A Novel Scheduling Algorithm for Video Traffic in High-Rate WPANs

Shahab Moradi, A. Hamed Mohsenian Rad, and Vincent W.S. Wong
Department of Electrical and Computer Engineering
The University of British Columbia, Vancouver, Canada
e-mail: {shahab, hamed, vincentw}@ece.ubc.ca

***Abstract*— The emerging high-rate wireless personal area network (WPAN) technology is capable of supporting high-speed and high-quality real-time multimedia applications. In particular, MPEG-4 video streams are deemed to be a widespread traffic type. However, in the current IEEE 802.15.3 standard for media access control (MAC) of high-rate WPANs, the implementation details of some key issues such as scheduling and quality of service (QoS) provisioning have not been addressed. In this paper, we first propose a mathematical model for the optimal scheduling scheme for MPEG-4 flows in high-rate WPANs. We also propose an RL scheduler based on the reinforcement learning (RL) technique. Simulation results show that our proposed RL scheduler achieves nearly optimal performance and performs better than F-SRPT [1], EDD+SRPT [2], and PAP [3] scheduling algorithms in terms of a lower decoding failure rate.**

I. Introduction

In the past few years, ultrawide-band (UWB) technology has received increasing attention in the wireless world. It provides short-range connectivity, low transmit power levels, and high-data rates, which make UWB to be the physical layer of choice for high-rate wireless personal area networks (WPANs). UWB-enabled WPANs can offer many new applications, such as home entertainment, real-time multimedia streaming, and wireless USB. In order to fully exploit UWB technology in high-rate WPANs, upper layers, including the media access control (MAC) layer, must be properly designed for high-rate applications. Video transmission is one such application for high-rate WPANs. Real-time video flows are delay-sensitive and require quality of service (QoS) guarantee. However, in the IEEE 802.15.3 standard for MAC [4], which is designed for WPANs, details of scheduling and QoS support are left to the developers. Consequently, in this paper, we aim to design a scheduling algorithm for MAC layer to provide the required QoS for video traffic.

In order to reduce bandwidth consumption, video traffic is usually compressed with variable bit rate (VBR) encoders, among which MPEG-4 is the most widely used. Similar to other real-time traffic, MPEG-4 stream is delay sensitive, and its frames are dropped at the receiver if their delay exceeds the maximum tolerable delay. This is the base of *job failure rate* (JFR) metric for evaluating the performance of MAC schedulers. It is defined as the fraction of frames that fail to meet their transmission deadlines (and thus become useless). However, MPEG-4 stream has a few unique characteristics that make QoS support more challenging than other real-time traffic. It has high burstiness, large peak-to-average ratio of the frame sizes, and hierarchical structure with dependency among its frames [3]. Therefore, a more accurate metric which takes frame dependencies into account is required. We use the *decoding failure rate* (DFR) criterion, which is defined as the ratio of the total number of undecodable frames to the total number of frames [5]. DFR can be viewed as an objective measure of user-perceived degradation of quality. Thus, it should be minimized for better QoS performance.

Recently, various MAC scheduling algorithms have been proposed for high-rate WPANs [1]–[3], [6]–[11]. For impulse-based UWB, scheduling problems can be formulated as rate and power allocation problems. These problems can be modeled as a joint optimization problem, so as to minimize the total power consumption [8] or maximize the total system throughput [10]. The concept of exclusion region is also used for such schedulers [11]. On the other hand, with no assumption on the type of physical layer, Mangharam *et al.* proposed the fair shortest remaining processing time (F-SRPT) scheduler [1]. SRPT schedules different jobs in the system in the order of their remaining processing time, from the shortest to the longest. F-SRPT is a variation of SRPT that maintains fairness among flows with different data rates. In [2], the earliest due date (EDD) method is used along with SRPT. Kim and Cho proposed a scheduling algorithm designed for MPEG-4 flows [3]. Each MPEG-4 frame type is scheduled with a pre-assigned priority (PAP) in I, P, and B order. In [12], we proposed a frame-decodability aware (FDA) technique to improve the performance of scheduling MPEG-4 flows.

In this paper, we propose a novel scheduling algorithm which minimizes the average DFR. The contributions of our work are as follows:

- We formulate the scheduling of MPEG-4 flows in high-rate WPANs as a Markov decision process (MDP) problem. The model takes into account the number and pattern of MPEG-4 flows, and the hierarchical structure.
- Using reinforcement learning (RL), we find the optimal (or near-optimal) scheduling policy defined by the MDP model. The optimal policy minimizes the average DFR.
- Simulation results show that our proposed RL scheduler reduces the average DFR by 42%, 49%, and 53% when compared to EDD+SRPT [2], PAP [3], and F-SRPT [1] schedulers, respectively.

The rest of the paper is organized as follows. In Section II, we summarize IEEE 802.15.3 standard and the hierarchical structure of MPEG-4 streams. In Section III, we present the problem formulation and our proposed scheduling algorithm for video traffic. Performance comparisons are given in Section IV, and conclusions are drawn in Section V.

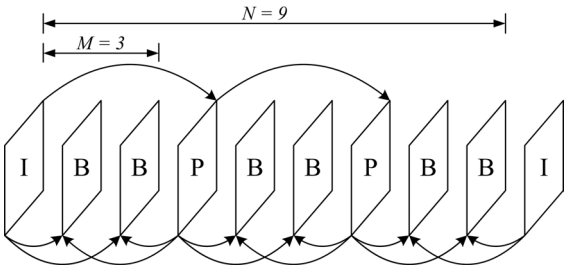

Fig. 1. GOP structure (N = 9, M = 3). The arrows indicate the direction of decoding dependencies.

II. Background

A. IEEE 802.15.3 MAC Standard

An 802.15.3 *piconet* is a wireless ad-hoc data communications system that allows a number of independent data devices (DEVs) to communicate with each other. Within a piconet, one DEV, called the piconet coordinator (PNC), is responsible to provide basic timing, perform scheduling, manage QoS, and control media access. The timing in a piconet is based on *superframes*. Each superframe includes a contention free period, the use of which is determined by the scheduling decisions that PNC makes.

B. MPEG-4 Streams

An MPEG-4 stream consists of a sequence of compact frames which can have three types: intra-coded (I), predictive (P), or bidirectional (B) frames. The type of frames in the sequence is according to a predefined pattern called group of pictures (GOP). This pattern is characterized by two parameters (N, M), where N is the I-to-I frame distance, and M is the I-to-P frame distance [13]. This pattern is generally fixed for a given video sequence, and N is a multiple of M. Different frame types are encoded using different compression schemes. Therefore, I frames *tend* to be larger (less compressed) than P and B frames, and P frames *tend* to be larger than B frames. Furthermore, MPEG-4 streams have a hierarchical structure, meaning that there are decoding dependencies among the frames (see Fig. 1). For a frame to be decodable at the receiver, all other frames that it depends on must be available at the receiver. Otherwise, it is undecodable [5]. As shown in Fig. 1, all the frames in a GOP depend on the I frame in that GOP; therefore, if that I frame does not meet its deadline, the whole GOP is undecodable.

III. Optimal Scheduling Algorithm for Video Traffic

In this section, we first formulate the problem of finding the scheduling policy with minimum average DFR in the form of a Markov decision process (MDP) with average reward criterion [14]. We next relate the gain of the scheduling policy with the average DFR that it yields. As a result of the linear relationship between gain and average DFR, we use the reinforcement learning (RL) techniques [15], [16] to determine the optimal scheduling policy which minimizes the average DFR.

A. System Model

Consider a WPAN and let F denote the number of MPEG-4 flows. We denote $\mathcal{F} = \{1, 2, \ldots, F\}$ as the set of all flows. We assume that the deadline of all MPEG-4 frames is fixed, and is equal to the frame inter-arrival time γ. The superframe size η is also assumed to be fixed and less than γ. We denote the GOP pattern of the flow $i \in \mathcal{F}$ by (N_i, M_i), where N_i and M_i are defined in Section II-B. The GOP pattern of all the flows is assumed to remain fixed during the flows' lifetime.

To model the scheduling problem as an MDP, we need to determine the corresponding decision epochs, system states, actions, reward function, and state transition probabilities [14]. In our model, the scheduler makes a decision at the beginning of each superframe, and determines which flows should be scheduled in that superframe. For simplicity, we show the decision epochs in terms of superframe size. For example, the actual time of decision epoch n, which is the beginning of n^{th} superframe, is $(n-1)\eta$. Therefore, the set of decision epochs can be denoted as $\{1, 2, \ldots, T\}$, for $T < \infty$.

For any flow $i \in \mathcal{F}$, let l_i denote the length of the frames in the queue (in bytes), d_i denote the number of full superframes left until the arrival of the next frame[1], and g_i denote the offset of the queued frame with respect to the beginning of the frame's GOP. We also set δ_i to 0 if the frame of flow i is undecodable, and set to 1 otherwise. We define the set of all possible system states as $\boldsymbol{S}$. A state $(\underline{l}, \underline{d}, \underline{g}, \underline{\delta})$ belongs to $\boldsymbol{S}$ if and only if for all $i \in \mathcal{F}$ we have:

$$\begin{aligned} 0 \le l_i \le L_i^{max}, \qquad & d_i \in \{0, \ldots, D^{max}\}, \\ g_i \in \{0, \ldots, N_i - 1\}, \qquad & \delta_i \in \{0, 1\}, \end{aligned} \tag{1}$$

where L_i^{max} is the maximum frame size of the i^{th} flow, and $D^{max} = \lfloor \frac{\gamma}{\eta} \rfloor$ is the maximum frame deadline in units of superframe size η. In vector notation, we have $\underline{l} = [l_1 \ l_2 \ \cdots \ l_F]$, $\underline{d} = [d_1 \ d_2 \ \cdots \ d_F]$, $\underline{g} = [g_1 \ g_2 \ \cdots \ g_F]$, and $\underline{\delta} = [\delta_1 \ \delta_2 \ \cdots \ \delta_F]$. In order to find the value of $\underline{\delta}$ based on $\underline{l}$, $\underline{d}$ and $\underline{g}$, the scheduler incorporates the frame decodability aware (FDA) technique which we proposed in [12].

Let $\boldsymbol{A}$ be the set of all possible actions and $\underline{a} = [a_1 \ a_2 \ \cdots \ a_F]$. An action $\boldsymbol{a} = (\underline{a}, i^{partial})$ is a possible action if and only if $i^{partial} \in \mathcal{F} \cap \{0\}$ and for all $i \in \mathcal{F}$, we have:

$$a_i \in \{0, 1\}, \qquad a_i + a_{i^{partial}} \le 1, \tag{2}$$

where a_i is equal to 1 if the scheduler allocates enough channel time to flow i so that it can *fully* transmit its frame. Otherwise, it is equal to 0. The parameter $i^{partial}$ is the flow that can only transmit parts of its frame during the channel time that the scheduler allocates to it. If no such flow exists, $i^{partial} = 0$. Here we assume that in each superframe, the scheduler allows at most one partial frame transmission, and the rest are full frame transmissions. The inequality $a_i + a_{i^{partial}} \le 1$ implies that a frame cannot be both fully and partially transmitted simultaneously.

[1] Since the maximum tolerable delay for MPEG-4 frames is the frame inter-arrival time, d_i can alternatively be interpreted as the deadline of the frame in the queue of flow i, in terms of superframe size.

At the beginning of each superframe, the scheduler chooses an action depending on the current state. The chosen action must satisfy a few constraints. First, the channel time required to transmit scheduled frames should not exceed the superframe length η:

$$\sum_{i=1}^{F} \text{tx}(l_i \times a_i) \leq \eta, \tag{3}$$

where the function tx$(\cdot)$ gives the transmission time of its argument. The simplest form of this function is tx$(x) = x/$channel_data_rate. However, depending on the acknowledgement policy, inter-frame spacing times, and maximum MAC fragment size, this function may have a different form.

Second, all the scheduled flows must be eligible:

$$a_i \leq e_i, \qquad \forall\, i \in \mathcal{F}, \tag{4}$$

$$e_{i^{partial}} = 1, \qquad \text{if } i^{partial} \neq 0, \tag{5}$$

where e_i denotes the *eligibility* of the flow i for being scheduled. Flow $i \in \mathcal{F}$ is eligible ($e_i = 1$) if it has a frame that is not expired ($d_i > 0$) and is decodable ($\delta_i = 1$); otherwise, it is ineligible for being scheduled ($e_i = 0$). Equation (4) implies that $a_i = 0$, if $e_i = 0$. In other words, the ineligible flows are never scheduled. After the scheduler makes its decision about which frames should get fully transmitted in a superframe, there may still remain some channel time in that superframe that is not enough for full transmission of any unscheduled frame. The amount of data that can be sent in this remaining time is $l^{partial} = \text{tx}^{-1}(\eta - \sum_{i=1}^{F} \text{tx}(l_i \times a_i))$, where tx$^{-1}(\cdot)$ is the inverse of tx function, and gives the amount of data that can be sent within a channel time equal to its argument. The fact that $l^{partial}$ is not enough for full transmission of any eligible frame that is not scheduled, can be formally expressed as the following constraint:

$$l_i > l^{partial}, \qquad \forall\, i \in \mathcal{F},\ e_i = 1,\ a_i = 0. \tag{6}$$

The idle channel time is allocated to the flow $i^{partial}$. If no such flow exists, $i^{partial}$ is set to 0. Consequently, the set of possible actions in state $\boldsymbol{s} \in \boldsymbol{S}$, denoted by $\boldsymbol{A_s}$, is the largest subset of $\boldsymbol{A}$, whose members satisfy all the constraints (3)–(6). These constraints guarantee that the scheduler accommodates as many eligible frames as possible. Note that $\boldsymbol{A_s}$ is stationary and only depends on the system states.

Because of the hierarchical structure and inter-dependency of MPEG-4 frames, some frames may be undecodable. We should properly choose the reward function to take this into account. We give a reward of one unit when a frame is scheduled. On the other hand, if the deadline of a frame expires, the scheduler receives a penalty (negative reward) of W units, where W is the number of frames that depends on the expired frame. Table I shows the number of dependencies between GOP frames of an (N, M) MPEG-4 flow.

Let $c_i(\boldsymbol{s})$ be the state-dependent penalty (or cost) function for flow i, and $c_i(\boldsymbol{s}) =$

$$\begin{cases} \big(N_i + (M_i - 1)\big) \cdot e_i \cdot U_{\{d_i=1\}}, & y_i(g_i) = I, \\ e_i \cdot U_{\{d_i=1\}}, & y_i(g_i) = B, \\ \big(N_i - 1 - (\frac{g_i}{M_i} - 1)M_i\big) \cdot e_i \cdot U_{\{d_i=1\}}, & y_i(g_i) = P, \end{cases} \tag{7}$$

TABLE I

NUMBER OF FRAME DEPENDENCIES FOR EACH MPEG-4 FRAME

Frame type	Number of frames
I	$N + (M-1)$
$P_k, k = 1, \ldots, \frac{N}{M} - 1$	$N - 1 - (k-1)M$
B	1

where the function $y_i(g) : \{0, \ldots, N_i - 1\} \rightarrow \{I, B, P\}$ maps g to the frame types as follows:

$$y_i(g) = \begin{cases} I, & g = 0; \\ B, & g \bmod M_i \geq 1; \\ P, & g \bmod M_i = 0 \text{ and } g \neq 0. \end{cases} \tag{8}$$

Furthermore, $U_{\{\cdot\}}$ is the indicator function and is equal to 1 if its argument is true, and is 0 otherwise. The product $e_i \cdot U_{\{d_i=1\}}$ indicates if the flow i has an urgent eligible frame. Hence, the scheduler should receive $c_i(\boldsymbol{s})$ units of penalty if it does not schedule flow i in the current superframe. As a result, we can express the state- and action-dependent reward that the scheduler receives at decision epoch n by:

$$r(\boldsymbol{s}(n), \boldsymbol{a}(n)) = \sum_{i=1}^{F} \Big[a_i(n) - c_i(\boldsymbol{s}(n))(1 - a_i(n)) \Big]. \tag{9}$$

In order to show the merits of the reward function in equation (9), we study the policy gain that it yields. The gain of policy π under the average reward criterion is the average accumulated reward [14]. In our model, the policy gain ρ^π is given by:

$$\begin{aligned} \rho^\pi &= \frac{1}{T} \sum_{n=1}^{T} r(\boldsymbol{s}(n), \boldsymbol{a}(n)) \\ &= \frac{1}{T} \sum_{i=1}^{F} \left(\sum_{n=1}^{T} a_i(n) - \sum_{n=1}^{T} c_i(\boldsymbol{s}(n))(1 - a_i(n)) \right). \end{aligned} \tag{10}$$

The total number of frames for each flow is total_frames $= \frac{\text{total\_time}}{\text{inter\_arrival\_time}} = \frac{\eta T}{\gamma}$. Furthermore, the terms $\sum_{n=1}^{T} a_i(n)$ and $\sum_{n=1}^{T} c_i(\boldsymbol{s}(n))(1 - a_i(n))$ in equation (10) are in fact the total number of scheduled frames and the total number of undecodable frames for each flow, respectively. In addition, these two terms add up to the total number of frames for flow i. Consequently, equation (10) can be rewritten as:

$$\begin{aligned} \rho^\pi &= \frac{\eta}{\gamma} \sum_{i=1}^{F} \Big(\frac{\text{total\_scheduled} - \text{total\_undecodable}}{\text{total\_frames}} \Big) \\ &= \frac{\eta}{\gamma} \sum_{i=1}^{F} \Big(1 - \frac{2 \times \text{total\_undecodable}}{\text{total\_frames}} \Big) \\ &= \frac{\eta}{\gamma} \sum_{i=1}^{F} (1 - 2 \times \text{DFR}_i^\pi) \\ &= \frac{\eta F}{\gamma} - \frac{2\eta}{\gamma} \sum_{i=1}^{F} \text{DFR}_i^\pi, \end{aligned} \tag{11}$$

where DFR_i^π denotes the decoding failure rate of flow i under the policy π. Let $\overline{\text{DFR}}^\pi \triangleq \frac{1}{F}\sum_{i=1}^{F} \text{DFR}_i^\pi$ denote the average DFR under the policy π. Using equation (11), we have

$$\overline{\text{DFR}}^\pi = \frac{1}{2} - \frac{\gamma}{2\eta F}\rho^\pi. \tag{12}$$

Equation (12) shows that in our formulation, the average DFR is a linear function of gain. Therefore, we conclude that:

$$\arg\max_\pi \rho^\pi = \arg\min_\pi \overline{\text{DFR}}^\pi. \tag{13}$$

Hence, an optimal (maximum) gain policy yields the optimal (minimum) average DFR, which is what we aim to find.

To define the state transition probabilities, assume that at superframe n, the system is in state $\boldsymbol{s}(n)$ and chooses the action $\boldsymbol{a}(n)$. For the rest of this subsection, we describe how to determine the system state at superframe $n+1$, $\boldsymbol{s}(n+1)$, which depends on $\boldsymbol{s}(n)$, $\boldsymbol{a}(n)$ and new frame arrivals. We determine the state transitions per flow. The whole system state is updated by performing the same procedure for all the flows.

When $d_i(n) \geq 1$, no new MPEG-4 frame will arrive for flow i. The time left till the next arrival is reduced by one superframe. If the flow is scheduled, its length will become zero. And if it is partially transmitted, its length will reduce as much as $l^{partial}$. Otherwise, the length remains unchanged. The frame offset within GOP does not change. The value of δ_i changes only when an I or P frame of flow i expires, or when a new GOP starts, which is indicated by arrival of a new I frame for flow i. None of these happens if $d_i(n) \geq 1$. Thus, if $d_i(n) \geq 1$, the state deterministically changes as follows:

$$d_i(n+1) = d_i(n) - 1, \tag{14}$$
$$l_i(n+1) = l_i(n)(1 - a_i(n)) - l^{partial} U_{\{i=i^{partial}\}}, \tag{15}$$
$$g_i(n+1) = g_i(n), \tag{16}$$
$$\delta_i(n+1) = \delta_i(n). \tag{17}$$

When $d_i(n) = 0$, it means that the frame in the queue of flow i has expired, if it has not already been sent, i.e. if $l_i(n) \neq 0$. It also means that a new frame will arrive for flow i within superframe n. Thus, when $d_i(n) = 0$, the state changes with probability $\text{prob}_i(l_i^{new}(n))$ as follows:

$$d_i(n+1) = d_i^{new}(n), \tag{18}$$
$$l_i(n+1) = l_i^{new}(n), \tag{19}$$
$$g_i(n+1) = (g_i(n)+1) \bmod N_i, \tag{20}$$
$$\delta_i(n+1) = \begin{cases} 1, & \text{if } y_i(g_i(n+1)) = I, \\ 0, & \text{if } y_i(g_i(n)) = I, P,\ l_i(n) \neq 0, \\ \delta_i(n), & \text{otherwise,} \end{cases} \tag{21}$$

where $\text{prob}_i(\cdot)$ is the frame size distribution probability of flow i. Parameters $d_i^{new}(n)$ and $l_i^{new}(n)$ denote the deadline and length of the newly arrived frame for flow i, in superframe n.

B. Algorithm Implementation: RL Scheduler

Since the frame size distribution may not be available a priori and the state space is huge, we use reinforcement learning (RL) to determine the optimal policy. An algorithm for average reward RL called SMART (Semi-Markov Average Reward Technique) [17] is used to find the optimal gain policy. This algorithm calculates the *value* of taking action a in state s, denoted by $R(s,a)$, which is a measure of the action's appropriateness. The higher the action value, the better the action is and the more it is favored by the scheduler. In order to accrue a lot of rewards, the scheduler should be greedy. In other words, at each superframe, it should choose the action with maximum value. However, the scheduler should occasionally deviate from the greedy manner (i.e., perform exploration) in order to search for potentially better actions.

Initialize the superframe number $n = 0$, action values $R(s,a) = 0$ for all $s \in S$ and $a \in A_s$, cumulative reward $CM = 0$, and the average reward $\rho = 0$. Superframe size is η. Suppose that the system starts in state s.

while $n <$ MAX_STEPS do

1) Calculate exploration probability p_n and learning rate μ_n using the DCM method.
2) With probability $1 - p_n$, choose the greedy action $a \in A_s$ that maximizes $R(s,a)$; otherwise, choose a random exploratory action from the set $\{A_s \setminus a\}$.
3) Execute the chosen action. Let the system state at the next superframe be s', and the immediate received reward be $r(s,a)$.
4) Calculate the target value $R_{tar}(s,a) =$ $(1-\mu_n)R(s,a) + \mu_n\{r(s,a) - \eta\rho + \max_{a'} R(s',a')\}$
5) **if** a greedy action was chosen in step 2, **then** Update $CM \leftarrow CM + r(s,a)$ and $\rho \leftarrow CM/n\eta$; **else**, go to Step (2).
6) Update the action value representation based on the approximation error $R_{tar}(s,a) - R(s,a)$.
7) Go to the next superframe, i.e., update $n \leftarrow n+1$ and $s \leftarrow s'$.

Fig. 2. Pseudo-code of the RL scheduler.

When the state space is large, as in our MDP model, it requires a more compact representation for the state space. This will lead to a reduction of memory requirement and an increase of the convergence rate for RL. For large state space, it may be possible to *aggregate* similar states with a slight degradation in accuracy. Furthermore, depending on the learning problem, there may exist some *features* in the state and action, that can fully capture the important aspects of system that influence the learning process. We take advantage of aggregation within $\boldsymbol{S}$ as well as features to form a compact representation of the state space [18]. Similarly, an action value representation is required for generalizing among the value of similar actions. We use the sparse distributed memory for this matter [19], [20].

Using SMART, we determine the scheduling policy that achieves the minimum average DFR. We call this scheduler as the *RL scheduler*. The pseudo-code of the RL scheduler is given in Fig. 2. The exploration probability and the learning rate decay are based on Darken-Chang-Moody (DCM) search-then-converge procedure [21]. Using DCM method, the exploration probability and learning rate at superframe n are given by $p_0/[1 + (\frac{n^2}{p_r+n})]$ and $\mu_0/[1 + (\frac{n^2}{\mu_r+n})]$, respectively. The parameters p_0, p_r, μ_0, and μ_r are constants. The structure of our proposed RL scheduler is illustrated in Fig. 3.

The implementation of proposed scheduler requires a signaling scheme to pass the required info from DEVs to the

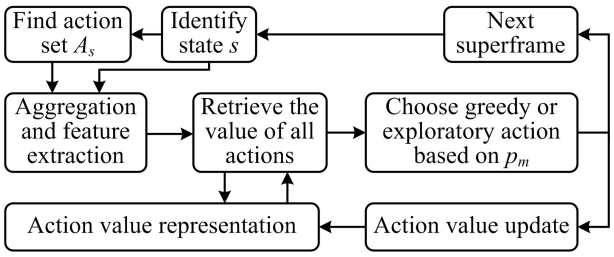

Fig. 3. Structure of the RL scheduler.

PNC. For example, it can use the signaling method proposed in [22] which is compatible with the 802.15.3 standard.

IV. Performance Evaluation

In this section, we evaluate the performance of our proposed scheduling algorithm. In the simulation model, each iteration (i.e., simulation run) lasts 500 s. The superframe size is $\eta = 8$ ms; thus, each iteration consists of $500/0.008 = 62500$ superframes (i.e., decision epochs). The parameters p_0 and μ_0 of the DCM scheme algorithm depend on how fast the scheduler can learn the optimal policy. We set $p_0 = \mu_0 = 0.1$ and $p_r = \mu_r = 10^{10}$, as it gives the RL scheduler enough time to explore and find the optimal policy without too many iterations. We end the simulation when both the learning rate and exploration probability fall below 0.005.

The rest of the simulation parameters are as follows. The channel data rate is 100 Mbps, the number of MPEG-4 flows varies from 2 to 10. The GOP pattern of the flows is (12, 3), and their mean data rate is 8 Mbps. Moreover, the frame inter-arrival time is 1/30 s, maximum tolerable delay is 1/30 s, and maximum MAC fragment size is 2048 bytes. We use the average DFR as the performance metric. For F-SRPT, EDD+SRPT, and PAP scheduling algorithms we use the performance that is already improved by the FDA technique described in our previous work [12].

Fig. 4 compares the average DFR achieved by RL, EDD+SRPT, PAP, and F-SRPT algorithms when the number of MPEG-4 flows varies from 2 to 10. We can see that RL scheduler performs better than the others regardless of the number of MPEG-4 flows. The relative reduction of average DFR is up to 42%, 49%, and 53% for EDD+SRPT, PAP, and F-SRPT scheduler, respectively. This improvement can also be translated to system capacity enhancement. Suppose that the acceptable user perceived quality is equivalent to the average DFR being less than 5%. Thus, the capacity of the system can be defined as the number of MPEG-4 flows that can be admitted to the system, while the average DFR is less than the maximum allowable value of 5%. Using this definition, the system capacity is 7 flows for the conventional schedulers, as opposed to 8 flows for the RL scheduler. Consequently, in this example, the RL scheduler increases the system capacity by 14.3%.

The start time of different flows in the system affects the burstiness of traffic load, and thus influences the overall performance. In order to show this fact, we assume that the

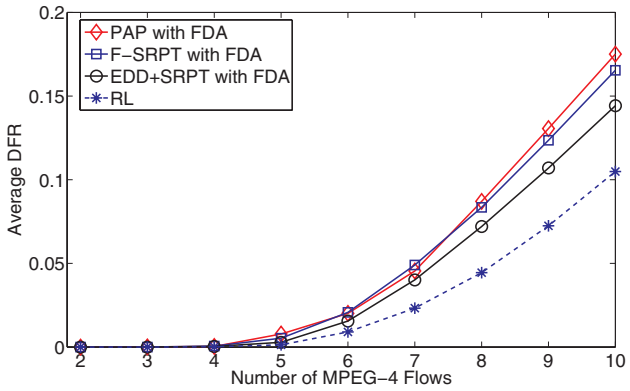

Fig. 4. Comparison of RL scheduler and other schedulers when the number of MPEG-4 flows varies from 2 to 10.

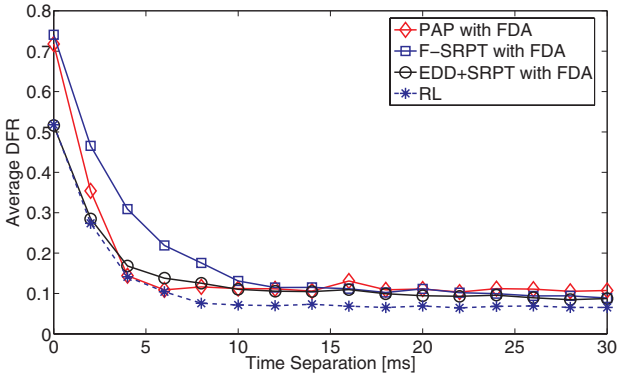

Fig. 5. Effect of time separation on average DFR. The RL scheduler has the smallest average DFR for all ϕ.

start times of flows are separated by ϕ seconds. In other words, flow i starts at time $i\phi$ [1]. Fig. 5 compares the average DFR achieved by RL, F-SRPT, EDD+SRPT, and PAP scheduling algorithms when ϕ varies from 1 to 30 ms. The number of MPEG-4 flows is equal to 9. We can see that our proposed RL scheduler performs better than the other three for all values of ϕ. Furthermore, the performance of RL scheduler is less sensitive to ϕ.

To evaluate the optimality of the RL algorithm, we consider the special case of interest which is when $\phi = 0$, i.e., when all the flows in the system start at the same time. In [18], we show that for this case, the SRPT scheduler is the optimal scheduler. Fig. 6 compares the average DFR achieved by RL algorithm with the optimal case when the number of MPEG-4 flows varies from 2 to 10. We can see that in all cases, RL scheduler provides nearly optimal performance.

As mentioned in Section III, the policy gain and DFR have a linear relationship. We can verify the validity of equation (12) as follows. First, the estimated average DFR is calculated by substituting the policy gain ρ in equation (12). Second, the exact average DFR is measured by counting the number of scheduled frames. Fig. 7 compares these two values. As one can see, $\overline{\text{DFR}}$ in equation (12) under-estimates the exact average DFR, because the gain is only updated when the

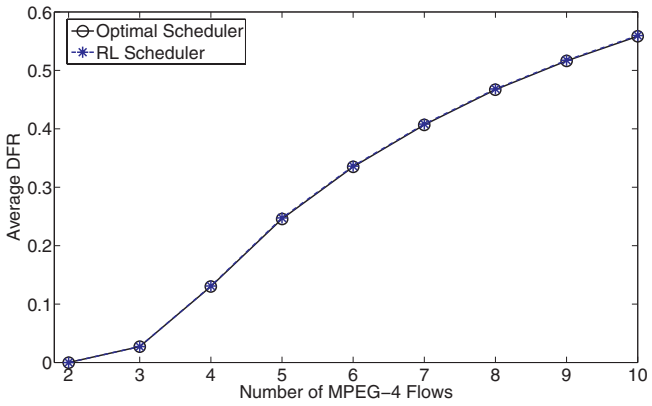

Fig. 6. Comparison of RL scheduler and optimal scheduler for $\phi = 0$. RL scheduler is nearly optimal in this case.

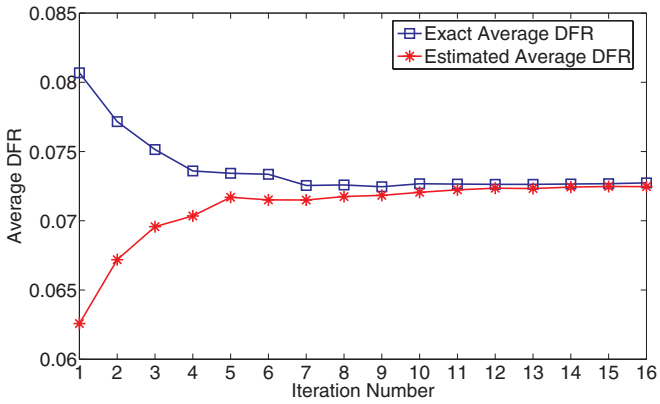

Fig. 7. Comparison of the exact and estimated DFR ($F = 9$).

scheduler takes a greedy action. However, both greedy and exploratory actions affect the exact average DFR. Over time, with more iterations, as the RL scheduler learns the optimal policy and the exploration probability decays, the exact and estimated average DFR converge together. This result verifies the fact that the optimal gain policy yields the minimum average DFR.

V. Conclusions

In this paper, we presented an MDP model for scheduling MPEG-4 flows in high-rate WPANs. The model takes into account the number, hierarchical structure, and pattern of MPEG-4 flows. Using an RL technique called SMART, we proposed a practical algorithm that can lead to finding optimal (or near-optimal) schedules based on the MDP formulation. Simulation results show that our proposed RL scheduler is nearly optimal and performs better than some other scheduling algorithms including F+SRPT [1], EDD+SRPT [2], and PAP [3] regardless of the number of MPEG-4 flows and the value of the time separation parameter.

A high-rate WPAN should be able to support a variety of applications with different QoS requirements. We focused on the video traffic class in this paper. Integrating our proposed scheme with a more versatile scheduler that can handle and recognize different traffic types is part of our future work.

Acknowledgment

This work is supported by Bell Canada and the Natural Sciences and Engineering Research Council of Canada (NSERC) under grant number CRDPJ 320552-04.

References

[1] R. Mangharam, M. Demirhan, R. Rajkumar, and D. Raychaudhuri, "Size matters: size-based scheduling for MPEG-4 over wireless channels," in *Proc. of SPIE/ACM Multimedia Computing and Networking (MMCN)*, Santa Clara, CA, Jan. 2004.

[2] A. Torok, L. Vajda, A. Vidacs, and R. Vida, "Techniques to improve scheduling performance in IEEE 802.15.3 based ad hoc networks," in *Proc. of IEEE Globecom*, St. Louis, MO, Nov. 2005.

[3] S. M. Kim and Y. J. Cho, "Scheduling scheme for providing QoS to real-time multimedia traffics in high-rate wireless PANs," *IEEE Trans. on Consumer Electronics*, vol. 51, no. 4, pp. 1159–1168, Nov. 2005.

[4] "IEEE Std 802.15.3-2003: Wireless medium access control (MAC) and physical layer (PHY) specifications for high rate wireless personal area networks (WPANs)," Sept. 2003.

[5] A. Ziviani, B. E. Wolfinger, J. F. Rezende, O. C. Duarte, and S. Fdida, "Joint adoption of QoS schemes for MPEG streams," *Multimedia Tools and Applications*, vol. 26, no. 1, pp. 59–80, May 2005.

[6] X. Shen, W. Zhuang, H. Jiang, and J. Cai, "Medium access control in ultra-wideband wireless networks," *IEEE Trans. on Vehicular Technology*, vol. 54, no. 5, pp. 1663–1677, Sept. 2005.

[7] C. Hu, H. Kim, J. Hou, D. Chi, and S. Shankar, "Provisioning quality controlled medium access in UWB-operated WPANs," in *Proc. of IEEE Infocom*, Barcelona, Spain, Apr. 2006.

[8] Y. C. Chu and A. Ganz, "Joint scheduling and resource control for QoS support in UWB-based wireless networks," in *Proc. of IEEE MILCOM*, Monterey, CA, Nov. 2004.

[9] R. Zeng and G. S. Kuo, "A novel scheduling scheme and MAC enhancements for IEEE 802.15.3 high-rate WPAN," in *Proc. of IEEE WCNC*, New Orleans, LA, Mar. 2005.

[10] C. Y. Zou and Z. Haas, "Optimal resource allocation for UWB wireless ad hoc networks," in *Proc. of IEEE PIMRC*, Berlin, Germany, Sept. 2005.

[11] K. H. Liu, L. Cai, and X. Shen, "Performance enhancement of medium access control for UWB WPAN," in *Proc. of IEEE Globecom*, San Francisco, CA, Nov. 2006.

[12] S. Moradi and V. W. S. Wong, "Technique to improve MPEG-4 traffic schedulers in IEEE 802.15.3 WPANs," in *Proc. of IEEE ICC*, Glasgow, Scotland, June 2007.

[13] "MPEG-4 part 2: Visual (IS 14496-2), doc. N2502," Oct. 1998.

[14] M. L. Puterman, *Markov decision processes: discrete stochastic dynamic programming*. Wiley-Interscience, 1994.

[15] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. Cambridge, MA: MIT Press, 1998.

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

[17] T. K. Das, A. Gosavi, S. Mahadevan, and N. Marchalleck, "Solving semi-Markov decision problems using average reward reinforcement learning," *Journal of Management Science*, vol. 45, no. 4, pp. 560–574, 1999.

[18] S. Moradi, "A novel scheduling algorithm for video flows in high-rate WPANs," Master's thesis, The University of British Columbia, Vancouver, BC, Canada, Apr. 2007.

[19] P. Kanerva, *Sparse distributed memory*. Cambridge, MA: MIT Press, 1988.

[20] B. Ratitch and D. Precup, "Sparse distributed memories for on-line value-based reinforcement learning," in *Proc. of European Conference on Machine Learning (ECML)*, Pisa, Italy, Sept. 2004.

[21] C. Darken, J. Chang, and J. Moody, "Learning rate schedules for faster stochastic gradient search," in *Proc. of IEEE Workshop on Neural Networks for Signal Processing*, Copenhagen, Denmark, Sept. 1992.

[22] X. Liu, Q. H. Dai, and Q. F. Wu, "Scheduling algorithms analysis for MPEG-4 traffic in UWB," in *Proc. of IEEE VTC-Fall*, Los Angeles, CA, Sept. 2004.