Cross-layer optimization of a multimedia streaming system via dynamic programming

Alice Combernoux, Cyrile Delestre, Nesrine Changuel, Bessem Sayadi, Michel Kieffer

CROSS-LAYER OPTIMIZATION OF A MULTIMEDIA STREAMING SYSTEM VIA DYNAMIC PROGRAMMING

Alice Combernoux[1], *Cyrile Delestre*[1], *Nesrine Changuel*[2], *Bessem Sayadi*[2], *Michel Kieffer*[1,3,4]

[1] L2S, CNRS - SUPELEC - Univ Paris-Sud, F-91192 Gif-sur-Yvette
[2] Alcatel-Lucent - Bell-Labs, France
[3] LTCI, CNRS-Télécom ParisTech, F-75013 Paris
[4] Institut Universitaire de France

ABSTRACT

This paper addresses the problem of efficient video streaming to mobile users. A cross-layer optimization of various parameters of the coding and transmission chain (coding parameters, buffer management, MAC-layer management) is performed to account for the time-varying nature of the characteristics of the transmitted contents and of the wireless channel. The problem is cast in the framework of Markov Decision Processes (MDP). This formalism provides efficient tools to compute a *foresighted* control policy maximizing some long-term discounted sum of rewards linked to the video quality received by the user. Experimental results illustrate the benefits in terms of average PSNR of this approach compared to a short-term (*myopic*) policy. The robustness of the proposed control policy to variations of the transmitted contents is also illustrated.

Index Terms— Discrete-time MDP, Data transmission, Cross-layer optimization, Video coding

1. INTRODUCTION

The increase of available bandwidth in wireless networks allows a larger diversity of services provided to users. Media streaming, video conferencing, video-on-demand, are examples of applications attracting an increasing number of users. Supporting these applications is a major challenge for current bandlimited wireless networks. In fact, existing wireless networks provide dynamically varying resources with only limited support for the quality of service required by delay-sensitive, bandwidth-intense, and loss-tolerant multimedia applications. One of the key challenges associated with multimedia transmission over wireless networks is the time-varying nature of the characteristics of transmitted contents and of the wireless channel [1, 2].

In this context, cross-layer optimization has been extensively investigated [3] to improve the quality of the contents decoded by the receivers. In [4], packets are scheduled for transmission over a channel characterized by a constant packet error rate to minimize the distortion at application layer while satisfying the delay constraint. The channel conditions observed at each time instant are considered, without paying attention to data heterogeneity. Dependencies between the multimedia packets are expressed as a direct acyclic graph in [5] and the packet scheduling is optimized to reach a rate-distortion compromise. Recently, [6] also considers the characteristics of multimedia data, as well as time-varying network conditions and adaptation capability of the user at the various layers of the protocol stack. Packet scheduling is optimized according to the application layer and transmission strategy adaptation at the adaptive medium access control (MAC) and physical layers. However, packet-size optimization at the MAC layer, which can result in good performance in terms of the multimedia quality, as shown in [7] is not considered in previously-cited works. In fact, in [7] a joint application and MAC technique is applied to minimize the distortion impact and fulfill delay constraints of the various packets. The cross-layer algorithm proposed in [7] uses Lagrangian formulation which allows maximizing the instantaneous utility, without considering the impact of the users current action on its long-term performance. In wireless multimedia applications, such myopic strategy design can result in unacceptable deterioration in long-term multimedia quality due to the heterogeneous characteristics of the media traffic. Therefore, cross-layer strategies need to be optimized in a foresighted way by considering the effect of current actions on the future performance.

Cross-layer optimization using Markov Decision Process (MDP) framework [8] was proposed in [9]. A joint control of packet scheduling at the transmitter and content-aware playout at the receiver is proposed to maximize the quality of media streaming over a wireless link. But, no adaptive MAC packet size selection is considered. In [10], packet scheduling and buffer management in both Application and MAC layers are jointly considered for single scalable video transmission. In [10], the buffer in the MAC layer is observed to get information about the state of the channel, but no action at the MAC layer is applied.

This paper focuses on determining the optimal cross-layer transmission policy for an individual wireless user (a transmitter-receiver pair) streaming video traffic over a wireless network in a time-varying environment. We develop a cross-layer optimization mechanism, where the application layer collaborates with the MAC layer to jointly determine the optimal quantization parameter per frame, MAC packet size, and scheduling decision, see Section 2. In Section 3, the optimization problem is cast in the MDP framework that explicitly considers the cooperation at the application and MAC layers, the heterogeneity of the video data, and post-encoding buffer control so as to overcome the variations of the channel and maximize the perceived video quality. This formalism allows deriving a foresighted control policy maximizing some long-term discounted sum of rewards. Due to

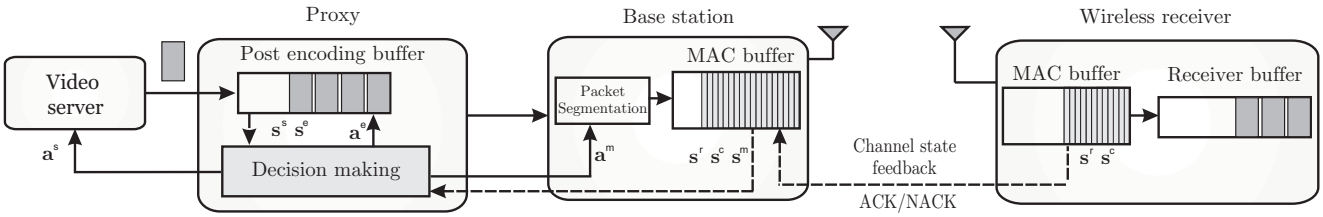

Fig. 1. Considered single-user streaming system

the dimensionality of the problem, values taken by the states have to be strongly quantized. Nevertheless, experimental results, detailed in Section 4 illustrate the benefits of this approach compared to a short-term (myopic) policy.

2. SYSTEM AND CONSTRAINTS

We consider a single user unicast streaming system to a mobile user. It consists of a streaming server, a post-encoder buffer, a MAC buffer, a wireless channel, a receiver buffer, and a video decoder, see Figure 1. The MAC buffer is located in the base station. The aim of this work is to maximize the average quality of the video decoded at the receiver side.

The streaming server encodes videos with the H.264/AVC encoder. The quantization parameter (QP) may be adjusted at a frame level to fit transmission conditions. The frame rate is assumed constant. A post-encoder buffer with finite capacity follows the encoder. It can send, keep, or drop packets. The MAC buffer carves up packets into Packets Data Units (PDUs) whose length is variable. A binary symmetric channel is considered with crossover probability described by a finite-state Markov process. An ARQ mechanisms is implemented at MAC layer: base station retransmits a MAC PDU through the channel until its reception is acknowledged. When the MAC buffer is full, packets entering the MAC buffer are dropped. The receiver consists of a receiver buffer and an H.264/AVC decoder. They are considered asynchronous from the other elements: as soon as enough MAC PDUs are received to form again a video packet, this packet is rebuilt, decoded, and stored in a frame buffer before being played out. Acknowledgments, the state of the channel, as well as the level of the receiver buffer are assumed to be fed back to the transmitter without delay.

Due to the behavior of the post-encoder and MAC buffers, one has to avoid empty or full buffer states. The receiver buffer has on the contrary to be as full as possible to increase the playout margin, which improves the continuity of display of the video stream in case of bad channel conditions.

3. MARKOV DECISION PROCESS

The system introduced in Section 2 is modeled with a discrete-time MDP [8]. All elements of the system are represented by a discrete-time MDP, except the decoder. An MDP is defined by a 5-tuple $(\mathcal{S}, \mathcal{A}, \mathbf{P}, \mathbf{R}, \pi)$, where $\mathcal{S}$ is the set of states, $\mathcal{A}$ the set of actions, $\mathbf{P} = (p(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t))$ the matrix of transition probabilities from State $\mathbf{s}_t$, applying Action $\mathbf{a}_t$ at time t to State $\mathbf{s}_{t+1}$ at time $t+1$. The reward matrix $\mathbf{R}$ is a function of $\mathbf{s}_{t+1}$ and $\mathbf{a}_t$. Finally, the policy $\pi(\mathbf{s}) \in \mathcal{A}$ describes the action to select when the state is $\mathbf{s}$. The time t corresponds to the time at which the encoding instant of the t-th frame of the streamed video content. All actions selected at time t are applied over the interval $[t, t+1)$.

To evaluate the optimal policy, $\pi^*(\mathbf{s})$, one may evaluate the optimal value function [8] $V^* : \mathcal{S} \mapsto \mathbb{R}$ which satisfies the following Bellman optimality equation

$$V^*(\mathbf{s}) = \max_{\mathbf{a}\in\mathcal{A}} \left\{ R(\mathbf{s}, \mathbf{a}) + \gamma \sum_{\mathbf{s}'\in\mathcal{S}} p(\mathbf{s}'|\mathbf{s}, \mathbf{a}) V^*(\mathbf{s}') \right\}, \quad (1)$$

where γ is some discount factor indicating the relative importance of the current reward and future rewards. The optimal policy for state value $\mathbf{s}$ is then obtained as

$$\pi^*(\mathbf{s}) = \arg\max_{\mathbf{a}\in\mathcal{A}} \left\{ R(\mathbf{s}, \mathbf{a}) + \gamma \sum_{\mathbf{s}'\in\mathcal{S}} p(\mathbf{s}'|\mathbf{s}, \mathbf{a}) V^*(\mathbf{s}') \right\}. \quad (2)$$

The value iteration algorithm is an efficient tool to provide $V^*(\mathbf{s})$ and $\pi^*(\mathbf{s})$, see [8].

Actions, states, transition, and reward matrices have now to be identified for the considered streaming system.

3.1. States

The state vector $\mathbf{s} = (s^{\mathrm{s}}, s^{\mathrm{e}}, s^{\mathrm{m}}, s^{\mathrm{c}}, s^{\mathrm{r}})^{\mathrm{T}}$ gathers the state of the source (and encoder) s^{s}, of the post-encoder buffer s^{e}, of the MAC buffer s^{m}, of the channel s^{c}, and of the receiver buffer s^{r}. All these state components should provide enough information to allow designing an efficient policy π, with a reasonable complexity. The dimension of the state-space has to satisfy a compromise between efficiency and complexity.

The state of the source s^{s} is chosen to represent the importance of the frame to be transcoded/layer filtered at time t. Here, $s^{\mathrm{s}} = I$ indicates that the t-th frame is a key frame, whereas $s^{\mathrm{s}} = P$ indicates that it is a (possibly bidirectionnaly) predicted frame. The choice between I and P is assumed to have been performed by the video encoder, based, *e.g.,* on scene change detection.

The state components s^{e}, s^{m}, and s^{r} indicate the level of the post-encoder, MAC, and receiver buffers. Each frame is stored in a single packet in the post-encoder and receiver buffer. MAC packets are stored in the MAC buffer. The level of the buffers may be measured in number of stored bits or packets. Keeping a precise description of these numbers would lead to an explosion of the size of the state-space. In order to monitor the load of the buffers, we choose to consider three possible state values for the MAC buffer, namely U for underflow, G for good and O for bit overflow. The state of the post-encoder and receiver buffer may take five values, U, G, O, UG, and GO, which are intermediate states between underflow and good, and good and overflow, respectively. These additional states, albeit they increase the system complexity, help to better account.

The channel is modeled as a binary symmetric channel, with constant rate R_c, and with crossover probability

$\varepsilon_{s_t^c} \in \{\varepsilon_1 \dots \varepsilon_n\}$ constant over a time interval $[t, t+1)$ and corresponding to the value of s^c at time t, described by an n-state first-order Markov process.

3.2. Actions

Many parameters may be adjusted to optimize the behavior of a video streaming chain over a wireless channel. Here, we focus on the adjustment of three tuning parameters of the system. The average frame quantization parameter (QP) is represented by a^s. One also selects at each time instant the number a^e of packets to take from the post-encoder buffer and to store in the MAC buffer (when $a^e \geq 0$) or to drop from the post-encoder buffer (when $a^e < 0$). Finally, we choose to adjust the size in bits a^m of MAC packets transmitted to the lower layers of the protocol stack, which has a direct impact on the size of packets transmitted on the channel. Small packets are more likely to reach the receiver in case of a noisy channel than big packets. The price to be paid is a higher overhead due to headers, which size is more or less independent of the size of the packet.

Many other choices for the actions are possible, for example, one could represent the modulation and coding scheme at physical layer.

The vector of actions $\mathbf{a} = (a^s, a^e, a^m)$ gathers all actions chosen at time t and applied over the time interval $[t, t+1)$.

3.3. Transitions and rewards

3.3.1. State transition matrix

The components of the state transition matrix may be factorized as follows

$$\begin{aligned} p(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t) \quad = \quad & p(s^s_{t+1}|s^s_t)p(s^e_{t+1}|s^s_t, s^e_t, a^s_t, a^e_t) \\ & p(s^m_{t+1}|s^m_t, s^c_t, a^e_t, a^m_t)p(s^c_{t+1}|s^c_t) \\ & p(s^r_{t+1}|s^c_t, s^r_t, a^m_t) \end{aligned} \quad (3)$$

The source and channel transition probabilities are independent of actions and other state components. The value s^e_{t+1} of the state of the post-encoder buffer is influenced by the state of the source, since it will be more filled by the encoding of an I frame than with a P frame, and it also depends on a^s and a^e. The evolution of the level of the MAC buffer is determined by the state of the channel, and by a^e and a^m. Finally, the state of the receiver buffer is determined by that of the channel and by a^m.

The evaluation of each transition probabilities involved in $p(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t)$ is done off line using training video sequences. $p(s^s_{t+1}|s^s_t)$ depends on the size of the GOP and of the frequency of scene changes. $p(s^e_{t+1}|s^s_t, s^e_t, a^s_t, a^e_t)$ depends on the rate-distortion characteristics of the considered video sequence, which has a direct impact on the size of packets generated at the output of the video encoder. $p(s^m_{t+1}|s^m_t, s^c_t, a^e_t, a^m_t)$ depends on the size of encoded frames, and thus indirectly on the rate-distortion characteristics of the source. $p(s^c_{t+1}|s^c_t)$ depends on the channel characteristics. Finally, $p(s^r_{t+1}|s^c_t, s^r_t, a^m_t)$ depends again on the size of encoded frames.

The entries of the state transition matrix may also be updated on line at each time instant and the optimal policy may be updated accordingly. When some transition probabilities are really difficult to obtain, one may resort to learning techniques, such as reinforcement learning to estimate the optimal policy, see for example [8, 11].

3.3.2. Reward matrix

Ideally, the reward matrix should directly account for the PSNR or for any other quality measure of the frames decoded at the receiver side. Unfortunately, the impact of actions on the received PSNR is not immediate. Thus, the following alternative reward is considered

$$\mathbf{R}(\mathbf{s}, \mathbf{a}) = \mathrm{PSNR}(s^s, a^s) + f_e(s^e, a^e) + f_m(s^c, s^m, a^m) + f_r(s^r). \quad (4)$$

The first part $\mathrm{PSNR}(s^s, a^s)$, corresponds to the PSNR in dB at the output of the encoder. It is function of the frame type and of the selected quantization parameter. The functions $f_e(s^e, a^e)$ and $f_m(s^c, s^m, a^m)$ provide very negative reward when the state of the buffer is in the U, O states. A small negative reward is provided when the buffers are in the intermediate states UG, and GO. Finally, $f_r(s^r)$ is the part of the reward associated with the receiver buffer. Negative rewards are provided when the buffer is in the U, UG, and O states.

3.4. Optimal policies

Using the value iteration algorithm [11], one is able to obtain the optimal value function $V^*(\mathbf{s})$ and the optimal policy $\pi^*(\mathbf{s})$ for various values of the discount factor γ. Here, a myopic policy $\pi^*_m(\mathbf{s})$ is considered for the case $\gamma = 0$. This policy corresponds to maximizing the immediate reward, without paying attention on the consequence the current decision may have on future rewards. A foresighted policy $\pi^*_f(\mathbf{s})$ is obtained for $\gamma \in]0, 1[$. In this case the impact of the current decision on future rewards is taken into account.

4. SIMULATION RESULTS

We consider an H.264/AVC video coder encoding QCIF frames at 15 frames/s. A constant GOP size of 12 frames is considered, which corresponds to a key frame every 0.8 s. The action a^s indicates the quantization parameter used to encode the current frame. It is chosen in the set $\mathcal{A}^s = \{21, 25, 30, 35\}$.

The post-encoder buffer is 600 kbits large. The intervals corresponding to the various values of the state s^e are $[0, 85]$ for U, $[85, 228]$ for UG, $[228, 371]$ for G, $[371, 515]$ for GO, and $[515, 600]$ for O. Four possible actions are considered for packets stored in this buffer $\mathcal{A}^e = \{-1, 0, 1, 2\}$, where 0, 1, or 2 correspond to the transmission to the MAC buffer of 0 (the packet is kept in the buffer), 1, or 2 packets, whereas -1 consists of dropping 1 packet.

The MAC buffer is also 600 kbits large. The intervals corresponding to the three possible values of the state are $[0, 150]$ for U, $[150, 450]$ for G, and $[450, 600]$ for O. The possible values of a^m are in $\mathcal{A}^m = \{256, 2048\}$, corresponding to the size of the transmitted payload.

The rate of the channel is $R_c = 180$ kbits/s. The channel state may take two possible values, namely G, associated to a crossover probability $\varepsilon_G = 10^{-5}$ and B, associated to $\varepsilon_B = 10^{-3}$. The channel state transition probabilities are $p_{GG} = \Pr(s^c_{t+1} = G | s^c_t = G) = 0.95$ and $p_{BB} = \Pr(s^c_{t+1} = B | s^c_t = B) = 0.7$.

Finally, the reception buffer is also 600 kbits wide. Its states and corresponding intervals are the same as those of the post-encoder buffer.

All parameters of the transition matrix have been tuned for the foreman.qcif sequence, with the previously introduced numerical values. The optimal myopic policy $\pi^*_m(\mathbf{s})$ is then

evaluated, setting $\gamma = 0$ in (2). The optimal foresighted policy $\pi_{\mathrm{f}}^{*}(\mathbf{s})$ is obtained with $\gamma = 0.9$.

All buffers are initially empty. The results are evaluated in permanent regime: 300 frames are used to drive the system in its permanent regime. The performance are evaluated on the 300 next frames (video sequences are played in loop).

The evolution of the PSNR for the foreman.qcif sequence

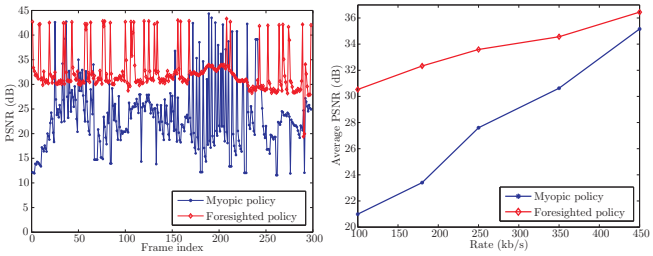

Fig. 2. Evolution of the PSNR as a function of time (left) and of the channel rate (right) for the forman.qcif sequence when $\pi_{\mathrm{m}}^{*}(\mathbf{s})$ and $\pi_{\mathrm{f}}^{*}(\mathbf{s})$ are applied

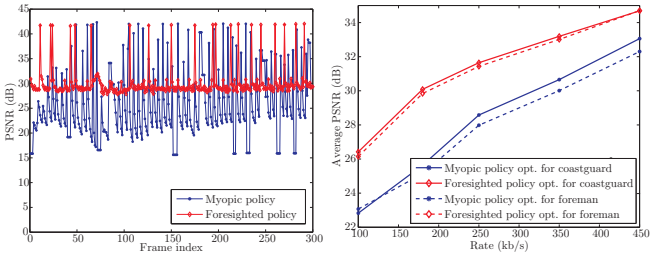

Fig. 3. Evolution of the PSNR as a function of time (left) and channel rate (right) for the coastguard.qcif sequence when $\pi_{\mathrm{m}}^{*}(\mathbf{s})$ and $\pi_{\mathrm{f}}^{*}(\mathbf{s})$ optimized for the foreman.qcif sequence (dashed) and when $\pi_{\mathrm{m}}^{*}(\mathbf{s})$ and $\pi_{\mathrm{f}}^{*}(\mathbf{s})$ optimized for the coastguard.qcif sequence (plain)

as a function of time is represented in Figure 2 (left). In average, the foresighted policy provides an increase of 9 dB in PSNR. The PSNR variations are much larger for the myopic policy, which tries to transmit the best quality at any time instant, with as consequence, the impossibility to transmit good quality frames after having transmitted large frames.

To evaluate the robustness of the proposed approach to variations of characteristics of the system, The optimal policy evaluated for a channel rate $R_c = 180$ kbits/s is now applied for different channel rates, see Figure 2 (right). The discrepancy between the foresighted and myopic policies decreases when the channel rate increases. This is due to the fact that the channel becomes less constrained.

The robustness to variations of the content are now characterized. The optimal policies obtained for the foreman.qcif video sequence are now applied to the coastguard.qcif video sequence, see Figure 3. At $R_c = 180$ kbits/s, a gain of more than 4 dB in PSNR is observed with the foresighted policy compared to the myopic policy. The behavior in presence of rate variations is similar to that observed with the forman.qcif sequence. One sees also that the mismatch in the optimization leads to a small loss for the myopic policy. The loss becomes negligible for the foresighted policy. This control technique is thus very robust to variations of the transmitted content.

5. CONCLUSION

This paper shows that the MDP framework is suitable for performing a cross-layer optimization of a video streaming chain. More than 3 dB improvements are observed with a foresighted policy, compared to a myopic policy, maximizing the immediate quality of the received video contents. The approach is also quite robust to variations of the system parameters, such as variations of characteristics of the transmission channel, or changes in the content of the video sequence.

The off line evaluation of the optimal policy is still quite challenging, since the complexity of this task increases significantly with the number of possible state values. A way to address this issue is to consider layered MDP techniques, as proposed in [12] for the design of efficient video coders. The layered architecture of protocol stacks fits well this formalisms.

6. ACKNOWLEDGMENTS

The authors would like to thank Prof. M. van der Schaar for valuable discussions on dynamic programming and its applications to multimedia content delivery.

7. REFERENCES

[1] V. Pahalawatta and A. Katsaggelos, "Review of content-aware resource allocation schemes for video streaming over wireless networks: Research articles," *Wirel. Commun. Mob. Comput.*, vol. 7, pp. 131 – 142, 2007.

[2] Y. Zhang, F. Fu, and M. van der Schaar, "On-line learning and optimization for wireless video transmission," *IEEE Transactions on Signal Processing*, vol. 58, no. 6, pp. 3108 –3124, 2010.

[3] M. van Der Schaar and Sai Shankar N, "Cross-layer wireless multimedia transmission: challenges, principles, and new paradigms," *IEEE Wireless Communications*, vol. 12, no. 4, pp. 50 – 58, 2005.

[4] A. Faridi and A. Ephremides, "Distortion control for delay-sensitive sources," *IEEE Transactions onInformation Theory*, vol. 54, no. 8, pp. 3399 –3411, 2008.

[5] P.A. Chou and Zhourong Miao, "Rate-distortion optimized streaming of packetized media," *IEEE Transactions on Multimedia*, vol. 8, no. 2, pp. 390 – 404, 2006.

[6] F. Fu and M. van der Schaar, "Structural solutions for cross-layer optimization of wireless multimedia transmission," *CoRR*, vol. abs/0905.4087, 2009.

[7] M. van der Schaar and D. S. Turaga, "Cross-layer packetization and retransmission strategies for delay-sensitive wireless multimedia transmission," *IEEE Transactions on Multimedia*, vol. 9, no. 1, pp. 185 –197, 2007.

[8] R.S Sutton and A.G Barto, *Reinforcement Learning: An Introduction*, MIT Press, 1998.

[9] Y. Li, A. Markopoulou, J. Apostolopoulos, and N. Bambos, "Content-aware playout and packet scheduling for video streaming over wireless links," *IEEE Transactions on Multimedia*, vol. 10, no. 5, pp. 885 –895, 2008.

[10] N. Changuel, N. Mastronarde, M. van der Schaar, B. Sayadi, and M. Kieffer, "Adaptive scalable layer filtering process for video scheduling over wireless networks based on MAC buffer management," in *IEEE ICASSP*, 2011, pp. 2352 – 2355.

[11] W. P. Powell, *Approximate dynamic programming: solving the curse of dimensionnality*, Wiley interscience, 2007.

[12] Nicholas Mastronarde and Mihaela van der Schaar, "Designing autonomous layered video coders," *Signal Processing: Image Communication*, vol. 24, no. 6, pp. 417 – 436, 2009.