Published in IET Communications
Received on 9th January 2008
Revised on 17th May 2008
doi: 10.1049/iet-com:20080015

ISSN 1751-8628

Dynamic resource allocation for supporting real-time multimedia applications in IEEE 802.15.3 WPANs

W.-K. Kuo   S.-Y. Lien

Institute of Computer and Communication Engineering, Department of Electrical Engineering, National Cheng Kung University, Tainan, Taiwan, ROC
E-mail: wkuo@ee.ncku.edu.tw

Abstract: The IEEE 802.15.3 medium access control (MAC) protocol is an emerging standard for high-rate wireless personal area networks (WPANs), especially for supporting high-quality real-time multimedia applications. Despite defining quality of service (QoS) signalling mechanisms for interoperability between devices, IEEE 802.15.3 does not specify resource allocation algorithms that are left to manufacturers. To guarantee the QoS of real-time variable bit rate (VBR) videos and utilise the radio resource efficiently, the authors propose a dynamic resource allocation algorithm. The proposed bandwidth allocation algorithm is based on a novel traffic predictor. Recently, the variable step-size normalised least mean square (VSSNLMS) algorithm was employed for on-line traffic prediction of VBR videos. However, the performance of the VSSNLMS algorithm significantly degrades due to the abrupt traffic variation occurring at the scene boundary. To tackle this problem, the authors design a novel traffic predictor based on a simple scene detection algorithm and the VSSNLMS algorithm. Analyses using real-life MPEG video traces indicate that the proposed traffic predictor significantly outperforms the VSSNLMS algorithm with respect to the prediction error. The performance of the proposed bandwidth allocation algorithm is also investigated by comparing several existing algorithms. Simulation results demonstrate that the proposed bandwidth allocation algorithm surpasses other mechanisms in terms of channel utilisation, buffer usage and packet loss rate.

1 Introduction

Recently, Ultra-Wideband (UWB) has gained considerable attentions for its usability in wireless communication systems. Being able to provide low power consumption, high data rate transmission and low interference to existing radio technologies, UWB has become the promising candidate of high rate wireless personal area networks (WPANs). UWB-based WPANs can support many novel applications, such as home entertainments, real-time multimedia streaming and wireless universal serial bus (WUSB) that connects computers to peripherals. IEEE 802.15.3 working group is organised to work towards a common standard for UWB medium access control (MAC) layer and physical layer. One of the main features of the IEEE 802.15.3 MAC [1] is to support quality of service (QoS) guarantee of multimedia applications.

MPEG videos (e.g. MPEG-1 used by VCD, MPEG-2 used by DVD and MPEG-4 used for network streaming) are expected to be one of the most important indoor applications for WPANs [2]. However, due to the nature of MPEG variable bit rate (VBR) videos such as burstiness and strong long range dependence (LRD), the design of efficient transport mechanisms that are capable of achieving high resource utilisation while still preserving the required QoS of MPEG videos has become a challenging problem [3, 4]. Furthermore, although IEEE 802.15.3 defines the QoS signalling mechanism, it does not specify the resource allocation algorithm which is left to vendor implementations. Due to the above reasons, reliable transport of MPEG VBR video and a guarantee of its QoS over 802.15.3 networks have become difficult. To solve this problem, Tseng *et al.* [5] proposed an on-line traffic predictor called the variable step-size normalised least mean square

(VSSNLMS) algorithm. Based on this predictor, they proposed a bandwidth allocation algorithm to support real-time MPEG VBR video transmission on 802.15.3 networks. Nevertheless, as shown in Section 2, the VSSNLMS algorithm can induce large prediction errors and yield slow convergence upon the occurrence of scene changes. If the traffic predictor cannot provide precise estimation outcomes, the bandwidth allocation algorithm that dynamically regulates video streams using the prediction results cannot furnish fast and accurate control actions in order to achieve high channel utilisation and satisfy the QoS requirements of VBR videos.

In this paper, we first propose a novel traffic predictor that aims at providing simple yet accurate online MPEG video prediction. The proposed predictor is based on a simple scene detection algorithm and the VSSNLMS algorithm. Due to its rapid-tracking ability, the VSSNLMS algorithm is used to predict the traffic within the same scene. Upon detecting a scene change, we update the filter coefficients using only the signals belonging to the new scene. This is done in order to accelerate the re-convergence of the coefficients and to avoid the creation of a large prediction error at the scene boundary. Therefore the proposed predictor offers adaptability not only in the adjustment of the step size and coefficients but also in the dynamic update of the inputs. Based on the proposed predictor, we then design a dynamic bandwidth allocation algorithm to support the QoS transport of real-time MPEG videos over 802.15.3 networks. By means of simulations using real-life MPEG video traces, it is shown that the proposed scheme significantly outperforms other mechanisms with respect to channel utilisation, buffer usage and packet loss rate.

The remainder of this paper is organised as follows. Section 2 discusses related works on real-time VBR video traffic prediction and the deficiencies in previous proposals. Section 2 also presents the proposed traffic predictor and compares the outcomes with the VSSNLMS algorithm. Section 3 provides descriptions of the 802.15.3 MAC and the proposed dynamic resource allocation algorithm. Simulation results validating the claimed performance improvements are presented and discussed in Section 3. Section 4 concludes the paper.

2 Related work on MPEG traffic perdiction and proposed traffic predictor

2.1 MPEG traffic prediction and related work

An MPEG video consists of a sequence of group of pictures (GOPs). Each GOP is composed of three types of frames arranged in a repetitive structure. These three types of frames are, I-frame (Intra-frame), P-frame (Predictive-frame) and B-frame (Bidirectional-Predictive-frame), each with different encoding methods. A GOP pattern is denoted by GOP(M, N), where M is the frame distance from one I-frame to the next I-frame and N is the frame distance between two successive P-frames. For instance, GOP (12, 3) represents the frame sequence 'IBBPBBPBBPBB'. Traffic generated by MPEG and observed at the frame level exhibits a highly fluctuating and nonlinear bit-rate variation resulting from the encoding algorithm itself and the complex content of the audiovisual data, such as camera motion, scenes of high activity and scene changes. Moreover, recent studies using complete video traces reveal that MPEG videos possess strong LRD (i.e. self-similarity) [6–8].

For VBR video, a methodology commonly used to tackle the effects of burstiness and strong LRD is the dynamic resource allocation. The aim of dynamic resource allocation is to adaptively allocate resources to capture the traffic characteristics of VBR video. One of the popular techniques in dynamic resource allocation is the use of traffic prediction, which anticipates the dynamics of future traffic in order to provide faster and more efficient traffic management mechanisms. Several works have been proposed to predict MPEG VBR video [9]. Among these works, Adas [10] presented a traffic predictor which separates the MPEG video sequence into subgroups I, P and B and forecasts each type of frame using the normalised least mean square (NLMS) algorithm which is explained as follows: Given an input vector with p samples $\boldsymbol{X}(n) = [x(n), x(n-1), \ldots, x(n-p+1)]^{\mathrm{T}}$ and a coefficient vector of p coefficients $\boldsymbol{W}_n = [w_n(0), w_n(1), \ldots, w_n(p-1)]^{\mathrm{T}}$, the estimated signal of a pth-order linear predictor is calculated as follows

$$\hat{x}(n+k) = \sum_{l=0}^{p-1} w_n(l)x(n-l) = \boldsymbol{W}_n^{\mathrm{T}}\boldsymbol{X}(n) \tag{1}$$

The prediction error, $e(n)$, is

$$e(n) = x(n+k) - \hat{x}(n+k) = x(n+k) - \boldsymbol{W}_n^{\mathrm{T}}\boldsymbol{X}(n) \tag{2}$$

The coefficient vector is recursively updated as follows

$$\boldsymbol{W}_{n+1} = \boldsymbol{W}_n + \frac{\mu e(n)\boldsymbol{X}(n)}{||\boldsymbol{X}(n)||^2} \tag{3}$$

where $||\boldsymbol{X}(n)||^2 = \boldsymbol{X}(n)^{\mathrm{T}}\boldsymbol{X}(n)$. μ is a constant called step size. Using a large μ leads to faster convergence and a quicker response to traffic changes; whereas, after convergence, the predicted traffic exhibits significant fluctuations. In contrast, the use of a small μ results in slower convergence with less fluctuation thereafter.

The advantage of Adas's method is that it neither demands any prior information regarding video statistics nor assumes the signal to be stationary. Moreover, it is easy to implement. Thus this method is particularly suitable for

doi: 10.1049/iet-com:20080015

on-line VBR video prediction. However, its performance significantly deteriorates for applications with rapid traffic variations and frequent scene changes (e.g. entertainment and broadcast videos). This is because the NLMS algorithm converges slowly when there is highly fluctuating traffic caused by scene changes, thus resulting in large prediction errors at the scene boundaries. To overcome this problem, Tseng *et al.* [5] proposed a traffic predictor by modifying the variable step-size least mean square (VSSLMS) algorithm [11]. The idea of VSSLMS is that a large prediction error increases the step size in order to provide faster tracking, whereas a small prediction error decreases the step size to reduce the misadjustment. The VSSLMS is the same as the LMS algorithm with the exception of the dynamic adjustment of the step size, μ_n, as follows

$$\mu_n' = \alpha\mu_{n-1} + \gamma e_{n-1}^2 \tag{4}$$

with $0 < \alpha < 1$, $\gamma > 0$ and

$$\mu_n = \begin{cases} \mu_{\max} & \text{if } \mu_n' > \mu_{\max} \\ \mu_{\min} & \text{if } \mu_n' < \mu_{\min} \\ \mu_n' & \text{otherwise} \end{cases} \tag{5}$$

where $0 < \mu_{\min} < \mu_{\max}$. The constant $\mu_{\max}$ is chosen to ensure that the mean square error (MSE) is bounded; $\mu_{\min}$ is selected as a compromise between the desired level of steady-state misadjustment and the required tracking capability. The parameter α is selected from the range (0, 1) to provide exponential forgetting; usually, a small value of γ is chosen in order to control the convergence speed and the level of misadjustment. Tseng *et al.* [5] normalised the coefficient vector of VSSLMS as (4) and used VSSNLMS to denote their predictor.

Although the VSSNLMS can automatically adjust the step size to accommodate the rapid traffic variation at scene boundaries, it still has some shortcomings. Fig. 1 shows the traffic prediction of VSSNLMS for a part of 'Soccer' [12] I, P, and B subsequence, respectively. In this experiment, we set $p = 5$, $\mu_{\max} = 0.58$, $\mu_{\min} = 0.03$, $\alpha = 0.98$ and $\gamma = 6 \times 10^{-9}$. All these parameters are selected by exhaustive search in order to give the minimal MSE. From Fig. 1, we have the following observations: First, for I- and B-frame, the bit rate varies steeply at the points of scene change, while within the scenes, the traffic fluctuates within a limited range. Moreover, VSSNLMS produces large prediction error and converges slowly when scene changes happen. This is because the step size increases drastically due to the abrupt traffic variation at scene boundaries. Although larger step size helps the coefficients move towards the optimum faster, an extreme change of step size may make the coefficients jump over the optimum. This can produce large misadjustment and excess MSE, which further necessitates the re-convergence of coefficients. Although one can choose a small value of $\mu_{\max}$ to relieve the effect of drastic increase of step size, a small $\mu_{\max}$ will reduce the predictor's ability to track scene changes. Furthermore, the selection of γ and α values also affects the convergence speed and the degree of prediction error. A small value of γ makes the predictor unable to detect scene change and produce large prediction error at scene boundary. Oppositely, a large value of γ increases the value of step size even through there is actually no scene change,

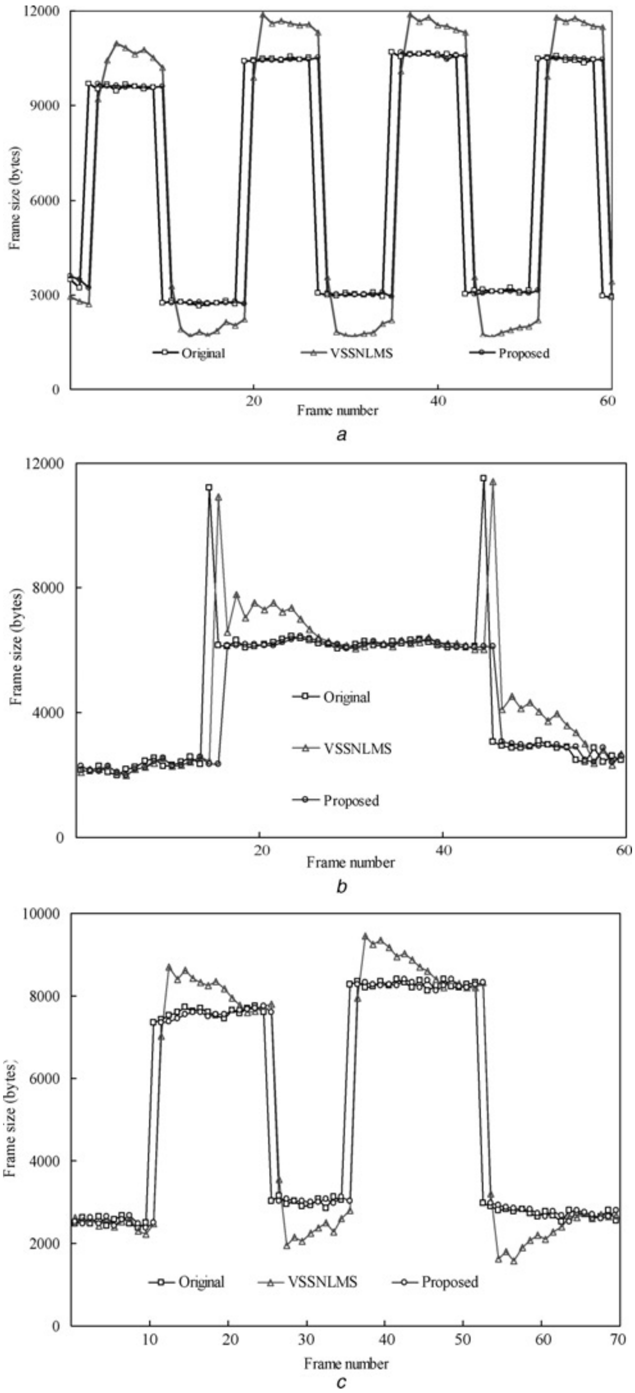

Figure 1 *Prediction results for VSSNLMS and the proposed traffic predictor*

a VSSNLMS prediction result of I-frame
b VSSNLMS prediction result of P-frame
c VSSNLMS prediction result of B-frame

leading to a large traffic fluctuation within a scene. The value of α depends on the autocorrelations of VBR videos. Since different videos have distinct traffic-variation characteristics, it is difficult to pre-determine proper values of $\mu_{\max}$, γ and α so as to guarantee the fast convergence and the creation of small prediction errors when scene changes happen, particularly for real-time VBR videos. Second, for P-frame, the bit rate varies as a spike with a scene change, whereas traffic fluctuates within a certain extent inside a scene. We observe that VSSNLMS produces large prediction error at the frame next to a scene change and converges slowly subsequent to the spike. This is because the spike makes the step size vary violently and makes coefficients overleap the optimum. After a sharp increase in the bit rate of the frame corresponding to the scene change, the traffic of the next frame decreases. However, VSSNLMS increases the bit rate of the frame next to the scene change with large coefficients, leading to large prediction error. Furthermore, it also requires the re-convergence of coefficients following the spike to accommodate the traffic variation of the new scene, resulting in slow convergence.

2.2 Proposed traffic predictor

In this section, we propose a novel traffic predictor for I-, P-, B-frame based on the discussion in the previous section. The objective of the proposed predictor is to rapidly track the traffic variation as well as produce smaller prediction errors, especially when scene changes occur. We apply the same predictor to I- and B-frames since the traffic variations in these two types of frames are similar. Another predictor is developed for sub-sequence P because of its unique characteristics regarding bit-rate variation.

2.2.1 I- and B-frame predictor:

The procedure of the proposed I- and B-frame predictor is shown in Fig. 2.

In L1, we set the initial value for *scene_mark*. In L3, we identify a scene change if the traffic variation between two consecutive frames exceeds a certain multiple of the average frame size within a scene, where *scene_mark* denotes the starting frame of the current scene, i is the index of the

L1 *scene_mark*=1; //initially, *scene_mark* is the first video frame

L2 For (i=1; i<=*EndOfVideo*; i++) { // i=1 is the first video frame

L3 If ($i \geq 2$ and $\dfrac{|x(i) - x(i-1)|}{\sum_{n=scene\_mark}^{i-1} x(n) \Big/ (i - scene\_mark)} > gap$) //scene change occurs

L4 $scene\_mark = i$;

L5 $\hat{x}(i+1) = x(i)$;

L6 Else // within a scene

L7 If ($i - scene\_mark + 1 < p$)

L8 $\hat{x}(i+1) = \dfrac{\sum_{n=scene\_mark}^{i} x(n)}{i - scene\_mark + 1}$;

L9 Else

L10 $\hat{x}(i+1)$ = VSSNLMS ();

L11 }

Figure 2 *Procedure of the proposed I- and B-frame predictor*

current frame, $x(i)$ is the size of the ith frame, *gap* is a constant selected based on the traffic characteristics. This is based on the observation that the scene change corresponds to the frame with a sharp variation in the bit rate. In L4, if the scene change is identified, we assign the frame at which the scene change occurs to *scene_mark* (i.e. the start of a new scene). In L7, we check whether the number of frames belonging to the new scene is less than the order (i.e. p). If yes, we forecast the traffic as L5 and L8, where $\hat{x}(i+1)$ is the frame size averaged over the sizes of the first frame to the current frame corresponding to the new scene. It is noteworthy that in L5 and L8, we predict the first p frames according to their average. There are two reasons for considering this: (1) within a scene, the traffic varies within a limited range and (2) the MPEG traffic demonstrates a high degree of self-similar behaviour. Hence, a significant correlation exists between the variability in the sizes of adjacent frame of the subgroups I, P and B [6–8]. Indeed, L5 and L8 can eliminate the generation of a large prediction error caused by a drastic change in the step size. In L10, once there are p frames that relate to the new=scene, we update the coefficients and inputs and then predict the successive frames using VSSNLMS until the next scene change occurs. (When a scene change is detected, we update coefficient vectors as $\boldsymbol{W}_n = [1/p \ 1/p \cdots 1/p]^{\mathrm{T}}$. This is the same as taking the average value). More specifically, we remove all the coefficients generated by the frames of the previous scene and use only the frames corresponding to the new scene to train the VSSNLMS algorithm. This is because the frames belonging to the same scene have similar traffic-variation properties (i.e. traffic fluctuates within a limited range), while the characteristics of interscene bit-rate variation are different. Hence, using the frames of the same scene to train the filter helps in the rapid re-convergence of the coefficients and self-adaptation to different traffic-variation properties.

2.2.2 P-frame predictor:

Since the spike makes the VSSNLMS produce large prediction errors at both the frame corresponding to the spike and the frame next to the spike and slows the re-convergence rate of coefficients, it is important to eliminate the effect of the spike on P-frame prediction. Therefore, we modify the scene change detection algorithm used in predicting I and B subsequences, and the manner of updating VSSNLMS inputs when scene changes occur. The procedure of the proposed P-frame predictor is shown in Fig. 3.

In L1 and L2, we set the initial value for *scene_mark* and $\delta(i)$, respectively. In L4 and L5, if the frame belongs to the same scene, we update the average frame size within the scene. When computing the average frame size, we exclude the bit rate of the frame corresponding to the spike (i.e. $x(scene\_mark)$), *scene_mark* is used to mark the frame at which a spike occurs). In L8 and L9, if a scene change is decided upon, we assign the starting frame of the new scene (i.e. the spike) to *scene_mark*. In L10, we assign $\hat{x}(scene\_mark + 1)$ as $\hat{x}(scene\_mark)$ instead of $x(scene\_mark)$

doi: 10.1049/iet-com:20080015

```
L1   scene_mark=0;   // initial value of scene_mark
L2   δ(i)=0;         // initial value of δ(i)
L3   For (i=1; i<=EndOfVideo; i++) {                 // i=1 is the first
        video frame
L4       If (i > scene_mark +1)
L5           δ(i)= |x(i) - x(i - 1)| / ( Σ_{n=mark_scene+1}^{i-1} x(n) / (i - scene_mark - 1) );
L6       If (i = scene_mark +1)
L7           x̂(i+1)= x(scene_mark +1);
L8       Elseif (δ(i)(> gap)
L9           scene_mark = i;    // scene_ mark is the spike
L10          x̂(scene_mark + 1) = x̂(scene_mark);
L11      Elseif (i = scene_mark < p)
L12          x̂(i+1) = Σ_{n=scene_mark+1}^{i} x(n) / (i - scene_mark);
L13      Else
L14          x̂(i+1)= VSSNLMS ();
L15  }
```

Figure 3 *Procedure of the proposed P-frame predictor*

to eliminate the effect of the spike on traffic prediction. In L7 and L12, prior to having p frames that relate to the new scene (i.e. from the $(scene_mark+1)$th frame to the $(scene_mark+p)$th frame with the exception of the $x(scene\_mark)$, which is discarded), we predict traffic by the average frame size. In L14, on obtaining the p frames, we update the coefficients and train the filter using the new set of signals. (When a scene change is detected, we update coefficient vector as $\boldsymbol{W}_n = [1/p \;\; 1/p \cdots 1/p]^{\mathrm{T}}$. This is the same as taking the average value).

2.3 Performance results and comparisons

In this section, we evaluate and compare the performance of the proposed predictor with that of VSSNLMS. For the purpose of simulations, we use MPEG-4 video traces generated by the Technical University of Berlin [12]. We employ

$$\mathrm{SNR}^{-1}(\%) = 100\frac{\sum e(n)^2}{\sum x(n)^2} = 100\frac{\sum (x(n)-\hat{x}(n))^2}{\sum x(n)^2}$$

as the performance measurement.

Fig. 1 shows the prediction result of the VSSNLMS algorithm and the proposed scheme for a part of 'Soccer' I, P and B subsequences. As mentioned previously, for the VSSNLMS algorithm, we set $p=5$, $\mu_{\max}=0.58$, $\mu_{\min}=0.03$, $\alpha=0.98$ and $\gamma=6\times10^{-9}$. For the proposed scheme, the p, $\mu_{\max}$, $\mu_{\min}$, α and γ are the same as those used in the VSSNLMS algorithm; further, for I-, P- and B-frames, the *gap* values are 0.2, 0.8 and 0.2, respectively. From this figure, we observe that for I, P and B subsequences, the proposed scheme converges faster and produces smaller prediction errors, particularly when scene changes occur. This is because the proposed scheme can quickly adapt itself to different traffic-variation characteristics, for instance scene changes or the introduction of a new video.

Table 1 lists the SNR^{-1} values of VSSNLMS and the proposed scheme for subgroups I, P and B of different video traces. For comparison, we provide one set of VSSNLMS prediction results, namely, $\mathrm{VSSNLMS}_{\mathrm{optimal}}$. For $\mathrm{VSSNLMS}_{\mathrm{optimal}}$, different video traces use different parameter values in order to achieve the optimal performance. The values of the VSSNLMS parameters for different movies are determined by exhaustive search in order to give the minimal MSE. Similarly, for the proposed scheme, we provide two sets of prediction results, namely, $\mathrm{Proposed}_{\mathrm{Soccer}}$ and $\mathrm{Proposed}_{\mathrm{optimal}}$. In the former, all the parameter values are the same as those in the previous experiment (i.e. Fig. 1, the optimal parameter values of 'Soccer'), and in the latter, the parameter values for different movies are the same as those of $\mathrm{VSSNLMS}_{\mathrm{optimal}}$. Moreover, for all the video

Table 1 SNR^{-1} comparison with VSSNLMS

	$\mathrm{VSSNLMS}_{\mathrm{optimal}}$			$\mathrm{Proposed}_{\mathrm{optimal}}$			$\mathrm{Proposed}_{\mathrm{Soccer}}$		
	I	P	B	I	P	B	I	P	B
Mr. Bean	3.93	8.47	4.97	2.19	4.53	2.78	2.95	5.17	3.45
Formula 1	3.16	6.61	2.76	1.77	3.24	1.28	2.63	4.54	1.86
Soccer	5.82	9.83	2.84	3.37	5.06	1.19	3.37	5.06	1.19
The Firm	2.08	10.95	4.72	0.93	5.67	3.37	1.42	6.32	4.23
Star Trek	4.74	7.72	4.03	2.65	3.14	2.21	3.14	4.23	3.27
Robin Hood	2.65	7.34	2.65	1.04	3.51	1.04	2.06	4.46	1.78
Boulevard	3.51	8.86	6.78	1.72	4.32	3.42	2.79	4.91	4.32
N3 talk	4.67	5.98	3.19	2.48	2.79	1.25	3.58	3.85	2.06

doi: 10.1049/iet-com:20080015

traces, the *gap* values for I-, P-, and B-frames are 0.2, 0.8 and 0.2, respectively. From Table 1, we can see that (1) comparing the results of $VSSNLMS_{optimal}$ with those of $Proposed_{optimal}$, the proposed scheme decreases the error in I-, P- and B-frames by approximately 42–61%, 47–59% and 41–61%, respectively. Apparently, the proposed scheme significantly outperforms the VSSNLMS algorithm. Second, comparing the results of $VSSNLMS_{optimal}$ with those of $Proposed_{Soccer}$, the performance of the proposed scheme is still better than that of $VSSNLMS_{optimal}$ for all the video traces. This is because, for the proposed scheme, the VSSNLMS algorithm is only used to predict the intra-scene traffic variation that changes within a limited extent. Thus, it is much easier to select one set of parameter values that can achieve a better performance than the VSSNLMS algorithm for different videos. Moreover, since the prior traffic statistics of real-time video are unavailable, it is very difficult to predetermine the optimal values of the predictor parameters for different videos in advance. It is more practical that one selects a set of parameter values that can achieve good performance for different videos and uses the same set of parameter values to predict different videos. The results of Table 1 reveal that in real situations, the proposed predictor can easily achieve better performance than the optimal performance of the VSSNLMS algorithm.

3 IEEE 802.15.3 MAC and proposed dynamic bandwidth allocation algorithm

3.1 Introduction of 802.15.3 MAC

The IEEE 802.15.3 defines a wireless *ad hoc* network (piconet) that allows a number of devices to communicate directly with each other. One device of the piconet is elected as the piconet coordinator (PNC) that is responsible for coordinating radio resources among the other devices within a piconet, associating and disassociating devices, authenticating new devices etc. The channel time of the 802.15.3 MAC is divided into superframes (SFs). Each SF consists of three parts: a beacon frame, an optional contention access period (CAP) and a channel time allocation period (CTAP). At the beginning of each SF, the PNC broadcasts the beacon frame that is used to provide management information, network-wide timing synchronisation and the channel time allocation information for traffic streams. The CAP is used for transmission of commands and asynchronous data (i.e. traffic with no specific QoS requirements). During the CAP, devices access the channel using the carrier sense multiple access/collision avoidance with back-off procedure. The CTAP is composed of channel time allocations (CTAs) and optional management CTAs (MCTAs). The CTAP is used to support both asynchronous and isochronous traffic (i.e. applications with specific QoS requirements) transmission. In each SF, during the CAP, devices request CTAs for the next SF to the PNC which is responsible for the resource allocation. Specifically, if a device needs CTA, it will issue a Channel Time Request command (CTRq) during the CAP that indicates the required channel time of the next SF. On reception of this request, the PNC responds with a Channel Time Response (CTRp) command and, if the required radio resource is available, it allocates the requested channel time to the stream. Then, in the next SF, the PNC broadcasts the allocated CTAs in the beacon frame to all streams. MCTAs are a type of CTA used for communication between devices and the PNC. MCTAs are either assigned to a specific source/destination pair and use time division multiple access to communicate or they are shared CTAs that are accessed using the slotted aloha protocol.

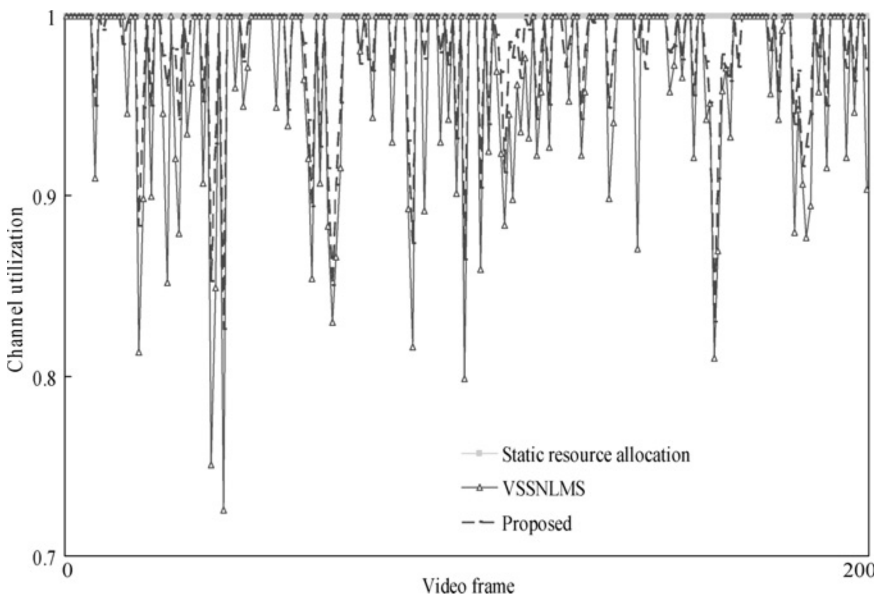

Figure 4 *Channel utilisation comparison*

doi: 10.1049/iet-com:20080015

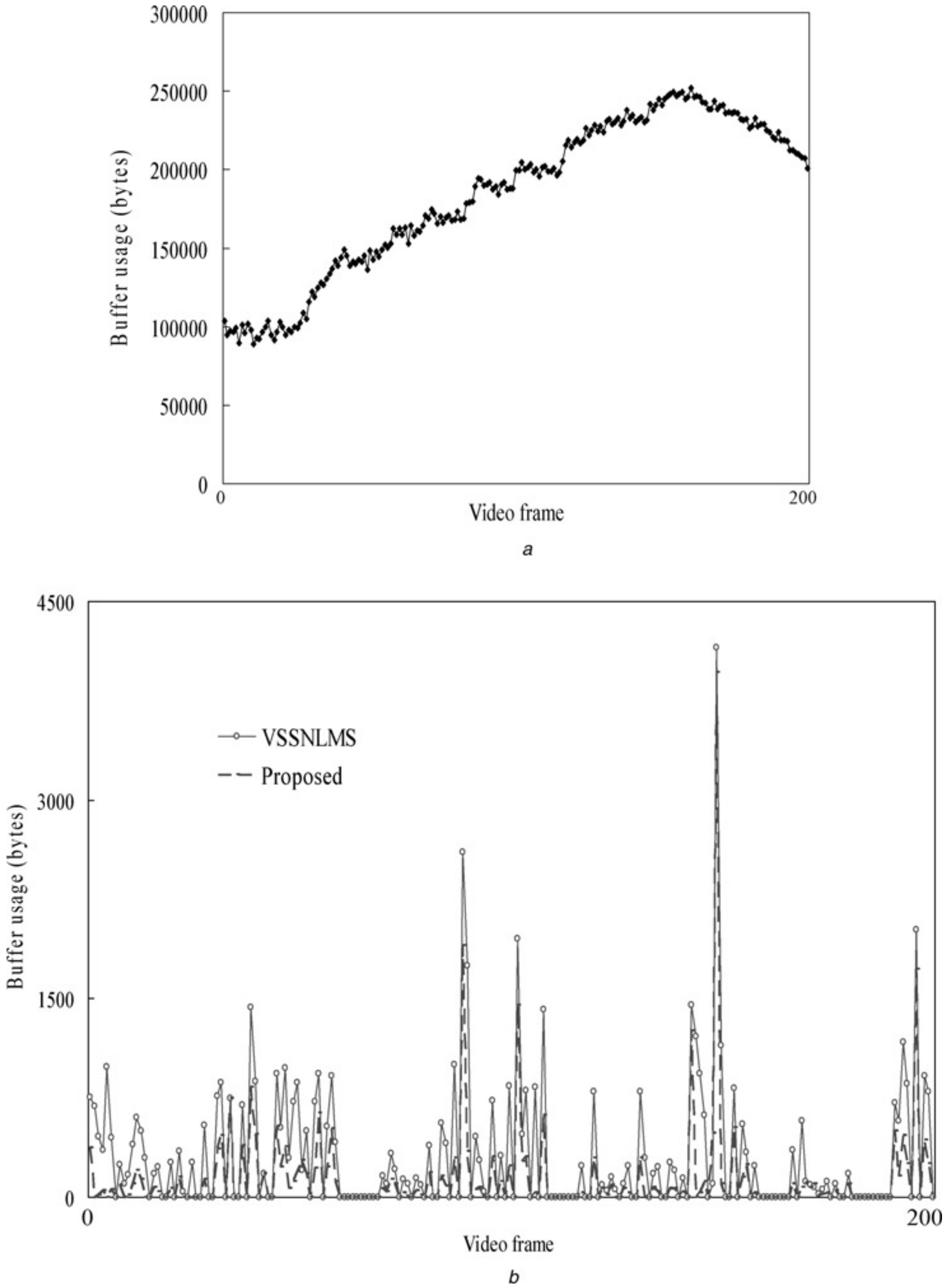

Figure 5 *Buffer usage comparison*

a Buffer usage of static resource allocation scheme
b Buffer usage of proposed scheme and the scheme in which the PS_{n+1} in (6) is fed with the prediction results of the VSSNLMS algorithm

Although 802.15.3 standard defines signalling mechanism for CTA negotiation, it does not specify the resource allocation algorithm that can properly distribute CTAs in order to guarantee QoS requirements of different applications. To address this issue, we propose a dynamic bandwidth allocation algorithm which allows the 802.15.3 standard to provide QoS transport of real-time VBR videos.

3.2 Proposed dynamic bandwidth allocation algorithm

Apparently, static resource allocation (i.e. allocating a fixed bandwidth for an entire video transmission) is not suitable for real-time VBR videos. If resources are over-allocated, the network incurs low resource utilisation. On the other hand, if resources are under-allocated, the video stream may suffer intolerable delays, jitter or packet loss. To achieve high resource utilisation while supporting the QoS transport of real-time VBR videos over 802.15.3 WPANs, we propose a dynamic bandwidth allocation algorithm described as follows.

In the proposed scheme, we choose the duration of a SF as the inter-arrival time of video frame. During the CAP in each SF, the device sends the CTRq command that indicates the bandwidth requirement of the next SF to the PNC.

doi: 10.1049/iet-com:20080015

Moreover, to achieve a more reliable transmission for CTRq/CTRp, an MCTA is allowed to be used, instead of a CAP. The bandwidth requirement includes the amount of data in the buffer (if it has any) and the predicted size of the video frame in the next SF as (6) shows

$$\mathrm{BR}_{n+1} = \mathrm{BS}_n + \mathrm{PS}_{n+1} \tag{6}$$

where BR_{n+1} is the bandwidth request for the next (i.e. the $(n+1)$th) SF, BS_n is the bandwidth requirement for transmitting all queued packets in the current (i.e. the nth) SF, PS_{n+1} is the predicted size of the $(n+1)$th video frame. The proposed scheme adopts the prediction results of the traffic predictor presented in Section 2.2 to request the bandwidth requirement for real-time VBR video. After receiving the bandwidth request, the PNC will allocate the resource which is equal to BR_{n+1} to the stream in the next SF and responds with a CTRP command. In the beacon of the next SF, the PNC broadcasts the CTA information to devices within the same piconet. Therefore the PNC can dynamically allocate radio resource in order to achieve high channel utilisation and sustain the QoS requirements of real-time VBR videos. If the predicted size of the video frame is less than the actual required bandwidth, the device queues the difference which is considered as a part of bandwidth request (i.e. PS_{n+1} in (6)) of the next SF.

3.3 Performance evaluation and discussion

In this section, we evaluate the performance of the proposed scheme, the scheme in which the PS_{n+1} in (6) is fed with the prediction results of the VSSNLMS algorithm, and the static resource allocation scheme in terms of channel utilisation, buffer usage and packet loss rate. We implement different resource allocation algorithms on NS-2 [13] with the 802.15.3 MAC module developed by Intel [14], and modify the module to satisfy our needs. We assume ideal error-free wireless channels, since the focus is on the 802.15.3 MAC, and traffic prediction and dynamic resource allocation for real-time VBR videos. In our simulation, a device transmits a VBR video to another device within the same piconet. The data rate of each DEV is set as 55 Mbps, which is the maximum data rate of IEEE 802.15.3. The MPEG video used in the simulation is 'Soccer' with a frame rate of 25 fps (i.e. the 802.15.3 superframe duration is 40 ms), mean-rate 1.1 MKbps and peak-rate 3.6 Mbps. The simulation runs for 3600s, which is the length of the video stream. For the proposed scheme and the scheme in which the PS_{n+1} in (9) is fed with the prediction results of the VSSNLMS algorithm, the parameters are all the same as $\mathrm{Proposed}_{\mathrm{optima}}$ and $\mathrm{VSSNLMS}_{\mathrm{optimal}}$ described in Section 2.3. For the static resource allocation scheme, we allocate mean rate of the video stream in each SF.

Fig. 4 illustrates the channel utilisation of a part of Soccer for the three schemes. We observe that the proposed scheme achieves higher channel utilisation than the scheme in which the PS_{n+1} in (6) is fed with the prediction results of the VSSNLMS algorithm. This is because the proposed traffic predictor is more accurate than the VSSNLMS. For the entire movie, the average channel utilisation of the proposed scheme, the scheme in which the prediction results using VSSNLMS, is about 97.8% and 95.6%, respectively.

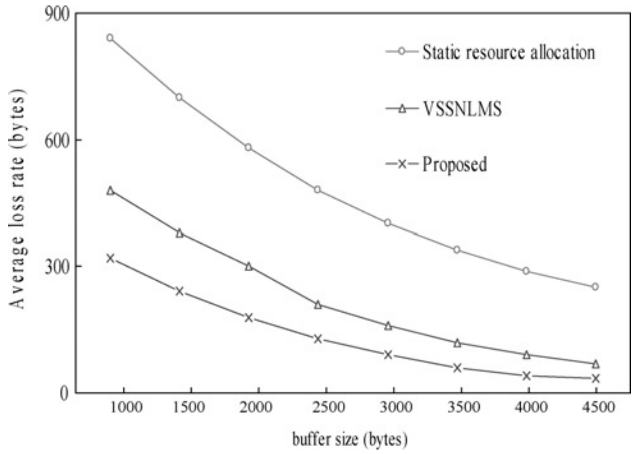

Figure 6 *Average loss rate comparison*

Fig. 5 depicts the buffer usage of a portion of Soccer for the three schemes. We observe that the proposed scheme shows the best buffer usage, whereas the performance of the static resource allocation scheme is the worst. For the entire movie, the average buffer usage of the proposed scheme, the scheme in which the PS_{n+1} in (6) is fed with the prediction results of the VSSNLMS algorithm, and the static resource allocation scheme is 263, 437 and 12 361 bytes, respectively. Moreover, the maximum buffer occupancy of the three schemes is 9684, 13 985 and 594 746 bytes, respectively. Apparently, the proposed scheme can save buffer space most effectively. Furthermore, comparing Fig. 4 with Fig. 5*a*, we find that although the static resource allocation scheme has the highest channel utilisation among the three schemes, it also results in the longest queue size. This is because static bandwidth allocation is not sufficient to transmit the bursts of packet arrivals and most packets are thus queued at the transmitter's buffer.

Fig. 6 shows the average packet loss rate for the three schemes. When the buffer capacity is fixed, we would prefer the scheme with the lowest packet loss rate since it can provide the best video quality. From Fig. 6, we find that the proposed scheme has the lowest packet loss rate and thus achieves the best video quality among the three schemes for the same buffer size.

4 Conclusion

In this paper, we first presented that the VSSNLMS algorithm produces large prediction errors and converges

doi: 10.1049/iet-com:20080015

slowly when scene changes occur. Furthermore, it is very difficult to determine beforehand the appropriate parameters of the VSSNLMS algorithm to guarantee fast convergence and small prediction errors when scene changes occur, particularly for real-time VBR video in which traffic characteristics are unknown beforehand. According to our observations, we propose a novel traffic predictor designed for MPEG VBR videos, which are expected to become a major application in WPANs. The novelty of the proposed scheme, as compared to that of previous proposals, is its capability of self-adaptation to varying video characteristics, for example scene changes or the introduction of a new video. This adaptability is with regard to not only the adjustment of the step size and filter coefficients but also the dynamic update of inputs. The conducted simulation results, in comparison with the results for the VSSNLMS algorithm, show that the proposed scheme can significantly reduce the prediction error. Based on the proposed predictor, we developed a dynamic resource allocation mechanism for QoS transport of real-time VBR videos over IEEE 802.15.3 WPANs. Simulation results comparing the proposed mechanism with other resource allocation schemes indicated that the proposed scheme can achieve better performance in terms of packet delay, channel utilisation, buffer usage and packet loss rate.

5 References

[1] IEEE: 'Part 15.3: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for High Rate Wireless Personal Area Networks (WPANs)', IEEE Std 802.15.3–2003, September 2003

[2] STROH S.: 'Ultra-Wideband: Multimedia unplugged', *IEEE Spectrum*, 2003, **40**, (9), pp. 23–27

[3] GARRETT M., WILLINGER W.: 'Analysis, modeling and generation of self-similar VBR video traffic'. Proc. ACM SIGCOMM Comput. Commun. Rev., 1994, **24**, (4), pp. 269–280

[4] KNIGHTLY E., SHROFF N.: 'Admission control for statistical QoS: theory and practice', *IEEE Netw.*, 1999, **13**, (2), pp. 20–29

[5] TSENG Y.-H., WU H.-K., CHEN G.-H.: 'Scene-change aware dynamic bandwidth allocation for real-time VBR video transmission Over IEEE 802.15.3 wireless home networks', *IEEE Trans. Multimedia*, 2007, **9**, (3), pp. 642–654

[6] SEELING P., FITZEK F.H.P., REISSLEIN M.: 'Video traces for network performance evaluation' (Springer, 2006)

[7] FITZEK F.H.P., REISSLEIN M.: 'MPEG-4 and H.263 video traces for network performance evaluation', *IEEE Netw.*, 2001, **15**, (6), pp. 40–54

[8] SEELING P., REISSLEIN M., KULAPALA B.: 'Network performance evaluation using frame size and quality traces of single-layer and two-layer video: a tutorial', *IEEE Commun. Surv. Tutorials*, 2004, **6**, (2), pp. 58–78

[9] DOULAMIS A., DOULAMIS N., KOLLIAS S.: 'An adaptable neural network model for recursive nonlinear traffic prediction and modeling of MPEG video sources', *IEEE Trans. Neural Netw.*, 2003, **14**, (1), pp. 150–166

[10] ADAS A.M.: 'Using adaptive linear prediction to support real-time VBR video under RCBR network service model', *IEEE/ACM Trans. Netw.*, 1998, **6**, (5), pp. 635–644

[11] KWONG R.H., JOHNSTON E.W.: 'A variable step size LMS algorithm', *IEEE Trans. Signal Process.*, 1992, **40**, pp. 1633–1642

[12] Mpeg-4 and h.263 video traces for network performance evaluation, http://www.tkn.tu-berlin.de/research/trace/trace.html

[13] NS-2: 'NS-2 network simulator', http://www.isi.edu/nsnam/ns/

[14] Intel Lab: IEEE 802.15.3 MAC Model, http://www.macroangel.com/nslast.zip