Real-Time State-Dependent routing based on User Perception

Hai Anh TRAN, Abdelhamid MELLOUK,
University of Paris-Est Creteil Val de Marne (UPEC)
Image, Signal and Intelligent Systems Lab-LiSSi Lab
Transport Infrastructure and Network Control Group - TINC
122 rue Paul Armangot, 94400 Vitry sur Seine, France
{hai-anh.tran, mellouk}@u-pec.fr

***Abstract*—In order to successfully resolve the network infrastructure's problems the network provider has to improve the service quality. However in traditional ways, maintaining and improving of the service quality are generally determined in terms of quality of service criteria, not in terms of satisfaction and perception to the end-user. The latter is represented by Quality of Experience (QoE) that becomes recently the most important tendency to guarantee the quality of network services. QoE represents the subjective perception of end-users using network services with network functions such as admission control, resource management, routing, traffic control, etc. In this paper, we focus on routing mechanism driven by QoE end-users. Today, NP-complete is one of the most routing algorithm problems when trying to satisfy multi QoS constraints criteria simultaneously. In order to avoid the classification problem of these multiple criteria reducing the complexity problem for the future Internet, we propose two protocols based on user QoE measurement in routing paradigm to construct an adaptive and evolutionary system. Our first approach is a routing driven by terminal QoE basing on a least squares reinforcement learning technique called Least Squares Policy Iteration. The second approach, namely QQAR (QoE Q-learning based Adaptive Routing), is a improvement of the first one. QQAR basing on Q-Learning, a Reinforcement Learning algorithm, uses Pseudo Subjective Quality Assessment (PSQA), a real-time QoE assessment tool based on Random Neural Network, to evaluate QoE. Experimental results showed a significant performance against over other traditional routing protocols.**

***Index Terms*—Quality of Service (QoS), Quality of Experience (QoE), Network Services, Routing System, Autonomous System, Pseudo Subjective Quality Assessment (PSQA), Reinforcement Learning.**

I. Introduction

In his position, the customer is able to choose between different competing network service providers. Except of pricing schemes which are a decision aid for many users, the user choices are influenced by expected and experienced quality. Consequently, the providers interest mainly in how users perceive network services. In fact, providers have to observe and react quickly on quality problems before the customer perceives them and considers churn. Based on this kind of quality competition, the new term of Quality of Experience (QoE) has been introduced, combining user perception, experience and expectations without technical parameters such as QoS parameters. In fact, the network provider's aim is to provide a good user experience at minimal network resource usage. It is important from the network operator to be aware of the degree of influence of each network's factor on the user perception. For users, also for operators and Internet service providers, the end-to-end quality is one of the major factors to be achieved. QoE takes into account the needs and the desires of the subscribers when using network services, while the concept of QoS just attempts to objectively measure the service delivered. Furthermore, e2e QoS with more than two non correlated criteria is NP-complete [1]. With the evolution of the Internet, both technologies and needs continue to develop, so complexity and cost become limiting factors in the future evolution of networks. In order to reduce this complexity problem, one has integrated QoE in network systems. Firstly, 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. Secondly, with QoE paradigm, we can reach a better solution and prevent the NP-complete problem because our goal is just maintaining QoE criteria instead of optimizing multiple QoS criteria.

In the recent years, there are many researches, proposals that are made in order to measure, evaluate, and improve QoE in networks. Our work aimed to construct an adaptive routing method that can retrieve environment information and adapt to the environment changes. This adaptive routing mechanism maintaining the required QoE of end-users is very necessary for network systems that have great dynamics (i.e. unreliable communication) and multiple user profiles where the required QoE levels are different. For better user's perception, it is preferable that a routing protocol adapts itself to these QoE levels.

Our two proposals are routing systems driven by terminal QoE based on Reinforcement Learning (RL) concept [2] [3]. They aimed to maintain user satisfaction using QoE feedback of end-users. The first algorithm is based on Least Squares Policy Iteration (LSPI) [4], a RL technique that combines least squares function approximation with policy iteration. The second algorithm, an improvement of the first one, is based on Q-Learning [3] which is one of the RL algorithms. However, native Q-Learning taking into account rewards at all nodes

in the system is inadequate to our target problem because the QoE reward is available only at the last node (QoE is evaluated at end-users). In order to improve the first algorithm, we evaluate the QoE at any node in the whole system. The QoE measurement is realized by using a hybrid between subjective and objective evaluation method called Pseudo Subjective Quality Assessment (PSQA) tool [5]. We now present briefly preliminaries of PSQA tool and Q-Learning algorithm and explain the reason for which we have chose them to our approach.

PSQA tool measures QoE in an automatic and efficient way, such that it can be done in real time. It consists of training a Random Neural Network (RNN) to behave as a human observer and to deliver a numerical evaluation of quality, which must be close to the average value that a set of real human observers would give to the received streams. PSQA method includes the following steps: a) Identifying a set of parameters having an important impact on the perceived quality, b) Building a platform allowing to send a video sequence through an IP connection, c) Performing a subjective testing experiment, d) Training the RNN.

PSQA used RNN for the learning phase. Sequences are input data of RNN that will give a real function as output. So for any configuration, the function returns a number close to the associated MOS (Mean Opinion Score)[1] value [6]. Our testbed using PSQA tool is described in detail in section IV. Using PSQA method gives us a function f expressed as:

$$f : R^3 \rightarrow R \tag{1}$$

This function f takes a combination of three parameter values as mentioned above (*delay time*, *loss rate* and *conditional loss rate*) to obtain a single output equivalent to the appropriate MOS score. So from now onwards the expression of "applying PSQA tool" means using function f in Equation 1 with three parameters as input to obtain the MOS score.

Regarding the Q-Learning algorithm [7], it is a technique to solve specific problems by using the RL approach [3]. In 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.

One of the most important breakthroughs in RL is an off-policy temporal difference (TD) control algorithm known as Q-learning. One-step Q-learning is defined as:

$$Q(s,a) = (1-\alpha)Q(s,a) + \alpha[r + \gamma \max_{a'} Q(s',a')] \tag{2}$$

where $Q(s,a)$ is the Q-value of state s when choosing action a. s' is the next state of s. α represents the learning rate, which models the rate of updating Q-values. γ is the discount factor. $\max_{a'} Q(s',a')$ is the maximum Q-value of the next state.

The paper is structured as follows: Section II surveys briefly related works. We present our approaches in section III. The experimental results are shown in section IV. Paper is ended with conclusion and some future works in section V.

[1]Mean Opinion Score (MOS) gives a numerical indication of the perceived quality of the media 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

II. Related work

Routing mechanism is key to the success of large-scale, distributed communication and heterogeneous networks. However the goal of every traditional algorithm is to maximize many QoS criteria simultaneously. So they meet the NP-complete problem as we mentioned before. We present in this section some related works.

The idea of applying reinforcement learning to routing in networks was firstly introduced by [8]. Authors described the Q-routing algorithm for packet routing. Reinforcement learning module is embedded into each node of a switching network. In [8], each node to keep accurate statistics on which routing decisions lead to minimal delivery times uses only local communication. However, this proposal focus on optimizing only one basis QoS metric (delivery times). So user perception (QoE) is not yet considered in this approach. [9] proposed an application of gradient ascent algorithm for RL to a complex domain of packet routing in network communication. This approach updates the local policies while avoiding the necessity for centralized control or global knowledge of the networks structure. The only global information required by the learning algorithm is the network utility expressed as a reward signal distributed once in an epoch and dependent on the average routing time. In [10], K-Optimal path Q-Routing Algorithm (KOQRA) is presented as a QoS based routing algorithm based on a multi-path routing approach combined with the Q-routing algorithm. The global 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 [11] 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 all of these approaches above do not take into account the perception and satisfaction of end-users. In other words, QoE concept is ignored. That poses the problem of choosing the best QoS metric that is often complex. However QoE comes directly from the use and represents the true criteria to optimize. In taking into account this lack, other proposals are presented in [12].

In [13], authors presented an overlay network for end-to-end QoE management. The purpose is QoE optimization by routing around failures in the IP network and 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, their proposal does not use any adaptive mechanism.

[14] presented a new 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 measured using PSQA tool, the same measurement tool we have used. After receiving a client demand, the video server chooses an initial strategy and an initial scheme to start the video streaming. Then, client uses PSQA to evaluate the QoE of the received video in real time and sends this feedback to server. After examining this feedback, the video server will decide to keep or to change its strategies. This approach has well considered end-users perception. However the adaptive mechanism is quite simple because it is not based on the learning method. Furthermore, the problem of this approach is the fact that authors use source routing.

Instead of trying to optimizing many QoS criteria like approaches above, our algorithm just based on user perception (QoE). We present in the next section our two approaches driven by terminal QoE.

III. Proposed Approaches

Our idea to take into account end-to-end QoE consists to develop adaptive mechanisms that can retrieve the information from their environment (QoE) and adapt to initiate actions. The action choice should be executed in response to end-users feedback, in other words the QoE feedback.

Concretely, the system integrates the QoE measurement

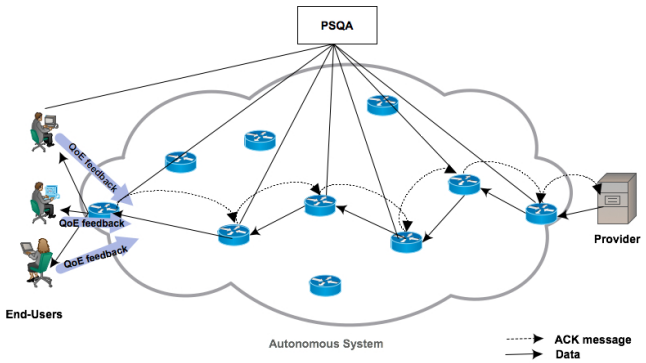


Fig. 1: QQAR routing system

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. 1). So in that way, the routing process is build according to maintaining the best user perception.

In this section, we present our two approaches: our routing system driven by terminal QoE and his improvement, QQAR algorithm.

A. Routing system driven by terminal QoE

In order to integrate RL notion into our routing system, we have mapped RL model to our routing model in the context of learning routing strategy. We consider each router in the system as a state. The states are arranged along the routing path. Furthermore, we consider each link emerging from a router as an action to choose. The system routing mechanism corresponds to the policy π.

After data reach end-users, QoE evaluation is realized to give a QoE feedback to the system. We consider this feedback as environment reward and our purpose is to improve the policy π using this QoE feedback. Concretely, the policy π is chosen so that it is equal to $argmax$ of action value function Q in policy π:

$$\pi_{t+1}(s_t) = argmax Q^{\pi_t}(s_t, a) \tag{3}$$

Least-Squares Policy Iteration (LSPI) [4] is a recently introduced reinforcement learning method. Our choice is based on the fact that this technique learns the weights of the linear functions, thus can update the Q-values based on the most updated information regarding the features. It does not need carefully tuning initial parameters (e.g., learning rate). Furthermore, LSPI converges faster with less samples than basic Q-learning.

In this technique, instead to calculate directly action-value function Q, this latter is approximated with a parametric function approximation. In other words, the value function is approximated as a linear weighted combination:

$$\hat{Q}^{\pi}(s, a, \omega) = \sum_{i=1}^{k} \phi_i(s, a)\omega_i = \phi(s, a)^T \omega \tag{4}$$

where $\phi(s, a)$ is the basis features vector and ω is weight vector in the linear equation. The k basis functions represent characteristics of each state-action pair.

We have to update the weight vector ω to improve system policy.

Bellman equation and Eq 4 can be transformed to $\Phi\omega \approx R + \gamma P^{\pi}\Phi\omega$, where Φ represent the basis features for all state-action pairs. This equation is reformulated as:

$$\Phi^T(\Phi - \gamma P^{\pi}\Phi)\omega^{\pi} = \Phi^T R \tag{5}$$

Basing on equation 5, the weight ω of the linear functions in equation 4 is extracted:

$$\omega = (\Phi^T(\Phi - \gamma P^{\pi}\Phi))^{-1} \times \Phi^T R \tag{6}$$

For a router s and forwarding action a, s' is the corresponding neighbor router with $P(s'|s, a) = 1$. Our learning procedure is realized as follows: when a packet is forwarded from node s to s' by action a, which has been chosen by current Q-values $\phi(s, a)^T\omega$, a record $< s, a, s', \phi(s, a) >$ is inserted to the packet. The gathering process (Fig. 2) is realized until the packet arrives at the destination (end-users). Thus with

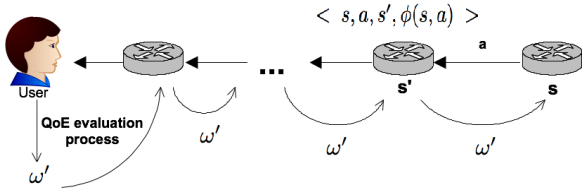


Fig. 2: Learning procedure

this way, one can trace the information of the whole routing path. At the end-users, a QoE evaluation process is realized to give a QoE score that is the value of *R* vector in equation 6. Furthermore, with the gathered information, the new value of ω is determined using equation 6. Then this new weight value ω' is sent back to the system along the routing path in order to improve policy procedure in each router on the routing path. With the new weights ω', policy improvement is realized in each router on the routing path by selecting the action *a* with the highest Q-value:

$$\pi(s|\omega') = argmax_a \phi(s,a)^T \omega' \quad (7)$$

The next subsection presents the improvement of this algorithm in using QoE measurement tool at all nodes of the routing system.

B. QQAR algorithm

In order to evaluate QoE in entire system, we have applied PSQA tool into all nodes (routers) including the last one representing end-user station (Fig. 1). In fact, measuring QoE anywhere in the system facilitates applying Q-Learning into our model with the reward at any node. That is the improvement factor of our first approach.
The proposed routing mechanism can be formulated as follows:

- *First step - Data packet flow*: the provider sends data packet to end-user. After receiving this data packet, each node in the routing path forwards the packet to the next node with a selection process presented in detail in subsection 2. It simultaneously evaluates QoE by using PSQA tool and saves this result.
- *Second step - At end-user side*: After data reach end-user, QoE evaluation is realized by using PSQA tool to give a QoE feedback as ACK message to the routing path that this data flow just passes through.
- *Third step - ACK message flow*: Each time a node receives a ACK message, it updates the Q-value of the link that this ACK message just passes through. The update process is introduced ci-below (subsection 1). It then attaches the QoE measurement result that it saved above into the ACK message and forwards it to the previous neighbor.

With regard to a selection process in each node after receiving a data packet, we have to consider the tradeoff between *exploration* and *exploitation*. This tradeoff is one of the challenges that arises in RL and not in other kinds of learning. To obtain a lot of reward, 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 has to try 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. In our approach, we choose softmax method as selection process that will be presented in the subsection 2.

1) Learning process: In our model, each router has a routing table that indicates the Q-values of links emerging from this router. For example in Fig. 3, node y has a routing table containing Q values: $Q_{yz_1}, Q_{yz_2}, Q_{yz_3}...Q_{yz_n}$ 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.

As mentioned above, after receiving a data packet, the last

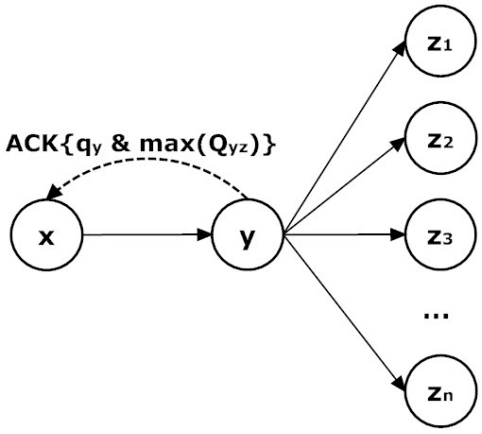


Fig. 3: Learning process

node representing end-user evaluates the QoE, it then sends back feedback as ACK message to the routing path. Each router x in this routing path receives this message containing information: the PSQA result of the previous router (q_y) and the maximum value of Q-values ($\max Q_{yz}$) in the routing table of node y (Fig. 3). Router x then updates the Q-value of the link connecting to y. Our update function based on the native Q-Learning (Eq 2) is defined in Equation 8:

$$\underbrace{Q_{xy}}_{\text{new value}} = \underbrace{Q_{xy}}_{\text{old value}} + \alpha \Big[\underbrace{\beta\,(q_y - q_x) + \gamma \max_i Q_{yz_i}}_{\text{new estimation}} - \underbrace{Q_{xy}}_{\text{old value}} \Big] \quad (8)$$

Where Q_{xy} and Q_{yz_i} are Q-values of links xy and yz_i. q_x and q_y are results obtained (MOS score) by using PSQA tool at node x and y. α is the learning rate, which models the rate updating Q-value. The two discount factors β and γ balance the value between future reward and immediate reward.

2) Selection process: As mentioned above, our selection process must balance between the *exploration* and *exploitation* phase. 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. Therefore, we have chosen softmax method using Boltzmann distribution [3]. So with this softmax action selection rules, after receiving a packet, node x chooses its neighbor y_k among its n neighbors y_i ($i = 1..n$) with probability presented in Equation 9:

$$p_{xy_k} = \frac{e^{\frac{Q_{xy_k}}{\tau}}}{\sum_{i=1}^{n} e^{\frac{Q_{xy_i}}{\tau}}} \quad (0 \le k \le 1) \quad (9)$$

Where Q_{xy_i} represents Q-value of link xy_i and τ represents a temperature parameter of Boltzmann distribution. High tem-

perature causes the link selection to be all equi-probable. Low temperatures generates a greater difference in selection probability for links that differ in their Q-values. In other words, more we reduce the temperature τ, the more we exploit the system. In that way, we reduce the temperature τ after each time of forwarding packet as shown in Equation 10:

$$\tau = \phi \times \tau \quad (0 < \phi < 1) \tag{10}$$

where ϕ is the weight parameter.
So in that way, we initially balance *exploration* and *exploitation*. After that system is quite converged, we then increasingly exploit the system.

IV. Experiment

In order to validate our proposed approach, this section presents firstly our testbed for collecting dataset to PSQA tool. We then describe our simulation results using OPNET simulator.

A. Testbed for PSQA

Training RNN for PSQA tool needs a real dataset of the impact of the network on the perceived video quality. To construct this dataset, we conducted an experiment consist of selecting a number of human and asked him to score the perceived quality of video using the MOS score. The testbed is composed by client-server architecture and a network emulator. The client is VLC video client [15] and the server is VLC video streaming server [15]. The traffic between client and server is forwarded by the network emulator NetEm [16]. NetEm provides the way to reproduce a real network in a lab environment.
The current version of NetEm can emulate variable delay, loss, duplication and re-ordering.
The experimental setup consists on forwarding video traffic between server and client. Then, we introduce artificial fixed delay, variable delay and loss on the link to disturb the video signal.
According to ITU-R [17], the length of the video should be at least 5sec. We choose the sintel video trailer [18]. This video is of 52 seconds duration, 1280 x 720 dimensions and 24 frames per second cadence and uses H.264 codec. This video was chosen because it alternates high and slow movements.
Experiments were conducted with fixed delay values of [25, 50, 75, and 100ms], variable delays of [0, 2, 4, 6, 8, 16, 32ms], loss rate values of [0, 2, 4, 6, 10, 15, 20, 25, 30%] and successive loss probability of [0, 30, 60, 90%]. These values were chosen to cover the maximum of QoE range.
To collect data, we chooses viewers with a strong cinematic experience.
Nowadays, as a major part of monitors are LCD, we used the same ones. The particular screen used is a 19" size screen "LG flatron L194wt–SF" which has 1440x900 resolution.
The obtained dataset of this testbed is used for learning process of PSQA tool. We then obtain the function f in Equation 1 and apply the latter to our system as QoE measurement tool (PSQA).

B. Routing system simulation results

We have used Opnet simulator version 14.0 to implement our approach in an autonomous system. Regarding network topology, we have implemented an irregular network with 3 separated areas including 38 routers, each area is connected to each other by one link, all links are similar (Fig. 4). The network system is dynamically changed with an average period of 200 seconds.

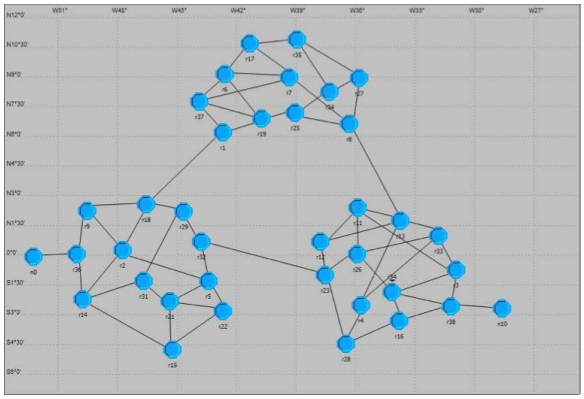


Fig. 4: Simulation network topology

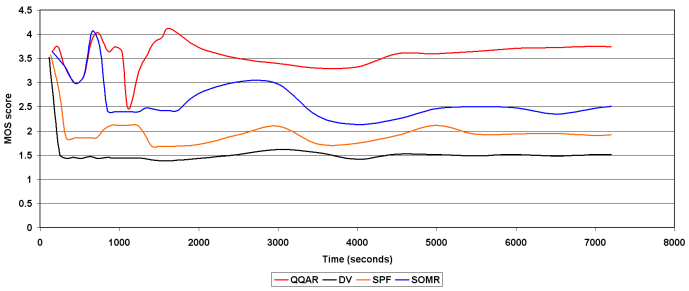
To validate our results, we compare our approach with three kinds of algorithm:

- *Those based on Distance-Vector (DV) algorithm*. 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 roads are updated every 30 seconds.
- *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 route to each network known in the LSDB.
- *Those based on Standard Optimal QoS Multi-Path Routing (SOMR) algorithm* where routing is based on finding K Best Optimal Paths and used a composite function to optimize delay and link cost criteria simultaneously.

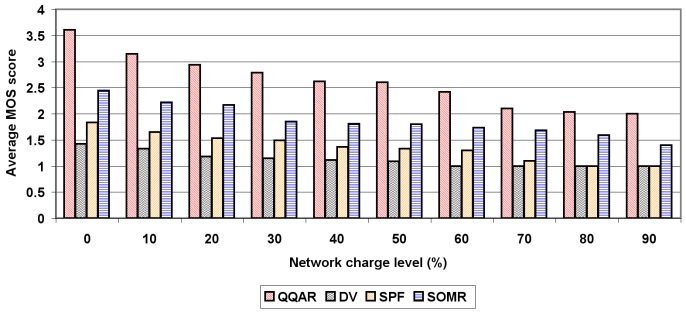
Fig. 5 illustrates the result of average MOS score of four algorithms (non-charged network in 5a and different charge levels in 5b). In the charged system case, we have generated a traffic that stress the network. The charge level represents the rate of number of charged link and number of total link:

$$\text{level} = \frac{n_s}{N} \tag{11}$$

where n_s is number of charged links and N is number of total links in the system.
In observing Fig. 5a, we can see that results of all four



(a) User perception in low load traffic network



(b) User perception in different load traffic levels of network

Fig. 5: User perception

algorithms fluctuate very much in the first 1500 seconds. That is explained by the execution of initialization process. In other words, these four algorithms do not have any information about the system in this first period, they try to explore it. In these first 1500 seconds, the MOS score of QQAR varies between 2.5 and 4, SOMR between 2.4 and 4.1, SPF between 1.7 and 3.6, DV between 1.4 and 3.5. After the first 1500 seconds, protocols gradually become stable. QQAR varies between 3.4 and 3.7. DV and SPF are quite stable with average results respectively 1.4 and 1.9. SOMR varies much more but the maximum value (obtained in period from 2100th to 2300th second) is still lower than QQAR.

Fig. 5b gives us the average of these four algorithms in different charge levels formulated in (11): from 10% to 90%. We can see that more the system is charged, the higher the average score decreases. However, at any charge level, the average MOS score of QQAR is better than three other algorithms. With a charge level lower than 10%, the MOS score of QQAR is higher than 3 (in MOS score range, 3 represents a fair quality). Regarding the three other protocols, the maximum value obtained by SOMR is just 2.4 with charge level 10%.

QQAR gives a better e2e QoE perception in both cases than three other algorithms in the same delay. So with our approach, despite network environment changes, we can maintain a better QoE without any other e2e delay or any other QoS metric. Thus, QQAR is able to adapt its decisions rapidly in response to changes in the network dynamics.

Our experiment works consist also the survey of overheads met by these protocols. The type of overhead we observe is control overhead that is determined by the proportion of control packet number to the number of all packets emitted. To monitor this overhead value, we have varied node number in adding routers to network system. So the observed node numbers are [38, 50, 60, 70, 80]. The obtained result is showed in Fig. 6. We can see that the control overhead of

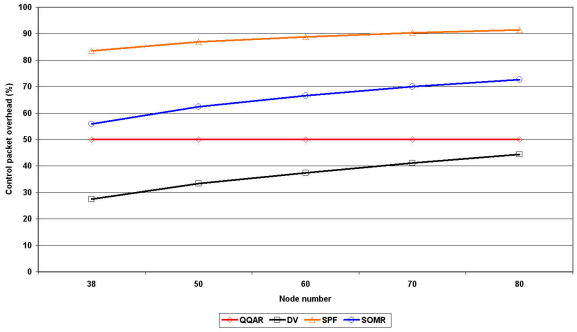


Fig. 6: Control overhead

our approach is constant (50%). That is explained by the equal of control packet number and data packet number in QQAR. Each generated data packet leads to an acknowledge packet generated by 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. 6. Whereas the SPF algorithm has the highest value because of the highest control packet rate (0.4 packet/second) with multiple type of packet such as Hello packet, Link State (LS) Acknowledgement 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, the higher the overhead is. So with a stable overhead, our approach is more scalable than these three others.

V. Conclusion

We present in this paper two approaches for routing systems driven by terminal QoE. We have integrated QoE measurement to routing paradigm for an adaptive and evolutionary system. Our second approach based on Q-Learning algorithm uses PSQA tool, a hybrid of subjective and objective method, for QoE evaluation. The simulations obtained demonstrates that our proposed approach yields significant QoE evaluation improvements and scalability over traditional approaches.

Finally, extensions to the framework for using these techniques across hybrid networks to achieve end-to-end QoE needs to be further investigated. Also, some future works includes applying our protocol to large scalable real testbed to verify its feasibility.

References

[1] Z. Wang and J. Crowcroft, "Quality of service routing for supporting multimedia applications," *IEEE Journal on selected areas in communications*, vol. 14, no. 7, pp. 1228–1234, 1996.

[2] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement learning: A survey," *Journal of AI Research*, vol. 4, pp. 237–285, 1996.

[3] R. S. Sutton and A. G. Barto, "Reinforcement learning: An introduction," *IEEE transactions on neural networks*, vol. 9, 1998.

[4] M. G. Lagoudakis and R. Parr, "Least-squares policy iteration," *Journal of Machine Learning Research*, vol. 4, p. 1149, 2003.
[5] G. Rubino, "Quantifying the quality of audio and video transmissions over the internet: The psqa approach," *Design and operations of communication networks: a review of wired and wireless modeling and management challenges. Imperial College Press, London*, 2005.
[6] "Recommendation p.801: Mean opinion score (mos) terminology," ITU-T Rec P.801, 2006.
[7] Watkins and Daylan, "Technical note: Q-learning," *Machine Learning*, vol. 8, no. 3, pp. 279–292, 1992.
[8] J. A. Boyan and M. L. Littman, "Packet routing in dynamically changing networks: A reinforcement learning approach," *Advances in Neural Information Processing Systems*, p. 671, 1994.
[9] L. Peshkin and V. Savova, "Reinforcement learning for adaptive routing," in *Neural Networks, 2002. IJCNN'02. Proceedings of the 2002 International Joint Conference on Neural Networks*, vol. 2. IEEE, 2002, pp. 1825–1830.
[10] A. Mellouk, S. Hoceini, and S. Zeadally, "Design and performance analysis of an inductive qos routing algorithm," *Computer Communications*, vol. 32, no. 1371-1376, 2009.
[11] S. Hoceini, A. Mellouk, and B. Smail, "Average-Bandwidth Delay Q-Routing Adaptive Algorithm," in *ICC'08. IEEE International Conference on Communications*. IEEE, 2008, pp. 1840–1844.
[12] H. A. Tran and A. Mellouk, "Qoe model driven for network services," *Wired/Wireless Internet Communications*, pp. 264–277, 2010.
[13] B. D. Vleeschauwer, F. D. Turck, B. Dhoedt, P. Demeester, M. Wijnants, and W. Lamotte, "End-to-end qoe optimization through overlay network deployment," *International Conference on Information Networking*, 2008.
[14] G. Majd, V. Cesar, and K. Adlen, "An adaptive mechanism for multipath video streaming over video distribution network (vdn)," *First International Conference on Advances in Multimedia*, 2009.
[15] Videolan. [Online]. Available: http://www.videolan.org/
[16] S. Hemminger, "Network emulation with netem," in *Linux Conf Au*, April 2005.
[17] "Recommendation 500-10: Methodology for the subjective assessment of the quality of television pictures," ITU-R Rec. BT.500, 2000.
[18] Sintel video trailer. [Online]. Available: http://www.sintel.org/