



**HAL**  
open science

# QoE-based Content Distribution Network architecture

Hai Anh Tran

► **To cite this version:**

Hai Anh Tran. QoE-based Content Distribution Network architecture. Other [cs.OH]. Université Paris-Est, 2012. English. NNT : 2012PEST1108 . tel-02886838

**HAL Id: tel-02886838**

**<https://theses.hal.science/tel-02886838>**

Submitted on 1 Jul 2020

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



UNIVERSITÉ PARIS-EST  
ÉCOLE DOCTORALE MSTIC

Thèse de doctorat  
Spécialité : Informatique

**Hai Anh TRAN**

Titre :

**Une architecture orientée Qualité d'Expérience pour les  
réseaux de diffusion de contenus**

*QoE-based Content Distribution Network Architecture*

Thèse dirigée par :

Directeur de Thèse : Abdelhamid MELLOUK  
Co-encadrant : Said HOCEINI

Date de soutenance : Mardi 11 Décembre 2012

Lieu : Département Réseaux et Télécoms & Laboratoire LiSSi, Vitry sur Seine

**Jury :**

**Rapporteurs :**

Olivier FESTOR, INRIA Nancy  
Xavi MASIP-BRUIN, Université Polytechnique de Catalogne, Barcelone, Espagne

**Examineurs :**

Michel DIAZ, LAAS-CNRS Toulouse  
Patrick GALLINARI, UPMC  
Guy PUJOLLE, UPMC  
William RANG, IPLabel

---

**Titre :** Une architecture orientée Qualité d'Expérience pour les réseaux de diffusion de contenus.

**Résumé :** Avec le développement rapide des technologies réseaux et la constante progression des besoins des utilisateurs, les mécanismes traditionnels liés à la notion de Qualité de Service (QoS) ne permettent plus de garantir des niveaux élevés de performance et ne suffisent plus à satisfaire les attentes des utilisateurs, principalement dans le contexte des réseaux orientés contenus (Content-Oriented Network - CON). Les travaux développés dans le cadre de cette thèse s'intéressent à un nouveau paradigme, appelé qualité de l'expérience (Quality of Experience), permettant de mesurer le niveau de perception de l'utilisateur des services qui lui sont fournis. Au regard des contraintes de plus en plus dynamiques dans les applications actuelles, un mécanisme basé sur une boucle de décision -contrôle/commande- pour le routage adaptatif a été développé dans lequel la QoE usager tient le rôle de métrique régulatrice. Le domaine considéré concerne les réseaux de distribution de contenus pour lesquels de nouveaux algorithmes de routage et de sélection de serveurs ont été développés sur la base de l'apprentissage par renforcement. Les algorithmes proposés ont été validés et comparés aux approches traditionnelles à la fois en mode simulé qu'en mode réel. Les résultats obtenus montrent clairement l'intérêt de ce nouveau type de métrique pour un meilleur rendu des applications usagers.

**Mots-clés:** Qualité d'Expérience, Qualité de Service, Réseau de diffusion de contenus, Routage, Sélection de serveurs, Apprentissage par Renforcement.

**Title:** QoE-based Content Distribution Network architecture.

**Abstract:** Nowadays, with the rapid growth of high quality network applications, the Quality of Service mechanisms, which are used to guarantee required levels of performance to data flows in network, are not enough anymore to satisfy end-users, mostly in the context of Content-Oriented Networks (CON) that is used to redesign the current Internet architecture to accommodate content-oriented applications and services. We propose in this thesis a new system that introduces the Quality of Experience (QoE), a new paradigm that represents the user's perception, in an adaptive control/command network routing chain. Our proposed system has been applied to a Content Distribution Network (CDN) architecture. We focused on two main functions of a CDN, the routing and the server selection method, in order to improve the QoE level at end-user side. For the routing method, an algorithm named QQAR (QLearning QoE-based Adaptive Routing) has been proposed. This algorithm uses the technique of Reinforcement Learning based on Markov models and take into account as a metric the QoE perceived by the end user. Regarding the server selection method, we proposed a server selection algorithm that is motivated by the Multi-Armed Bandit problem. The two proposed algorithms are validated and compared to traditional approaches using a simulation model and a real platform.

**Keywords:** Quality of Experience, Quality of Service, Content Distribution Network, Routing algorithm, Server selection, Reinforcement Learning, Multi-Armed Bandit.

**Unité de recherche :** Laboratoire Images, Signaux et Systèmes Intelligents (LISSI), EA 3956, UPEC.

# Contents

<b>Acknowledgment</b>	<b>8</b>
<b>Introduction</b>	<b>11</b>
<b>1 Overview of QoE and CDN</b>	<b>16</b>
1.1 Introduction . . . . .	16
1.2 Quality of Experience . . . . .	17
1.2.1 QoE concept . . . . .	17
1.2.2 Importance of QoE . . . . .	19
1.2.3 QoE metrics . . . . .	21
1.2.4 QoE measurement methods . . . . .	25
1.2.5 QoS/QoE relationship . . . . .	27
1.2.6 Impact of Networking on QoE . . . . .	30
1.3 Content Distribution Network . . . . .	35
1.3.1 Routing layer . . . . .	38
1.3.2 Meta-routing layer . . . . .	44
1.4 Conclusion . . . . .	50
<b>2 Proposed routing algorithm for CDN architecture: QQAR</b>	<b>51</b>
2.1 Introduction . . . . .	51
2.2 Reinforcement Learning and Q-Routing . . . . .	53
2.2.1 Mathematical model of Reinforcement Learning . . . . .	55
2.2.2 Value functions . . . . .	57
2.2.3 Q-Learning . . . . .	59
2.2.4 Q-Routing . . . . .	60
2.2.5 Related works and motivation . . . . .	61
2.3 QQAR routing algorithm . . . . .	62
2.3.1 Formal Parametric model . . . . .	63
2.3.2 QQAR algorithm . . . . .	64
2.3.3 Learning process . . . . .	66
2.3.4 Simple use case-based example of QQAR . . . . .	69
2.3.5 Selection process . . . . .	75
2.4 Conclusion . . . . .	77

<b>3</b>	<b>Proposed server selection algorithm</b>	<b>78</b>
3.1	Introduction . . . . .	78
3.2	Multi-armed Bandit formalization . . . . .	80
3.2.1	MAB paradigm . . . . .	81
3.2.2	Applications of MAB . . . . .	84
3.2.3	Algorithms for MAB . . . . .	86
3.3	Server selection schemes . . . . .	91
3.4	Our proposal for QoE-based server selection method . . . . .	94
3.4.1	Proposed server selection scheme . . . . .	94
3.4.2	Proposed UCB1-based server selection algorithm . . . . .	97
3.5	Conclusion . . . . .	98
<b>4</b>	<b>Experiments</b>	<b>99</b>
4.1	Introduction . . . . .	99
4.2	Testbed for measuring QoE . . . . .	99
4.2.1	Video parameters . . . . .	100
4.2.2	Network parameters . . . . .	101
4.2.3	Application . . . . .	101
4.3	Experimental setup . . . . .	102
4.3.1	Platform setup . . . . .	102
4.3.2	Simulation setup . . . . .	107
4.4	QQAR protocol: experimental results . . . . .	117
4.4.1	Experimental setup . . . . .	117
4.4.2	Average MOS score . . . . .	118
4.4.3	Convergence time . . . . .	124
4.4.4	Capacity of convergence and fault tolerance . . . . .	127
4.4.5	Control overheads . . . . .	129
4.4.6	Packet delivery ratio . . . . .	130
4.5	Server selection algorithm: experimental results . . . . .	131
4.5.1	Simulation results . . . . .	131
4.5.2	Real platform results . . . . .	137
4.6	QQAR and proposed server selection: experimental results . . . . .	139
4.7	Conclusion . . . . .	141
<b>5</b>	<b>Conclusion</b>	<b>143</b>
5.1	Contributions . . . . .	144
5.2	Perspective . . . . .	145
<b>6</b>	<b>Version française abrégée</b>	<b>147</b>
	<b>Appendices</b>	<b>159</b>

<b>A</b>	<b>QoE measurement method</b>	<b>159</b>
A.1	Introduction . . . . .	159
A.2	PSQA-based QoE measurement method . . . . .	160
A.3	Parameters affecting QoE . . . . .	162
A.3.1	Network parameter . . . . .	162
A.3.2	Video characteristics . . . . .	163
A.3.3	User equipment . . . . .	164
A.3.4	Psychological factors . . . . .	165
A.4	Training Random Neural Network . . . . .	166
A.4.1	Random Neural Network tool . . . . .	166
A.4.2	RNN Learning . . . . .	167

# List of Figures

1	Adaptive control/command model . . . . .	14
1.1	Classification of Objective Measurement Methods . . . . .	24
1.2	Quality chain in an e2e service . . . . .	28
1.3	QoE and QoS in Quality Management module . . . . .	28
1.4	General mapping curve between QoS and QoE . . . . .	29
1.5	CDN service . . . . .	35
1.6	General Architecture of CDN system . . . . .	36
1.7	Two layers in a CDN architecture . . . . .	37
2.1	CDN processes . . . . .	52
2.2	Reinforcement Learning model . . . . .	54
2.3	Example of Markov process with 3 states and 2 actions . . . . .	56
2.4	Q-Learning-based model . . . . .	63
2.5	QQAR protocol . . . . .	65
2.6	QQAR diagram . . . . .	67
2.7	Learning process . . . . .	68
2.8	Sample of execution for QQAR . . . . .	69
2.9	Phase 1: with MOS = 4, the Q-values are updated . . . . .	70
2.10	Phase 2: with MOS = 3, the Q-values are updated . . . . .	70
2.11	Phase 3: with MOS = 1, the Q-values are updated . . . . .	71
2.12	Phase 4: with MOS = 0, the Q-values are updated . . . . .	71
2.13	Phase 5: with MOS = 2, the Q-values are updated . . . . .	72
2.14	Network before the failure . . . . .	73
2.15	QQAR detects exactly the link downed . . . . .	73
2.16	Average MOS score of QQAR-1 and QQAR-2 . . . . .	74
2.17	Irregular 6x6 grid network . . . . .	75
3.1	DNS-based server selection scheme . . . . .	92
3.2	QoE-based CDN model . . . . .	95
3.3	CDN model with four components . . . . .	95
3.4	QoE-based server selection scheme . . . . .	96
4.1	Conceptual diagram for testbed . . . . .	100

4.2	Evaluation testbed . . . . .	102
4.3	Subjective evaluation form . . . . .	103
4.4	Implemented Platform . . . . .	104
4.5	Functional diagram . . . . .	106
4.6	Incident timeline for router X . . . . .	110
4.7	Network topologies . . . . .	112
4.8	Poisson rate modulated by Markov chain . . . . .	115
4.9	Markov modulated Poisson process . . . . .	116
4.10	MOS score for 3AreasNet topology . . . . .	119
4.11	MOS score for 6x6GridNet topology . . . . .	120
4.12	MOS score for NTTnet topology . . . . .	121
4.13	Initialization phase for 3AreasNet topology . . . . .	124
4.14	Initialization phase for 6x6GridNet topology . . . . .	125
4.15	Initialization phase for NTTnet topology . . . . .	125
4.16	6x6GridNet topology after breaking down the R18-R19 link . . . . .	127
4.17	MOS score after topology change . . . . .	128
4.18	User perception in different load levels . . . . .	129
4.19	Control overhead . . . . .	130
4.20	Packet delivery ratio . . . . .	130
4.21	User perception results of four policies: UCB1, FASTEST, CLOSEST and Round Robin . . . . .	132
4.22	Initialization phase . . . . .	133
4.23	User perception results of three policies: $\epsilon$ -greedy, softmax and UCB1 . . . . .	134
4.24	Number of choice of iBox 55 . . . . .	136
4.25	Experimental results over the real platform . . . . .	138
4.26	MOS score of QCDNA for 45 hours of simulation . . . . .	140
4.27	MOS score of QCDNA for 5 minutes of initialization phase . . . . .	141
6.1	Le service de CDN . . . . .	149
6.2	L'algorithme de routage QQAR . . . . .	151
6.3	Le processus d'apprentissage . . . . .	152
6.4	Le modèle de CDN avec quatre composants . . . . .	154
6.5	Schéma pour le banc d'essai . . . . .	155
6.6	Real Platform . . . . .	156
6.7	Topologies réseaux en mode simulé . . . . .	157
A.1	PSQA-based QoE measurement model . . . . .	160

# Acknowledgment

Apart from the efforts of myself, the success of any thesis depends largely on the encouragement and guidelines of many others. This page is perhaps the most difficult one to write of this thesis as those who have had an influence on this work are numerous.

First and foremost, I would like to thank to my supervisor, Prof. Abdelhamid Melouk, for the valuable guidance and advice. He inspired me greatly to work in this thesis. I feel motivated and encouraged every time I attend his meeting. His willingness to motivate me contributed tremendously to my work. I would like to thank him for showing me a good direction for my research. I can't say thank you enough for his tremendous support and help.

I also wish to express my sincere gratitude to my co-supervisor, Mr. Said Hoceini, for guidance and encouragement in helping me to carry out my thesis. Without his encouragement and guidance this thesis would not have materialized.

I would like to show my greatest appreciation to Mr. Julien Perez who helped me with technical discussions, proposal and efficient suggests.

Deepest gratitude are also due to the members of LISSI lab (Signals Images & Intelligent Systems Laboratory): Sami Souihi, Benjamin Cogrel, Brice Augustin and other persons, without whose knowledge and assistance this thesis would not have been successful.

Finally, an honorable mention goes to my family and friends for their understandings, endless patience and supports on me in completing this thesis. Without helps of the particular that mentioned above, I would face many difficulties while doing this.

# Publications

## Journals:

1. H.A. Tran, A. Mellouk, S. Hoceini, B. Augustin, S. Zeadally, “*A Dynamic QoE Routing System for User-Centric Applications*”, in review for minor revisions in Transactions on Emerging Telecommunications Technologies (ETT), 11 pages, 2012.

## Book chapter:

1. H. A. Tran, A. Mellouk, S. Hoceini, “*QoE-based Routing for Content Distribution Network Architecture*”, in “*Media Networks: Architectures, Applications and Standards*”, CRC Press. Ed., (H. Moustafa et S. Zeadally Eds), pp. 256-271, 2011.

## Lecture Note Series:

1. H.A. Tran, A. Mellouk, “*User to User QoE routing system*”, LNCS Springer, vol 6649, pp. 362-373, 2011.
2. H.A. Tran, A. Mellouk, “*QoE Model Driven for Network Services*” LNCS Springer, vol. 6074, pp. 264-277, 2010.

## International Conferences:

1. H.A. Tran, A. Mellouk, S. Hoceini, J. Perez, ” *User-centric Content Distribution Network Architecture*”, IEEE ICUMT12, International Conference on Ultra Modern Telecommunications, Russia, 3-5 October, 2012.
2. H. A. Tran, A. Mellouk, S. Hoceini, B. Augustin, “*Global State-Dependent QoE based Routing*”, IEEE ICC 2012, International Conference on Communications, Canada, 5-9 June, 2012.
3. H. A. Tran, A. Mellouk, S. Hoceini, S. Souihi, “*User QoE-based adaptive routing system for future Internet CDN*”, IEEE CNIT 2012, China, January 10-14, 2012.
4. H. A. Tran, A. Mellouk and S. Hoceini, “*QQAR routing protocol User to User adaptive routing based on QoE*”, IEEE ISPS’11, Algiers, Algeria, April 25-27, 2011

- 
5. H. A. Tran, A. Mellouk and S. Hoceini, “*QoE Content Distribution Network for Cloud Architecture*”, in Conference on Network Cloud and Applications, NCCA-2011, Toulouse, France, November 21-23, 2011.
  6. H. A. Tran and A. Mellouk, “*User to User QoE routing system*”, 9th IFIP International Conference on Wired/Wireless Internet Communications (WWIC 2011), Vilanova i la Geltrú, Barcelona, Spain, June 15-17, 2011, (LNCS Springer, vol 6649).
  7. H. A. Tran and A. Mellouk, “*Real-Time State-Dependent routing based on User Perception*”, In IEEE International Conference on Communications and Information Technology – ICCIT 2011, Jordan, March 29-31, 2011.
  8. H. A. Tran and A. Mellouk, “*QoE model driven for network services*”, In IFIP 8th International Conference on Wired/Wireless Internet Communications (WWIC 2010), Sweden, June 1-3, 2010, (LNCS Springer, vol. 6074).

**National Conferences:**

1. H.A. Tran, A. Mellouk, S. Hoceini, “*Un routage inductif pour une meilleure garantie de service dans les réseaux de communication*”, Conférence Internationale Francophone d’Automatique (CIFA 2012), Grenoble, France, 4-6 Juillet, 2012.
2. H.A. Tran, A. Mellouk, S. Hoceini, “*La perception de l’usager comme paramètre pilote dans le routage inductif des applications temps réel au sein d’un réseau opérateur*”, Journées Doctorales / Journées Nationales MACS du GDR CNRS, Marseille, France, 2011.
3. H.A. Tran, A. Mellouk, S. Hoceini et J. Marty “*User to User QoE routing system*”, École d’été du GDR CNRS ASR, Toulon-Hyères, France, du 13 au 18 Juin 2010.

**Invited paper:**

1. A. Mellouk, S. Hoceini, B. Augustin, N.A. Saadi, H.A. Tran, S. Souihi, “*Adaptive Control based on Quality of Experience for Multimedia Communications*”, Vol.7, No.3, IEEE E-Letter of Multimedia Communications Technical Committee Communications Society, 2012.

**Submitted:**

1. H.A. Tran, A. Mellouk, J. Perez, S. Hoceini, S. Zeadally, “*QoE-based Server Selection for Content Distribution Networks*”, submitted to ACM Transaction on Computers.
2. H.A. Tran, A. Mellouk, S. Hoceini, B. Augustin, J. Marty, S. Fowler, “*QoE-based concepts and models for end-to-end network services based on the user perception*” submitted to ACM Transactions on Multimedia Computing, Communications and Applications.

# Introduction

Due to the emergence of different kinds of communication and networking technologies (core networks with intra-domain and inter-domain challenges, access networks, aggregation networks, spontaneous networks, Internet of objects, etc) and the current and foreseen proliferation of different and specific types of services supported by these technologies (real-time services, IPTV, VoD, social networking, gaming, etc.), traditional end-to-end Quality of Service (QoS) strategies implemented by network providers are not sufficient to match user expectations. Based on a convergence of network technologies, the Next Generation Network (NGN) is being deployed to carry high quality video and voice data. However, it is important for the operators and providers not to forget sight of the reason for this new infrastructure: to provide network service that user want to use. Accomplishing this idea means assuring positive experience of end-users. The operators always try to deliver network services in the most cost and resource efficient manner with ensured user perception. Therefore, service providers are switching the focus from traditional QoS to user satisfaction, which is the overall success of a network from the user perspective. According to Daniel R. Scoggin, “*The only way to know how customers see your business is to look at it through their eyes*”. In fact, the convergence of network technologies has been driven by the converging needs of end-users. The perceived end-to-end quality becomes one of the main goals required by users that must be guaranteed by the network operators and the Internet service providers, through manufacturer equipment. This is referred to as the Quality of Experience notion (QoE) that becomes commonly used to represent user perception. The QoE is not a technical metric, but rather a concept consisting of all elements of a user’s perception of the network services. To ensure good end-users perception, the service providers have encountered QoE challenging issues, such as QoE monitoring, measurement, diagnosis and management. Therefore, all of network functions such as admission control, access network selection, routing, resource allocation, transmission control and server selection are expected to be adaptive to user QoE. Among them, we focus on the routing and server selection function.

In the past, the network operators used QoS notion as a method to improve their ability to accommodate data flows and to ensure QoS guarantees from the network. We take an example of routing issue to show problems exist. The QoS routing allows the network operator to determine a path for every data flow through the network with sufficient resources to meet the flow’s requirements. These requirements are based just on

QoS parameters such as end-to-end delay, bandwidth, loss rate, etc. Thus, identifying a path capable of guarantying these requirements needs the accuracy of the knowledge/information of the availability of network resources. In other words, a successful QoS routing depends on the accurate information of the available resources in the network. Most existing routing algorithms assume the availability of precise state information. However, in many cases this information may be inaccurate and insufficient. The advertising updates in a dynamic network environment cause considerable imprecision for state information. So, the wide issue is the QoS routing with inaccurate state information [Guerin and Orda, 1999].

The second issue related to the problem of complexity. To meet the requirement of source node, apart from the resource parameters (e.g. e2e delay, loss rate, bandwidth, etc.), a QoS routing system may try to optimize also other criteria such as network utilization, carried load, number of flows successfully routed, etc. These criteria can influence the routing process but satisfying all of them is not easy. For example, the network provider may identify a path satisfying the requirement of the data flow, but cannot use it because it is too expensive in terms of the amount of consumed resources. Optimizing all of these criteria is really a NP-complete problem. In reality, an e2e QoS model with more than two non correlated criteria is NP-complete (this problem is proved in [Wang and Crowcroft, 1996] and [Michael and Johnson, 1979]). In order to resolve this problem of complexity, network provider should integrate the QoE into network systems.

In addition, QoS network system addresses the different challenges associated with the accurate computation and estimation of network parameters. Previously, most project/approaches were network oriented and only considered the Service Level Agreement (SLA) between network operators and service providers. However, this perspective is not sufficient anymore with the new types of content on the Internet: the high quality multimedia content. The latter has dramatically increased in the recent years. Users are downloading high quality video streams, watching videoconferencing and broadcasting their own video streams to the Internet. These sensitive traffics in terms of e2e delay, loss rate, impose tighter constraints on the end-users for a reliable content delivery, opposed to agreements of classic SLA. Thus, the interest of both network operators and service providers on providing a satisfied perception to the end-users becomes more important than just considering technical network parameters.

Hence, the QoE has been introduced to resolve these problems above. This new paradigm promotes the importance of the user in a whole system. With regard to QoE definitions, according to the International Telecommunication Union (ITU), the QoE is defined as the overall satisfaction of a service perceived subjectively by end users. It is a measurement of how well a service satisfying the end-user's expectations about the service and the complete e2e system effects. By [Hassan et al., 2010], QoE includes relevant factors such as user's internal state (e.g., predispositions, expectations, needs, motivation and mood), the characteristics of the designed system (e.g., complexity, purpose, usability, functionality and relevance) and the context (or environment) within which the service is experienced (e.g., organizational/social setting, meaningfulness of

the activity, voluntariness of use). Today, despite the meaningful progress made, QoE is always a significant interest and continuing challenge of academy and research industry.

The trend towards using QoE in network management systems is explained by the appearance of emerging multimedia services in the Internet such as mobile online gaming, immersive environments, 3D virtual world, social networking, IPTV applications. The service quality depends not only on technical QoS parameters but also on end-user perception. The challenge between network operators becomes satisfying end-users who play an important role in quality management model. In fact, the clients are not only “*the spectator*” but also the key “*actor*” of network management. The Quality of Experience (QoE) has been introduced to face this new challenge.

Improving the quality of the service as perceived by the users is a significant challenge of network services. In this context, the goal is to minimize the customer churn yet and to maintain their competitive edge. Basing on this kind of quality trade-off, QoE [Shaikh et al., 2010a] [Brooks and Hestnes, 2010] [Tran et al., 2010] has been introduced as a combination of user perception, experience and expectations instead of using only technical parameters mainly gathering from QoS measurement. The QoE management takes into account the needs and the desires of the subscribers when using network services, while the concept of QoS only attempts to objectively measure the service delivered. As a critical measure of the end-to-end performances at the service level from the user’s perspective, the QoE is a kind of metric for the system design and engineering processes that can only be measured dynamically at the very end of any transmission activity.

For these reasons, we focus on the idea of how to integrate the QoE into a control/-command chain in order to construct an adaptive network system. More precisely, in the context of Content-Oriented Networks (CON) that is used to redesign the current Internet architecture to accommodate content-oriented applications and services, we apply the end-to-end QoE model to a Content Distribution Network architecture that is composed mainly of two layers: the *routing* and the *meta-routing* layer. The routing layer takes into account the routing process to route data packet between clients and servers. In this layer, we proposed a routing algorithm based on the user perception. On the other hand, the meta-routing layer includes various modules such as *server placement*, *cache organization* and *server selection*. We focused on the third one: the server selection, which selects the most appropriate replica server among a servers set.

### **Contributions**

As mentioned previously, the QoE becomes today a decisive criterion for the network management success. We propose an adaptive control model to carry the QoE notion. Our idea is to design an adaptive loop for improving the service quality of a network system in taking into account the end-user feedback. In our model (Fig. 1), the *QoE measurement module* evaluates the user feedback and sends the result to *the control module*. The latter analyses the received results, makes a decision and sends it to *the command module* that will apply the chosen action to the network system. This loop model refers to an adaptive system based on the e2e QoE.

To enhance the practical significance of the proposed model in figure 1, we applied

the e2e QoE model above to a real network infrastructure: the Content Distribution Network (CDN). The latter has been introduced to improve the delivery media content service. It is an effective solution to improve network quality and to optimize the resource utilization. The main idea of CDN operation is to move the content from the origin server to the servers close to end-users, namely replica servers. Based on the idea of applying the e2e QoE model to a CDN, we designed a QoE-based Content Distribution Network Architecture (QCDNA) whose the goal is to maintain the best QoE in a dynamic network environment.

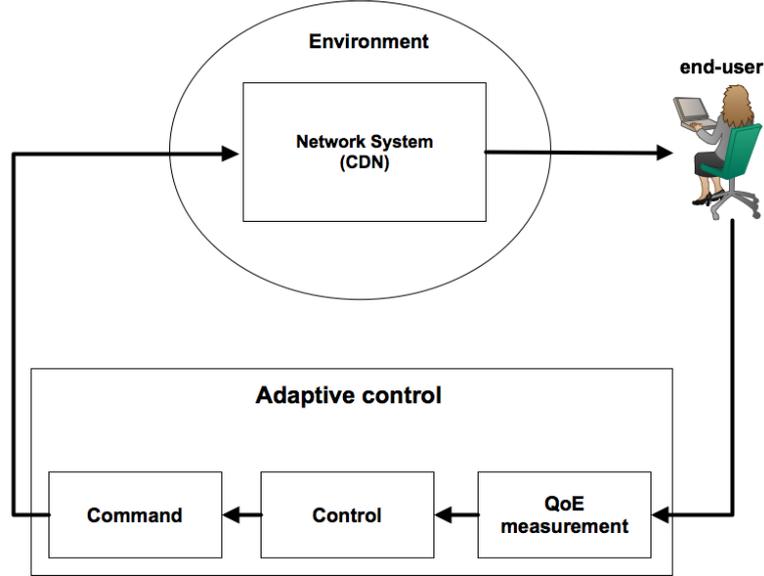


Figure 1: Adaptive control/command model

More precisely, we propose two approaches for two separated layers in a QoE-based CDN models: the *routing layer* and the *meta-routing layer*.

Our contributions include:

1. QoE-based Content Distribution Network architecture (QCDNA): We applied the E2E QoE adaptive control/command model to a CDN architecture. QCDNA will take into account the QoE feedback of end-users and put it into this adaptive control chain for maintaining a sufficient E2E QoE level of entire system. In addition, it can avoid issues of QoS-based systems mentioned above.
2. QoE measurement method: We implement a real testbed to create a measurement method that is a hybrid between objective and subjective method.
3. QoE-based routing protocol: We propose a QoE-aware routing algorithm that is based on a Reinforcement Learning. We implement it to the routing layer of the CDN architecture.

4. QoE-based server selection: We propose a QoE-based server selection method in formalizing selection process to multi-armed bandit formalization. This proposal focuses on the meta-routing layer of the CDN architecture.
5. Combination of QQAR and the new server selection scheme: We combined the proposed routing protocol (for routing layer) and the proposed server selection scheme (for meta-routing layer) to introduce a novel QoE-based CDN Architecture (QCDNA)

### Organization of the thesis

In chapter 1, we survey briefly the Quality of Experience notion. First, we talk about the QoE concept, what the QoE is and how important it is in the networking context. We present also some standard organizations studying the QoE, such as ITU-T, Broadband-forum and TeleManagement-forum. Then, we talk about the relationship between the QoS, the widely used notion, and the QoE, the new view of network operators. In this section, we talk also the impact of networking on the QoE. Also in this chapter, as this thesis focuses on integrating the QoE metric into a Content Distribution Network architecture, we present the CDN in describing their two main layers: the routing and the meta-routing layer. For the routing layer, we present some routing approaches with different algorithms. For the meta-routing layer, we talk about three main modules: *server placement*, *cache organization* and *server selection*.

Chapter 2 presents the proposed routing algorithm, called QQAR. In this chapter, we first present the mathematical RL model and the Q-Learning algorithm based on RL theory. We describe also the Q-Routing, a routing algorithm that based on the Q-Learning, and consider Q-Routing as the fundamental theory to construct our routing protocol QQAR. After, the QQAR algorithm is presented in detail with the description of their two main processes: learning process and selection process.

In chapter 3, we describe the muti-armed bandit formalization we used to propose a server selection algorithm to apply to the meta-routing layer of a CDN architecture. Then, we explain how we have formalized the muti-armed bandit to our selection algorithm. We cited the advantages to explain our choice compared with traditional approaches.

In chapter 4, we present our work on experiments. First, we describe the testbed we used to generate a dataset for supervised learning phase of the QoE measurement method. Then, we present the experimental setup with setting parameters as well as experimental scenario. Finally, we show and analyze the obtained results.

# Chapter 1

## Overview of QoE and CDN

We begin this chapter with the definitions and explanation of importance of QoE in the network context. The QoS/QoE relationship and some QoE measurement methods will be shown also. After, we will give an overview of the Content Distribution Network architecture and its functionality in two main layers: routing and meta-routing layer. The state of art of different approaches for each layer will be described in detail.

### 1.1 Introduction

Nowadays, Next Generation Networks (NGN) trend includes deploying the Fixed Mobile Internet Convergence (FMIC) and migrating the traditional telecom networks to Internet Protocol (IP) technology. NGN will employ multiple networking technologies for the best network service. Network operators and service providers try to simplify network operations and management in order to offer multiple services over a single network. While NGN network experts are going to employ a common network layer protocol in core networks to accomplish the current network services, the access networks will use numerous technologies, such as WLAN, WPAN, Ethernet cable, DSL, 2G/3G, LTE, WiMAX, UWB, optical fiber, etc. to meet the diversified requirements of end-users [Zhang and Ansari, 2011]. Using a network environment with multiple-operators and multiple-network, end-users expect to use a heterogeneous wired and wireless high-bandwidth ubiquitous network access and diversified services. Hence, the end-users have multitude of offers that make service prices decrease. Consequently, the competition between network providers increases. So, the customer is today in a strong position, being able to select between different competing providers. Apart from similar pricing schemes, which are an useful decision aid for users, their choices are also influenced by expected and experience quality. So, the interest of service providers in how users perceive usability, reliability, quality and price-worthiness has increased. As a result, the QoE notion has been introduced to help service providers to take into account the user perception and satisfaction.

As mentioned in the Introduction chapter, we have proposed an adaptive E2E QoE control model and applied it to a real infrastructure, the CDN, to construct a QoE-based

Content Distribution Network architecture (QCDNA). Before presenting our QCDNA, we need to clarify what QoE is and how important it is in the network context. After, it is also necessary to describe the CDN architecture to show the need of applying an E2E QoE model. Therefore, in this chapter, we survey the QoE notion in the networking context. Then, we talk about the CDN architecture with two main layers where the proposed algorithms are implemented.

## 1.2 Quality of Experience

### 1.2.1 QoE concept

The era of user-centered network service has begun, where user's perception plays a key role in the entire system. The quality perceived by end-users when using a service is called Quality of Experience. Taking the image processing as an example, the question of which features are included in a multimedia service becomes how well such features are addressed and which impact they have on end-users. The data usually pass through many processing phases before delivering to end-users host. These phases include: acquisition, processing, coding, transmission, enhancement, decoding, etc. The quality of multimedia data may be affected by each of these phases. This affection decreases the end-to-end QoE. There are also some other factors that impact on the QoE such as usability, human factors, context.

Traditionally, service providers carry only the notion of Quality of Service, which objectively measures and guarantees characteristics of service from the provider perspective. QoS metrics take into account system components characteristics, not the human perception. Over the last few years, QoE has brought a new look to end-to-end quality in multimedia systems. QoE represents aspects related to not only subjective perception, but also user behavior and needs, appropriateness, context, usability and human factors of the delivered content.

The first concern refers to what the QoE is. There are different definitions of QoE. According to International Telecommunication Union [ITU, 1865], QoE is defined as *"The overall acceptability of an application or service, as perceived subjectively by the end-user"*. In fact, QoE is more a qualitative measure than a quantitative one. By [PC Mag, 1994], QoE is defined as *"a subjective measure of performance in a system. QoE relies on human opinion and differs from quality of service (QoS), which can be precisely measured"*. According to [Fiedler et al., 2010a], *if a user does not feel that (s)he is getting "value for money", i.e., good-enough QoE for what (s)he is paying for, (s)he might be ready to abandon the particular service provider and/or network operator in question.*

To clarify the above definitions, many standards organizations have studied the concept of QoE. We cite here three relevant standards organizations:

1. **ITU-T:** Study Group 12 is the main group studying the QoS and QoE in the ITU Telecommunication Sector (ITU-T). Some of important recommendations are the following:

- *ITU-T G.1010* [Recommendation, 2001a] constructs a model for multimedia QoS categories from an end-user viewpoint. This recommendation studies not only the key parameters impacting the user such as the delay, delay variation and information loss, but also the performance targets for different applications that would meet user expectations. The intent of this Recommendation is to provide guidance on the key factors that influence Quality of Service (QoS) from the perspective of the end-user.
  - *ITU-T G.1030* [Recommendation, 2005] proposes a framework to get the IP network performance, evaluates the application performance and assesses user satisfaction with the end-to-end performance. The recommendation shows a model of Web browsing. This guideline has 3 steps: network performance evaluation, application performance evaluation and perceptual performance evaluation. The third step has introduced the user perception. This model is similar to the E-model [Recommendation, 2011b] that makes a link from end-user experience of network layer to application layer.
  - *ITU-T G.1070* [Recommendation, 2007] provides a model that assesses video-phone quality to the QoE/QoS planners. This model is used to guarantee the user satisfaction with end-to-end service quality. It has three functions: *video quality assessment*, *speech quality assessment* and *multimedia quality integration functions*. The model is based on specific terminals and environments.
2. **Broadband-Forum** [Broadband Forum, 1995]: Besides their focus on triple play applications (i.e., video, audio and best-effort data), Broadband forum also takes into account the QoE/QoS relationship. With similar idea on QoE definition of ITU-T, they consider QoE as the measure of overall performance from user's perspective. They try to define a clear relationship between these two concepts so that one could evaluate QoE from QoS measurements. Furthermore, they propose the idea of determining the required network parameters with a target of QoE. The Broadband forum provides a complete methodology and some specific metrics (e.g, delay, bandwidth, loss rate, etc.), but they do not show how QoE would be affected if such requirements are not met.
  3. **TeleManagement-Forum** [TM forum, 1988]: Forum uses Service Level Agreement(SLA) to assess QoE. TM Forum considers Key Quality Indicator (KQIs) and Key Performance Indicators (KPIs) as measurements of perceived quality rather than network performance. They focus also on QoE management and try to find an exact definition of end-to-end QoE in entire network system.

All these standards organizations focus on providing a clear and uniform definition of QoE and QoS. Moreover, they try also to establish the relationship between QoS and QoE to contribute to the QoE measurement and management.

The classification is necessary for understanding any concepts. The QoE is also not an exception. We classify this notion into three different classes (Tab. 1.1): *QoE as QoS extension*, *QoE for management system* and *QoE as hedonistic concept*. We have chosen

this method of classification because these three approaches can cover the whole QoE notion. Indeed, what makes us interested in considering the user perception concept are:

- QoE as QoS extension: refers to the fact that QoE is generally viewed as an extension of a known concept, QoS. From a historical point of view, the QoS metric came first to designate a set of techniques to ensure the routing of network traffic such as voice or sensitive multimedia applications. Since then, QoS has been highlighting the performance improvement of network systems. QoE concept appeared more recently. As presented as a comprehensive approach to quality (measured end to end), QoE directly affects end-users and considered as an extension of QoS.
- QoE for management system: A management system of network services has responsibility for ensuring availability, performance and efficiency of all modules in the framework. Over the last few decades, the network management systems just based on technical parameters such as delay time, loss-rate, bandwidth, etc. However, satisfying all these metrics does not ensure the success of the system. Service providers unintentionally forgot the most important factor: the user experience, which plays a critical role in the whole system. That is why we consider QoE as the key factor for network management system.
- QoE as hedonistic concept: In the context of QoE, Hedonistic concept is related to the pleasure and satisfaction of the end-user when using service. QoE represents the overall end-user satisfaction with a service. Even though QoE is not always numerically quantifiable, it is the most important single factor to assess the user experience. Hence, QoE can be viewed as a hedonistic concept.

Table 1.1: Three QoE approaches

QoE approaches	Descriptions
Extension of QoS	QoE is considered as an extension of QoS. That reflects the QoS influences on end-users perception.
Key factor for management system	QoE is considered as a key factor for management system. It reflects the user's perception, and has vital importance for management tasks.
Hedonistic concept	QoE determines the user satisfaction when using a service. The key idea of hedonistic concept is that pleasure is the only factor that has intrinsic value. This concept, in the context of QoE is used to evaluate service in terms of how much pleasure and how little pain end-users.

### 1.2.2 Importance of QoE

E2E-QoE (End-to-end QoE) measures the degree of satisfaction of services perceived by the end-user both objectively and subjectively. It has become a major issue for

telecommunication companies because their businesses are highly dependent on customer satisfaction, and thus, the average revenue per user can only be increased by value-added services which requires taking into account the complete end-to-end system effects. E2E-QoE involves the entire end-to-end path chain including the Quality of Design (QoD)<sup>1</sup> of terminal equipment and QoS supported by network components. In the context of communication services, E2E-QoE is influenced by content, network, device, application, user expectations and context of use. Other external factors that can also have an impact on E2E-QoE include user's terminal hardware, mobility, and the importance of the application.

Technical metrics such as packet delay, loss rate, jitter can represent QoS notion, and for some of the high quality conferencing applications, these QoS metrics become certainly important. However, the QoS parameters are not the only crucial factor that determines service quality. Therefore, QoE aims to estimate or measure what happens in the customer's mind, i.e. service quality perceived by the customers. As we know, simply improving technical metrics by itself would never be enough to prevent customers from giving up the service. Instead, operators should use some metrics which can react in advance and improve customer perception, i.e. QoE. Consequently, network providers who provide good QoE have a significant competitive advantage, while providers who ignore the importance of QoE may suffer unnecessary costs. [Soldani et al., 2006] confirms this point in showing statistical results: around 82% of customer defections (churning) are due to frustration over the product or service and the incompetence of the provider and 90% of users will not complain before defecting. The results indicate the necessity of a strategy to manage and improve QoE proactively for customer satisfaction.

Today, with the highly competitive environment of various service providers, end-users have the option to choose the best one. As a consequence, the service availability is not enough. In this context, the service providers must improve those services in order that users can enjoy a rich experience with a reasonable price. In addition, data traffic is today no longer just a best-effort traffic. To meet user needs and requirements, many applications of data traffic need high quality of service. Thus, many standards organizations (e.g, International Telecommunication Union (ITU), Broadband Forum, TeleManagement-Forum as mentioned in previous section) have proposed many classifications of services and QoS parameters. However, they do not clearly capture the real QoE. The most important difference between QoS and QoE is that while QoS just focuses on measuring performance from network perspective, the QoE is how well a service meets the user's expectations. Therefore, the challenge for service providers is to take into account both QoE and QoS. In many cases, acceptable QoS does not always infer acceptable QoE. For example, in a session of VoIP [Batteram et al., 2010], the packet overflow is monitored at the aggregate queue level. Assume that an edge router has an inadequate buffer size. As the VoIP traffic volume is low compared to other types of traffic, the aggregate buffer statistics gives an acceptable QoS. Despite of this, the QoE in this case is poor because of dropped packets. So, the VoIP subscribers are unsatisfied.

---

<sup>1</sup>The QoD is the quality perceived by end-user when using end-devices. In other words, QoD is how end-user is satisfied with design and usability of end-devices.

### 1.2.3 QoE metrics

Even though QoE is difficult to predict due to its subjective nature, measurement of QoE is necessary for the assessment of overall service quality. Therefore, one needs an effective strategy to measure QoE as realistically as possible. One way to achieve this is top-down approach. In the top-down approach, the factors that influence user perception are determined at the beginning. Then, this information is used to generate operating requirements. Finally, a method to constantly evaluate these factors and improve them is created.

One can group the expectations obtained in the initial phase into two main categories: *reliability* and *quality* [Soldani et al., 2006]. Quality in this case refers to the quality of the service and application software features. Key Performance Indicators (KPI) for the *reliability dimension* in the network service include:

- Service availability (anywhere)
- Service accessibility (anytime)
- Service access time (service setup time)
- Continuity of service connection (service retainability)

KPI for the *quality dimension* include:

- Quality of session
- Bit rate
- Bit rate variation
- Active session throughput
- System responsiveness
- End-to-end delay
- Delay variation

The KPIs are measured for each service in order to calculate an estimate of each service score. Based on the comparison between these scores and the initial “QoE targets” which was set before measuring each KPI, the values of each of these metrics will be mapped to assess the level of impact on the actual QoE.

Some KPI will be totally irrelevant in one case while being the most important in another. It all depends on the type of service application the user is running. For example, important characteristics of voice and video conferencing are time relation preserving, guaranteed resource allocation without re-transmissions and real time traffic. In applications like Web browsing, remote server access and interactive gaming, however,

characteristics such as request/response pattern of use, transmission delay, dynamic resource allocation and best-effort traffic should be considered more.

Another way to classify QoE metrics is based on *subjective* or *objective* method. Subjective methods are conducted to obtain information on the quality of multimedia services using opinion scores, while objective methods are used to estimate the network performance using models that approximate the results of subjective quality evaluation. Subjective parameters assess how audio and video data are collected from users, namely, what is their opinion about the quality of sequences of audio/video. A typical example of qualitative metrics (subjective) is the Mean Opinion Score (MOS) [Recommendation, 2006], which was first standardized by the International Telecommunication Union (ITU). With this metric, the system quality is assessed subjectively from the mean opinion of a panel or group of users. Each user rates the service on a five-point scale: 1=Bad, 2=Poor, 3=Fair, 4=Good, and 5=Excellent. The MOS is used typically in subjective tests of audio listening, where a number of people assess the quality of the audio. Another example of a qualitative metric is the R-factor [Recommendation, 2011b], which can be used in a manner similar to MOS. Subjective evaluation of speech quality uses the R-factor in voice transmission models. Examples of subjective tests were summarized in ITU-R Rec. BT.500-11 [Recommendation, 2002] and ITU-T Rec.P.910 [Recommendation, 1999] and cited in [Chikkerur et al., 2011].

- Double Stimulus Continuous Quality Scale (DSCQS) [Recommendation, 2002]: In this test, the evaluators watch twice the reference and processed video. They then rate the video quality on a continuous quality scale of 0–100 (with 0 being *Bad* and 100 *Excellent*).
- Single Stimulus Continuous Quality Evaluation (SSCQE) [Recommendation, 2002]: the evaluators watch only the processed video sequence with long duration (about 20–30 minutes). They use a slider to rate the instantaneous perceived quality on the DSCQS scale of *bad* to *excellent*.
- Absolute Category Rating (ACR) [Recommendation, 1999]: It is similar to SSCQE with only the processed video being shown to the evaluators. However, it is very common to include hidden references that are rated in the same way as the processed [Rohaly et al., 2000]. The evaluators rate the overall video quality using a discrete five-level scale ranging from *bad* to *excellent*.
- Pair Comparison (PC) [Recommendation, 1999]: In this method, the evaluators watch test clips from the same scene but under varying conditions. These clips are paired in all possible combinations and screened to the evaluators for preference judgment about each pair.
- Double Stimulus Impairment Scale (DSIS) [Recommendation, 2002]: Unlike the DSCQS, in the DSIS, the evaluators have to watch only one presentation sequence. The evaluators rate the sequences on a discrete five-level scale ranging from *very annoying* to *imperceptible*.

However, subjective QoE measurement is time consuming, tedious, expensive and is not particularly applicable in a production environment. Instead of directly collecting quality information, objective methods can be used to estimate the MOS score. This way, the scores reflecting real-time user perception can be estimated, and immediately, the service providers can utilize the information to maintain the network condition thereby providing best QoE.

The objective quality measurement methods have been classified into the following five main categories [Chikkerur et al., 2011] [Takahashi et al., 2008]:

- Media-layer models: The QoE is computed using the speech and video signal. This model does not require any information about the system under testing (e.g., codec type, packet-loss rate). It can be applied to the evaluation of unknown systems.
- Parametric packet-layer models: This model predicts the QoE from the packet-header information and do not analyze media signals.
- Parametric planning models: This model uses quality planning parameters for networks and terminals to predict the QoE.
- Bit-stream-layer models: In this model, encoded bit-stream information and packet-layer information is used to measure QoE.
- Hybrid models: This model is a combination of previously mentioned technologies.

Among these categories, the media-layer objective quality measurement methods are subdivided into three types (Fig. 1.1) including Full Reference (FR), Reduced Reference (RR) and No Reference (NR). Taking video signals as an example, these subtypes can be explained as follows:

- Full-Reference: the degraded signal is compared pixel by pixel with the original signal. It is a very detailed comparison that requires the original signal. Sending the reference signal requires considerable computing resources. In addition, this method may not be applicable because the reference image is often not available. Some of these metrics have been standardized by the ITU [Recommendation, 2004a] [Recommendation, 2004b] [Recommendation, 2008b].
- No-Reference: a video analysis (stream) on receipt without comparing the original signal. Quality information is extracted from the original signal, as no reference is available. These methods are very small, making them very suitable for analysis in real time. However, KPIs output is low, especially compared to methods “Full Reference”. In addition, NR metrics cannot distinguish the difference between quality problems of the very reference and additional disturbance by the network. This method is an extremely difficult task (especially general-purpose metrics that are applicable to a wide variety of image distortion types).

- **Reduced-Reference:** These methods actually build upon representative parameters (typically statistical values) that allow for the quantification of the change of quality between the original and the distorted version. This method provides a solution that lies between FR and NR models. With this method, the perceptual quality of distorted images is predicted based on partial information of reference images. Some of these metrics have been standardized by the ITU [Recommendation, 2008a] [Recommendation, 2010] [Recommendation, 2011c].

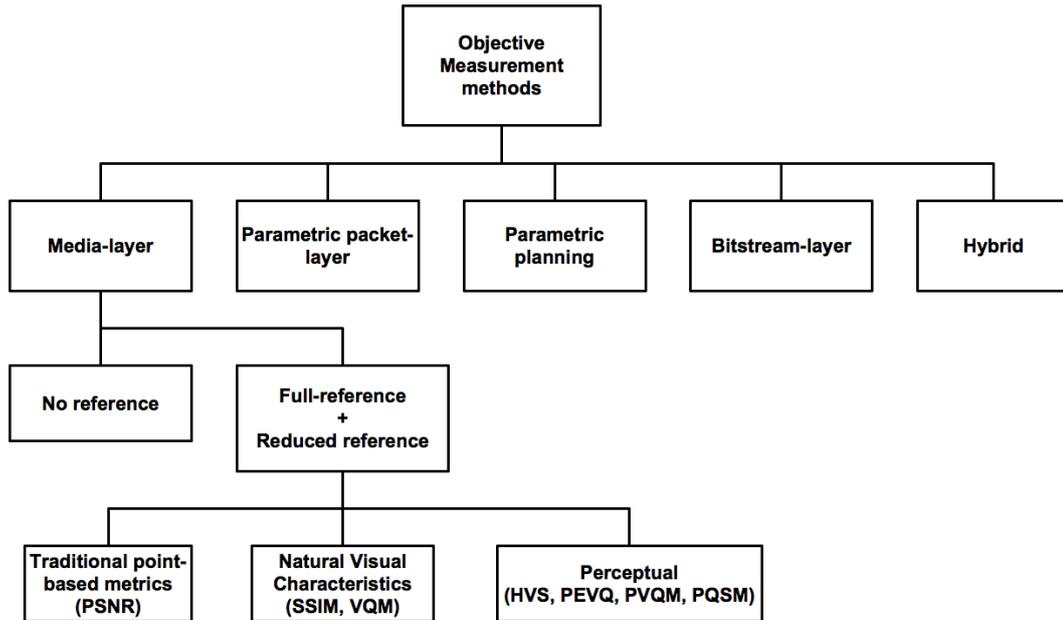


Figure 1.1: Classification of Objective Measurement Methods

Fig. 1.1 shows us the classification of objective quality measurement methods. The full-reference (FR) and reduce-reference (RR) are classified into Traditional point-based metrics (e.g. PSNR, MSE), Natural Visual Characteristics oriented metrics(e.g. SSIM, VQM) and Perceptual oriented metrics (e.g. HVS, PEVQ, PESQ, PQSM). We present in detail now some representative objective metrics:

- PSNR (Peak Signal to Noise Ratio) [Huynh-Thu and Ghanbari, 2008], is the proportion between the maximum signal power and the corruption noise power. The original data is called the signal, and the error is noise in this case. However, this metric has several limitations and it can only be used as a measure of quality of reconstruction of loss compression codecs (e.g., for image compression).
- SSIM (Structural SIMilarity) [Wang et al., 2004] metric is used for measuring the similarity between two images. It is a full reference metric. The image quality measurement is based on distortion-free image as reference.

- PEVQ (Perceptual Evaluation of Video Quality) [Recommendation, 2008b] takes a Full Reference approach to provide MOS scores of the video quality for IPTV, streaming video, mobile TV and video telephony. This method analyzes the degraded video signal through the network to evaluate the degradation. PEVQ has become part of ITU-T Recommendation J.247.
- PESQ (Perceptual Evaluation of Speech Quality) [Rix et al., 2002] is a method to evaluate autonomously speech quality as experience of a telephony system user. It is standardised as ITU-T recommendation P.862 [Recommendation, 2001b].
- PSQA (Pseudo-Subjective Quality Assessment) [Rubino, 2005] is a measurement method that uses a learning tool to know how user perceives when receiving data. The authors use a Random Neural Network model, which is a network having positive and negative customers.
- IETF proposed the Media Delivery Index (MDI) [Welch and Clark, 2006], a metric that is based on parameters such as a measure of deviation from nominal flow rates, traffic jitter, and packet loss measurements for a particular flow, and does not consider the codec or video information in its quality assessment process.

QoE measurement possibility helps operators to maintain network's performance in the overall level of customer satisfaction in terms of reliability, availability, scalability, speed, accuracy and efficiency.

#### 1.2.4 QoE measurement methods

We survey now several QoE measurement models that have been recently proposed in the literature. Y. Goong et al. [Goong et al., 2009] proposed a pentagram model for measuring QoE with five factors: integrality, retainability, availability, usability, and instantaneousness. The authors defined a model of experience taking into account these factors in quality evaluations. The model was applied to measure experience of a VoIP service. QoE KPIs and the most important measures were defined as the following:

- Service integration (delay, jitter, and packet loss ratio) (symbol:  $a$ )
- Service retainability (Service interruption ratio) (symbol:  $b$ )
- Service availability (The success ratio of user access service) (symbol:  $c$ )
- Service usability (Service usability) (symbol:  $d$ )
- Service instantaneousness (The response time(s) of establish and access service) (symbol:  $e$ )

Taking the each factor into account, the authors propose a formula (Eq. 1.1) to compute QoE:

$$QoE = \frac{1}{2} \sin(\lambda)(ab + bc + cd + de + ea) \quad (1.1)$$

Where:  $\lambda$  represents the 72-degree angle between the two sides in the QoE pentagram model. This model facilitated QoE measurement during a service use process in order to improve and represent new methods to evaluating perception of VoIP services.

H. Kim et al. [Kim et al., 2008] propose a method for the objective QoE measurement using traditional QoS parameters. QoS and QoE correlation model was described, and the QoE evaluation method using QoS parameters was studied in the converged network environment. The QoS parameters associated with QoE evaluation and the service classes include delay (D), jitter (J), loss rate (L), error rate (E), bandwidth (B) and signal success rate (S). The QoS-QoE correlation model was expressed as Eq. 1.2:

$$QoS = F(D, J, L, E, B, S) \quad (1.2)$$

$$QoE(QoS) = K \left\{ \frac{(e^{QoS-\alpha}) + e^{-QoS+\alpha}}{(e^{QoS-\alpha}) + e^{-QoS+\alpha} + \beta)} + 1 \right\} \quad (1.3)$$

Where:  $\alpha$  is the QoS quality class of the network level. The QoE class measured by the QoS quality parameter of a network-level is mapped similarly to the existing MOS grade with a 5 point-scale.  $\beta$  is determined according to the class of service as the grade of service constant.  $K$  is the scale constant to show the satisfaction with the use service. Using the quality of service information measured in a network level, authors explain the proposed QoS-QoE correlation model for the objective QoE.

Another approach by S. Moebs [Moebs, 2008] takes into account the trade-offs between QoS parameters and the flow requirements for e-learning, and aim to develop better evaluation method for QoE using QoS-QoE correlation. Authors proposed an adaptive multimedia e-learning system that consists of a user model, a domain model and an adaptation model. The domain model represents the concept of the subject domain and usually, these concept structures are described as concept maps, semantic networks or concept graphs. The user model represents general characteristics of the user such as location, preferences for devices, previous knowledge, knowledge state and learning goals. The adaptation model uses adaptation rules to connect the two previously outlined models. Flow-related and learning-related aspects are the principal elements of the QoE, and are both influenced by QoS. This relationship is represented by the following function:

$$QoE = f(QoL(QoS), QoF(QoS)) \quad (1.4)$$

QoS is based on packet loss, delay and jitter in background application, video, audio and data. QoL (Quality of Learning) represents aspects of the user model (feedback, learning styles, assessment of the previous knowledge) and the domain model. QoF (Quality of Flow) is based on feedback, interaction, the balance of skill and challenges, which contains use of technology and emotions.

A. Hamam et al.[Hamam et al., 2008] propose a taxonomy that is modeled by a mathematical model where the QoE is computed as the weighed linear combination of the QoS and User Experience (UE) for a particular haptic user interface:

$$QoE = \zeta \times QoS + (1 - \zeta) \times UE \quad (1.5)$$

Where:  $\zeta$  controls the relative weight given to the QoS parameters compared to the UE parameters. UE is a linear combination of perception measures, rendering quality, physiology measures and quality of service measures. If the quality factors are restricted between 0 and 1, then the overall quality of experience will also have a value between 0 and 1. To achieve this condition, the sum of constant coefficients must be equal to 1.

The four previous approaches have showed different methods for measuring the QoE. However, all these methods are based just on linear-combination of QoS metrics (objective elements) and on the QoS/QoE correlation, from which authors calculate QoE. It seems that they have ignored subjective perception, an indispensable factor of QoE concept. QoE measurement methods must reflect the user's perception and satisfaction. Concretely, the QoE of audio or video media should be evaluated in subjective terms. Subjective quality measurement is the most basic method of evaluating subjective quality. This deficiency motivates us to use a hybrid tool that mixes the two types of measurement: the subjective and objective method. Our measurement method is presented in the appendix A.

The meaning of QoE in network context is so important that we have to find out the relationship between the QoS and the QoE. Actually, understanding the relationship between a familiar concept (QoS) and the new one (QoE) is the best way to understand this new concept. The next subsection clarifies this.

### 1.2.5 QoS/QoE relationship

The notion of Quality of Service [Recommendation, 2011a] [Recommendation, 2001a] has been discussed for more than a decade. It concerns essentially the technical view on service quality. The QoS architectures (e.g. Integrated Services or Differentiated Services) are used to pave the way for high-quality real-time services like VoIP or video streaming. However, the lack of user experience consideration raises unresolved issues. Consequently, in order to redirect the focus towards the end user and quantify the subjective experience gained from using a service, the notion of QoE has been introduced to describe quality as perceived by the human user.

QoS is defined as “*ability of the network to provide a service at an assured service level*” [Soldani et al., 2006], or “*capability of a network to provide better service to selected network traffic described by the following parameters: delay and jitter, loss probability, reliability, throughput and delivery time*” [Markaki et al., 2007]. These definitions show that QoS is fundamentally a technical concept, which usually has little meaning to users. There are also some QoS's definitions that correlate with the QoE: “*collective effect of service performance which determines the degree of satisfaction of a user of the service*” [Recommendation, 1994], “*a set of qualities related to the collective behavior of one or more objects*” [ISO/IEC, 1998], or “*the network's capability to meet the requirements of users and applications*” [Kilkki, 2002].

Compared to QoS, the notion QoE means the user perception about the quality of a particular service or network. It is expressed by human feelings like “good”, “excellent”, “bad”, etc.

According to the QoS/QoE relationship, for us, QoE covers the QoS concept, whereas there is a common belief that QoE is a part of QoS. Fig. 1.2 illustrates our theoretical point. The quality of a network service including core network and access network is determined by QoS Access and QoS Backbone. The quality perceived by end-users when using end-devices is called QoD (Quality of Design). So, we claim that the end-to-end QoE is a conjunction of QoD and QoS.

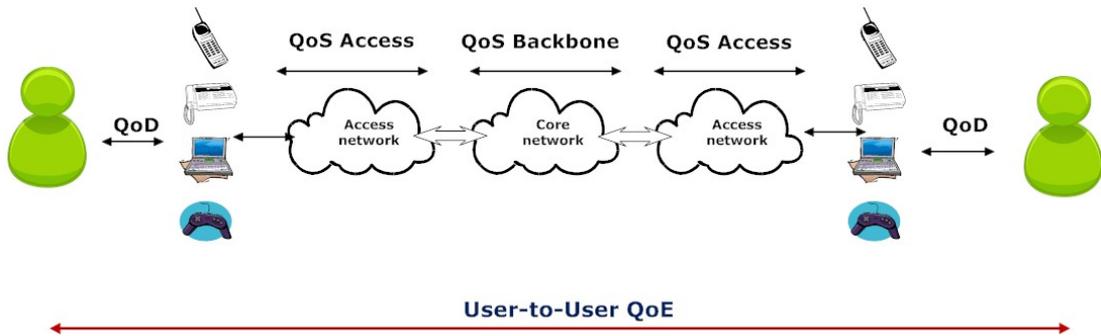


Figure 1.2: Quality chain in an e2e service

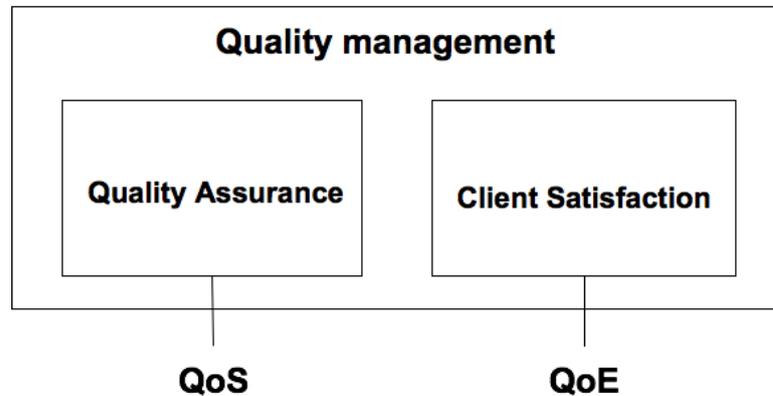


Figure 1.3: QoE and QoS in Quality Management module

Regarding another point of view of relationship between QoS and QoE, we analyze now the *Quality Management* (QM), an indispensable term for every network operators. QM aims not only to ensure that a network service is consistent but also to achieve it. QM includes two main modules: *quality assurance* and *client satisfaction* (Fig. 1.3). The term *Quality Assurance* (QA) is defined as systematic activities implemented in a quality system to fulfill all quality requirements. While QA tries to design and

meet worthy service specifications, Client Satisfaction (CS) metrics seek to capture the degree to which those requirements translate into services/applications that fulfill user's expectations. The QoS is actually related to the QA and the QoE is related to CS. This relation is explained by the fact that QoS is often part of enforceable SLAs (Service Level Agreements) based on specific performance levels while QoE refers to the perceived quality of the user engagement.

According to the qualitative dependent relationship between QoE and QoS, M.Fiedler et al. [Fiedler et al., 2010b] give a schematic relation describing the impact of QoS problems on QoE. In the Fig. 1.4, the x-axis is the QoS disturbance and the y-axis is the QoE value, which is described by MOS score (Mean Opinion Score) [Recommendation, 2006] (i.e., a scale of score of 1 to 5 where 5 is the best and 1 is the worst). As a function of QoS disturbance, the QoE is divided in three areas, separated by *threshold 1* and *threshold 2*. We now analyze each of these areas (Fig. 1.4):

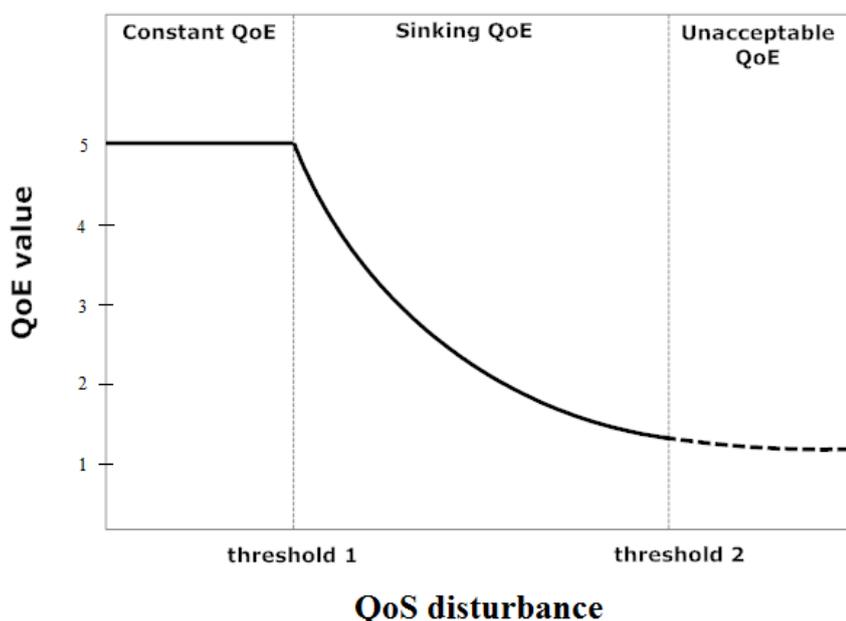


Figure 1.4: General mapping curve between QoS and QoE

- *Area of Constant QoE*: In this area, the QoS is good enough to give a good and constant QoE. A little growth of QoS disturbance does not affect the QoE value. For example, the jitter buffer can eliminate small delays and delay variations, without the user recognizing of the additional delays.
- *Area of decreasing QoE*: When the QoS disturbance exceeds the *threshold 1*, the QoE value begins to decrease. After, the higher the QoS disturbance grows, the higher the QoE value decreases. In case of high QoE, a little additional QoS disturbance might have a considerable impact on the QoE. On the contrary, for

low QoE, the particular additional QoS disturbance might not be that critical anymore.

- *Area of unacceptable QoE*: After reaching the *threshold 2*, the perceived quality becomes unacceptable. For example, the service might stop working because of timeouts problem. The dashed line illustrates that user might quit the service from this point (threshold 2). For another example, threshold 2 represents the cancellation rate of web browsing when the loading time exceeds a limit.

Concerning the mapping between QoS and QoE in the figure 1.4, the network provider should detect the moment of *threshold 1* and propose solutions to improve service quality as long as the QoE falling into the *decreasing area*. The goal is not only to solve the problem of quality degradation but also to avoid the churn problem (i.e. users give up the operator). Hence, the QoE is the most important factor for operator to properly design and manage its network. A good QoS gives a good QoE. However, maintaining all the QoS parameters will not ensure a satisfied user. For example, in a wireless network, if there is no coverage in a short distance away, this causes a bad QoE although there is an excellent throughput in one part of a network.

In order to guaranty a good QoE in network system, apart from understanding the QoE/QoS relationship, we need also to comprehend the impact of networking on QoE. The next subsection discusses this issue.

## 1.2.6 Impact of Networking on QoE

According to impact of networking in QoE, with the continuous development of network services, it is giving new features and advances. It is not easy to explain how networking can affect the QoE because there are many research topics, technologies and application related with the networking topic. First, based on [Martinez-Yelmo et al., 2010], we present the layered approach similar to TCP/IP model to explain this issue. Then, we talk about the impact of some network criteria on the QoE.

### 1.2.6.1 Layered classification of impacts on QoE

**1.2.6.1.1 Link layer** The limit of the maximum speed that is transmitted in a link gives to the link layer a fundamental role in the user experience. In fact, it decides how the data is transported over the links and how the end-users can access to a link. In addition, the link layer can prioritize some kind of traffic in adjusting QoS parameters. On the other hand, this layer supports not only wired link technologies but also wireless ones. We cite here some examples: Wifi [Henry and Luo, 2002], Wimax [Nuaymi, 2007] and the UMTS [Kaarainen et al., 2001] that provides the wireless infrastructure for 3G networks. Besides the shared access technologies we mention previously, there are also point-to-point technologies such as Digital Subscriber Line (DSL) [Starr et al., 1999] and Fiber To The Home (FTTH) [Koonen, 2006], which can provide QoS. The link layer technologies have the same aim but different performance, capacity and behavior. In

addition, the deployed scenario and the external conditions influent their performance. Therefore, the context and scenario description is very important in QoE measurement.

**1.2.6.1.2 Network layer** The network layer includes Internet Protocols (IP). The IP performance depends on some factors. First, it is the routing path that data packets must follow to reach the destination. The routing path is influenced by three factors: the end points, the connectivity from ISPs to the customers and the relationship between ISPs. Building the routing tables in the routers of ISPs needs BGP (Border Gateway Protocol) [Ramalho, 2000] that manages the ASs (Autonomous Systems). Thus, the performance of Internet is based on the way the ISPs establish routing policies through BGP. Therefore, many research projects related with IP carry the relationship between ASs and ISPs. Considering the QoE makes these projects more realistic and closer to the results in a real deployment. The other factor affecting IP is the mobility. IP was originally designed in considering only wired network where hosts are fixed. However, the appearance of wireless technology as well as the laptops, hand-held devices, gives a challenge to IPs. In fact, the QoE perceived by end-users changes due to the fact that the mobility affects delay, jitter and loss rate. So, the QoE is different due to the scenario and on the adopted technical solutions.

**1.2.6.1.3 Transport layer** The transport layer plays an important role in the QoE perceived by end-user. The main protocols of this layer are TCP (Transmission Control Protocol) [Aweya, 2002] and UDP (User Datagram Protocol) [Kurose and Ross, 2005]. The connection-oriented protocol TCP realizes different end-to-end flow congestion control mechanisms, which affect the QoE [Trinh et al., 2010]. Besides TCP, there are also other similar protocols such as SCTP(Stream Control Transmission Protocol) [Stewart and Metz, 2001] or DCCP(Datagram Congestion Control Protocol) [Kohler et al., 2006]. On the other hand, the connectionless protocol UDP is used for real time transmission where the flow control is not necessary. Actually, the peer-to-peer traffic that has been growing for years affect quality of others flows in the Internet because peer-to-peer applications open lots of connections both UDP and TCP. Furthermore, different flows competing for the available network resource implies the overload issue of network system. According to [Martinez-Yelmo et al., 2010], The IETF Low Extra Delay Background Transport (LEDBAT) WorkingGroup (WG) carries this problem to develop lower than best effort transport protocols that avoid to decrease the QoE appreciated in the rest of applications running on the Internet.

**1.2.6.1.4 Application layer** The application layer is an important layer in terms of QoE because it is the only thing that users can really appreciate and interact with. In fact, the end-users do not take into account the technical metrics under their applications. They just consider the subjective media quality of applications they are using. For example, the simplicity of *install and use* is a important feature of a VoIP application. If the latter was able to cross NATs and firewalls without reconfiguring any network equipment and the users could install it in any computer and network without any com-

plex configuration, this application would obtain a good QoE. Skype [Chen et al., 2006] is such an application that has much success due to the good QoE with the simplicity of install and use. QoE means differently in different kinds of applications. While the sound fidelity and ability to smoothly take turns in a conversation give a good QoE for a VoIP application, a multimedia streaming application with large and clear video image has a positive QoE. For a web browser, good QoE means the Web content is retrieved fast enough.

### 1.2.6.2 Impact of user mobility on QoE

In next generation wireless networks, the IP-based transport through the radio and core network parts is provided by the all IP-paradigm solution. However, this concept requires a precise management of user mobility. Therefore, the impacts of user mobility on QoE are important issue to consider, and studied by [Bernardo et al., 2008] and [Ciubotaru and Muntean, 2009].

F. Bernardo et al. [Bernardo et al., 2008] propose mechanisms in order to evaluate the QoE that users perceive with different QoS-aware mobility management strategies. In their work, the authors use a testbed that can evaluate the QoE in a real heterogeneous wireless access network. By contrast, B. Ciubotaru et al. [Ciubotaru and Muntean, 2009] propose another novel quality-aware approach (algorithm SASHA) to handover on load balancing among different networks using a comprehensive function for decision-making. These approaches are described concretely as follows:

In [Bernardo et al., 2008], they presented a brief overview of the testbed in the Mobility Management (MM). This testbed reproduces a Beyond 3G heterogeneous radio access network that consists of three Radio Access Networks (RANs): UTRAN, GERAN, and WLAN interfacing a common Core Network based on DiffServ/MPLS protocols. In this testbed, User Equipment connected to Ingress Routers (IRs) through Traffic Switch. The authors present three handover types:

- Horizontal Handover where handover is realized between base stations of the same RAN, called intra-RAN handover.
- Intra-IR Vertical Handover where the handover is performed between base stations of different RANs.
- Inter-IR Vertical Handover where MM plays a crucial role with an IR change. On the core network side, the data information of the User Under Test (UUT) is encapsulated into MPLS tunnels from the Egress Router to one of the IRs for downlink and for uplink.

After testing different handover types, the QoE of the UUT is depicted by MOS values. The authors used a full reference model-based objective metric to obtain a satisfaction level measurement.

Studies of [Ciubotaru and Muntean, 2009] focused in a heterogeneous wireless network environment where the mobile device (MN) uses different wireless technologies to

access to several networks. In order to maximize user's perceived quality in a dynamic environment a quality oriented mobility management solution with efficient resource allocation is required. Multimedia Mobility Management System (M3S) is an application level framework for delivering high quality multimedia content to mobile clients. A novel Quality of Multimedia Streaming metric (QMS) is presented to depict and quantify the impact of separated network parameters to the multimedia delivery quality. M3S provides handover management and efficient quality oriented resource allocation using the Smooth Adaptive Soft-Handover Algorithm (SASHA), which performs handover between different networks by transferring the load from one network to the other. QMS is described by the Equation (1.6) with the communication channel  $i$ :

$$\begin{aligned} QMS^i = & w_1 QoS_{grade}^i + w_2 \times QoE_{grade}^i \\ & + w_3 \times Cost_{grade}^i + w_4 \times Peff_{grade}^i \\ & + w_5 \times UPref_{grade}^i \end{aligned} \quad (1.6)$$

User preference grade ( $UPref_{grade}^i$ ) is the score of the user that expresses the user's preference for the network interface used by the communication channel  $i$ .

User's perceived quality grade ( $QoE_{grade}^i$ ) is the user's perceived quality that is calculated based on the received content's video quality. QoE grade for each communication channel is determined by distributing the overall QoE according to the channel's corresponding rate share:

$$QoE_{grade}^i = QoE_{overall} \times RateShare^i \quad (1.7)$$

This algorithm executes a quality-oriented handover by sending the load from one network to another. The handover decision-making algorithm is based on the QMS grades for each communication channel. QMS represents a more comprehensive metric to construct a handover management solution oriented on user's perceived quality by taking several QoS and QoE related parameters into account. This approach aims to maximize the end-users perceived quality (QoE) when streaming multimedia content by efficiently using all the communication resources available. M3S uses the novel algorithm (SASHA) to distribute the load over the communication channels based on their estimated contribution in order to deliver high quality multimedia content.

The two approaches above show influence of user mobility on the QoE. The next subsection discusses the impact of network resource utilization and management on the QoE.

### 1.2.6.3 Impact of network resource utilization and management on QoE

Proper management of the network resource is a good way to handle and maintain the system quality, so it also has a significant impact on end user perception. Indeed, there were many systems taking into account this feature such as [Yamada et al., 2007] and [Agboma and Liotta, 2008].

In [Yamada et al., 2007], a network resource management system for Resource Access Control Facility (RACF) based on the end-user QoE is presented. The authors

constructed a concept of Network Resource Management system (NetRM) to provide service-oriented network services via Web services interface (WSI). NetRM distributes and supervises the suitable end-to-end network resources to meet the service request. NetRM furnishes guaranteed and scheduled network services to network service clients.

In [Agboma and Liotta, 2008], the authors propose a more concrete approach based on a QoE management methodology, which aims at maximizing the user's QoE while preserving network resources at the same time. A statistical modeling technique, Discriminant Analysis, has been employed to implement this methodology. Due to the prediction model, network operators can anticipate the user's experience and then allocate network resources accordingly. A statistical technique can be used to predict a qualitative attribute of an individual from known quantitative variables. In order to perform discriminated analysis, one needs a set of observations whose group memberships are already known. The general formula for a discriminated function is given in below:

$$f_{km} = u_0 + u_1X_{1km} + u_2X_{2km} + \dots + u_pX_{pkm} \quad (1.8)$$

where:

$f_{km}$  is the predicted discriminated score for case  $m$  in the group  $k_i$ ,

$X_{ikm}$  is the value of the quantitative predictors  $X_i$  for case  $m$  in group  $k$ ,

$u_i$  is coefficients for variable  $i$ , for  $i = 1..p$ .

The results taken from this paper illustrate how to maximize the utilization of network resources and provide different levels of service quality.

We consider these two approaches [Yamada et al., 2007] [Agboma and Liotta, 2008] as adaptive systems. They both used resource management functions to maintain required QoE and QoS. The main differences between the two approaches are the function used. In [Yamada et al., 2007], in order to maintain the QoE, network resource management function was implemented in each access network and utilized when the communication service was set up to allocate required bandwidth. Instead, [Agboma and Liotta, 2008] used discriminated function and implemented a statistical modeling technique to maximize the user's QoE while preserving network resources.

#### 1.2.6.4 Impact of billing and pricing

We will discuss how billing and pricing can have an impact on QoE. [Takahashi et al., 2008] studied using a particular application based on IPTV service and focus on QoE requirements in consideration of Service Billing for this application. In their work, QoE was defined as how well an IPTV service satisfies the customer's expectation. It was found that the service billing had a strong correlation with the customer's expectation for a service quality. Authors gave an example where a service has a long channel zapping delay. An IPTV customer subscribing a basic class accepted this channel zapping time. On the other hand, the premium IPTV customer who pays more expensive service charge were dissatisfied with the same channel zapping time. Since customers tend to expect a better service quality as the service charge increases, the customer's evaluation for the service strongly depends on its service billing.

As an example of QoE metric with service billing, authors propose *End-user's Utility* ( $U_s$ ), which is a relative satisfaction level for a service quality. Another metric is *Service Quality Metric* ( $T_s$ ) for IPTV, e.g., Channel zapping delay, Control command response time, STB startup time, Picture quality, etc. The third metric is *Service Billing* ( $B_s$ ) that is the service charge for a service S. Finally  $U_s$  can be defined as:  $U_s = f(T_s, B_s)$ . In this formula,  $f$  is a weight function that transforms the absolute service quality metric,  $T_s$ , with service billing as a relative end-user utility value  $U_s$ . Therefore, even if two customers have the same service quality metrics, their satisfaction levels ( $U_s$ ) can be different because each customer may pay a different service charge. The weight function is based on the types of services, the types of customers and the policies of service billing.

So, after this section we got a clear idea of the QoE notion and his role in the networking context. However, QoE will remain an abstract concept if it is not applied to any real network architecture. In this thesis, we applied the proposed e2e QoE model to a CDN architecture that will be presented in the next section. We have chosen the CDN to implement the QoE model due to the fact that CDN is an user-centered infrastructure. The latter always puts the user perception on top priority and considers it as motivation for network management and Content-Oriented Network (CON).

### 1.3 Content Distribution Network

In this section, first, we present the general architecture of a CDN system with their components and functions. After, we describe in detail its two main layers: the routing and the meta-routing layer.

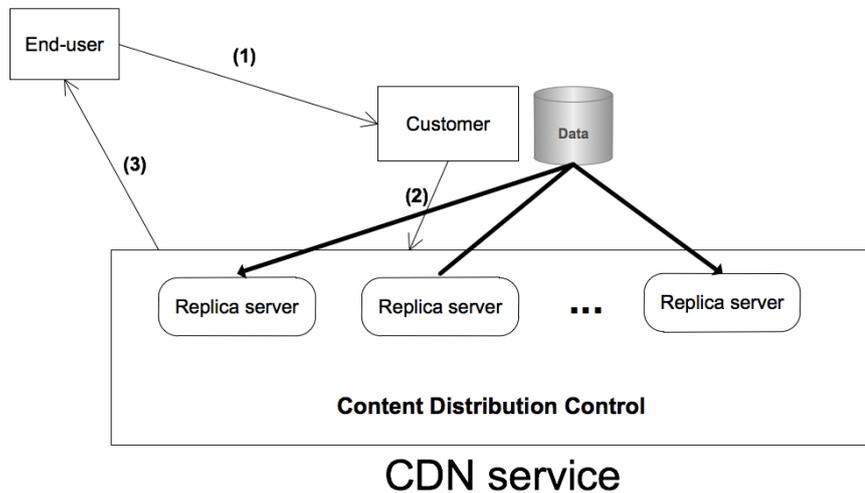


Figure 1.5: CDN service

Nowadays, the demand for content in the Internet increases rapidly. That needs an efficient and scalable content distribution solution. Implementing multiple servers

placed at geographically distributed locations replaces using single servers schemes. The general idea of this evolution is to bring geographically the servers closer to clients. This new architecture, called Content Distribution Network (CDN), will improve the performance of the system, reduce network load and provide better fault tolerance. Currently, there are many successful CDN providers like Akamai [Akamai, 1998], Digital Island [Digital Island, 1996], etc.

Generally, operation mechanism of CDN service is depicted in the Fig. 1.5. When a customer (e.g., a web site, VoD service, etc.) subscribes to a CDN service, the customer provides data to the CDN provider. The latter will implement a distribution network in using replica server scheme. More precisely, CDN provider replicates the content data from origin server of the customer to the replica servers scattered over the Internet. The operation mechanism of a user's request is described step by step as following. First, the end-user send a request to the web site (step 1). Then, the web site forwards the request to the CDN service (step 2). After, the CDN provider uses several mechanisms to select the most appropriate server to serve the end-user (step 3).

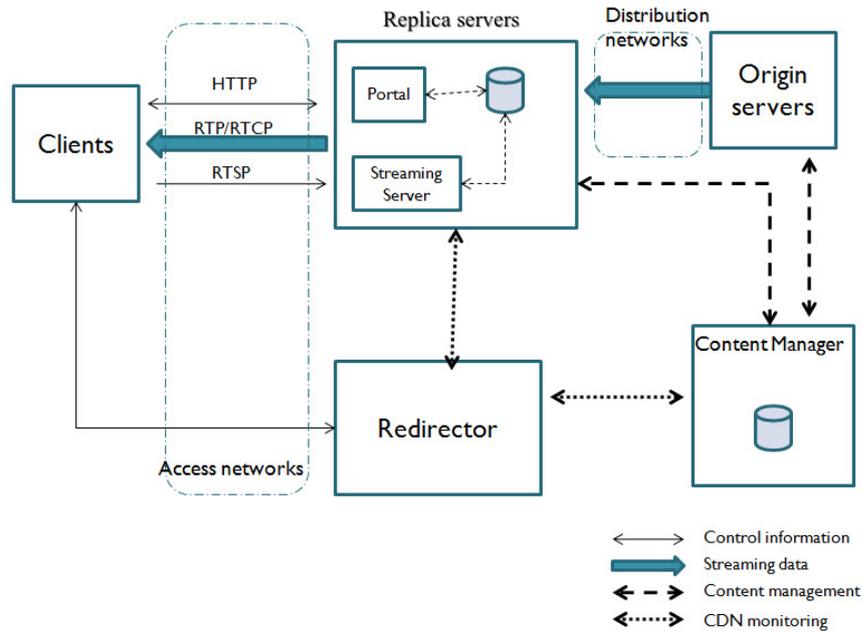


Figure 1.6: General Architecture of CDN system

Fig. 1.6 shows a detailed architecture of CDN system [Peng, 2004] [Pathan and Buyya, 2008]. The CDN focused on serving both web content and streaming media. We present now some main components of this architecture: Origin Server, Replica servers, Clients, Access Network, Distribution Network, Content Manager and Redirector.

*Origin server* contains the information to be distributed or accessed by clients. It distributes the delegated content to the replica servers of the CDN through the *Dis-*

*tribution Network*. The replica servers act as proxy/cache servers with the ability to store and deliver content. The structure, number and location of replica servers are used to classify the CDN and the selection algorithm executed to determine the server who serves each issued request. *Clients* are represented by PCs or special set-top boxes which request and download a particular piece of content stored somewhere in the CDN. The CDN usually works with clusters of clients rather than individual clients. The clients access the service provided by the CDN through different *Access Networks*, depending on the ISP.

Distribution network interconnects the Origin Servers with the Multimedia Streaming Servers (MSSs) to deliver media objects within the streaming media CDN. The Content Manager controls the media objects stored in each replica server, providing this information to the *Redirector* in order to get each client served by the most suitable replica server. The Redirector module provides intelligence to the system, because it estimates the most adequate replica server for each different request and client.

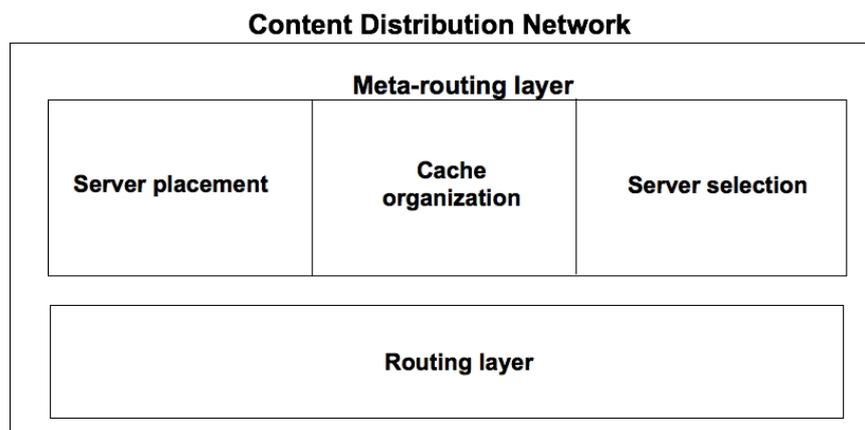


Figure 1.7: Two layers in a CDN architecture

A functional CDN architecture includes two key layers: the *Routing layer* and *Meta-Routing layer* (Fig. 1.7).

- The Routing layer: The core network components (e.g. routers, switches, etc.) take into account the routing process in providing the basic network connectivity and guaranteed QoS for specific request or traffic. This layer plays no active role in server selection process.
- The Meta-routing layer: CDN providers replicate content to replica servers. This layer takes into account some modules such as server placement, replica server selection, etc.

These two above independent layers are the brain of a CDN architecture (Fig. 1.7). While the routing layer takes into account the path finding process to optimize data

delivery task, the meta-routing layer carries out some modules. In this thesis, we focus on server selection module and propose a novel server selection policy based on the user perception.

Concretely, for each layer we proposed a novel approach:

1. **In the routing layer:** We proposed a routing algorithm, called QQAR. The latter is based on QoE in using Q-Learning algorithm. We considered the user's feedback as reward to formalize our routing model to a Reinforcement Learning model. In fact, the goal is to optimize the end-user's QoE of the routing system.
2. **In the meta-routing layer:** Among several modules in this layer, we focused on the server selection module. We proposed a new server selection algorithm in formalizing it to the MAB (Multi-armed bandit) formalization. In fact, this algorithm not only chooses the most appropriate server that gives the good QoE level, but also takes into consideration the exploitation-exploration trade-off.

### 1.3.1 Routing layer

As a layer below the meta-routing layer in a CDN, the routing layer takes into account the routing process to transfer data between end-user and selected server. The goal is to choose the best path that gives good user's perception at end-user side. For traditional routing process in CDN, providers use usually QoS routing systems, which based on technical parameters and widely used routing protocols. In this section, first, we survey the QoS routing approaches they used in their CDN systems. Then, we present some classical routing algorithms. Finally, we discuss QoS routing with different protocols.

#### 1.3.1.1 Routing in telecommunication network

Routing is one of the main features of the network layer which is responsible for deciding on which output line, an incoming packet must be re-transmitted. In general, there are direct delivery, which corresponds to the transfer of a datagram between two computers on the same network, and indirect delivery, which is implemented as at least a router separates the original sender and final recipient. Indirect delivery is necessary to determine which router to send an IP datagram according to its final destination. This is possible to use a specific routing table for each router to determine which route to send a datagram output for any network. The main contents of a routing table consists of quadruples (*destination, gateway, mask, interface*)

There are many routing protocols, which can be divided into two following categories:

- The intra-domain protocols "*Interior Gateway Protocol (IGP)*" (such as RIP, OSPF, IGRP, IS-IS) calculating the routing tables within the same AS.
- The inter-domain protocols "*Exterior Gateway Protocol (EGP)*" (such as EGP or BGP) that calculate the routing tables between nodes belonging to different AS.

In this subsection, we focus particularly on the intra-domain protocols that can be divided into two main categories: *Distance-vector protocols* and *Link-state protocols*.

### 1.3.1.2 Classical routing algorithms

**1.3.1.2.1 Distance-vector routing** For distance-vector routing algorithm (Bellman Ford routing) [Heineman et al., 2008], each router has a routing table indicating for each destination, the best known distance and which line to reach it. This algorithm is used by the protocol RIPv1, RIPv2 [Atkinson and Fanto, 2007] and IGRP [Sharma et al., 2010]. The estimated distance in the routing table assumes that a metric must be calculated locally for each router (e.g. the hop-count, packet number in the queue, routing time, link cost, etc.). The distance-vector routing algorithm can determine the optimal route by exchanging information in the list of metric distances. Furthermore, this information is exchanged between neighbour routers. First, each router starts registering in its routing table links with neighbour routers indicating the line and the estimated distance. Then, periodically, each router sends it to all its neighbour table extracted from its routing table containing for each destination the estimated distance, thereby to update the routing table. The principle idea of distance-vector routing is to find the shortest path following the imposed metric.

**1.3.1.2.2 Link-state routing** With the link-state routing algorithms, each router maintains the same database of network topology. Each router executes the same algorithm in parallel, by building a tree, whose it is the root, to find the shortest path. This tree of the shortest path has the information of the entire path to any destination network, but only the next hop is stored in routing tables. After detecting the change of a link state, the router transmits information via a mechanism of broadcast to all other routers. Like distance-vector routing, the notion of shortest path uses a distance defined by a metric such as hop count, the number of packets in the queue, etc. First, link-state routing algorithm begins researching neighbors and assessing the distance to reach them. Then, each estimated distance is sent to all routers in the network. Each router can include in its database the complete network topology, and then build the tree to calculate the shortest path. A typical link-state routing protocol is the OSPF (Open Shortest Path First) [Moy, 2008] [Rétvári et al., 2009].

### 1.3.1.3 QoS-based routing

The user's needs of telecommunications are characterized by an increase in bandwidth requirements (television services like VoD, Internet access, telecommuting, video telephony). The use of next generation network (NGN) as part of multimedia applications, services of guaranteed quality, mailing services, mobile services, etc.. requires that the delivery service is guaranteed by a Quality of Service control (throughput, delay, jitter, reliability, etc.). Certainly, the mentioned above routing techniques do not meet these characteristics. Thus, it is necessary to design new routing mechanisms. Routing must take into account the real state of links and routers to perform QoS routing. Taking into account the quality of service routing is organized around three stages: 1) Evaluating the state of network links; 2) Collecting the information by the routers and 3) Determining routers satisfying these QoS constraints.

However, the authors of [Wang and Crowcroft, 1995] and [Michael and Johnson, 1979] proved that QoS-based routing system met NP-complete problem in finding a path satisfying at the same time more than two non-correlative constraints (e.g. delay, loss rate, jitter and cost). Apart from the complexity problem, QoS-based routing faces another issues as follows. In fact, identifying a path for guarantying the QoS requirements needs the accuracy of the knowledge of the availability of network resources. Most existing routing algorithms assume that availability of precise state information. However, in many cases this information may be inaccurate and insufficient. Another issue of QoS-based routing is that this kind of routing is not sufficient anymore for the new types of content on the Internet: the high quality multimedia content. Such type of content is sensitive to e2e delay, loss-rate, etc., opposed to agreements of classic SLA (Service Level Agreement) between network operators and service providers.

The approaches of QoS routing can be classified into four categories [Hoceini, 2004]:

- Those based on the principle of switching (MPLS).
- Those derived from existing protocols (QOSPF, multipath).
- Those based on multiple metrics (SW, WS).
- Those based on learning techniques (Q-Learning, Ants, ...)

We present in detail now the four categories:

#### **1.3.1.3.1 Switching-based protocol: MPLS (Multiprotocol label switching)**

MPLS [Bocci et al., 2010] is a protocol that imposes a fixed route to different flows to arrive at their destinations. It is based on the concept of label switching. Traffic characterization defining the required QoS is associated with each flow. It is not strictly a protocol of layer 3 (network layer), but rather an intermediate between the level 2 (link layer) and layer 3 (network layer) of the OSI model. MPLS is a technology that offers circuit mode to an IP, to the image of X25 or ATM. The advantages of MPLS can be summarized as follows:

- Fast switching at the level of core equipment.
- Ability to associate specific routing policies with certain flows.

The idea of label switching is to replace the search for the longest match between the destination address of IP packets, and present prefixes in the routing tables by inserting a label of fixed-length header between the network and link header of the packets. The determination of the next is then made through the label. This solution is more advantageous, since it does not rely anymore on finding the longest match of prefixes of variable length, but on finding a fixed-length label.

Prospects for MPLS to improve the quality of services are numerous. Cisco contributed to the standardization of MPLS by offering the Tag Switching, which uses 3 bits in the *Service Type* field of the IPv4 header of incoming IP packets and place them

in the field *CoS* (*Class of Service*) of the MPLS label. In the MPLS domain, the routing can be performed so as to favor the flow having the highest CoS. Moreover, one of the most interesting aspects of MPLS is the ability to build explicit end-to-end routing, the LSP (Label-Switched Paths). These paths can be determined to meet the QoS constraints. Thus, multiple parallel paths can exist between two routers in the MPLS domain. Each path offers unique characteristics in terms of delay or bandwidth.

**1.3.1.3.2 Protocols derived from existing protocols** We survey now some protocols that are derived from existing protocols. The first one is QOSPF (Quality Of Service Path First) [Zhang et al., 2007], an extension of OSPF. Combined with a reservation protocol, this routing protocol can inform to all routers the capacity of links to support constraints of QoS. QOSPF is based on the use of two messages, the RES-LSA (Link Resource Advertisement message) and the RRA (Resource Reservation Advertisement message). The purpose of the RES-LSA messages is to inform all the routers of the link state. This information is used to determine paths. This message type is used when a new router is added to the network or when the load of a link varies. The role of message RRA is to indicate the resources used by a stream (or rather reserved for this flow). Each router is notified of the resources and the route taken by each flow. The number of messages RRA can become very high if the number of flows increases, which poses a problem of scale.

Classical routing algorithms use a single route to route packets from their source to their destination. To make best use of network resources, one conducts a multi-path routing [He and Rexford, 2008]. Indeed, the routing algorithms can find multiple paths with the same minimal cost. The choice of one of these paths is currently arbitrary. The techniques, namely Equal Cost Multipath (ECMP) [Gojmerac et al., 2003], have been proposed in order to evenly distribute the traffic on these paths. Others suggest an uneven distribution on different paths (Optimized Multipath, OMP). There are also extensions to OSPF to perform routing on several paths. OSPF-ECMP can divide the traffic equally on optimal paths. Since these routing techniques are not sensitive to the load, there is no risk of oscillations.

**1.3.1.3.3 Protocols based on multiple metrics** Metrics combination is used to combine different metrics in a single metric and compute the path that minimizes the resulting metric. Metrics composition can use linear, non-linear, and Lagrange relaxation composition. The key idea in QoS routing is to find a path satisfying multiple constraints on routing metrics like desired bandwidth, delay time and e2e QoS. Nowadays, many studies have been introduced to integrate the QoS requirements problematic for the routing algorithm. However, only few proposed routing algorithms are suitable for the today's Internet such as QOSPF (Quality Of Service Path First), TE (Traffic Engineering) and WCA (Wang Crowcroft Algorithm). Enhanced Interior Gateway Routing Protocols (EIGRP) [Lemma et al., 2010] is also one of the multi-metrics routing algorithms.

In [Guerin and Orda, 1999], the authors proposed Guerin-Orda algorithm to solve

both bandwidth-constrained and delay-constrained problem with imprecise network states. The goal of the bandwidth-constrained routing is to find the path with highest probability to accommodate a connection having a precise bandwidth requirement. The goal of the delay-constrained routing is to find a path having highest probability to satisfy a given e2e delay. In fact, it is NP-hard to find the path satisfying at the same time these constraints. The idea of this algorithm is to transform a global constraint into local constraints. More precisely, the approach is to split the e2e delay constraint of intermediate links in such a way that all links in the path have the same probability to satisfy its local constraint.

Ma and Steenkiste [Ma and Steenkiste, ] presented a novel approach to solve the constraints on delay, jitter and buffer space in rate-based scheduling networks. In [Wang and Crowcroft, 1996], the authors proposed a hop-by-hop distributed routing system. Every routers pre-determine a forwarding neighbor for every possible destination. This approach uses link-state protocol to maintain a complete global state, which is used to determine the shortest-widest path (i.e. the path that has maximum bandwidth and smallest delay).

**1.3.1.3.4 Protocols based on learning techniques** In order to design adaptive algorithms for dynamic networks routing problems, the researchers based on different learning techniques. For Reinforcement Learning (RL) technique is defined as learning a policy, a mapping of observations into actions, based on feedback from the environment. In RL, the network represents the environment whose state is determined by the number and relative position of nodes, the status of links between them and the dynamics of packets. Each node is considered as an agent who has a choice of actions. Another learning technique is based on the behavior of various biological entities such as Ant/Bee Colony [Mellouk et al., 2011] [Bitam et al., 2010a]. Ant colony optimization (ACO) [Dorigo et al., 2006] is influenced by the foraging behavior of some ant species. These ants leave pheromone on the ground to give information about some favorable path that should be followed by other members of the colony. Ant colony optimization is widely used in routing systems. G. Di Caro et al. [Di Caro et al., 2005] proposed AntHocNet, an algorithm for routing in mobile ad hoc network. This hybrid algorithm combines reactive route setup with proactive route probing, maintenance and improvement. At the start of a data session, it sets up paths when they are needed instead of maintaining routes to all destinations. Backward ants return to the source to set up paths in the form of pheromone tables indicating their quality as well.

Besides these techniques, there are also various routing algorithms based on a neural network for finding the optimal path [Patel and Yadav, 2012] [Schuler et al., 2009]. In [Schuler et al., 2009], the authors propose a hybrid intelligent method for routing combining Hopfield Neural Networks (HNN) and simulated annealing (SA). The proposed method is considered as a modified version of the discrete time equation used to determine the new neuron input. The novel version of the equation improved the HNN convergence and decreased the computation cost. In addition, the SA algorithm is used to obtain the optimal parameters of the HNN. Gelenbe et al. [Gelenbe and Lent, 2004]

propose Cognitive Packet Networks (CPN) that are based on random neural networks. These are store-and-forward packet networks in which the intelligence is constructed into the packets rather than at the routers or in the high-level protocols. CPN is a reliable packet network infrastructure which incorporates packet loss and delays directly into user QoS criteria and use these criteria to do routing. The major drawback of algorithms based on cognitive packet networks is the convergence time, which is affected when the network is heavily loaded.

Q-Routing approach was firstly proposed by Boyan et al. [Boyan and Littman, 1994]. That is a technique where each node makes its routing decision based on the local routing information, represented as a table of Q values which estimate the quality of the alternative routes. These values are updated each time the node sends a packet to one of its neighbors. However, when a Q value is not updated for a long time, it does not necessarily reflect the current state of the network and hence a routing decision that is based on such an unreliable Q value will not be accurate. The update rule in Q-Routing does not take into account the reliability of the estimated or the updated Q value because it depends on the traffic pattern and load levels. In fact, most of the Q values in the network are unreliable.

A. Mellouk et al. [Mellouk et al., 2012] propose a continuous state-dependent routing algorithm based on a multi-path routing approach exploiting the Q-Routing algorithm. The global learning algorithm finds K best paths in terms of cumulative link cost and optimizes the residual bandwidth and the average delivery time on these paths. The technique used to estimate the residual bandwidth and the e2e delay are based on the reinforcement learning approach to take into account dynamic changes in networks. The proposed system, called Bandwidth Delay K Optimal paths Q Routing Algorithms (BD-KOQRA), takes into account static and dynamic QoS criteria and is based on inductive approaches with continuous learning of dynamic network parameters such as link capacity, end-to-end delay, and residual bandwidth. A load balancing policy that depends on the traffic dynamic path probability distribution function is also supported.

On the last few decades, many research areas are massively influenced by the behavior of various biological entities and phenomena. Hence, there are also routing algorithms inspired by bee life [Bitam et al., 2010b]. Teodorovic and Dell'Orco proposed Bee Colony Optimization (BCO) [Teodorović and Dell'Orco, 2005] which is an improvement and generalization of the Bee System. This proposal is able to solve combinatorial problems characterized by uncertainty. The authors used Fuzzy Bee system representing one of the possible BCO to the ride-sharing problem. Inspired by BCO, Markovic et al. [Marković et al., 2007] tailored BCO-RWA for the Routing and Wavelength Assignment (RWA) problem in all-optical networks regardless wavelength conversion in intermediate nodes. This proposal has been applied for static case in which lightpath requests are known in advance. Authors proved that BCO-RWA can produce optimal or near-optimal solutions in a reasonable amount of computer time.

So, we can see that all approaches above have taken into account just the QoS criteria (technical parameters) in ignoring the QoE notion. This lack motivates us to integrate the QoE into a routing system. This work is described in chapter 2. We work not

only on routing layer but also on the meta-routing layer, which is presented in the next subsection.

### 1.3.2 Meta-routing layer

The basic idea of a CDN is to move the content from the places of origin servers to the places closer to end-users, at the edge of network. Apart from routing mechanism in core network (that is realized by Routing layer modules), we explain in this subsection how the CDN provider places the replica servers, how they build caching policy and how they select the appropriate replica server to serve the user request. Among various modules of meta-routing layer, we survey in this subsection three main modules (Tab. 1.2): *Server placement*, *Cache organization* and *Server selection*. Then, we explain why we have chosen the server selection module among the three ones. Finally, we focus on the “Server selection” module in proposing a new selection algorithm, which will be presented in chapter 3.

#### 1.3.2.1 Server placement

In order to minimize the average content access latency perceived by end-users and the network bandwidth consumption, CDN needs a server placement module to decide where on the network to place the servers, which store replicates of data. Intuitively, replica servers should be placed closer to the clients to reduce latency and bandwidth consumption. In this subsection, we give a survey of the approaches concerning server placement issues.

To resolve the server placement problem, there are some theoretical approaches such as *minimum K-center problem* [Jamin et al., 2000], the *facility location problem* [Qiu et al., 2001] or *k-hierarchically well-separated trees* [Bartal, 1996]. For the *minimum K-center problem*, the goal is to minimize the maximum distance between a node and the nearest center with a given number of centers. Similar to the latter, the *facility location problem* has a given constrain that is the total cost in terms of building facilities and services clients. The third issue, *k-hierarchically well-separated trees*, is based on graph theory.

These theoretical approaches described above are not easy to implement to real CDN systems because they are computationally expensive and they do not consider the characteristics of the network. Therefore, some heuristic solutions have been developed. Qiu et al. [Qiu et al., 2001] propose a server placement approach based on greedy algorithm [Krishnan et al., 2000]. The goal of this algorithm is to choose  $M$  servers among  $N$  potential sites. One site is chosen in a time. First, to determine the suitability for hosting a server, each of  $N$  sites is evaluated individually. The site having the lowest cost (e.g., bandwidth consumption) is chosen. Then, the iteration continues until  $M$  servers have been chosen. This algorithm assumes that users direct their accesses to the server having the lowest cost. The *l-backtracking greedy algorithm* is discussed in [Jamin et al., 2001]. The difference with the basic one is that in each iteration, the *l-backtracking greedy algorithm* verifies all the possible combinations by deleting  $l$  of the already placed servers

Table 1.2: Approaches for Meta-routing modules

Modules	Approaches
Server placement	minimum K-center problem [Jamin et al., 2000] facility location problem [Qiu et al., 2001] k-hierarchically well-separated trees [Bartal, 1996] topology-informed placement strategy [Jamin et al., 2001]
Cache organization	<b>Caching techniques:</b> [Pathan and Buyya, 2008] query-based scheme digest-based scheme directory-based scheme hashing-based scheme <b>Cache update techniques:</b> [Pathan and Buyya, 2008] Periodic update Update propagation On-demand update Invalidation
Server selection	<b>Algorithms:</b> Round-Robin [Szymaniak et al., 2003] Globule project [Pierre and Van Steen, 2006] [Karger et al., 1999] [Andrews et al., 2002] [Ardaiz et al., 2001] <b>Mechanisms:</b> Client multiplexing [Bhattacharjee et al., 1997, Beck and Moore, 1998] HTTP redirection DNS indirection Anycasting Peer-to-peer

and replacing them with  $l + 1$  new servers.

Jamin et al. [Jamin et al., 2001] propose a *topology-informed placement strategy*. The authors assume that the nodes with the highest out-degrees<sup>2</sup> can reach more nodes with smaller latency. They place servers on candidate hosts in descending order of outdegrees. However, as one has not detailed information of network topology, each node represents a single Autonomous System (AS). Each AS-level BGP peering corresponds to a node link.

The experiments of these above strategies show that the *topology-informed placement strategy* can perform almost as well as *the greedy placement*. They show that using router-level topology information gives better performance than exploiting only AS-level topology. In addition, they proved that the performance improvement degrades when

<sup>2</sup>The node outdegree is the number of other nodes it is connected to.

increasing server number.

After placing servers in the network, another important task of CDN service provider is to organize the server cache. This work is clarified in the next subsection.

### 1.3.2.2 Cache organization

We present now the cache organization of a CDN system. *Cache organization* includes caching techniques and cache update techniques. The goal is to guarantee the availability, freshness and reliability of content. The performance of a CDN system will be improved if the cache is well organized. In this subsection, we present cache organization in two parts: caching techniques and cache update techniques [Pathan and Buyya, 2008].

The content caching can be classified into two classes: *intra-cluster* and *inter-cluster* caching. The intra-cluster involves different schemes such as *query-based scheme*, *digest-based scheme*, *directory-based scheme*, *hashing-based scheme*. In the query-based scheme [Wessels and Claffy, 1997], CDN server broadcasts a query to other cooperating CDN servers for each cache miss. The CDN server has to wait for the last “miss” reply of their neighbours to conclude that there is no peer having the requested content. Hence, the problems of this scheme are the significant query traffic and the delay. The digest-based scheme [Rousskov and Wessels, 1998] focuses on the problem of the flooding queries in query-based scheme. In digest-based scheme, each CDN server maintains a digest of content stored by other cooperating replica servers. The updating CDN server informs the content update to cooperating replica servers. Based on the content digest, a CDN server can decide which particular replica server to route a content request to. However, due to the frequent exchange of update message to ensure the correct information about each other, the update traffic overhead is a drawback of this scheme. The directory-based scheme [Gadde et al., 1997] is introduced as a centralized version of the digest-based scheme. In this scheme, the centralized server has the information of all of replica servers in a cluster. The CDN server only informs the directory server when local updates occur. If there is a local cache miss, CDN server will query the directory server. The directory server receives all the information from cooperating replica servers. The problems of this centralized scheme are potential bottleneck and single point of failure. In hashing-based scheme [Karger et al., 1999, Valloppillil and Ross, 1998], all servers use the same hashing functions. The designated server keeps the content based on IP addresses of the CDN servers and the hashing function. This designated server carries out all requests for that particular content. Due to the smallest implementation overhead and highest content sharing efficiency, the hashing-based scheme is more efficient than other schemes.

The inter-cluster cooperative caching can not apply the hashing-based scheme because representative servers of different clusters are normally distributed geographically. There is only the query-based scheme can be applied for inter-cluster caching. A cluster will query his neighbour cluster if it fails to serve a content request. The neighbour will reply a hit message if it finds the requested content. On the contrary, if the neighbour does not find the requested content, it will forward this request to other neighbour. The hashing-based scheme is used by all the servers of a cluster. A server only queries the designated server of that cluster to serve a content request.

To ensure the freshness of content, CDN provider should take into account *the cache update mechanism*. Clearly, with the delays involved in propagating the content, the content may be inconsistent or expired. Hence, the CDN provider has to interest the problems about how long different content is to be considered fresh, when to check back with the origin server for updated content, etc. We discuss now some cache update techniques: *periodic update*, *update propagation*, *on-demand update* and *invalidation*.

In the *periodic update* technique, caches are updated regularly. The drawback of this approach is significant levels of unnecessary traffic generated from update traffic at each interval. With the *update propagation* technique, the active content is performed and pushed to the CDN cache servers. So, the updated version of a document is sent to all caches even there is a change of document at the origin server. Due to the frequently changing of content, this approach creates excess update traffic. According to *on-demand update*, one propagates the latest copy of a document to the replica server based on the prior request for that content. This technique assumes that the content is not updated unless it is requested. The drawback of this technique is the back-and-forth traffic between the cache and origin server. In the *invalidation* technique, the invalidation message is broadcasted to all surrogate caches when a document is modified at the origin server. While the document is being modified, the surrogate caches are blocked. After that, the cache has to fetch an updated version of the document. The disadvantage of this technique is that it does not make full use of the distribution network for content delivery.

Due to the cache organization, CDNs help the content provider to ensure a freshness content and the consistency of CDN sites. In fact, the content providers themselves construct the policies and heuristics for caching policies. Then, the content providers give the caching policies to CDN provider, which will deploy to its caches. Based on these rules, the caches know how to maintain the freshness of content through ensuring consistency.

The tasks of server placement and cache organization are considered preparation-related process of a CDN service. The first one, server placement module, is used to place the replica servers in order to minimize the average content access latency and the network bandwidth consumption. The second one, cache organization module, takes into account the cache management process to guarantee the availability, freshness and reliability of content. Despite of this, these two modules are launched in the preparation phase where one designs the infrastructure for a CDN service. In this thesis, we focus on the module that appears in the main phase of a CDN: the server selection module.

### 1.3.2.3 Server selection

As explained above, we focus on the server selection module among the modules in meta-routing layer. Server selection module is used to direct the user requests to the suitable replica server. It could direct the request to the closest replica server. However, the closest server may not be the best one for servicing this request. So, apart from hop-count criteria, they should consider another metrics such as network proximity, client perceived latency, distance and replica server load. In this subsection, we survey several

server selection algorithms and mechanisms.

**1.3.2.3.1 Server selection algorithms** Server selection algorithms are classified into two types [Pathan and Buyya, 2008]: *adaptive* and *non-adaptive*. The difference between these two types is considering the current condition. In fact, while the adaptive algorithm consider the current condition of environment to select the replica server, the non-adaptive algorithm just uses some heuristics to select the replica server.

The most common non-adaptive server selection algorithm is *round-robin* [Szymaniak et al., 2003], which balances load among them in sending all requests to replica servers in round order. This simple algorithm is efficient for clusters, where all the replica servers are installed at the same place. However, the round-robin server selection algorithm is not suitable for wide area distributed system. An another non-adaptive algorithm assumes that the replica server load and the client-server distance are the most influencing factors. This algorithm takes into account the client-server distance to direct requests to the replica servers in such a way that load is balanced among them. The Cisco DistributedDirector [Delgadillo, 1999] applied some non-adaptive server selection algorithms. They consider also the percentage of received client requests and assume that the server receiving more requests is more powerful. They support random request distribution that chooses randomly the replica server. In addition, there are other non-adaptive algorithms that direct requests to the closest replica.

According to adaptive server selection algorithms, we survey now some representative proposals. Globule project [Pierre and Van Steen, 2006] proposes an adaptive server selection algorithm that always chooses the server closest to the users in terms of network proximity. They use path length as estimation metric that is updated periodically. This algorithm seems inefficient (that is proven in [Huffaker et al., 2002]) because the used metric is passive and it does not introduce any additional traffic to the network. In another proposal, Karger et al. [Karger et al., 1999] propose an algorithm to adapt to hotspots. They base on content URL to generate a hashing function from various identifiers. They use this hashing function to route the user requests to a logical ring involving cache servers with IDs from the same space. This algorithm directs user requests to the cache server holding the smallest ID. In two proposals [Andrews et al., 2002, Ardaiz et al., 2001], the authors use user-server latency as evaluation metric. They use client access logs and passive server-side latency to measure the client-server latency. The replica server having the minimal latency is chosen. The drawback of this approach is that it needs the maintenance of central database measurements, which restricts the scalability of the system. In another approach, Cisco DistributedDirector [Delgadillo, 1999] proposes an adaptive server selection algorithm, which carries out a weighted combination of three metrics: inter-AS distance, intra-AS distance and end-to-end latency. Using three metrics makes this approach flexible. However, the deployment for metric measurement is complex and costly. In addition, the active measurement technique gives more additional traffic to the system. Akamai [Akamai, 1998, Dilley et al., 2002] propose an adaptive algorithm to avoid the flash crowds problem. Their used metrics are replica server load, the reliability of loads be-

tween the client and each of replica server and also the available bandwidth of a server.

**1.3.2.3.2 Server selection mechanisms** There are many techniques to direct the user requests to the appropriate server among various replica servers. These techniques are classified into five categories [Peng, 2004]:

1. *Client multiplexing*: For this mechanism, the client holds a address list of candidate replica servers. The client uses this list to choose the replica server to send the requests. Concretely, the DNS server sends the IP addresses of servers to client. After, DNS resolver at client side chooses a server among these based on evaluation metrics such as response time. To obtain these metrics, DNS resolver makes probes to the servers and also collects reports from the clients on performance of past accesses to these servers. Two approaches [Bhattacharjee et al., 1997, Beck and Moore, 1998] have applied this mechanism. The advantage of this mechanism is that there is no additional communication of request processing. However, there are following drawbacks. First, the client has to use customized DNS resolver. So, their software (e.g. web browser, proxy) must be modified in such a way that the resolver relies on client performance. Second, the DNS infrastructure is dependent on DNS response caching. Consequently, one can not change dynamically replica servers at the risk of using stale replica server sets. Moreover, reducing the caching time increases the DNS queries, that causes the bottleneck issue for the DNS infrastructure.
2. *HTTP redirection*: This mechanism is very simple and not efficient for a server selection technique. Generally, all of client requests are sent to the origin server. Then, the origin server re-directs the browser to a new URL at the HTTP protocol level. In that way, the origin server is the only node that carries out redirecting requests. So, it could become a bottleneck.
3. *DNS indirection*: The main difference between *DNS indirection* and the *Client multiplexing* mechanism is that while in *Client multiplexing* mechanism, choosing a replica server occurs at the client side, the *DNS indirection* realizes the server selection at DNS infrastructure. Generally, some domain name server implementations allow DNS server to assign a host domain name to a set of IP addresses and choose one of them for every client request. With this mechanism, the replica server infrastructure can be easily changed without any danger of using out of date information.
4. *Anycasting*: For this mechanism, an anycast address (an IP anycast address or a URL of content) is a notion that defines a group of servers providing the same service. The client adds an anycast address in the destination address field of packets and sends them to the server group. Then, anycast-aware routers direct the packet to at least one of the servers in this anycast group. This anycast-based routing can be implemented into the existing Internet routing infrastructure.

5. *Peer-to-peer*: A peer-to-peer system is today a broad trend on the Internet for disseminating content. Each affiliation in a peer-to-peer system has participant nodes. The affiliations constitute an ad-hoc network. Since the network changes continuously, each node can not have the complete information in time about the entire network. The problem of how to select efficiently the best server in a distributed manner without incurring high overhead of propagating the routing information becomes actually a trend for network operator.

## 1.4 Conclusion

Nowadays, service providers should observe and react quickly on quality problems before users quit the service (customer churn problem). Facing this quality competition, the notion of Quality of Experience has been introduced, combining user perception, experience and expectations. Certainly, the known QoS concept designates a set of technical criteria to ensure network service, but QoE is a notion that represents the overall level of end-user satisfaction of a service. The QoE is expressed by human feelings like good, excellent, bad, etc. QoE-aware network system is a promising new notion in which service providers are aware of user perception and can consequently adapt to the dynamic environment to obtain acceptable and predictable QoE. Furthermore, the QoE impacts on the setting of internal parameters of the network. In fact, the end-to-end (e2e) quality is the most important aspect to be achieved for operators, service providers and also for users. QoE is a new concept to make e2e QoS more clearly taking into account the user's experience. QoE also takes into account the satisfaction and the perception of the users when using network services. Recently, many researches and proposals have been made in order to measure, evaluate, and improve QoE in network systems.

The e2e QoE model is necessary for every network system. That motivates us to apply this model to an user-centered architecture: the Content Distribution Network. This chapter has surveyed the QoE notion and the general CDN architecture. The chapter plays an important role to clarify our way of integrating the QoE model into the CDN architecture. That is also the basic premise to explain the need of applying QoE model to a CDN.

## Chapter 2

# Proposed routing algorithm for CDN architecture: QQAR

In this chapter, we will present the proposed QoE-based routing algorithm QQAR, which is based on the Q-Learning, an algorithm of Reinforcement Learning (RL). First, this chapter will present briefly the knowledge base of the RL and Q-Routing. After, we will focus on our proposed routing algorithm QQAR.

### 2.1 Introduction

Over the last few decades QoS played an important role in the QoS framework of the Internet. The QoS routing strategy determines the suitable paths for different types of data traffic sent by various applications. The goal is to maximize the utilization of network resources while satisfy the traffic requirements in the network. To achieve this objective, it is required to develop a routing algorithm that determines multi-constrained paths taking into account the network state and the traffic needs (e.g., delay, loss-rate, bandwidth, etc.). In fact, the core of any QoS routing strategies is the path computation algorithm. The algorithm has to select several alternative paths, which can satisfy a set of constraints like end-to-end delay and bandwidth requirements. However, the algorithms to solve such a problem meet generally high computational complexity. In fact, Wang et al. [Wang and Crowcroft, 1996] proved that an end-to-end QoS model with more than two non-correlated criteria is NP-complete. Apart from the problem of complexity, current QoS routing strategies also have other problem: the lack of consideration of end-user perception. Actually, current QoS routing strategy just focuses on technical parameters in the network in ignoring the end-user perception. Unfortunately, from time to time end-user satisfaction plays the key role in the competition of network operators. This challenge has devised a new human-centered metric, called Quality of Experience.

QoE-based network system is a promising new notion in which service providers are aware of user perception and can consequently adapt to the dynamic environment to obtain acceptable and predictable QoE. E2E-QoE (End-to-End QoE) refers to the overall satisfaction and acceptability of an application or service as perceived by the end-user.

E2E-QoE expresses a subjective measure of user's experiences and becomes a major issue for telecommunication companies because their businesses are highly dependent on customer satisfaction. The average revenue per user can only be increased by value-added services taking into account the complete end-to-end system effects. As an important measure of the end-to-end performance at the service level from the user's perspective, the QoE is an important metric for the design of systems and engineering processes. With the QoE paradigm, we can reach a better solution and prevent the NP-complete problem due to the goal of maintaining only QoE criterion instead of optimizing multiple QoS criteria. Furthermore, QoE is really a contribution to real-adaptive control network's components in the future internet.

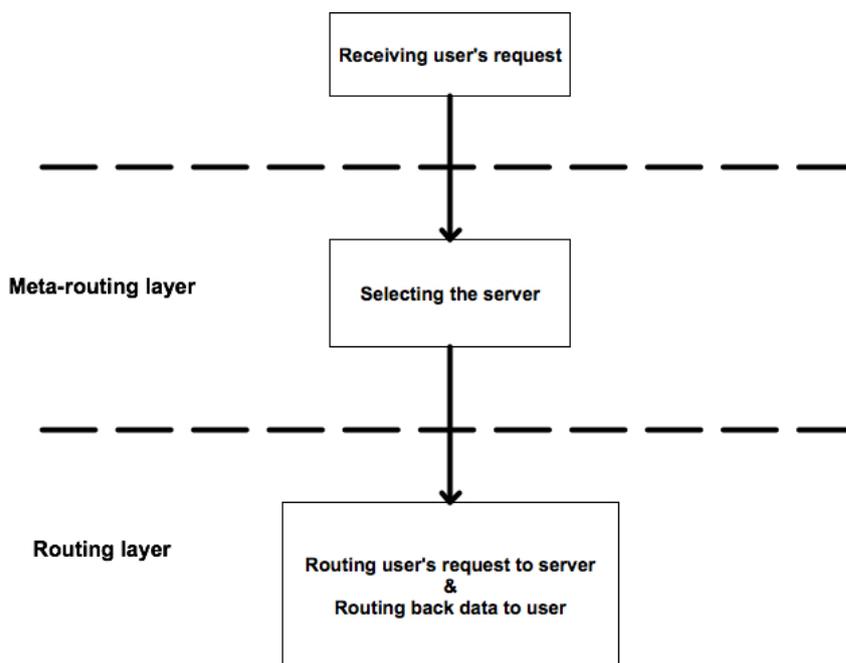


Figure 2.1: CDN processes

Such advantages of QoE-based network system motivate us to devise a new kind of routing algorithm for the CDN architecture: QoE-based routing algorithm. Concretely, in the CDN context, after receiving a user's request, the server selection module is used to choose the most appropriate server in using a selection algorithm (Fig. 2.1). Then, IP address of the chosen server is attached to the request, which is sent down to the routing layer. The latter will take into account the routing process in using a routing strategy. Our proposed routing strategy adapts to the network environment changes to improve the user perception. It is a solution to react in real-time to maintain a good QoE level. QoE-based routing system does not wait until the client leave the service. In contrast, it reacts quickly after detecting quality degradation to improve the QoE value of the network system. In addition, this new kind of routing can resolve problems of QoS

routing framework. First, QoE-based routing strategy is able to avoid the NP-complete problem. It optimizes only the QoE metric instead of optimizing multiple QoS criteria due to the fact that satisfying end-user is the most important goal of network operators. Users disregard the technical parameters in the core network when using services. All they care about is service quality they perceive. Secondly, QoE-based routing can avoid also the churn problem (i.e., the client leaves a service after being not satisfied). In fact, clients will never leave the service if their experience is put on top priority.

The scalability of a CDN predicts fluctuations of the environment in the core network. In order to react and adapt to changes of network environment, routing strategy needs an efficient learning method. There are lots of learning methods such as supervised learning (e.g., neural network) that was applied to various routing approaches. In this thesis, we have chosen the Reinforcement Learning (RL), which is defined as learning a policy, a mapping of observations into actions, based on feedback from the environment. RL is considered as natural framework for the development of adaptive strategies by trial and error in the process of interaction with the environment. Such method is appropriate for a routing approach.

In this chapter, we present the proposed routing algorithm, called QQAR (QoE QLearning-based Adaptive Routing protocol). First, we survey the Reinforcement Learning theory. Then, we describe the QQAR algorithm in mapping the RL model into our routing model.

## 2.2 Reinforcement Learning and Q-Routing

As mentioned previously, based on Reinforcement Learning we propose a new routing algorithm that takes into consideration the QoE as environment feedback. Before focusing on the proposal, we present the fundamental of RL as well as Q-Learning algorithm.

The basis of RL is the *Markov Decision Process* (MDP), which provides a mathematical framework for modeling decision-making. MDP is a discrete time stochastic control process. The MDP model can be considered as a Markov chain to which one adds a decision component. Like the rest of his family, he is among others used in artificial intelligence for control of complex systems such as intelligent agents. If the probabilities or rewards in a MDP are unknown, the problem becomes Reinforcement Learning (RL) [Sutton and Barto, 1998]. The latter is a learning method for which only a qualitative measure of error is available. In this case, the agent receives rewards from its environment and reacts to it by choosing an appropriate action for his behavior. His reaction is then judged against a predefined objective called “reward”. The agent receives this “reward” and learns to modify its future actions and achieve an optimal behavior. Action leading to a negative rating will be used less than a positive action.

The RL results from two simple principles:

- For a given state, if an action immediately causes something bad, then the system learns not to take that action when it is in this state in the future.

- For a given state, if all possible actions lead to something bad, then the system will learn to avoid being in this state.

The RL is a learning method for an autonomous agent behavior to adapt to its environment. The goal is to achieve objectives without any external intervention. Figure 2.2 illustrates the principle of reinforcement learning.

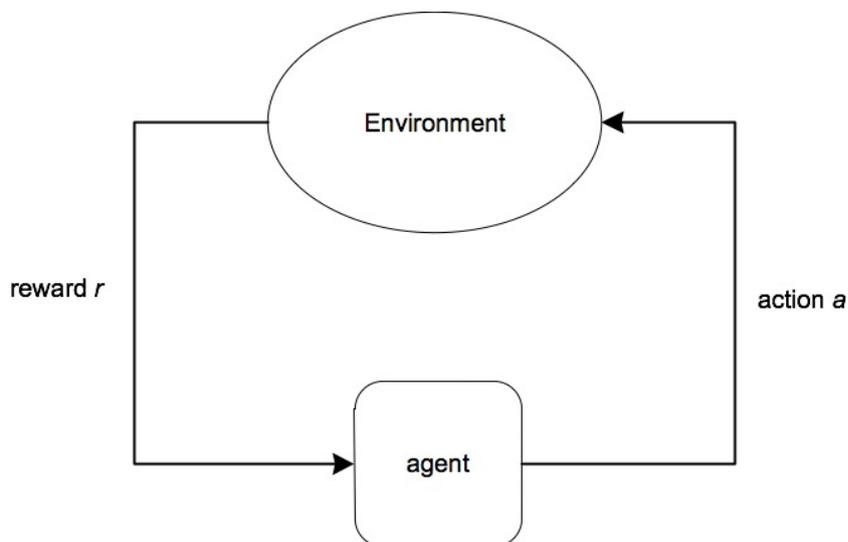


Figure 2.2: Reinforcement Learning model

In a reinforcement learning (RL) model [Kaelbling et al., 1996] [Sutton and Barto, 1998], an agent is connected to its environment via perception and action (Fig. 2.2). Whenever the agent receives as input the indication of current environment state, the agent then chooses an action,  $a$ , to generate as output. The action changes the state of the environment, and the value of this state transition is communicated to the agent through a scalar reinforcement signal,  $r$ . The agent is perform to choose actions that tend to increase by systematic trial and occurrences, guided by any algorithm.

Formally, based on the MDP, the RL model consists of:

- a discrete set of environment states:  $\mathcal{S}$ ;
- a discrete set of agent actions:  $\mathcal{A}$ ;
- a set of scalar reinforcement signals:  $\mathcal{R}$ ;

So the agent's goal is to find a policy  $\pi$ , mapping states to actions, that maximizes some long-run measure of reinforcement. In RL paradigm, one uses two types of value functions to estimate how good is for the agent to be in a given state:

- State-Value function  $V^\pi(s)$ : expected return when starting in  $s$  and following  $\pi$  thereafter.

$$V^\pi(s) = E_\pi\{R_t | s_t = s\} \quad (2.1)$$

- Action-Value function  $Q^\pi(s, a)$ : expected return starting from  $s$ , taking the action  $a$ , and thereafter following policy  $\pi$

$$Q^\pi(s, a) = E_\pi\{R_t | s_t = s, a_t = a\} \quad (2.2)$$

where:  $t$  is any time step and  $R$  is the reward.

Regarding policy iteration notion, the reason to compute the value function for a policy is to find better policies. Once a policy,  $\pi$ , has been improved using  $V^\pi$  to yield a better policy,  $\pi'$ , one can then compute  $V^{\pi'}$  and improve it again to yield an even better  $\pi''$ . One can thus obtain a sequence of monotonically improving policies and value functions:

$$\pi_0 \xrightarrow{E} V^{\pi_0} \xrightarrow{I} \pi_1 \xrightarrow{E} V^{\pi_1} \xrightarrow{I} \pi_2 \xrightarrow{E} \dots \xrightarrow{I} \pi^* \xrightarrow{E} V^* \quad (2.3)$$

where  $E$  denote a policy evaluation and  $I$  denotes a policy improvement.

The goal is to find an optimal policy that maps states to actions so as to maximize cumulative reward, which is equivalent to determining the following Q function (Bellman equation):

$$Q^*(s, a) = r(s, a) + \gamma \sum_{s'} P(s'|s, a) \max_{a'} Q^*(s', a') \quad (2.4)$$

where  $\gamma$  is the discount rate. The next subsection presents in detail the mathematical model of RL.

### 2.2.1 Mathematical model of Reinforcement Learning

We take an example of a model with three states (Fig. 2.3). Each state can make two possible actions  $a_1$  and  $a_2$ . The transition from one state to another one is performed according to the probability  $P_{21}^{a1}$ ,  $P_{21}^{a2}$ , leading to a reward signal respectively  $r_{23}^{a1}$ ,  $r_{23}^{a2}$ .

These are some notations:

- $X$ : the state set of the system and  $x_t \in X$  is a state of the system at time  $t$ .
- $A_x$ : the set of possible actions of state  $x$ .
- $A$ : set of all actions.
- $a_t \in A_x$ : Applied action in time  $t$  of state  $x_t$
- $P_{xy}^a = Pr(x_{t+1} = y | x_t = x, a_t = a)$ : the probability of transition from state  $x$  to  $y$  by action  $a$ .

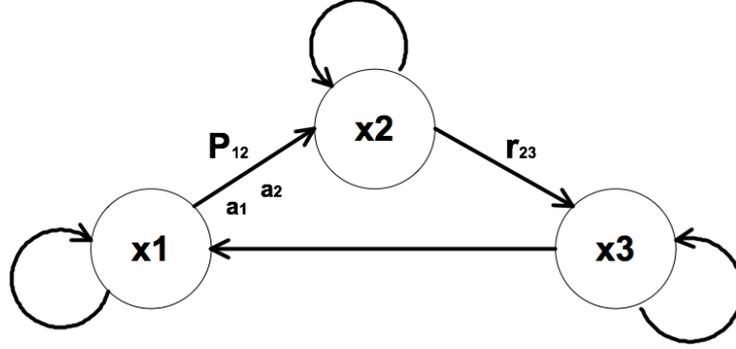


Figure 2.3: Example of Markov process with 3 states and 2 actions

- $R$ : the set of rewards the system can receive from environment, such as:

$$\begin{aligned} r : (X, X, A) &\rightarrow R \\ (x_t, y_t, a_t) &\mapsto r_{x_t y_t}^{a_t} \end{aligned} \quad (2.5)$$

where  $r$  is reward of environment after applying action  $a_t$  during the transition of state  $x_t \rightarrow y_t$

- $\pi$ : the adopted policy that chooses the action  $a$  in state  $x : a = \pi(x)$ , such as:

$$\begin{aligned} \pi : X &\rightarrow A \\ x &\mapsto a \end{aligned} \quad (2.6)$$

- $\Pi$ : the set of policies, such as:  $V^\pi$ : the state value function:

$$\begin{aligned} V^\pi : (X, \Pi) &\rightarrow R \\ (x, \pi) &\mapsto V^\pi(x) \end{aligned} \quad (2.7)$$

- $V^\pi(x)$ : the measure of quality of policy  $\pi$  of state  $x$ .

- $U$ : the set of the couples action-state.

- $Q^\pi$ : the value function of couple action-state, such as:

$$\begin{aligned} Q^\pi : (U, \Pi) &\rightarrow R \\ (x_0, a, \pi) &\mapsto Q^\pi(x_0, a) \end{aligned} \quad (2.8)$$

The function  $Q^\pi(x_0, a)$  measures the quality of the policy  $\pi$  when the system is in the state  $x_0$  and the action  $a$  is chosen.

The definition of the functions  $Q$  and  $V$  (Q-value and V-value) is fundamental because the optimal policy is the one leading to the best quality, which is defined by:

$$\pi^* = \arg \max_{a \in A_x} Q^*(x, a) \quad (2.9)$$

The optimal policy maximizes the criteria  $V^\pi$ . Therefore, the reinforcement learning algorithms aim to improve iteratively the function  $\pi$ .

### 2.2.2 Value functions

In this subsection, we clarify the relationship between the value functions and decision policies.

#### 2.2.2.1 Rewards

A reward in a given time  $r_t$  is the received signal by the agent during the transition from state  $x_t$  to state  $x_{t+1}$ , with the realization of action  $a_t$ . It is defined by:  $r_t = r_{xy}^a$ .

The global reward represents the sum of all punctual rewards. It is used to evaluate the system behavior over time. When a trial is finished, this sum is written as:

$$R_{et} = \sum_{k=0}^T r_{t+k+1} \quad (2.10)$$

One defines a sum of damped-off rewards by:

$$R_{et} = \sum_{k=0}^T \gamma^k r_{t+k+1} \quad (2.11)$$

where  $0 < \gamma < 1$  and  $T$  are the end of the test.

#### 2.2.2.2 Definitions of functions of V-value and Q-value

The value function, V-value, represents expectation of global sum of punctual rewards using the policy  $\pi$ . The goal of this function is to assess the quality of states under a given policy. It is defined as follows:

$$V^\pi(x) = E_\pi[R_{et}|x_t = x] = E_\pi \left[ \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | x_t = x \right] \quad (2.12)$$

Similarly, a function Q-value,  $Q^\pi(x, a)$ , represents expectation of global sum when the system realizes the action  $a$  under the policy  $\pi$ , such as:

$$Q^\pi(x, a) = E_\pi \left[ \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | x_t = x, a_t = a \right] \quad (2.13)$$

The estimated value of these functions is necessary in most RL algorithms. These value functions can be estimated from experiments in [Sutton and Barto, 1998] [Littman, 1994].

#### 2.2.2.3 Value function and optimal policy

A policy  $\pi = \pi^*$  is optimal if  $V^* = V^{\pi^*}(x) \geq V^{\pi^i}(x)$  with  $\forall x \in X$  and for all policies  $\pi^i$  [Littman, 1994].

The goal of RL is to find an optimal policy  $\pi^*$  that maximizes the expectation of global reward:

$$\pi^* = \arg \max_{\pi} V^{\pi}(x) = \arg \max_{\pi} E_{\pi} \left[ \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | x_t = x \right] \quad (2.14)$$

The equation 2.12 can be expressed in the form:

$$V^{\pi}(x) = \sum_{y \in X} P_{xy}^{\pi(x)} \left[ r_{xy}^{\pi(x)} + \gamma V^{\pi}(y) \right] \quad (2.15)$$

This equation means that the utility of a state under policy  $\pi$  is an expectation over all available states of the sum of the global reward received by applying  $\pi(x)$  and the value of  $y$  under the policy  $\pi$ .

This equation also applies to all policies, in particular the optimal policy  $\pi^*$ :

$$V^*(x) = V^{\pi^*} = \sum_{y \in X} P_{xy}^{\pi^*(x)} \left[ r_{xy}^{\pi^*(x)} + \gamma V^*(y) \right] \quad (2.16)$$

By definition, all optimal policies have the same value function  $V^*$ . Therefore, the action selected by the optimal policy must maximize the following expression:

$$\sum_{y \in X} P_{xy}^a \left[ r_{xy}^a + \gamma V^*(y) \right] \quad (2.17)$$

Which implies:

$$V^*(x) = \max_{a \in A_x} \sum_{y \in X} P_{xy}^a \left[ r_{xy}^a + \gamma V^*(y) \right] \quad (2.18)$$

Finally, one can determine the following optimal policy:

$$\pi^*(x) = \arg \max_{a \in A_x} \sum_{y \in X} P_{xy}^a \left[ r_{xy}^a + \gamma V^*(y) \right] \quad (2.19)$$

These equations require the knowledge of the system model  $(r_{xy}^a, P_{xy}^a)$ . However, if this model was unknown, the Q-value would solve this problem.

The expression  $Q^{\pi}(x, a)$  can be expressed in the following recurrent form:

$$Q^{\pi}(x, a) = \sum_{y \in X} P_{xy}^a \left[ r_{xy}^a + \gamma V^{\pi}(y) \right] \quad (2.20)$$

The function  $Q^* = Q^{\pi^*}$  is corresponding to the optimal policy, it does not necessarily choose the action  $\pi(x)$  with the policy  $\pi$ .

$V^*$  can be expressed in the following form (optimal equation of Bellman):

$$V^* = \max_{a \in X} [Q^*(x, a)] \quad (2.21)$$

and the expression of  $\pi^*(x)$  becomes:

$$\pi^*(x) = \arg \max_{a \in X} [Q^*(x, a)] \quad (2.22)$$

These last two equations can be evaluated without knowledge of the system model. Therefore, the problem of RL is to learn the Q-values and V-values. In other words, it is to estimate it in the absence of prior knowledge of the model. One of the efficient methods to solve this problem is the Q-Learning we present in the next section.

### 2.2.3 Q-Learning

The Q-learning [Watkins and Dayan, 1992] is a learning method that does not require any model. The goal is to find an approximation of the Q-function approaching  $Q^*$ . This involves solving the following Bellman optimal equation [Mellouk, 1994] [Bellman and Kalaba, 1959]:

$$V^* = \max_{a \in A_x} [Q^*(x, a)] \quad (2.23)$$

The Q-learning algorithm can be summarized in algorithm 2.2.3.1. Watkins [Watkins and Dayan, 1992] has proved that the values of  $Q(x, a)$  reach the optimal values of  $Q^*(x, a)$  with the probability of 1.

---

**Algorithm 2.2.3.1** Q-learning algorithm

---

**Input:**  $t = 0$ ,  $x$ : an initial state

**repeat**

    Adjust  $Q$ , such as:

$$Q(x, a) = Q(x, a) + \alpha [r_{xy}^a + \gamma \max_{b \in A_y} Q(y, b) - Q(x, a)]$$

$x = y$

**until**  $Q \simeq Q^*$

---

The Q-learning method has obvious advantages. First, it requires no prior knowledge of the system. Secondly, the perspective of the agent is very local, which is appropriate for the problem of dynamic routing where each router has to decide the routing path without knowing the status of all routers. Furthermore, the implementation requires only a Q-values table comparable to a routing table. This Q-value table is updated after each executed action. However, this algorithm has a drawback that is found generally in all reinforcement learning methods: it is the guarantee of convergence from  $Q$  to  $Q^*$ . So, the strong assumptions have to be admitted: the process must be Markovian and stationary, and all couples of state-action  $(x, a)$  must be executed with an infinite number of times. The last assumption is not crucial in all topic of reinforcement learning. Indeed, with the problems in robotics where the robot learns to move in a given environment or learning issues like learning games (e.g. backgammon, chess, etc.), one gives a learning period where the agent behavior is irrelevant and where one can afford many explorations. Usually, at the end of this learning period, one continues the exploration leading to better results. In our problem, the environment is not static, but dynamic and therefore

there is no separation between learning period and operation period. The number of executions of pairs  $(x, a)$  must be not only finite, but enough to ensure the convergence to  $Q^*$  without falling into a local minimum. However, it should be small enough so as not to saturate the system. To propose an adaptive routing based on Q-Learning algorithm, the study of a solution to the problem of exploration is developed.

There were already several routing approaches based on Q-Learning, called “Q-Routing”. The difference between these approaches and the ours is that they used just network parameters as reward of environment, while we consider user’s feedback as rewards. The next subsection present generally the Q-Routing.

### 2.2.4 Q-Routing

This subsection discusses the attempt of applying the Q-learning algorithm to adaptive routing protocol, called Q-Routing [Boyan and Littman, 1994]. The exploitation and exploration framework used in Q-Routing is the basis of proposed our routing algorithm.

First, we present the routing information stored at each node as Q-tables. Q-Routing is based on Q-Learning to learn a representation of network’s state, Q-values, and it uses these values to make control decisions. The goal is to learn an optimal routing policy for the network. The Q-values represent the state  $s$  in the optimization problem of routing process. The node  $x$  uses its Q-tables  $Q_x$  to represent its own view of the network’s state. The action  $a$  at node  $x$  is used to choose neighbor  $y$  such that it maximizes the obtained Q-value for a packet destined for node  $d$  (destination).

In Q-Routing, each node  $x$  has a table of Q-values  $Q_x(y, d)$ , where  $d \in V$  the set of all nodes in the network, and  $y \in N(x)$  is the set of all neighbors of node  $x$ . The  $Q_x(y, d)$  is the best Q-value that a packet would obtain to reach the destination  $d$  from node  $x$  when sent via the neighbor  $y$ .

- Exploitation phase: In the exploitation phase, the Q-values stored at each node are used for making locally greedy routing decisions. More precisely, when a node  $x$  receives a packet  $P(s, d)$  destined for node  $d$ . Node  $x$  looks at the vector  $Q_x(*, d)$  in his Q-table and selects his neighbor node  $\hat{y}$  so that the  $Q_x(\hat{y}, d)$  value is maximum<sup>1</sup>. In that way, the node  $x$  makes a locally greedy decision by forwarding the packet to the neighbor so that the Q-value is the maximal one.
- Exploration phase: After receiving the packet  $P(s, d)$  sent from node  $x$ , destined for node  $d$ , node  $y$  sends back to  $x$  its best estimate  $Q_y(\hat{z}, d)$  for the destination  $d$ :  $Q_y(\hat{z}, d) = \max_{z \in N(y)} Q_y(z, d)$ . Using the estimate value, the node  $x$  updates its  $Q_x(y, d)$  value:

$$Q_x(y, d)^{\text{new}} = Q_x(y, d)^{\text{old}} + \alpha(Q_x(y, d)^{\text{est}} - Q_x(y, d)^{\text{old}}) \quad (2.24)$$

---

<sup>1</sup>In this thesis, we are working on optimizing the obtained QoE value, so for us the optimized value is the maximal one. Probably, in other approaches where they work on delay optimization, the optimized value is the minimal one.

The goal of Q-Routing is to balance the exploitation and exploration phase. First, we survey some related works of Q-Routing to highlight our motivation to propose a novel routing algorithm.

### 2.2.5 Related works and motivation

The crucial issue of the success of distributed communication and heterogeneous networks in the Next Generation Network (NGN) is the routing mechanism. Every routing protocol tries to transport traffic from source to destination in maximizing network performance, while minimizing costs. In fact, changes of network environment influence the end-user perception [Shaikh et al., 2010a]. For that reason, most existing adaptive routing techniques are designed to be capable to take into account the dynamics of the network.

The idea of applying reinforcement learning to routing in networks was firstly introduced by J. Boyan et al. [Boyan and Littman, 1994]. The authors described the Q-routing algorithm for packet routing. Each node in the network uses a Reinforcement Learning module. They try to keep accurate statistics on which routing decisions lead to minimal delivery time using only local communication. However, this proposal focuses on optimizing only one basis QoS metric: the delivery times. User perception (QoE) is not considered.

The proposals in [Mellouk et al., 2009] and [Esfahani and Analoui, 2008] present two QoS based routing algorithms based on a multi-path routing approach combined with the Q-routing algorithm. The learning algorithm finds K best paths in terms of cumulative link cost and optimizes the average delivery time on these paths. The technique used to estimate the end-to-end delay is based on the Q-Learning algorithm to take into account dynamic changes in networks. In the proposal of Hoceini et al. [Hoceini et al., 2008], AV-BW Delay Q-Routing algorithm uses an inductive approach based on trial/error paradigm combined with swarm adaptive approaches to optimize three QoS different criteria: static cumulative cost path, dynamic residual bandwidth and end-to end delay. Based on KOQRA, the approach presented here adds a new module to this algorithm dealing with a third QoS criterion which takes into account the end-to-end residual bandwidth.

We can see that the perception and satisfaction of end-users are ignored in all of these approaches above. This poses the problem of choosing the best QoS metric that is often complex. However, QoE comes directly from the user perception and represents the true criterion to optimize. Some other proposals are introduced to solve this problem.

An overlay network for end-to-end QoE management is proposed by B. Vleeschauwer et al. [Vleeschauwer et al., 2008]. Routing around failures in the IP network improves the QoE in optimizing the bandwidth usage on the last mile to the client. Components of overlay network are located both in the core and at the edge of the network. However, this proposal does not use any adaptive mechanism.

G. Majd et al. [Majd et al., 2009] present a new QoE-aware adaptive mechanism to maximize the overall video quality at the client. Overlay path selection is dynamically done based on available bandwidth estimation, while the QoE is evaluated using the

PSQA tool [Rubino, 2005]. In this model, the video server will decide to keep or change video streaming strategies based on end-user feedback. However, the adaptive mechanism is quite simple because it is not based on any learning method. Furthermore, using source routing is also a drawback of this approach because this fixed routing is really not adaptive to environment changes.

M. Wijnants et al. [Wijnants et al., 2010] propose a QoE optimization platform that can reduce problems in the delivery path from service provider to users. The proposed architecture supports overlay routing to avoid inconsistent parts of the network core. In addition, it includes proxy components that realize last mile optimization through automatic bandwidth management and the application of processing on multimedia flows. In [Lopes Gomes et al., 2011], the authors propose a wireless mesh networks (WMN) routing protocol based on optimized link state routing (OLSR) to assure both the QoS and the QoE for multimedia applications. They used the dynamic choice of metrics and a fuzzy link cost (FLC) for routing multimedia packets. Concretely, fuzzy system uses two link quality metrics: expected transmission count (ETX) and minimum delay (MD). However, the QoE in these two approaches above are considered as linear combination of QoS criteria. In fact, as mentioned in chapter 1, QoE is the subjective factor that represents user perception, not a linear combination of QoS parameters.

Our approach gives a routing mechanism based on Q-learning algorithm in order to construct an adaptive, evolutionary and decentralized routing system. That is presented in the next section.

## 2.3 QQAR routing algorithm

This section presents the proposed routing algorithm for the routing layer of a CDN architecture, called QQAR (QoE QLearning-based Adaptive Routing protocol). This QoE-aware routing protocol is based on the Q-Learning algorithm.

Our idea to take into account end-to-end QoE consists to develop an adaptive mechanism that can retrieve the information from its environment and adapt to initiate actions. The action choice should be executed in response to end-users feedback (the QoE feedback). We named this algorithm *QQAR* (QoE QLearning-based Adaptive Routing).

Concretely, the system integrates the QoE measurement in an evolutionary routing system in order to improve the user perception based on Q-Learning algorithm to choose the “best optimal QoE paths” (Fig. 2.4). In our approach, we have mapped RL model to our routing model in the context of learning routing strategy. So in that way, the routing process is built according to maintaining the best user perception.

As mentioned previously, integrating QoE into a system reduces the complexity problem associated with NP-complete of the QoS model. The objective function is related to user satisfaction, i.e the selection of certain parameters, or even one, not several uncorrelated parameters. We work on the oriented selection gradient on this parameter. In this section, first, we present our formal parametric model based on continuous time evolving function to prove the convergence of our approach. We then describe in detail the QQAR algorithm. The learning and selection processes explain the way we apply

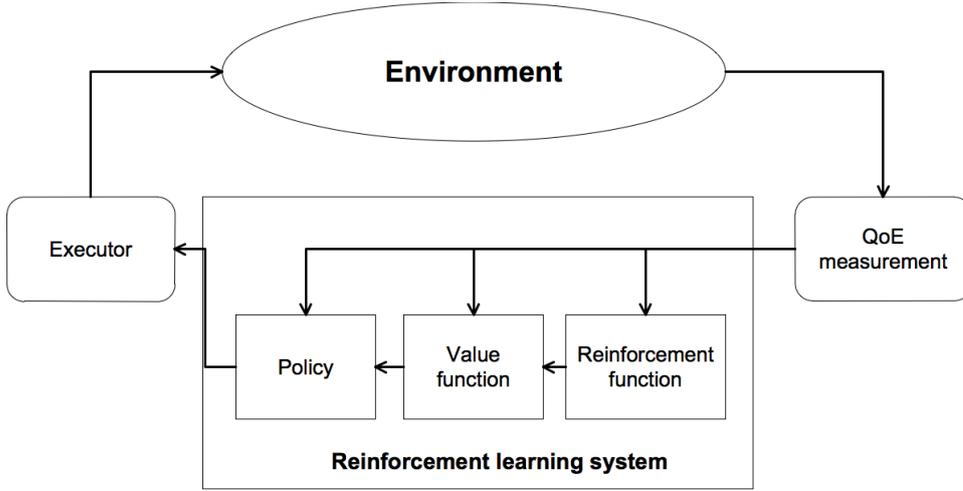


Figure 2.4: Q-Learning-based model

Q-learning algorithm to QQAR.

### 2.3.1 Formal Parametric model

Our system is based on different modules. Each one optimizes a local cost function. To achieve this, we propose an unified formal model [Mellouk et al., 2012] to determine the convergence of our approach. Continuous learning in our system involves changing the parameters of the network model using an adaptive update rule as follows:

$$w(n) = w(n - 1) - \epsilon(n)F(x(n), w(n - 1)) \quad (2.25)$$

where  $w(n)$  denotes the parameters of the whole system at time  $n$ ,  $x(n)$  is the current traffic and  $\epsilon$  is the modification step.  $F$  is either the gradient of a cost function or a heuristic rule.  $x$  can be defined by a probability density function  $p(x)$ , and  $w$  the parameters of the learning system. The concept behind our system is based on the fact that each iteration uses an instance of a real flow instead of a finite training set. The goal of this learning system consists of finding the minimum of a function  $C(w)$  named the expected risk function and the average update is therefore a gradient algorithm which directly optimizes  $C(w)$ . For a given state of the system, we can define a local cost function  $J(x, w)$  which measures how well our system behaves on  $x$ . The goal of learning is often the optimization of some function of the parameters and of the concept to learn which we will call the *global cost function*. It is usually the expectation of the local cost function over the space  $X$  of the concept to learn:

$$C(w) = \int_x J(x, w)p(x)dx = E_X\{J(x, w)\} \quad (2.26)$$

Most often we do not know the explicit form of  $p(x)$  and therefore  $C(w)$  is unknown.

Our knowledge of the random process comes from a series of observations  $x_i$  of the variable  $x$ . We are thus only able to calculate  $J(x, w)$  from the observations  $x_i$ .

A necessary condition of optimality for the parameters of the system is:

$$\Delta C(w) = E_X\{\Delta_w J(x, w)\} = 0 \quad (2.27)$$

where  $\Delta$  is the gradient operator. Since we do not know  $\Delta C(w)$  but only the result of  $\Delta_w J(x, w)$ , we cannot use classical optimization techniques to reach the optimum. One solution is to use an adaptive algorithm as follows:

$$w(n) = w(n-1) - \gamma(n)\Delta_w J(x(n), w(n-1)) \quad (2.28)$$

where  $\gamma$  is the gradient step. There is close similarity between Eq. 2.25 and Eq. 2.28 and when learning in the second stage aims at minimizing a cost function, there is a complete match. This formulation is particularly adequate for describing adaptive algorithms that simultaneously process an observation and learn to perform better. Such adaptive algorithms are very useful in tracking a phenomenon that evolves over time, as in user perception.

### 2.3.2 QQAR algorithm

Our approach is based on Q-Learning algorithm. In this section, we summarize some general idea of RL and Q-Learning.

Q-Learning [Watkins and Daylan, 1992] is a technique to solve specific problems by using the RL approach [Sutton and Barto, 1998]. Even the Reinforcement Learning has been introduced for a long time; this concept is used/applied recently in many works in many disciplines, such as game theory, simulation-based optimization, control theory, Genetic Algorithms, etc. We cite here some recent RL advances: RL-Glue [Tanner and White, 2009], PyBrain [Schaul et al., 2010], Teachingbox [Ertel et al., 2009], etc.

We explain now how a RL framework works. In a RL framework, an agent can learn control policies based on experience and rewards. In fact, Markov Decision Process (MDP) is the underlying concept of RL.

An MDP model is represented by a tuple  $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R})$  where  $\mathcal{S}$  is the set of states,  $\mathcal{A}$  is the set of actions,  $\mathcal{P}(s'|s, a)$  is the transition model that describes the probability of entering state  $s'$  after executing action  $a$  at state  $s$ .  $\mathcal{R}(s, a, s')$  is the reward obtained when the agent executes  $a$  at state  $s$  and enters  $s'$ . The quality of action  $a$  at state  $s$  is evaluated by Q-value  $Q(s, a)$ .

Solving an MDP is equivalent to finding an optimal policy  $\pi : \mathcal{S} \rightarrow \mathcal{A}$ , that maps states to actions such that the cumulative reward is maximized. In other words, we have to find the following Q-fixed points (Bellman equation):

$$Q^*(s, a) = r(s, a) + \gamma \sum_{s'} P(s'|s, a) \max_{a'} Q^*(s', a') \quad (2.29)$$

where the first term on the right side represents the expected immediate reward of executing action  $a$  at  $s$ . The second term is the maximum expected future reward.  $\gamma$  is the discount factor.

One of the most important breakthroughs in RL is an off-policy temporal difference (TD) control algorithm known as Q-learning, that directly approximates the optimal action-value function (Equation 2.29), independent of the policy being followed. After executing action  $a$ , the agent receives an immediate reward  $r$  from the environment. By using this reward and the long term reward it then updates the Q-values influencing future action selection. One-step Q-learning is defined as:

$$Q(s, a) = (1 - \alpha)Q(s, a) + \alpha[r + \gamma \max_{a'} Q(s', a')] \quad (2.30)$$

where  $\alpha$  is the learning rate, which models the rate of updating Q-values.

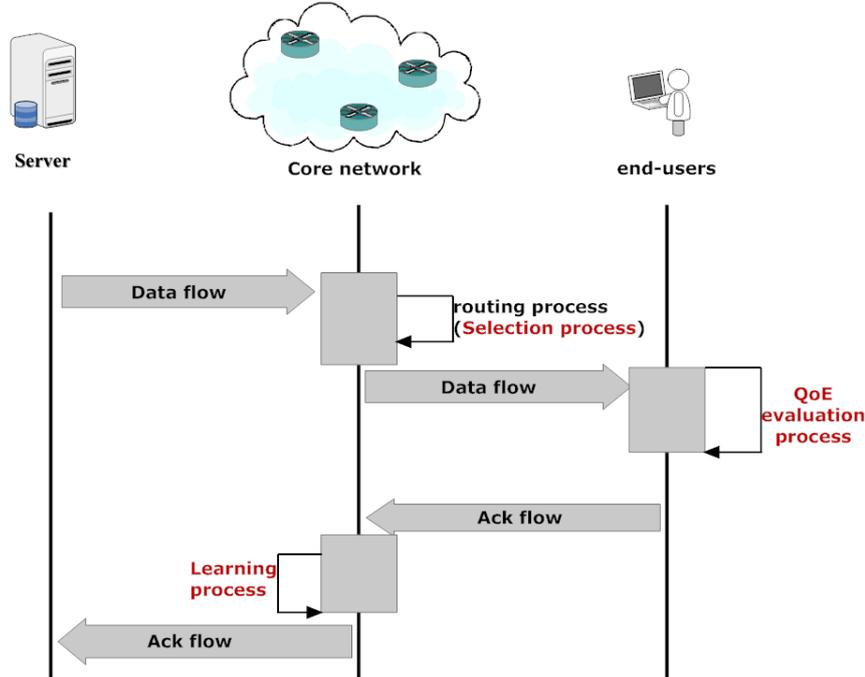


Figure 2.5: QQAR protocol

In order to apply the Q-Learning algorithm to our routing system, we have mapped the RL model to our routing model in the context of learning a routing strategy. We consider each router in the system as a state. The states are arranged along the routing path from the client to the chosen server in a CDN. Furthermore, we consider each link emerging from a router as an action to choose. The system's routing mechanism refers to the policy  $\pi$  (Tab. 2.1). Regarding QoE measurement, we used a method that evaluates the QoE of all nodes (appendix A).

The proposed routing mechanism is illustrated in a diagram in Fig. 2.5

Table 2.1: Mapping RL-model onto routing model

RL-model	Routing model
State	Router
Actions	Links
Policy $\pi$	Routing policy

- **First step** - *Data packet flow*: the server chosen by meta-routing layer sends a data packet to end-user. After receiving this data packet, each node in the routing path evaluates QoE and forwards the packet to the next one in using a selection process (presented in subsection 2.3.5).
- **Second step** - *At user side*: After data reaches end-users, the QoE evaluation process is realized to send back a QoE feedback as an ACK message to the routing path.
- **Third step** - *ACK message flow*: Once a node receives a ACK message, it updates the Q-value of the link that this ACK message just passed through. The update process is introduced below (subsection 2.3.3). It then attaches the QoE measurement result to the ACK message and forwards it to the previous neighbour.

The following diagram (Fig. 2.6) describes the QQAR algorithm:

In the next two subsections, we clarify the two main processes of QQAR: the learning process and the selection process.

### 2.3.3 Learning process

As the analysis of Q-Learning algorithm presented previously, we propose a new method for network exploration. This technique relies on forward exploration [Boyan and Littman, 1994] to which we added a probabilistic mechanism exploration. This mechanism allows us firstly, to explore from time to time optimal paths and secondly, to avoid further congestion of the network as data packets are also used for exploration.

In our model, each router has a routing table that indicates the Q-values of links emerging from this router. For example in Fig. 2.7, node  $y$  has a routing table containing Q values:  $Q_{yz_1}, Q_{yz_2}, Q_{yz_3} \dots Q_{yz_n}$  corresponding to  $n$  links from  $y$  to  $z_i$  with  $i = 1..n$ . Based on this routing table, the optimal routing path can be trivially constructed by a sequence of table look-up operations. Thus the task of learning optimal routing strategy is equivalent to finding the Q-fixed points (Equation 2.29).

As mentioned above, after receiving a data packet, the destination node (end-user) evaluates the QoE. Then, it sends back the feedback (QoE value) as ACK message to the routing path. Each router  $x$  in the routing path receives this message including: the QoE evaluation result (the MOS score<sup>2</sup>) of the previous router ( $q_y$ ) and the maximum value of Q-values ( $\max Q_{yz}$ ) in the routing table of node  $y$  (Fig. 2.7).

---

<sup>2</sup>Mean Opinion Score (MOS) gives a numerical indication of the perceived quality of the media

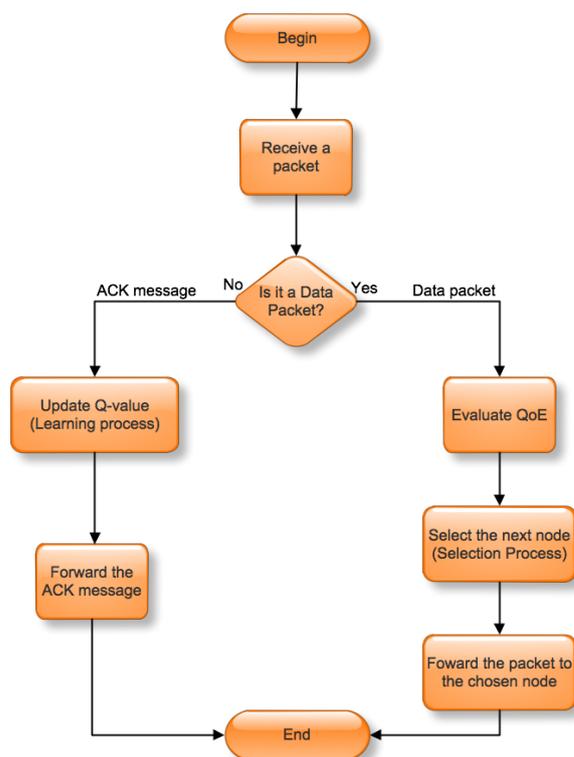


Figure 2.6: QQAR diagram

Based on the native Q-Learning algorithm (Eq 2.30), the update function is expressed as:

$$Q_{\text{new}} = Q_{\text{old}} + \alpha [Q_{\text{new estimation}} - Q_{\text{old}}] \quad (2.31)$$

where  $Q_{\text{new}}$  and  $Q_{\text{old}}$  are the the new and old Q-value, respectively. The  $Q_{\text{new estimation}}$  is the new estimation of Q-value from node  $x$  to the destination, depending on two following factors:

1. The estimation of the link  $xy$ , expressed by the difference between the Q-value of node  $x$  and  $y$ . We call that the *immediate reward*. The reason of choosing this estimation is explained below.
2. The optimal estimation from node  $y$  to the destination, expressed by the maximal value among the Q-values in the Q-table of node  $y$ . We call that the *future reward*.

---

received after being transmitted. MOS is expressed in a number, from 1 to 5, 1 being the worst and 5 the best. The MOS is generated by averaging the results of a set of standard, subjective tests where a number of users rate the service quality

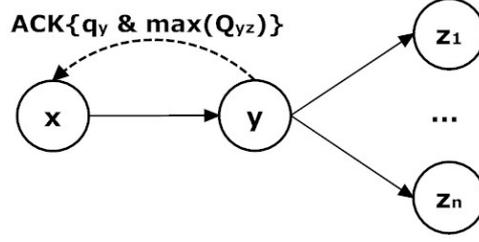


Figure 2.7: Learning process

So, we have the Q-value of the new estimation expressed in equation 2.32:

$$Q_{\text{new estimation}} = \underbrace{\beta(q_y - q_x)}_{\text{immediate reward}} + \underbrace{\gamma \max_i Q_{yz_i}}_{\text{future reward}} \quad (2.32)$$

where  $\beta$  and  $\gamma$  are two discount factors balancing the values between *future reward* and *immediate reward*.

From two equations 2.31 and 2.32, router  $x$  updates the Q-value of the link  $xy$  in using the update function defined in equation 2.33:

$$\underbrace{Q_{xy}}_{\text{new value}} = \underbrace{Q_{xy}}_{\text{old value}} + \alpha \left[ \underbrace{\beta(q_y - q_x) + \gamma \max_i Q_{yz_i}}_{\text{new estimation}} - \underbrace{Q_{xy}}_{\text{old value}} \right] \quad (2.33)$$

Where  $Q_{xy}$  and  $Q_{yz_i}$  are Q-values of links  $xy$  and  $yz_i$ .  $q_x$  and  $q_y$  are QoE value evaluated (MOS score) by using QoE measurement method (presented in appendix A) at node  $x$  and  $y$ .  $\alpha$  is the learning rate, which models the rate updating Q-value. Actually, it is worthwhile noting here the close similarity between (Eq. 2.33) and (Eq. 2.28) given in the global cost function. Parameters  $\alpha$  and  $Q$  represent the gradient step  $\gamma$  and the function  $J$  in the formal model respectively (Eq. 2.28). In fact, QQAR becomes reactive with the update function in Eq. 2.33. The router will reduce the Q-value of a link when its quality is degraded. This reduction helps QQAR avoiding actively bad quality links.

---

**Algorithm 2.3.3.1** Learning algorithm

---

**Input:**  $[Q_0, Q_1, Q_2, \dots, Q_m]$ : Q-values in Q-table.

$m$ : The number of neighbor nodes.

$N$ : the number of node of the routing path.

**for**  $i = 0$  to  $N$  **do**

    Receive the ACK message;

    Update the Q-value;

    Find the maximal Q-value and attach it to the ACK message;

    Forward the ACK message;

**end for**

---

According to the time complexity of the learning algorithm (Algo. 2.3.3.1), the learning process is executed  $N$  times with  $N$  the number of node throughout the routing path. In addition, the number of operations for finding maximal Q-value in Q-table is  $m$ , which is the number of neighbors of this node. As shown in algo. 2.3.3.1, the number of elementary operations performed by learning algorithm for a loop is  $(m + 3)$ . In fact, there are totally  $N$  loops. So, the asymptotic time complexity of learning algorithm is  $O(N \times m)$ .

### 2.3.4 Simple use case-based example of QQAR

We show now some samples of execution of QQAR. We consider a simple network topology as shown in figure 2.8. This network consist of 11 routers (named  $R1, R2, \dots, R11$ ), 1 source node, 1 destination node (the chosen server in a CDN) and 14 connection links. With this sample network, we have 3 paths from source to destination:  $R1-R2-R3-R4-R5-R6$ ,  $R1-R7-R8-R9$  and  $R1-R11$ . The path  $R1-R7-R8-R10$  achieves never the destination. All routers are implemented the QQAR algorithm. We fixed the three parameters  $\alpha = \beta = \gamma = 0.9$ . In order to show easily the execution of exploration phase, we have chosen this high value for these parameters to make the learning changed rapidly. Note that the numbers appear on the links in these figures are their Q-values.

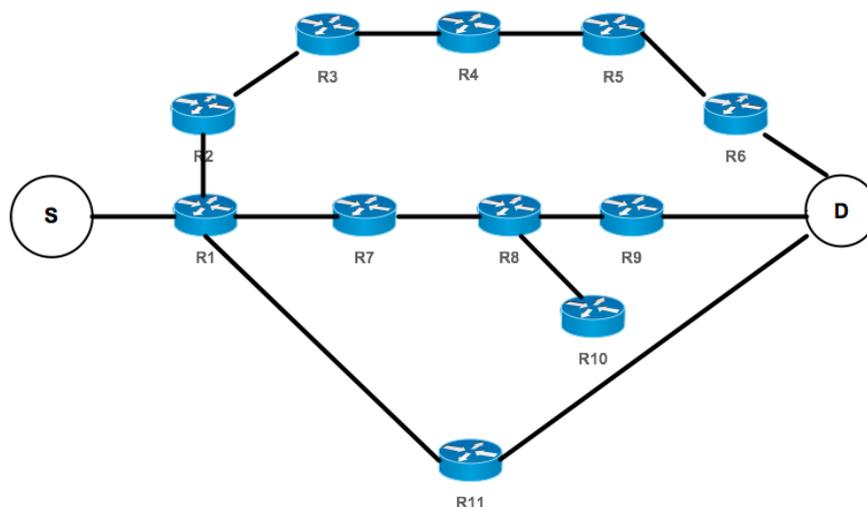


Figure 2.8: Sample of execution for QQAR

In the first phase (Fig. 2.9), assume that the path  $R1-R2-R3-R4-R5-R6$  is chosen and the destination gives a MOS score of 4. We can see that after the update process, all the links of the routing path receive high Q-value.

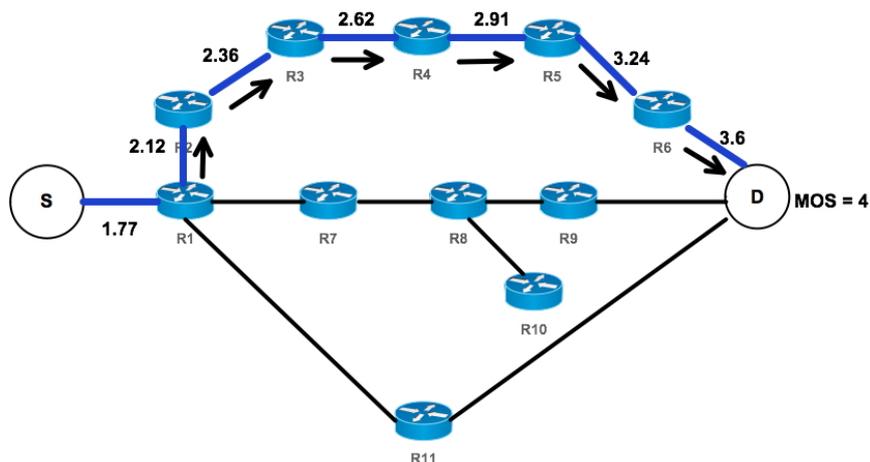


Figure 2.9: Phase 1: with  $MOS = 4$ , the Q-values are updated

In the second phase (Fig. 2.10), the path  $R1-R7-R8-R9$  is chosen. The destination has given a MOS score of 3.

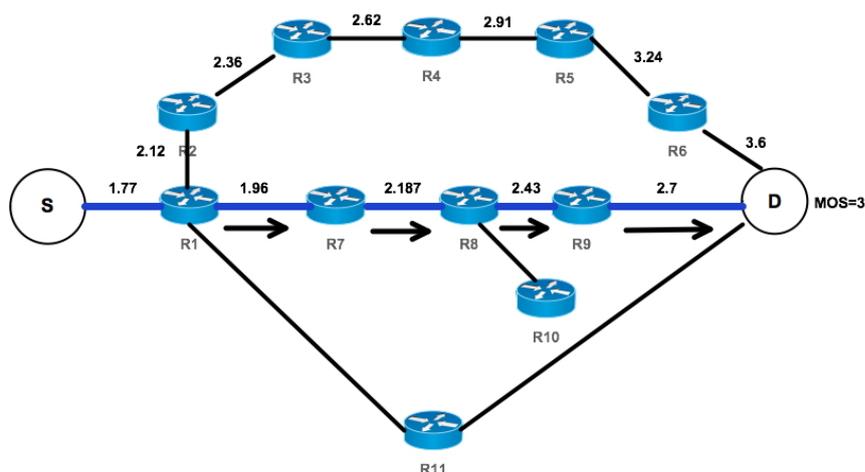


Figure 2.10: Phase 2: with  $MOS = 3$ , the Q-values are updated

The path  $R1-R11$  has been chosen in third phase (Fig. 2.11). The destination has given a MOS score of 1. The Q-values in this routing path are updated in using the equation 2.33. This phase represents the case where a short routing path in terms of hop-count will have low QoE value if the end-user is not satisfied with the received quality.

In 4th phase, the path  $R1-R7-R8-R10$  is chosen (Fig. 2.12). The packet cannot reach the destination, so the QoE value obtained is 0. We can see that with our update

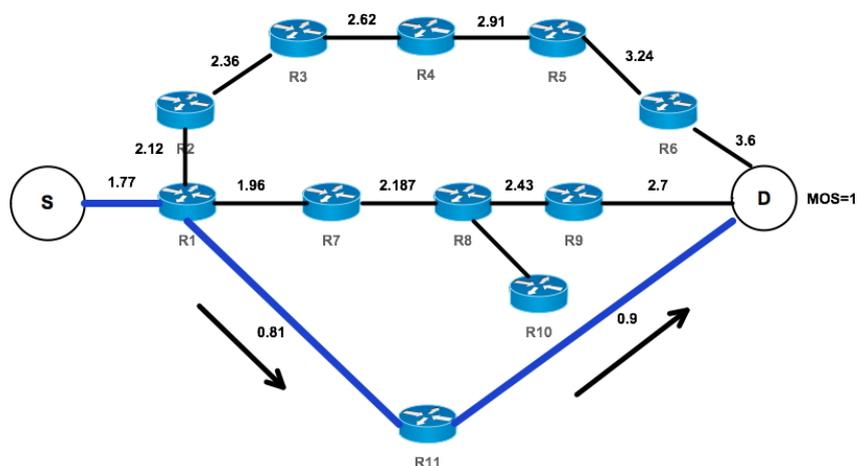


Figure 2.11: Phase 3: with  $MOS = 1$ , the Q-values are updated

function of equation 2.33 does not influence the Q-value in the paths  $R1-R7$  and  $R7-R8$ . Therefore, the router  $R8$  will never choose the path  $R8-R10$  to forward the packet.

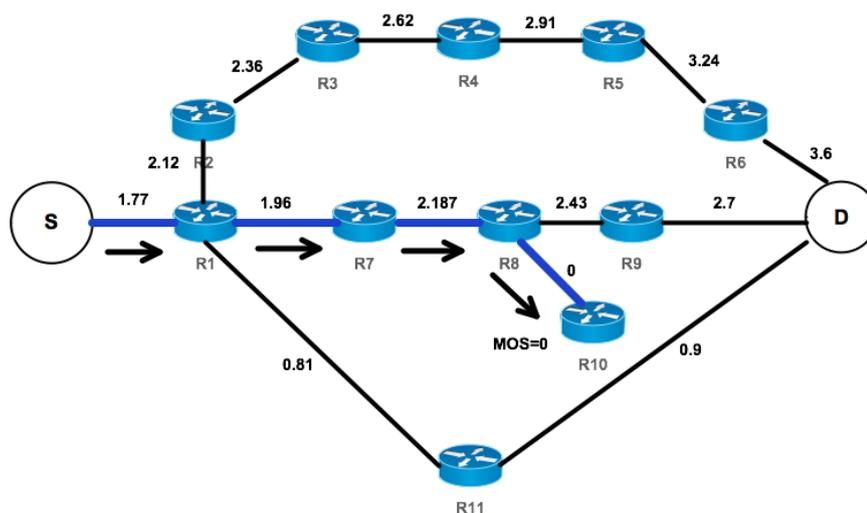


Figure 2.12: Phase 4: with  $MOS = 0$ , the Q-values are updated

In the 5th phase, the path  $R1-R2-R3-R4-R5-R6$  is chosen for the second time (Fig. 2.13). We assume that the path quality is degraded compared with the first time. So, the MOS value given is just 2. The update process has decreased all Q-values of this path.

The samples of algorithm execution above show some characteristics of exploration process of QQAR:

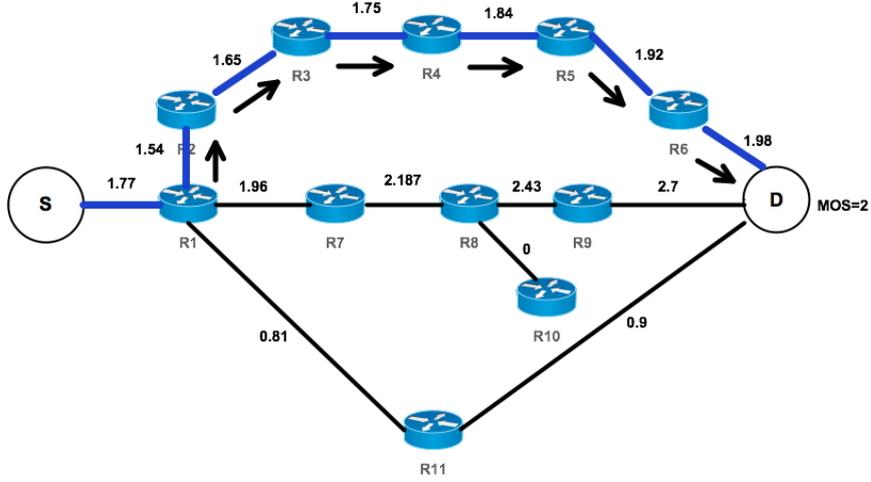


Figure 2.13: Phase 5: with  $MOS = 2$ , the Q-values are updated

- If path  $P1$  obtains a better QoE value (MOS score) than path  $P2$ , all the links of path  $P1$  obtain better Q-values than  $P2$  (phase 1 and 2).
- A path has low Q-values if it obtains a poor quality at end-user even it is close to the destination in terms of hop-count (phase 3).
- If a path cannot reach the destination, this one would not be chosen in the future. (phase 4).
- The decreasing of MOS score at end-user will decrease all Q-values of the corresponding routing path. (phase 5)

In order to explain why the new estimation in equation 2.32 and 2.33 depends on the difference of Q-values between two adjacent nodes,  $(q_y - q_x)$ , we give another sample network in figures 2.14 and 2.15. We keep the value of 0.9 for three parameters  $\alpha$ ,  $\beta$  and  $\gamma$ . This network includes 9 routers, 1 destination and 16 connections links. The router  $R4$  is connected to all other ones. We make router  $R4$  connect to all other routers in order to analyze the Q-routing issue that is solved by our proposed difference in formulation 2.33.

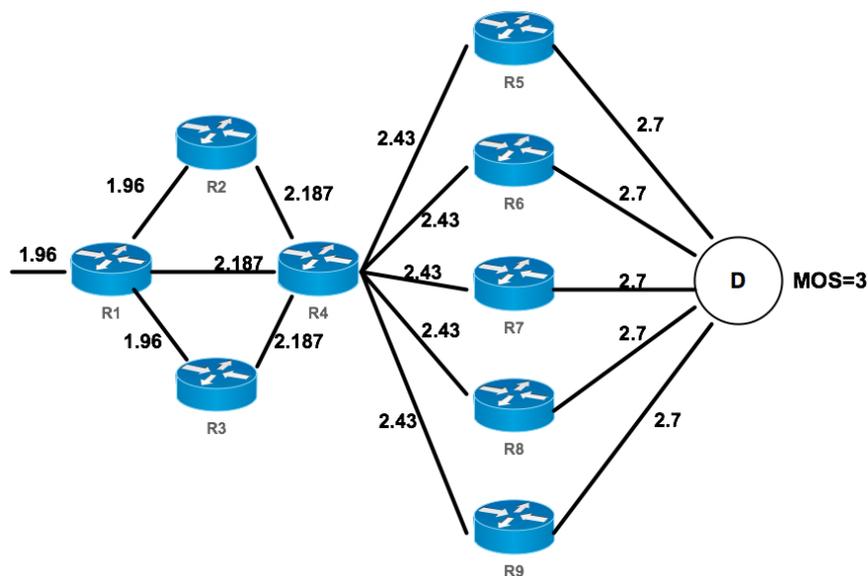


Figure 2.14: Network before the failure

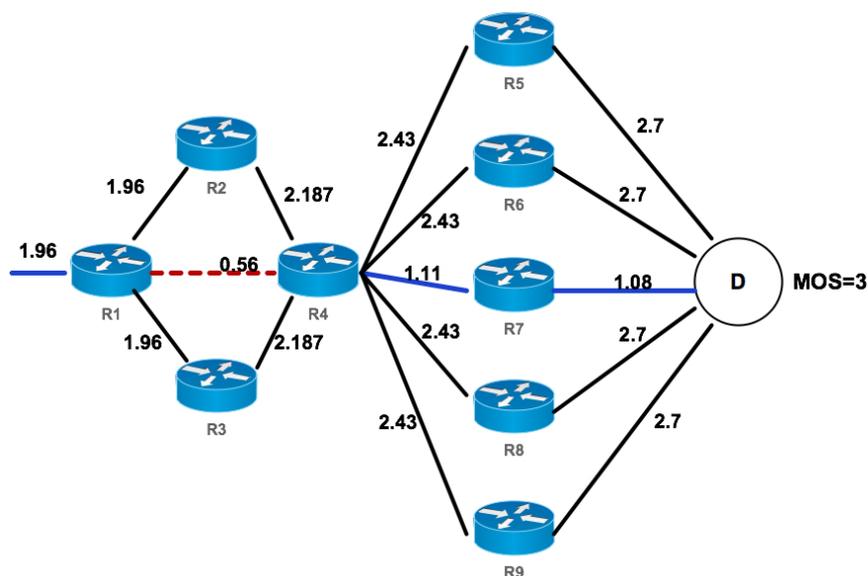


Figure 2.15: QQAR detects exactly the link downed

In fact, due to the difference in the update function, the QQAR is able to detect exactly where the failure comes from. The figure 2.14 describes network state before the failure. With the MOS score of 3 obtained at the destination, all links have their own Q-values assigned by QQAR algorithm. When the failure occurs in the link  $R1-R4$  (figure 2.15) and the routing path is  $R1-R4-R7$ , the MOS score decreases from 3 to 1.

The router  $R7$  and  $R4$  update the Q-values of links  $R7-D$  and  $R4-R7$ , which are equal to 1.08 and 1.11, respectively. We analyze now the update process of router  $R1$ . Without the difference of  $q_{R4}$  and  $q_{R1}$ , the update function becomes:

$$\underbrace{Q_{R1-R4}}_{\text{new value}} = \underbrace{Q_{R1-R4}}_{\text{old value}} + \alpha \left[ \underbrace{\gamma \max_i Q_{R4-Ri}}_{\text{new estimation}} - \underbrace{Q_{R1-R4}}_{\text{old value}} \right] \quad (2.34)$$

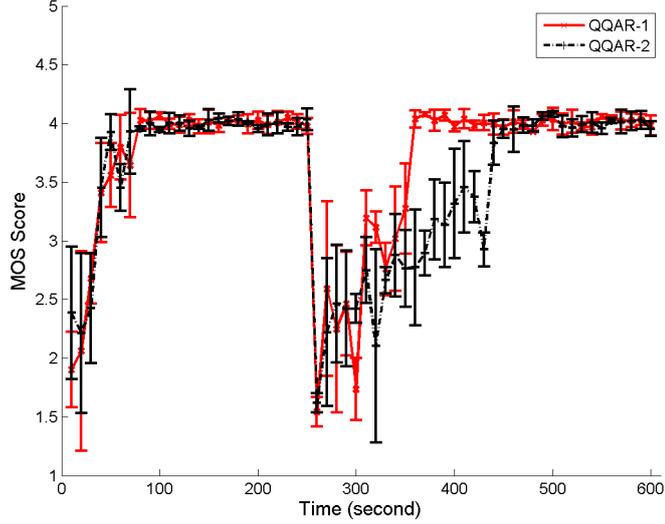


Figure 2.16: Average MOS score of QQAR-1 and QQAR-2

Since the highest value of Q-values of links emerging from  $R4$ ,  $\max_i Q_{R4-Ri}$ , is still 2.43, the Q-value of link  $R1-R4$  is still 2.187. In reality, the problem of quality degradation comes from the link  $R1-R4$ . In order to solve this problem, we added the difference of Q-values between two adjacent routers (equation 2.33). Adding this difference helps us to find the the link that causes the quality degradation problem. In fact, this difference makes the Q-value of link  $R1-R4$  decreased from 2.187 to 0.56 (figure 2.15). With this new low Q-value, the future routing paths of QQAR will avoid this low quality link.

In order to prove this advantage, we have tested two versions of the algorithm QQAR: *QQAR-1* using the formulation 2.33 as an update function and *QQAR-2* using the formulation 2.34 to update their Q-values. The only difference between them is the difference ( $q_y - q_x$ ). We used the irregular network topology 6x6grid (figure 2.17). The test lasts 10 minutes. After 250 seconds, we make link  $R16-R25$  broken down to see how these two versions of QQAR react to this problem.

As shown in figure 2.16, after about 80 seconds of initial period where two algorithms learn the network topology, they obtain an average MOS score of 4.0. At the 250th minutes, the MOS score obtained of two algorithms decreases to 1.5 since the link  $R16-R25$  is broken down. After, while *QQAR-2* takes 220 seconds to get back the MOS

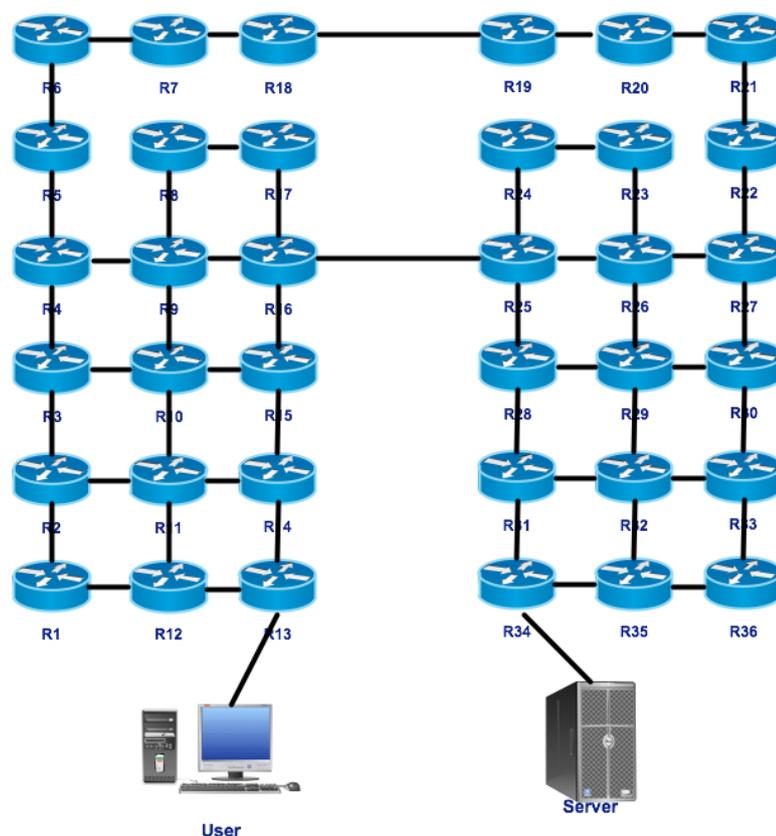


Figure 2.17: Irregular 6x6 grid network

value of 4.5, the *QQAR-1* takes just about 100 seconds. Therefore, in this example, the capacity of QQAR with the difference of Q-value between two adjacent nodes for reacting to the environment changes is better than the version without this difference.

So, with this efficient learning process, all nodes can use their updated Q-values in the Q-table to select the next node for routing the packet received. The selection process is presented in the next subsection.

### 2.3.5 Selection process

With regard to the selection process in each node after receiving a data packet, we have to consider the trade-off between *exploration* and *exploitation*. It cannot always *exploit* the link that has the maximum Q-value because each link must be tried many times to reliably estimate its expected reward. This trade-off is one of the challenges that arises in RL and not in other kinds of learning. To obtain good rewards, a RL agent (router) must prefer actions that it has tried in the past and found to be effective in producing reward. But to discover such actions, it tries actions that it has not selected before. The agent has to *exploit* what it already knows, but it also has to *explore* in order to make

better action selection in the future. There are some mathematical issues to balance exploration and exploitation. We need a method that varies the action probabilities as a graded function of estimated value. The greedy action (i.e. the action that gave the best quality feedback in the past) has still the highest selection probability. All the other actions are ranked and weighted according to their value estimates. Therefore, we have chosen the *softmax method* basing on Boltzmann distribution [Sutton and Barto, 1998].

With this softmax action selection rules, after receiving a packet, node  $x$  chooses a neighbor  $y_k$  among its  $n$  neighbors  $y_i$  ( $i = 1..n$ ) with probability expressed by Equation 2.35 to forward the packet:

$$p_{xy_k} = \frac{e^{\frac{Q_{xy_k}}{\tau}}}{\sum_{i=1}^n e^{\frac{Q_{xy_i}}{\tau}}} \quad (1 \leq k \leq n) \quad (2.35)$$

Where  $Q_{xy_i}$  is Q-value of link  $xy_i$  and  $\tau$  is the temperature parameter of Boltzmann distribution. High temperature makes the link selection to be all equi-probable. On the contrary, low temperatures generates a greater difference in selection probability for links that differ in their Q-values. In other words, the more we reduce the temperature  $\tau$ , the more we exploit the system. Initially, we begin by exploring the system with the initial value of the temperature  $\tau$ . Then, we increasingly exploit the system by decreasing this temperature value. In that way, we reduce  $\tau$  after each forwarding time as shown in Equation 2.36:

$$\tau = \phi \times \tau \quad (0 < \phi < 1) \quad (2.36)$$

where  $\phi$  is the weight parameter.

More precisely, the selection process can be summarized in Algo. 2.3.5.1:

---

**Algorithm 2.3.5.1** Selection algorithm

---

**Input:**  $[Q_0, Q_1, Q_2, \dots, Q_n]$ : Q-values in Q-table.

**repeat**

    Receive the data packet;

    Look the Q-table to determine the Q-value for each link;

    Calculate the probabilities with respect to Q-values (equation 2.35);

    Select link with respect to the probabilities calculated;

    Decrease the temperature value (equation 2.36);

**until** Packet reaches the destination

---

We analyze now the algorithm complexity of the selection algorithm (Algo. 2.3.5.1). There are totally  $N$  loops, where  $N$  is the number of node in the routing path. Calculating the probabilities for each link has a time complexity of  $O(m)$ , where  $m$  is the number of neighbors. As  $N \leq N_{TTL}$ , where  $N_{TTL}$  is *time to live* we fixed before (i.e. the number of hops the packet is allowed to pass before it dies), the worst-case time complexity for the selection algorithm is  $O(N_{TTL} \times m)$ .

## 2.4 Conclusion

In this chapter, we focused on the proposed routing algorithm, called QQAR, that was applied to the routing layer of a CDN. First, we presented generally the Reinforcement Learning theory with mathematical model and value functions. After, we presented the Q-Learning algorithm and explain why this algorithm is suitable for Q-Routing. Our routing system has taken into account the QoE feedback as the reward of a RL model. Regarding this reward, the routing policy is adapted for environment changes. Then, the QQAR is presented in basing on the Q-Routing algorithm in adding a particular improvement. We explained the advantage of our improvement in showing the concrete example and some results. We showed clearly step by step the processes of the algorithm to prove the mentioned advantages. We depicted also how our change of QLearning's formula improved the time convergence of the routing algorithm. Two main processes of QQAR are presented in detail: the learning and selection process. With its advantages, the QQAR is totally suited to be applied to the routing layer of a CDN architecture. The experimental results of this routing algorithm are presented in chapter 4.

For conclusion, this chapter has resolved the routing issue in a CDN system. Since a routing process needs information of address of source-destination pair. The source is the end-user who sends the request. To determine the server to send the request, we need a server selection process that is located in the meta-routing layer. In the next chapter, we will present the proposed server selection algorithm.

## Chapter 3

# Proposed server selection algorithm

This chapter will present the proposed server selection method that is inspired by the Multi-Armed Bandit problem (MAB) for the meta-routing layer of a CDN architecture. In fact, we consider the QoE as the reward in MAB formalization that is able to take into account the exploration-exploitation trade-off. Among algorithms for MAB problem, we chosen the algorithm called UCB1 (Upper Confidence Bound 1).

### 3.1 Introduction

In the previous chapter, we have presented the proposed routing algorithm for the routing layer in a CDN. This chapter focuses on the meta-routing layer in proposing a new server selection method, which is inspired by the multi-armed bandit problem.

The E2E QoE model is efficient and indispensable in every today's network systems. It carries the user experience to guaranty the satisfaction of end-users instead of focusing only on technical parameters as current QoS systems. In fact, as proven previously, satisfying end-user is the new trend of development of Internet as well as emerging high quality applications. However, the E2E QoE model will be abstract if it is not applied to any real network architecture. In this thesis, we decide to apply this model to a CDN architecture, which has been introduced to improve the delivered service quality.

As presented in chapter 1, Content Distribution Network (CDN) [Yin et al., 2010] is considered as the user-centered solution in moving the content from the origin servers to the servers close to end-users. This approach is used to improve the scalability of a network service due to the fact that it is able to provide services to a large number of clients with a reasonable service quality. Actually, the user requests are transferred to replica servers instead of sending to the origin server. That helps the system to avoid congestion as well as overload issues at the core network and server side. In addition, this cache-related approach minimizes the delay time. Theoretically, a CDN architecture has two main layers: the routing layer and the meta-routing layer. The second one is composed of several modules such as *server placement*, *cache organization* and

*server selection.* The two first modules belong to the preparation-related phase. More precisely, the server placement module tries to place the replica servers in an optimized way to minimize the delivery delay and the bandwidth consumption. Providers use the cache organization module to organize the content stored in replica servers in order to guarantee the availability, freshness and reliability of content. Besides these two preparation-related modules, we focus on the server selection module, which plays an important role in launching operation phase of a CDN. The fundamental objective of servers selection is obviously to offer better performance than from the origin server. Another adding value of this selection process is for lowering the costs of network resources. It is not easy to choose an appropriate server to provide service to users. The appropriate server may not be the closest one in terms of hop-count or E2E delay, nor the least loaded server. The best server is the one makes end-user satisfied when using the provided service. So, the server selection process plays a key role in the decisive success of a CDN. We survey now some related researches on server selection methods in the context of CDN. Then, we explain our motivation to develop a server selection scheme based on a multi-armed bandit formalization.

Zhou et al. [Zhou et al., 2008] present a server selection method that is based on load information of replica servers to choose the appropriate server. This method uses a reliable server pooling framework named RSerPool. Unfortunately, this approach is not always feasible because the accessibility of CDN servers is not always available. In fact, server polling or checking with a high frequency increases the server load.

Han et al. [Han et al., 2009] propose a server selection method that uses RTTs (round trip times) between client and servers to choose the server that has the minimum RTT to client. The RTTs is measured by passive-based probing. The drawback of this proposal is that probing may increase traffic overhead of the network.

Chellouche et al. [Chellouche et al., 2010] use anycasting technique to realize server selection process. This mechanism chooses the appropriate server among server set in an anycast group based on network metrics. However, deploying an anycasting system is not easy and common. Even it is applicable to IPv6 infrastructure (RFC 1546), the IPv4 network infrastructure does not support this kind of delivering operation because there is not an address range set aside for anycast use on the public IPv4 Internet (presented in IETF RFC 3330).

Wendell et al. [Wendell et al., 2010] propose a distributed system, named DONAR, that uses an algorithm to consider both client performance (e.g. sudden changes in client request rates) and server load. This system deploys mapping nodes for server selection. This algorithm can maintain a weighted split of requests to a client's replicas. This approach uses a weighted function for decision making. So, choosing the appropriate weighted vector is a problem of this approach because one cannot obtain an optimized solution.

In several projects such as the RingServer Project in Japan [Ringserver, 2009], the Round-Robin mechanism has also been used for server selection policy. However, since each server is selected by a classic Round Robin algorithm, the network criteria and server load are both ignored. This simple policy is not applicable in a dynamic network

system like CDN.

Guo et al. [Guo et al., 2002] design and evaluate server selection techniques with a fixed number of channels for replicated video-on-demand servers. The authors propose several heuristic techniques, to outperform a basic policy, that always route requests to the closest server. For the same purpose, G. Pierre et al. [Pierre and Van Steen, 2006] propose Globule. The closest server in terms of network proximity is always chosen. However, the closest server is not always the server that gives the best quality. Similarly to above works, Almeida et al. [Almeida et al., 2004] try to minimize the objective function of a weighted sum of network and server bandwidth usage to resolve the server selection problem.

In this point of our work, a fundamental point must be made, all of the approaches presented above consider only the QoS criteria such as network and server parameters as the measure of performance to optimize but not the QoE that represents the actual end-user perception. The assumption made behind this decision is that a good QoS measurement strictly induces a good QoE performance. However, this hypothesis is not always correct because the data is delivered to end-user through the network in different contexts. So, the subjective end-user context has to be considered carefully, or at least not neglected. In addition, these approaches do not take into account the exploitation-exploration trade-off (i.e. finding a balance between uncharted knowledge and current knowledge) that must be taken into account in any online decision making task such as in a CDN environment. Consequently, these above problems motivate us to propose a novel server selection algorithm, which takes into consideration the end-user's perception as well as the exploitation-exploration trade-off. To acquire this purpose, we have to use a selection algorithm based on a learning method.

Taking an example of a learning problem, one has to choose repeatedly among  $n$  different options (*actions*). After each choice, one receives a numerical reward of a stationary probability distribution dependent on the chosen action. The goal is to maximize the expected total reward over some time period. This is the form of the Multi-Armed Bandit problem that we based on to formalize the server selection issue in a CDN.

This chapter presents the Multi-Armed Bandit formalization and explains why we chose it to formalize our selection algorithm. Then, we focus on the proposed server selection algorithm.

## 3.2 Multi-armed Bandit formalization

Multi-armed bandit (MAB) [Vermorel and Mohri, 2005, Antos et al., 2008] takes its name from a traditional slot machine (one-armed bandit). With the motivating applications in statistics, multiple levers of this machine are considered. After pulling a lever, the latter provides a reward drawn from a distribution associated with that specific lever. The goal of the gambler is to maximize his total rewards earned in a series of trials. Indeed, many learning methods and optimizing problems are formalized in this way.

In this section, we present briefly the notion and general formulation of the MAB and discuss also some common policies for MAB problem.

### 3.2.1 MAB paradigm

We chose the MAB formalization for our proposal due to the following reasons:

- MAB formalization does not make any assumption concerning the reward function. This approach always carries the exploration and exploitation trade-off dilemma for any reward function. The reward function we take into account is the *MOS* score, the evaluated QoE value.
- Unlike other approaches of QoS optimization problems, MAB makes no heuristic. While other classic approaches try to optimize a concrete heuristic such as E2E delay, hop count, etc., MAB explicitly minimizes the regret  $\rho$  (equation 3.2).
- Among the approaches capable of solving the MDP, there is only the MAB approach that needs only one state. The future action is chosen based on the obtained reward and this unique state. This property is suitable for the server selection issue we are working on because we just know the current state of each server.

In MAB problem, the expected reward given by the selected action is called the *value* of this action. The MAB would be easily solved if one knew the value of each action, one would always choose the action with highest value. In reality, one does not know the action values, although one may have estimates. In fact, among actions with different action values, there is at any time at least one action whose estimated value is greatest. This action is called a *greedy action*. Selecting the greedy action means *exploiting* the current knowledge of the value of the actions. On the other hand, selecting one of the non-greedy actions means *exploring* because this enables us to improve the estimate of the non-greedy action's value. To compare exploitation and exploration, while exploitation is used to maximize expected reward on the one choice, exploration is used to produce greater total reward in the long run. For example, we know certainly the greedy action's value, while other actions are evaluated to be nearly as good but with substantial uncertainty. However, at certain point, the uncertainty means that one of other actions becomes probably better than the greedy action, but we do not know which. It is clearly better to explore the non-greedy actions and determine which of them are better than the greedy one. Reward may be lower in the short run, during exploration phase, but higher in the long run after you have determined the better actions to exploit them. Since we cannot both explore and exploit in a single selection action, we called that *conflict between exploration and exploitation*.

There are some algorithms for evaluating the value of actions and for using the estimates to select the action. One simple way to estimate this is by averaging the rewards received if the action was selected. So, after  $t$  decisions, the action value of action  $a$  is estimated after  $n_a$  times as follows:

$$Q_t(a) = \frac{r_1 + r_2 + \dots + r_{n_a}}{n_a} \quad (3.1)$$

Where  $Q_t(a)$  is the estimated value after  $t$  choices. As  $n_a \rightarrow \infty$ ,  $Q_t(a)$  converges to  $Q^*(a)$ .

The method above, called *sample-average*, is one way to evaluate action-values, not the best one. For example, the algorithm UCB1 that we used in our approach has the estimate function, called UCB index, of  $\hat{r}_a + \sqrt{\frac{2 \ln(t)}{n_a}}$  where the  $\hat{r}_a$  is the estimate for the true expected rewards.

In many real-world contexts, one makes decisions to maximize the expected numerical reward. The chosen actions are used not only to bring in more reward but also to discover new knowledge that could help to improve future decisions. Such situations include adaptive server selection efforts for improving the service quality in a CDN system. All decision-making systems face a particular problem of balancing reward maximization based on the already acquired knowledge and attempting new actions to further increase knowledge. That is known as the exploitation-exploration trade-off in reinforcement learning.

The multi-armed bandit has been introduced by Robbins [Robbins, 1952] to solve this general problem. The main difference with a traditional slot machine (one-armed bandit) is that multi-armed bandit has more than one lever. Pulling a lever provides a reward drawn from a distribution associated with that specific lever. In the beginning, the gambler has no knowledge about the levers, but after some repeated trials, he can determine the most rewarding levers. In fact, the multi-armed bandit problem is actually determining the best policy for the gambler.

The MAB problem is a formalization of a sequential decision making task composed by a decision-maker and fixed decision actions  $a_i$  with  $i = 1..N$  ( $N \geq 1$ ). Each time the decision-maker chooses an action  $a_i^t$  at time  $t$ , it receives a reward  $r_{a_i^t}^t$  drawn from an unknown distribution associated with that specific action. The goal of the decision-maker is to maximize the sum of rewards received through a sequence of choosing actions. So in each period,  $t = 1, 2, \dots, T$ :

- The decision-maker chooses  $a^t$ , based on its previous rewards and chosen actions  $a^1, r_{a^1}^1, a^2, r_{a^2}^2, \dots, a^{t-1}, r_{a^{t-1}}^{t-1}$ .
- The decision-maker receives only the reward  $r_{a^t}^t$ .

The MAB problem can be modeled by a set of real distributions  $\Psi = \{\Phi_1, \Phi_2, \dots, \Phi_N\}$  called arms. Each distribution  $\Phi_i$  is associated with the rewards delivered by action  $a_i$ . The average values associated with these reward distributions are  $\xi_1, \xi_2, \dots, \xi_N$ .

The regret  $\rho$  after  $T$  rounds is defined as the difference between the reward sum associated with an optimal strategy and the sum of the collected rewards:

$$\rho = T\xi^* - \sum_{t=1}^{t=T} r_t \tag{3.2}$$

Where  $\xi^*$  is the maximal reward average,  $\xi^* = \max(\xi_i)$ , and  $r_t$  is the reward at time  $t$ .

All of policies or algorithms proposed for a MAB problem tend to minimize the regret per round. A strategy is called *zero-regret policy* if his average regret per round tends to zero with probability of 1 for any bandit problem when the horizon tends to infinity.

With enough played round, the zero-regret policies are guaranteed to converge to an optimal policy. This formalism is a common description of a discrete action based on reinforcement learning problem [Sutton and Barto, 1998] with a unique state.

There are many extensions to the problem of stochastic multi-armed bandit:

- **Bandit in MDP** [Jaksch et al., 2010]: Exploration Strategy “optimistic uncertainty” (based on USB) to explore a Markov Decision Process. In [Auer et al., 2009], the authors proposed algorithm, called UCRL, that achieves logarithmic expected regret:  $\mathbb{E}[R_T^A] = O(\log T)$ . Fix some time  $t$ . UCRL keeps an estimate of the transition probabilities ( $p_t : X \times A \times X \rightarrow [0, 1]$ ) and the immediate rewards ( $r_t : X \times A \rightarrow [0, 1]$ ). Furthermore, UCRL constructs confidence bounds,  $c_t^{(p)} : X \times A \times X \rightarrow \mathbb{R}^+$ ,  $c_t^{(r)} : X \times A \rightarrow \mathbb{R}^+$ , for these estimates:  $c_t^{(p)}(x, a, y)$  is a confidence bound for the transition probability estimate  $p_t(x, a, y)$ , and  $c_t^{(r)}(x, a)$  is a confidence bound for the reward estimate  $r_t(x, a)$ . Together these specify the set of plausible transition probabilities and rewards given the learner’s experience up to time  $t$ : In particular,  $p$  is a plausible transition probability function if it is transition probability function and  $|p(x, a, y) - p_t(x, a, y)| \leq c_t^{(p)}(x, a, y)$  holds for all  $(x, a, y) \in X \times A \times X$ . Similarly,  $r$  is plausible if  $r : X \times A \rightarrow [0, 1]$  and  $|r_t(x, a) - r(x, a)| \leq c_t^{(r)}(x, a)$  holds for all  $(x, a) \in X \times A$ . Denote the set of plausible MDPs at time  $t$  by  $M_t$ . The “optimistically best policy at time  $t$ ” is  $\pi_t = \operatorname{argmax}_\pi \{\rho^\pi(M) : M \in M_t\}$ .
- **Bandits with contextual information:** At each time  $t$ , there is information  $x_t \in X$  and one makes a decision  $a_t \in A$ . The reward is a function of  $a_t$  and  $x_t$ . It compares to a class of strategies  $\pi : X \rightarrow A$ .
- **Bandit with a countable number of arms** [Wang et al., 2008]: Each new arm has a probability  $\epsilon^\beta$  to be  $\epsilon$ -optimal. In [Wang et al., 2008], the authors focus on the case when the number of arms is infinite (or larger than the available number of experiments). In this case, the exploration of all the arms is impossible to achieve: if no additional assumption is made, it is not easy to find a near-optimal arm. They consider a stochastic assumption on the average-reward of any new selected arm. After pulling a new arm  $k$ , its average-reward  $\mu_k$  is assumed to be an independent sample of a fixed distribution. They write  $v_n = O(u_n)$  when for some  $n_0, C > 0$ ,  $v_n \leq C u_n (\log(u_n))^2$ , for all  $n \geq n_0$ . They assume that the rewards of the arms lie in  $[0, 1]$ . The regret bounds depend on whether  $\mu^* = 1$  or  $\mu^* < 1$ . For  $\mu^* = 1$ , the algorithms are such that  $\mathbb{E}R_n = O(n^{\beta/(1+\beta)})$ . For  $\mu^* < 1$ , one has  $\mathbb{E}R_n = O(n^{\beta/(1+\beta)})$  if  $\beta > 1$ , and  $\mathbb{E}R_n = O(n^{1/2})$  if  $\beta < 1$ . In addition, they derive the lower-bound: for any  $\beta > 0$ ,  $\mu^* \leq 1$ , any algorithm satisfies  $\mathbb{E}R_n \geq C n^{\beta/(1+\beta)}$  for some  $C > 0$ . The authors propose an algorithm having the anytime property, which is based on an arm-increasing rule. The proposed algorithms essentially consist in pulling  $K$  different arms randomly chosen, where  $K$  is of order  $n^{\beta/2}$  if  $\mu^* < 1$  and  $\beta < 1$ , and  $n^{\beta/(1+\beta)}$  otherwise, and using a variant of the UCB algorithm on this set of  $K$  arms, which takes into account the

empirical variance of the rewards. This last point is crucial to get the proposed rate for  $\mu^* = 1$  and  $\beta < 1$ .

- **Linear arms** [Mersereau et al., 2009]: One chooses an arm  $x_t \in X \subset \mathbb{R}^d$ . The average reward is a linear function  $r_t = x_t \cdot \alpha$ , where  $\alpha \in \mathbb{R}^d$  is an unknown parameter. In [Mersereau et al., 2009], the authors propose a simple model that demonstrates the benefits of exploiting the underlying structure of the rewards. They consider a bandit problem where the expected reward of each arm is a linear function of an unknown scalar, with a known prior distribution. Since the reward of each arm depends on a single random variable, the average rewards are perfectly correlated. They prove that, under certain assumptions, the cumulative Bayes risk over  $T$  periods under a greedy policy admits an  $O(\log T)$  upper bound, independent of the number of arms. This approach is extended in [Rusmevichientong and Tsitsiklis, 2010] to the setting where the expected reward of each arm depends linearly on a multivariate random vector  $Z \in \mathbb{R}^r$ . The authors in [Rusmevichientong and Tsitsiklis, 2010] concentrate on the case where  $r \geq 2$ , which is fundamentally different from the model in [Mersereau et al., 2009] due to the fact that the average rewards now depend on more than one random variable, and thus, they are no longer perfectly correlated. The bounds on the regret and Bayes risk and the policies found in [Mersereau et al., 2009] no longer apply.
- **Bandits in metric spaces** [Kleinberg et al., 2008] [Bubeck et al., 2008]: One chooses an arm  $x_t \in X$  in metric space. The average reward  $f(x_t)$  is assumed Lipschitz. It compares to  $\sup_{x \in X} f(x)$ . This is an optimization problem online. In [Kleinberg et al., 2008], the authors considered generic metric spaces assuming that the mean-payoff function is Lipschitz with respect to the (known) metric of the space. They proposed an algorithm that achieves essentially the best possible regret with respect to these environments.
- **Bandits hierarchical**: Algorithms UCT [Kocsis and Szepesvári, 2006], BAST [Coquelin and Munos, 2007], HOO [Bubeck et al., 2008], application to the game of go [Gelly et al., 2006].

Nowadays, there are lots of applications of MAB in different fields. The next subsection surveys briefly some of them.

### 3.2.2 Applications of MAB

Multi-armed bandit problems play a key role for decision problems to maintain the “exploration versus exploitation” trade-off. The latter is considered as the gambling scenario described previously. After the gambler has known a slot machine having good average payoff, there are two options to choose: continuing to play this slot machine (exploitation) and trying other alternatives that have never been tested or that have only been tested infrequently (exploration). Since this type of trade-off appears widely in the field of online decision problems, there are many applications of multi-armed bandit algorithms. We cite here some usually used applications as follows:

**Web search** [Radlinski et al., 2008]: For Internet infrastructure, Web search has become an essential component and has attracted significant interest from the machine learning community. The conventional approach is associated to the learning-to-rank problem. In [Radlinski et al., 2008], an online learning approach for learning from usage data is presented. Since training data is being collected, it immediately impacts the rankings shown. The learning problem they address is regret minimization, where the goal is to minimize the total number of poor rankings displayed over all time. In particular, in this setting there is a natural trade-off between exploration and exploitation. The authors propose a learning algorithm to compute an ordering of search results from a diverse set of orderings. By iterating through all documents in each of the positions while holding fixed the documents in the other positions, they attempt to learn a “best” ranking of documents using user clicks. Their approach naturally produces a diverse set of results, as user feedback through clicks will diminish the value of similar documents.

**Design of ethical clinical trials** [Hardwick and Stout, 1992]: In this application, one has to evaluate  $n$  possible treatments for a disease in order to find which one is most effective. The appropriate solution is to design a clinical trial in assigning  $T$  subjects partitioned randomly into  $n$  groups of equal size. Each treatment is administered to one group of subjects, and the results across different groups are compared to see which treatment is most effective.

The experiment is launched over time so that some of the subjects arrive after the treatments have already been tested on earlier subjects. The rule is adjusted in assigning subjects into groups as time progresses. After some stages of the trial, a greater fraction of the subjects should be assigned to treatments which have performed well during the earlier stages of the trial. This increases the expected benefit of the treatments to the subjects in the clinical trial, and it also has the effect of increasing the amount of data collected for the most successful treatments. Because a subject’s response to a treatment can always be evaluated before the next subject arrives, this problem becomes a multi-armed bandit problem where the set of strategies is the set of treatments. The time periods is the sequence of subjects participating in the trial.

**Internet ad placement** [Gabillon, 2009]: In the case where a customer is launching an Internet advertising campaign for a new product. He has bought advertising space on a website, one of  $n$  possible advertisements should be chosen to be displayed each time a user visits the site. If the user clicks on the advertisement, the payoff of displaying an advertisement to a user is 1, 0 otherwise. In any case, this is a MAB problem in which the set of strategies is the set of advertisement that user might display.

**Pricing:** A seller uses a website to sell something with zero production cost. When a user visits the site, the seller offers a price and the user makes a purchase if and only if this price is less than or equal to the maximum amount the user is willing

to pay. This is also a MAB problem in which the set of strategies is the set of possible prices. The payoff of charging price is 0 if the user is unwilling to pay that price, or 1 if the user is willing to pay the price.

**Overlay routing** [Awerbuch and Kleinberg, 2008]: In an overlay network, two hosts communicate in using a multi-hop path. The network experiences point-to-point delays which change unpredictably over time. Two hosts can only measure the end-to-end delay on a multi-hop path. The goal is to choose a routing path having small delay time. That can be modeled as a MAB problem in which the set of strategies is the set of multi-hop paths. The cost of a path is equal to its end-to-end delay.

Since this thesis focuses on CDN service, the application we chose to apply the MAB is server selection method in a CDN system. In fact, there are many policies to apply the MAB to the applications above. The next subsection surveys some widely used algorithms for the MAB problem. Then, we compare these algorithms to choose the best one, in terms of QoE, to apply to our CDN architecture.

### 3.2.3 Algorithms for MAB

MAB formalization of a sequential decision problem has to take into account the exploration versus exploitation dilemma, i.e. the balance between exploring the less chosen actions in order to find the most profitable actions while exploiting the currently identified best action as often as possible. Many algorithms provide policies to solve a MAB formalization of a given sequential decision problem. The measure that assesses the success of a policy is *the regret*. Lai et al. [Lai and Robbins, 1985] indicate that the regret for MAB formalization has to grow at least logarithmically in the number of plays. In this section, we present three currently used algorithms for MAB formalization:  *$\epsilon$ -greedy*, *softmax* and *upper confidence bounds*.

#### 3.2.3.1 $\epsilon$ -greedy algorithm

First, the  $\epsilon$ -greedy rule [Watkins, 1989] [Koulouriotis and Xanthopoulos, 2008] (Alg. 3.2.3.1) is the simplest and well-known policy for the bandit problem.  $\epsilon$ -greedy policy includes choosing randomly a lever with  $\epsilon$ -frequency, and otherwise choosing the lever having the highest estimated reward. More precisely, for this algorithm, the decision-maker chooses with a probability of  $(1 - \epsilon)$  the action with the highest average reward, and with a probability  $\epsilon$  a random action uniformly drawn in the set of available ones.

The simplest variant of the  $\epsilon$ -greedy policy is called  *$\epsilon$ -first policy*. The  $\epsilon$ -first policy consists of realizing the exploration at the beginning. More precisely, for a number  $T \in \mathbb{N}$  of rounds, one has a pure exploration phase where the actions are randomly chosen during  $\epsilon T$  first rounds. In the remaining  $(1 - \epsilon)T$  rounds, the action having highest estimated average reward is chosen. One calls it the pure exploitation phase. The  $\epsilon$  must be in the interval  $(0, 1)$  and it is chosen by user. The PAC framework of [Even-Dar et al., 2002]

and [Mannor and Tsitsiklis, 2004] have presented the  $\epsilon$ -first policy. There is another  $\epsilon$ -greedy policy called  $\epsilon$ -decreasing policy, which uses the decreasing  $\epsilon$  for getting arbitrarily close to the optimal policy asymptotically. Like the general  $\epsilon$ -greedy policy, a random action is chosen with  $\epsilon$  frequency and the action having the highest average reward is chosen with the  $(1 - \epsilon)$  frequency. The difference is the  $\epsilon_t$  where  $t$  is the index of the current round. The decreasing  $\epsilon_t$  is given by  $\epsilon_t = \min\{1, \frac{\epsilon_0}{t}\}$  where  $\epsilon_0 > 0$ . The value of  $\epsilon_0$  is chosen by the user. Cesa-Bianchi et al. [Cesa-Bianchi and Fischer, 1998] use an algorithm called *GreedyMix* to analyze the  $\epsilon$ -decreasing policy. The slight difference between the *GreedyMix* and  $\epsilon$ -decreasing policy is the use of a decreasing factor of  $\frac{\log(t)}{t}$  instead of  $\frac{1}{t}$ . The authors prove a  $\mathcal{O}(\log(T)^2)$  regret for *GreedyMix* where  $T$  is the number of rounds. Auer et al. [Auer et al., 2002] improve this result and they achieve a  $\mathcal{O}(\log(T))$  regret for the  $\epsilon$ -decreasing policy with some constraint of the choice of the value  $\epsilon$ .

---

**Algorithm 3.2.3.1**  $\epsilon$ -Greedy algorithm

---

**Input:**  $[a_0, r_0, a_1, r_1, \dots, a_{t-1}, r_{t-1}]$ ,  $N \geq 1$ ,  $0 < \epsilon < 1$

**Output:**  $a_t$

Play each action once;

**for**  $i = 0$  to  $N$  **do**

$\hat{r}_{a^i} \leftarrow$  current average reward of  $a^i$

**end for**

**if** probability is  $1 - \epsilon$  **then**

**return**  $a_t^i = \operatorname{argmax}_i(\hat{r}_{a^i})$

**else**

**return**  $a_t^i$  is selected randomly

**end if**

---

According to [Auer et al., 2002], the  $\epsilon_t$  is defined by

$$\epsilon_t = \min \left\{ 1, \frac{cK}{d^2 t} \right\} \quad (3.3)$$

where  $c > 0$  and  $0 < d < 1$  are parameters. For all  $K > 1$  and for all reward distributions  $P_1, P_2, \dots, P_K$  with support in  $[0, 1]$ , if policy  $\epsilon$ -greedy is run with input parameter:  $0 < d \leq \min_{i: \mu_i < \mu_*} \Delta_i$ , then the probability that after any number  $n \geq cK/d$  of plays  $\epsilon_t$ -greedy chooses a sub-optimal machine  $j$  is at most:

$$\frac{c}{d^2 n} + 2 \left( \frac{c}{d^2} \ln \frac{(n-1)d^2 e^{1/2}}{cK} \right) \left( \frac{cK}{(n-1)d^2 e^{1/2}} \right)^{c/(5d^2)} + \frac{4e}{d^2} \left( \frac{cK}{(n-1)d^2 e^{1/2}} \right)^{c/2} \quad (3.4)$$

**3.2.3.2 Softmax algorithm**

Secondly, the Softmax policy [Luce, 1959, Koulouriotis and Xanthopoulos, 2008] uses a function to assign average reward into action probabilities. The used function  $p^i$  is

demonstrated in Alg. 3.2.3.2. The decision-marker chooses an action based on probability  $p^i$  assigned to each action  $a^i$ . This algorithm allows to explore less used actions at a given time but try also to focus on best action in terms of gains expectation with respect to a Gibbs sampling distribution [Georgii, 2011].

The action  $k$  is chosen with the probability:

$$p^i = \frac{e^{\frac{\hat{r}_{a^i}}{\tau}}}{\sum_{j=1}^N e^{\frac{\hat{r}_{a^j}}{\tau}}} \quad (3.5)$$

where  $\hat{r}$  is the estimated mean of the rewards brought by the action  $i$  and  $\tau \in \mathbb{R}^+$  is a parameter called *temperature*. The user chooses the value of  $\tau$ .

Like  $\epsilon$ -greedy policy, the Softmax policy (Boltzman Exploration) can be modified into *decreasing Softmax* whose temperature  $\tau$  decreases with the number of round. More precisely, the decreasing Softmax has a temperature  $\tau_t = \frac{\tau_0}{t}$  that depends on the index  $t$  of the current round. The user has to choose the value of  $\tau_0$ . Bianchi et al. [Cesa-Bianchi and Fischer, 1998] use *Softmix algorithm* to analyze the decreasing Softmax. The Softmix differs from the decreasing Softmax with a  $\frac{\log(t)}{t}$  factor instead of a  $\frac{1}{t}$  factor.

There is another variant of the Softmax algorithm: the Exp3 “exponential weight algorithm for exploration and exploitation” [Auer et al., 1995]. At the round  $t$ , the probability of choosing the action  $k$  is defined as follows:

$$p_k(t) = (1 - \gamma) \frac{\omega_k(t)}{\sum_{j=1}^K \omega_j(t)} + \frac{\gamma}{K} \quad (3.6)$$

where  $\omega_j(t+1) = \omega_j(t) \exp\left(\gamma \frac{r_j(t)}{p_j(t)K}\right)$  if the action  $j$  has been chosen at time  $t$  with  $r_j(t)$  being the observed reward,  $\omega_j(t+1) = \omega_j(t)$  otherwise. The user chooses the parameter  $\gamma \in (0, 1]$ . The algorithm divides the reward  $r_j(t)$  by the probability  $p_j(t)$  that the action was chosen. In a modified version of *Exp3* [Auer et al., 2003] with decreasing  $\gamma$ . This approach achieved a regret of  $\mathcal{O}(\sqrt{KT \log(K)})$ .

---

**Algorithm 3.2.3.2** Softmax algorithm

---

**Input:**  $[a_0, r_0, a_1, r_1, \dots, a_{t-1}, r_{t-1}]$ ,  $N \geq 1$ ,  $0 < \tau < 1$

**Output:**  $a_t$

Play each action once;

**for**  $i = 0$  to  $N$  **do**

$\hat{r}_{a^i} \leftarrow$  current average reward of  $a^i$

$$p^i = \frac{e^{\frac{\hat{r}_{a^i}}{\tau}}}{\sum_{j=1}^N e^{\frac{\hat{r}_{a^j}}{\tau}}}$$

**end for**

**return**  $a_t^i$  is chosen with probability  $p^i$

---

### 3.2.3.3 UCB algorithm

Unlike the two algorithms above, the policy UCB1 (Alg. 3.2.3.3) [Auer et al., 2002] is related to an index-based policy. The UCB-index is defined as the sum of the current average reward and the size of the one-sided confidence interval for the average reward within which the true expected reward falls with overwhelming probability. In the case where the reward generating process can be represented by a set of random variables, the confidence bounds provide an useful method to deal with the exploitation-exploration trade-off. The estimated means of the random variables represent the current knowledge and guide further exploitation. On the contrary, the widths of the confidence bounds reflect the uncertainty of the knowledge and then guide further exploration. Relating means and widths gives a criteria to decide when to explore and when to exploit.

Lai et al. [Lai and Robbins, 1985] prove that an optimal algorithm achieves a regret (Eq. 3.2)  $\rho(T) = \Theta(\log T)$  as  $T \rightarrow \infty$  when the variances of the distributions  $\Phi_i$  are finite. Agrawal [Agrawal, 1995] propose a simple learning algorithm to obtain such performance. The authors used upper confidence bounds of the form  $\hat{r}_i(t) + \sigma_i(t)$  for the expected rewards  $r_i$  of the distributions  $\Phi_i$ . The  $\hat{r}_i(t)$  is an estimate for the true expected reward  $r_i$  and  $\sigma_i(t)$  is chosen such that  $\hat{r}_i(t) - \sigma_i(t) \leq r_t \leq \hat{r}_i(t) + \sigma_i(t)$  with high probability. In each round  $t$ , the decision-maker selects the action having the maximal upper confidence bound  $\hat{r}_i(t) + \sigma_i(t)$ . So, an action  $i$  is chosen if  $\hat{r}_i(t)$  is large or if  $\sigma_i(t)$  is large. If the  $\sigma_i(t)$  is large, this choice is an exploration trial. On the contrary, if an action with large  $\hat{r}_i(t)$  is selected, we call it an exploitation trial. The number of exploration trials is limited because the  $\sigma_i(t)$  decreases rapidly with each choice of action  $i$ . If  $\sigma_i(t)$  is small then  $\hat{r}_i(t)$  is close to  $r_i$  and an action is chosen in an exploitation trial only if it is the optimal action with maximal  $r_i$ . Therefore, the upper confidence bounds can really solve the exploitation-exploration trade-off.

---

**Algorithm 3.2.3.3** UCB1 algorithm

---

**Input:**  $[a_0, r_0, a_1, r_1, \dots, a_{t-1}, r_{t-1}]$ ,  $N \geq 1$

**Output:**  $a_t$

Play each action once;

**for**  $i = 0$  to  $N$  **do**

$\hat{r}_{a^i} \leftarrow$  current average reward of  $a^i$

$UCBindex_{a^i} = \hat{r}_{a^i} + \sqrt{\frac{2 \ln(N)}{n_{a^i}}}$

**end for**

**return**  $a_t^i = \operatorname{argmax}_i(UCBindex_{a^i})$

---

We denote the average (or mean or expected) reward of the best action as  $\mu^*$  and of any other action  $j$  as  $\mu_j$ . There are a total of  $K$  actions.  $T_j(n)$  is the number of times of trying action  $j$  in a total of  $n$  action. Formally, the regret after  $n$  actions is defined as:

$$\text{regret}(n) = \mu^* n - \sum_{j=1}^K \mathbb{E}[T_j(n)] \quad (3.7)$$

Regret is defined in terms of the average reward. So, if we can estimate average reward we can minimize regret. A confidence interval is a range of values within which we are sure the mean lies with a certain probability. For example, we could have believe the mean is within  $[0.2, 0.5]$  with probability 0.95. If we have tried an action less often, our estimated reward is less accurate so the confidence interval is larger. It shrinks as we get more information (i.e. try the action more often). Then, instead of trying the action with the highest mean we can try the action with the highest upper bound on its confidence interval. This is called an optimistic policy. We believe an action is as good as possible given the available evidence. There is one question: How do we calculate the confidence interval? We can turn to the classic Chernoff-Hoeffding bound [Kargin, 2005] to get an answer. Let  $X_1, X_2, \dots, X_n$  be independent random variables in the range  $[0, 1]$  with  $\mathbb{E}[X_i] = \mu$ . Then for  $a > 0$ :

$$\mathbb{P}\left(\frac{1}{n} \sum_{i=1}^n X_i \geq \mu + a\right) \leq e^{-2a^2 n} \quad (3.8)$$

The other side also holds:

$$\mathbb{P}\left(\frac{1}{n} \sum_{i=1}^n X_i \leq \mu - a\right) \leq e^{-2a^2 n} \quad (3.9)$$

So, for  $1 \leq n$  fixed:

$$\mathbb{P}\left(\mu_{\hat{k},n} + \sqrt{\frac{2 \log t}{n}} \leq \mu_k\right) \leq e^{-4 \log(t)} = t^{-4} \quad (3.10)$$

And also:

$$\mathbb{P}\left(\mu_{\hat{k},n} - \sqrt{\frac{2 \log t}{n}} \geq \mu_k\right) \leq e^{-4 \log(t)} = t^{-4} \quad (3.11)$$

The number of time that each action  $k$  is drawn on average at most is:

$$\mathbb{E}T_k(n) \leq 8 \frac{\log n}{\Delta_k^2} + \frac{\pi^2}{3} \quad (3.12)$$

So, the average cumulative regret of UCB is bounded as follows:

$$\mathbb{E}R_n = \sum_k \Delta_k \mathbb{E}T_k(n) \leq 8 \sum_{k:\Delta_k>0} \frac{\log n}{\Delta_k} + K \frac{\pi^2}{3} \quad (3.13)$$

This result deduces that the average cumulative regret is logarithmic in  $n$ .

We have the uniform bound on the regret as follows:

$$\mathbb{E}R_n \leq \sqrt{8Kn(\log n + \frac{\pi^2}{3})} \quad (3.14)$$

In this thesis, our server selection algorithm was inspired by MAB problem. More precisely, we formalized the MAB to our server selection model and based on UCB1 to design our selection algorithm. Besides the algorithm, we need a concrete scheme to implement our server selection method. The next section surveys the provider-oriented scheme, which is a widely used scheme in today's CDN services. We will show their issues and explain why we have chosen the user-oriented scheme for our proposal.

### 3.3 Server selection schemes

Nowadays, there are lots of CDN providers with different strengths and weakness all based on many different needs of a specific customer. Some representative providers include:

Akamai [Akamai, 1998]: It offers distribution services like http content and streaming media as well as other services such as network monitoring and geographic targeting. Akamai purchased InterVu (April 2000), Speedera (in 2005), Netli (in 2007). These are some big customers names of Akamai: Adobe, Yahoo, IBM, NBA, Travelocity, Sony, CNET, NTT, Myspace, US Airforce, Monster, AMD, Reuters, NASA, etc.

Limelight Networks [Limelight, 2001]: This provider gives an advanced CDN for Internet distribution of high-bandwidth media like video, music, games, etc. It has also some customers such as Amazon, Akimbo, iFilm, MSNBC. LonelyPlanet, Valve, VideoJug, etc.

EdgeCast [Edgecast, 2006]: This CDN provider offers video, games, music, content as well as live events solutions. It also serves website acceleration to increase web site performance and decrease page load times. Its customers are some Web 2.0 companies like Jaxtr, Mashable, Blip.tv, Mahalo, etc.

Besides different implementations, majority of CDN providers apply actually similar DNS-based server selection schemes [Pan et al., 2003]. These schemes are transparent to end-users. This kind of server selection is based on the DNS resolution <sup>1</sup>.

---

<sup>1</sup>Domain Name System, defined as a hierarchical distributed naming system for computer network, is an essential and effective component of the functionality of the Internet. It uses Domain Name Service to resolve queries for these names into IP addresses for the purpose of locating computer services and devices worldwide [Wikipedia, 2001]

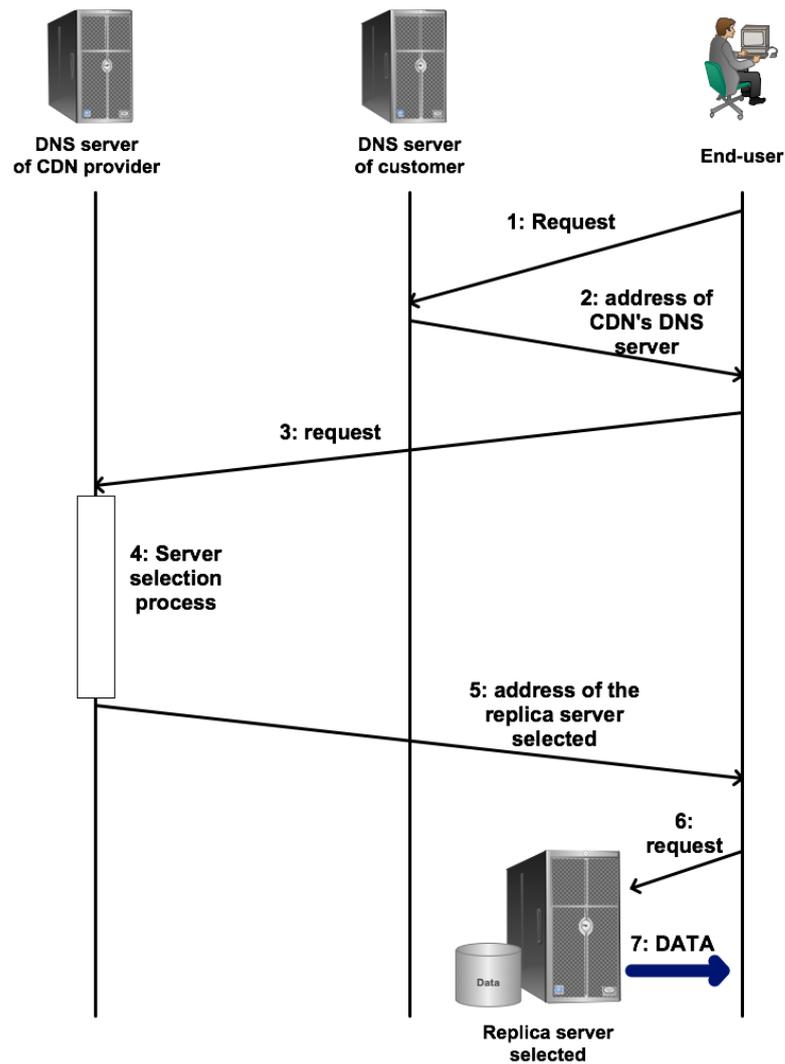


Figure 3.1: DNS-based server selection scheme

The DNS-based server selection scheme has three principal characters: *End-user*, *DNS server of customer* and *DNS server of CDN provider*. As shown in figure 3.1, this scheme is described step-by-step as follows:

- Step 1: The user sends his request to the DNS server of the customer after some basic DNS resolution phases.
- Step 2: DNS server of the customer replies to the user with DNS server's address of CDN provider.
- Step 3: After receiving the address, the end-user sends the request to DNS server of CDN provider.

- Step 4: The server selection process is launched to choose an appropriate server for the user.
- Step 5: CDN provider sends the address of the server selected to end-user.
- Step 6: The user sends request to the replica server selected.
- Step 7: The replica server provides data to end-user.

Since the sever selection is launched at the DNS server of CDN provider, this approach is considered as *provider-oriented* one. In other words, the CDN provider makes a decision of which replica server to serve the end-user. By [Pan et al., 2003], this kind of server selection has some advantages in taking advantage of the existing DNS infrastructure in making no change at content providers as well as end-user side. That makes this approach immediately deployable in the Internet. Despite of this, provider-oriented server selection has following disadvantages:

- This server selection scheme operates at the application layer. So, the IP addresses of selected replica server are treated differently. Concretely, with certain applications that are not interested in the TTL (Time To Live), an IP address of the replica server is used deterministically, regardless of how CDN provider dynamically switches between selected servers. In other words, the end-user is served always by the replica server selected even when network condition and server load change dramatically during the session. This fact reduces the effect of dynamic selection of the DNS-based method.
- There is a problem coming from the TTL setting of DNS record. If the TTL is small, end-user must send more requests to the DNS server. That causes more traffic and more DNS-related delay. Consequently, this approach faces the congestion and packet loss issues.
- Requests coming from the same local DNS of different clients are treated equally. The DNS does not take into consideration the protocol and object identifiers in requested URLs. As a result, there is no distinction of different services such as http, ftp, etc., even each service has proper requirement.
- For DNS-based approach, the CDN provider manages users as clusters. Each cluster is managed by a local DNS server, which takes into account the DNS-related resolution phase. There is a critical assumption of DNS-based server selections: all users are close to their local DNS server. Then, all users managed by a local DNS server are treated equally. However, this assumption is not always true, especially in the actual Internet with large corporations and ISP networks having wide coverage, multiple connections. This fact is proven in [Mao et al., 2002] that found that only 64% of all users have a local DNS server with the same autonomous system (AS) number. In [Shaikh et al., 2001], the authors prove that the user-DNS proximity assumption is far from realistic in showing that 15% of web client and

its DNS server pairs are in 1-hop clusters. They showed also the median cluster size is 5 hops.

- Since DNS servers cannot know location of local DNS servers and do not have complete knowledge about the network status and connectivity, provider-oriented server selection cannot select the most appropriate replica server, even when end-users are close to their local DNS server.

The disadvantages above motivate us to apply an user-oriented server selection scheme to the CDN system. According to Yoshikawa et al [Yoshikawa et al., 1997], “*in many cases the client, rather than the server, is the right place to implement transparent access to network services*” [18]. Our proposed scheme based on QoE feedback of end-user and select the replica server at the user side. Theoretically, that will resolve all these problems above.

## 3.4 Our proposal for QoE-based server selection method

Server selection is an important operation in any CDN, aiming at choosing the appropriate server according to some predefined metrics. Indeed, if an inappropriate server was chosen (e.g. the overloaded server or the very far one), the quality of delivered data would be quickly degraded. Previous researches on server selection methods are mainly based on Quality of Service metrics (QoS) such as round trip time (RTT), hop-count between a client-server pair, loss rate, etc. However, these studies did not take into account a crucial concept that constitutes a determining factor for success of the deployment of future services which is based on user perception, the QoE.

In this section, we present the proposed server selection scheme and our UCB1-based selection algorithm.

### 3.4.1 Proposed server selection scheme

In this section, we present our QoE-based CDN model (Fig. 3.2), which operates as other CDN models, except that it is based on QoE feedback of end-users instead of only QoS parameters.

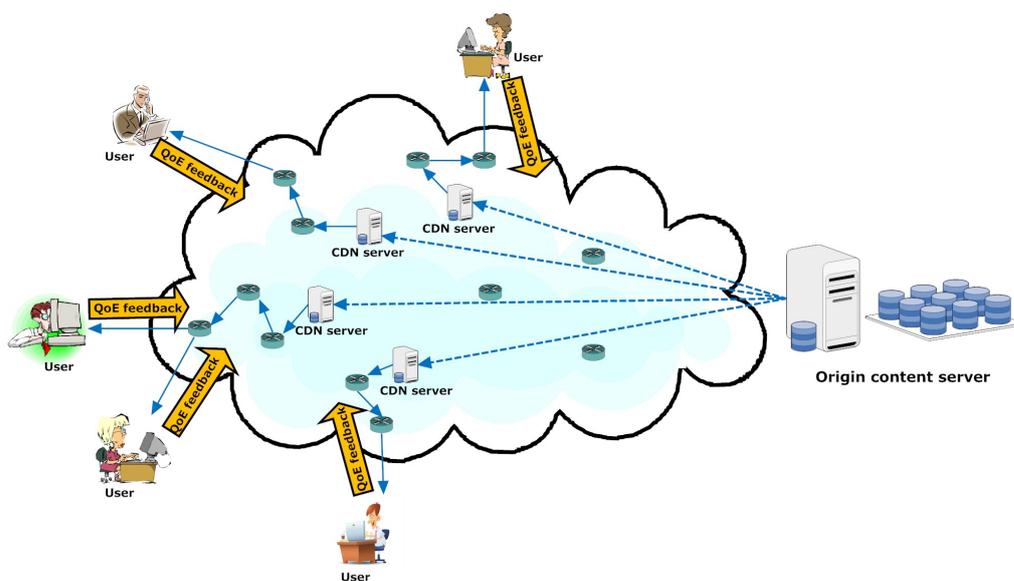


Figure 3.2: QoE-based CDN model

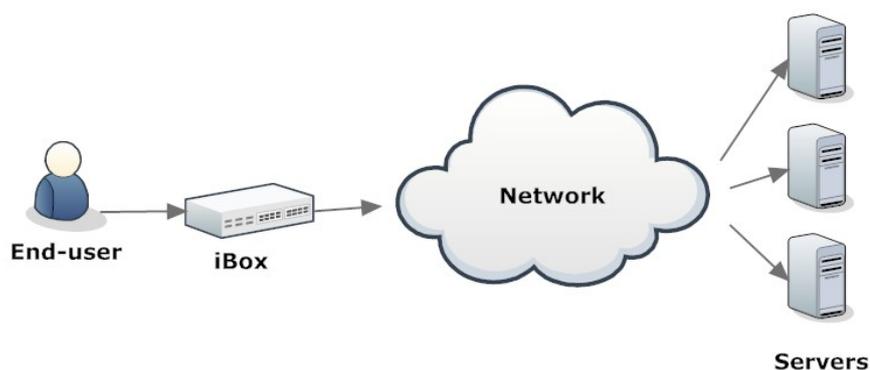


Figure 3.3: CDN model with four components

The model (Fig. 3.3) is organized in 4 components: *end-user*, *iBox*, *network* and *servers*. The iBoxes are placed at the user side, typically serving only one user. The users send requests to the system. These requests are transparently passed through the associated iBox. Based on information in its database and on a server selection policy, the iBox realizes the server selection process and chooses the server to forward the request. The iBox supports also the QoE measurement method. Each iBox has a QoE table that consists of *MOS* values, a composite metric representing QoE score. Each *MOS* entry is corresponding to a replica server. So, the number of *MOS* entries is equal to the number of replica servers.

In order to apply our proposed approach to another CDN architecture, notice that the iBox we used in our model is not necessarily a network equipment. That may be a

software embedded in any equipment with the same functionality.

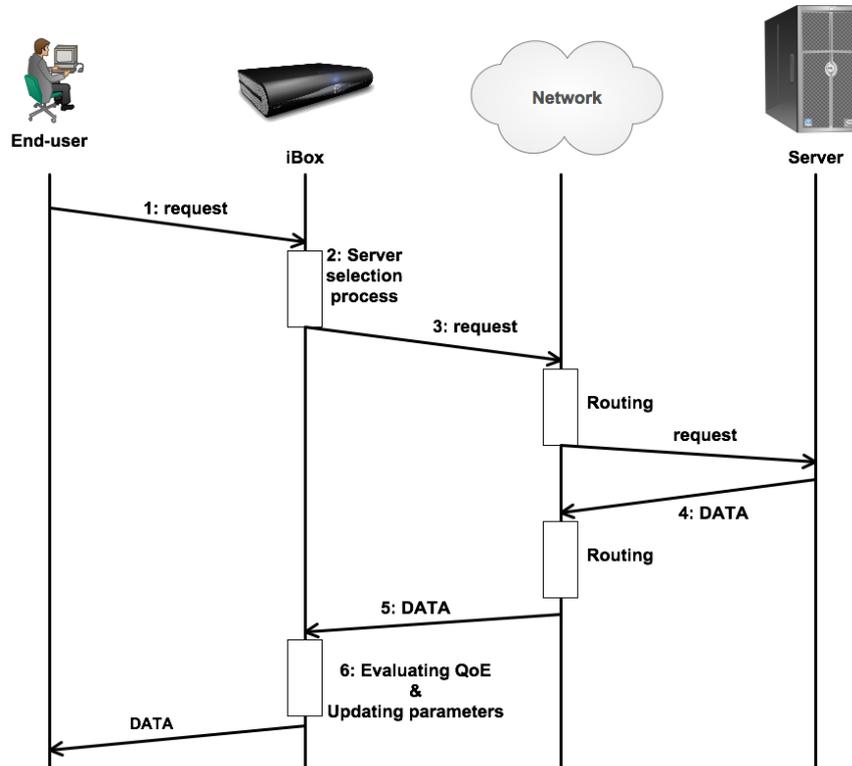


Figure 3.4: QoE-based server selection scheme

We present a step-by-step description of the server selection operations (Fig. 3.4):

1. User sends request to his iBox.
2. This iBox uses server selection algorithm to choose the adequate server based on his *MOS* table. In this step, we implement several algorithms of server selection, based on the state of the art, to compare their performances.
3. After assigning the selected server in the request packet, the iBox forwards it to the network.
4. The chosen server receives data request and serves the requested data to user. Without loss of generality, we assume that all the requested data can be found in any replica server.
5. Network system forwards data from server to the associated iBox.
6. After receiving data from server, iBox evaluates QoE to update parameters if this server is in the selection table.

### 3.4.2 Proposed UCB1-based server selection algorithm

We decide to formalize the server selection method into MAB formalization presented previously.

We present now the numerical details of our server selection based on algorithm UCB1 (Algo. 3.2.3.3). As explained previously, our selection policy aims at selecting servers based on indexes that provide UCB on the rewards (*MOS* score) associated to the server.

Each iBox has a *MOS-table* that contains *MOS* scores of the servers. The iBox uses this table to select server in using the UCB1 algorithm. In our model, we have  $N$  servers,  $s_i$  with  $i = 0 \dots (N - 1)$ . In the time  $t$ , each iBox has a vector of information  $\{s_0, M_0, s_1, M_1, \dots, s_{t-1}, M_{t-1}\}$  where  $s_j$  and  $M_j$  are respectively selected server and the measured *MOS* score in time  $j$ . The general approach to assess the average reward (*MOS* score) of a server  $s_i$  is to consider a confidence bound for its sample mean. Let  $\hat{M}_i$  be the sample mean of server  $s_i$  after being selected  $n_{s^i}$  times. The steps to select server (applied for all of iBoxes) are enumerated below:

1. Each server is selected at least once.
2. Compute the current average *MOS* score of each server

$$\hat{M}_i = \frac{\sum_{j=0}^{t-1} M_j \times \mathbf{1}_{\{s^i \text{ is chosen}\}}}{n_{s^i}} \quad (3.15)$$

Where  $\mathbf{1}_{\{s^j \text{ is chosen}\}}$  is equal to 1 if the server  $s^j$  is chosen, 0 if not.  $n_{s^i}$  is the number of chosen times of the server  $s_i$ .

3. Compute the upper bound confidence index,  $UCB_i$ , for each server  $s^i$ :

$$UCB_i = \hat{M}_i + \sqrt{\frac{2 \ln(t)}{n_{s^i}}} \quad (3.16)$$

For all servers,  $UCB_i$  gives an optimistic estimation of the expected reward obtained when the iBox selects the server  $s_i$  at a time  $t$  after being tested  $n_{s^i}$  times.

4. Choose the server having the maximum UCB index:  $s_t = \operatorname{argmax}_i(UCB_i)$ .

As shown in formula 6.4, the UCB index is composed of two part: the average *MOS* score of server  $i$ ,  $\hat{M}_i$ , and the confidence bound  $\sqrt{\frac{2 \ln(t)}{n_{s^i}}}$ . Hence, the server  $i$  is chosen if  $\hat{M}_i$  is large or if  $\sqrt{\frac{2 \ln(t)}{n_{s^i}}}$  is large. If the average *MOS* score is large, this choice is an exploitation trial. On the contrary, if a server with large confidence bound is selected, it is an exploration trial. The confidence bound is large if the number of chosen time of server  $i$ ,  $n_{s^i}$ , is small compared to the total chosen time  $t$ . In other words, the less the server is chosen, the more it has opportunity to be chosen.

### 3.5 Conclusion

Content Distribution Network is commonly an efficient solution to address the problem of scale associated with distributed services of the Internet. With this approach, the server selection policy plays an important role in CDN management. Actually, the end-users will be satisfied with the service of the most appropriate replica server. Several approaches were proposed to resolve the server selection issue. However, they all met two grave problems. First, they ignored the user's perception in assuming the QoE will be good with good QoS criteria. This assumption is proven false previously in the last chapters. Secondly, these approaches do not take into consideration the exploitation-exploration trade-off.

The chapter began with a brief presentation of MAB paradigm to give an overview of this notion and explain why we have chosen MAB to apply to our server selection method. We presented also some applications of MAB in different fields such as clinical research, network services, etc. Among them, we have chosen application of server selection that is related to the CDN infrastructure, our goal in this thesis. Then, we surveyed also three widely used algorithms:  $\epsilon$ -greedy, Softmax and UCB1. As shown in the experiment results of comparing these three algorithms, the UCB1 algorithm gives the best result in term of QoE. It is also an optimal decision method in terms of minimizing the regret. Hence, that motivates us to choose UCB1 as selection algorithm to apply in our server selection module. Finally, we proposed a novel server selection method based on Multi-Armed Bandit formalization. Our approach is implemented in a QoE-based Content Distribution Network architecture (QCDNA). The MAB formalization allows to explicitly maximize a QoE measurement without any assumption on the shape, the noisiness or the stationarity of the considered QoE function.

## Chapter 4

# Experiments

In order to validate our works, this chapter will show the experimental results of our two proposals: the routing algorithm QQAR and the new server selection method. We will present also a testbed for the dataset of the QoE measurement method and the experimental setup.

### 4.1 Introduction

Over the last few years, the CDN infrastructure becomes really a new solution for the emerging network applications such as VoIP, VoD, Video Conference, etc. As mentioned previously, the key functions of a CDN are located in two main layers: the routing and meta-routing layer. In previous chapters, we presented the two proposals for a CDN architecture: a routing algorithm, called QQAR, for the routing layer and a new server selection method for the meta-routing layer. To validate our approaches, we try to implement a testing network environment. More precisely, we deploy a simulation model and a real platform. The goal is to construct a network environment where we can test the proposed approaches.

We compare the performance of our approaches with traditional and current methods, which are applied commonly in actual network system. The obtained results show that our approaches yield indisputable progress in QoE improvement.

In this chapter, first, we describe in detail the testbed we implemented to construct a dataset for the QoE measurement method (in appendix A). Then, we present the experimental setup that consists of a real platform setup and a simulation setup. This section describes the network context and environment where we test the approaches. Finally, we show, analyze and explain the obtained experimental results.

### 4.2 Testbed for measuring QoE

The key brain of our QoE measurement method (appendix A) is training neural network that needs a real dataset as input data. To construct this dataset, we conducted an

experiment that consists of selecting a number of human and asking them to score the perceived quality of video using the MOS score.

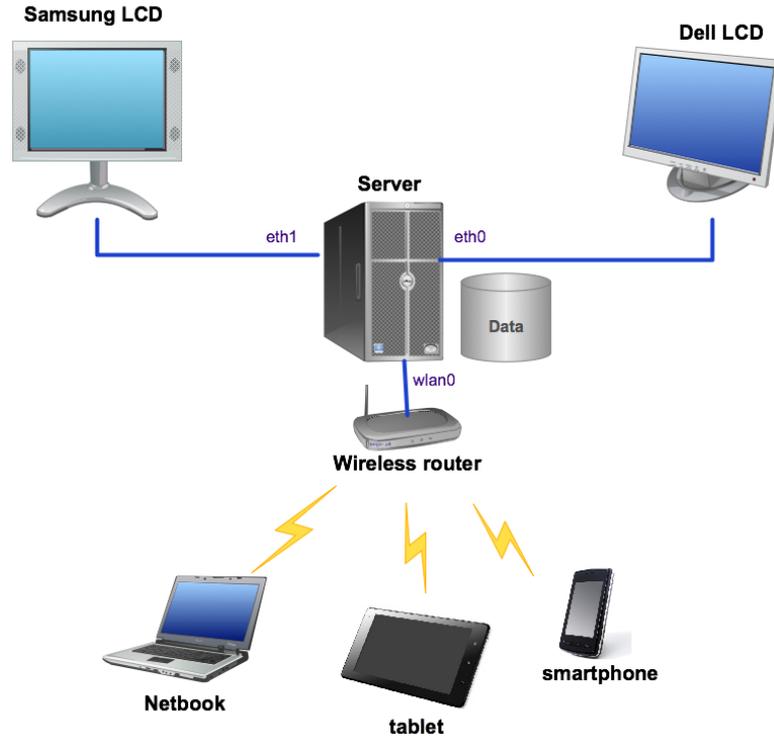


Figure 4.1: Conceptual diagram for testbed

The experimental devices (Fig. 4.1) include a *video streaming server*, *video clients* and *network emulator*, called *NETEM* [Hemminger, 2005]. The server has three network adapters. Two Ethernet adapters are intended for fixed stations and one wireless card is designed for mobile devices. More precisely, two interfaces *eth1* and *eth0* are connected to two PCs with two types of screen: *Samsung LCD* and *Dell LCD*. The interface *wlan0* is used by a wireless router that provides wifi connection to three types of devices: Netbook, tablet and smartphone. Network parameters are varied by the emulator network *NETEM* at the three network interfaces on the server in order to reproduce a real network. A conceptual diagram is shown in the figure 4.1.

We discuss now some testbed's details including video parameters, network parameters and applications used.

#### 4.2.1 Video parameters

Eighteen different video sequences were chosen for the implementation of the testbed. According to ITU-R [itu, 2000], the length of the video should be at least 5 seconds,

each video lasts 30 seconds. The video format are converted to *XviD H264* “. mp4” so they can be played on the Smartphone and Tablet. For each of the eighteen videos selected, we modified the properties of the resolution, the bit-rate and the rate of frames per second (fps) to the values indicated in Table 4.1. Consequently, each sequence could have  $3^3$  possible combinations. At the end we got 27 movies.

Table 4.1: Video parameters

Resolution	Bit-rate	fps
$1920 \times 1072$	1024	$30 \pm 7$
$720 \times 480$	512	15
$320 \times 240$	252	10

### 4.2.2 Network parameters

We have chosen four network parameters that influence the video quality at end-users: delay, jitter, loss-rate and conditional loss-rate. So, the parameter set is  $\Gamma = \{\gamma_d, \gamma_j, \gamma_l, \gamma_{cl}\}$  where  $\gamma_d, \gamma_j, \gamma_l, \gamma_{cl}$  are respectively delay, jitter, loss-rate and conditional loss-rate. The experimental setup consists of forwarding video traffic between the server and the client. Then, we introduce artificial fixed delay, variable delay and loss on the link to disturb the video signal. The design of the experiment is shown in Fig. 4.1.

According to set of values for each parameter, experiments were conducted with delay values of 0, 30, 60, 100 and 120ms, jitter values of 0, 4, 8, 16 and 32ms, loss rate values of 0, 0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5 and 5% and conditional loss probability of 0, 30, 60 and 90%. These values (Tab. 4.2) were chosen to cover the maximum of QoE range.

Table 4.2: Network parameters

	Values $\mu_i$
<b>Delay</b>	0, 30, 60, 100 and 120 ms
<b>Jitter</b>	0, 4, 8, 16 and 32 ms
<b>Loss rate</b>	0, 0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5 and 5%
<b>Conditional loss probability</b>	0, 30, 60 and 90%

### 4.2.3 Application

A web application was developed to the following processes:

- Start streaming video.
- Run the NETEM emulator.
- Play the videos on the screen.

- Collect feedback from users.
- Store all these parameters in a database.

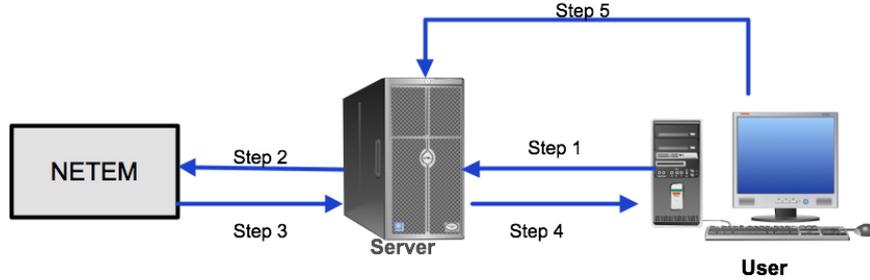


Figure 4.2: Evaluation testbed

The figure 4.2 shows the steps of our testbed. The first step (step 1) is to collect the user’s personal information (age, sex, interests, etc.). Then, the client will watch and assess 10 video sequences. In using the web browser, the client sends a request to the server indicating that he is ready to receive a video. Before the launch of video streaming, network emulator NETEM runs with random network parameters, network interface takes the new values (step 2 and 3). The fourth step is to send the streaming video. After the end of each clip, the user uses the form in figure 4.3 to give a score on the quality of video and specify the types of audio or video problems he encountered. Finally, the score is sent to the server (step 5).

After, we analyze these obtained scores to calculate the MOS score. The entire entry (combination of the configuration and the subjective score) will be added to the dataset, which will be used in the learning process of neural network (appendix A).

## 4.3 Experimental setup

In this section, we present the experimental setup including the platform and simulation setup. The platform setup consists of settings and scenario we conducted to implement a testing network environment.

### 4.3.1 Platform setup

#### 4.3.1.1 Platform settings

**4.3.1.1.1 Platform infrastructure** We have implemented a real platform with client-server infrastructure in the laboratory environment. As shown in Fig. 4.4, we have used twelve PCs: one client and eleven video servers. We used an additional PC that plays the role of an iBox at the client side. We wrote in Python language for this iBox a program having two modules:

**La qualité est:**

Excellente   
 Bonne   
 Assez bonne   
 Médiocre   
 Mauvaise

**Problèmes audio**

Le son n'était pas naturel ou présentait des distortions  
 Le son était entrecoupé  
 J'entendais de l'écho  
 J'entendais des parasites  
 J'ai remarqué des décalages

**Problèmes vidéo**

Impossible de voir la vidéo  
 L'image se fiseait ou était de mauvaise qualité  
 Les mouvements de la vidéo n'étaient pas naturels  
 La vidéo était en retard ou en avance par rapport au son

Figure 4.3: Subjective evaluation form

- Server selection: This module selects the server based on the selection policy. We installed the proposed selection algorithm as well as traditional ones to compare them.
- QoE evaluation: the QoE measurement method is described in detail in appendix A. It is a combination of objective and subjective methods. The subjective factor refers to real score given by human in using multimedia service. On the other hand, the objective factor gives the capacity of measuring the QoE in real-time. This fact is suitable for our proposals that is applied in an adaptive environment.

**4.3.1.1.2 Network Emulator: NetEm** We used a network emulator, called NetEm [Hemminger, 2005], to make the network environment change dynamically. The goal is to reproduce the network behavior in a controlled environment. By [Shaikh et al., 2010b], *network emulation is one way to evaluate the network performance in a controlled and repeatable environment.*

In our platform, the NetEm is used to introduce the variable delay, loss-rate and conditional loss-rate for the incoming packets. NetEm is one of network emulator that provide network emulation by emulating network properties. It is composed of two main parts: the kernel module for queuing discipline and command line utility for configuration. Netlink socket interface supports the communication between command line and Linux kernel. NetEm uses the FIFO queuing as the queuing discipline, which is between the network device and the protocol output. NetEm is controlled by a command line tool, called *Traffic Control(tc)*, which is a part of *iproute2* package. It has four basic operations: variable delay, packet loss, duplication and re-ordering of packets. We give here some following examples of command used in the Linux kernel to add delay to the packets:

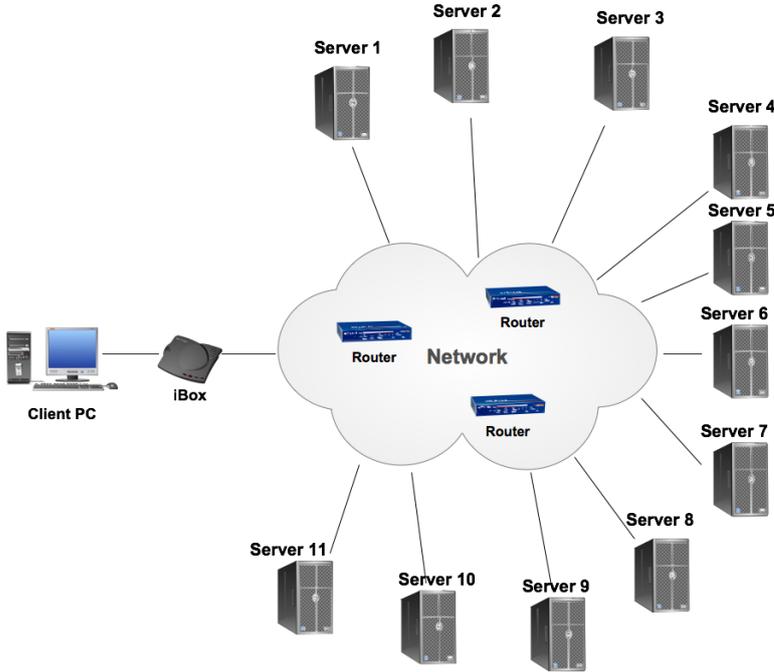


Figure 4.4: Implemented Platform

- `tc qdisc add dev ethX root NetEm delay 100ms`
- `tc qdisc change dev ethX root NetEm delay 100ms 10ms`

where `tc` command is used for configuration of the traffic control in the Linux kernel. “`qdisc`” is the queuing discipline (e.g. FIFO queue). In the first command, 100ms is added to all packets going out of a local Ethernet interface `ethX` and the second one adds an added delay of 10ms to the delay 100ms.

**4.3.1.1.3 Technical parameters** All servers used in the platform had the same technical characteristics (Tab. 4.3). We worked on the Linux OS version 2.6.32 (Debian). The PC’s model is Dell Optiplex300 with processor of Intel Pentium Dual CPU E2180 2.00GHz. The used memory is DDR2 SDRAM 4GB.

Table 4.3: Technical characteristics for PCs of the platform

Operating System	Linux version 2.6.32 (Debian)
Processor	Intel Pentium Dual CPU E2180 2.00GHz
Memory	800MHz DDR2 SDRAM 4GB

**4.3.1.1.4 Video player: VLC application** At client and server side, we use VLC application [VLC, 2001] for video streaming. VLC is released by GNU General Public License, which supports lots of formats such as AVI, MKV, MPEG, etc. and encoding formats including DIV3, H.264, MPEG-4 etc. It can function as a streaming server and support several protocols including RTP, RTSP, HTTP, UDP etc. This video player has features like codec and media information. It shows the information of the playing media file such as codec information, frame rate, bit rate, total number of frames in the file, etc. It is able to take snapshots of a video. VLC player supports also the web interface. So, the user can play or stop the video when the video is streamed. The option to view the video at a faster or slower rate. There are also the advanced options to check the the video frame by frame. Subtitles can be easily added to the video. One can zoom the playing video to maximum 2:1 double. Some video effects are available such as rotating the video, sharpness, adding contrast, etc. There are also some filter options to change the effects of the playing video.

We have chosen VLC among another applications of streaming video server (e.g. Windows Media Server, etc.) because VLC is developed for research purpose rather than for entertainment. In addition, it introduces less latency into the network and it supports several transcoding techniques, file formats and protocols.

VLC player has several transcoding techniques and some codecs are allowed to stream the video over a network. In this platform, we used MPEG4 codec and the used protocol is UDP. The international standard codec, MPEG4, provides tools to compress audio and video. It is standardized by ITU [Van der Auwera et al., 2008]. We used MPEG-4 with HD (High-definition) resolution videos ( $1280 \times 720$  pixels - 720p). For MPEG-4 video codec, we have no frame skip of quality loss compromise [Zheng et al., 2001]. We choose the real time protocol UDP for streaming videos. The drawback of using UDP is that it does not support reordering of packets.

**4.3.1.1.5 Video used for platform** All of eleven servers store the same video clip that lasts 4000 seconds (about 66.66 minutes). We have chosen the film “*Sintel*” [Ton Roosendaal, 2010]. This independently produced film is initiated by the Blender Foundation. The goal is to further improve and validate the free and open source 3D creation suite Blender. It becomes really a viable development model for both open 3D technology as for independent animation film. This film is realized by an international team of artists and developers. Several crucial technical and creative goals have been realized online by developers and artists and teams all over the world.

#### **4.3.1.2 Platform scenario**

**4.3.1.2.1 Server selection algorithms** The goal is to compare the obtained QoE results of three following algorithms:

- FASTEST policy: This policy chooses always the server that has the shortest round-trip time (RTT)
- Round Robin policy: This policy follows the circular order to select the server.

- UCB1: this algorithm is presented in chapter 3.

**4.3.1.2.2 Functional Diagram** For server selection at client side, we wrote in Python language a selection module and installed it in the iBox. The selection module is executed every twenty seconds during the video playing. In other words, the server selection is executed every twenty seconds for switching to another server or staying in the current one. We installed also the QoE measurement module at the iBox, which takes network parameters as input data and gives the *MOS* score as output.

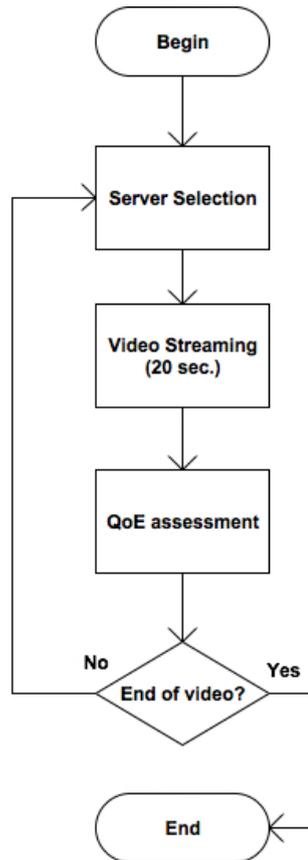


Figure 4.5: Functional diagram

As shown in figure 4.5, the platform scenario is launched by a loop of following steps:

1. Server selection: In this module, we installed three algorithms mentioned above to compare their performance in terms of satisfying user perception.
2. Video Streaming: This module lasts 20 seconds.
3. QoE assessment: It measures the QoE value. The measurement method is presented in appendix A.

### 4.3.2 Simulation setup

We implement also a network simulation model to validate the proposals. We present now the simulation setup including setting parameters, simulator, simulation scenario.

Table 4.4 lists the setting parameters:

Table 4.4: Simulation Setup

<b>OS</b>	Windows XP 32bit
<b>CPU</b>	Intel Core 2 6700 2.66GHz
<b>RAM</b>	2.0 GB
<b>OPNET Version</b>	Opnet Modeler v14.0
<b>Topologies</b>	6x6 grid 3AreasNet NTT
<b>6x6 grid network</b>	Router number: 36 User/iBox number: 3 Server number: 2 Link number: 50
<b>3AreasNet network</b>	Router number: 35 User/iBox number: 10 Server number: 5 Link number: 63
<b>NTT network</b>	Router number: 57 User/iBox number: 10 Server number: 5 Link number: 162
<b>Simulation time</b>	45 hours
<b>Confidence interval</b>	95%
<b>Arrival request distribution</b>	MMPP (Markov-Modulated Poisson Process)

#### 4.3.2.1 Simulator

There are many network simulators allowing the implementation and performance evaluation of network systems and protocols. They are either prototypes from academic research or marketed products. In this subsection, we present three common simulators: NS, MaRS and OPNET. After, we present in detail the OPNET and explain why we have chosen this one to simulate our proposals.

**4.3.2.1.1 NS** NS [Network Simulator, 1989] is distributed by the University of Berkeley (USA). This is a discrete event simulator designed for research in the field of networks. It provides substantial support for simulations using TCP and another routing protocols. The simulator is written in *C++* and uses *OTcl* as the language interface and configuration control.

The advantage of NS is offering many extensions to improve its basic characteristics. The user specifies the network topology by placing the list of nodes and links in a topology file. Traffic generators can be attached to any node to describe its behavior (e.g., FTP or Telnet client). There are different ways to collect information generated during a simulation. Data can be displayed directly during the simulation or be saved to a file for processing and analyzing.

The two monitoring modes in NS are, firstly, the registration of each packet as it is transmitted, received and rejected and secondly, the monitoring of specific parameters, such as the date of arrival of packets, size, etc.. Data generated during the simulation are stored in files and can be treated. The *Network Animator* is a tool to analyze these files. Its graphical interface displays the configured topology and packet exchanges.

The NS is an open source simulator (its source code is available in [Network Simulator, 1989]). It is widely used for research in the field of networks, and recognized as a tool for experimenting with new ideas and new protocols.

**4.3.2.1.2 MaRS** MaRS [Maryland Routing Simulator, 1991] is a discrete event simulator that provides a flexible platform for the evaluation and comparison of routing algorithms. It is implemented in language *C* under UNIX, with two GUIs: *Xlib* and *Motif*.

MaRS allows the user to define a network configuration consisting of a physical network, routing algorithms and traffic generators. The user can control the simulation, record the value of certain parameters, and save, load and modify network configurations. MaRS consists of two parts: 1) a simulation engine, which controls the event list and the user interface; 2) a collection of components to model the network configuration and to manipulate certain features of the simulation. The configuration of the simulation (network algorithms, connections, traffic) can be done via configuration files or via a graphical *X-Windows*.

**4.3.2.1.3 OPNET** The OPNET project was launched in 1987. Currently, this software is marketed by MIL3 and is the first commercial tool for simulation of communication networks.

This is a simulation tool based on discrete events, written in *C*. It has a graphical interface used in various modes, which facilitates the development of new models and simulation programs. Many models of protocols are provided with the standard distribution of OPNET, which is a definite advantage over other simulators. In fact, the other simulators often offer only an implementation of *TCP/IP* and some classical routing protocols. OPNET is a very flexible tool, which simulates a network at any level of granularity. It is used extensively in many development projects.

The advantages of this simulator tool motivate us to choose it to simulate and validate our proposals in this thesis. The next subsection presents in detail simulation settings in using OPNET simulator.

### 4.3.2.2 Simulation scenarios

The Internet today has a dynamic environment with lots of unexpected changes that cause different network issues. The *load balance* is important factor that influences the network performance. There are different load balancing methods have been introduced to avoid this issue. In fact, overloaded routers or servers can break down rapidly the entire network system. In addition, *unexpected incidents* affect the network service quality too. A failed router or link can change the network topology. This change really decreases the network quality and requires an effective fault-tolerance mechanism of network operator. Therefore, we simulate two following scenarios:

1. **Highload Scenario:** We have generated a traffic that stresses the network. The load level represents the rate of number of overloaded links and number of total links:

$$\text{level} = \frac{n_s}{N} \times 100\% \quad (4.1)$$

where  $n_s$  is the number of loaded links and  $N$  is the total number of links in the system. Therefore, the more we increase the *level*, the more network system is loaded. We tested in using seven levels 10%, 20%, 30%, 40%, 50%, 60% and 70%.

On the other hand, we increase the bandwidth of all links in the network. As shown in the equation 4.2, the new bandwidth is assigned the new value:

$$\text{BW}_{\text{new}}^i = \text{level} \times \text{BW}_{\text{old}}^i \quad (4.2)$$

where the  $\text{BW}_{\text{new}}^i$  and  $\text{BW}_{\text{old}}^i$  are respectively the new and old value of bandwidth of link  $i$ . The level is varied from 10% to 70% as mentioned above.

2. **Incident Scenario:** During the simulation time, we make routers or servers broken down randomly. The duration time of this broken phase is long enough to test the fault tolerance capacity of the approaches. This mechanism can evaluate the capacity of detecting and reacting to broken issues. That represents the dynamic of the approaches to the environment changes. Given a router X, figure 4.6 illustrates the timeline for incident scenario for this router. The broken time lasts from  $t_x$  to  $t_y$ . In this broken period, the router X neither receives nor forwards packets. The  $t_x$  and  $T_{\text{broken time}}$  are chosen randomly so that the  $T_{\text{broken time}}$  is long enough to test the reacting method of the approach.

### 4.3.2.3 Simulation metrics

In this thesis, we used the following metrics to evaluate the performance of the approaches:

- **Average MOS score:** This metric gives the overall MOS score of entire network system. It is equal to the average obtained MOS score in the client side. The equation 4.3 is used to calculate the average MOS score:

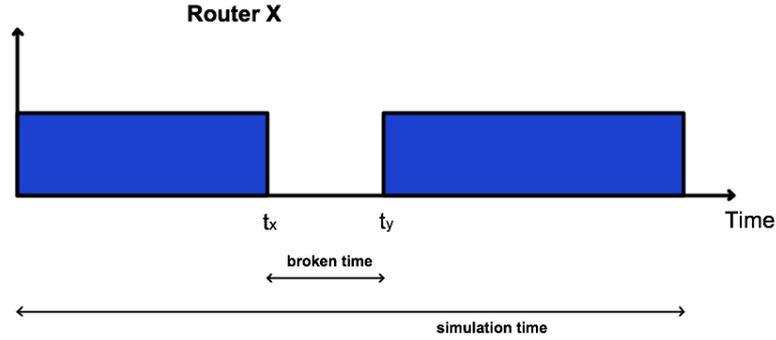


Figure 4.6: Incident timeline for router X

$$\text{Average MOS score} = \frac{\sum_{i=1}^n m_i}{n} \quad (4.3)$$

where  $n$  is the number of evaluating the MOS score and  $m_i$  is the evaluated value at the  $i$ th time.

- **Packet Delivery Ratio:** This metric gives the ratio of the data packets successfully received at the destination node and the total number of packets sent at source node. We used the equation 4.4 to calculate the PDR:

$$\text{PDR} = \frac{D_R}{D_S} \times 100 \quad (4.4)$$

where  $D_R$  is the data packets received by the destination and  $D_S$  is the data packets sent by the source.

- **Normalized Routing Overhead (NRO):** This metric is the number of control packets (e.g. error, request, acknowledge, etc.) transmitted per data packet delivered to the destination. The equation 4.5 is used to calculate the NRO:

$$\text{NRO} = \frac{(P_S + P_F)}{D_R} \quad (4.5)$$

where  $P_S$  is control packets sent by all nodes and  $P_F$  is control packets forwarded by all nodes.  $D_R$  is data packets received by the destination.

#### 4.3.2.4 Dynamic network system

One of the most important capacity of a routing algorithm is adapting to changes in crucial system parameters during network operation. In addition, the Internet today always changes dynamically. Therefore, we tested our approaches on networks whose topology, end-user's requests distribution and traffic load levels were changing dynamically.

- **Topology:** some routers in the network will break down for a period of time to change the network topology. This is a challenge for the algorithms to react quickly to such changes and continue working efficiently.
- **End-user request distribution:** The request distribution is based on MMPP distribution (Markov-modulated Poisson Process), which is presented below (subsection 4.3.2.5.3).
- **Traffic load:** We use an automatic method to change dynamically the load of links in the network in varying their bandwidth. The way to change bandwidth was discussed in section 4.3.2.2. The effect of load levels is shown more clearly in experimental results.

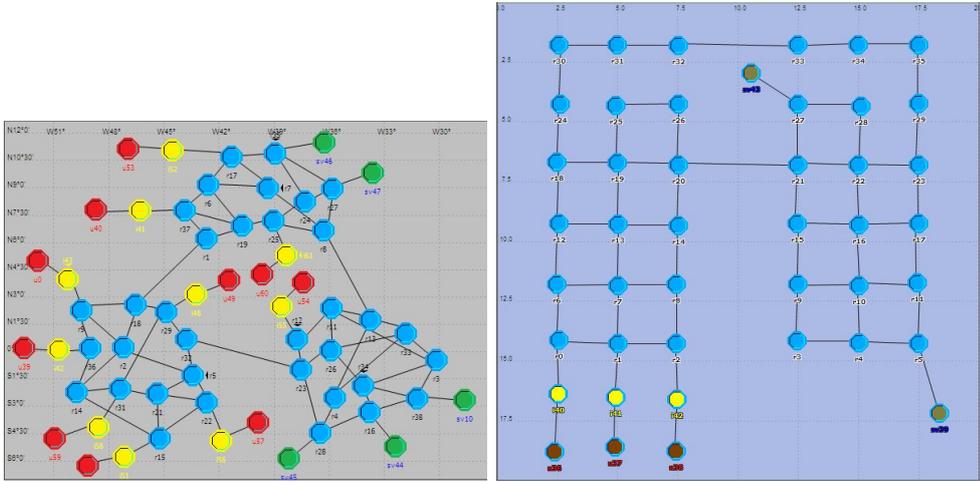
#### 4.3.2.5 Simulation settings

For performance validation of the proposals, we have chosen the simulator OPNET for several reasons. It allows to visualize the physical topology of a local, metropolitan, remote or embedded network. The protocol-specified language is based on the formal description using finite state automata. It also easier to understand the problems of transit times and network topology.

OPNET has three nested hierarchical levels: the *network domain*, the *domain node* and the *process domain*. In the next subsections, we present our simulation settings of these levels.

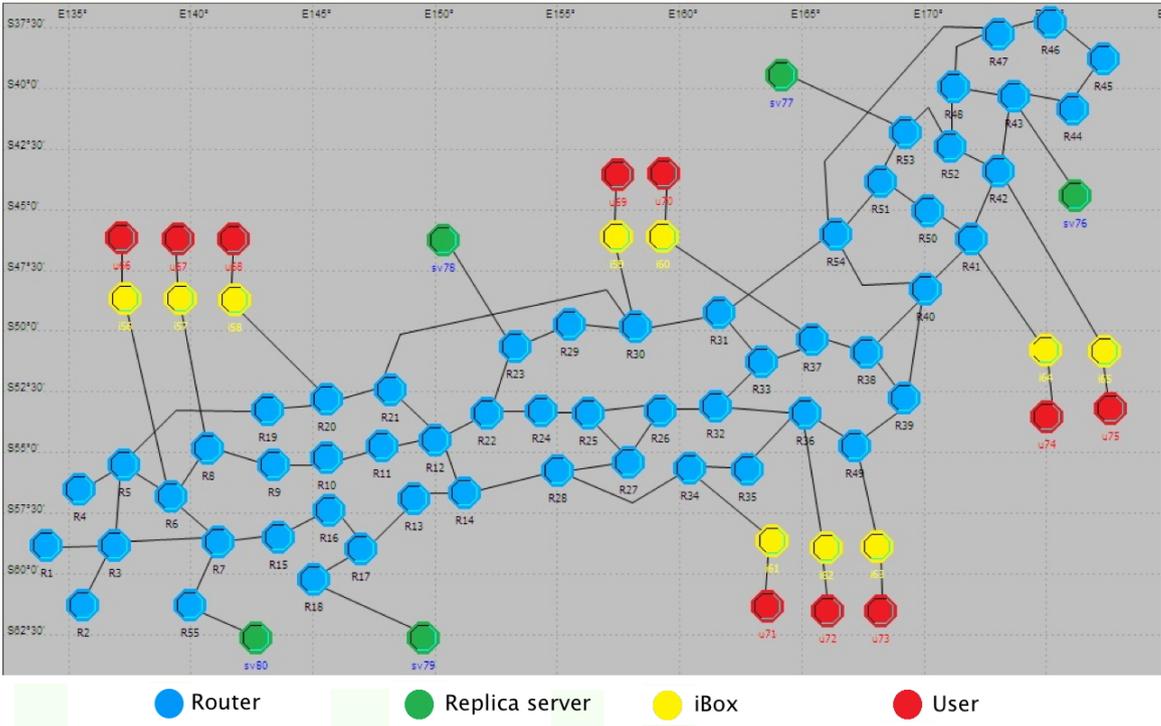
**4.3.2.5.1 Network domain** This is the highest level of the hierarchy of OPNET. It defines the network topology by deploying hosts, links and active equipment such as switches or routers. Defined by its model, each communication entity (called node) is configurable. The graphical user interface associated with the network domain takes into account the spatial deployment of the network. According to the network, we worked on three following topologies: 6x6GridNet, 3AreasNet and NTT. We chosen a testing methodology as follows.

- **6x6GridNet** (Fig. 6.7(b)): First, we tested our routing algorithm on a simple network, the 6x6GridNet, to verify its functionality. This irregular network was firstly used by Boyan and Littman [Boyan and Littman, 1994]. Then, it was widely used and accepted in routing protocols studies. The network consists of two connected areas with a bottleneck of traffic on two bridging links. After obtaining sufficient result for the first topology, we continue testing our algorithm in using the second topology, the 3AreasNet.
- **3AreasNet** (Fig. 6.7(a)): For the second testing step, we design ourselves an simulated irregular network, called 3AreasNet, with 3 separated areas including 35 routers. Each area is connected to each other by a unique link and all links are similar. This constitutes our second beta testbed, which is used to test the functionality of algorithms. The 3AreasNet topology is based on Waxman topology



(a) 3AreasNet

(b) 6x6grid



(c) NTTnet

Figure 4.7: Network topologies

[Waxman, 1993]. This method distributed the node uniformly in the network, and the edges are added considering to probability defined by equation 4.6 that depend on the distances between nodes.

$$P = \alpha \times a \exp\left(-\frac{d}{L} \times \beta\right) \quad (4.6)$$

where  $d$  represent the distance between two nodes,  $L$  is the maximum distance in the network,  $\alpha > 0$  and  $\beta \leq 1$ . The parameter  $\alpha$  increases or decreases the probability of edges between two nodes and  $\beta$  controls the ratio of long/short edges.

- **NTT** [Di Caro et al., 2005]: For the third step, we worked on NTT network that is modeled on the exact characteristics of a real-world network, the former NTT (Nippon Telephone and Telegraph company) fiber-optic corporate backbone. NT-Tnet is a 57 nodes, 162 bi-directional links network (Fig. 6.7(c)). Unlike the 3AreasNet and 6x6GridNet topologies that are used to test the functionality of algorithms, this one is used to validate the proposed approaches.

In each network, we used 10 end-users (red color), ten iBoxes (yellow color) and five replica servers (green color). According to the server placement, we used a server placement method proposed in [Qiu et al., 2001]. This method is based on a greedy algorithm [Krishnan et al., 2000]. For this method, we have to choose five servers in potential sites. The site having the lowest cost (hop-count to end-users) is chosen. Then, the iteration continues until five servers have been chosen. According to the cache organization, we assume that all requested data is stored in each server.

**4.3.2.5.2 Node domain** *Node domain* is used to define the formation of nodes (e.g. routers, workstations, hubs, etc.). The model is defined by using blocs called modules. Some modules are not programmable: it is primarily *transmitters* and *receivers*, whose unique function is to interface the node and links. On the other side, another modules are programmable such as *processors* and *queues*.

*Processors* are modules that perform a specific task node. They symbolize the different parts of the operating system of a machine, and the different layers mainly implemented in the node network (e.g. Ethernet, IP ...). In addition, these modules can communicate with each other via packet streams, which allow a packet to transit from one layer to another within the same machine. This kind of organization provides a clear vision for the protocol stack implemented in a node, and quickly understand their interactions. For example, the IP module is connected to the layer 4 modules such as TCP, UDP, and those of layer 2 (Ethernet). The statistic wires are the second type of links for communication between modules. They can be traced back the statistics from one module to another one, such as the size and the delay of queues of the transmitter.

*Process domain* defines the role of each programmable module. A module has a default main process, which can add sub-processes performing a sub-task. OPNET provides mechanisms for all processes created within a process domain to communicate with

each other via a shared memory block or scheduling interrupts. The role of a module is determined by the process model, which is described as a finite state machine. Each block represents a different state, wherein the machine executes a specific code. Transitions refer to links between blocks and are determined by conditions (interruptions, variable with some value, etc..). Actions to perform are written in language *C*. OPNET provides a library of over 400 proprietary functions specific to the use of networks (creating, sending and receiving packets, extracting values in different fields of a header, etc.).

**4.3.2.5.3 Network traffic model** In order to model the real world Internet Traffic, we studied the performance of algorithms with increasing packet traffic load. According to the traffic model, we used Markov Modulated Poisson Process (MMPP) [Okamura et al., 2009] as arrival traffic process. First, this subsection presents briefly the traditional Poisson process. Then, we describe the MMPP and explain why we have chosen this traffic model.

**Traditional Poisson process:**

In traditional traffic theory, the Poisson arrival process [Kingman, 1993] was very suitable for data and voice transmission. It is a discrete probability distribution, which describes a random variable representing the number of events that occur in a time interval. In Poisson arrivals model, there is a convenient relationship between *Poisson* (discrete) and *Exponential* (continuous) distributions. The Poisson distribution represents the number of events in a time interval. On the other hand, the exponential distribution represents time between events. The Poisson arrival model gives the following probability of  $k$  arrivals in an unit of time (1 second):

$$p_k = \frac{(\lambda)^k e^{-\lambda}}{k!} \quad (4.7)$$

According to inter-arrival time, equation 4.8 expresses the exponential distribution of inter-arrival times:

$$f_x(t) = \lambda e^{-\lambda t} \quad (4.8)$$

The Poisson arrival process has some properties for analysis [Chen, 2007]:

- It is memoryless. If the previous arrival occurred  $T$  time ago, the time to the next one will be exponentially distributed with mean of  $\frac{1}{\lambda}$  regardless of  $T$ .
- The number of arrivals in an interval of length  $t$  has a Poisson distribution with mean  $\lambda t$ .
- The sum of two independent Poisson arrival processes is a Poisson process with rate of the sum of these Poisson processes. For example, the sum of two Poisson processes with rates  $\lambda_1$  and  $\lambda_2$ , is a Poisson process with rate  $\lambda_1 + \lambda_2$ .

**Markov modulated Poisson process (MMPP):**

Since the arrival rate  $\lambda$  of Poisson process is constant, it is obviously unrealistic as a traffic model. For example, with the aggregation of multiple packet speech flows, the starting and terminating time are random times, so the flow rate is not constant. Markov modulated Poisson Process (MMPP) has been introduced to solve this issue. MMPP method modulates the Poisson rate as the states of a continuous-time Markov chain. The MMPP has the memoryless property of the Poisson process and can be analyzed by Markov theory. The complexity of the analysis increases with the number of states, so this number is usually small.

Concretely, we have designed a Markov chain with three states of rate values: 1, 4 and 10 (fig. 4.8). In the figure 4.8, three states represent three rate values. The number on each transition arrow is the probability of state transition.

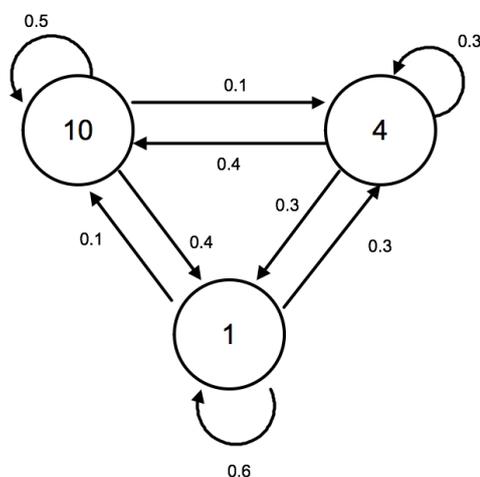


Figure 4.8: Poisson rate modulated by Markov chain

Applying MMPP method gives us the experiment results in figure 4.9, which shows the switch between the states following the Markov chain in figure 4.8. The axis  $y$  in these figures represents the number of arrival process in one second and the axis  $x$  is the time(in sec.). Taking an example of figure 4.9(a), it makes a switch from state of  $\lambda = 1$  to state of  $\lambda = 4$  at 300th second. After, at 650th second, it returns to the state of  $\lambda = 1$ . Then, during about 120 seconds from the 780th second to the 900th second, the state is  $\lambda = 10$ . The switch between the states follows the Markov chain in figure 4.8. The during time follows the exponential distribution in equation 4.8 with the  $\lambda$  is the value of current  $\lambda$ .

In this thesis, we used the MMPP model to implement the arrival process for the user request model. Actually, using the MMPP model gives a traffic model with different load level for network system. In fact, the higher the value of  $\lambda$  is, the more the network system is loaded.

The next section shows the obtained results after launching the experiments in using

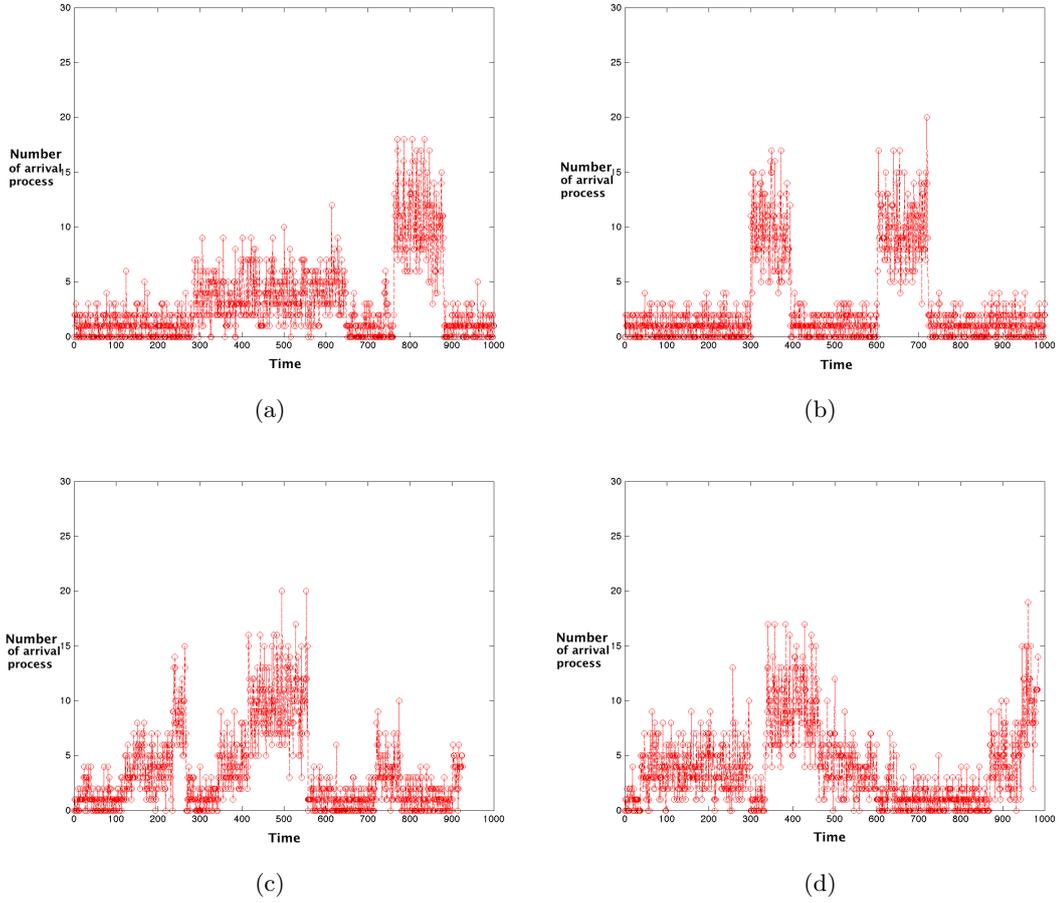


Figure 4.9: Markov modulated Poisson process

a simulation model and a real platform, which were described in the previous section. We divide this section into three main parts: 1) the results obtained for QQAR protocol, 2) results obtained for the proposed server selection and 3) results of the QCDNA model that is a combination of our two proposals in two layers of a CDN architecture. In the two first parts, we tested separately two proposals. For QQAR algorithm, we applied the same Round Robin server selection method for all routing algorithms. For testing server selection algorithms, we applied the standard routing protocol OSPF (in the routing layer) for all server selection algorithm. The goal of this separation is to validate each one before combine them in a CDN model (subsection 4.6), called QCDNA. In addition, this kind of separated test shows that each one of our two proposals is able to be applied in any CDN model, not necessary to implement both of them.

## 4.4 QQAR protocol: experimental results

### 4.4.1 Experimental setup

To validate the results of QQAR, we compare our approach with three kinds of algorithm:

- *Those based on Distance-Vector (DV) algorithm.* DV algorithm builds routing tables where no router has the overall vision of the network. The distribution of paths is being gradually. The term “distance-vector” is that the protocol manipulates vectors (arrays) of distances to other nodes in the network. The “distance” is the number of hops to reach the neighboring routers. In this algorithm, routers can maintain the optimal route by storing the address of the next router in the routing table so that the number of hops to reach the destination is minimal. The routes are updated periodically (e.g., the update period of RIB protocol is 30 sec.). DV algorithm uses the Bellman-Ford algorithm to find paths. It is a dynamic programming algorithm for finding shortest paths from a given source node in a weighted directed graph. Unlike Dijkstra’s algorithm, which can only be used when all edges have non-negative weights, the Bellman-Ford algorithm allows the presence of some arcs of negative weight and detects the existence of a circuit absorbent, that is to say the total weight of negative, accessible from the top source.
- *Those based on Link-State algorithm: SPF (Short Path First).* In this algorithm, each router establishes relations with its neighbors by sending hello messages. Each router then forwards the list of networks it is directly connected by messages (LSA-Link State Advertisements) to spread gradually to all routers in the network. The set of LSAs forms the database links Link-State Database (LSDB), which is identical for all participating routers. Each router then uses Dijkstra’s algorithm to determine the shortest path to each network known in the LSDB.
- *Those based on Standard Optimal QoS Multi-Path Routing (SOMR) algorithm* where routing is based on finding the K-Best Optimal Paths and uses a composite function to optimize delay and link cost criteria simultaneously.

While we can use static routing for small network like LAN (Local Area Network), larger (or huge) network like Internet having complex topologies needs adaptive routing protocols, which attempt to solve this problem by constructing routing tables automatically. Three adaptive routing algorithms above are representative ones of today’s widely used routing systems. That motivates us to use them to compare with our proposed routing algorithm.

The comparison in our experiment is based on the average MOS score, packet delivery ratio and normalized routing overhead that represents the QoE value. For the video streaming, we used two types of video: the *SD Video* (Standard Definition Video) and the *HD Video* (High-Definition Video). The resolution of SD Video is  $768 \times 576$ , while the resolution of HD Video is  $1280 \times 720$  pixels (720p). We use the HD Video with

large volume files to test network with lots of loaded flows. In other words, we intend to test our network in different load levels. The first set of experiments were conducted in using the SD video files for video streaming. Then, we use the HD Video files to test our approach in a stressed network. Finally, we stressed the network in adding some data flow with different load levels.

We observe the average MOS score at ten access points and the experiment has been relaunched ten times as a cross validation method. The average MOS value is calculated with sliding windows of one hour. We also present the confidence interval of 95%. The duration time is 45 hours.

#### 4.4.2 Average MOS score

Figures 4.10, 4.11 and 4.12 illustrate the experimental result of average MOS score of four algorithms in three network topologies: *the 3AreasNet*, the *6x6GridNet* and *the NTTnet*, respectively. As mentioned previously, for each topology, we implement two types of video: the SD Video and the HD Video. For these experiments, we collect the MOS score every 5 hours.

In observing Fig. 4.10(a) and 4.10(b), it is seen that after the first five hours, the four protocols obtain the stable MOS score. These stable values are 4.1, 3.3, 2.95 and 2.6 for QQAR, SOMR, SPF and DV, respectively (Tab. 4.5).

Table 4.5: MOS score for SD Video streaming of 3AreasNet topology

	QQAR	DV	SOMR	SPF
MOS score	4.1	2.6	3.3	2.95
Improvement of QQAR	-	157.7%	124.2%	138.98%

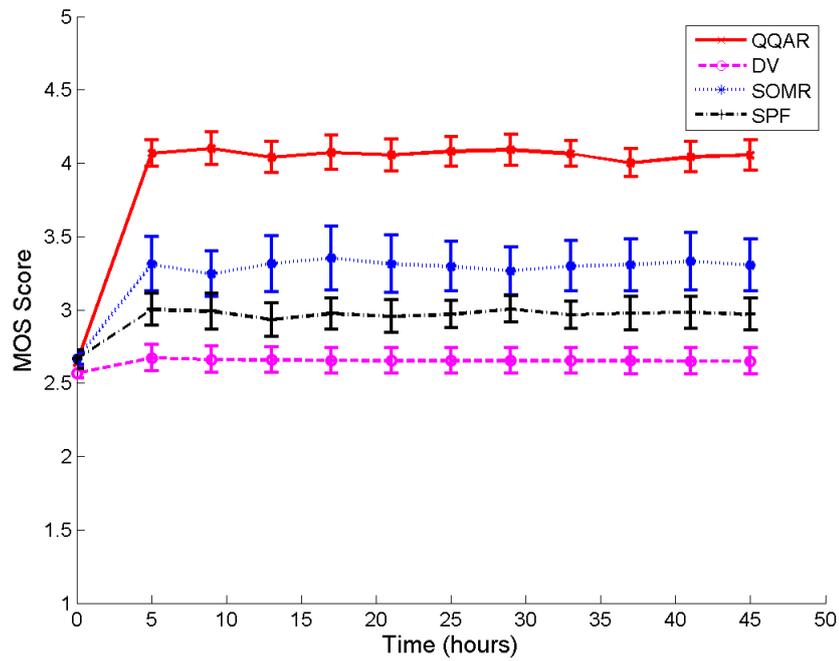
According to the effect of network load on the results, video streaming with HD Video depresses QoE value of four protocols with the decrease from 4.1 down to 3.65, 3.3-3.2, 2.95-2.8 and 2.6-2.45 for QQAR, SOMR, SPF and DV, respectively (Tab. 4.6).

Table 4.6: MOS score for HD Video streaming of 3AreasNet topology

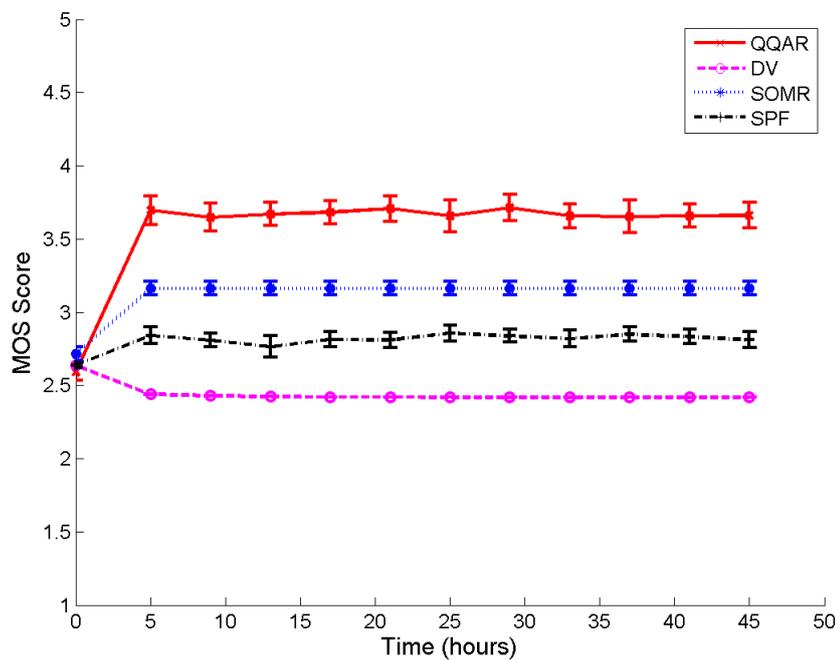
	QQAR	DV	SOMR	SPF
MOS score	3.65	2.45	3.2	2.8
Improvement of QQAR	-	148.97%	114,06%	130.35%

Even the HD video streaming decreases the QoE value of protocols, the obtained MOS score of QQAR is better than the three other ones with the improvement of 148.97%, 114,06% and 130.35% for DV, SOMR and SPF, respectively.

Similarly, in the 6x6GridNet topology (Fig. 4.11), the four protocols obtain the stable MOS scores: 4.1, 3.2, 3.5 and 3.35 for QQAR, DV, SOMR and SPF, respectively. The HD video streaming lowers the MOS scores of approaches: 4.1 down to 3.9, 3.2-3.15, 3.35-3.3 for QQAR, DV and SPF, respectively. This decrease of MOS score is negligible. The SOMR is not affected by the increase of load.

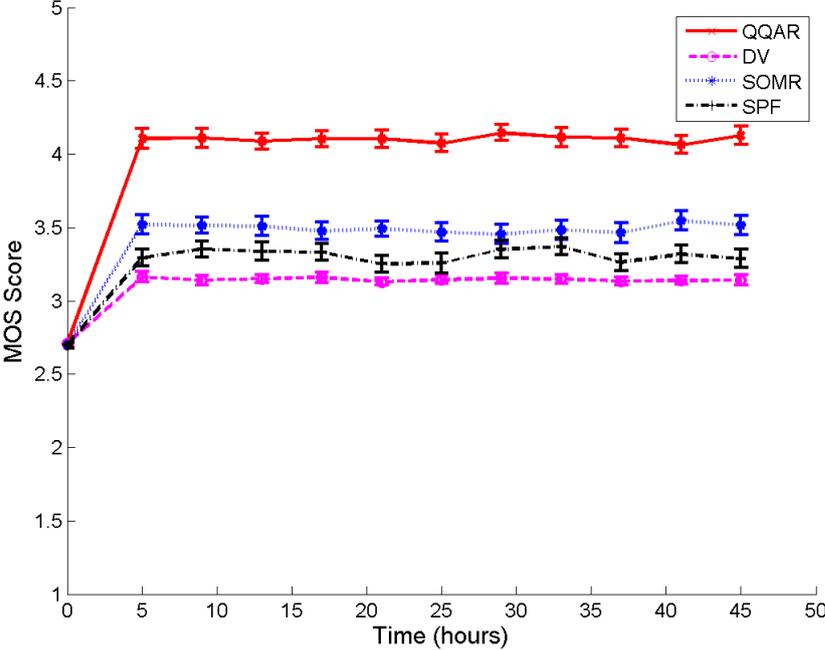


(a) Using SD Video

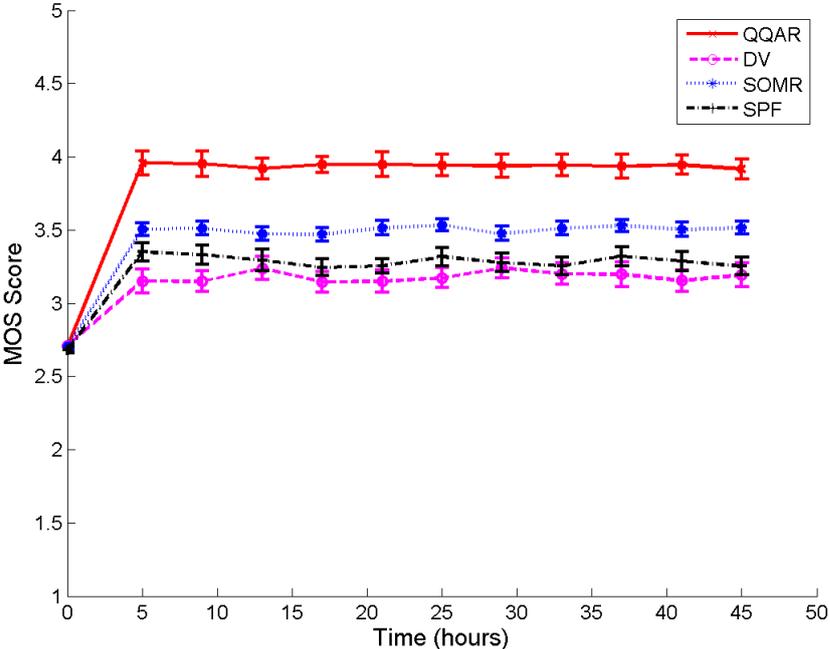


(b) Using HD Video

Figure 4.10: MOS score for 3AreasNet topology

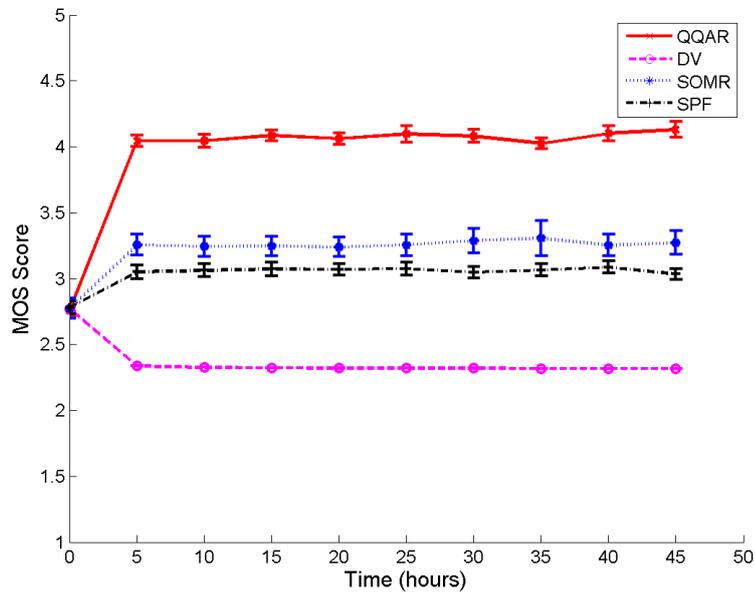


(a) Using SD Video

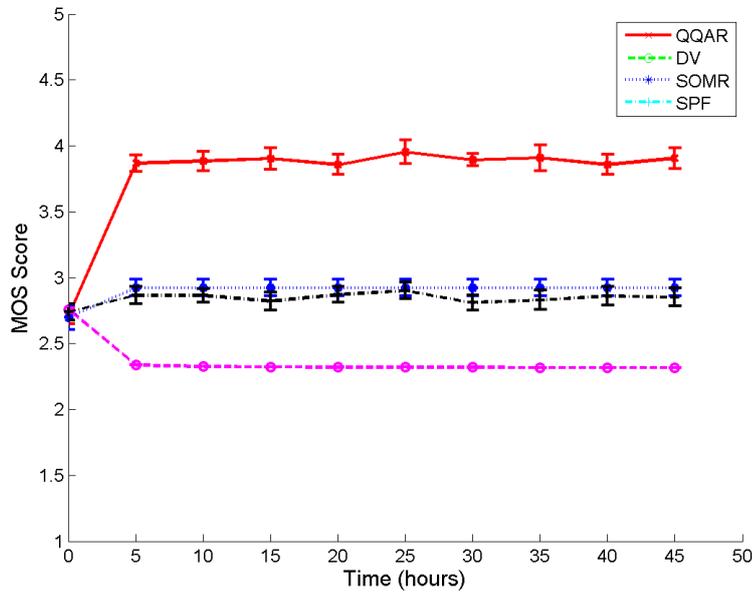


(b) Using HD Video

Figure 4.11: MOS score for 6x6GridNet topology  
120



(a) Using SD video



(b) Using HD video

Figure 4.12: MOS score for NTTnet topology

Table 4.7: MOS score for SD Video streaming of 6x6GridNet topology

	QQAR	DV	SOMR	SPF
MOS score	4.1	3.2	3.5	3.35
Improvement of QQAR	-	128.12%	117.14%	122.39%

Table 4.8: MOS score for HD Video streaming of 6x6GridNet topology

	QQAR	DV	SOMR	SPF
MOS score	3.9	3.15	3.5	3.3
Improvement of QQAR	-	123.8%	111.42%	118.18%

We analyze now the results in using NTTnet topology (Fig. 4.12). According to the SD video streaming, the average MOS scores of QQAR, SOMR, SPF and DV are 4.1, 3.25, 3.05 and 2.3, respectively (Tab. 4.9). For the HD video streaming, the QQAR, SOMR, SPF and DV obtain respectively MOS scores of 3.9, 2.9, 2.8 and 2.3.

Table 4.9: MOS score for SD Video streaming of NTT topology

	QQAR	DV	SOMR	SPF
MOS score	4.1	2.3	3.25	3.05
Improvement of QQAR	-	178.26%	126.15%	134.42%

We have the same conclusion as the previous topology: the obtained MOS score of QQAR is better than the three other ones in both types of video streaming.

In three networks: *3AreasNet*, *6x6GridNet* and *NTTnet* (figures 4.10, 4.11 and 4.12), the algorithm QQAR gives better QoE value than three other ones. That is explained by following reasons:

- While the other protocols are based on Dijkstra algorithm, which finds the shortest paths in terms of a cost function, QQAR algorithm is based on Q-learning algorithm of Reinforcement learning, which can learn the environment changes and the obtained feedback to be increasingly adaptive.
- QQAR is based on a non-linear combination of QoS parameters. More precisely, we used four network criteria: *delay*, *jitter*, *loss-rate* and *conditional loss-rate* (more details in appendix A). On the other hand, the SOMR used a composite linear function to optimize delay and link cost criteria. As mentioned previously, the user perception does not depend linearly on the technical parameters. So, a linear combination of QoS parameters is not suited to be the cost function to optimize.
- As proven in section 2.3.3, QQAR is able to detect exactly where the network issues come from without any broadcast mechanism.

According to the DV algorithm, it is the most widely accepted routing protocol. It is straightforward in broadcasting their entire current routing database periodically,

Table 4.10: MOS score for HD Video streaming of NTT topology

	QQAR	DV	SOMR	SPF
MOS score	3.9	2.3	2.9	2.8
Improvement of QQAR	-	169.56%	134.48%	139.28%

typically every 30 seconds. The name “distance-vector” means that there is a distance (a cost) and a vector (a direction) for each destination. The DV algorithm works fine for small, stable network. That explains it obtains a better QoE result in 3AreasNet and  $6 \times 6$  grid topology than NTTnet topology. In addition, there are some drawbacks of DV algorithms:

- For this algorithm, the best path is the one having the least number of node hops. However, the routing path having the least hop-count is not enough to guarantee good service quality.
- There is no protection from routing loops.
- It uses a limited hop-count (usually 15 sec.), which is an important reason makes DV algorithm unsuitable for large networks.

These arguments above explain the worst QoE results of DV compared with the three other ones.

For SPF algorithm, it is considered as a link-state algorithm. In contrast to DV algorithm, where each node *tells all neighbors about the world*, the link-state node *tells the world about their neighbors*. The SPF is better than DV suited for large, dynamic networks. That explains the fact that SPF obtain better results than DV algorithm. However, in a large network, a link change can infer a flood of thousands of link-state messages that propagate across the entire network. In each router, the database storing these messages can be quickly overloaded. Each time there is a network change, routers must redetermine new paths. Fortunately, SPF is designed to resolve this issue by devising the SPF area. SPF areas are subdivisions of an entire network. More precisely, in the 3AreasNet topology (Fig. 6.7(a)), SPF configures in dividing the network in three separated areas. In the 6x6GridNet topology (Fig. 6.7(b)), the network is divided in two separated parts. SPF routers in one area do no exchange topology updates with the routers in the other areas. This reduces the update message number of flow through the network. This improvement is showed clearly in our convergence time studies.

According to Standard Optimal QoS Multi-Path Routing (SOMR) algorithm, it is useful for finding multiple paths between a source and destination in a single discovery. This multiple paths-based algorithm can be used to react to the dynamic and unpredictable topology change in networks. For example, when the shortest path found in using Dijkstra’s algorithm is not available for some reason, it is necessary to find the second shortest path. These paths are considered as k-shortest paths. This advantage explains the good MOS scores of the previous experiments: 3.3-3.2 in 3AreasNet, 3.5 in 6x6GridNet and 3.25-2.9 in NTTnet. The only drawback of this kind of algorithm

is time taken to calculate the  $k$ -shortest path. This algorithm is used several times to calculate  $k$ -shortest path with  $k > 1$ . Due to this computational complexity, the SOMR obtains a lower quality than QQAR. In addition, the considered cost of SOMR is a linear combination of network parameters. However, the QoE depends not only on technical criteria but also on the human perception, which is an important subjective factor we have considered in our proposals. More precisely, in our QoE measurement method presented in appendix A, we determine the QoE in using a hybrid way that combines objective and subjective methods.

#### 4.4.3 Convergence time

We discuss now the convergence time of these approaches. Network convergence is a notion that is used under various interpretations. In a routing system, every participating router will attempt to exchange information about the network topology. Network convergence is defined as the process of synchronizing network forwarding tables from the beginning (the initialization phase) or after a topology change. When there are no more forwarding tables changing for “some reasonable” amount of time, the network is said “converged”. For the learning routing algorithms like QQAR, the convergence time refers to the initialization phase for exploring the network environment. After this phase, such algorithm becomes stable.

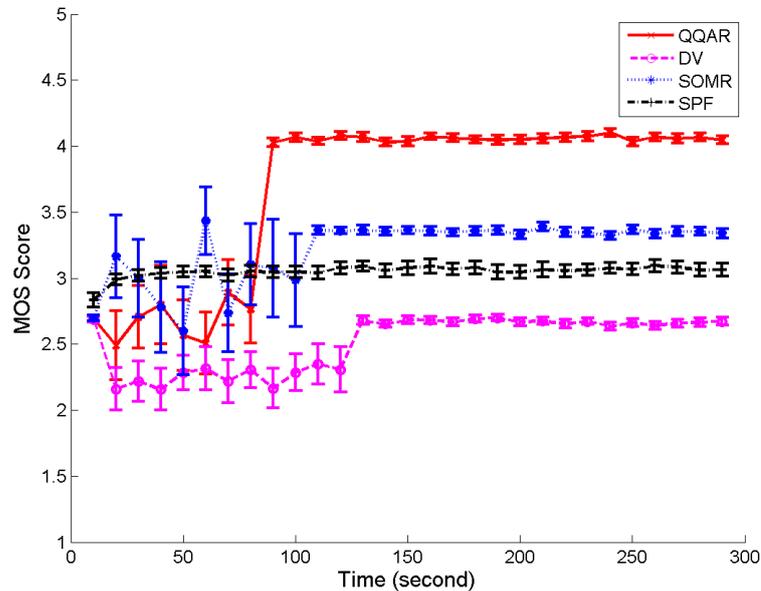


Figure 4.13: Initialization phase for 3AreasNet topology

The three figures 4.13, 4.14 and 4.15 show the obtained MOS scores of four algorithms in the first five minutes of simulation. The MOS scores are taken every 10 seconds. We

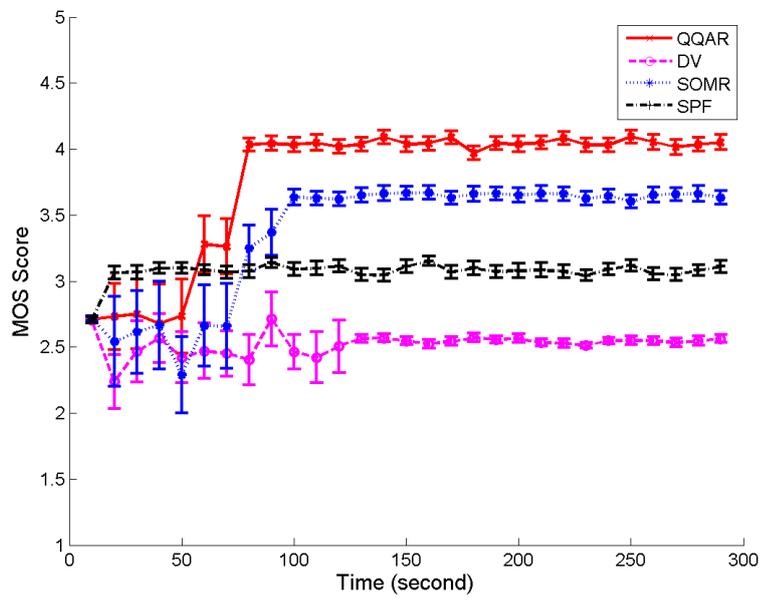


Figure 4.14: Initialization phase for 6x6GridNet topology

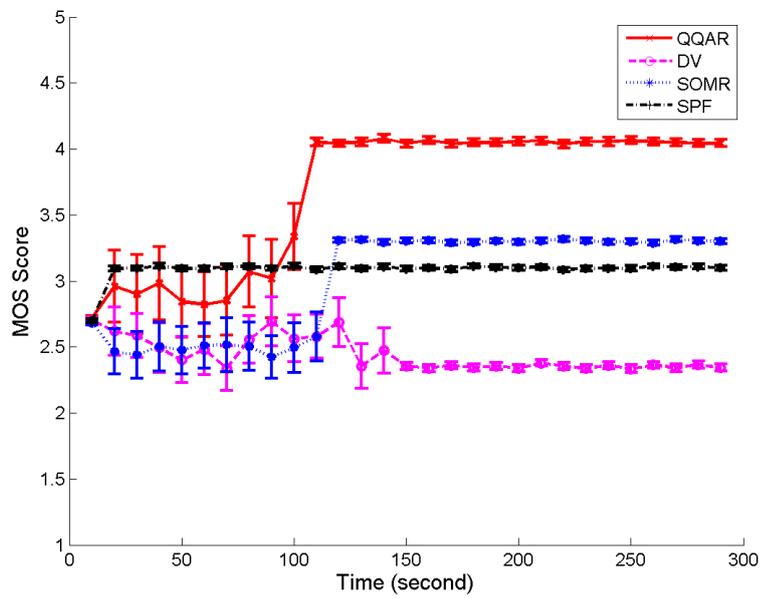


Figure 4.15: Initialization phase for NTTnet topology

consider three cases: for 3AreasNet (Fig. 4.13), 6x6GridNet (Fig. 4.14) and NTTnet (Fig. 4.15) topology. The goal is to study the convergence time of each approach in the initialization phase. All of four algorithms have a common feature: in the initialization period, they have no information about the system. So, during this period, they try to explore the network. We call this period *initialization phase*.

As shown in Fig. 4.14, in the 6x6GridNet topology, the convergence time are 20, 90, 110 and 130 respectively for SPF, QQAR, SOMR and DV. For the 3AreasNet network, the convergence time are 20, 90, 110 and 130 seconds for SPF, QQAR, SOMR and DV, respectively. For the NTTnet, the convergence time are 20, 110, 120 and 150 seconds for SPF, QQAR, SOMR and DV, respectively (Tab. 4.11). The more large the network is, the more time it takes for convergence.

Table 4.11: Convergence time

	QQAR	DV	SOMR	SPF
Convergence time for 3AreasNet (sec.)	90	130	110	20
Convergence time for 6x6GridNet (sec.)	80	130	100	20
Convergence time for NTTnet (sec.)	110	150	120	20

According to DV, it is considered as a slow convergence routing algorithm due to the slow propagation of update messages. The DV routers broadcast periodically their routing database every 30 seconds. Especially, in large or slow links networks, some routers may still publicize a disappeared route. That explains the long time of convergence of DV algorithm shown in the Tab. 4.11. It obtains the stable QoE value after 130 seconds in 3AreasNet and 6x6GridNet network and 150 seconds in NTTnet network. DV convergence is the slowest one among four algorithms.

In contrast to the DV, SPF algorithm converges quickly and accurately. In both network topologies, the small one (3AreasNet and 6x6GridNet) and the large one (NTTnet), SPF works very well with the obtained result shown in Tab. 4.11. It archives an impressive result of 20 seconds of convergence. As explained above, the SPF algorithm divides 3AreasNet in three separated areas and 6x6GridNet in two areas. The routers within one area do not exchange topology updates with the ones in the others areas. That reduces the number of update messages to alleviate the network load. So, these advantages make SPF to be the fastest algorithm in term of convergence time among four algorithms.

For QQAR algorithm, the routers do not have the knowledge of entire network. Each QQAR router has a routing table containing Q-values of their neighbors. The exchange phase between each router and their neighbors takes time for convergence phase, especially in the large network like NTTnet. That is explained the obtained results in Tab. 4.11: 80, 90 seconds in 6x6GridNet and 3AreasNet, but 110 seconds in NTTnet. In fact, 110 seconds is still acceptable duration for convergence time of a routing protocol.

The SOMR-based routing uses Diffusing Update Algorithm (DUAL) algorithm [Garcia-Lunes-Aceves, 1993] to obtain loop-freedom at every instant throughout a route

computation, while DV algorithm meets temporary micro-loops during the time of re-convergence in the topology due to inconsistency of forwarding tables of different routers. Therefore, as shown in the Tab. 4.11, SOMR gives a better convergence time results than DV algorithm. More precisely, these are 100, 110 and 120 seconds in 6x6GridNet, 3AreasNet and NTTnet, respectively.

#### 4.4.4 Capacity of convergence and fault tolerance

We have also tested the algorithm's capacity of convergence as well as fault tolerance when a change of topology occurred in the network. To implement this situation, we broke down the link  $R18-R19$  of the 6x6GridNet topology (Fig. 4.16) at the 20th hour. After this change of topology, the two areas (the left one and the right one) of the 6x6GridNet network are connected by only one link:  $R16-R25$ .

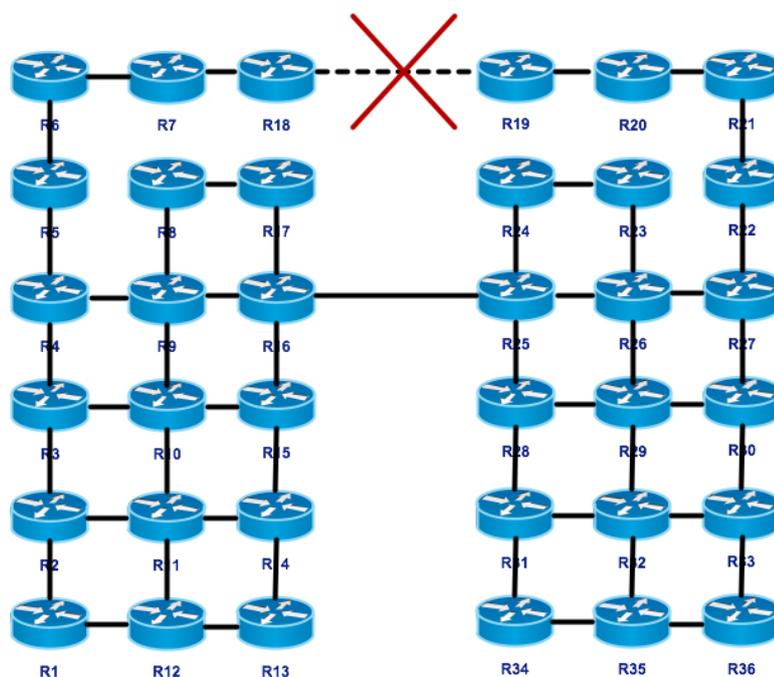


Figure 4.16: 6x6GridNet topology after breaking down the R18-R19 link

As shown in figure 4.17, after the 7200th second where the link  $R18-R19$  was broken, the MOS scores of four algorithms decreased down to value of 1.7. According to the convergence time (Tab. 4.12), the three algorithms QQAR, SOMR and SPF take the same time as the convergence time for initialization phase (Fig. 4.14). However, even after the convergence phase, the re-obtained MOS scores of these algorithms are lower than the ones before the topology change. More precisely, the QoE value decreases from 4 to 3.7 for QQAR, 3.6-3.4 for SOMR, 3-2.8 for SPF and 2.55-2.4 for DV. This fact can be explained as follows. After the topology change, the data exchange between two areas

of 6x6GridNet network is realized on only one link: the link  $R16-R25$ . Therefore, the congestion issue of this link caused the quality degradation. One more special comment, the DV algorithm has no convergence phase in this context because even before the topology change, it used just the link  $R16-R25$  to exchange data between two areas. So, the disappearance of the link  $R18-R19$  does not much influence on DV algorithm result.

Table 4.12: Convergence time for a topology change

	QQAR	DV	SOMR	SPF
Convergence time (sec.)	80	-	130	20

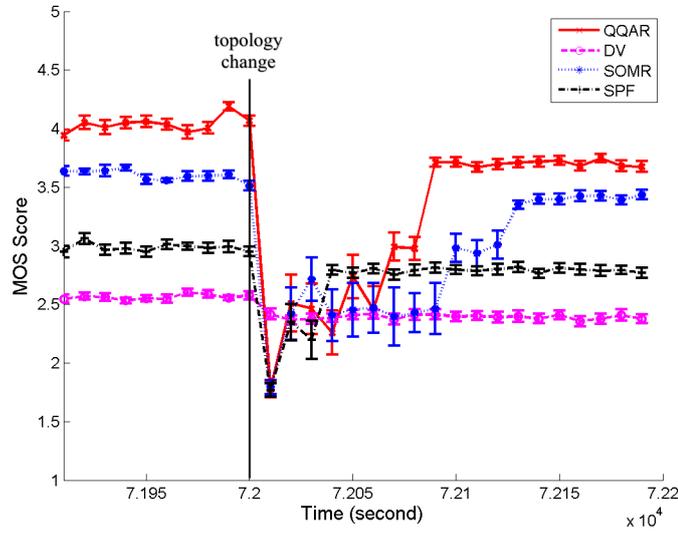


Figure 4.17: MOS score after topology change

Fig. 4.18 gives us the average of these four algorithms in different load levels formulated in (4.1): from 10% to 70%. We can see that the more the system is loaded, the more the average MOS score decreases. However, at any load level, the average MOS score of QQAR is higher than the three other algorithms. For a load level of 10%, the MOS score of QQAR is 4.1 (note that 4 represents a good quality in MOS score range). Regarding the three other protocols, the maximum value obtained by SOMR is just 3.3 with a load level of 10%.

QQAR gives a better e2e QoE result than three other algorithms in any load level and in the same network context. So, with our approach, despite network environment changes, we can maintain a better QoE than traditional algorithms. Thus, QQAR is able to adapt its decisions rapidly in response to changes in network dynamics.

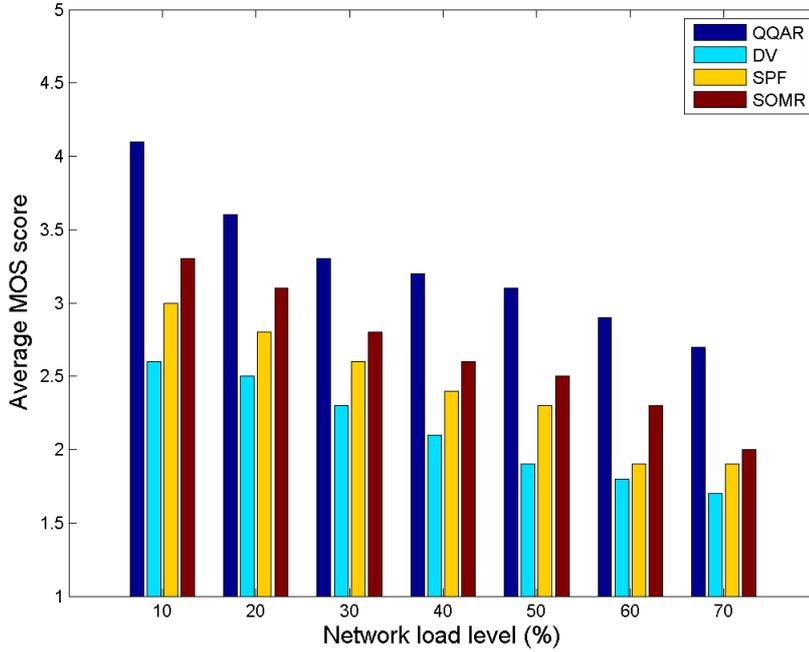


Figure 4.18: User perception in different load levels

#### 4.4.5 Control overheads

Our experiment works consist also in analyzing overheads of these protocols. The type of overhead we observe is control overhead that is determined by the proportion of control packet number as compared to the total number of packets emitted. To monitor the overhead value, we have varied the node number in adding routers to the network system. So the observed node numbers are 38, 50, 60, 70, 80. The obtained results are showed in Fig. 4.19.

We can see that the control overhead of our approach is constant (50%). That is explained by the equal number of control packets and data packets in QQAR. Each generated data packet leads to an acknowledgement packet generated by the destination node. The control packet rates of DV, SPF and SOMR are respectively 0.03, 0.4 and 0.1 (packet/second). This order explains the control overhead order in Fig. 4.19. While the SPF algorithm has the highest value because of the highest control packet rate (0.4 packet/second) with multiple types of packets such as Hello packet, Link State (LS) Acknowledgment packet, LS Update packet, LS State request packet, etc., DV algorithm has the smallest overhead value with a control packet rate value of 0.03. We can see also that the higher the number of node is, the higher the overhead is. So, with a stable overhead, our approach is more scalable than these three others.

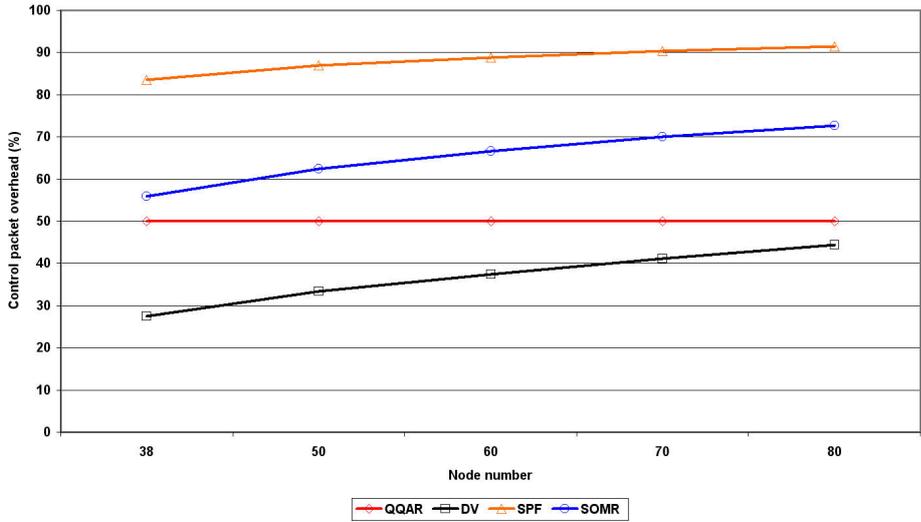


Figure 4.19: Control overhead

#### 4.4.6 Packet delivery ratio

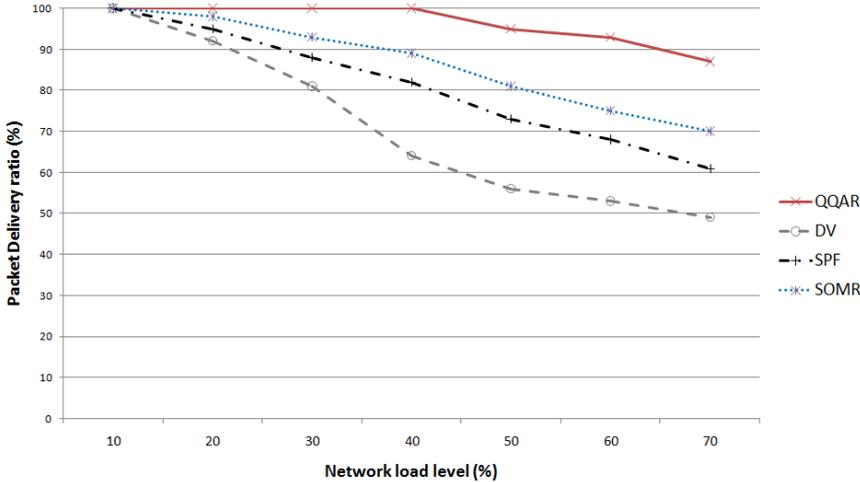


Figure 4.20: Packet delivery ratio

According to packet delivery issue, Fig. 4.20 shows the packet delivery ratio of the algorithms. Packet delivery ratio is the ratio of number of packets received by the destination (client nodes) to the total number of sent packets at server nodes. That is calculated in using the equation 4.4. Generally, the more network is loaded, the more packet delivery ratio decreases. The QQAR gives a good result in keeping the ratio at 100% until the network load level of 30%. The minimal value obtained is 87% at 70% of load level. The packet delivery ratio of the three other algorithms decreases with the

increase of network load level. When network load level reaches 70%, the packet delivery ratios are 87%, 70%, 61% and 49% for QQAR, SOMR, SPF and DV, respectively. So, the QQAR obtains the best result in terms of delivery ratio.

## 4.5 Server selection algorithm: experimental results

To validate the proposal, we used the same methodology as routing experiments. More precisely, first, we tested the selection algorithm in a simulation model to validate the functionality of our proposal. Then, we began implementing a real platform to verify its performance and compare it with other approaches. The two next subsections present the obtained experimental results.

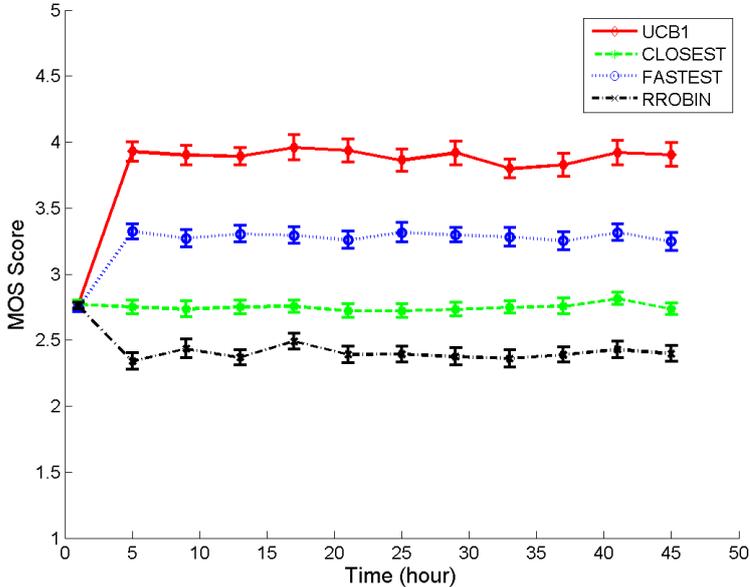
### 4.5.1 Simulation results

We first analyze the obtained simulation results. We compare our approach with the following three server selection policies that have been applied in previous research efforts. As explained later, these approaches are mainly based on the optimization of QoS performance metrics:

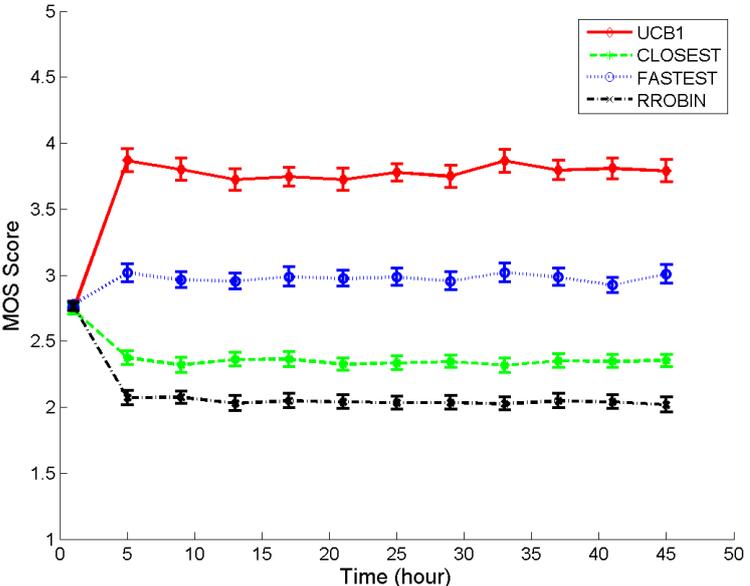
- **FASTEST policy** [Wendell et al., 2010, Han et al., 2009]: The iBox always chooses the server that has a shortest round-trip time (RTT) (i.e., the time it takes for a data packet to be sent and an acknowledgment of that data packet to be received) between a client and a server. The RTT value between client-server pair is updated online.
- **CLOSEST policy** [Pierre and Van Steen, 2006, Guo et al., 2002]: The closest server of the iBox in terms of hop-count (i.e., the number of routers through which data must pass between user and server) is always chosen. The hop-count between client-server pair is updated online.
- **ROUND ROBIN policy**[Ringserver, 2009]: Each iBox has a server list. For a Round Robin policy, the first request is sent to a server picked randomly from the list. For subsequent requests, the iBox follows a circular order fixed before to select the server.

The reason we chose these server selection policies above is that they are widely used server selection algorithms of today's CDN providers. For example, the FASTEST policy is used by Akamai [Akamai, 1998] and NLARN [NLARN, 2006] who is developing a project of server selection based on RTT. The CLOSEST policy is implemented in project Globule [Pierre and Van Steen, 2006] that based on the hop-count to selection the appropriate server. The ROUND ROBIN policy is applied to the Ring Server project.

In chapter 3, we have presented three currently used algorithms for MAB formalization:  $\epsilon$ -greedy, softmax and *Upper Confidence Bounds 1 (UCB1)*. We compared now the performance of them to choose the best one to apply to our proposal.

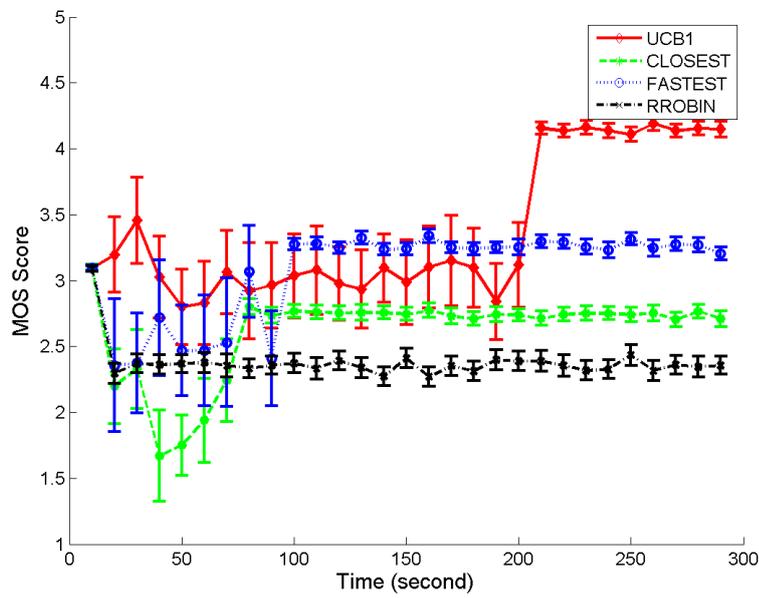


(a) 3AreasNet

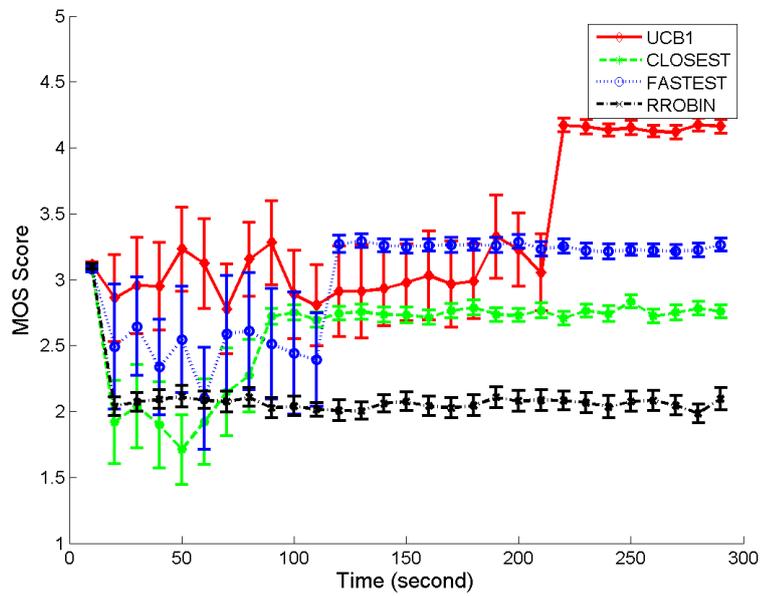


(b) NTTnet

Figure 4.21: User perception results of four policies: UCB1, FASTEST, CLOSEST and Round Robin

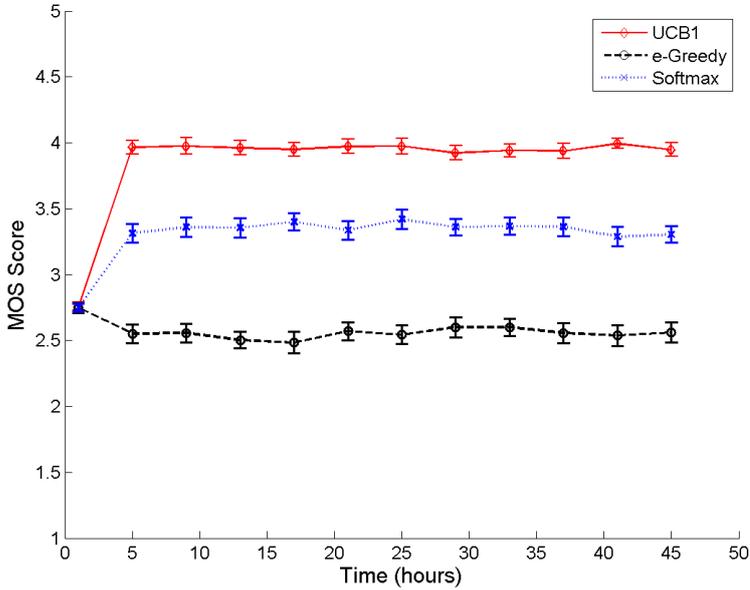


(a) 3AreasNet

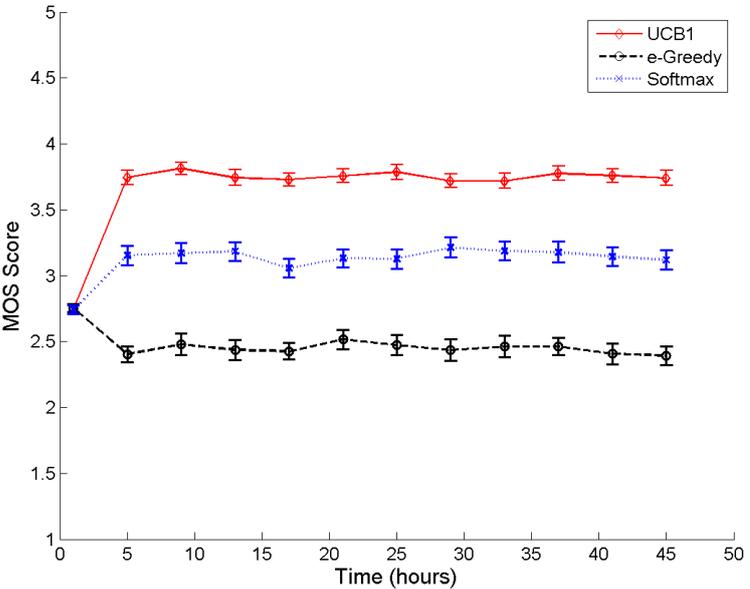


(b) NTTnet

Figure 4.22: Initialization phase



(a) 3AreasNet



(b) NTTnet

Figure 4.23: User perception results of three policies:  $\epsilon$ -greedy, softmax and UCB1

The experiments are launched in the platform presented in chapter 4 in using two network topologies: the *3AreasNet* (Fig. 4.23(a)) and the *NTTnet* (Fig. 4.23(b)). The two figures 4.23(a) and 4.23(b) show that UCB1 is explicitly better than these two policies in term of *MOS* score. The UCB1 reaches mean *MOS* value of 4.0 in Fig. 4.23(a) and 3.7 in Fig. 4.23(b). After the 5th hour, UCB1 becomes increasingly better than the two others. UCB1 gives a *MOS* score that varies between 3.9-4.0 (Fig. 4.23(a)) and 3.7-3.8 (Fig. 4.23(b)). The average *MOS* score of UCB1 is always higher than score of 3.5.

On the other hand, the  $\epsilon$ -greedy gives the worst result with the *MOS* score between 2.4-2.5 in two figures. The Softmax obtains the *MOS* score of 3.3 and 3.1 in figures 4.23(a) and 4.23(b), respectively.

Obviously, the results above show that the algorithm UCB1 outperformed the two other ones in terms of *MOS score*. That motivates us to apply the UCB1 to our server selection method.

In figure 4.21, the UCB1 algorithm has a mean *MOS* value that is greater than *MOS* value of 3.5, while the three others are less than 3.5. For example, in the 28th hour of the simulation on the *3AreasNet* (Fig. 4.21(a)), the obtained *MOS* scores are 3.9, 3.3, 2.7 and 2.4 for UCB1, FASTEST, CLOSEST and Round Robin, respectively (Tab. 4.13 and 4.14).

Table 4.13: Experimental results for 3AreasNet

	UCB1	CLOSEST	FASTEST	RROBIN
Average MOS	3.9	2.7	3.3	2.4
Convergence time (sec.)	210	80	100	-
Improvement of UCB1 (MOS)		144.44%	118.18%	162.5%

Table 4.14: Experimental results for NTTnet

	UCB1	CLOSEST	FASTEST	RROBIN
Average MOS	3.7	2.4	3.0	2.05
Convergence time(sec.)	220	90	120	-
Improvement of UCB1 (MOS)		154.16%	123.33%	180.48%

According to the Round Robin policy, as shown in both figures 4.21(a) and 4.21(b), Round Robin gives the worst result (*MOS* score of 2.4 and 2.05) which results in the selection of inappropriate servers. For example, all servers are chosen periodically even if a server has a long routing path to the user or it is already overloaded.

We now analyze the results of the CLOSEST algorithm which always selects the closest server in terms of hop-count. In many cases, the closest server does not give good result in terms of service quality (e.g., in the case where a server is close to many users). Taking an example in the *3AreasNet* topology (Fig. 6.7(a)), the server *sv47* is close to four users *u53*, *u40*, *u60* and *u54*. In contrast, in the *NTT* topology 6.7(c), the server *sv78* is close to five users *u56*, *u57*, *u58*, *u59* and *u60*. For the CLOSEST

algorithm, all the users select the closest server. This fact overloads the selected server and leads to the degradation of quality. This explains the weak result obtained of the CLOSEST algorithm in Fig. 4.21. The decrease of *MOS* score from 2.7 (Fig. 4.21(a)) down to 2.4 (Fig. 4.21(b)) is explained by the change of topology. The NTTnet topology is based on a real network called NTT where choosing the closest server is not always suitable (in the case of server *sv78*).

The FASTEST algorithm gives a quite good result (3.3 in fig. 4.21(a) and 3.0 in fig. 4.21(b)). This algorithm always selects the server having the shortest round-trip time. As mentioned previously, there are four criteria that influence the service quality: *delay time*, *jitter*, *loss rate* and *conditional loss rate*. Taking into account only one of these three criteria is not enough to give a good QoE.

Results in both figures 4.21(a) and 4.21(b) show that the UCB1 algorithm obtains good QoE feedback with the *MOS* score greater than 3.5 (it is worth noting that the *MOS* score of 3.0 corresponds to Fair quality).

Concerning the convergence time issue, in the first 5 minutes of simulation, we observe *MOS* score results of 4 algorithms every 10 seconds for two topologies (Fig. 4.22). Note that time unit used in both figures 4.22(a) and 4.22(b) is in “second”. We observe that 4 policies have a common initialization period where the policies do not have any information about the system. We call this period the *initialization phase*. As each algorithm has a particular policy, the convergence time of each algorithm is different.

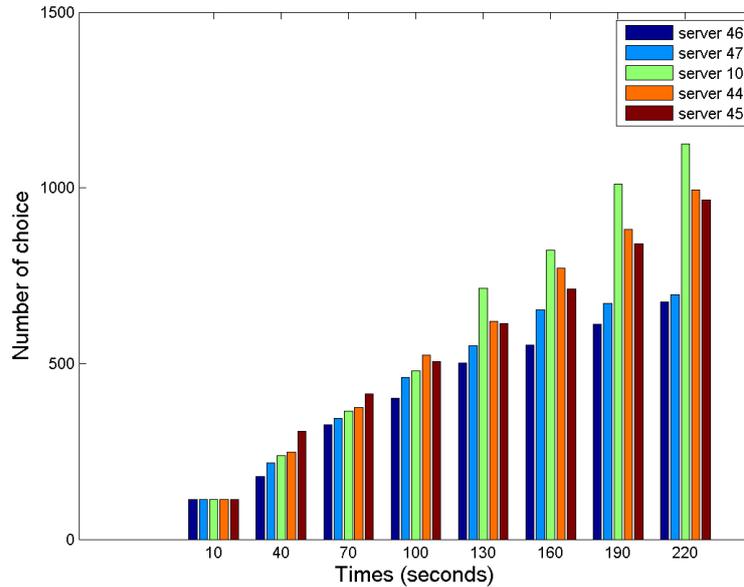


Figure 4.24: Number of choice of iBox 55

For the 3AreasNet network (Fig. 4.22(a)), after about 210 seconds, the UCB1 is stable with a *MOS* score of 3.9. The convergence times of FASTEST and CLOSEST

are 100 seconds and 80 seconds respectively. To explain the strong fluctuation in this period, we observed the UCB1 in the first 220 seconds (Fig. 4.24). The figure 4.24 shows the number of choices of one iBox in our experiment (iBox 55) for the first 220 seconds in the 3AreasNet topology. In this case (Fig. 6.7(a)), the minimum hop-counts from iBox55 to the five servers *sv45*, *sv44*, *sv10*, *sv47*, *sv46* are 4, 5, 6, 6, 7, respectively. In the first 70 seconds, *iBox55* has mainly chosen the *server45* due to the fact that *server45* is the closest one. Then, from the 100th second, *server10* is the most chosen one. The overloading of *server45* caused by choices of *iBox56*, *iBox51* and *iBox48* explains this selection switch. In addition, the number of choice of *server47* and *server46* decrease because the data flows have passed through the link *8-13*. The latter is highly overloaded because it is the connected link between two zones. This kind of adaptive behavior possibly explain the results obtained of our proposed method.

For the NTTnet network (Fig. 4.22(b)), UCB1 reaches the *MOS* score of 3.7 after 220 seconds. The convergence time of FASTEST is 120 seconds and its *MOS* score is 3.0. CLOSEST algorithm takes 90 seconds respectively during the initialization phase.

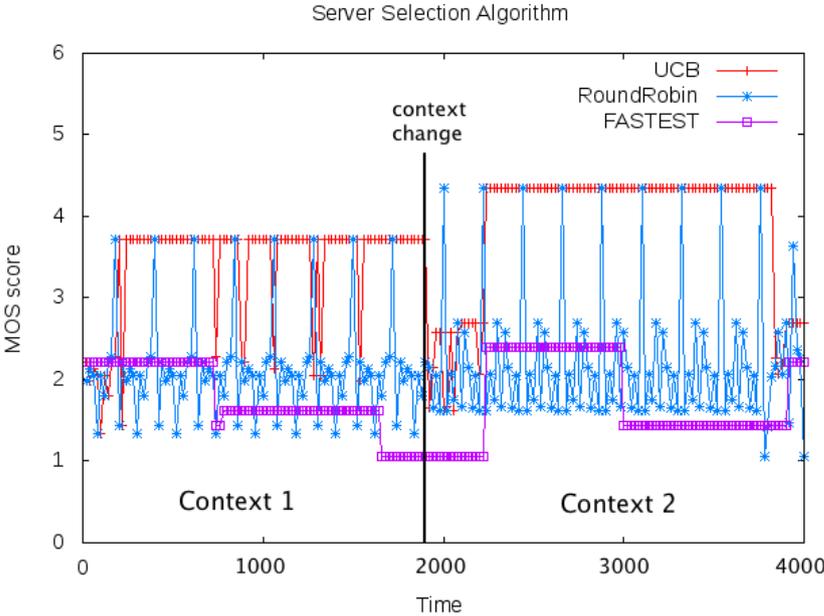
#### 4.5.2 Real platform results

We now analyze the results obtained of the real platform. As mentioned in section 4.3.1, we used a video of 4000 seconds. The server selection module is executed every 20 seconds. So, we divided the video into 200 tracks, each one lasting 20 seconds. To evaluate the robustness of the proposed approach when the network environment changes, we made a change in the network context at about the 2000th second for each test (we marked a vertical line in two figures 4.25(a) and 4.25(b)). Making a change in the network context is achieved by changing the network parameters for each server. We give here an example in Tab. 4.15 that shows the network parameters for eleven servers (from *sv1* to *sv11*) in two contexts. The three numbers in each box are *jitter*, *loss-rate* and *loss-rate conditional*. As shown in Tab. 4.15, the parameters are changed in passing from *Context 1* to *Context 2*. The figure 4.25 shows the results of two executions of the test.

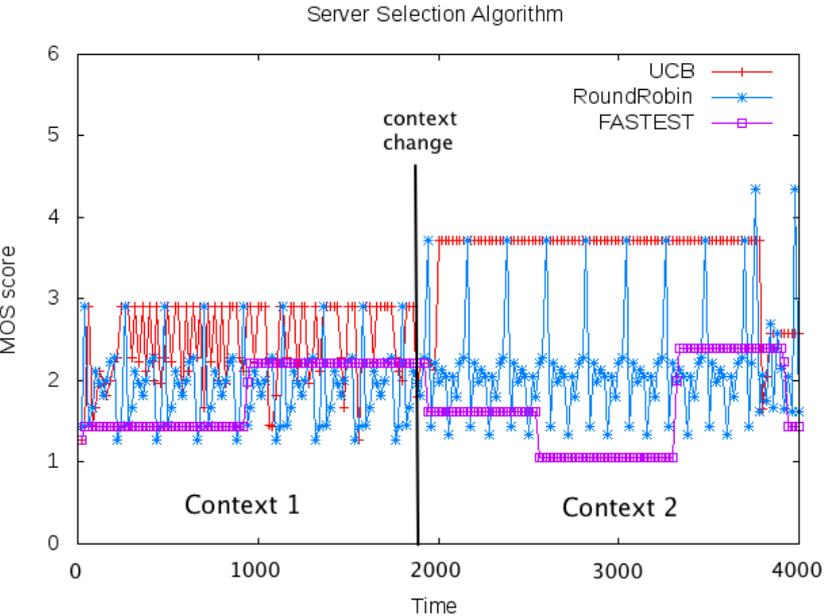
Table 4.15: Example of change of network parameters

Context	sv1	sv2	sv3	sv4	sv5	sv6	sv7	sv8	sv9	sv10	sv11
1	6 3 18	10 3 20	4 5 10	15 2 50	20 5 10	12 4 25	9 5 16	7 4 32	8 3 27	7 4 28	6 3 31
2	4 3 15	10 1 20	8 4 25	5 3 18	0 2 10	7 3 15	6 5 24	7 4 18	6 4 32	7 5 35	2 4 19

In the figure 4.25(a), the UCB1 algorithm takes about 200 seconds, from 0th to 200th second, to converge to the *MOS* score of 3.7. In the figure 4.25(b), this initialization phase is from 0th to 220th second and the converged *MOS* value is 2.91. We can observe that during this first period (marked “Context 1” in the two figures), the values 3.7 and 2.91 are the best obtained values (among 11 servers) in the figures 4.25(a) and 4.25(b) respectively. Due to the exploration task, UCB1 chooses sometimes another server at 740th, 880th, 900th, 1060th, 1280th ... second in figure 4.25(a) and at 300th, 340th, 380th, 420th ... second. In other words, UCB1 does not always choose the server giving the best result to guarantee the exploration-exploitation trade-off. The frequency of



(a)



(b)

Figure 4.25: Experimental results over the real platform

switching in figure 4.25(b) is higher than the one in figure 4.25(a). This is explained by the fact that the maximum value obtained in figure 4.25(a) is greater than the value obtained in figure 4.25(b)(i.e.,  $3.7 > 2.91$ ).

In the case of the context change (the vertical line with the text “*context change*”), the UCB1 algorithm takes 320 seconds (from 1920th to 2240th second) in the figure 4.25(a) and 240 seconds (from 1760th to 2000th second) in the figure 4.25(b) for the adaptation phase. In the second context (marked “Context 2” in two figures), the converged *MOS* value is 4.34 and 3.71 shown in figures 4.25(a) and 4.25(b) respectively. As shown in both figures, there is no switching task from 2240th to 3820th second (figure 4.25(a)) and from 2000th to 3780th second (figure 4.25(b)) because these two values (3.71 and 4.34) are already good in the scale of *MOS* score of 1 to 5 for video quality at the client side.

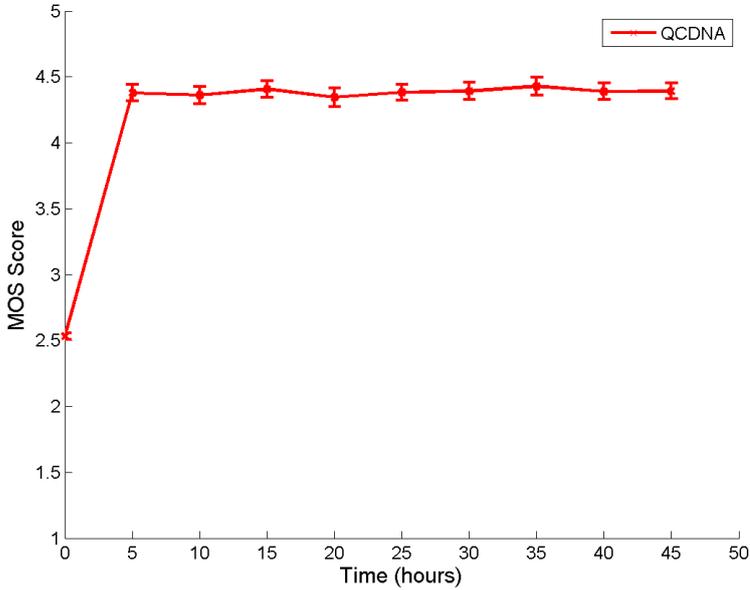
For the RoundRobin and the FASTEST algorithms: the RoundRobin algorithm follows a round-robin rule in choosing the servers. On the other hand, the FASTEST algorithm chooses the server that gives the shortest delay. We can see that the fastest one is not the best one because the video quality does not depend only on the delay parameter.

These experimental results demonstrate that our proposed selection policy based on MAB formalization explicitly optimizes the QoE result. The results obtained also show that the proposed adaptive control model (fig. 1) gives a better QoE result (the *MOS score*) than traditional model based on QoS metrics only.

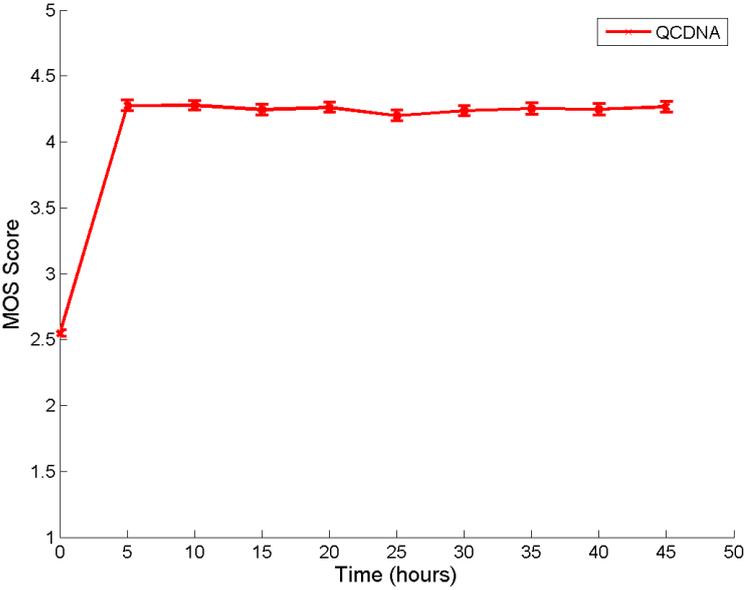
## 4.6 QQAR and proposed server selection: experimental results

In the last two subsections, we tested separately our two proposals: the proposed routing algorithm QQAR in the routing layer and the proposed MAB-formalized server algorithm in the meta-routing layer. The obtained results showed that our proposals yielded better results than traditional approaches. Therefore, it is now time to combine the two proposed approaches for a complete QoE-based CDN Architecture, called QCDNA. Generally, QCDNA has also two layers: the routing layer and the meta-routing layer. For the server selection module of the meta-routing layer, we apply the proposed server selection scheme as well as selection algorithm presented in Chapter 3. This module chooses the most appropriate replica server in term of QoE. Then, for the routing layer, we apply the proposed routing algorithm QQAR (Chapter 2). This layer takes into account the routing process between the source (end-user) and the determined destination (selected server).

To validate the proposed QCDNA, we tested it in the same scenario as previous experiments including the highload scenario and incident scenario (section 4.3.2.2) to stress the network out. The used network topology is NTTnet. For each topology, we implement two types of video: the SD Video and the HD Video. The experiment lasts 45 hours. We collect the *MOS* score every 5 hours. As shown in figure 4.26, QCDNA reaches the average *MOS* score of 4.4 and 4.3 for SD Video and HD Video,



(a) Using SD video



(b) Using HD video

Figure 4.26: MOS score of QCDNA for 45 hours of simulation

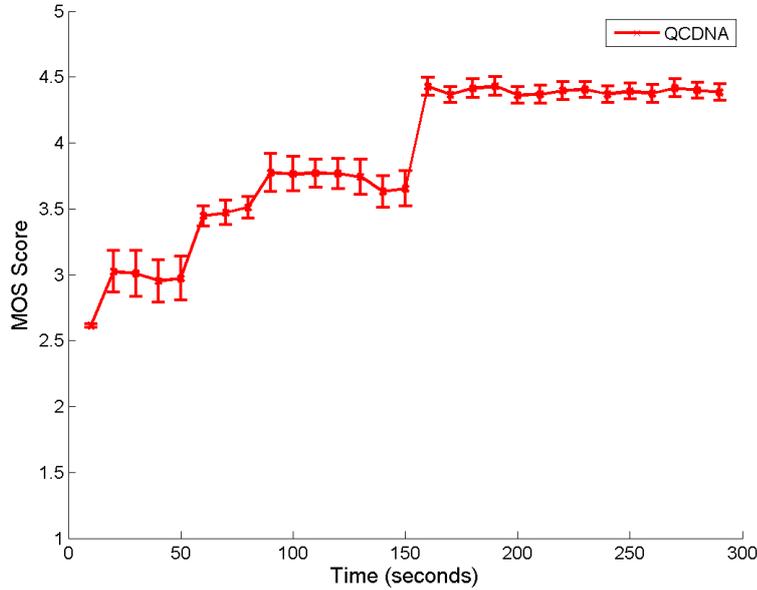


Figure 4.27: MOS score of QCDNA for 5 minutes of initialization phase

respectively. According to the convergence time, figure 4.27 shows the obtained MOS scores of QCDNA in the first five minutes of simulation. The MOS scores are taken every 10 seconds. The MOS score of QCDNA begins to obtain a stable value of 4.4 after 160 seconds.

These good results confirm the correctness of our two proposals above. The combination of our QoE-based algorithms in two layers, the routing layer and the meta-routing layer, gives a good experimental result in terms of user’s perception, the QoE.

## 4.7 Conclusion

In this chapter, we have presented three main parts. First, we presented in detail the testbed implemented for measuring the QoE value. The measurement method is presented in appendix A. The testbed is used to construct a real dataset for the input data of supervised learning of the measurement method. Concretely, the goal is to collect the MOS score given by real human in watching the video with the quality changed by NetEm tool, which controls the network parameters to generate different network configurations. Therefore, we reach a dataset with the MOS scores given subjectively corresponding to different network configurations. This helps supervised learning of QoE measurement method.

Then, we described the experimental setup including both real platform and simulation setup. We cited in detail the setting parameters and the experimental scenarios. The models are illustrated clearly by figures. This setup description is the basic premise for

experimental results obtained. Finally, we combine our two proposals in a CDN model where the routing algorithm is applied to routing layer and server selection scheme is applied to meta-routing layer.

The experimental results showed that our proposals yielded better results than traditional ones. More precisely, the proposed routing algorithm, QQAR, gave better QoE result than DV, SPF and SOMR algorithm. On the issue of convergence time, the QQAR converges faster than DV and SOMR. Although QQAR convergences slower than SPF algorithm, which is known for short convergence time, the convergence time of QQAR is acceptable for a routing protocol. We tested also the fault tolerance in making topology change. The algorithms are tested under different network load levels. The results showed that the QQAR gives better MOS score under any load level.

With regard to the proposed server selection algorithm, the results in both simulation model and real platform show that this new approach is better than widely used methods in terms of MOS score and convergence time. In addition, the experimental diagram shows that our selection algorithm is effective in balancing the exploitation-exploration trade-off. This approach is an appropriate formalism that can efficiently optimize the QoE while other implemented baseline approaches, mainly based on linear combinations of QoS parameters, are clearly sub-optimal in this context.

Finally, to confirm these two experimental results above, we combined our two proposals in a CDN architecture, called QCDNA. The obtained results showed that QCDNA gave good MOS scores in both kinds of video streaming (SD and HD video). Furthermore, the experimental results in the initialization phase show that the convergence time of QCDNA is acceptable for a CDN model.

## Chapter 5

# Conclusion

Network services have grown increasingly over the past several years with variety of high quality applications such as IPTV, VoIP, Videoconferencing, Telepresence, online games, etc. Facing such rapid development, the widely used QoS framework, which is defined as aspect ensuring service quality of network traffic, is not suitable enough. In fact, the traditional QoS networking based on the technical parameters (e.g., delay, jitter, bandwidth, loss-rate, etc.), which are too trivial to be competitive factors between service providers in today's Internet. Actually, a traffic flow delivered with excellent QoS metrics may risk delay experience of a video-conference client or an echo of voice of a VoIP user. In addition, satisfying such set of constraints risks meeting problem of complexity due to the fact that a QoS model with more than two non-correlate criteria meets NP-complete problem. All service operators and providers gradually know that the most important goal is to satisfy end-users who directly "pay" for these services. Ignoring user's perception risks client dissatisfaction, churn and lost revenue. Therefore, satisfying end-users becomes today a new trend for network management systems. Service providers need a new criterion that refers the user experience in order to improve the traditional QoS parameters. In this thesis, we showed the inefficiency of QoS-based algorithms for user satisfaction optimization. As this latter is, by definition, directly related to end-users, it seems to be reasonable to describe it as a non-linear function of the QoS parameters [Tran et al., 2011]. In addition, the QoS is important but not determinant in order to realize the mechanisms allowing to obtain and optimize the user satisfaction that is the important goal of the operators. Consequently, Quality of Experience has been introduced to meet this request. This new notion is defined as levels of user's satisfaction when using the network services. Unlike QoS networking, which focuses on just technical parameters, QoE networking navigates the target to end-users in ensuring an acceptable experience level. More precisely, Internet today, mostly the Content-Oriented Network (CoN), needs an efficient end-to-end QoE model to assure the satisfaction of end-users. An E2E QoE model is able to guaranty the user satisfaction between network hosts. Proactive monitoring enables the system to detect in real-time degradation of the QoE and to react adaptively to avoid the system's fall.

## 5.1 Contributions

To meet the requirement mentioned above, we proposed an E2E QoE model and applied it to a Content Distribution Network (CDN) architecture. CDN is an effective approach to improve the service quality of Internet in replicating the content from the place of origin to the replica servers scattered over the network system. So, the user's requests are served by replica servers close to user's place. Despite this, a CDN service should guarantee an E2E QoE to satisfy end-user who plays a key role in network management. This fact motivates us to apply the proposed E2E QoE model to the CDN architecture in focusing on two main layers: the routing layer and the meta-routing layer. We summarize our contributions as follows:

- We proposed an end-to-end model based on QoE with an adaptive control including three modules: QoE measurement, control and command. The QoE measurement evaluates the QoE in basing on the user's feedback. On his turn, the control module takes the QoE value evaluated to make decision in choosing an appropriate action. Finally, the command module applies this chosen action to the environment. This adaptive loop makes the system become an evolutionary system, which operates adaptively and infinitely in order to maintain a sufficient QoE level of entire system. The model is also able to take into account the user's perception and network's changes. The goal of the adaptive control to optimize the QoE level at the user side.
- We applied the E2E QoE model to a CDN architecture. We chosen CDN architecture for the following reasons. First, there are many CDN providers in today's Internet. They have different policies to provide services to customers with different network applications with high availability and high performance. Secondly, CDN is an user-oriented network service that refers to QoE model. The goal is to guarantee a good QoE level of CDN service. Concretely, we focused on their two main layers: *routing* and *meta-routing*. In each layer, an E2E QoE model is proposed to ensure the user's experience.
- In chapter 2, for the routing layer, we proposed a routing algorithm, called QQAR, which is based on the QoE. Our algorithm used Q-Learning, an algorithm of Reinforcement Learning (RL). The latter is defined as learning a routing policy in mapping observations of feedback into selecting paths actions. Our idea of applying Q-learning, an algorithm of RL, is to use QoE as user's feedback to improve the quality of exploration in routing process for adaptive packet routing in communication networks. Each node based on the QoE value, stored in Q-tables, and used the QQAR algorithm for routing packets in order to optimize the QoE in the whole traffic. After testing in different network load level, the experiments showed indisputably that our approach yielded better result than traditional routing algorithms in terms of: MOS score, convergence time, fault tolerance capacity, packet delivery ratio.

- In chapter 3, for the meta-routing layer, we proposed a novel server selection algorithm that was formalized by the Multi-armed bandit formalization. We consider the user's feedback as rewards of the system. We chosen MAB to formalize our server selection method due to three main reasons. First, MAB does not make any assumption of the reward function. It just take into account the exploration-exploitation trade-off. Secondly, MAB makes no heuristic. Unlike other classic approaches optimizing a concrete heuristic such as E2E delay, hop count, etc., MAB explicitly minimizes the regret. For the third reason, among the approaches capable of solving the MDP (Markov Decision Processes), there is only the MAB approach that needs only one state. The future action is chosen based on the obtained reward and this unique state. This property is suitable for the server selection issue of a CDN service. In order to validate the proposal, we tested it on a real platform and a simulation model. The obtained results of both testing platforms showed that the proposed selection algorithm gave better QoE value than commonly used approaches.
- As presented in appendix A, we implemented a QoE measurement method based on the PSQA tool. The latter uses a Random Neuron Network (RNN) for supervised learning method. This RNN takes the dataset of the network configurations (i.e. parameters such as delay, jitter, loss-rate, etc.) and MOS value scored by real user as input. Then, it will give a function as output. This function is used to calculate the MOS score from network parameter values (configurations). We chosen this measurement method because it is considered as a hybrid of subjective and objective methods. The *subjective* idea is the scores given by real user for the dataset. On the other hand, the *objective* idea is the function of output that is used to calculate automatically the MOS score from network parameters.

## 5.2 Perspective

Some of our future research directions include:

- In this thesis, for the meta-routing layer, we focused only on the server selection module. For the future works, we will research this layer to propose a whole meta-routing layer including *server placement*, *cache organization* and *server selection* modules. For this purpose, besides the server selection module, we will focus also on two modules of the preparation phase as follows. First, for the *server placement* module, we try to propose a novel approach to place the servers in the optimized way to minimize the average content access latency and the network bandwidth consumption and improve the user's perception (QoE). There were different algorithms for server placement module [Jamin et al., 2000] [Qiu et al., 2001] [Krishnan et al., 2000] but they focused only on QoS parameters, mostly the latency and hop-count, without considering the user's perception. That motivates us to research on a novel QoE-based server placement method. Secondly, in a similar way for the *cache organization* module, we will try to propose a QoE-based

cache organization with novel caching techniques and update techniques. The aim is to satisfying end-users in guarantying the availability, freshness and reliability of content.

- Improving the QoE measurement in adding more of samples of human score for the input of supervised learning. For the supervised learning, the more we increase the number of trials (samples), the more obtained MOS score is accurate. To accomplish this, we will develop a *add-on tool* that will be attached to a video-sharing website like Youtube so that any users who have Internet connexion can score the perceived video quality after watching it. This method allows us to expand the scale of our testbed in order to increase the volume of the dataset.
- Comparing the proposed routing algorithm with different approaches. Besides three types of routing algorithm we used in this thesis, we are implementing other kinds of routing algorithms that take into account the QoE value at end-users.
- Improving the proposed server selection algorithm in taking into consideration another algorithms of MAB formalization. In other words, besides the UCB1, we will implement another algorithms of MAB formalization for the server selection issue.

## Chapter 6

# Version française abrégée

### Introduction

Le premier chapitre a pour objet d'introduire le contexte général ainsi que les objectifs et enjeux des travaux menés dans le cadre de cette thèse. Dans un premier temps, nous nous positionnons par rapport aux travaux existants reposant sur la notion de Qualité de Service (QoS) en mettant en avant les enjeux de la Qualité d'Expérience (QdE). Nous décrivons ensuite la nécessité d'introduire dans tout système de contrôle/commande une boucle adaptative auto-régulée par un retour issu de la qualité perçue par l'utilisateur. Enfin, nous détaillons les contributions de cette thèse.

L'approche traditionnelle consiste à optimiser les paramètres techniques de QoS tout au long du chemin entre un client et un serveur, on parle alors de paramètres de QoS de bout-en-bout. Ce type d'optimisation pose deux difficultés majeures. Elle suppose une connaissance précise de l'ensemble des ressources disponibles sur le réseau. Cette hypothèse n'est souvent pas soutenable car ces informations peuvent être imprécises, insuffisantes et extrêmement variables. Par ailleurs, cette optimisation doit considérer plusieurs paramètres de QoS comme le délai, le taux de perte ou la bande passante. Or, cette optimisation devient un problème NP-complet [Wang and Crowcroft, 1996] [Michael and Johnson, 1979] lorsqu'au moins deux de ces critères ne sont pas corrélés.

Le paradigme de QdE peut offrir une approche alternative contournant les contraintes mentionnées ci-dessus. La QdE s'intéresse directement à la satisfaction de l'utilisateur. Dans cette thèse, nous nous concentrons sur l'intégration de la QdE dans le mécanisme de contrôle/commande pour rendre la prise de décision adaptative. Plus précisément, nous appliquons le système proposé à une architecture de réseau de diffusion de contenus (Content Distribution Network, CDN) qui est composée principalement de deux niveaux: le routage et le méta-routage.

Les principales contributions de cette thèse sont donc les suivantes:

1. *Une architecture de réseau de diffusion de contenus contrôlée par la QdE.* Ce système prend en compte le retour des utilisateurs finaux pour adapter les fonctions

de routage et de méta-routage.

2. *Une méthode d'estimation de la QdE.* Nous avons mis en place un banc d'essai réel pour créer une méthode de mesure basée sur une quantification objective et une perception subjective.
3. *Un algorithme de routage basé sur la QdE.* Cet algorithme utilise l'apprentissage par renforcement. Nous l'avons mis en œuvre dans l'architecture de CDN que nous proposons.
4. *Un algorithme de sélection de serveurs basé sur la QdE.* Cette méthode est formalisée comme un problème du bandit multi-bras.
5. *L'intégration des algorithmes de routage et de sélection de serveurs.* Cette intégration a donné lieu à une évaluation dans une architecture CDN.

## État de l'art de la QdE et des CDNs

Le chapitre 1 présente le cadre général de cette thèse. Il décrit les travaux existants autour de la notion de QdE ainsi que l'architecture et le fonctionnement général d'un réseau de diffusion de contenus (CDN). Ce chapitre motive le choix d'un CDN comme architecture d'accueil de notre modèle de notre approche de contrôle/commande auto-régulée par la QdE.

### Qualité d'Expérience

La QdE est définie comme une métrique subjective de la performance d'un système. Elle s'appuie sur l'opinion humaine et diffère de la QdS qui représente des paramètres techniques du réseau. Afin de clarifier la notion de QdE, plusieurs organisations de normalisation étudient ce concept comme l'Union Internationale des Télécommunications (ITU), le Broadband-Forum et le TeleManagement-Forum. Elles ont comme objectif de proposer une définition universelle de la QdS et la QdE. En outre, elles tentent également d'établir une relation directe entre la QdS et la QdE afin de rendre cette dernière aisément mesurable.

Pour notre part, nous proposons de classifier la notion de QdE en trois catégories: *La QdE comme une extension de la QdS, la QdE comme paramètre de gestion d'un système et la QdE comme un concept hédoniste.*

De plus, la QdE de bout-en-bout estime le degré de satisfaction des services perçus subjectivement par l'utilisateur final. Cette satisfaction est devenue un enjeu majeur pour les opérateurs de télécommunication. Dans le cadre des services en ligne, la QdE de bout-en-bout est influencée par plusieurs paramètres : le contenu, le réseau, les applications et le contexte d'utilisation. D'autres paramètres externes comprennent le matériel terminal de l'utilisateur, la mobilité et l'importance de l'application. L'amélioration des métriques techniques comme les paramètres de QdS n'est pas toujours satisfaisante pour les utilisateurs. Tous les opérateurs s'orientent de plus en plus vers l'utilisation de

la QdE pour être plus réactifs aux changements de perception des utilisateurs et ainsi obtenir un avantage concurrentiel significatif.

Même si la QdE est difficile à prédire en raison de sa nature subjective, elle est indispensable pour évaluer la performance du service. La mesure de la QdE peut être distinguée en deux types de méthodes : subjectives et objectives. Les méthodes subjectives se basent sur la collecte d'un avis sous la forme d'un score appelé le score moyen d'opinion (Mean Opinion Score - MOS). Pour ce type d'évaluation, chaque utilisateur évalue le service sur une échelle de cinq mesures : 1 = mauvais, 2 = médiocre, 3 = moyen, 4 = bon, et 5 = excellent. Cependant, la mesure subjective de la QdE prend du temps, est fastidieuse et coûteuse. De plus, elle n'est pas applicable dans un environnement dynamique. Au lieu de recueillir directement des valeurs de QdE en provenance de l'utilisateur, la QdE peut être obtenue de manière non intrusive à partir de méthodes objectives ne nécessitant pas l'intervention de l'utilisateur. Ces méthodes permettent d'estimer la valeur de la QdE en temps réel. Citons quelques exemples de ces méthodes objectives : PSNR (Peak Signal to Noise Ratio), SSIM (Structural SIMilarity), PEVQ (Perceptual Evaluation of Video Quality), PESQ (Perceptual Evaluation of Speech Quality) et Media Delivery Index (MDI).

### Réseau de diffusion de contenus

Aujourd'hui, le besoin d'une solution efficace pour la distribution de contenus volumineux et à forte demande est présent chez tous les opérateurs, l'objectif étant d'améliorer la performance du réseau et de fournir une meilleure tolérance aux pannes. Les mécanismes de fonctionnement d'un service offert par un réseau de diffusion de contenus (CDN) sont représentés dans la figure 6.1. Un opérateur de service de vidéo à la demande publie par exemple certains de ses contenus auprès d'un autre opérateur appelé fournisseur CDN. Ce dernier réplique ensuite ces contenus sur ses serveurs distribués dans diverses zones géographiques.

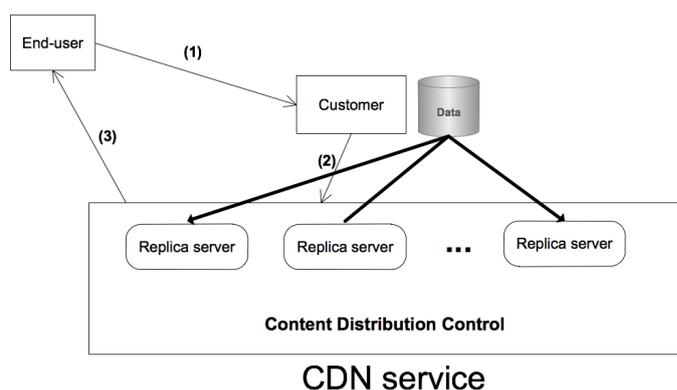


Figure 6.1: Le service de CDN

Par ailleurs, une architecture de type CDN comprend deux couches principales : le

routage et le méta-routage.

- *La couche de routage.* Les composants du réseau de cœur prennent en compte le processus de routage en fournissant la connectivité de réseau et la QoS garantie pour le trafic. Plusieurs approches de routage intégrant la QoS ont été proposées par le passé. Nous retrouvons celles basées sur le principe de la commutation (MPLS, etc.) [Stallings, 2001] [Partridge, 1992], celles dérivées des protocoles classiques de routage (QOSPF, etc.) [Rajagopalan et al., 1998], celles utilisant plusieurs métriques [Wang and Crowcroft, 1996], Widest-Shortest Path [Apostolopoulos and Orda, 1999], Shortest-Widest Path [Guerin et al., 1997], TAMCRA [Kuipers and Van Mieghem, 2005], Approximation lagrangienne [Jaffe, 1984], et enfin celles basées sur l'apprentissage automatique : Q-Routing [Boyan and Littman, 1994], CQ-Routing [Kumar and Miikkulainen, 1998], CPN [Gelenbe and Lent, 2004], AntNet [Di Caro and Dorigo, 2004] et KOQRA [Mellouk et al., 2009].
- *La couche de méta-routage.* Cette couche prend en compte certains mécanismes comme l'emplacement du serveur, l'organisation du cache et la sélection du serveur. L'emplacement du serveur vise à minimiser la latence moyenne d'accès au contenu ainsi que la consommation de bande passante. Le module s'occupant de l'organisation du cache est basé sur des techniques de mémorisation [Rousskov and Wessels, 1998] [Gadde et al., 1997] [Karger et al., 1999] et de mise à jour du cache. L'objectif est de garantir la disponibilité, la fraîcheur et la fiabilité du contenu. Le troisième module, la sélection de serveurs, comprend deux types de sélection: la sélection adaptative [Pierre and Van Steen, 2006] [Karger et al., 1999] et la sélection non-adaptative [Szymaniak et al., 2003] [Delgadillo, 1999]. Dans cette thèse, nous nous concentrons sur le module se chargeant de la sélection de serveurs.

## Algorithme de routage proposé pour un CDN : le QQAR

Dans le chapitre 2, nous présentons notre première contribution qui consiste dans un algorithme itératif de routage qui se base sur l'apprentissage par renforcement (AR) et qui considère les mesures issues de la QdE dans la recherche de chemins.

Des algorithmes d'AR rencontrent les mêmes problèmes que les algorithmes traditionnels distribués, mais avec quelques contraintes supplémentaires. Tout d'abord, l'environnement est modélisé stochastiquement (en particulier les liens, les coûts de liaison, le trafic et la congestion), de sorte que les algorithmes de routage peuvent prendre en compte la dynamique du réseau. Toutefois, aucun modèle dynamique n'est donné a priori. Cela signifie que les algorithmes d'AR doivent échantillonner, estimer et peut-être construire des modèles de caractéristiques pertinentes de l'environnement. Par ailleurs, dans certains cas, l'approche d'AR, contrairement à d'autres algorithmes d'apprentissage automatique, n'a pas de phase d'apprentissage explicite suivie d'une évaluation. L'AR,

considéré comme une approche réactive, consiste à permettre à un agent autonome l'exécution de la tâche souhaitée dans un environnement de réseau très dynamique. La politique doit être définie en fonction de la boucle de perception - décision - action.

En nous basant sur l'AR, nous avons proposé un algorithme de routage, appelé QQAR (QLearning QoE-Adaptive Routing), notre objectif étant de maintenir un bon niveau de QdE de l'utilisateur sur les routes empruntées par le trafic de ce dernier. Dans QQAR, nous considérons chaque routeur comme un état discret. Les états sont disposés tout au long de la route entre le client et le serveur choisi dans le CDN.

Le mécanisme de routage proposé est illustré dans un diagramme de la figure 6.2 et s'opère selon les étapes suivantes :

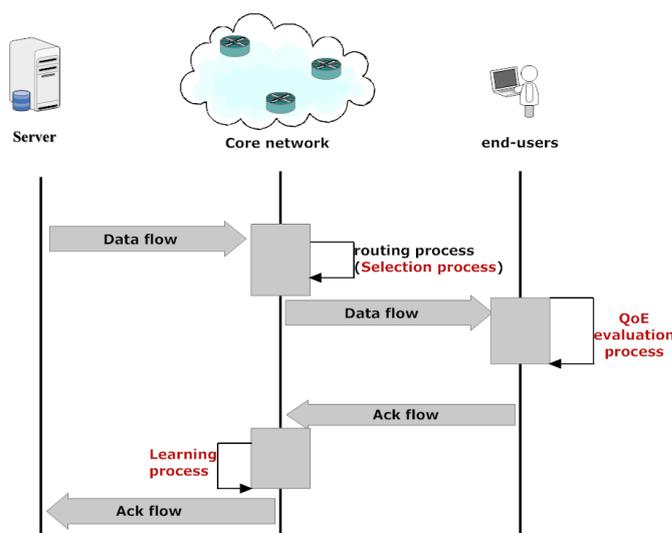


Figure 6.2: L'algorithme de routage QQAR

- **Première étape** - Flux usager : le serveur, choisi par la couche de méta-routage, envoie un paquet à l'utilisateur final. Après la réception de ce paquet, chaque nœud situé sur la route empruntée évalue la QdE et transmet le paquet en utilisant un processus de sélection.
- **Deuxième étape** - Du côté de l'utilisateur : Une fois les données arrivées aux utilisateurs finaux, un processus d'évaluation de la QdE est réalisé. Un retour de QdE est renvoyé sous la forme d'un accusé de réception tout au long de la route.
- **Troisième étape** - Les accusés de réception : Après qu'un nœud situé sur la route ait reçu l'accusé de réception sur un lien donné, il met à jour la Q-valeur de ce lien.

Concernant le processus d'apprentissage de QQAR, nous proposons une nouvelle méthode pour l'exploration du réseau. Cette technique repose sur un mécanisme probabiliste d'exploration. Concrètement, après avoir reçu le message contenant l'accusé de

réception, le nœud  $x$  met à jour la Q-valeur du lien  $xy$  en utilisant la formule 6.1 (Fig. 6.3) :

$$\underbrace{Q_{xy}}_{\text{nouvelle valeur}} = \underbrace{Q_{xy}}_{\text{ancienne valeur}} + \alpha \left[ \underbrace{\beta(q_y - q_x) + \gamma \max_i Q_{yz_i}}_{\text{nouvelle estimation}} - \underbrace{Q_{xy}}_{\text{ancienne valeur}} \right] \quad (6.1)$$

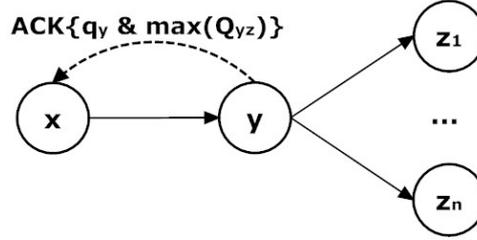


Figure 6.3: Le processus d'apprentissage

où  $Q_{xy}$  et  $Q_{yz_i}$  sont les Q-valeurs associés aux liens  $xy$  et  $yz_i$ .  $q_x$  et  $q_y$  sont des valeurs de QdE évaluées sur la base du MOS par la méthode d'estimation de la QdE au niveau des nœuds  $x$  et  $y$ .  $\alpha$  est le taux d'apprentissage.

Concernant le processus de sélection de chaque nœud après la réception d'un paquet, une solution de compromis entre les phases exploration et exploitation a été mise en œuvre. Il s'agit d'une méthode qui fait varier les probabilités d'action en fonction de la valeur estimée. La meilleure action est celle qui dispose de la plus grande probabilité de sélection. Toutes les actions sont pondérées en fonction de leur valeur Q conformément à la méthode *softmax* qui utilise la distribution de Boltzmann [Sutton and Barto, 1998].

Avec ce type de sélection, après avoir reçu un paquet, le nœud  $x$  choisit un voisin  $y_k$  parmi ses  $n$  voisins  $y_i$  ( $i = 1..n$ ) avec la probabilité exprimée par l'équation 6.2 pour transmettre le paquet :

$$p_{xy_k} = \frac{e^{\frac{Q_{xy_k}}{\tau}}}{\sum_{i=1}^n e^{\frac{Q_{xy_i}}{\tau}}} \quad (1 \leq k \leq n) \quad (6.2)$$

## Algorithme de sélection de serveurs proposé

Le chapitre 3 présente notre deuxième contribution qui porte sur une nouvelle méthode de sélection de serveurs inspirée par le problème dit de "Bandit Multi-Bras (MAB)".

La formalisation du problème Bandit Multi-Bras [Vermorel and Mohri, 2005] [Antos et al., 2008] s'inspire de la machine à sous traditionnelle. Après avoir tiré un bras, ce dernier offre une récompense tirée d'une distribution associée à ce bras. Le but du joueur est de maximiser les récompenses cumulées sur un nombre fixe de tirage.

Tous les algorithmes proposés pour le problème MAB ont tendance à minimiser *le regret*, qui est défini par la différence entre la somme de récompenses associées à une stratégie optimale et celle des récompenses obtenues.

Notre choix pour une telle formalisation est motivé par les raisons suivantes:

- La formalisation MAB ne fait aucune supposition sur la fonction de récompense. Cette approche prend en charge le compromis exploration-exploitation pour tout type de fonction de récompense. Pour notre travail, la fonction de récompense considérée est le MOS.
- Parmi les approches susceptibles de résoudre le Processus de Décision Markovien (MDP) inhérent à l'approche AR utilisé dans le routage, MAB ne nécessite qu'un seul état. L'action est choisie en fonction de la récompense obtenue pour unique cet état. Cette propriété est appropriée pour notre problème de sélection de serveurs du fait que seul l'état courant de chaque serveur est connu.

Afin de maximiser le gain dans MAB, nous nous sommes intéressés à trois méthodes d'exploration différentes :  *$\epsilon$ -greedy*, *softmax* and *upper confidence bounds*.  *$\epsilon$ -greedy* est l'algorithme le plus simple, il consiste à choisir aléatoirement une action avec la probabilité  $\epsilon$  et à prendre la meilleure action avec la probabilité  $(1 - \epsilon)$ . Concernant Softmax, il s'agit d'un algorithme qui utilise une fonction attribuant la récompense moyenne aux probabilités d'action. Cet algorithme permet d'explorer les actions moins utilisées à un moment donné, et met aussi l'accent sur les meilleures actions en termes d'espérance des gains.

Contrairement aux deux algorithmes ci-dessus, l'algorithme UCB1 est lié à une politique basée sur un indice qui est défini comme la somme de la récompense moyenne actuelle et la taille de l'intervalle de confiance de la récompense moyenne.

Pour notre proposition, nous avons comparé dans un premier temps les trois politiques exploratrices définies ci-dessus et retenu la méthode l'UCB1 pour concevoir notre algorithme de sélection.

Notre proposition se base sur un modèle qui se compose de quatre entités (Fig. 6.4): l'utilisateur final, l'iBox, le réseau et les serveurs.

Les différentes étapes de sélection de serveurs embarquées dans le composant iBox, sont énumérées ci-dessous :

1. Chaque serveur est choisi au moins une fois.
2. Calculer la moyenne des MOS obtenus pour chaque serveur :

$$\hat{M}_i = \frac{\sum_{j=0}^{t-1} M_j \times \mathbf{1}_{\{s^j \text{ est choisi}\}}}{n_{s^i}} \quad (6.3)$$

où  $\mathbf{1}_{\{s^j \text{ est choisi}\}}$  est égal à 1 si le serveur  $s^j$  est choisi, 0 sinon.  $n_{s^i}$  est le nombre de sélection du serveur  $s_i$ .

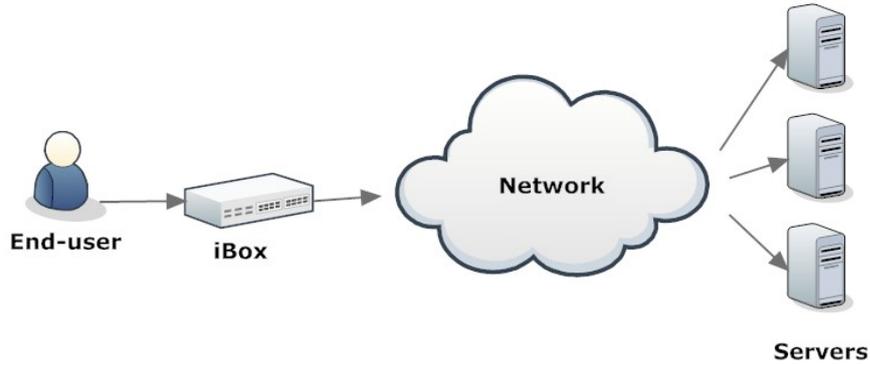


Figure 6.4: Le modèle de CDN avec quatre composants

3. Calculer l'indice de  $UCB_i$  pour chaque serveur  $s^i$  :

$$UCB_i = \hat{M}_i + \sqrt{\frac{2 \ln(t)}{n_{s^i}}} \quad (6.4)$$

4. Choisir le serveur ayant l'indice UCB maximal.

## Travaux expérimentaux

Afin de valider nos travaux, le chapitre 4 montre les résultats expérimentaux obtenus pour nos deux propositions : l'algorithme de routage QQAR et la nouvelle méthode de sélection de serveurs. Nous y décrivons aussi notre banc d'essai pour la méthode d'estimation de la QdE.

Cette dernière s'inspire de l'approche PSQA [Rubino, 2005], elle se fait par le biais d'un modèle neuronal entraîné à partir d'un ensemble de test regroupant plusieurs utilisateurs amenés à donner des scores traduisant leur perception d'un rendu vidéo. Les dispositifs expérimentaux (figure 6.5) de ce banc d'essai comprennent un serveur de vidéo streaming, des clients et un outil de réseau appelé Netem [Hemmingner, 2005] qui permet de tester différentes configurations en faisant varier les paramètres du réseau. A des fins expérimentales, nous nous sommes concentrés sur les quatre paramètres suivants qui peuvent influencer la qualité sur le rendu de la vidéo : le délai, la gigue, le taux de perte et le taux de perte conditionnelle.

- *Plateforme réelle.* Cette plateforme, que nous avons implantée et illustrée dans la figure 6.6 contient douze PCs : un client et onze serveurs vidéo. Un PC supplémentaire est utilisé pour jouer le rôle d'un iBox du côté du client.
- *Simulation.* L'ensemble de nos algorithmes ont été testés en utilisant le simulateur OPNET. Différentes configurations de topologie réseau ont été sélectionnées et mises en œuvre. La figure 6.7 décrit quelques unes de ces topologies.

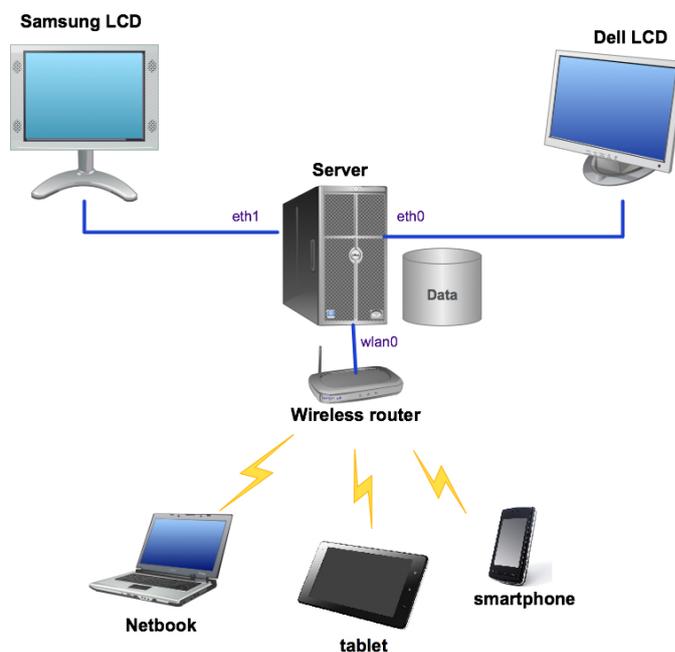


Figure 6.5: Schéma pour le banc d'essai

Trois métriques ont été sélectionnées pour évaluer la performance des approches testées : le MOS moyen, le taux de réception des paquets et le trafic de signalétique généré par nos algorithmes de routage.

Nos approches ont été comparées à d'autres techniques aussi bien en mode simulé qu'en mode réel. Les résultats obtenus à la fois pour le routage et pour la sélection de serveurs ont démontré l'intérêt de l'intégration des mesures de QdE dans tout processus de décision.

## Conclusion

Aujourd'hui, avec le développement rapide des technologies réseaux et la constante progression des besoins des utilisateurs, les mécanismes traditionnels liés à la notion de Qualité de Service (QoS) ne permettent plus de garantir des niveaux élevés de performance et ne suffisent plus à satisfaire les attentes des utilisateurs, principalement dans le contexte des réseaux orientés contenus (Content-Oriented Network - CON). Les travaux développés dans le cadre de cette thèse s'intéressent à un nouveau paradigme, appelé qualité de l'expérience (QdE), permettant de mesurer le niveau de perception de l'utilisateur des services qui lui sont fournis. Au regard des contraintes de plus en plus dynamiques dans les applications actuelles, un mécanisme basé sur une boucle de décision-contrôle/commande- pour le routage adaptatif a été développé dans lequel la QdE usager tient le rôle de métrique régulatrice. Le domaine considéré concerne les réseaux de

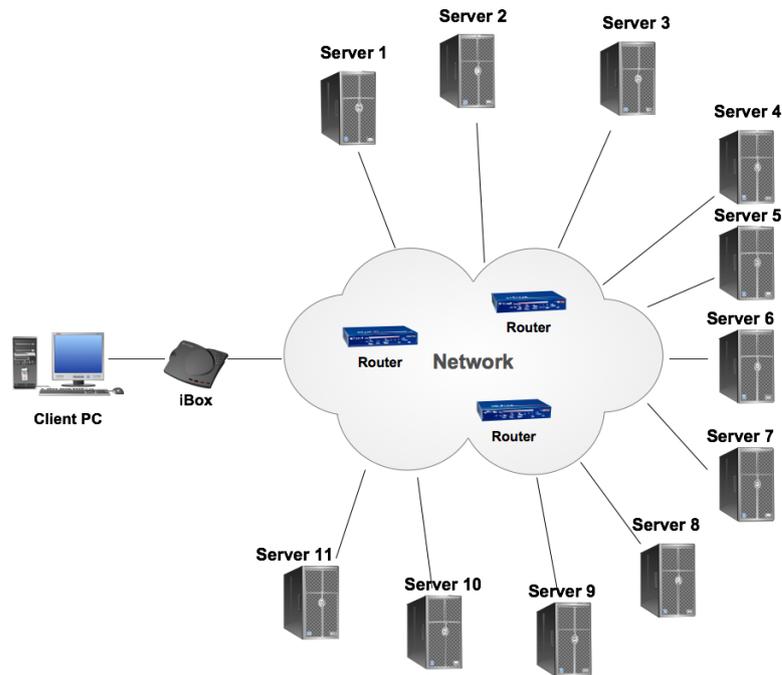


Figure 6.6: Real Platform

distribution de contenu pour lesquels de nouveaux algorithmes de routage et de sélection de serveurs ont été développés sur la base de l'apprentissage par renforcement. Les algorithmes proposés ont été validés et comparés aux approches traditionnelles à la fois en mode simulé et en mode réel. Les résultats obtenus montrent clairement l'intérêt de ce nouveau type de métrique pour un meilleur rendu des applications usagers.

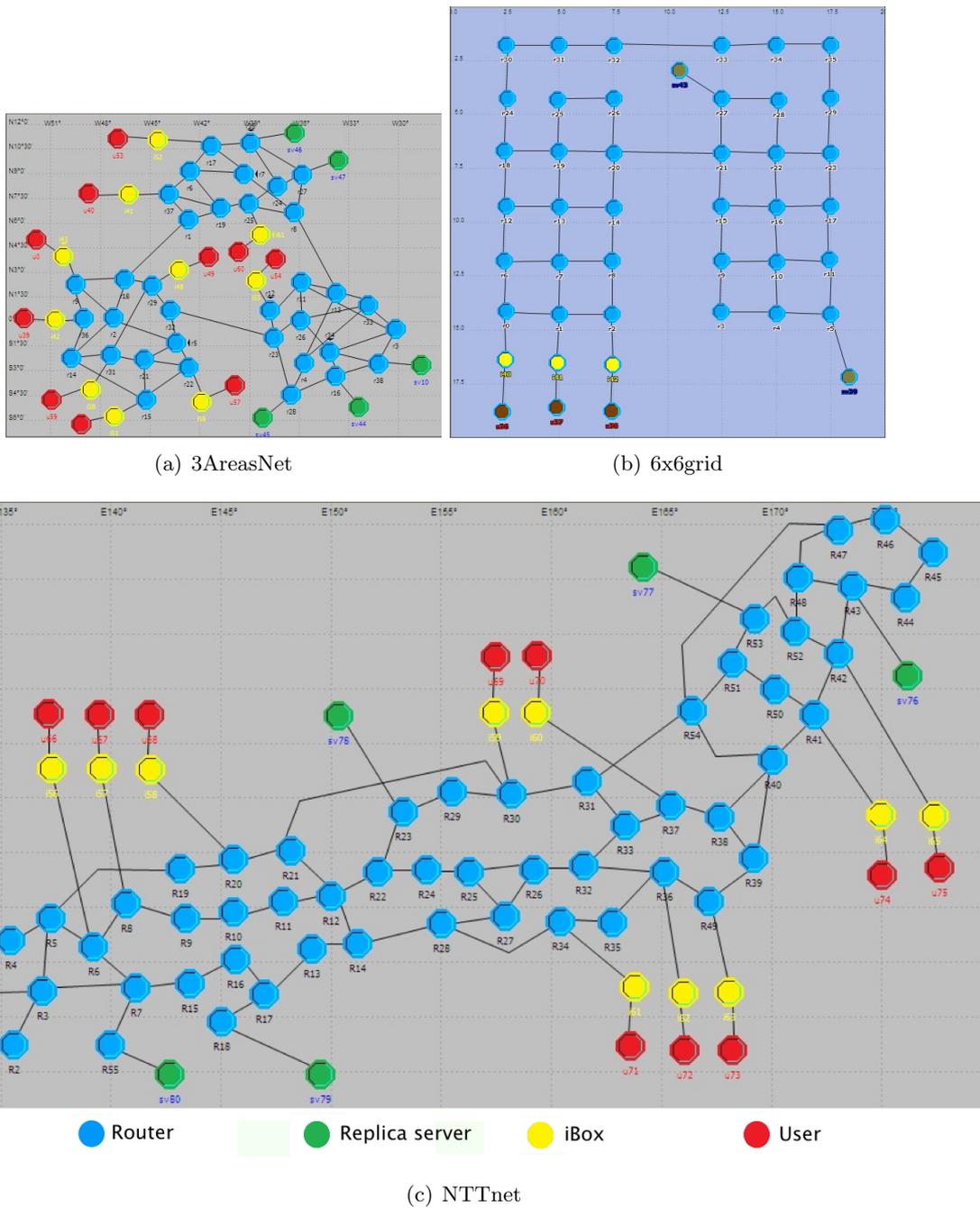


Figure 6.7: Topologies réseaux en mode simulé

# Appendices

# Appendix A

## QoE measurement method

### A.1 Introduction

As mentioned previously in chapter 1, there are lots of QoE measurement methods that assess the user perception in different ways. In fact, the perceived quality of a multimedia stream is not easy to be evaluated because assessing perceived quality requires real people to evaluate it subjectively. The ITU-P.800 recommendation [Recommendation, 1996] for telephony, or the ITU-R BT.500-10 [Recommendation, 2002] for video are two representative standard methods for organizing *subjective* quality assessments. However, subjective evaluations are expensive in terms of time and manpower because it is hard to repeat often. In addition, the subjective method cannot be a part of an automatic process.

As subjective assessment is useless for real time and automatic evaluations, the researchers have made lots of significant research effort to obtain similar evaluations in using *objective* methods (i.e. algorithms and formulas that measure the quality of a stream). We cite here some commonly used objective measurement methods for audio: ITU E-model [Recommendation, 2011b], Signal-to-Noise Ratio (SNR), Enhanced Modified Bark Spectral Distortion (EMBSD) [Yang, 1999], Segmental SNR (SNRseg), Perceptual Speech Quality Measure (PSQM) [Beerends and Stemerdink, 1992], Measuring Normalizing Blocks (MNB) [Vorán, 1997], PSQM+ [Beerends et al., 1997], Perceptual Analysis Measurement System (PAMS) [Rix, 2002], etc. Regarding video quality, there are several measurement methods such as: the ITS' Video Quality Metric (VQM) [Vorán, 2002], Color Moving Picture Quality Metric (CMPQM) [van den Branden Lambrecht, 2002], and Normalization Video Fidelity Metric (NVFM) [van den Lambrecht, 1996]. However, as explained in chapter 1, the service quality is a subjective concept in the network context today. These objective methods cannot replace subjective ones because their provided assessments do not correlate with human perception and their use as quality measurement method is really limited. Apart from the ITU E-model, these methods propose several ways to compare the received data with the original one. The E-model gives an approximation of the perceived quality as a linear function of network parameters. However, its results is proven not correlate with

subjective evaluation in [Hall, 2001, Recommendation, 2011b].

In this thesis, we based on a method that is considered as a hybrid between subjective and objective evaluation: the Pseudo-Subjective Quality Assessment (PSQA) [Rubino, 2005]. Our QoE measurement method is used to assess the quality of a video streaming application.

This appendix presents our PSQA-based QoE measurement method to evaluate the user perception. We describe in detail also the implemented testbed to collect the dataset for this measurement method.

## A.2 PSQA-based QoE measurement method

In this section, we present our PSQA-based QoE measurement method. The key brain of the PSQA method is the supervised learning of a Random Neural Network (RNN) [Gelenbe, 1993] to learn the relation between the parameters causing the distortion and the perceived quality. We divide the measurement method into five steps (Fig. A.1):

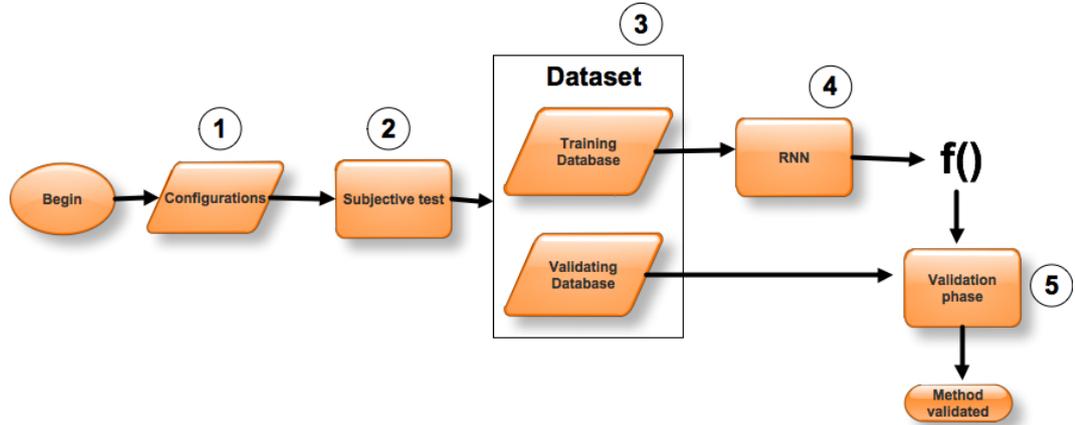


Figure A.1: PSQA-based QoE measurement model

- **Step 1: Generating configurations.** We have to determine  $N$  parameters that influence the perceived quality. This parameter set is  $\Gamma = \{\gamma_1, \dots, \gamma_N\}$ . After determining these parameters, we need a set of representative values for each  $\gamma_i$  with a minimal value  $\gamma_{\min}$  and maximal value  $\gamma_{\max}$ , according to the condition under which we expect the system to work. Let  $\{\mu_{i1}, \dots, \mu_{iH_i}\}$  be this set of values with  $\gamma_{\min} = \mu_{i1}$  and  $\gamma_{\max} = \mu_{iH_i}$ . The number of value  $H_i$  for each  $\gamma_i$  depends on the size of the chosen interval. In this context, let *configuration* be a set with the form  $\psi = \{\nu_1, \dots, \nu_n\}$  where  $\nu_i$  is one of the chosen values for  $\gamma_i$ .

The total number of possible configuration is equal to  $\prod_{i=1}^n H_i$ . It is very large to realize all of them. Hence, we need to select a subset of these possible configurations. This selection can be realized randomly. The uniform distribution is not

advisable. We sample further more points in the regions near the configurations which are most likely to happen during normal use. Then, we generate a set of “distorted samples” of the original media over the network by using the parameters of these configurations. In the next section, we present in detail our testbed and the way to generate these samples.

- **Step 2: Subjective test.** We select a set of  $m$  media samples:  $\theta_j$  with  $j = [1, m]$ . It is necessary to have  $C$  configurations,  $\{\psi_1, \dots, \psi_C\}$  where  $\psi_c = \{\nu_{c1}, \dots, \nu_{cN}\}$ ,  $\nu_{cn}$  is the value of parameter  $\gamma_n$  in configuration  $\psi_c$ . So, for each media sample  $\theta_j$ , we build a set  $\{\theta_{j1}, \dots, \theta_{jC}\}$  of samples equivalent to the set of  $C$  configurations  $\{\psi_1, \dots, \psi_C\}$ . In other words, sequence  $\theta_{jc}$  is the sequence the receiver receives after the sender sent  $\theta_j$  through the network system where the  $N$  chosen parameters had the values of configuration  $\psi_c$ .

After generating the distorted samples, we realize a *subjective test* [Recommendation, 1996, Recommendation, 1999, Recommendation, 2002] for each piece  $\theta_{jc}$  received. Then, the sequence  $\theta_{jc}$  receives the value  $\varepsilon_{jc}$ . For this value of evaluation, we use the MOS score (Mean Opinion Score) presented in chapter 1, which is given by ITU-T Rec P.801 [Recommendation, 2006] with the scale of: 1 - Bad, 2 - Poor, 3 - Fair, 4 - Good and 5 - Excellent.

The idea is to assign each configuration  $\psi_c$  to the value of assessing:

$$\varepsilon_c = \frac{1}{m} \sum_{i=1}^m \varepsilon_{ic} \quad (\text{A.1})$$

This step is presented in section 4.2.

- **Step 3: Dataset.** The subjective test in the last step gives us  $C$  configurations, which are divided into two separate types: *Training Database* and *Validation Database*, such as  $C = C_T + C_V$  where  $C_T$  and  $C_V$  are the number of configurations of Training Database and Validation Database, respectively. The Validation Database is reserved for further use in the last step of the process.
- **Step 4: Training RNN.** The next step is the process of training the Random Neural Network, a statistical learning tool, to learn the mapping between configurations of Training Database and MOS score values. All values of parameters are standardized into interval  $[0, 1]$  and the same with the output quality. Once the tool has been trained, we obtain a function  $f()$  that mapping any configuration into the quality score. This step is presented in detail in section A.4.
- **Step 5: Validation phase.** This step is the validation phase that consists of comparing the values obtained by  $f()$  with the  $\varepsilon$  of the Validation Database. If they are close enough for all of them, the RNN is validated.

## A.3 Parameters affecting QoE

In this section, since the chosen application is video streaming, we analyze different parameters of video streaming that affect the QoE. The goal is to determine a set of parameters  $\Gamma$  mentioned in section A.2.

### A.3.1 Network parameter

The QoS parameters are network related elements and have an effect on QoE. These parameters are packet loss, delay, jitter and throughput. Each of these considered parameters leads separately or together to blocking, blurriness or even blackouts with different level of degradation. Certainly, QoS parameters have relative important degrees and affect each video application category in different ways.

#### A.3.1.1 Packet loss

The packet loss is considered as the first cause that has a direct effect on the video presented to the end-user. It happens, either due to the loss of packets during its transmission caused by the network congestion or due to its late arrival at the destination codec buffer.

Missing information causes impairments along the communication and at the user machine. In fact, video is usually encoded in a format that handles the packet loss. We consider for example the MPEG format, which uses 3 types of frames:  $I$ ,  $P$  and  $B$  [Huszak and Imre, 2010]. It is characterized by its intra-frame and inter-frame compression in which  $I$  is the intra coded picture,  $P$  is the predictive coded picture and  $B$  is the bidirectional predictive coded picture. The repeated pattern of  $I$ ,  $P$  and  $B$  frames in a video stream is called the Group of Pictures (GOP).

While the loss of packets in a frame can degrade the video quality, the more problematic situation is the loss of the  $I$ -frame and the propagation of errors to dependent frames. The more packet loss rate increases, the more the decoded frames degrade.

#### A.3.1.2 Jitter

Jitter is generally a performance parameter, which characterizes the variance in packet arrival times at the destination [Winkler and Mohandas, 2008]. It is generally caused by the packet arrival time influenced by, first, the use of different paths to reach the same destination, second, the queuing management algorithms.

Even if most receivers implement a play-out buffer, which buffers incoming packets and delays their playout in order to nullify the effects of jitter. This metric, beyond a certain threshold, is similar to network loss. If packets arrive out of order and later than the buffering time, they will be discarded at the receiver

### A.3.1.3 Delay

It is the time taken by the packet to travel from its source to its destination. This value depends on the application nature and includes delays in the terminal, networks and servers. In reality, video streaming is sensitive to the delay. If it exceeds a certain value, the video will be displayed with frozen screen and lost blocks.

## A.3.2 Video characteristics

This subsection presents different video characteristics that affect user perception while watching the video.

### A.3.2.1 Frame rate and resolution

The reduction of the bit rate can be very important in different applications of video streaming. One of the possible manners to reduce it, is to adjust the video's transmission with the available bandwidth. It can be done by adjusting the spatial and temporal information, i.e. resolution and frame rate as it plays a crucial role in describing sequence differences while ensuring an acceptable quality of service from a user perspective.

The resolution and the frame rate are two video parameters easy to be measured, and they affect directly the QoE. Janowski et al. [Janowski and Romaniak, 2010] presented the results of an experiment confirming this theory. This experiment is initially based on changing the resolution of the video for a constant frame rate and then on varying frame rate from 5 to 30 frames per seconds for a fixed resolution.. The results showed that even if these two parameters had significant effects on the video quality, a simple metric based on sequence frame rate or resolution cannot predict QoE properly.

### A.3.2.2 Motion complexity

The influence of the resolution and frame rate on different video contents is different. For example, we consider a sequence with a little motion complexity, the image quality remains the same for different values of frame rate. It is not the case for a complex video. Hence, one needs a metric that takes into consideration the degree of motion complexity in the video. In this context, new researches try to adjust the size of the motion interpolation structure (or GOP length) according to the motion activity along the sequence. In fact, GOP length extracted from video characteristics can be considered in the determination of the QoE [Ascenso et al., 2006].

Janowski et al. [Janowski and Romaniak, 2010] discussed the influence of motion complexity on the QoE. Their experience was based on three different videos, which had the same characteristics (resolution and frame rate): 1- Blue sky, 2-Crowd run, 3-Park Joy. The results showed that unlike video 2 and 3, video 1 having the least movements had a good quality even by the frame rate changes.

### A.3.2.3 Video content

Zinner et al. [Zinner et al., 2010] showed that the content of the video is also important in the QoE context. In [Zinner et al., 2010], the selection of participants was based on their interest in football. The results showed that football fans found the quality of only 6 frames per second (fps) video acceptable 80% of the time when the average of frame rate commonly used is 25 fps. We can conclude that, from this result, the human visual system can tolerate a certain degree of distortion and that participants may accept a poor quality of the video, if there is enough interest in the topic.

### A.3.2.4 Used codec

Sending an uncompressed video over the Internet network requires large amounts of bandwidth and storage space. Therefore, the videos are compressed with very effective and lossy compression algorithms. A lot of video codecs are supported for the video services such as H.262, H.263, H.264, WMV3 and WVID.

### A.3.2.5 Video delivery

Video delivery refers to the ability of an application to ensure continuous and synchronized streams of video. We discuss now the influence of the application and transport protocol used to deliver the video streams on the final video quality.

While one generally assumes that using both RTP and RTCP feedback on the application/session layer and UDP on the transport layer is suited for the delivery of multimedia streams, a brief overview of some popular services showed that videos can also be delivered through RTMP/TCP (a proprietary protocol developed by Adobe) or even HTTP/TCP.

In contrast to the UDP, TCP provides mechanisms to perform re-transmission. As a result, the use of TCP as the transport protocol (instead of UDP) might smooth the effect of loss rates in the resulting video QoE.

## A.3.3 User equipment

With the advances in the telecommunication industry, the consumer electronic industry has offered a large scale of products for multimedia services. New generation end-user devices substantially vary in size, processing power, usage, and other respects. We can distinguish three classes of end user devices: Personal computers, mobile video devices (cell phones and iPods) and TV sets. For reason, we need an idea about these parameters as they affect the QoE. Such information can be used to avoid useless calibrations on the network such as sending a HD video to a small terminal with standard resolution or changing the resources allocation, when the user machine is the source of quality degradation.

### A.3.3.1 Display size

The network providers can evaluate subscriber's QoE using QoS parameters weight but also taking into account the terminal type with his different characteristics. Kim et al. [Kim and Choi, 2010] present a numerical formula model to evaluate the subscriber's IPTV video QoE:

$$QoE = Qr * (1 - QoS)^{((QoS*A)/R)} \quad (A.2)$$

Where  $Qr$  is a coefficient limiting the range of the IPTV image QoE according to the display size and the resolution of the terminal. For example two users with the same conditions of reception may have different qualities due to the resolution of their terminal is different.

### A.3.3.2 System and navigator CPU usage

It describes the computing resources used by the system and the web navigator, which affect directly the quality of the video. In fact, when the CPU is too loaded, information received by the user machine cannot be treated easily. As a result, the video will be blocked with frozen images.

### A.3.3.3 Browser type

In video streaming, the browser is the software application often used to display the received video. As every browser has its performance, its own way of rendering web pages, its sensitivity toward loading speed, the type of browser and even the used version are important in considering the QoE.

### A.3.4 Psychological factors

QoS has long been used to ensure service guarantees for parameters like bandwidth, delay, jitter, and loss-rate to enhance application performance. However, QoS lacks an important element in characterizing videos streams: human perception.

QoE is a new concept based on the user's perception. Since the human behaviour is unpredictable, the operator must take into consideration a great number of parameters and metrics that depend on the user emotional behaviour in order to determine the adequate quality level for a specific multimedia application. Otherwise, the operator has to understand what happens in the human mind while watching the video sequence to assess his perception.

The estimation of the quality level changes from a person to another and this change depends on several factors that are related to the person, his preferences and his environment. Jumisko et al. [Jumisko et al., 2005] specify some of these parameters:

- The user characteristics (knowledge background, language background, familiarity with the task, age, sex)
- The user behaviour (interest degree on the content)

## A.4 Training Random Neural Network

The section 4.2 has presented the implemented testbed for providing the dataset. In this section, this dataset obtained is used for training the Random Neural Network. This step is the key brain of the PSQA-based QoE measurement method. The latter maps a configuration onto a MOS score as output. In this section, we first present the RNN tool. After, we describe the neural network and how it works.

### A.4.1 Random Neural Network tool

Gelenbe [Gelenbe, 1989, Gelenbe, 1993, Gelenbe, 1990] has invented the Random Neural Network (RNN) that is actually an open queuing network with positive and negative units. This tool has been applied in different fields [Bakırcıoğlu and Koçak, 2000].

RNN is a stochastic dynamic network of interconnected neurons among which units circulate. The feed-forward RNN architecture is commonly used for learning applications. The neurons are classified into three sets:

- a set of *input* neurons  $\mathcal{I}$  having  $I$  elements.
- a set of *hidden* neurons  $\mathcal{H}$  having  $H$  elements.
- a set of *output* neurons  $\mathcal{O}$  having  $O$  elements.

The circulating units come from environment to the input layer. Then, they go to the hidden layer, after, to the output layer. Finally, they leave the network. Assume that the moving between neurons is done instantaneously. In other words, the units in the network are in one of its neurons at any point in time. Each neuron  $i$  in the network is associated a neuron's rate, a positive number  $\mu_i$ . The different units in any neuron are stored according to a FIFO schedule. If there are  $n > 0$  units at node  $i$  at time  $t$ , then, at a random time  $T$  a unit will leave neuron  $i$  where  $T - t$  is an exponential random variable with parameter  $\mu_i$ . In other words, neuron  $i$  follows the  $. / M / 1$  queue. If  $i \in \mathcal{O}$  then the unit will leave the network.

Each neuron  $i$  in set  $\mathcal{I}$  or  $\mathcal{H}$  has a probability distribution  $(p_{i,j}^+, p_{i,j}^-)_j$ . After leaving the neuron  $i$ , a unit goes to the neuron  $j$  as positive one with probability  $p_{i,j}^+$  or as negative one with probability  $p_{i,j}^-$ . That means:  $\sum_j (p_{i,j}^+ + p_{i,j}^-) = 1$ . Note that:

- If  $i \in \mathcal{I}$  and  $j \in \mathcal{I} \cup \mathcal{O}$ , then  $p_{i,j}^+$  and  $p_{i,j}^-$  are zero.
- If  $i \in \mathcal{H}$  and  $j \in \mathcal{I} \cup \mathcal{H}$ , then  $p_{i,j}^+$  and  $p_{i,j}^-$  are zero.

When a unit arrives as *positive* one at a neuron having  $n$  units, it joins the queue and the number  $n$  of units at the neuron increases by 1. On the other hand, when a unit arrives as *negative* one at a neuron, it disappears and if  $n > 0$ , one of the units is destroyed. If  $n = 0$ , only this negative unit disappears and nothing else happens. Units arrive from outside at neuron of set  $\mathcal{I}$  following independent Poisson flows. Generally, with Poisson distribution, the positive units arrive at node  $i$  with rate  $\lambda_i^+$  and negative

units arrive with rate  $\lambda_i^-$ . The *steady state*  $\varrho_i$  of a neuron  $i$  is defined as a state in which the neuron  $i$  is not empty (has at least one unit).

We assume that any node  $i$  in  $\mathcal{I}$  and any node  $h$  in  $\mathcal{H}$  can send its units outside. That means: for any  $i \in \mathcal{I}$  there is some  $h \in \mathcal{H}$  and some  $o \in \mathcal{O}$  such that  $(p_{i,h}^+ + p_{i,h}^-)(p_{h,o}^+ + p_{h,o}^-) > 0$ .

Assume that for all  $i \in \mathcal{I}$ , we have:

$$x_i = \frac{\lambda_i^+}{\mu_i + \lambda_i^-} < 1 \quad (\text{A.3})$$

We assume that for all  $h \in \mathcal{H}$ :

$$x_h = \frac{\sum_{i \in \mathcal{I}} x_i \mu_i p_{i,h}^+}{\mu_h + \sum_{i \in \mathcal{I}} x_i \mu_i p_{i,h}^-} < 1 \quad (\text{A.4})$$

Finally, for each  $o \in \mathcal{O}$ :

$$x_o = \frac{\sum_{h \in \mathcal{H}} x_h \mu_h p_{h,o}^+}{\mu_o + \sum_{h \in \mathcal{H}} x_h \mu_h p_{h,o}^-} < 1 \quad (\text{A.5})$$

Then, the network is ergodic and for all neuron  $i$ , we have  $\varrho_i = x_i$ .

#### A.4.2 RNN Learning

The goal is to determine the mapping of a configuration onto the quality (MOS score). Denote by  $\vec{\nu} = \nu_1, \nu_2, \dots, \nu_n$  the selected set of metrics, and by  $q$  the perceived quality. The experiments give  $K$  pairs of values of the metrics and their corresponding quality, denoted by:  $(\vec{\nu}^{(k)}, q^{(k)})_{k=1..K}$ . The RNN has now  $I = n$  input nodes,  $O = 1$  output nodes. We set the rates of negative units arriving from outside to 0, and we make the following association:  $\lambda_i^+ = \nu_i$  with  $i = 1..n$ , and  $\varrho_o = q$ .

All of input rates  $\nu_1, \nu_2, \dots, \nu_n$  are normalized in  $[0, 1]$ . We set  $H = 2n$ . We set also  $\mu_i = 1$  for all  $i \in \mathcal{I}$  and  $\mu_j = n$  for all other node. The cost function is defined as:

$$C = \frac{1}{2} \sum_{k=1}^K \left( \varrho_o^{(k)} - q^{(k)} \right)^2 \quad (\text{A.6})$$

where  $\varrho_o^{(k)}$  is the probability that neuron  $o$  is excited when the input variables are  $\lambda_i^+ = \nu_i^{(k)}$  for all  $i \in \mathcal{I}$ . The goal is to minimize this cost function  $C$ . The variable change  $\omega_{i,j}^+ = \mu_i p_{i,j}^+$ ,  $\omega_{i,j}^- = \mu_i p_{i,j}^-$  is denoted by  $\vec{\omega}$  this new set of variables, called *weights* in the area.

We have  $C = C(\vec{\omega})$ , the dependency on  $\vec{\omega}$  through the  $\varrho_o^{(k)}$ :

$$\varrho_o^{(k)} = \frac{\sum_{h \in \mathcal{H}} \frac{\sum_{i \in \mathcal{I}} (\nu_i^{(k)} / \mu_i) \omega_{i,h}^+}{\mu_h + \sum_{i \in \mathcal{I}} (\nu_i^{(k)} / \mu_i) \omega_{i,h}^-} \omega_{h,o}^+}{\mu_o + \sum_{h \in \mathcal{H}} \frac{\sum_{i \in \mathcal{I}} (\nu_i^{(k)} / \mu_i) \omega_{i,h}^+}{\mu_h + \sum_{i \in \mathcal{I}} (\nu_i^{(k)} / \mu_i) \omega_{i,h}^-} \omega_{h,o}^-} \quad (\text{A.7})$$

The minimization procedure should be adapted to the mathematical constrained problem of minimizing  $C(\vec{\omega})$  over the set:  $\sum_j (\omega_{i,j}^+ + \omega_{i,j}^-) = \mu_i$ .

Rubino [Rubino, 2005] proved that a well trained RNN extrapolates well by giving reasonable results even for parameters values outside the ranges considered during training. The interesting mathematical properties of this neural network model is that it can obtain the derivatives of the output with respect to any of the inputs, which is useful for evaluating the performance of the network under dynamic environment.

After the RNN training, the most important thing we got is the function  $F_{\text{QoE}}$

$$\begin{aligned} F_{\text{QoE}} : \Gamma &\rightarrow \text{MOS score} \\ (d, j, l, cl) &\mapsto q \end{aligned} \tag{A.8}$$

where  $\Gamma$  is the configuration that consists of QoS parameters:  $d$ ,  $j$ ,  $l$  and  $cl$ , which are *delay*, *jitter*, *loss-rate* and *conditional loss-rate*, respectively.

In this thesis, we used this function  $F_{\text{QoE}}$  to assess the QoE. Note that our testbed is realized in offline way. The goal is just to obtain the function  $F_{\text{QoE}}$  and use it to evaluate the QoE value.

# Bibliography

- [itu, 2000] (2000). Recommendation 500-10: Methodology for the subjective assessment of the quality of television pictures. ITU-R Rec. BT.500.
- [Agboma and Liotta, 2008] Agboma, F. and Liotta, A. (2008). QoE-aware QoS Management. In *Proceedings of the 6th International Conference on Advances in Mobile Computing and Multimedia (MoMM '08)*, Linz, Austria.
- [Agrawal, 1995] Agrawal, R. (1995). Sample mean based index policies with  $o(\log n)$  regret for the multi-armed bandit problem. *Advances in Applied Probability*, 27(4):1054–1078.
- [Akamai, 1998] Akamai (1998). Akamai. <http://www.akamai.com>.
- [Almeida et al., 2004] Almeida, J., Eager, D., Vernon, M., and Wright, S. (2004). Minimizing delivery cost in scalable streaming content distribution systems. *Multimedia, IEEE Transactions on*, 6(2):356–365.
- [Andrews et al., 2002] Andrews, M., Shepherd, B., Srinivasan, A., Winkler, P., and Zane, F. (2002). Clustering and server selection using passive monitoring. In *IN-FOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, New York, USA. IEEE.
- [Antos et al., 2008] Antos, A., Grover, V., and Szepesvári, C. (2008). Active learning in multi-armed bandits. In *The 19th International Conference on Algorithmic Learning Theory*, Budapest, Hungary. Springer.
- [Apostolopoulos and Orda, 1999] Apostolopoulos, G. and Orda, A. (1999). Qos routing mechanisms and ospf extensions. Technical report, RFC 2676.
- [Ardaiz et al., 2001] Ardaiz, O., Freitag, F., and Navarro, L. (2001). Improving the service time of web clients using server redirection. *ACM SIGMETRICS Performance Evaluation Review*, 29(2):39–44.
- [Ascenso et al., 2006] Ascenso, J., Brites, C., and Pereira, F. (2006). Content adaptive wyner-ziv video coding driven by motion activity. In *International Conference on Image Processing*, Atlanta, GA, USA. IEEE.

- [Atkinson and Fanto, 2007] Atkinson, R. and Fanto, M. (2007). RIPv2 Cryptographic Authentication. Technical report, RFC 4822.
- [Auer et al., 2002] Auer, P., Cesa-Bianchi, N., and Fischer, P. (2002). Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2):235–256.
- [Auer et al., 1995] Auer, P., Cesa-Bianchi, N., Freund, Y., and Schapire, R. (1995). Gambling in a rigged casino: The adversarial multi-armed bandit problem. In *36th Annual Symposium on Foundations of Computer Science*, Milwaukee, Wisconsin, USA. IEEE.
- [Auer et al., 2003] Auer, P., Cesa-Bianchi, N., Freund, Y., and Schapire, R. (2003). The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing*, 32(1):48–77.
- [Auer et al., 2009] Auer, P., Jaksch, T., and Ortner, R. (2009). Near-optimal regret bounds for reinforcement learning. *Advances in Neural Information Processing Systems*, (21):89–96.
- [Awerbuch and Kleinberg, 2008] Awerbuch, B. and Kleinberg, R. (2008). Online linear optimization and adaptive routing. *Journal of Computer and System Sciences*, 74(1):97–114.
- [Aweya, 2002] Aweya, J. (2002). Transmission control protocol. In *Encyclopedia of Telecommunications*. Wiley Online Library.
- [Bakırcıođlu and Koçak, 2000] Bakırcıođlu, H. and Koçak, T. (2000). Survey of random neural network applications. *European Journal of Operational Research*, 126(2):319–330.
- [Bartal, 1996] Bartal, Y. (1996). Probabilistic approximation of metric spaces and its algorithmic applications. In *37th Annual Symposium on Foundations of Computer Science*, pages 184–193, Burlington, Vermont. IEEE.
- [Batteram et al., 2010] Batteram, H., Damm, G., Mukhopadhyay, A., Philippart, L., Odysseos, R., and Urrutia-Valdés, C. (2010). Delivering quality of experience in multimedia networks. *Bell Labs Technical Journal*, 15(1):175–193.
- [Beck and Moore, 1998] Beck, M. and Moore, T. (1998). The internet2 distributed storage infrastructure project: An architecture for internet content channels. *Computer Networks and ISDN systems*, 30(22-23):2141–2148.
- [Beerends et al., 1997] Beerends, J., Meijer, E., and Hekstra, A. (1997). Improvement of the p. 861 perceptual speech quality measure. *ITU-T SG12 COM-34E*, 196.
- [Beerends and Stemerding, 1992] Beerends, J. and Stemerding, J. (1992). A perceptual audio quality measure based on a psychoacoustic sound representation. *Journal-Audio Engineering Society*, 40:963–963.

- [Bellman and Kalaba, 1959] Bellman, R. and Kalaba, R. (1959). On adaptive control processes. *Automatic Control, IRE Transactions on*, 4(2):1–9.
- [Bernardo et al., 2008] Bernardo, F., Vučević, N., Umberto, A., and López-Benítez, M. (September 2008). Quality of Experience Evaluation under QoS-aware Mobility Mechanisms. In *Proceedings of the 14th European Wireless Conference*, Prague, Czech Republic.
- [Bhattacharjee et al., 1997] Bhattacharjee, S., Ammar, M., Zegura, E., Shah, V., and Fei, Z. (1997). Application-layer anycasting. In *INFOCOM'97. Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, Kobe, Japan. IEEE.
- [Bitam et al., 2010a] Bitam, S., Batouche, M., and Mellouk, A. (2010a). Qosbeemanet: a new qos multipath routing protocol for mobile ad-hoc networks. In *GLOBECOM Workshops*, Florida, USA. IEEE.
- [Bitam et al., 2010b] Bitam, S., Batouche, M., and Talbi, E. (2010b). A survey on bee colony algorithms. In *IEEE International Symposium on Parallel & Distributed Processing, Workshops and Phd Forum (IPDPSW)*, Atlanta, Georgia, USA. IEEE.
- [Bocci et al., 2010] Bocci, M., Bryant, S., Frost, D., Levrau, L., and Berger, L. (2010). A framework for mpls in transport networks. *RFC5921*.
- [Boyan and Littman, 1994] Boyan, J. and Littman, M. (1994). Packet routing in dynamically changing networks: A reinforcement learning approach. *Advances in neural information processing systems*, 6:671–678.
- [Broadband Forum, 1995] Broadband Forum (1995). Broadband Forum. <http://www.broadband-forum.org/>.
- [Brooks and Hestnes, 2010] Brooks, P. and Hestnes, B. (2010). User measures of quality of experience: why being objective and quantitative is important. *Network, IEEE*, 24(2):8–13.
- [Bubeck et al., 2008] Bubeck, S., Munos, R., Stoltz, G., Szepesvári, C., et al. (2008). Online optimization in x-armed bandits. In *Neural Information Processing Systems conference (NIPS)*, Vancouver, B.C., Canada.
- [Cesa-Bianchi and Fischer, 1998] Cesa-Bianchi, N. and Fischer, P. (1998). Finite-Time Regret Bounds for the Multiarmed Bandit Problem. In *Proceedings of the 15th International Conference on Machine Learning*, Madison, Wisconsin, USA.
- [Chellouche et al., 2010] Chellouche, S., Negru, D., Borcoci, E., and LeBARS, E. (2010). Anycast-based context-aware server selection strategy for vod services. In *GLOBECOM Workshops (GC Wkshps), 2010 IEEE*, Florida, USA. IEEE.

- [Chen et al., 2006] Chen, K., Huang, C., Huang, P., and Lei, C. (2006). Quantifying Skype user satisfaction. In *ACM SIGCOMM Computer Communication Review*, volume 36, pages 399–410. ACM.
- [Chen, 2007] Chen, T. (2007). Network traffic modeling. *Handbook of Computer Networks*, pages 326–339.
- [Chikkerur et al., 2011] Chikkerur, S., Sundaram, V., Reisslein, M., and Karam, L. (2011). Objective video quality assessment methods: A classification, review, and performance comparison. *Broadcasting, IEEE Transactions on*, 57(2):165–182.
- [Ciubotaru and Muntean, 2009] Ciubotaru, B. and Muntean, G. (May 2009). SASHA—a quality-oriented handover algorithm for multimedia content delivery to mobile users. *IEEE Transactions on Broadcasting*, 55(2):437–450.
- [Coquelin and Munos, 2007] Coquelin, P. and Munos, R. (2007). Bandit algorithms for tree search. In *Uncertainty in Artificial Intelligence*, Vancouver, Canada.
- [Delgadillo, 1999] Delgadillo, K. (1999). Cisco distributeddirector. *Cisco White Paper*.
- [Di Caro and Dorigo, 2004] Di Caro, G. and Dorigo, M. (2004). Ant Colony Optimization and its application to adaptive routing in telecommunication networks. *Unpublished doctoral dissertation*.
- [Di Caro et al., 2005] Di Caro, G., Ducatelle, F., and Gambardella, L. (2005). AntHocNet: an adaptive nature-inspired algorithm for routing in mobile ad hoc networks. *European Transactions on Telecommunications*, 16(5):443–455.
- [Digital Island, 1996] Digital Island (1996). Digital Island. <http://www.digitalisland.com>.
- [Dilley et al., 2002] Dilley, J., Maggs, B., Parikh, J., Prokop, H., Sitaraman, R., and Wehl, B. (2002). Globally distributed content delivery. *Internet Computing, IEEE*, 6(5):50–58.
- [Dorigo et al., 2006] Dorigo, M., Birattari, M., and Stutzle, T. (2006). Ant colony optimization. *Computational Intelligence Magazine, IEEE*, 1(4):28–39.
- [Edgecast, 2006] Edgecast (2006). Edgecast networks. <http://www.edgecast.com>.
- [Ertel et al., 2009] Ertel, W., Schneider, M., Cubek, R., and Tokicy, M. (2009). The teaching-box: A universal robot learning framework. In *International Conference on Advanced Robotics (ICAR)*, Munich, Germany. IEEE.
- [Esfahani and Analoui, 2008] Esfahani, A. and Analoui, M. (2008). Widest K-shortest Paths Q-routing: A new QoS routing algorithm in telecommunication networks. In *International Conference on Computer Science and Software Engineering*, Wuhan, China. IEEE.

- [Even-Dar et al., 2002] Even-Dar, E., Mannor, S., and Mansour, Y. (2002). PAC bounds for multi-armed bandit and Markov decision processes. In *Computational Learning Theory*, Sydney, Australia. Springer.
- [Fiedler et al., 2010a] Fiedler, M., Hlavacs, H., Hackbarth, K., and Arlos, P. (2010a). Quality of Experience. *Annals of Telecommunications*, 65(1):1–2.
- [Fiedler et al., 2010b] Fiedler, M., Hossfeld, T., and Tran-Gia, P. (2010b). A generic quantitative relationship between quality of experience and quality of service. *Network, IEEE*, 24(2):36–41.
- [Gabillon, 2009] Gabillon, V. (2009). Machine learning tools for online advertisement. Technical report, INRIA Lille, France.
- [Gadde et al., 1997] Gadde, S., Rabinovich, M., and Chase, J. (1997). Reduce, reuse, recycle: An approach to building large internet caches. In *The Sixth Workshop on Hot Topics in Operating Systems*, pages 93–98, Cape Cod, Massachusetts. IEEE.
- [Garcia-Lunes-Aceves, 1993] Garcia-Lunes-Aceves, J. (1993). Loop-free routing using diffusing computations. *IEEE/ACM Transactions on Networking (TON)*, 1(1):130–141.
- [Gelenbe, 1989] Gelenbe, E. (1989). Random neural networks with negative and positive signals and product form solution. *Neural computation*, 1(4):502–510.
- [Gelenbe, 1990] Gelenbe, E. (1990). Stability of the random neural network model. *Neural computation*, 2(2):239–247.
- [Gelenbe, 1993] Gelenbe, E. (1993). Learning in the recurrent random neural network. *Neural Computation*, 5(1):154–164.
- [Gelenbe and Lent, 2004] Gelenbe, E. and Lent, R. (2004). Power-aware ad hoc cognitive packet networks. *Ad Hoc Networks*, 2(3):205–216.
- [Gelly et al., 2006] Gelly, S., Wang, Y., Munos, R., Teytaud, O., et al. (2006). Modification of uct with patterns in monte-carlo go. Technical report, INRIA, France.
- [Georgii, 2011] Georgii, H. (2011). *Gibbs measures and phase transitions*, volume 9. Walter de Gruyter.
- [Gojmerac et al., 2003] Gojmerac, I., Ziegler, T., Ricciato, F., and Reichl, P. (2003). Adaptive multipath routing for dynamic traffic engineering. In *Global Telecommunications Conference, 2003. GLOBECOM'03. IEEE*, San Francisco, USA. IEEE.
- [Goong et al., 2009] Goong, Y., Yang, F., Huang, L., and Su, S. (2009). Model-based approach to measuring quality of experience. In *First International Conference on Emerging Network Intelligence*, Sliema, Malta.

- [Guerin and Orda, 1999] Guerin, R. and Orda, A. (1999). QoS routing in networks with inaccurate information: theory and algorithms. *IEEE/ACM Transactions on Networking (TON)*, 7(3):350–364.
- [Guerin et al., 1997] Guerin, R., Orda, A., and Williams, D. (1997). QoS routing mechanisms and OSPF extensions. In *Global Telecommunications Conference, 1997. GLOBECOM'97.*, IEEE, Phoenix, Arizona. Ieee.
- [Guo et al., 2002] Guo, M., Ammar, M., and Zegura, E. (2002). Selecting among replicated batching video-on-demand servers. In *Proceedings of the 12th international workshop on Network and operating systems support for digital audio and video*, Miami, Florida. ACM.
- [Hall, 2001] Hall, T. (2001). Objective speech quality measures for internet telephony. In *Voice over IP (VoIP) Technology, Proceedings of SPIE*.
- [Hamam et al., 2008] Hamam, A., Eid, M., El Saddik, A., and Georganas, N. (2008). A quality of experience model for haptic user interfaces. In *Proceedings of the 2008 Ambi-Sys workshop on Haptic user interfaces in ambient media systems*, Quebec, Canada. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).
- [Han et al., 2009] Han, Y., Kim, M., and Park, H. (2009). A novel server selection method to achieve delay-based fairness in the server palm. *Communications Letters, IEEE*, 13(11):868–870.
- [Hardwick and Stout, 1992] Hardwick, J. and Stout, Q. (1992). *Bandit strategies for ethical sequential allocation*. Citeseer.
- [Hassan et al., 2010] Hassan, J., Das, S., Hassan, M., Bisdikian, C., and Soldani, D. (2010). Improving quality of experience for network services [guest editorial]. *Network, IEEE*, 24(2):4–6.
- [He and Rexford, 2008] He, J. and Rexford, J. (2008). Toward internet-wide multipath routing. *Network, IEEE*, 22(2):16–21.
- [Heineman et al., 2008] Heineman, G., Pollice, G., and Selkow, S. (2008). *Algorithms in a Nutshell*. O'Reilly Media.
- [Hemminger, 2005] Hemminger, S. (2005). Network emulation with NetEm. In *Australia national Linux conference*, Canberra, Australia. Citeseer.
- [Henry and Luo, 2002] Henry, P. and Luo, H. (2002). WiFi: What's next? *Communications Magazine, IEEE*, 40(12):66–72.
- [Hoceini, 2004] Hoceini, S. (2004). *Techniques d'Apprentissage par Renforcement pour le Routage Adaptatif dans les Réseaux de Télécommunication à Trafic irrégulier*. PhD thesis, University of Paris-Est Creteil.

- [Hoceini et al., 2008] Hoceini, S., Mellouk, A., and Smail, B. (2008). Average-Bandwidth Delay Q-Routing Adaptive Algorithm. In *ICC'08. IEEE International Conference on Communications*, Beijing, China. IEEE.
- [Huffaker et al., 2002] Huffaker, B., Fomenkov, M., Plummer, D., Moore, D., and Claffy, K. (2002). k claffy. distance metrics in the internet. In *in IEEE International Telecommunications Symposium*, Natal, Brazil. Citeseer.
- [Huszak and Imre, 2010] Huszak, A. and Imre, S. (2010). Analysing GOP structure and packet loss effects on error propagation in MPEG-4 video streams. In *Communications, Control and Signal Processing (ISCCSP), 2010 4th International Symposium on*, Limassol, Cyprus. IEEE.
- [Huynh-Thu and Ghanbari, 2008] Huynh-Thu, Q. and Ghanbari, M. (2008). Scope of validity of PSNR in image/video quality assessment. *Electronics letters*, 44(13):800–801.
- [ISO/IEC, 1998] ISO/IEC (1998). Quality of service: Framework”. ISO/IEC 13236.
- [ITU, 1865] ITU (1865). International Telecommunication Union. [www.itu.int](http://www.itu.int).
- [Jaffe, 1984] Jaffe, J. (1984). Algorithms for finding paths with multiple constraints. *Networks*, 14(1):95–116.
- [Jaksch et al., 2010] Jaksch, T., Ortner, R., and Auer, P. (2010). Near-optimal regret bounds for reinforcement learning. *The Journal of Machine Learning Research*, 11:1563–1600.
- [Jamin et al., 2000] Jamin, S., Jin, C., Jin, Y., Raz, D., Shavitt, Y., and Zhang, L. (2000). On the placement of internet instrumentation. In *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, Tel Aviv, Israel. IEEE.
- [Jamin et al., 2001] Jamin, S., Jin, C., Kurc, A., Raz, D., and Shavitt, Y. (2001). Constrained mirror placement on the internet. In *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, Anchorage, Alaska. IEEE.
- [Janowski and Romaniak, 2010] Janowski, L. and Romaniak, P. (2010). QoE as a function of frame rate and resolution changes. *Future Multimedia Networking*, 6157:34–45.
- [Jumisko et al., 2005] Jumisko, S., Ilvonen, V., and Väänänen-Vainio-Mattila, K. (2005). Effect of TV content in subjective assessment of video quality on mobile devices. In *Proceedings of SPIE*, volume 5684, pages 243–254.
- [Kaarainen et al., 2001] Kaarainen, H., Ahtiainen, A., Laitinen, L., Naghian, S., and Niemi, V. (2001). *UMTS networks*. Wiley Online Library.

- [Kaelbling et al., 1996] Kaelbling, L. P., Littman, M. L., and Moore, A. W. (1996). Reinforcement Learning: A survey. *Journal of AI Research*, 4:237–285.
- [Karger et al., 1999] Karger, D., Sherman, A., Berkheimer, A., Bogstad, B., Dhanidina, R., Iwamoto, K., Kim, B., Matkins, L., and Yerushalmi, Y. (1999). Web caching with consistent hashing. *Computer Networks*, 31(11-16):1203–1213.
- [Kargin, 2005] Kargin, V. (2005). On the chernoff bound for efficiency of quantum hypothesis testing. *The Annals of Statistics*, 33(2):959–976.
- [Kilikki, 2002] Kilikki, K. (2002). Differentiated services for the Internet. *Interpretation*, 1:00–99.
- [Kim and Choi, 2010] Kim, H. and Choi, S. (2010). A study on a QoS/QoE correlation model for QoE evaluation on IPTV service. In *The 12th International Conference on Advanced Communication Technology (ICACT)*, Gangwon-Do, Korea. IEEE.
- [Kim et al., 2008] Kim, H. J., Lee, D. H., Lee, J. M., Lee, K. H., Lyu, W., and Choi, S. G. (2008). The qoe evaluation method through the qos-qoe correlation model. In *Fourth International Conference on Networked Computing and Advanced Information Management*, Gyeongju, Korea.
- [Kingman, 1993] Kingman, J. (1993). *Poisson processes*. Wiley Online Library.
- [Kleinberg et al., 2008] Kleinberg, R., Slivkins, A., and Upfal, E. (2008). Multi-armed bandits in metric spaces. In *Proceedings of the 40th annual ACM symposium on Theory of computing*, pages 681–690, Victoria, British Columbia, Canada. ACM.
- [Kocsis and Szepesvári, 2006] Kocsis, L. and Szepesvári, C. (2006). Bandit based monte-carlo planning. *Machine Learning: ECML 2006*, 4212:282–293.
- [Kohler et al., 2006] Kohler, E., Handley, M., and Floyd, S. (2006). Designing DCCP: Congestion control without reliability. *ACM SIGCOMM Computer Communication Review*, 36(4):27–38.
- [Koonen, 2006] Koonen, T. (2006). Fiber to the home/fiber to the premises: what, where, and when? *Proceedings of the IEEE*, 94(5):911–934.
- [Koulouriotis and Xanthopoulos, 2008] Koulouriotis, D. and Xanthopoulos, A. (2008). Reinforcement learning and evolutionary algorithms for non-stationary multi-armed bandit problems. *Applied Mathematics and Computation*, 196(2):913–922.
- [Krishnan et al., 2000] Krishnan, P., Raz, D., and Shavitt, Y. (2000). The cache location problem. *IEEE/ACM Transactions on Networking (TON)*, 8(5):568–582.
- [Kuipers and Van Mieghem, 2005] Kuipers, F. and Van Mieghem, P. (2005). Conditions that impact the complexity of qos routing. *IEEE/ACM Transactions on Networking (TON)*, 13(4):717–730.

- [Kumar and Miikkualainen, 1998] Kumar, S. and Miikkualainen, R. (1998). Confidence-based q-routing: an on-queue adaptive routing algorithm. In *Proceedings of Neural Networks in Engineering*.
- [Kurose and Ross, 2005] Kurose, J. and Ross, K. (2005). *Computer networking*. Pearson/Addison Wesley.
- [Lai and Robbins, 1985] Lai, T. and Robbins, H. (1985). Asymptotically efficient adaptive allocation rules. *Advances in applied mathematics*, 6(1):4–22.
- [Lemma et al., 2010] Lemma, E., Hussain, S., and Anjelo, W. (2010). Performance Comparison of EIGRP/IS-IS and OSPF/IS-IS. Master’s thesis, Blekinge Tekniska Hogskolan, Sweden.
- [Limelight, 2001] Limelight (2001). Limelight. <http://www.limelight.com>.
- [Littman, 1994] Littman, M. (1994). Markov games as a framework for multi-agent reinforcement learning. In *Proceedings of the eleventh international conference on machine learning*, New Brunswick, NJ, USA.
- [Lopes Gomes et al., 2011] Lopes Gomes, R., Moreira Junior, W., Cerqueira, E., and Jorge Abelem, A. (2011). Using fuzzy link cost and dynamic choice of link quality metrics to achieve QoS and QoE in wireless mesh networks. *Journal of Network and Computer Applications*, 34(2):506–516.
- [Luce, 1959] Luce, R. (1959). *Individual choice behavior*. John Wiley.
- [Ma and Steenkiste, ] Ma, Q. and Steenkiste, P. On path selection for traffic with bandwidth guarantees. In *Network Protocols, 1997. Proceedings., 1997 International Conference on*, Atlanta, GA, USA. IEEE.
- [Majd et al., 2009] Majd, G., Cesar, V., and Adlen, K. (2009). An Adaptive Mechanism For Multipath Video Streaming Over Video Distribution Network (VDN). *First International Conference on Advances in Multimedia*.
- [Mannor and Tsitsiklis, 2004] Mannor, S. and Tsitsiklis, J. (2004). The sample complexity of exploration in the multi-armed bandit problem. *The Journal of Machine Learning Research*, 5:623–648.
- [Mao et al., 2002] Mao, Z., Cranor, C., Douglis, F., Rabinovich, M., Spatscheck, O., and Wang, J. (2002). A precise and efficient evaluation of the proximity between web clients and their local dns servers. In *USENIX Annual Technical Conference*, Monterey, Canada.
- [Markaki et al., 2007] Markaki, O., Charilas, D., and Nikitopoulos, D. (2007). Enhancing quality of experience in next generation networks through network selection mechanisms. In *Personal, Indoor and Mobile Radio Communications, 2007. PIMRC 2007. IEEE 18th International Symposium on*, pages 1–5, Athens, Greece. IEEE.

- [Marković et al., 2007] Marković, G., Teodorović, D., and Aćimović-Raspopović, V. (2007). Routing and wavelength assignment in all-optical networks based on the bee colony optimization. *AI Communications*, 20(4):273–285.
- [Martinez-Yelmo et al., 2010] Martinez-Yelmo, I., Seoane, I., and Guerrero, C. (2010). Fair Quality of Experience (QoE) measurements related with networking technologies. In *Wired/Wireless Internet Communications*, Lulea, Sweden.
- [Maryland Routing Simulator, 1991] Maryland Routing Simulator (1991). Maryland Routing Simulator. <http://www.cs.umd.edu/projects/netcalliper/software.html>.
- [Mellouk, 1994] Mellouk, A. (1994). *Etude des Sytemes modulaires Neuro-Preditif: application a la Reconnaissance Automatique de la Palire Continue*. PhD thesis, University Paris-Sud, Orsay, France.
- [Mellouk et al., 2009] Mellouk, A., Hoceini, S., and Zeadally, S. (2009). Design and performance analysis of an inductive QoS routing algorithm. *Computer Communications*, 32(1371-1376).
- [Mellouk et al., 2011] Mellouk, A., Hoceini, S., and Zeadally, S. (2011). A bio-inspired quality of service (qos) routing algorithm. *Communications Letters, IEEE*, 15(9):1016–1018.
- [Mellouk et al., 2012] Mellouk, A., Hoceini, S., and Zeadally, S. (2012). A state-dependent time evolving multi-constraint routing algorithm. *ACM Transactions on Autonomous and Adaptive Systems (TAAS), ACM*, 8(1).
- [Mersereau et al., 2009] Mersereau, A., Rusmevichientong, P., and Tsitsiklis, J. (2009). A structured multiarmed bandit problem and the greedy policy. *Automatic Control, IEEE Transactions on*, 54(12):2787–2802.
- [Michael and Johnson, 1979] Michael, R. and Johnson, D. (1979). Computers and intractability: a guide to the theory of np-completeness. *WH Freeman & Co., San Francisco*.
- [Moebs, 2008] Moebs, S. (2008). A learner , is a learner ,is a user ,is a customer - qos-based experience-aware adaptation. In *16th ACM International Conference on Multimedia*, Vancouver, Canada.
- [Moy, 2008] Moy, J. (2008). *OSPF complete implementation*. Addison-Wesley Professional.
- [Network Simulator, 1989] Network Simulator (1989). The Network Simulator. <http://www.isi.edu/nsnam/ns/>.
- [NLANR, 2006] NLANR (2006). National Laboratory for Applied Network Research. <http://www.nlanr.net/>.

- [Nuaymi, 2007] Nuaymi, L. (2007). *WiMAX: technology for broadband wireless access*. Wiley.
- [Okamura et al., 2009] Okamura, H., Dohi, T., and Trivedi, K. (2009). Markovian arrival process parameter estimation with group data. *IEEE/ACM Transactions on Networking (TON)*, 17(4):1326–1339.
- [Pan et al., 2003] Pan, J., Hou, Y., and Li, B. (2003). An overview of dns-based server selections in content distribution networks. *Computer Networks*, 43(6):695–711.
- [Partridge, 1992] Partridge, C. (1992). A proposed flow specification. Technical report, RFC 1363.
- [Patel and Yadav, 2012] Patel, A. and Yadav, M. (2012). Optimization of routing algorithm using fuzzy neural network. *Asian Journal of Computer Science and Information Technology*, 2(7).
- [Pathan and Buyya, 2008] Pathan, M. and Buyya, R. (2008). A taxonomy of cdns. *Content delivery networks*, pages 33–77.
- [PC Mag, 1994] PC Mag (1994). PC Magazine Digital Edition. <http://www.pcmag.com/>.
- [Peng, 2004] Peng, G. (2004). Cdn: Content distribution network. Technical report, Experimental Computer Systems Lab, Department of Computer Science, State University of New York, NY, USA.
- [Pierre and Van Steen, 2006] Pierre, G. and Van Steen, M. (2006). Globule: a collaborative content delivery network. *Communications Magazine, IEEE*, 44(8):127–133.
- [Qiu et al., 2001] Qiu, L., Padmanabhan, V., and Voelker, G. (2001). On the placement of web server replicas. In *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, Anchorage, Alaska. IEEE.
- [Radlinski et al., 2008] Radlinski, F., Kleinberg, R., and Joachims, T. (2008). Learning diverse rankings with multi-armed bandits. In *Proceedings of the 25th international conference on Machine learning*, California, USA. ACM.
- [Rajagopalan et al., 1998] Rajagopalan, B., Nair, R., Sandick, H., and Crawley, E. (1998). A framework for qos-based routing in the internet. *Request for Comments (RFC)*, 2386.
- [Ramalho, 2000] Ramalho, M. (2000). Intra-and inter-domain multicast routing protocols: A survey and taxonomy. *Communications Surveys & Tutorials, IEEE*, 3(1):2–25.
- [Recommendation, 1994] Recommendation (1994). ITU E.800: Terms and Definitions Related to Quality of Service and Network Performance Including Dependability. *ITU Telecom*.

- [Recommendation, 1996] Recommendation (1996). ITU-T P.800: Methods for subjective determination of transmission quality. *ITU Telecom*.
- [Recommendation, 1999] Recommendation (1999). ITU-T P.910: Subjective Video Quality Assessment Methods for Multimedia Applications. *ITU Telecom*.
- [Recommendation, 2001a] Recommendation (2001a). ITU-T G.1010: End-user multimedia QoS categories. *ITU Telecom*.
- [Recommendation, 2001b] Recommendation (2001b). ITU-T P.862: Perceptual evaluation of speech quality (PESQ): An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs . *ITU Telecom*.
- [Recommendation, 2002] Recommendation (2002). ITU-R BT.500-11: Methodology for the Subjective Assessment of the Quality of Television Pictures. *ITU Telecom*.
- [Recommendation, 2004a] Recommendation (2004a). Itu-r bt.1683: Objective perceptual video quality measurement techniques for standard definition digital broadcast television in the presence of a full reference. *ITU, 1683*.
- [Recommendation, 2004b] Recommendation (2004b). ITU-T J.144: Objective perceptual video quality measurement techniques for digital cable television in the presence of a full reference. *ITU Telecom*.
- [Recommendation, 2005] Recommendation (2005). ITU-T G.1030: Estimating end-to-end performance in IP networks for data applications. *ITU Telecom*.
- [Recommendation, 2006] Recommendation (2006). ITU P.801: Mean Opinion Score (MOS) Terminology. *ITU Telecom*.
- [Recommendation, 2007] Recommendation (2007). ITU-T G.1070: Opinion model for video-telephony applications. *ITU Telecom*.
- [Recommendation, 2008a] Recommendation (2008a). ITU-T J.246: Perceptual visual quality measurement techniques for multimedia services over digital cable television networks in the presence of a reduced bandwidth reference. *ITU Telecom*.
- [Recommendation, 2008b] Recommendation (2008b). ITU-T J.247 Objective perceptual multimedia video quality measurement in the presence of a full reference. *ITU Telecom*.
- [Recommendation, 2010] Recommendation (2010). ITU-T J.249: Perceptual video quality measurement techniques for digital cable television in the presence of a reduced reference. *ITU Telecom*.
- [Recommendation, 2011a] Recommendation (2011a). ITU-T G.1000 Qualité de service des communications : cadre et définitions. *ITU Telecom*.
- [Recommendation, 2011b] Recommendation (2011b). ITU-T G.107: The E-model: a computational model for use in transmission planning. *ITU Telecom*.

- [Recommendation, 2011c] Recommendation (2011c). ITU-T J.342: Objective multimedia video quality measurement of HDTV for digital cable television in the presence of a reduced reference signal. *ITU Telecom*.
- [Rétvári et al., 2009] Rétvári, G., Németh, F., Chaparadza, R., and Szabó, R. (2009). Ospf for implementing self-adaptive routing in autonomic networks: A case study. *Modelling Autonomic Communications Environments*, pages 72–85.
- [Ringserver, 2009] Ringserver (2009). RingServer project. <http://www.ring.gr.jp/>.
- [Rix, 2002] Rix, A. (2002). Advances in objective quality assessment of speech over analogue and packet based networks. In *Data Compression: Methods and Implementations (Ref. No. 1999/150)*, *IEE Colloquium on*, page 10. IET.
- [Rix et al., 2002] Rix, A., Hollier, M., Hekstra, A., and Beerends, J. (2002). Perceptual evaluation of speech quality (PESQ): The new ITU standard for end-to-end speech quality assessment part I-time-delay compensation. *Journal of the Audio Engineering Society*, 50(10):755–764.
- [Robbins, 1952] Robbins, H. (1952). Some aspects of the sequential design of experiments. *Bulletin of the American Mathematical Society*, 58(5):527–535.
- [Rohaly et al., 2000] Rohaly, A., Libert, J., Corriveau, P., Webster, A., et al. (2000). Final report from the video quality experts group on the validation of objective models of video quality assessment. *ITU-T Standards Contribution COM*, pages 9–80.
- [Rousskov and Wessels, 1998] Rousskov, A. and Wessels, D. (1998). Cache digests. *Computer Networks and ISDN Systems*, 30(22-23):2155–2168.
- [Rubino, 2005] Rubino, G. (2005). Quantifying the Quality of Audio and Video Transmissions over the Internet: The PSQA Approach. In *Design and operations of communication networks: a review of wired and wireless modeling and management challenges*, London, UK. Imperial College Press.
- [Rusmevichientong and Tsitsiklis, 2010] Rusmevichientong, P. and Tsitsiklis, J. (2010). Linearly parameterized bandits. *Mathematics of Operations Research*, 35(2):395–411.
- [Schaul et al., 2010] Schaul, T., Bayer, J., Wierstra, D., Sun, Y., Felder, M., Sehnke, F., Rückstieß, T., and Schmidhuber, J. (2010). Pybrain. *The Journal of Machine Learning Research*, 11:743–746.
- [Schuler et al., 2009] Schuler, W., Bastos-Filho, C., and Oliveira, A. (2009). A novel hybrid training method for hopfield neural networks applied to routing in communications networks. *International Journal of Hybrid Intelligent Systems*, 6(1):27–39.
- [Shaikh et al., 2001] Shaikh, A., Tewari, R., and Agrawal, M. (2001). On the effectiveness of dns-based server selection. In *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, Anchorage, Alaska. IEEE.

- [Shaikh et al., 2010a] Shaikh, J., Fiedler, M., and Collange, D. (2010a). Quality of Experience from user and network perspectives. *Annals of Telecommunications*, 65(1):47–57.
- [Shaikh et al., 2010b] Shaikh, J., Minhas, T., Arlos, P., and Fiedler, M. (2010b). Evaluation of delay performance of traffic shapers. In *Security and Communication Networks (IWSCN), 2010 2nd International Workshop on*, Karlstad, Sweden. IEEE.
- [Sharma et al., 2010] Sharma, M., Kaur, H., et al. (2010). Performance evaluation of hybrid network using rip & igrp for different applications. In *High-Capacity Optical Networks and Enabling Technologies (HONET), 2010*, Cairo, Egypt. IEEE.
- [Soldani et al., 2006] Soldani, D., Li, M., and Cuny, R. (2006). *QoS and QoE management in UMTS cellular systems*. Wiley Online Library.
- [Stallings, 2001] Stallings, W. (2001). Multi-protocol label switching (mpls). *White paper, Cisco Systems*.
- [Starr et al., 1999] Starr, T., Cioffi, J., and Silverman, P. (1999). *Understanding digital subscriber line technology*. Prentice Hall PTR.
- [Stewart and Metz, 2001] Stewart, R. and Metz, C. (2001). SCTP: new transport protocol for TCP/IP. *Internet Computing, IEEE*, 5(6):64–69.
- [Sutton and Barto, 1998] Sutton, R. S. and Barto, A. G. (1998). *Reinforcement Learning: An introduction*. MIT Press, Cambridge.
- [Szymaniak et al., 2003] Szymaniak, M., Pierre, G., and Van Steen, M. (2003). Netairt: A flexible redirection system for apache. In *International Conference WWW/Internet (ICWI)*, Algarve, Portugal.
- [Takahashi et al., 2008] Takahashi, A., Hands, D., and Barriac, V. (2008). Standardization activities in the ITU for a QoE assessment of IPTV. *Communications Magazine, IEEE*, 46(2):78–84.
- [Tanner and White, 2009] Tanner, B. and White, A. (2009). Rl-gluе: Language-independent software for reinforcement-learning experiments. *The Journal of Machine Learning Research*, 10:2133–2136.
- [Teodorović and Dell’Orco, 2005] Teodorović, D. and Dell’Orco, M. (2005). Bee colony optimization—a cooperative learning approach to complex transportation problems. In *Advanced OR and AI Methods in Transportation: Proceedings of 16th Mini-EURO Conference and 10th Meeting of EWGT (13-16 September 2005)*.—Poznan: Publishing House of the Polish Operational and System Research.
- [TM forum, 1988] TM forum (1988). TM Forum. <http://www.tmforum.org/>.
- [Ton Roosendaal, 2010] Ton Roosendaal (2010). Sintel Video Trailer. <http://www.sintel.org/>.

- [Tran et al., 2011] Tran, H., Mellouk, A., and Hoceini, S. (2011). User to User adaptive routing based on QoE. In *10th International Symposium on Programming and Systems (ISPS)*, Algiers, Algeria. IEEE.
- [Tran et al., 2010] Tran, H., Mellouk, A., Hoceini, S., Augustin, B., and Marty, J. (2010). QoE-based concepts and models for end-to-end network services based on user perception. Technical report, Laboratory LISSI, University Paris-Est Creteil.
- [Trinh et al., 2010] Trinh, T., Sonkoly, B., and Molnár, S. (2010). On the fairness characteristics of Fast TCP. *Annals of Telecommunications*, 65(1):73–85.
- [Vallopillil and Ross, 1998] Vallopillil, V. and Ross, K. (1998). Internet draft: Cache array routing protocol v1. 0. <http://icp.ircache.net/carp.txt>.
- [van den Branden Lambrecht, 2002] van den Branden Lambrecht, C. (2002). Color moving pictures quality metric. In *International Conference on Image Processing*, New York, USA. IEEE.
- [van den Lambrecht, 1996] van den Lambrecht, C. (1996). *Perceptual models and architectures for video coding applications*. PhD thesis, École Polytechnique Fédérale de Lausanne (EPFL), Switzerland.
- [Van der Auwera et al., 2008] Van der Auwera, G., David, P., and Reisslein, M. (2008). Traffic and quality characterization of single-layer video streams encoded with the H. 264/MPEG-4 advanced video coding standard and scalable video coding extension. *Broadcasting, IEEE Transactions on*, 54(3):698–718.
- [Vermorel and Mohri, 2005] Vermorel, J. and Mohri, M. (2005). Multi-armed bandit algorithms and empirical evaluation. In *Machine Learning: ECML 2005*, Porto, Portugal. Springer.
- [VLC, 2001] VLC (2001). Video LAN Organization. <http://www.videolan.org/>.
- [Vleeschauwer et al., 2008] Vleeschauwer, B. D., Turck, F. D., Dhoedt, B., Demeester, P., Wijnants, M., and Lamotte, W. (2008). End-to-end QoE Optimization Through Overlay Network Deployment. In *International Conference on Information Networking*, Busan, Korea.
- [Voran, 1997] Voran, S. (1997). Estimation of perceived speech quality using measuring normalizing blocks. In *Workshop on Speech Coding For Telecommunications*, Pennsylvania, USA. IEEE.
- [Voran, 2002] Voran, S. (2002). The development of objective video quality measures that emulate human perception. In *Global Telecommunications Conference, 1991. GLOBECOM'91. Countdown to the New Millennium. Featuring a Mini-Theme on: Personal Communications Services*, Phoenix, Arizona. IEEE.

- [Wang et al., 2008] Wang, Y., Munos, R., and Audibert, J. (2008). Algorithms for infinitely many-armed bandits. In *Neural Information Processing Systems*, Vancouver, Canada.
- [Wang et al., 2004] Wang, Z., Bovik, A., Sheikh, H., and Simoncelli, E. (2004). Image quality assessment: From error visibility to structural similarity. *Image Processing, IEEE Transactions on*, 13(4):600–612.
- [Wang and Crowcroft, 1995] Wang, Z. and Crowcroft, J. (1995). Bandwidth-delay based routing algorithms. In *Global Telecommunications Conference. GLOBECOM'95*, Singapore, Singapore. IEEE.
- [Wang and Crowcroft, 1996] Wang, Z. and Crowcroft, J. (1996). Quality-of-service routing for supporting multimedia applications. *Selected Areas in Communications, IEEE Journal on*, 14(7):1228–1234.
- [Watkins and Daylan, 1992] Watkins and Daylan (1992). Technical note: Q-learning. *Machine Learning*, 8(3):279–292.
- [Watkins and Dayan, 1992] Watkins, C. and Dayan, P. (1992). Q-learning. *Machine learning*, 8(3):279–292.
- [Watkins, 1989] Watkins, H. (1989). *Learning from Delayed Rewards*. PhD thesis, Cambridge University.
- [Waxman, 1993] Waxman, B. (1993). Performance evaluation of multipoint routing algorithms. In *INFOCOM'93. Proceedings. Twelfth Annual Joint Conference of the IEEE Computer and Communications Societies. Networking: Foundation for the Future. IEEE*, San Francisco, CA, USA. IEEE.
- [Welch and Clark, 2006] Welch, J. and Clark, J. (2006). RFC4445 - A Proposed Media Delivery Index (MDI). *Request for comments*.
- [Wendell et al., 2010] Wendell, P., Jiang, J., Freedman, M., and Rexford, J. (2010). Donar: decentralized server selection for cloud services. In *ACM SIGCOMM Computer Communication Review*, volume 40, pages 231–242. ACM.
- [Wessels and Claffy, 1997] Wessels, D. and Claffy, K. (1997). Application of internet cache protocol (icp), version 2.
- [Wijnants et al., 2010] Wijnants, M., Lamotte, W., De Vleeschauwer, B., De Turck, F., Dhoedt, B., Demeester, P., Lambert, P., Van de Walle, D., De Cock, J., Notebaert, S., et al. (2010). Optimizing user quality of experience through overlay routing, bandwidth management and dynamic transcoding. *International Journal of Adaptive, Resilient and Automatic Systems*, 1(2):64.
- [Wikipedia, 2001] Wikipedia (2001). Wikipedia, the free encyclopedia. <http://wikipedia.org>.

- [Winkler and Mohandas, 2008] Winkler, S. and Mohandas, P. (2008). The evolution of video quality measurement: from PSNR to hybrid metrics. *Broadcasting, IEEE Transactions on*, 54(3):660–668.
- [Yamada et al., 2007] Yamada, H., Fukumoto, N., Isomura, M., Uemura, S., and Hayashi, M. (2007). A qoe based service control scheme for racf in ip-based fmc networks. In *E- Commerce Technology and the 4th IEEE International Conference on Enterprise Computing*, Tokyo, Japan.
- [Yang, 1999] Yang, W. (1999). *Enhanced modified bark spectral distortion (EMBSD): An objective speech quality measure based on audible distortion and cognition model*. PhD thesis, Temple University.
- [Yin et al., 2010] Yin, H., Liu, X., Min, G., and Lin, C. (2010). Content delivery networks: a bridge between emerging applications and future IP networks. *Network, IEEE*, 24(4):52–56.
- [Yoshikawa et al., 1997] Yoshikawa, C., Chun, B., Eastham, P., Vahdat, A., Anderson, T., and Culler, D. (1997). Using smart clients to build scalable services. In *Proceedings of the 1997 USENIX Technical Conference*, California, USA.
- [Zhang and Ansari, 2011] Zhang, J. and Ansari, N. (2011). On assuring end-to-end QoE in next generation networks: challenges and a possible solution. *Communications Magazine, IEEE*, 49(7):185–191.
- [Zhang et al., 2007] Zhang, J., Hu, H., Wang, T., and Tang, H. (2007). QOSPF-improved routing algorithm with bandwidth and delay metrics. *Beijing Youdian Daxue Xuebao/ Journal of Beijing University of Posts and Telecommunication*, 30(4):134–138.
- [Zheng et al., 2001] Zheng, W., Ahmad, I., and Liou, M. (2001). Benchmark the software based MPEG-4 video codec. In *Electronics, Circuits and Systems, 2001. ICECS 2001. The 8th IEEE International Conference on*, Malta. IEEE.
- [Zhou et al., 2008] Zhou, X., Dreibholz, T., and Rathgeb, E. (2008). A new server selection strategy for reliable server pooling in widely distributed environments. In *Digital Society, 2008 Second International Conference on the*, Sainte Luce, Martinique. IEEE.
- [Zinner et al., 2010] Zinner, T., Abboud, O., Hohlfeld, O., Hossfeld, T., and Tran-Gia, P. (2010). Towards QoE management for scalable video streaming. In *21th ITC Specialist Seminar on Multimedia Applications-Traffic, Performance and QoE*, Miyazaki, Japan.