhead of BioReSS is significantly higher than that of the traditional. The periodic updates of the routing packets cause an increased overhead in the BioReSS. In addition, network topology changes also cause the increased overhead in the BioReSS. One way to reduce the overhead generated by an ant packet is to adapt the approach proposed by [28], where the method does not maintain routes to all possible destinations, but instead sets up routes only on demand. This approach results in a lower number of routing packets and control packets in the network. The lower number of packets eventually reduces the overhead that has been generated by the packets. The service paths are set up in the same way as BioReSS. During the data session, the paths are monitored, maintained and improved using the proactive forward ants [27]. In this way, the BioReSS can more quickly respond to link failures or other dynamic changes.

Even though the routing overhead of the BioReSS is significantly higher that of the traditional method, it does not affect our proposed system, as (a) it is negligible [26] compared to the resource consumption compensated by the added performance than the traditional approach; and (b) the streaming subsystem works independently from the rest of the multimedia conferencing application system. When a user wants to stream a video

during a conferencing session, it is assumed that the BioReSS program has already been running in the background and has been generating the network topology, structure and values. Thus, the extra overhead does not affect the proposed repurposing system. Finally, the BioReSS system is more robust than the traditional system. Even during service node or service link failure, throughput does not decay completely and it can also recover from that state.

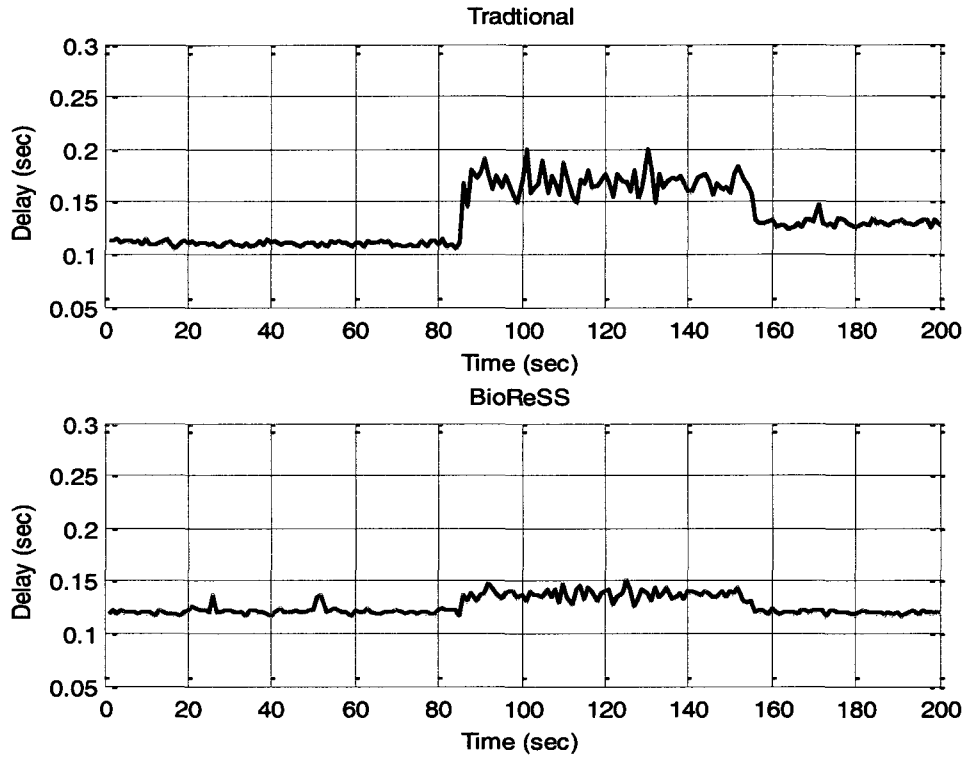

Figure 5-9. End to End to delay comparison for the traditional (top) and BioReSS algorithms (bottom)

Dynamicity or Adaptability

We evaluate the proposed algorithm in terms of its adaptability, dynamicity and self-management capabilities. We use the same test configuration as the overhead comparison. For this test, we demonstrated how the proposed algorithm and the traditional algorithm adapt to the changes of the service network, which are the link failure states (or service failure) at 85-155 seconds and link reconnection or alternative link (service join-

ing) states at 155-200 seconds. During the test, one of the links is broken after 85 seconds and at 155th seconds, it is reconnected.

In addition, the end-to-end delay (time required to travel a packet from source to destination) and jitter (variance of delay) are measured. During link failure, the proposed algorithm outperformed the traditional algorithm in terms of reduced mean jitter (45%) and mean delay (10%). The reason was that the traditional algorithm could not adapt to the changes of the service link efficiently. Moreover, after the failure (at 155-200 seconds), the proposed algorithm also outperformed the traditional one by 3.5% delay.

Figure 5-9 compares the end-to-end delay of the traditional algorithm with that of the proposed algorithm during link/service failure and link (service) joining. We observe that when the service or link failure occurs at 85-155 seconds, there is an increase in the average delay in both algorithms. However, this increase (8%) in the case of the traditional algorithm is more than that of the proposed algorithm.

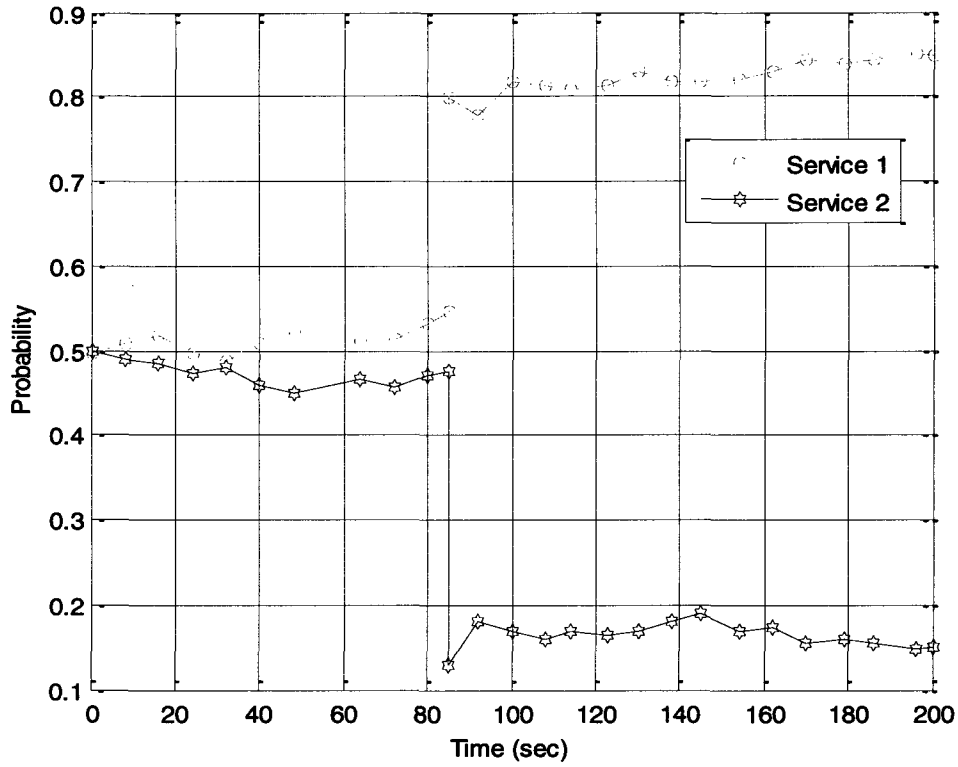

Figure 5-10. Probability of next service selection when one service is failed

Fig. 5-10 shows how the proposed approach works for the case of service failure. As seen from Fig. 5-10, when service 2 fails, the proposed algorithm selects the next service based on the probabilities of selecting un-failed services using (16). The probability for the service 2 reduces to the amount close to zero, while the probability for selecting service 1 increases as much as the amount of the decreased probability for the service 2.

5.4.2 Experiments with Multimedia Systems

We have conducted experiments to study how our biologically inspired repurposing service selection (BioReSS) algorithm behaves for real-time multimedia applications such as multimedia streaming. The results demonstrate the effectiveness of the proposed BioReSS algorithm for selecting repurposing services to repurpose multimedia content. We conducted the tests by deploying the proposed algorithm in a multimedia conferencing service environment. This application periodically scans for available multimedia repurposing services. It also determines the best available service by using the proposed service selection algorithm, in order to render the multimedia stream to different clients based on the user's satisfaction.

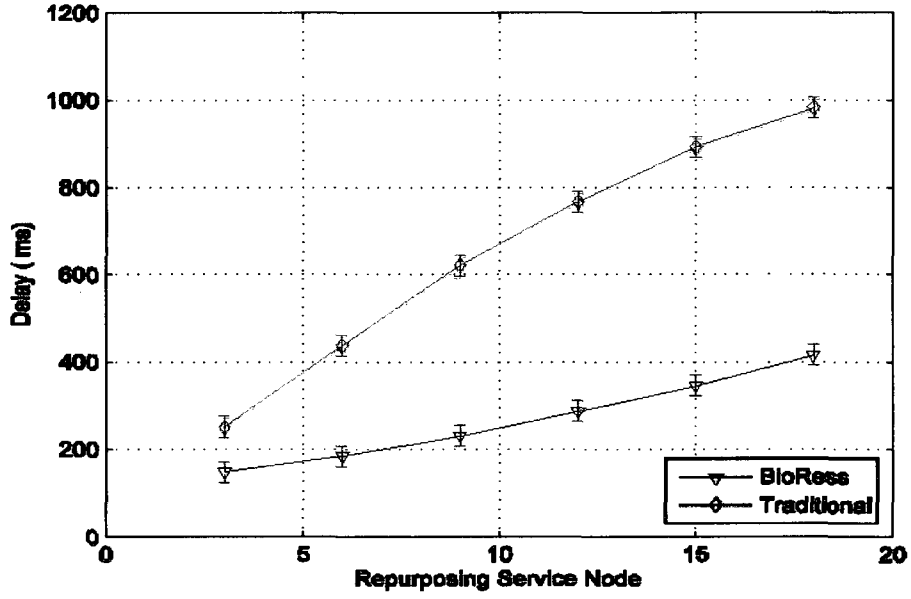

Figure 5-11. The repurposing delay in two algorithms after having ported on a multimedia application

After running the application, we measured the quality of the repurposed content. We used a widely accepted objective measure of visual quality metric called the Peak Signal to-Noise-Ratio (*PSNR*), which is defined in (18).

$$PSNR = \log_{10} \frac{255^2}{MSE} dB \cdots\cdots (18)$$

Where, *MSE* is the mean square error between the original content and the reconstructed visual content.

Table 3. Performance comparison of the system

	BioReSS	Traditional
Average path Generation Time	virtually 0 ms	209.1 ms
Repurposed Content Quality (PSNR)	32.05 dB	32.06 dB

The tests were conducted five times by using 20 repurposing services (node). The repurposing delay was measured as the delay in sending, repurposing, and receiving the stream from the media sender to the receiver. The results were compared to the previously developed system [56] that used a traditional repurposing service selection strategy.

As shown in Fig. 5-11, the BioReSS algorithm outperforms the traditional selection algorithm in terms of average delay. As illustrated in Table 3, the path generation time for the BioReSS is virtually zero, because the next node (sender or proxy) calculation is finding the neighbour with the best QoE from a hash table that is locally stored. Consequently, we find that the quality of the repurposed content in both cases is almost the same (e.g. 32 dB). In order to test this quality, the system takes a live stream of motion JPEG at 30 fps and repurposes it to H.263 at a frame rate of 15 fps.

There are two points regarding the path generation time for the proposed algorithm. One is the time spent by the algorithm for its calculations; the other is the time that a user of the system notices. In the case of a traditional algorithm, the actual path that the video/audio stream follows is computed at that time when the user wishes to send/receive the stream. As a result, the user notices the time that it takes to generate the path. The proposed ant-based algorithm decouples the path generation into a separate process. It

takes time for the ants to generate the complete path. The time might be larger than that of the traditional algorithms, but that time is spent before a user decides to send/receive the stream.

Therefore, from the user's point of view, the path generation time is 0 when he wishes to send/receive because of the following reasons: a) the path has already been generated, and, b) it is just a matter of searching some lookups (hash table) at each node to decide the next node along the service path.

Another point to mention is that the ant algorithm has a higher initialization cost (convergence time), but once converged, the path is updated in "real-time" in parallel with the video transmission. For the traditional algorithm, the path is computed each time when a video is sent or received.

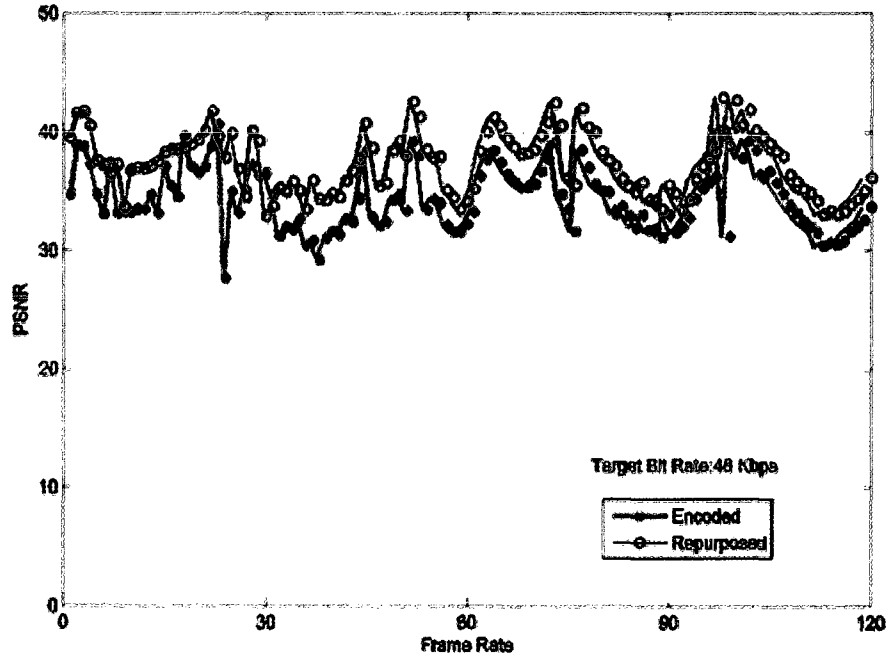

Figure 5-12. Quality comparisons of the repurposed stream and the encoded stream at 48 kbps

After deploying the BioReSS, we have reported some significant subsets of the results of the experiments conducted on our multimedia content repurposing system. One of the repurposing services that we have developed is able to repurpose video content

from Motion JPEG (MJPEG) to H.263 at different target channel bandwidths. As previously mentioned, we have measured the video quality by calculating and comparing the PSNR of the repurposed video content and encoded content. As shown in Fig. 5-12 and Fig. 5-13, the video content is repurposed from the MJPEG to the H.263 for the low bit rate target channels of 48 and 24 kbps. There is a sudden drop in both graphs in Fig. 5- 12 and 5-13 for the encoded data series, which is due to the changing of the scene. However, even when the scene changes occur at the 24th and the 99th frame, because of the repurposing, the quality of the repurposed frame is higher than that of the encoded stream. There is a quality gain of 2 dB after repurposing the video content into H.263. This quality gain is due to the repurposed multimedia stream, where bandwidth is controlled.

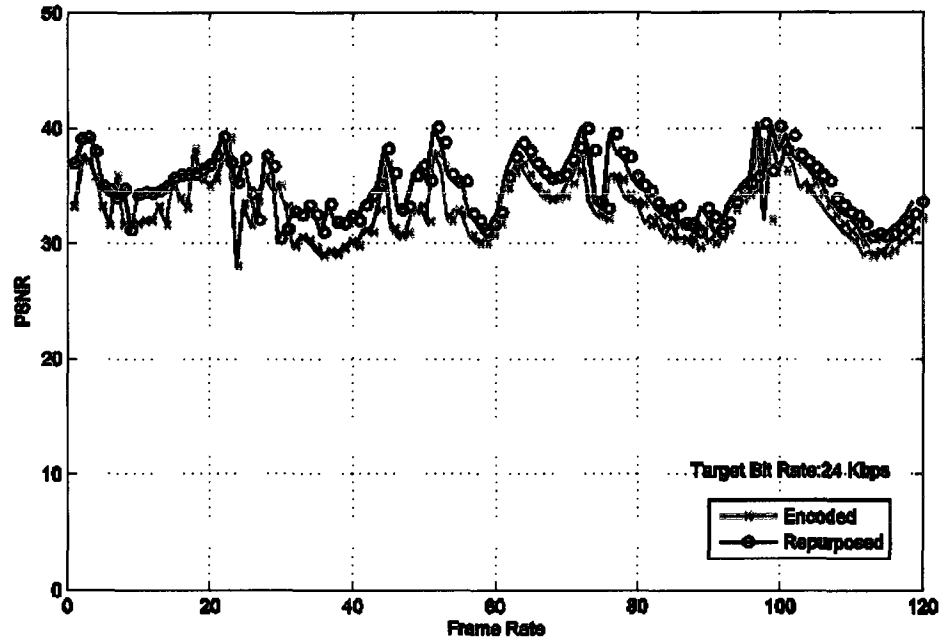

Figure 5-13. Quality comparisons of the repurposed stream and the encoded stream at 48 kbps

5.5. Summary

In this chapter, we have presented one of the novel biologically-inspired repurposing approaches for a multimedia system. The system adopts the biologically-inspired ant colony foraging behavior for multimedia content repurposing. With this proposed system, the main component, the BioReSS algorithm facilitates in selecting repurposing services and optimal service paths. The selection is based on Quality of Experience, by considering both the user's satisfaction (s_i) and the network constraints (e.g. network level QoS). From the obtained results, it is found that the bio-inspired solutions for multimedia can potentially fulfill heterogeneous services and the expected user's demands in modern and dynamically changing network environments. We have evaluated and compared our proposed system through implementation and simulation.

Chapter 6. The System Implementation and Validation

In this chapter, we presented the system implementation and validation of the proposed framework. We validated the proposed multimedia service management framework through implementation as well as simulation. We also used simulation to present composite service maintenance issues such as scalability, and load management. We used implementation to demonstrate multimedia service compositions for some target ubiquitous users. We have performed quantitative scalability evaluation through simulation and qualitative analysis through user studies in order to validate the suitability of our proposed framework. Finally, we have conducted the qualitative analysis through a small scale usability studies.

6.1. System Implementation

We have implemented the proposed multimedia service management framework by following the SOA paradigm, where services are registered, queried (or searched) and used. In this framework, we have not used UDDI as in [113]. We have instead used a private directory as the registry. In this case, the registry is a MySQL [83] database that contains three tables: the streaming services table, the proxy nodes table and the repurposing services table. Each table holds information regarding the type and quality of the stream provided by the corresponding service. The streaming service is implemented using the Java 6.0 [67] and Java Media Framework 2.1.1e. The streaming web service publishes, among others, a method that allows users to request a multimedia (e.g. video) stream of a particular format and resolution through SOAP messages. The bio-inspired service selection component, which is responsible for collecting QoS information, uses Java's capability to serialize and transmit ant objects to different repurposing services in the system.

Using SOAP over HTTP is not suitable for delivering multimedia streamed content because SOAP messages introduce unnecessary overhead. This overhead includes,

among others, processing cost, message parsing, and marshalling or un-marshalling process [124]. As a result, we divided the implementation of our multimedia web services into two parts. The first uses XML-based SOAP messages over HTTP, in order to gather all relevant information (such as port number, type of services, QoS etc.). The second part uses socket communication in order to stream the media content.

6.2. The Experimental Setup

In order to validate our framework; we have deployed five repurposing services for real-time multimedia streaming for different ubiquitous clients. Fig. 6-1 shows the top-level implementation of the proposed system as an example of content repurposing. Each of the repurposing services belongs to a different composition plan, as stated in section 3.1.2. In this test, an Intel Pentium 4 3.6 GHz Windows XP Pro SP2 with 1GB RAM PC was used as the media server. Five Intel Pentium 4 3.6 GHz Windows XP Pro SP2 with 1GB RAM PC were used as repurposing proxies, where repurposing services were running. An Intel Pentium 4 3.5 GHz Windows XP Pro SP2 with 512 RAM desktop PC, a PDA (blackberry), one cell phone, and notebook computers were used as clients. Among these, some of the clients were connected through Wireless LAN 802.11 and some of these were connected through GPRS. The Graphical User interface (GUI) is used to allow the user to modify the location of the server profile (as described in appendix F) and video bandwidth files (as described in appendix C), and to edit the port values and the name of the server as it will appear in the registry.

6.3. Multimedia Service Composition

During the validation tests, we have considered a number of repurposing services, as described in Table 4. In this table, we have listed some repurposing services, along with their input or output capabilities and different QoS attributes (repurposing delay, bit rate and frame rate). As shown in Table 4, the repurposing service 1 (RS_1) of row 1 is capable of taking MJPEG 320x240, 30 fps as input and delivering MJPEG 160x120 as output, which is suitable for rendering to the client (e.g. cell phone) at a bit rate of 48 kbps, frame rate of 8.6 fps, and a repurposing delay of 92ms. It should be mentioned that we have

measured the repurposing delay as the time it takes for the captured video stream to go through the individual repurposing service. We then have described two composition scenarios for the validation of the proposed framework.

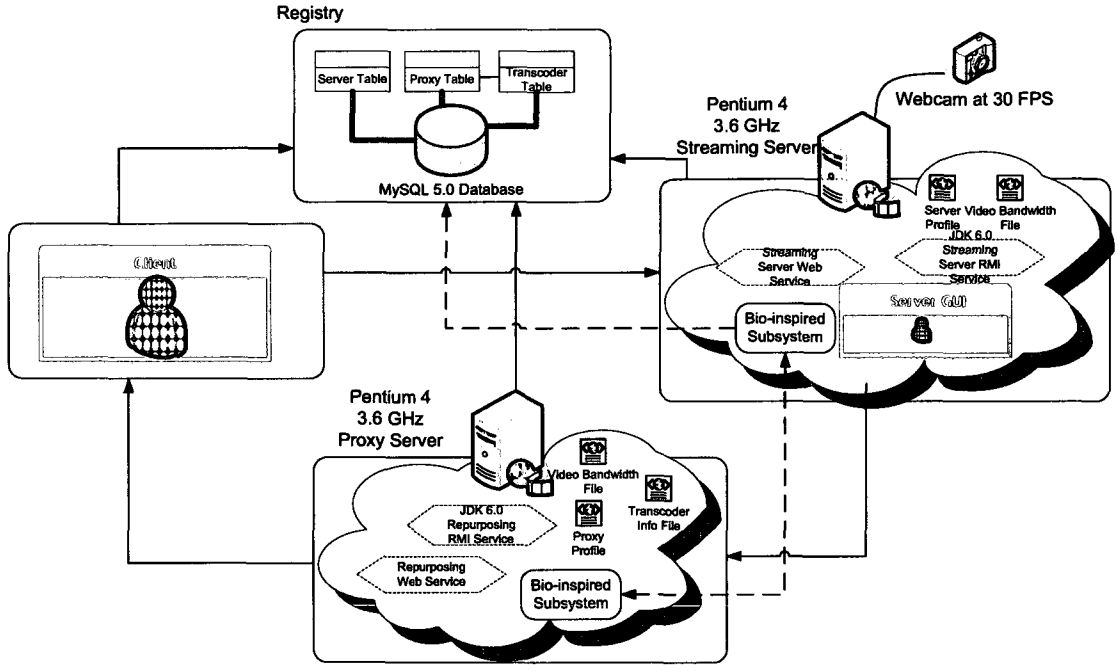

Figure 6-1. Detailed top-level implementation

At first, we considered an application scenario where a wireless client (e.g. a laptop connected to a Wireless LAN) accesses a live video stream. The stream was originally captured from a webcam as MJPEG 320x240, at 30 fps and then delivered to the client as H.263 352x288 at 15-20 fps with a bit rate of 245-330kbps, due to its capability of accepting such a media format. In this particular scenario, the client (Laptop) cannot play any MJPEG stream; it can only render an H.263 352x 288 formats, which requires that the source content be repurposed to the appropriate format. Given the list of repurposing services in Table 4, the composition plan we used in this scenario is:

$SC_1: MS \rightarrow RS_2 \rightarrow Laptop.$

Here, MS is the multimedia streaming services and RS is the repurposing service used. Note that only one repurposing service is used in this presented case; the service with the identification number 2. Therefore, we can call this an atomic composition, or a simple composition.

We have then examined a complex composition scenario where a wireless user (e.g. PDA) has a GPRS connection with a maximum bandwidth of 28.8 kbps and can accept videos with a frame rate of around 9 frames/second for SQCIF (128x96) resolution. Similar to the above simple composition scenario, the video stream is originally captured from a webcam as MJPEG 320x240, at 30 fps. In this particular case, the user is unable to attain the service according to his QoS demands from the available repurposing services' list. Now the user (PDA) has to search for and select the appropriate repurposing service in order to render the multimedia stream that satisfies his QoS requirements, namely the format (H.263), the PDA's small display (128x96), as well as the low network bandwidth (28kbps) and frame rate (8.1fps-9.4fps). Given the list of repurposing services in Table 4, the composition plan that we used in this scenario is as follows:

$SC_2: MS \rightarrow RS_3 \rightarrow RS_5 \rightarrow PDA$. Note that two repurposing services are selected and composed with the streaming service. Therefore, we call this composition a complex composition.

Table 4. Repurposing services with QoS (bit rate, frame rate, delay etc.)

Service no.	Possible Repurposing Services (RS)	Average bit rate (Kbps)	Average frame rate (fps)	Average repurposing Delay (ms)	Target clients for the service
RS_1	Input = MJPEG 320x240,30 fps Output = MJPEG 160x120	48	8.6	92	Cell phone
RS_2	Input = MJPEG 320x240, 30 fps Output = H.263 352x288	320	15.3	100.01	Laptop
RS_3	Input = MJPEG 320x240,30fps Output = H.263 176x144	96	14.5	98.7	PDA
RS_4	Input = H.263 352x288,30fps Output = H.263 176x144	64	9.2	99.1	PDA
RS_5	Input = H.263 176x144, 15fps Output = H.263 128x96	28	8.8	93.5	PDA

6.4. Quantitative Analysis

The quantitative analysis of the proposed framework is performed by evaluating scalability and load balancing.

6.4.1 Scalability Study

Scalability refers to the ability of a system to adapt under an increased load (e.g. the increasing number of concurrent request) without sacrificing performance [21]. In the context of our work, scalability investigates how well the system works with the increased number of concurrent composition requests while meeting the user's demand. The experimental results demonstrate our proposed framework's capability to scale under: a) an increased number of services in the system (i.e. service registry); and b) an increased number of concurrent composition service requests by the users. Our SOA-based system provides linear scalability to a certain extent, and increased throughput.

Scalability: Average Response Time vs. Concurrent Requests

We have conducted a qualitative scalability test where the work load of the system is defined as the number of concurrent composition requests, the number of service nodes and the server are defined as resource, and finally the response time is defined as performance metric. Response time is defined as follows:

"Response time is the difference from the instance at which the client initially makes a service request for streaming service to the time at which the server responds to the client notifying that the transmission has been started after having gone through all the required repurposing service nodes for target composition. Response time includes processing time, convergence time (initialization time), and service selection time."

We consider three randomly generated distributed service nodes topologies namely 49, 64 and 100. The service nodes have been connected with their neighbouring service nodes based on randomly generated user's satisfaction metric (0.1 to 1). The proposed BioReSS algorithm is used to select services for the service composition. We assume that 10% of the active users (12,000) in the system belong to concurrent users. Consequently, the results demonstrate that the proposed system comfortably scales to at least 1200 concurrent user request, representing a 12,000 to 15,000 active user request. Firstly, we ran the system with an increasing number of concurrent composition requests, from 10 to 1,200 for 49 services nodes. Then we increased service nodes from 49 to 64 and eventually to 100 in order to study the effects of resources on scalability. We used multithreaded clients in Java to emulate 1,200 concurrent requests. The test was one hour long, each repeated 10 times. Finally, the response time was recorded and averaged. The

test was conducted in mainly two phases. The former phase was emulated in a scenario where, a single server or proxy server was used and load distribution was not considered. In the later phase, more than a proxy server was used and extra load (concurrent request) was distributed to proxy servers.

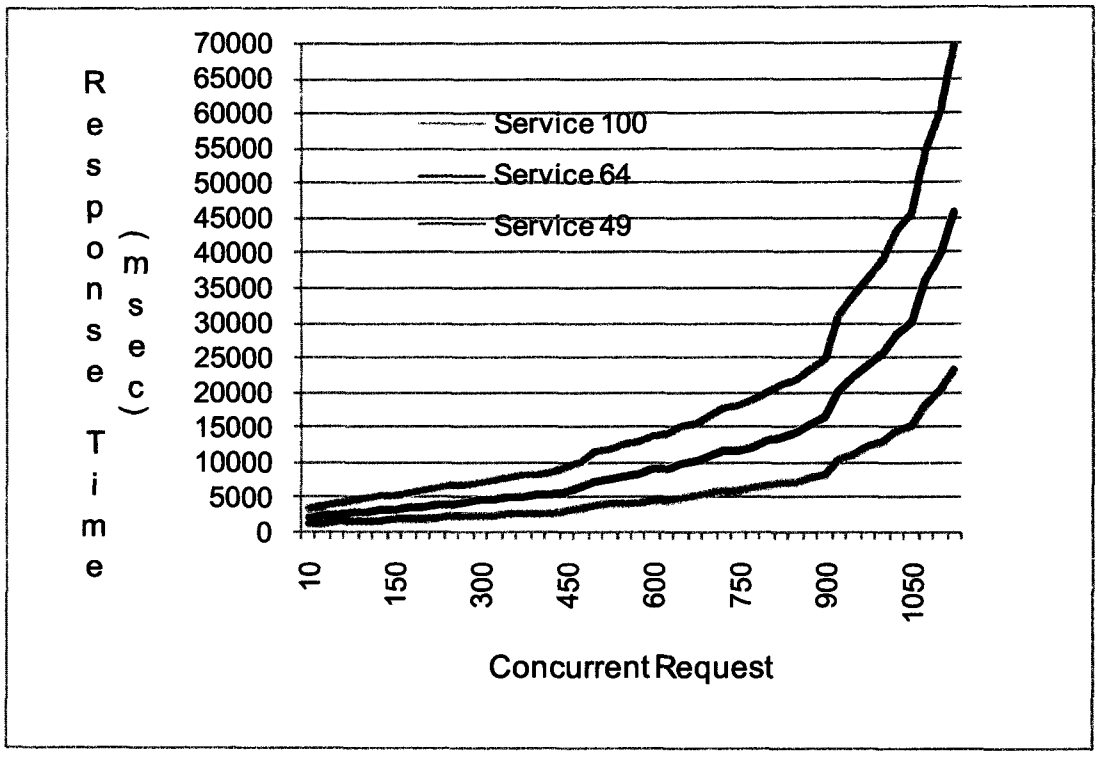

Figure 6-2. Scalability over increasing number of concurrent request with varying service nodes (without load distribution)

As shown in Fig. 6-2, the response time gradually increases in response to the increased number of concurrent requests whereas the average response time increases linearly with the number of concurrent requests. This linear scalability continues until a certain threshold (450 concurrent requests for this experiment). When the number of requests made exceeds the said threshold value, the increase becomes exponential, the performance deteriorates and response time increases unexpectedly (particularly when running with single server with single processor). The above contingency occurs when the number of concurrent requests that overloads the server has been reached. As a result, it may cause inability to handle the increasing number of concurrent requests or accept more requests. In order to avoid server overloading, keep response time within an accept-

able level, maintain scalability, and serve maximum number of request, load needs to be distributed.

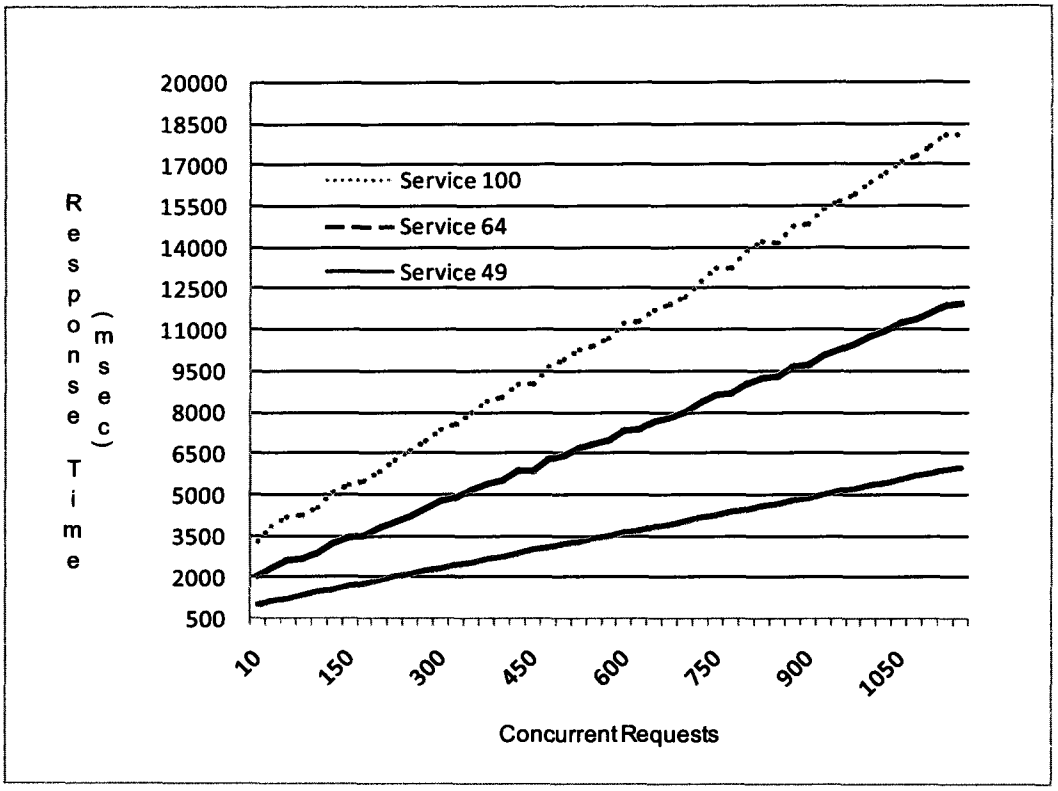

Figure 6-3. Scalability over increasing number of concurrent request with varying service nodes (with load distribution)

Fig. 6-3 shows scalability with load distribution while running more than one server to handle more concurrent requests. We have observed that as the load is distributed to next available proxy server, response time increases linearly with the increasing load (e.g. the number of concurrent request) compared to previous phase of scalability test described in Fig.6-2. We also found that by increasing the number of service nodes, the response time did not dramatically increase. As shown in Fig. 6-3, at 450 concurrent requests, the average response time slightly increases with 49 repurposing service nodes (around 2939.466 ms), compared to time with 64 repurposing services (around 3099.208 ms). While running 100 repurposing services, the response time has increased to 3163.221 ms. We conclude from the results that the system is linearly scalable for this

test phase. Moreover, load is distributed among the proxy servers, while serving the same rate of request per second (throughput).

It is worth mentioning that if we exclude convergence time (initialization time) from the response time, the response time increases with the number of requests as stated in Fig.6-3. However, it slightly decreases with the increase of service nodes from 49 to 64 and eventually to 100. The decrease is due to the availability of more services in the system and less time required in searching for available services.

Scalability: The System's Throughput vs. Concurrent Requests

In order to understand the predicted throughput in terms of service requests per second in our proposed system, we apply Little's law, which states that the average number of concurrent service composition requests (N) in the system is equal to the average arrival rate (λ) of the service request to that system, multiplied by the average time (T) spent in that system. This is expressed as follows:

$$N = \lambda T$$

$$\lambda = N/T$$

Where, N is the number of concurrent requests, λ is the throughput of N requests in terms of served requests per second in the system, and T is the response time in seconds.

We used the PushToTest Testmaker 5.1 [94] for testing scalability over the system's throughput. PushToTest Testmaker 5.1 is suitable for testing Service Oriented Architecture (SOA) based applications and web applications. In order to test the scalability, TestThread is required to send a request to the service. We ran one TestThread for each concurrent request and record, and saved the test results to the log file. We ran the test for 1200 concurrent requests. For this experiment, we used the same media server as a web server for accepting requests, and the desktop PC as the test client where a number of concurrent TestTthreads were running.

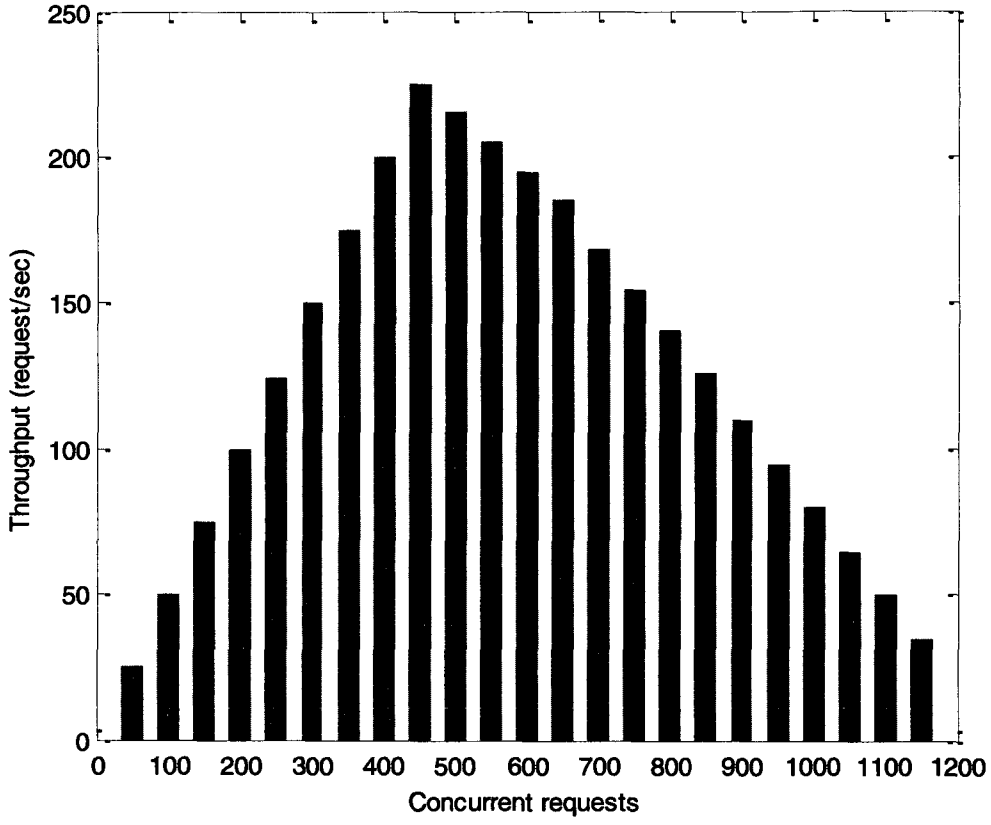

Figure 6-4. Scalability over the system's throughput (request per second)

Fig. 6-4 shows that increased throughput is directly proportional to the increased number of concurrent requests, or the throughput increases linearly with the number of received requests, while maintaining the average system response time constant no matter the number of concurrent requests. As shown in Fig. 6-4, at 150 concurrent requests, the proposed system handles 1500 requests in 20 seconds, which results in a throughput of 75 ($\lambda = \frac{N}{T} = \frac{1500}{20}$) requests per second, with a 2 ($T = \frac{N}{\lambda} = \frac{150}{75} = 2$) second response time. In order to keep the response time at 2 seconds, the same system at 300 concurrent request handles 3000 requests in 20 seconds, the outcome of which is a throughput of 150 ($\lambda = \frac{N}{T} = \frac{3000}{20}$) requests per second. Therefore, we find that with the increase in concurrent requests, the throughput also increases in the same manner, which demonstrates scalability. In other words, the increasing number of requests does not affect the system's response time. In this test (for the single server), scalability was achieved as long as we had

about 450 concurrent requests. When the number increases, the system's throughput begins to decrease. In order to have the desired throughput or to solve this problem, the media server load must be controlled or distributed to another serving site. In this case, the load is required to be distributed to different proxy servers. The dispatcher subsystem (described in section 3.2.2) facilitates this task in distributing load efficiently to different proxy servers.

6.5. Qualitative Analysis

We have conducted small scale usability study to qualitatively measure our proposed framework. This study consists of 10 undergraduate volunteers from different faculties. At first they were briefed about the system and some media related QoS parameters such as resolution, frame rate etc. Afterwards, the users were asked to use the system to select media service with their desired QoS. Based on their usage experience, they were asked to answer a number of questions on a Likert five-point scale. The questionnaire was divided into three different clusters (e.g. Q1-Ease of Use, Q2- User's Perception in Selection, Q3-Visual Quality). The answers for each section are then summarized and we calculated the average of the responses detailing their feedbacks. Fig. 6-6 shows the user's responses in percentage.

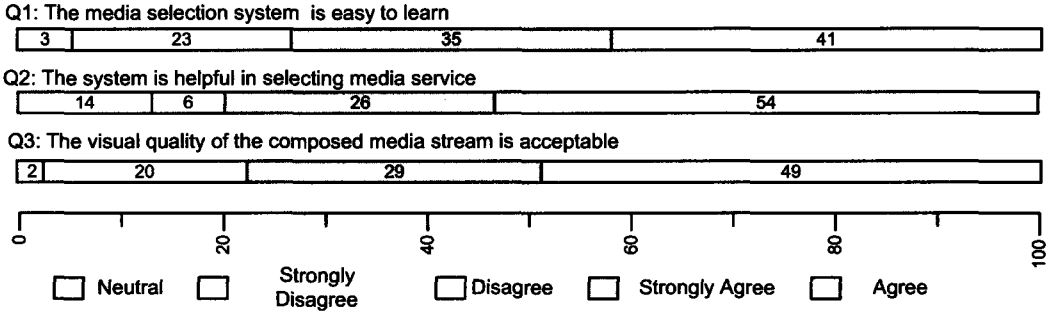

Figure 6-5. User response percentage during evaluation

Table 5 illustrates the overall rating of the user. The higher mean values *User's Perception in Selection*, and *Visual Quality* represent highly satisfactory response, while the moderate mean value of *Ease of Use* corresponds to comparatively low satisfactory response than those of the previous two.

Table 5. Satisfaction rating on the overall response on the Likert scale

	User's Perception in Selection	Visual Quality	Ease of Use
Std. Dev.	0.18	0.24	0.33
Mean	4.33	4.21	3.96

Fig. 6-6 shows the satisfaction rate of the system on Likert five-point rating scale in selecting media service for target composition. Approximately 26% of users have agreed that the system is highly satisfactory in selecting media service, and 54% of the users agree that it is satisfactory, while 6% of users have different opinion of the satisfaction.

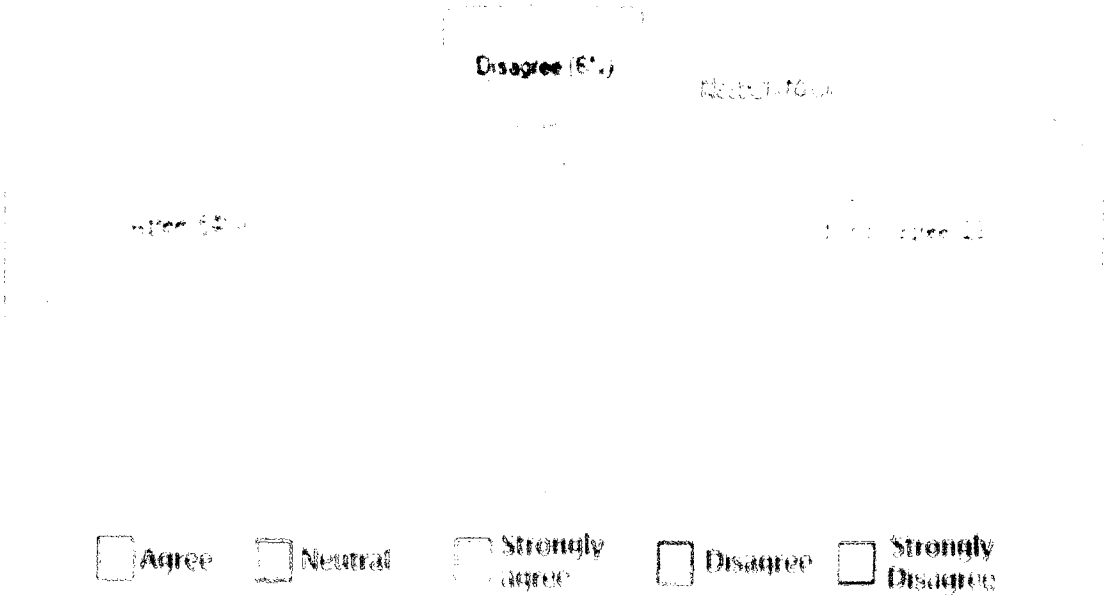

Figure 6-6. User satisfaction with the system in selecting media service

Fig. 6-7 illustrates the user's satisfaction based on visual quality of the final composed media stream. About 29% of users feel that the visual quality of the composed media was highly satisfactory, and 49% of the users agree that it is satisfactory. However, 20% of the users hold opposing views for the satisfaction score, as these users have video or image quality viewing experience.

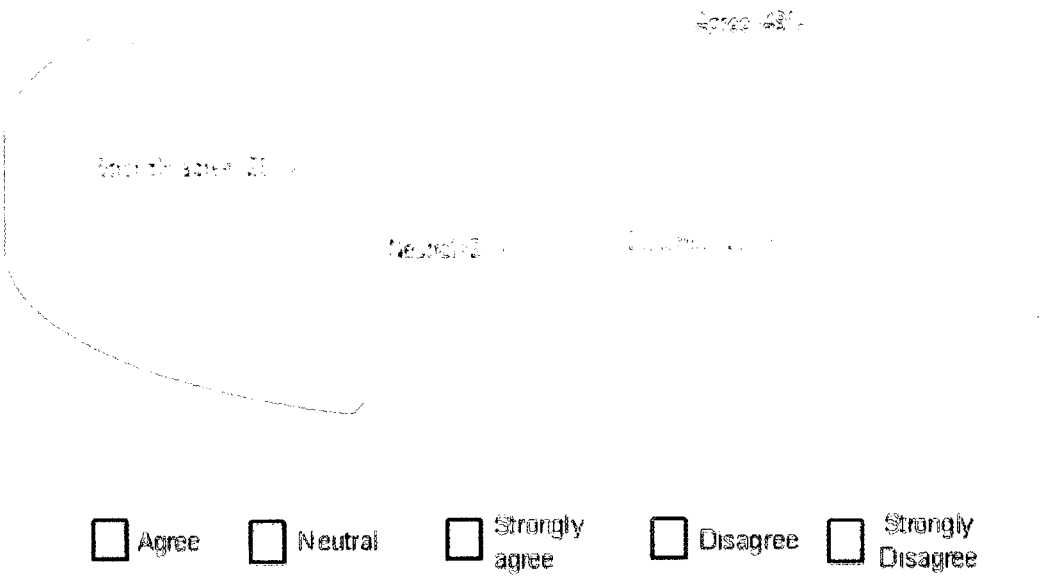

Figure 6-7. User acceptability on the evaluation of the visual quality of composed media

6.6. Summary

The combination of SOA and a biologically-inspired mechanism demonstrates the potential to overcome scalability, flexibility, and reusability of distributed multimedia delivery challenges. To the best of our knowledge, this is one of the first attempts that combine the SOA paradigm and a biologically-inspired mechanism into the multimedia domain and more specifically into multimedia service management for a ubiquitous environment. We evaluated and compared our proposed system through implementation and simulation. We discussed how our system scales under an increased number of composition requests. We also conducted some usability studies of the proposed system. Our future work includes exploring the robustness of the proposed framework, particularly under a dynamic ubiquitous environment.

Chapter 7. Conclusions and Future Works

This thesis describes the design and development of a framework for a biologically-inspired multimedia service management system. The framework focuses on selecting the appropriate repurposing services for a target composition by using a bio-inspired selection algorithm. Our bio-inspired-based service management approach achieves comparably better scalability and load balancing than the traditional (e.g. non bio-inspired selection algorithm) -based approach. To the best of our knowledge, this is one of the first attempts that combines the SOA paradigm and a biologically-inspired mechanism in the multimedia domain and more specifically, into multimedia service management for a ubiquitous environment. In this chapter, we summarize our major contributions and then we briefly discuss possible potential future research directions.

7.1. Contributions

In an attempt to design and develop a bio-inspired framework for multimedia service management to mitigate heterogeneity and scalability, this thesis makes the following contributions:

QoS-aware Service Selection Algorithms for Multimedia Repurposing Services
The proposed multimedia service management framework is an integrated framework/architecture with a focus on multimedia repurposing service selection. The service selection should be made in a way that it ensures that the delivered multimedia content is acceptable to the users, according to a desired level of QoS. Many available multimedia repurposing services provide similar functionalities, but with a different QoS. A challenge is to determine/find the best way to select appropriate repurposing services that fulfill the required QoS parameters. Thus, a service selection mechanism or algorithm is required to select the most suitable service, based on a metric (e.g. user satisfaction) that quantifies QoS. We implemented the previously mentioned selection algorithm which was described in chapter 4.

For a composite repurposing task, these services repurpose the requested media content in a chain fashion through different steps. Specifically, the output format of one repurposing service is supplied to the input format of the next repurposing service, until the requested content satisfies the end user's requirements and the outgoing target channel's requirements. Finally, the repurposed content is delivered to the clients from the last repurposing service. This method facilitates complex repurposing tasks by combining one or more simple repurposing services into different steps. Experimental results reveal that our approach performs complicated repurposing tasks by breaking them into a series of intermediate tasks. They are then executed in distributed repurposing proxies with a minimal degradation of the final delivered content quality.

Design and Development of an Ant-based Algorithm for Multimedia Service Selection

As mentioned in chapter 5, we presented one of the novel Biologically-inspired Repurposing Service Selection (BioReSS) approaches for a multimedia system. We adopted an AntNet-based repurposing service selection strategy in the context of real time multimedia streaming for ubiquitous users. With this proposed system, the main component, the BioReSS algorithm, facilitates in selecting repurposing services and optimal service paths based on the Quality of Experience. This is accomplished by considering both the user's satisfaction and the network level QoS. During the communication session, the algorithm uses biological foraging behaviour inspired from ant agents to find optimal service paths between the media sender and the destination. Experimental results demonstrate that the proposed algorithm provides significant performance gain over the traditional, state of the art selection algorithms and saves the average delay. Moreover, under an increased network load, the proposed bio-inspired selection algorithm outperforms the traditional approach.

Biologically-Inspired Framework for Multimedia Service Management

The objective of this research is to introduce a biologically-inspired multimedia service management framework through the composition of basic multimedia services including streaming services and different repurposing services. The biologically-inspired approach is used for collecting the QoS requirements from individual repurposing services, in order

to select the most suitable services for the desired composition process. The described framework is one of the few attempts toward bringing the combined potential of service-oriented architecture and the foraging behaviour of ant colony concepts into multimedia service management research.

We validated the feasibility of our proposed framework through implementation and simulation. We used implementation to demonstrate multimedia service compositions for target ubiquitous users in a live multimedia streaming application. We also simulated the multimedia service composition process, where the number of repurposing services was composed for a streaming service to get a customized stream for the user. Through simulation, we also presented some service management issues including scalability and load management. For the load management issue, we discussed how load is distributed when a system is overloaded. For the scalability issue, we discussed how our system scales under an increased number of composition requests. Finally, we conducted some usability studies to qualitatively evaluate the suitability of the proposed framework.

7.2. Future work

The proposed biologically inspired multimedia service management framework's level of performance draws attention to the significant potential of these bio-inspired approaches for multimedia service, web service and autonomic service management fields. However, there are still some aspects of this framework that can be improved and extended for further research. Some of these are outlined in the following as potential directions for future research works or topics:

Usability and Quality of Perception (QoP):

In this thesis, we have conducted usability studies in a small scale. It would be useful to have more usability study or other subjective evaluation to determine whether users actually find it constructive to their demands. One direction is to consider Quality of Perception with QoS. Quality of Perception (QoP) is determined from a user's subjective evaluation. QoP is the level of user satisfaction that is derived from the multimedia content. QoP has a direct impact of how well the multimedia service fulfills a user's need,

rather than the internal or the implied impact of QoS. Moreover, there could be cases when, in spite of a high QoS value, the associated QoP value may be low, and vice versa.

For example, it is possible to have excellent QoS but poor QoP, because the content available to the user is not considered to be satisfactory. On the other hand, even though the QoS may be poor, QoP could be excellent, as the user might still be able to watch a video of his interest, despite a lower level of quality (e.g. a low quality image of a beautiful lady may be more satisfying than a good quality image of a not-so-beautiful lady). Therefore, QoP could be a measurement of a user's level of satisfaction.

Multimedia AmI Service Management in Multi-sensor Network Environment

It is apparent that the goal of multimedia service management is to enable universal multimedia access [114] for heterogeneous multimedia services, which considers the streaming of multimedia content under different usage environments. Such a goal also drives the vision of Ambient Intelligence (AmI) [1], [34] that will eventually enable technology to become unobtrusive, embedded in a user's natural surroundings, adaptive to the response of people and their context, and predictive for human assistance.

In such a domain, rich multimedia content will be accessed ubiquitously through the limited capabilities of sensor node and sensor network. However, the key challenge will be to stream multimedia content efficiently and reliably through such limited capability multi-sensor network environments. Therefore, an important task in AmI paradigms will be the efficient management of multimedia streaming in such large-scale, highly distributed multi-sensor network environments. The technological aim is to bridge the available sensing infrastructure, and multimedia applications (e.g. streaming, healthcare) in an AmI environment.

Nature-inspired Autonomic Service Management

We have not addressed the issues regarding autonomic service management of real-time multimedia services. One of the challenges is the efficient, autonomous management of these real-time multimedia services over the future generation network. The autonomous service management is crucial for the self-management of real-time multimedia services. According to Ganek and Corbi of IBM [41], the autonomous or self-management aspects include self-optimizing, self-healing, self-configuring, and self-protecting. The existing

approach and framework contributes towards a system that is not fully autonomic in all four management aspects.

One of the future solutions of the said approach is inspired from the Bee colony metaphor. The allegory comprises how bee agents mimic functional services related to multimedia applications, in order to autonomously monitor and configure multimedia services. The objective of this future research will be to ensure complete autonomic behaviour of the four main management activities (configuration, repair, optimization and protection) of an autonomous system. Such direction could enable customization of the service for the current and future generation network conditions.

References

[1] E. Aarts, "Ambient Intelligence: A Multimedia Perspective.," *IEEE Multimedia*, vol. 11, no. 1, pp. 12-19, Jan.-Mar. 2004.

[2] "Advanced video coding for generic audiovisual services," ITU-T Recommendation H.264 March 2005.

[3] R. Agrawal, R. J. Bayardo, D. Gruhl, and S. Papadimitriou, "Vinci: A service-oriented architecture for rapid development of web applications," *Elsevier Computer Networks*, vol. 39, no. 5, pp. 523-539, 2002.

[4] I. Ahmed, X. Wei, Y. Sun, and Y. Q. Zhang, "Video transcoding: an overview of various techniques and research issues," *IEEE Trans. Multimedia*, vol. 7, no. 5, pp. 793-804, 2005.

[5] A. Alamry, M. Eid, and A. El Saddik, "Classification of the state of the art dynamic web services composition techniques," *Intl.l J. Web and Grid Services 2006*, vol. 2, no. 2, pp. 148-166, 2006.

[6] M. Amoretti, R. Bertolazzi, M. Reggiani, F. Zanichelli, and G. Conte, "Designing grid services for multimedia streaming in an e-learning environment," *Wiley Concurrency and Computation: Practice and Experience*, vol. 18, no. 8, pp. 911-923, 2006.

[7] J. G. Apostolopoulos, "Video communication and video Streaming II: error resilient video coding," MIT, [Online]. Available: http://www.mit.edu/~6.344/Spring2004/video_streaming2_2004.pdf.

[8] S. Ardon et al., "MARCH: a distributed content adaptation architecture," *Intl. J. Commun. Syst.*, vol. 16, pp. 97-115.

[9] O. Babaoglu et al., "Design patterns from biology for distributed computing," *ACM Trans.Autonomous and Adaptive Systems (TAAS)*, vol. 1, no. 1, pp. 26-66, 2006.

[10] D. Bertsekas and R. Gallager, *Data Networks*. NJ, USA: Prentice-Hall: Englewood Cliffs, December 1991.

[11] "BISON Project homepage," [Online]. Available:http://www.cs.unibo.it/bison/links.shtml,.

[12] P. Boonma and J. Suzuki, "BiSNET: A biologically inspired middleware architecture for self-managing wireless sensor networks," *Elsevier Computer Networks*, vol. 51, no. 16, pp. 4599-4616, 2007.

[13] D. Bruneo, M. Guarnera, A. Zaia, and A. Puliafito, "Grid based architecture for multimedia services management," in *Proc. 1st European Across Grids Conference*, Antiago de Compostela, Spain, February 2003.

[14] G. Canfora, M. D. Penta, R. Esposito, and M. L. Villani, "An approach for QoS-aware Service Composition Based on Genetic Algorithms," in *Proc. Genetic*

evolutionary computation conf. (GECCO'05), Washington DC. USA, June 25-29, 2005, pp. 1069-1075.

[15] L. Cao, M. Li, and J. Cao, "Cost-driven web service selection using genetic algorithm," in *Proc. Intl. Workshop Internet and network economics (WINE'05)*, X. Deng and Y. Ye (Eds.):, LNCS 3828, 2005, pp. 906-915.

[16] I. Carreras, I. Chlamtac, F. De Pellegrini, and D. Miorandi, "BIONETS: Bio-inspired networking for pervasive communication environments," *IEEE Trans.Vehicular Technology*, vol. 56, no. 1, pp. 218-229, 2007.

[17] L. Carrillo, J. L. Marzo, P. Vila, L. Fabrega, and C. Guadall, "A quality of Service Routing scheme for Packet switched Networks based on Ant Colony Behaviour," in *SPECTS 2004*, San Jose, California, USA, July 25 - 29, 2004.

[18] F. Chiang, R. Braun, and J. I. Agbinya, "Self-configuration of network services with biologically inspired learning and adaptation," *Springer Journal of Network and Systems Management*, vol. 15, no. 1, pp. 87-116, 2007.

[19] R. Chinnici, J.-J. Moreau, A. Ryman, and S. Weerawarana, "Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language, W3C Recommendation," [Online]. Available: http://www.w3.org/TR/wsdl20/, June 2007.

[20] L. Clement, A. Hately, C. V. Riegen, and T. Rogers, "UDDI Version 3.0.2, UDDI Spec Technical Committee Draft," OASIS UDDI Spec TC, 2004.

[21] F. Cohen, *FastSOA: The Way to Use Native XML Technology to Achieve Service Oriented Architecture Governance Scalability, and Performance*, 1st ed. San Fransisco, CA, USA: Morgan Kaufmann (Elsevier), November 2000.

[22] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 2nd ed.: MIT Press, Cambridge, 2001.

[23] F. Curbera, "Component contracts in service-oriented architectures," *IEEE Computer*, vol. 40, no. 11, pp. 74-80.

[24] A. K. Dey, "Understanding and Using Context," *Springer Personal and Ubiquitous Computing*, vol. 5, no. 1, pp. 4-7, 2001.

[25] G. Di Caro and M. Dorigo, "AntNet: Distributed Stigmergetic Control for Communications Networks," *Journal of Artificial Intelligence Research (JAIR)*, vol. 9, pp. 317-365, 1998.

[26] G. Di Caro and M. Dorigo, "Two ant colony algorithms for best-effort routing in datagram networks," in *Proc. 10th Intl. Conf. Parallel and Distrib. Comput. Syst.*, Las Vegas, Nevada, 1998.

[27] G. Di Caro, F. Ducatelle, and L. M. Gambardella, "AntHocNet: an adaptive nature-inspired algorithm for routing in mobile ad hoc networks," *Euro Trans. Telecom. (ETT)*, vol. 16, no. 5, pp. 443-455, 2005.

[28] G Di Caro, F Ducatelle, and L M Gambardella, "AntHocNet: an adaptive nature-inspired algorithm for routing in mobile ad hoc networks," *Euro Trans. Telecom. (ETT)*, vol. 16, no. 5, pp. 443-455, 2005.

[29] G. Di Caro, F. Ducatelle, L. M. Gambardella, and A. Rizzoli, "Building blocks from the biology for the design of algorithms for the management of modern

dynamic networks," *ECRIM News, Special Issue on Emergent Computing*, vol. 64, pp. 34-35, Jan 2006.

[30] E. W. Dijkstra, "A not on two problems in connection with graphs. Numer. Math. 1, 269–271 (1959)," *Numerische Mathematik*, vol. 1, pp. 269-271, 1959.

[31] E W Dijkstra, "A not on two problems in connection with graphs. Numer. Math. 1, 269–271 (1959)," *Numerische Mathematik*, vol. 1, pp. 269-271, 1959.

[32] Y. Ding, H. Sun, and K. Hao, "A bio-inspired emergent system for intelligent web service composition and management," *Elsevier Knowledge-Based Systems*, vol. 20, no. 5, pp. 457-465, 2007.

[33] M. Dorigo, G. Di Caro, and L. M. Gambardella, "Ant algorithm fordistributed discrete optimization," *Artificial Life*, vol. 5, no. 2, pp. 137-172, 1999.

[34] K. Ducatel, M. Bogdanowicz, F. Scapolo, J. Leijten, and J.- C. Burgelman, "Scenarios for Ambient Intelligence in 2010," [Online]. Available:ftp://ftp.cordis.lu/pub/ist/docs/istagscenarios2010.pdf, Seville, IST Advisory Group Final Report 2001.

[35] A. El Saddik and M. S. Hossain, "Multimedia content repurposing," in *Encyclopedia of Multimedia*, Borko Furht, Ed. Berlin, Germany: Springer Verlag, 2006.

[36] A. El Saddik and M. S. Hossain, "Multimedia Streaming for wireless communication," in *Encyclopedia of Wireless and Mobile Communications*, B Furht, Ed.: CRC Press, Taylor & Francis Group, 2007.

[37] I. Elsen, F. Hartung, U. Horn, M. Kampmann, and L. Peters, "Streaming technology in 3G mobile communication systems," *IEEE Comput.*, vol. 34, no. 9, pp. 46-52, 2001.

[38] A. Erradi, S. Padmanabhuni, and N. Varadharajan, "Differential QoS support in web services management," in *Proc. IEEE Intl. Conf. Web Services (ICWS '06)*, Chicago, U.S.A., September 2006, pp. 781-788.

[39] F. Fitzek, P. Seeling, and M. Reisslein, "Video streaming in wireless internet," in *Wireless Internet: Technologies and Applications*, A Salkintzis and A Poularikas, Eds.: CRC Press, pp. 11.1-11.41.

[40] T. Friedman, R. Caceres, and A. Clark, "RTP Control Protocol Extended Reports (RTCP XR)," [Online]. Available :http://www.networksorcery.com/enp/rfc/rfc3611.txt 2003.

[41] A. G. Ganek and T. A. Corbi, "The dawning of the autonomic computing era," *IBM Systems J.*, vol. 42, no. 1, pp. 5-18, 2003.

[42] Y. Gao, J. Na, B. Zhang, L. Yang, and Q. Gong, "Optimal Web Services Selection Using Dynamic Programming," in *Proc. Intl. Sympo. Computers and Communications (ISCC 2006)*, Sardinia, Italy, 26-29 June, 2006, pp. 365-370.

[43] M. R. Garey and D. S. Johnson, *Computers and Intractability:AGuide to the Theory of NP-Completeness*. San francisco: W.H. Freeman and Company, 1979.

[44] C. Gomila and P. Yin, "New features and applications of the H.264 video coding standard," in *Proc. of Intl. Conf. Info. Tech.: Research and Education*, 2003, pp. 6-10.

[45] M. Gudgin et al., "SOAP version 1.2 part," [Online]. Available: http://www.w3.org/TR/2007/REC-soap12-part1-20070427/, December 2007.

[46] X. Gu, K. Nahrstedt, and B. Yu, "Spidernet: An integrated peer-to-peer service 1 composition framework," in *Proc. 13th IEEE International Symposium on High performance Distributed Computing*, Honolulu, Hawaii, USA, 4–6 June 2004, pp. 110-119.

[47] J. R. Han et al., "Dynamic adaptation in an image transcoding proxy for mobile WWW browsing," *IEEE Personal Commun.*, vol. 5, no. 6, Dec 1998.

[48] M. Handley and V. Jacobson, "SDP: Session description protocol," IETF RFC 2327, 1998.

[49] M. Handley, C. Perkins, and E. Whelan, "SAP: Session announcement protocol," RFC 2974, 2000.

[50] R. Han and J. R. Smith, "Internet Transcoding for Universal Access," Multimedia Communications Handbook, ed. Jerry Gibson 1999.

[51] D. W. Hong and C. S. Hong, "A QoS management framework for distributed multimedia systems," *Int. J. Netw. Manag.*, vol. 13, no. 2, pp. 115-127, March 2003.

[52] M. S. Hossain, A. Alamri, and A. El Saddik, "A framework for qos-aware multimedia service selection for wireless clients," in *Proc. the 3rd ACM Workshop on Wireless Multimedia Networking and Performance Modeling (WMuNeP 07)*, Chania, Crete Island, Greece, October 22 - 22, 2007.

[53] M. S. Hossain, A. Alamry, and A. El Saddik, "Biologically-Inspired framework for Multimedia Service Management in Ubiquitous Environment," *Wiley Concurrency and Computation: Practice and Experience*, vol. 21, no. 11, pp. 1450-1466, August 2009.

[54] M. S. Hossain, A. Alamry, and A. El Saddik, "QoS-Aware Service Selection for Multimedia Transcoding," in *Proc. Intl. Instrum. Meas. Technol. Conf. (I2MTC'08)*, Victoria, BC, Canada, May 12-15, 2008.

[55] M. S. Hossain and A. El Saddik, "A Biologically Inspired Multimedia Content Repurposing System in Heterogeneous Network Environments," *ACM/Springer Multimedia Systems J.*, vol. 14, no. 3, pp. 135-144, 2008.

[56] M. S. Hossain and A. El Saddik, "Proxy-based Visual Content Repurposing using Selection Algorithm," in *Proc. Intl. Conf. Networking (ICN 2007)*, Martinique, French Caribbean, 22-28 April, 2007.

[57] M. S. Hossain and A. El Saddik, "Scalability measurement of a proxy-based multimedia content repurposing system," *Intl. J. Advanced Media and Commun.*, vol. 2, no. 3, pp. 267-287, 2008.

[58] X. Hu, T. Wang, and D. Li, "A New Approach of Color Quantization Based On Ant Colony Clustering Algorithm," in *Proc. Intl. conf. infor. Technol.: Coding and Computing (ITCC-05)*, Las Vegas, Nevada, USA, 2005.

[59] "IBM transcoding solution and service," [Online]. Available: http://www.research.ibm.com/networked_data_systems/transcoding/transcodef.pdf, Special Report.

[60] "Information Technology—JPEG 2000 Image Coding System–Part 3: Motion JPEG 2000," ISO/IEC 15 444-3, 2002.

[61] "Intermediaries and Transcoding," [online]. Available: http://www.almaden.ibm.com/cs/wbi/wbi-and-transcoding.html,.

[62] "ISO/IEC 14496-2 (MPEG-4), Information Technology – Coding ofAudio-Visual Objects, Part 2, Visual.," ISO/IEC JTC1, 1998.

[63] "ISO/IEC 21000-7:200x AMD/1, Information technology - Multimedia framework (MPEG-21) - Part 7: Digital Item Adaptation, AMENDMENT 1: DIA Conversions and Permissions," Final Proposed Draft Amendment 1 (FPDAM/1) 2005.

[64] "ITU-T H.263 (1998) Video Coding for Low Bitrate Communication, ," RITU-T Recommendation H.263, Version 2 November 1998.

[65] R. Jain, "Quality of experience," *IEEE MultiMedia*, vol. 11, no. 1, pp. 95-96, 2004.

[66] "Java Media Framework (JMF 2.1.1a) API," [Online]. Available: http://java.sun.com/javase/technologies/desktop/media/jmf/index.jsp, 2003.

[67] "Java SE Development Kit (JDK) 6 ," [Online]. Available: http://java.sun.com/javase/downloads/index.jsp 2007.

[68] J. Jin and K. Nahrstedt, "QoS-aware service management for component-based distributed applications," *ACM Trans.Internet Techn.2008*, vol. 8, no. 3, pp. 1-31, 2008.

[69] I. Jureta, S. Faulkner, Y. Achbany, and M. Saerens, "Dynamic web service composition within a service-oriented architecture," in *Proc. IEEE Intl. Web Services (ICWS 2007)*, Salt Lake City, UT, U.S.A, July 2007, pp. 304-311.

[70] S. Kalasapur, M. Kumar, and B. Shirazi, "Seamless service composition (SeSCo) in pervasive environments," in *Proc. the First ACM Intl. Workshop Multimedia Service Composition*, Hilton, Singapore, 2005, pp. 11-20.

[71] S. Kalasapur, M. Kumar, and B. Shirazi, "Seamless service composition (SeSCo) in pervasive environments," in *Proc. the First ACM Intl. Workshop Multimedia Service Composition*, Hilton, Singapore, 2005, pp. 11-20.

[72] M. Kloppmann, D. König, F. Leymann, G. Pfau, and D. Roller, "Business process choreography in WebSphere: Combining the power of BPEL and J2EE," *IBM Systems J.*, vol. 43, no. 2, pp. 270-296, 2004.

[73] C. Lee, P. Champrasert, and J. Suzuki, "iNet: A biologically-inspired adaptation mechanism for autonomic network applications," in *Proc. IEEE Workshop commun. and Networking*, Rochester, NY, 2005.

[74] Z. Lei, "Video Transcoding Techniques for wireless Video," School of Informatio Technology and Engineering, University of Ottawa, Ottawa, Ph.D. Thesis 2004.

[75] Z. Lei and N. D. Georganas, "Video transcoding gateway for wireless video access," in *Proc. IEEE Canadian Conf. Electrical and Computing Engineering (CCECE'03)*, Montreal, QC, Canada, 2003.

[76] Y. Liang, F. Chebil, and A. Islam, "Compressed domain transcoding solutions for MPEG-4 visual simple profile and H.263 baseline videos in 3GPP services and applications," *IEEE Trans. Consum. Electron.*, vol. 52, no. 2, pp. 507-515, 2006.

[77] Z. Liu, M. Z. Kwiatkowska, and C. Constantinou, "A biologically inspired QoS

routing algorithm for mobile ad hoc networks," in *Proc. Advanced Information Networking and Applications (AINA-2005)*, Taipei, Taiwan, 2005.

[78] W. Y. Lum and F. C. M. Lau, "On Balancing between Transcoding Overhead and Spatial Consumption in Content Adaptation," in *Proc. MobiCom'02*, Atlanta, Georgia, USA, Sep. 23-26 2002, pp. 239-250.

[79] A. Maheshwari, A. Sharma, K. Ramamritham, and P. Shenoy, "TransSquid:Transcoding and caching proxy for heterogeneous e-commerce environments," in *Proc. 12th IEEE Int. Workshop Research Issues in Data Engg*, San Jose, California, USA, 26 Feb. - 1 March, 2002, pp. 50-59.

[80] Z. M. Mao, H. W. So, B. Kang, and R. H. Katz, "'Network support for mobile multimedia," in *Proc. 11th Intl. Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV-2001),* New York, USA, 2001.

[81] J. M. Martínez, "MPEG-7 Overview," ISO/IEC JTCI/SC/29/WG11 N6828, [Online]. Available: http://www.chiariglione.org/MPEG/standards/mpeg-7/mpeg-7.htm., 2004.

[82] S. B. Musunoori and G. Horn, "Ant-based approach to the quality aware application service partitioning in a grid environment," in *Proc. IEEE Congress on Evolutionary Computation*, Vancouver, BC, Canada, 16–21 July 2006, pp. 2604-2611.

[83] "MySQL Enterprise," [Online]. Available: http://www.mysql.com/products/enterprise/server.html.

[84] K. Nahrstedt and W. T. Balke, "A taxonomy for multimedia service composition," in *Proc. 12th ACM Conf. Multimedia (ACM MM 04)*, New York, NY, USA, 10–16 October 2004, pp. 88-95.

[85] K. Nahrstedt and W. T. Balke, "Towards building large scale multimedia systems and applications: Challenges and status," in *Proc. the First ACM Intl.Workshop Multimedia Service Composition*, Hilton, Singapore, 2005, pp. 3-10.

[86] V. A. Nguyen and Y. P. Tan, "Efficient video transcoding between H.263 and H.264/AVC standards," in *Proc. IEEE Intl. Symposium on Circuits and Systems (ISCAS'05)*, Kobe, Japan, May 23-26, 2005.

[87] "The Ninja Project Enabling Internet-scale Services from Arbitrarily Small Devices," [Online]. Available: http://ninja.cs.berkeley.edu, 1997.

[88] M. F. Nowlan and M. B. Blake, "Intelligent agent communication and collaboration for web services management," in *Proc. the 16th IEEE Intl. Workshops Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE'07)*, Paris, France, 18-20 June 2007, pp. 18-23.

[89] K. Oida and M. Sekido, "An agent-based routing system for QoS guarantees," in *Proc. Intl. Conf. Systems, Man, and Cybernetics (SMC-99)*, Tokyo, Japan, 1999, pp. 833-838.

[90] M. P. Papazoglou and W. J. van den Heuvel, "Service oriented architectures: Approaches, technologies and research issues," *Intl. J. Very Large Data Bases (VLDB)*, vol. 16, no. 3, pp. 389-415, 2007.

[91] M. P. Papazoglou and W. J. van den Heuvel, "Web services management: A

survey," *IEEE Internet Computing*, vol. 9, no. 6, pp. 58-64, 2005.

[92] M P Papazoglou and W J van den Heuvel, "Web services management: A survey," *IEEE Internet Computing*, vol. 9, no. 6, pp. 58-64, 2005.

[93] K. Park and Y. Kim, "Analysis of AntNet routing scheme by using queueing model," *Computer Communications*, vol. 31, pp. 2951-2958, 2008.

[94] "Pushtotest testmaker 5.1," http://docs.pushtotest.com/docs/index.html, April 2008.

[95] B. Raman et al., "The SAHARA Model for Service Composition Across Multiple Providers," in *Proc. 1st Intl. Conf. Pervasive Computing*, Springer-Verlag, Lecture Notes in Computer Science; Vol. 2414, August 26 - 28, 2002, pp. 1-14.

[96] B. Raman and R. H. Katz, "Load Balancing and Stability Issues in Algorithms forService Composition," in *Proc. IEEE Joint Conf. Computer and Communications Societies, IEEE INFOCOM*, San Francisco, CA, USA, March 30 - April 3, 2003.

[97] A. Richards, G. Rogers, V. Witana, and M. Antoniades, "Mapping user level QoS from a single parameter," in *In 2nd IFIP/IEEE Intl. Conf. Manage. Multimedia Networks and Services*, Versailles, November 1998.

[98] J. Rosenberg et al., "SIP: session initiation protocol. IETF RFC 3261," 2002.

[99] R. Y. Rubinstein, "The Cross-Entropy Method for Combinatorial and Continuous Optimization," *Methodology and Computing in Applied Probability*, pp. 127-190, 1999.

[100] E. Sanchez-Nielsen, S. Martin-Ruiz, and J. Rodriguez-Pedrianes, "An open and dynamical service oriented architecture for supporting mobile services," in *Proc. 6th Intl. Conf. Web Engineering (ICWE '06)*, New York, NY, U.S.A, 2006, pp. 121-128.

[101] H. Schulzrinne, "The Real-Time Streaming Protocol (RTSP), http://www.cs.columbia.edu/~hgs/rtsp/," 1998.

[102] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications," [Online]. Available: ftp://ftp.rfc-editor.org/in-notes/rfc3550.txt, 2003.

[103] I. Shin and K. Koh, "Hybrid Transcoding for QoS Adaptive Video-on-Demand Services," *IEEE Trans.Consumer Electronics*, vol. 50, no. 2, 2004.

[104] "Simple Service Discovery Protocol (SDP)," [Online]. available: http://www.upnp.org/download/draft-cai-ssdp-v1- 03.txt, 2003.

[105] G. Singh, "Guest Editor's Introduction: Content Repurposing," *IEEE Multimedia*, vol. 11, no. 1, pp. 20-21, Jan 2004.

[106] J. R. Smith, R. Mohan, and C. S. Li, "Scalable Multimedia Delivery for Pervasive Computing," in *Proc. ACM Multimedia' 99*, Orlando, FL, USA, Oct.30 - Nov.5, 1999.

[107] R. Stewart, "Stream Control Transmission Protocol (SCTP)," http://tools.ietf.org/html/rfc4960 RFC 4960, September 2007.

[108] Z. Subing and L. Zemin, "A QoS routing algorithm based on ant algorithm," in *Proc. IEEE Intl. Conf. Communications (ICC'01)*, Helsinki, Finland, June 11-14, 2001, pp. 1581 - 1585.

[109] S. Su, C. Zhang, and J. Chen, "An Improved Genetic Algorithm for Web Services," in *Proc. DAIS'07, LNCS 4531*, J. Indulska and K. Raymond (Eds.), 2007, pp. 284-295.

[110] J. Suzuki and T. Suda, "A middleware platform for a biologically inspired network architecture supporting autonomous and adaptive applications," *IEEE J. Selected Areas Commun*, vol. 23, no. 2, pp. 249-260, 2005.

[111] S. Tadrus and L. Bai, "A QoS network routing algorithm using multiple pheromone tables," in *Proc. IEEE Intl. Conf. on Web Intelligence (WIC'01)*, Halifax, Canada, 13-16 Oct. 2003, pp. 132- 138.

[112] D. Tsesmetzis, I. Roussaki, and E. Sykas, "QoS-aware service evaluation and selection," *European J. Operational Research*, vol. 191, no. 3, pp. 1101-1112, 2008.

[113] F. L. Verdi, M. F. Magalhaes, E. Cardozo, E. R. M. Madeira, and A. Welin, "A service oriented architecture-based approach forinterdomain optical network services," *Springer J. Network and Systems Mangt.*, vol. 15, no. 2, pp. 141-170.

[114] A. Vetro, C. Christopoulos, and T. Ebrahami, "Editorial," *IEEE Signal Processing, Special issue on Universal Multimedia Acsess*, vol. 20, no. 2, March 2003.

[115] A. Vetro and C. Timmerer, "Digital Item Adaptation: Overview of Standardization and Research Activities," *IEEE Trans. Multimedia*, vol. 7, no. 3, pp. 418-426, June 2005.

[116] A. Vetro, J. Xin, and H. Sun, "Error resillence video transcoding for wireless communications," *IEEE Wirel. Commun*, vol. 12, no. 4, pp. 14-21, 2005.

[117] M. Vukovic, "Context aware service composition," University of Cambridge, Cambridge, UK, Ph D Thesis October 2007.

[118] M. Vukovic and P. Robinson, "Application Modeling for Context Awareness. Building and Evaluating Ubiquitous System Software.," *IEEE Pervasive Computing Mag.*, vol. 3, no. 3, pp. 59-59, July-October 2004.

[119] H. F. Wedde, M. Farooq, and Y. Zhang, "Beehive: an efficient fault tolerant routing algorithm inspired by honey bee behavior," in *ANTS -2004*, LNCS 3172. Springer, Berlin, 2004.

[120] O. Wittner, P. E. Heegaard, and B. E. Helvik, "Swarm-based distributed search in the amigos environment," Department of Telematics, Norwegian University of Science and Technology, AVANTEL Tech. Rep. Dec. 2003.

[121] W. Wu, G. Fox, H. Bulut, A. Uyar, and T. Huang, "Service oriented architecture for VoIP conferencing," *Intl. J. Commun. Syst.*, vol. 19, no. 4, pp. 445-461, 2006.

[122] D. Xu and K. Nahrsted, "Finding service paths in a media proxy network," in *Proc. Multimedia Computing Networking (MMCN 02)*, San Jose, California, USA, Jan. 18-25, 2002.

[123] D. Xu, D. Wichadakul, and K. Nahrstedt, "Resource-aware configuration 1 of ubiquitous multimedia services," in *Proc. IEEE Intl. Conf. Multimedia and Expo (ICME-2000)*, New York, NY, USA, July 30 - August 2, 200, pp. 851-854.

[124] M. Younas, I. U. Awan, and D. Duce, "An efficient composition of web services with active network support," *Elsevier Expert Systems with Applications*, vol. 31,

no. 4, pp. 859-869, 2006.

[125] T. Yu, Y. Zhang, and K. -J. Lin, "Efficient algorithms for web services selection with end-to-end QoS constraints," *ACM Trans. Web (TWEB)*, vol. 1, no. 1, pp. 1-26, 2007.

[126] J. Zhang and J. Y. Chung, "A soap-oriented component-based framework supporting device-independent multimedia web services," in *Proc. 4th Intl. Sympo. Multimedia Software Engineering (ISMSE '2002)*, Newport Beach, CA, USA, 11-13 December 2002, pp. 40-47.

[127] S. Ziane and A. Melouk, "A swarm intelligent multi-path routing for multimedia traffic overmobile ad hoc networks," in *Proc. ACM workshop Quality of Service and Security in Wireless and Mobile Networks*, Montreal. Quebec, Canada, 2005.

Appendix A: The Streaming Service WSDL

```
<?xml version="1.0" encoding="UTF-8"?><definitions
xmlns="http://schemas.xmlsoap.org/wsdl/"
xmlns:tns="http://services.withServices.repurposing.mcrlab.uottawa.ca/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" targetName-
space="http://services.withServices.repurposing.mcrlab.uottawa.ca/"
name="StreamingServiceService">
  <types></types>
  <message name="getStream">
    <part name="arg0" type="xsd:string"></part>
    <part name="arg1" type="xsd:string"></part>
    <part name="arg2" type="xsd:int"></part>
    <part name="arg3" type="xsd:int"></part>
    <part name="arg4" type="xsd:string"></part>
    <part name="arg5" type="xsd:int"></part>
    <part name="arg6" type="xsd:int"></part>
  </message>
  <message name="getStreamResponse">
    <part name="return" type="xsd:boolean"></part>
  </message>
  <portType name="StreamingService">
    <operation name="getStream" parameterOrder="arg0 arg1 arg2 arg3
arg4 arg5 arg6">
      <input message="tns:getStream"></input>
      <output message="tns:getStreamResponse"></output>
    </operation>
  </portType>
  <binding name="StreamingServicePortBinding"
type="tns:StreamingService">
    <soap:binding style="rpc" trans-
port="http://schemas.xmlsoap.org/soap/http"></soap:binding>
    <operation name="getStream">
      <soap:operation soapAction=""></soap:operation>
      <input>
        <soap:body use="literal" name-
space="http://services.withServices.repurposing.mcrlab.uottawa.ca/"></s
oap:body>
      </input>
      <output>
        <soap:body use="literal" name-
space="http://services.withServices.repurposing.mcrlab.uottawa.ca/"></s
oap:body>
      </output>
    </operation>
  </binding>
  <service name="StreamingServiceService">
    <port name="StreamingServicePort" bind-
ing="tns:StreamingServicePortBinding">
```

```
      <soap:address loca-
tion="http://137.122.91.89:8080/MyVideo"></soap:address>
    </port>
  </service>
</definitions>
```

Appendix B: Repurposing Service WSDL

```
<?xml version="1.0" encoding="UTF-8"?><definitions
xmlns="http://schemas.xmlsoap.org/wsdl/"
xmlns:tns="http://server.services.withServices.proxy/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" targetNames-
pace="http://server.services.withServices.proxy/"
name="RepurposingServiceService">
  <types></types>
  <message name="isReady">
    <part name="arg0" type="xsd:string"></part>
    <part name="arg1" type="xsd:string"></part>
    <part name="arg2" type="xsd:int"></part>
    <part name="arg3" type="xsd:int"></part>
  </message>
  <message name="isReadyResponse">
    <part name="return" type="xsd:boolean"></part>
  </message>
  <message name="receivingStream">
    <part name="arg0" type="xsd:string"></part>
    <part name="arg1" type="xsd:string"></part>
    <part name="arg2" type="xsd:string"></part>
  </message>
  <message name="receivingStreamResponse">
    <part name="return" type="xsd:boolean"></part>
  </message>
  <portType name="RepurposingService">
    <operation name="isReady" parameterOrder="arg0 arg1 arg2 arg3">
      <input message="tns:isReady"></input>
      <output message="tns:isReadyResponse"></output>
    </operation>
    <operation name="receivingStream" parameterOrder="arg0 arg1 arg2">
      <input message="tns:receivingStream"></input>
      <output message="tns:receivingStreamResponse"></output>
    </operation>
  </portType>
  <binding name="RepurposingServicePortBinding"
type="tns:RepurposingService">
    <soap:binding style="rpc" trans-
port="http://schemas.xmlsoap.org/soap/http"></soap:binding>
    <operation name="isReady">
      <soap:operation soapAction=""></soap:operation>
      <input>
        <soap:body use="literal" names-
pace="http://server.services.withServices.proxy/"></soap:body>
      </input>
      <output>
        <soap:body use="literal" names-
pace="http://server.services.withServices.proxy/"></soap:body>
```

```
      </output>
    </operation>
    <operation name="receivingStream">
      <soap:operation soapAction=""></soap:operation>
      <input>
        <soap:body use="literal" names-
pace="http://server.services.withServices.proxy/"></soap:body>
      </input>
      <output>
        <soap:body use="literal" names-
pace="http://server.services.withServices.proxy/"></soap:body>
      </output>
    </operation>
  </binding>
  <service name="RepurposingServiceService">
    <port name="RepurposingServicePort" bind-
ing="tns:RepurposingServicePortBinding">
      <soap:address loca-
tion="http://137.122.89.112:8081/H263_352"></soap:address>
    </port>
  </service>
</definitions>
```

Appendix C: Video Bandwidth File

```
<bandwidthTable xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
xsi:noNamespaceSchemaLocation=
http://www.site.uottawa.ca/research/MCRLab/videoBandwidthTableDefinitio
n.xsd'>
  <row resolutionWidth="352" resolutionHeight="288" bitRate="307200">
    <Format>MPEG-4</Format>
  </row>
  <row resolutionWidth="352" resolutionHeight="288" bitRate="153600">
    <Format>H.263</Format>
  </row>
  <row resolutionWidth="352" resolutionHeight="240" bitRate="1572864">
    <Format>MPEG-1</Format>
  </row>
  <row resolutionWidth="720" resolutionHeight="480" bitRate="15728640">
    <Format>MPEG-2</Format>
  </row>
  <row resolutionWidth="352" resolutionHeight="288" bitRate="1048576">
    <Format>H.261</Format>
  </row>
  <row resolutionWidth="720" resolutionHeight="480" bitRate="10485760">
    <Format>JPEG/NITF</Format>
  </row>
  <row resolutionWidth="720" resolutionHeight="480" bitRate="7864320">
    <Format>WAVELETS</Format>
  </row>
  <row resolutionWidth="352" resolutionHeight="240" bitRate="1572864">
    <Format>Cinepak</Format>
  </row>
  <row resolutionWidth="352" resolutionHeight="288" bitRate="1048576">
    <Format>Indeo</Format>
  </row>
  <row resolutionWidth="352" resolutionHeight="288" bitRate="1572864">
    <Format>MPEG-4</Format>
  </row>
  <row resolutionWidth="320" resolutionHeight="240" bitRate="1572864">
    <Format>JPEG_RTP</Format>
  </row>
  <row resolutionWidth="320" resolutionHeight="240" bitRate="1572864">
    <Format>video/jpeg</Format>
  </row>
  <row resolutionWidth="352" resolutionHeight="288" bitRate="1048576">
    <Format>video/h263</Format>
  </row>
</bandwidthTable>
```

Appendix D: Proxy Profile

```
<proxyNode xmlns:xsi='http://www.w3.org/2001/XMLSchema-
instance'xsi:noNamespaceSchemaLocation='
http://www.site.uottawa.ca/research/MCRLab/proxyNodeDefinition.xsd'>
      <RepurposingServiceDirectory>
http://www.site.uottawa.ca/research/MCRLab/AntTranscProxy/AONIT09Dir.xm
l</ RepurposingServiceDirectory>
      <IP>137.122.89.112</IP>
      <pathPort>22222</pathPort>
      <antPort>20202</antPort>
      <neighborList>
            <neighbor neighborDir="h:/MCR/senderProfile.xml"
IP="137.122.91.89" dataPort="22222" pathPort="22222" antPort="23000"
bandwidth="500000" type="in"/>
            <neighbor neighborDir=
"http://www.site.uottawa.ca/research/MCRLab/AntTranscoderProxy2/proxy2.
xml" IP="137.122.89.112" dataPort="22235" pathPort="22235" ant-
Port="20235" bandwidth="500000" type="out"/>
            <neighbor neighborDir="h:/MCR/receivProxy.xml"
IP="137.122.91.223" dataPort="22240" pathPort="23000" antPort="23000"
bandwidth="813700" type="out"/>
      </neighborList>
</proxyNode>
```

Appendix E: Repurposing Service Profile

```
<RepurposingService xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:noNamespaceSchemaLocation=http://www.site.uottawa.ca/research/MCRLa
b/repurposingV1.xsd'>

<input>
      <inputName>video/jpeg</inputName>
      <iproperty>
            <property>resolution</property>
            <value>320x240</value>
      </iproperty>
</input>

<input>
      <inputName>video/h263</inputName>
      <iproperty>
            <property>resolution</property>
            <value>128x96</value>
      </iproperty>
</input>

<output>
      <outputName>video/h263</outputName>
      <oproperty>
            <property>resolution</property>
            <value>352x288</value>
      </oproperty>
      <visualQuality>
            <videoFormat>H263_RTP</videoFormat>
            <Frame height="288" width="352" />
</visualQuality>
</output>

<RepurposeHardware>
      <processor>
      <processorName>Athlon 1700+</processorName>
      <cyclesPer>1500000</cyclesPer>
      </processor>
      <memory>512</memory>
</ RepurposeHardware >
<proxyNode location="
http://www.site.uottawa.ca/research/MCRLab/AntTranscProxy
/AONIT09.xml"/>
</ RepurposingService>
```

Appendix F – Server Profile

```
<Profile xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
xsi:noNamespaceSchemaLocation='file:/H:/MCR/senderProfileDefinition.xsd
'>
  <output>
    <outputName>video/jpeg</outputName>
    <oproperty>
        <property>resolution</property>
        <value>320x240</value>
    </oproperty>
    <visualQuality>
        <videoFormat>JPEG_RTP</videoFormat>
        <Frame height="240" width="320"/>
    </visualQuality>
  </output>

  <neighborProxy>
    <neighbor neighborDir="
http://www.site.uottawa.ca/research/MCRLab/BellfordTransc/AONIT10.xml"
IP="137.122.89.112" dataPort="22222"
        pathPort="22222" antPort="20202" bandwidth="3400000"/>
  </neighborProxy>
</Profile>
```

Appendix G: Pseudo code of the RP-AL -2 Algorithm

Algorithm 3: The Non Bio-inspired Repurposing Path Selection Algorithm (RP-AL-2)

1) Let s_i^e be user's satisfaction beween nodes u and v
   It may note that s_i^e may also also represent link satification
   for all nodes that have edges between nodes u, and v
2) **if** *graph size is 1* **then**
   TERMINATE (FAILURE)
3) **foreach** *repurposing service node* $v \in V(u,v)$ **do**
   a) satisfaction $[s_i]$=0
   b) satisfaction of all other adjacent service node $s_i = \infty$
4) **foreach** $i = 1 \cdots (|v|-1)$ **do**
   foreach *Edge* $e(u,v) \in E$ **do**
   a) Relax the graph
   b) Calculate user's satisfaction for each service node subject to QoS constraints
      Optimize(s_i, Q_{in}, Q_{out}, QoS constraints)
      i) $s_i[v] = \max\{s_i[v], s_i[v] + s_i^e(u,v)\}$
      if $s_i[v] \succ s_i[u]$ **then**
   $s_i[v] = \max\{s_i[v], s_i[v] + s_i^e(u,v)\}$
   Add repurposing node u to the repurposing path
5) Try and relax graph if required
   if *graph is relaxable* **then**
   TERMINATE (NEGATIVE LOOP)
6) Print the repurposing path from the Media Sender to Receiver.