Research Article

Implementing Statistical Multiplexing in DVB-H

Mehdi Rezaei,[1] Imed Bouazizi,[2] and Moncef Gabbouj[3]

[1] *Faculty of Electrical and Computer Engineering, University of Sistan & Baluchestan, Zahedan 98135-987, Iran*
[2] *Media Laboratory, Nokia Research Center, 33720 Tampere, Finland*
[3] *Department of Signal Processing, Tampere University of Technology, 33720 Tampere, Finland*

Correspondence should be addressed to Mehdi Rezaei, mehdi.rezaei@ieee.org

A novel technique for implementing statistical multiplexing (StatMux) of broadcast services over Digital Video Broadcasting for Handhelds (DVB-H) channels is proposed. DVB-H uses a time-sliced transmission scheme to reduce the power consumption used for radio reception part in DVB-H receivers. Due to the time-sliced transmission scheme, the implementation of known StatMux methods for DVB-H application presents some challenges which are addressed in this paper. The proposed StatMux technique is implemented in conjunction with the time-slicing transmission scheme. The combination is similar to a time division multiplexing (TDM) scheme. The proposed StatMux method considerably decreases the end-to-end delay of DVB-H services while it maximizes the usage of available bandwidth. Moreover, the proposed method can effectively decrease the channel switching delay of DVB-H services. Simulation results show a high performance for the proposed StatMux method.

1. Introduction

DVB-H (Digital Video Broadcasting for Handheld terminals) is an ETSI standard specification for bringing broadcast services to battery-powered handheld receivers [1]. DVB-H is mostly based on the successful DVB-T standard specification for digital terrestrial television, adding to it a number of features designed to take into account the limited battery life of small handheld devices, and the particular environments in which such receivers typically operate [2].

To reduce the power consumption in handheld terminals, the service data is time sliced and then transmitted over the channel as bursts at a significantly higher bit rate than the bit rate of the audiovisual service [3]. Time-slicing enables a receiver to become active during only a fraction of the time when receiving the bursts of the requested service to conserve battery power used for radio reception parts.

To indicate to the receiver when to expect the next burst, the time (delta-*t*) to the beginning of the next burst is indicated within the burst. Between bursts, data of the same service is not transmitted, allowing other services to use the bandwidth.

Time-slicing also enables the receiver to monitor neighboring cells during the off-times (between bursts). By switching of the reception from one *Transport Stream* to another during an off period, it is thus possible to accomplish a quasioptimum handover decision as well as seamless service handover.

DVB-H also utilizes additional *Multiprotocol Encapsulation Forward Error Correction* (MPE-FEC) to further improve mobile and indoor reception performance of DVB-T. Time Slicing and MPE-FEC are implemented in a network element called *Internet Protocol (IP) Encapsulator*. Moreover, a *Time Division Multiplexing* (TDM) is implemented by the IP encapsulator on a number of time-sliced services to fill a DVB-T transmission channel.

In video broadcasting over DVB-H, the video signals are encoded to variable bit rate (VBR) bit streams to provide a better average quality for reconstructed video. VBR encoding can provide a better average quality and compression performance at the expense of more delay in the system [4, 5]. Relationships between variations in the bit rate (or delay), video quality, and power consumption of DVB-H receiver are explored in [5]. When VBR bit streams are broadcasted over DVB-H, it would be beneficial to use a type of time

domain statistical multiplexing instead of a deterministic TDM at the IP encapsulator. In statistical multiplexing, a fixed bandwidth communication channel is virtually divided into several VBR channels. The link sharing is adapted to the instantaneous traffic demands of the data streams that are transferred over each channel. Statistical Multiplexing is used in many communication applications to improve the overall performance of communication channels in terms of bandwidth efficiency, end-to-end delay, and data drop rate.

Due to the time-sliced transmission scheme in DVB-H, implementation of StatMux in DVB-H has some associated difficulties. In the time-sliced transmission scheme, when a time slice is processed by the IP encapsulator, the time division information for the next time slice (typically in few seconds later) of the same service should be known to be signaled to the receivers. In StatMux the time divisions should vary proportionally to the instantaneous bit rate of bit streams while the estimation of exact time division information for the next time cycle is almost impossible.

In this paper a novel technique for performing StatMux in conjunction with time slicing in DVB-H, implemented in the IP encapsulator, is proposed. To the best of our knowledge, this is the first work in the relevant literature where a method for performing StatMux in conjunction with time slicing for DVB-H is proposed.

The rest of the paper is organized as follows: background information for StatMux in DVB-H application is provided in Section 2. In Section 3, the proposed StatMux and time-slicing methods for DVB-H application are explained. Simulation results are provided in Section 4. The paper is closed with conclusions in Section 5.

2. Statistical Multiplexing in DVB-H

StatMux has been studied in many digital communication applications. Terrestrial and Satellite Digital TV, Internet TV, Video on Demand, and other forms of video communication over (Asynchronous Transfer Mode) ATM networks are relevant examples to DVB-H application. Significant past research efforts have developed statistical models for VBR video traffic [6–12]. Video traffic models are used for modeling multiplexing processes and producing synthetic traffics for research simulations. Many other research efforts have modeled the performance of StatMux that is required for resource management and controlling the QoS in some application [13–26]. Moreover, a number of researchers have tried to improve the performance of StatMux by different means such as traffic smoothing, synchronization and scheduling [27–33]. Generally, the performance of StatMux can be defined based on the operating point of a multiplexer in a three-dimensional space including data drop rate, bandwidth usage, and delay dimensions.

Video streaming over DVB-H channel is very different from other applications. Therefore, StatMux in DVB-H should be verified according to the special conditions that exist in this application. From the traffic modeling point of view, most of the earlier works focused on uncontrolled VBR video, while due to standard buffering constraints in DVB-H

standard, the use of controlled VBR video is preferred [34]. While an almost constant quantization parameter (QP) is used for encoding video in uncontrolled VBR, a controlled VBR video bit stream is encoded by a loose rate control with a buffering constraint. The buffer size that defines the amount of variations in the bit rate is limited by the video coding standards and also by DVB-H standard. The statistical properties of a controlled VBR video bit stream differ from those of uncontrolled VBR bit stream. Therefore, the performance models proposed for StatMux of uncontrolled VBR video traffics may not be accurate for the controlled VBR case, for example, see the proposed model in [20]. Moreover, in many studied applications the transmission channel is shared between encoded video packets (in IP networks) or smaller cells (e.g., in ATM networks) but due to the time-sliced transmission scheme in DVB-H application, the transmission channel is shared between the bursts that each includes a large number of video packets. Therefore, the StatMux performance in DVB-H depends on the statistics of the merged video packets or bursts that are different from the statistics of individual video packets and cells. From the channel point of view, the bandwidth efficiency depends on the number of multiplexed services. In DVB-H application a DVB-T channel with relative small bandwidth is allocated to a small number of services (typically 10 to 15) while in other applications the shared bandwidth and the number of services can be much higher. Moreover, in many studied applications the number of active services may vary during time, and significant past research efforts have focused on admission control for connecting new requested services. However, in DVB-H the number of broadcast services can be fixed for a longer time and a bandwidth for the channel is guaranteed. From the QoS point of view, not only the data loss and multiplexing delay should be controlled in DVB-H such as other applications, but also the effect of StatMux on the bottleneck of channel changing delay that exists in DVB-H should be considered [5]. Moreover, concerning the conjunction of time slicing and StatMux, the power consumption of DVB-H receiver is a serious constraint that can be affected by StatMux. Finally, from the implementation point of view, due to the time slice signaling in DVB-H, the TDM information should be known few seconds earlier, which is a hard to achieve.

Statistical multiplexing in DVB-H can be implemented by the IP encapsulator. Multiplexed services may share the bits in one time slice or they may share the time via separate time slices [3]. When the multiplexed services share the bits in one time slice, a number of services can be encapsulated to one MPE-FEC and one burst. When the multiplexed services share the time, as a simple case each service can be encapsulated to one MPE-FEC and one time slice. In this case, the transmission channel is shared between the time slices as TDM. The proposed method in this paper is directed to the statistical multiplexing case in which each service is encapsulated to one time slice.

2.1. Objectives of StatMux in DVB-H. Due to the essential differences that exist between DVB-H and other video

communication applications, the objectives of StatMux in DVB-H may differ from those of other applications. The objective of StatMux in many applications is to increase the number of services for a given bandwidth while the QoS is kept above a given threshold. In StatMux of VBR traffics, the efficiency of transmission bandwidth depends on the number of multiplexed services and also on the variations that exist in the bit rate of bit streams. The bandwidth efficiency increases when the number of services increases and when the variations in the bit rate decrease. Note that the gain of StatMux differs from bandwidth efficiency. For example, multiplexing few services with high variations in bit rate may provide a high gain for StatMux but still a low efficiency for the bandwidth. In DVB-H application in which a DVB-T channel is allocated to a small number of DVB-H services, using uncontrolled VBR bit streams with high variations in the bit rate cannot provide a high efficiency for the transmission channel. Moreover, due to the bottleneck of channel changing delay that has been enlarged by the time-sliced transmission scheme in DVB-H, and also due to buffering constraints that exist in DVB-H standard on a hypothetical receiver buffering model, it is preferred to use controlled VBR service bit streams instead of uncontrolled VBR bit streams [5, 34–36]. Generally, a controlled VBR bit stream encoded with a buffering constraint can be transmitted through a constant bandwidth channel with a limited delay and without any loss in transmission bandwidth. Therefore, in DVB-H application, if all the broadcast services are constrained to a buffering limit, even without utilizing StatMux and just by a deterministic multiplexing (DetMux), it is possible to design a network with a limited guaranteed end-to-end delay while the maximum bandwidth is used.

Generally delay and bandwidth can be considered as two resources in communication networks that compensate each other. For example a lower end-to-end delay in DVB-H with DetMux can be achieved at the expense of a higher bandwidth that means a lower efficiency for the transmission channel. Therefore, StatMux method not only can decrease the end-to-end delay but it can also improve the bandwidth efficiency. To evaluate the performance of StatMux, as a fixed reference operating point, it can be compared with DetMux in terms of end-to-end delay when the bandwidth efficiency is maximized. However, they can also be compared at any other operating point in terms of delay and bandwidth efficiency.

Channel changing delay in DVB-H, that is, required time for switching from one audiovisual service to another, is part of the end-to-end delay that is perceived by the user. Channel changing delay in DVB-H has been increased due to the time-sliced transmission scheme. Several factors contribute to the delay, and the major ones include time-slicing parameters, frequency of random access points, and variations in the bit rate of video bit stream. An initial buffering delay is required to compensate for the variations in the bit rate. StatMux can decrease the channel changing delay by minimizing the required initial buffering delay.

2.2. Challenges of StatMux in DVB-H. According to the time-sliced transmission scheme used in DVB-H, during transmission of a data burst, a *delta-t* or the time to the beginning of the next burst of the same service is signaled to the receiver in order to indicate to the receiver when to expect the next burst. In DetMux the whole bandwidth may be allocated to a number of services with fixed burst sizes and determined delta-*t*. Unlike DetMux, in StatMux, the burst sizes and the duration of time slices may vary over time according to the temporal bit rate of service bit streams. The problem that arises due to the variation over time of the duration of time slices is how to calculate the delta-*t* for each service. When the data for the current burst is encapsulated, the time-slice boundaries of the next burst of the same service are unknown and therefore, it is difficult to compute an exact delta-*t* to be signaled. While a typical time cycle can be about few seconds, even the estimation of the time-slice boundaries according to the variations in bit rate is difficult. Furthermore, any estimation error may lead to even worse results for StatMux compared to the DetMux case. It is possible to compute an exact delta-*t* by a long-time look ahead or by buffering of service data for a relatively long-time (typically several seconds) before data encapsulation. However, a long-time buffering imposes a long delay to the system that is in contradiction with the objectives of StatMux. Comparing to DetMux, the overall delay of such a system increases in the order of seconds without any gain in bandwidth.

StatMux may be implemented in the DVB-H standard in such a way that a number of services are multiplexed and encapsulated into one time slice [3]. To consume any multiplexed service, a receiver must receive the whole time slice or burst. Therefore, the power consumption of the receiver increases proportionally to the duration of the time slice or the number of multiplexed services. As a simple case when all services are multiplexed into one time slice that consumes the whole bandwidth, a DVB-H receiver should be switched on continuously. In this case the percentage of power saving resulting from time-slicing decreases from typical values of 80%–90% to zero. In this paper we are seeking a novel method to implement StatMux without such a large penalty in power consumption of a DVB-H receiver.

2.3. Burst Statistics. To reach the stated objective, we first investigate the properties of video traffics. VBR video traffics generally exhibit self-similar properties [7, 8]. The main feature of self-similar processes is that they exhibit long range dependence (LRD), that is, their autocorrelation function $r(k)$ decays less than exponentially fast, and is nonsummable, that is, $r(k) \sim k^{-\beta}$, as $k \rightarrow \infty$, for $0 < \beta \leq 1$. The quantity $H = 1 - \beta/2$ is called *Hurst parameter* or *Hurst exponent*. The Hurst exponent was originally developed in hydrology [37]. It shows whether the data is a purely random walk or has underlying trends. The Hurst exponent is related to the fractal dimension, and it is a measure of the smoothness of fractal time series. However, the statistical properties of controlled VBR video traffics differ from those of uncontrolled VBR traffics [12, 38–40]. The uncontrolled

VBR video traffics are persistent, that is, the samples are positively correlated and the Hurst exponent $H > 0.5$. On the other hand, the controlled VBR traffics are antipersistent, that is, the samples are negatively correlated and the Hurst exponent $H < 0.5$. However, a loose VBR rate controller with a relatively large buffer size may generate video traffics close to the middle range of the Hurst exponent, that is, $H = 0.5$. From the self-similarity, both controlled and uncontrolled VBR video traffics exhibit self-similarity with LRD. From the communication network point of view, persistent (LRD with positive correlation) traffics need more resources than uncorrelated traffics in terms of bandwidth and delay. On the other hand, less resources are required for transmitting antipersistent (LDR and negative correlation) traffics than uncorrelated traffics.

In an ideal case for StatMux over DVB-H, if only the audiovisual services are considered, the overall delay is minimized if all time slices or bursts would carry the amount of service data that corresponds to a fixed play-out period. Accordingly, an *Ideal Burst* is defined such that it carries the amount of service data corresponding to play-out in a fixed time cycle. For example a typical ideal burst may carry 90 video frames of a video service with a frame rate of 30 f/s that is corresponding to a play-out duration of 3 seconds. Due to time slicing and data encapsulating scheme in DVB-H, the statistics related to the size of the ideal burst are more relevant to StatMux process than the statistics related to individual video frames. When controlled VBR video traffics are used in DVB-H, the size of video frames and also the size of the ideal burst are antipersistent. The video rate controller uses a smoothing buffer with a size in the standard range, typically corresponding to few (less than 3) seconds buffering period. A typical ideal burst in DVB-H includes at least one intraprediction picture as a random access point and a number of interprediction pictures corresponding to a play-out period of one to few seconds (typically 3). Considering these typical figures, a small variance for the size of ideal bursts is expected. However, the variations in the video frame size can still be large in comparison to constant bit rate video bit streams. Considering a constant bit rate for audio data, only the video statistics are considered for development of the StatMux algorithm in this paper. However, the developed method can be used without any changes for VBR audio.

To evaluate the challenge of StatMux based on real video traffics, some statistics related to the size of an ideal burst on a number of 5 video bit streams including different contents encoded with a buffering constraint (buffering period of 2 seconds) for a bit rate of 300 kb/s, a frame rate of 15 f/s, and QVGA picture format were collected. Moreover, the video frame size statistics were collected. The Nokia H.264/AVC codec with the introduced VBR rate controller in [41] was used for encoding the video contents. The sample histograms of the video frame size and the ideal burst size are depicted in Figures 1 and 2, respectively. The collected statistics show that unlike the video frame size, the ideal burst size has a relatively smaller variance and its probability density function (PDF) is approximately normal. Figure 3 compares the distribution of the ideal burst size against the normal distribution. Note that the size of video frames can have
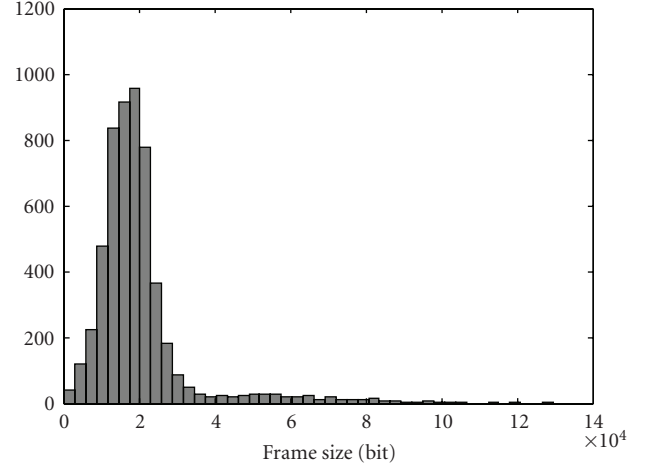

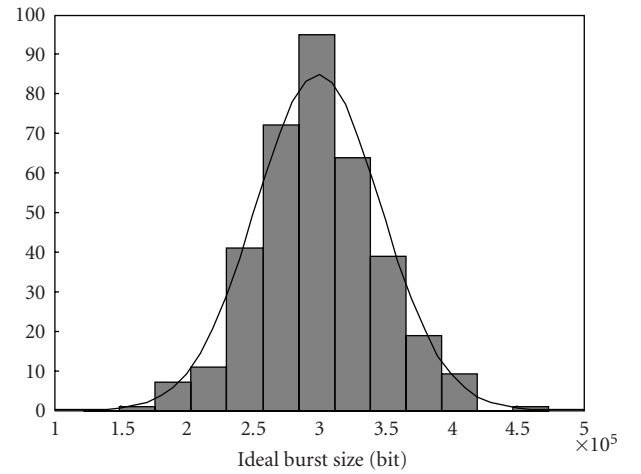FIGURE 1: Histogram of video frame sizes.

FIGURE 2: Histogram of ideal burst size.

any distribution but when a fixed number of video frames are considered as a burst, the distribution of the ideal burst size depends on the buffer constraint and the rate controller that drives the long-term average bit rate toward a target bit rate. For example, if an ideal burst contains one GOP and the rate controller tries to distribute a bit budget among the GOPs equally, then the size of resulting ideal burst can have a distribution very different from the distribution of frame size. Therefore, a normal distribution for the ideal burst size is possible even when the frame size has a distribution far from normal. However, in practice, the exact shape of the distribution depends on the rate control algorithm, the size of smoothing buffer used for the rate control, and the average burst size. Assuming a normal distribution for the ideal burst size and considering μ and σ as mean and standard deviation for the normal PDF, the collected statistics yield $\sigma/\mu = 0.15$. For this typical value, consider the following probability:

$$P(\mu - 2\sigma < \text{Ideal Burst Size} < \mu + 2\sigma)$$
$$= P(0.7\mu < \text{Ideal Burst Size} < 1.3\mu) \approx 0.95. \tag{1}$$

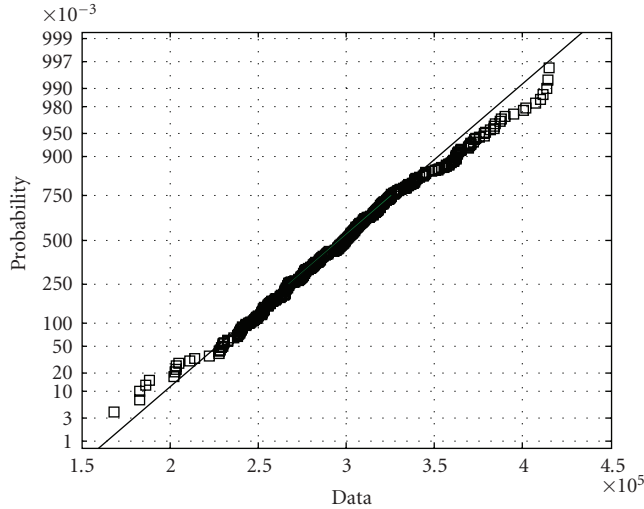FIGURE 3: Normal Probability Plot, Comparing Normal PDF with the PDF of the Ideal Burst size.

FIGURE 4: Example of timing diagram for the proposed time slicing and StatMux method.

The last equation comes from the fact that the ideal burst size is approximately normal. Consequently, if the time slice durations could have small variations (here within ±30% of the average ideal burst size), then with a high probability (approximately 0.95), the range of the variation would be enough to carry the ideal bursts. This means that a StatMux is close to the ideal case in which all bursts are ideal bursts. According to these results time slicing and StatMux techniques are proposed in which the delta-t calculation problem is solved with no look ahead and no special buffering. The next section presents the details of the proposed StatMux and time-slicing techniques.

3. Proposed Statistical Multiplexing Method

A new StatMux method for IPDC over DVB-H application is proposed for the case of a relative small variance of the ideal burst size. According to the proposed method, flexible burst duration and boundaries, within an acceptable range, allow statistical multiplexing to be performed in conjunction with time slicing in a DVB-H network by an IP encapsulator. Similar to DetMux, the proposed method splits the transmission time into several time cycles and allocates, for each service, a time slot from the total time cycle according to the average bit rate of the service. However, the boundaries of each time slice are allowed to vary in such a way that the time slice duration can grow or shrink in a limited range. This allows for allocating variable size data to the services. The signaling of the delta-t is performed according to the earliest allowed time such that service data is not missed. Sometimes, the receiver switches on but there is no service data ready to be received yet. However, in this case, data from the previous service might be received and discarded or used for another purpose.

3.1. Definitions and Conditions. According to the proposed method, the IP encapsulator receives IP packets that belong
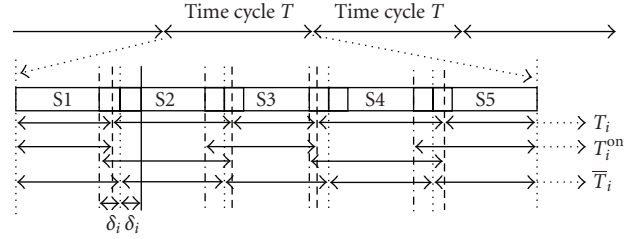
to a number of services which are going to be encapsulated and transported. A number of services, as well as an average bit rate of each service, can be planned such that the available bandwidth is substantially completely used. A significant part of the transmission channel bandwidth is typically used for MPE-FEC code, MPE protocol overhead, transmission stream (TS) protocol overhead, and the program specific information/service information (PSI/SI) signaling. A relatively small percentage of each MPE-FEC frame may be wasted because the IP packets do not typically fit exactly into the *application data table* of the MPE-FEC frame. The remaining bandwidth is available to be allocated to IP packets of the DVB-H services.

A fixed time cycle (T) is determined for all services based on the channel bandwidth, average bit rate of services, and the desired percentage of power saving for the receivers. Techniques for determining the time cycle are well known and are explained in more detail in the ETSI standard document; see [3–5] for more details. According to the proposed method, for each service, an *Average_Time_Slice_Duration* is computed as

$$\overline{T}_n = \frac{R_n T}{R_1 + R_2 + \cdots + R_n + \cdots + R_N}, \qquad (2)$$

where $\overline{T}_n$ denotes the *Average_Time_Slice_Duration* of service n. R_n represents the average bit rate of service n, and N stands for the number of services. As shown in Figure 4, the time line during transmission is partitioned into fixed time cycles and a fixed order for the services in the time cycle is determined. The burst durations are denoted by T_i in the figure. T_i^{on} shows the time duration in which the receiver of the ith service should be switched on. $S1, \ldots, S5$ stand for 5 multiplexed services. More details about the service ordering are presented later in Sections 3.4 and 3.5.

According to the proposed technique, the duration of time slices can have some variations around the average values to allow the use of StatMux. The maximum variations in time-slice durations are controlled by a set of numbers $\{\delta_i\}$, $i = 1, \ldots, N - 1$, called *Delta Burst Duration* set. Except for the first and last services which are started and ended, respectively, with the time cycle, the other services can start and end earlier or later than the time instants that correspond to DetMux. The time instants corresponding to DetMux are determined based on $\overline{T}_i$ boundaries.

To define the delta-t properly, a set of conditions is imposed on the time-slice durations as follows:

$$-\delta_n + \sum_{i=1}^{n} \overline{T}_i \leq \sum_{i=1}^{n} T_i \leq \delta_n + \sum_{i=1}^{n} \overline{T}_i, \quad n = 1, \ldots, N \quad (3)$$

Note that as shown in Figure 4, for the last service $\delta_N = 0$ maximizes the bandwidth usage by fixing the end of last time slice to the end of time cycle. However, a nonzero δ_N is possible at the expense of a small loss in bandwidth. More related details about δ_N are presented in Section 4.

According to conditions (3), the range of variations of the time-slice durations in each time cycle can be define by a minimum ($T_i^{\min}$) and a maximum ($T_i^{\max}$) value as

$$T_i^{\min} \leq T_i \leq T_i^{\max}, \quad (4)$$

where the minimum and the maximum values are defined as below. For the first service

$$\begin{aligned} T_1^{\min} &= \overline{T}_1 - \delta_1, \\ T_1^{\max} &= \overline{T}_1 + \delta_1. \end{aligned} \quad (5)$$

For the other services

$$\begin{aligned} T_n^{\min} &= \sum_{i=1}^{n} \overline{T}_i - \sum_{i=1}^{n-1} T_i - \delta_n, \\ T_n^{\max} &= \sum_{i=1}^{n} \overline{T}_i - \sum_{i=1}^{n-1} T_i + \delta_n. \end{aligned} \quad (6)$$

Note that for the last service, if $\delta_N = 0$, then $T_N^{\min} = T_N^{\max}$.

3.2. Delta-t Calculation. According to the imposed conditions above, the time-slice boundaries can be located in limited ranges over the time line. The delta-t values are computed based on the earliest data that can be received to prevent missing data at the receiver. Therefore, for the first service in the time cycle, the delta-t is computed as

$$\Delta t_1^j = T - T_{\text{pass}}^j, \quad (7)$$

and the delta-t of other services is computed as

$$\Delta t_n^j = T - T_{\text{pass}}^j - \delta_{n-1} + \sum_{i=1}^{n-1} \overline{T}_i, \quad n = 2, \ldots, N, \quad (8)$$

where Δt_n^j denotes the delta-t signaled by the jth MPE-FEC section of service n, and T_{pass}^j represents the past time from the start of the current time cycle when the MPE-FEC section is transmitted.

According to this signaling technique there are some short time intervals in which the radio receiver is active but there is no data to be received. This increases the receiver power consumption. Analytical and experimental results show that the overall increase in power consumption is very small (few percent), and it is the only cost that is paid for the proposed StatMux method.

3.3. StatMux and IP Encapsulation. A solution for the delta-t calculation problem was presented. Now a multiplexing algorithm is needed to dynamically distribute the available bandwidth among the multiplexed services proportionally to their bit rates. If only the audiovisual services are considered, in the ideal case of StatMux, the delay is minimized if all bursts carry the exact amount of data that corresponds to a fixed play-out period. However, such an ideal StatMux is impossible with maximum bandwidth usage. In the proposed StatMux, the attempt is to adapt StatMux as close as possible to the ideal case. The proposed StatMux algorithm is implemented in conjunction with the IP encapsulation according to the following algorithm.

(1) The IP packets received by the IP encapsulator are stored in a small size buffer or in a number of N separate buffers corresponding to the services. When separate buffers are used the size of each buffer is approximately equal to the maximum burst size.

(2) The IP packets related to service n are fetched by the IP encapsulator to be encapsulated and transmitted in time slice in accordance with the following rules.

(a) A *Target_Time_Stamp* is defined for media bit streams in the current time slice based on the time cycle T and the previous *Target_Time_Stamp* (in the previous time cycle) such that the media packets are synchronous to the time cycles and also to each other. For each media bit stream, a fixed value is added to the previous *Target_Time_Stamp* to compute the current *Target_Time_Stamp*. The fixed value depends on time cycle T.

(b) The IP packets related to service n are fetched to the encapsulator in time stamp order to reach the *Target_Time_Stamp* while the condition $T_n^{\min} \leq T_n \leq T_n^{\max}$ is met.

(c) If the fetched IP packets fill the time slice to $T_n^{\max}$ and the time stamp of the last fetched packets is close to the *Target_Time_Stamp*, packet fetching is stopped before reaching the *Target_Time_Stamp*. A time stamp is considered to be close to the *Target_Time_Stamp* if

$$Target\_Time\_Stamp - \text{time-stamp} < \triangle, \quad (9)$$

where $\triangle$ is a constant that is proportional to the media frame interval.

(d) If the fetched IP packets fill the time slice to $T_n^{\max}$ and the time-stamp of the last fetched packets is significantly lower than the *Target_Time_Stamp*, concerning random access points, a number of packets with older time-stamps can be dropped and more packets with newer time stamps can be fetched. A time stamp is considered significantly lower than the *Target_Time_Stamp* if

$$Target\_Time\_Stamp - \text{time-stamp} > c \cdot \triangle, \quad (10)$$

where c is a constant and $\triangle$ is a constant proportional to the media frame interval.

(e) If the *Target_Time_Stamp* is reached but the fetched IP packets do not fill the time slice to $T_n^{\min}$, then more packets are fetched to fill up the time slice up to at least $T_n^{\min}$.

(f) If the fetched IP packets do not fill the time slice to $T_n^{\min}$ and there are no more packets in the buffer, padding is used and the min time-slice duration $T_n^{\min}$ is used as the end of the current time slice.

(3) The fetched packets are encapsulated and transmitted.

(4) Go To Next Service.

3.4. Service Order Effect. The described StatMux algorithm permits some variations in the temporal allocated bandwidth to the DVB-H services proportional to their bit rates. The permitted variations and also the performance of StatMux depend on the order of services within the time cycle as well as on the values of δ_n. Some simulation results are presented in Section 4. The range of permitted variations for service n may be controlled by δ_n. However, due to the interaction that exists between time-slice boundaries, the range of permitted variations for a service is affected by all previous services in the time cycle. In other words, the performance of StatMux on service n depends on all δ_i, $i \leq n$, while it is mainly controlled by δ_n. A larger δ_n provides more flexibility in allocating the bandwidth. As a special case if $R_1 = \cdots = R_N$ and $\delta_1 = \cdots = \delta_N$, the earlier services in the time cycle have more flexibility in the bandwidth than later services. In this case, StatMux has a higher performance on the earlier services in multiplexing order. This means that the order of services in the time cycle establishes a sort of prioritization for the service multiplexing. This prioritization can be compensated by using proper values for δ_n. A method for computing δ_n is presented in the sequel.

3.5. Compensation of Service Order Effect. Experimental results show that when a similar flexibility for the burst durations of multiplexed services is used, the performance of StatMux decreases almost exponentially according to the order of services in the time cycle. The idea is to increase the values of δ_n exponentially according to the service order to provide a similar performance for StatMux over all services. Considering a number of N services, with service index $1, \ldots, n, \ldots, N$, which are multiplexed. To compensate the effect of service order on StatMux performance, a compensation function is extracted experimentally as follows:

$$\alpha_n = 2.10 - 0.28 e^{-(n-1)/2.1} - 1.35 e^{-(n-1)/18.0},$$
$$n = 1, 2, \ldots, N, \tag{11}$$

where α_n is a compensation coefficient for δ_n of service n. The service order compensation function is depicted in Figure 5. As an example, numerical values for the first 6 services are

$$\{0.47, 0.65, 0.78, 0.89, 0.98, 1.05\}. \tag{12}$$

The service order compensation function has been provided based on a heuristic optimization on a large number of
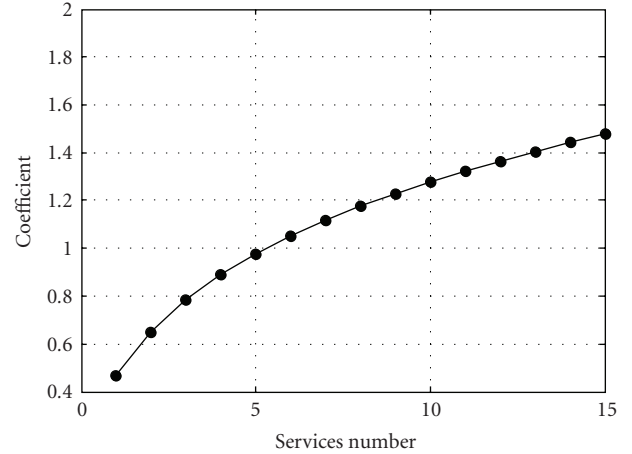

FIGURE 5: Service order effect compensation function.

different video bit streams that are multiplexed with different multiplexing parameters. As reported in Section 4, simulation results show that computed compensation coefficients by provided function perform well for a wide range of different bit streams multiplexing parameters over a large number of services.

When the bit streams have similar average bit rates, selecting values for δ_n proportional to α_n compensates the effect of service order. If the bit streams differ in bit rate, δ_n should be adapted to the bit rates. Therefore, normalizing the compensation coefficients with respect to their average value, delta burst duration δ_n can be computed as

$$\delta_n = \lambda \overline{T}_n \frac{\alpha_n}{\overline{\alpha}}, \qquad n = 1, 2, \ldots, N, \tag{13}$$

where $\overline{\alpha}$ denotes the average value of α_n over N services. λ is a constant coefficient that defines the flexibility of burst durations and also the overall performance of StatMux. Some simulation results related to the compensation of service order effect are presented in Section 4.

3.6. Heterogeneous Bit Streams. When multiplexed bit streams are encoded with a similar buffering constraint and the service order effect is compensated as above, a similar performance over all services is expected for the StatMux algorithm. In a general case in which the multiplexed services are heterogeneous and the bit streams are encoded with different buffering constraints, the performance of the proposed StatMux algorithm is not the same for all services. However, in this case, simulation results (presented in Section 4) show that the requirements of multiplexed services are close to the case in which the services are encoded with a similar buffering constraint.

3.7. Performance Criteria. The performance criteria for the proposed StatMux method should be defined properly with respect to the objectives of StatMux in a DVB-H application. The main objective of StatMux in a DVB-H application is to minimize the end-to-end delay of broadcast services while

the maximum available bandwidth is used and the data drop rate is limited.

The end-to-end delay in DVB-H networks depends mainly on the buffering delay that is required for several buffers in the system. A buffer at the IP encapsulator before encapsulation and another at the receiver before the decoder are two major buffers in the system. Forgetting the time-sliced transmission scheme, a continuous transmission channel between these two buffers can be assumed that is a constant bandwidth channel in DetMux case. In StatMux the channel has a VBR bandwidth for each service. In DetMux the encapsulator buffer compensates the difference between the VBR input and the constant bit rate output and also the decoder buffer compensates for the difference between the constant bit rate input and the VBR output. If jitter in the IP network is ignored, the size of the two buffers and their buffering period mainly depend on the variations that exist in the bit rate of the service bit streams. Therefore, the two buffers are expected to behave symmetrically. On the other hand, in StatMux the two buffers have variable bit rate input and output. While the variations in bandwidth are in accordance to the variations in bit rates, the difference between the input and the output is less than in the case of DetMux. As a result the buffer sizes and buffering delays can be smaller in StatMux than in DetMux. The other delay sources that exist in the system are common for both StatMux and DetMux. Therefore, the performance of StatMux can be evaluated by measuring the minimum required buffering delays for the IP encapsulator buffer and the decoder buffer. However, assuming symmetric behavior for the two buffers, the performance can be evaluated based on one of the two buffers. Regarding the buffering delay, the performance of the proposed StatMux is studied in two cases in the sequel.

3.8. Performance of StatMux on Minimum Buffering Delay for Zero Data Drop Rate. Regarding the receiver buffer, the minimum buffering delay means the minimum initial buffering period at the receiver buffer that is required before the start of the decoding for continuous play-out without buffer underflow and without data drop. Due to the symmetric operation of the receiver buffer and the IP encapsulator buffer, underflow at the receiver buffer corresponds to the overflow at the encapsulator buffer and both result in data drop. No underflow at the receiver buffer means a zero data drop rate. Note that, while the bit streams are encoded with buffering constraints, the receiver buffer can be prevented from overflow by a proper buffer size.

The performance of the proposed StatMux method in the zero drop rate case was evaluated experimentally. As expected, experimental results show that the proposed StatMux method decreases the min buffering delay in average over all multiplexed services. Using the compensation techniques presented by (13), the expected minimum buffering delay can be fixed for all services. However, in practice it is required to signal the exact value of the initial buffering period rather than the expected value to the receiver.

From the signaling point of view, the distribution of the minimum buffering delay around the average value is more important than the average value itself. The histograms of minimum buffering delay for the multiplexed services by StatMux were compared against those by DetMux to evaluate the performance of the proposed StatMux from this point of view. Details of the simulation results are presented in Section 4. The simulation results show that in StatMux, a lower initial buffering period compared to DetMux can be used even for a zero data drop rate. However, to use the advantage of StatMux more efficiently it is proposed to accept a limited data drop rate to achieve less end-to-end delay. Some quantified results are presented in Section 4.

3.9. Performance of StatMux on Buffering Delay with Data Drop. To use the advantage of the proposed StatMux method more efficiently, it is possible to decrease the buffering delay at the expense of a limited data drop rate. In this case the initial buffering delay signal should be computed statistically in order to to optimize the buffering delay and the data drop rate. In practice a model is needed to predict the performance of StatMux regarding the buffering delay and data drop rate.

A performance model for the proposed StatMux method is analytically provided based on the effective bandwidth theory that verifies the experimental results. The effective bandwidth theory attempts to provide a measure of bandwidth and buffer size, which adequately represents the trade-off between sources of different types, taking proper account of their varying statistical characteristics and QoS requirements [24].

In general if a bursty traffic $X = \{x(n)\}$ is offered to a server with the buffer size of B and channel capacity of C, then the buffer will overflow after loading a number of p data packet to the buffer if $[x(1) + \cdots + x(p)] > (p.C + B)$ [42]. Therefore, if the user demands a loss probability less than ε then the channel capacity and the buffer size should be such that

$$\Pr[[x(1) + \cdots + x(p)] > (p \cdot C + B)] < \varepsilon, \quad \forall p. \quad (14)$$

A study of real video traffics shows that the ideal bursts have a PDF very close to a Gaussian while the PDF of video frames is more complex. To simplify the modeling task, the ideal burst is chosen as a packet data instead of video frames for modeling. When the traffics X is an i.i.d. (independent and identically distributed) Gaussian random process with mean μ and variance σ^2, from the large deviation theory, the required channel capacity or effective bandwidth C is given by [18, 24]

$$C = \mu + \frac{\sigma^2}{2}\delta, \quad (15)$$

where

$$\delta = \left\lceil \frac{\ln(\gamma) - \ln(\varepsilon)}{B} \right\rceil, \quad 0 < \gamma \le 1, \quad (16)$$

γ is considered as a constant value. A more accurate form of this formula can be expressed as

$$\varepsilon = (2\pi)^{-1/2}(1+y)^{-1} \exp\left(-\frac{y^2}{2}\right), \qquad (17)$$

where

$$y = 2\sigma^{-1}(C-\mu)^{1/2}B^{1/2} \qquad (18)$$

is an approximation of the residual distribution function of the standard Gaussian distribution.

To use the results of effective bandwidth theory for the proposed StatMux method a number of simplifications are assumed as below

(a) While the performance of StatMux is equalized over all multiplexed services, a similar effective bandwidth is assumed for all multiplexed services.

(b) While the effective bandwidth for all multiplexed service is similar, one service in the multiplexing order is used for modeling.

(c) The effective bandwidth is assumed to be the average bandwidth plus a part that is proportional to the overall flexibility of bandwidth that is, $C = \mu + \beta\lambda\mu$, where β is the proportionality coefficient.

(d) The IP encapsulator buffer is assumed to operate similarly to a server buffer.

(e) As explained earlier, in the ideal case the decoder buffer at the receiver and the IP encapsulator operate symmetrically such that overflow at the IP encapsulator buffer corresponds to under flow at the decoder buffer. Note that in practice, a minimum buffering period is signaled to the receiver to prevent under flow. Due to symmetric operation of the buffers, the buffer size at the IP encapsulator corresponds to the fullness of the decoder buffer after the initial buffering delay, that is, $B = \mu D$, where D stands for the initial buffering delay.

Using the assumptions above, the results of large deviation theory can be rewritten for the burst data packets as follows:

$$\varepsilon = (2\pi)^{-1/2}(1+y)^{-1} \exp\left(-\frac{y^2}{2}\right), \qquad (19)$$

$$y = 2\mu\sigma^{-1}(\beta\lambda D)^{1/2}, \qquad (20)$$

where μ and σ^2 are the mean and variance of the ideal burst size. The value of β is related to the proposed time-slicing process, and it is assumed independent of the bit stream properties. This value can be found experimentally once for an operating range and then it can be used forever. Note that γ in (16) is the approximated value of $(2\pi)^{-1/2}(1+y)^{-1}$ in (19) that is sometimes set to 1 [43]. However, due to small σ in this application, the value of γ is closer to zero than 1. The provided model presents the drop rate based on the ideal burst packets. In fact, the model computes the

lower bound for the Fractional Brownian Motion (FBM) model that corresponds to the Hurst exponent $H = 0.5$. For the controlled encoded bit streams in which $H < 0.5$, the proposed model may have a slight over estimation of drop rate [20]. However, the results presented in Section 4 show that the model performs accurately enough to be used.

3.10. StatMux and Receiver Power Consumption. Unlike DetMux in which delta-t of a multiplexed service can be computed based on the exact determined time-slice boundaries, in the proposed StatMux method, delta-t is computed based on the earliest possible received data to prevent missing data at the receiver. Therefore, there are short time periods in which the radio receiver is active but there is no service data to be received. This increases the power consumption of the DVB-H receiver and consequently decreases the battery life time.

As shown in Figure 4, the time slice of the first service always started with the time cycle that is known and delta-t computed for the first service is a deterministic parameter. As a result there is no increase in power consumption of the receiver for the first service. For the other services, the increase in the power consumption depends on the values of δ_n. In average over all time slices, the radio receiver of service n is active for time duration equal to δ_{n-1} more than in the DetMux case. In DVB-H, the performance of time-sliced transmission scheme is evaluated by the percentage of power saving that is defined for the radio reception parts of DVB-H receiver as [3]

$$PPS = 100\left(1 - \frac{T_{\text{on}}}{T}\right) = 100\left(1 - \frac{T_S + T_R}{T}\right), \qquad (21)$$

where PPS denotes the percentage of power saving for the radio reception parts. T_{on} represents the time duration in which the radio receiver is active. The radio receiver is switched on for a synchronization time (T_S) before service data reception period (T_R). When the proposed StatMux technique is used, the percentage of power saving for the receiver of service n can be computed as

$$PPS_n = 100\left(1 - \frac{T_S + T_R + \delta_{n-1}}{T}\right), \quad n = 2,\dots,N. \quad (22)$$

Therefore, the reduction in the percentage of power saving for StatMux can be expressed as

$$\Delta PPS_n = PPS - PPS_n = 100\frac{\delta_{n-1}}{T}. \qquad (23)$$

To provide a numerical example consider a simple case in which $R_1 = \cdots = R_N$. In this case,

$$\Delta PPS_n = 100\frac{\delta_{n-1}}{T} = 100\frac{\overline{T}_n}{T}\frac{\delta_{n-1}}{\overline{T}_n} = 100\frac{1}{N}\frac{\delta_{n-1}}{\overline{T}_n}. \qquad (24)$$

For example, if $N = 10$ and $\delta_{n-1}/\overline{T}_n = 0.30$ then $\Delta PPS_n = 3\%$. Such a reduction for a typical system that operates with PPS of about 80% to 90% is a small cost for the proposed statmux method. note that the consumed power by the radio reception parts is only a part of the whole consumed power by the receiver. Therefore, the overall increase in the receiver power consumption is very small.

4. Simulation Results

During the design and development of the proposed StatMux algorithm several simulations were run. For each simulation a set of service bit streams is to be multiplexed. Furthermore, to collect statistical results, each simulation was repeated many times with different service bit streams.

The service bit streams were generated by a model that is targeted for controlled VBR video traffic [12, 39]. According to the model a Gamma distribution is considered for each video frame type (i.e., I, P, B) in each video scene. The model can generate video traffics according to some descriptive properties of video content such as motion activities and also according to the encoding parameters. The model was parameterized based on some statistics collected from real video traffic targeted for DVB-H application [12, 39]. The model was tuned to generate video traffics corresponding to bit streams with a bit rate of 300 kb/s, frame rate of 15 f/s, QVGA picture format, and different contents.

To evaluate the performance of the proposed StatMux method for different numbers of multiplexed services, a set of simulations was run separately for 4, 5, 6, 8, and 10 services. In each simulation, the results of StatMux were compared against DetMux in terms of minimum required buffering delay for zero drop rate conditions when the bandwidth usage is maximized. Simulations were repeated 300 times for different bit streams with similar statistical properties. In these simulations, the same δ_n was used for all services, that is, $\delta_n = 0.25\overline{T}_n$. The minimum buffering delay for multiplexed bit streams in both DetMux and StatMux cases was measured. The ratios of buffering delays in StatMux to buffering delays in DetMux in average over all 300 repetitions are depicted for different number of services (NOSs) in Figure 6. As the graphs show, the delay reduction (StatMux buffering delay to DetMux buffering delay) result of the StatMux depends on the service index in the multiplexing order. While in DetMux all services need a similar buffering delay, the required buffering delay for the earlier services is smaller than in later services in StatMux. According to the average results, all multiplexed services, except the last service, need considerably lower (less than 50%) buffering delays in StatMux compared to DetMux. For the last service the required buffering delay increased in StatMux because there is no flexibility at the end of the last time slice to use the whole bandwidth. Moreover, the flexibility at the start of the last time slice that is controlled in favor of the previous service may be in contradiction with the variations in the bit rate of the last service. In practice the last time slice can be allocated to non-time-critical services that exist in DVB-H. Alternatively, it is possible to allocate a δ_N to the last service as well as the other services to provide a similar multiplexing performance for all services. This costs a small extra bandwidth. For example, if the number of services is 10 and $\delta_N = 0.25\overline{T}_n$, then the extra bandwidth is about 2.5% of whole bandwidth, which is a very small cost. Additional simulations reveal that when a δ_N is allocated to the last service, depending on the variations in the bit streams and also depending on the δ_N, the required buffering delay for the last service can be lower, higher or equal to the delay of the
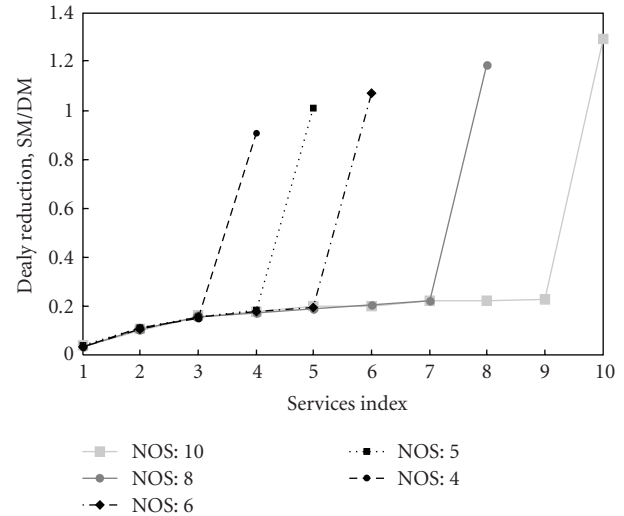

FIGURE 6: Delay reduction (StatMux delay to DetMux delay) for different number of multiplexed services (NOS).
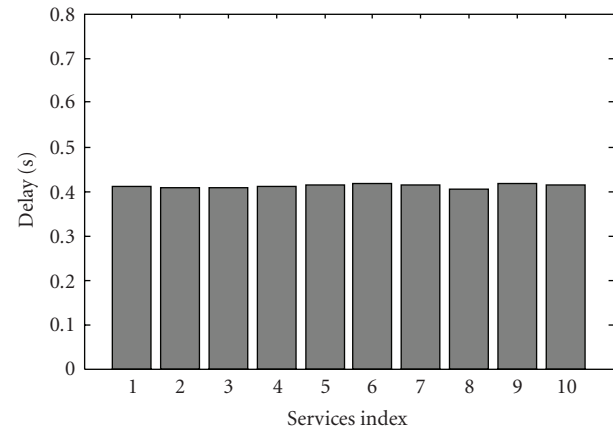

FIGURE 7: Buffering delay for 10 multiplexed services DetMux.

previous service. For simplicity in the rest of the simulations a δ_N is allocated to the last service only.

To study the effect of the service order on the StatMux performance more specifically, a simulation was run as above but only for 10 services and $\delta_n = 0.35\overline{T}_n$. The simulation was repeated 500 times and for each service, the average buffering delay over the 500 repetitions was measured in both StatMux and DetMux cases. The average buffering delays for DetMux and StatMux are depicted in Figures 7 and 8, respectively. These results show that while in DetMux case, the multiplexed services have similar buffering delays, in StatMux the delay increases with the service index. However, StatMux provides a considerable reduction in the delay of all services. Based on these results, a method was proposed in Section 3.5 to compensate the effect of service order and to provide similar buffering delays for all services in StatMux. A compensation function was derived by a heuristic optimization based on results of a large set of simulations that were run for different bit streams and different multiplexing parameters.
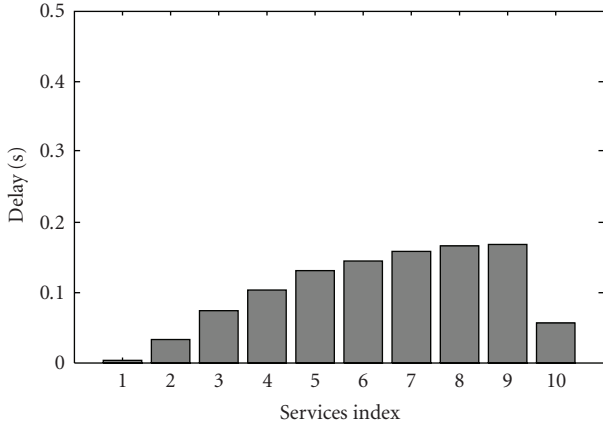
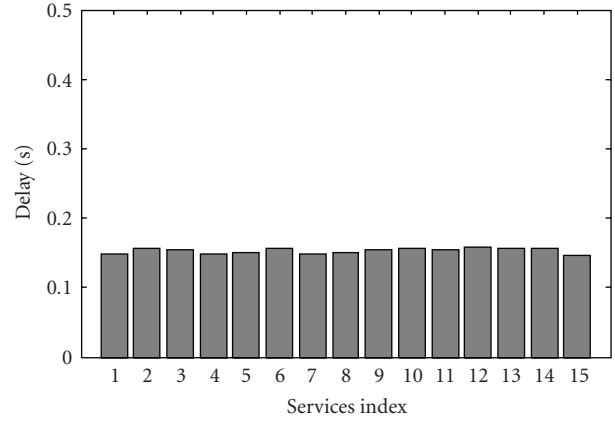FIGURE 8: Buffering delay in StatMux for the bit streams used in Figure 7.

FIGURE 10: Buffering delay in StatMux with service order compensation for the bit streams used in Figure 9.
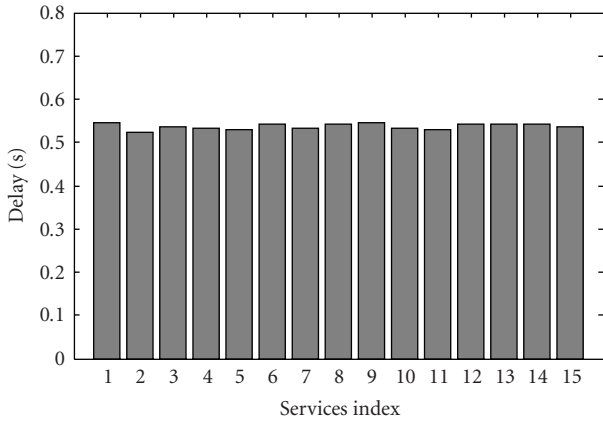

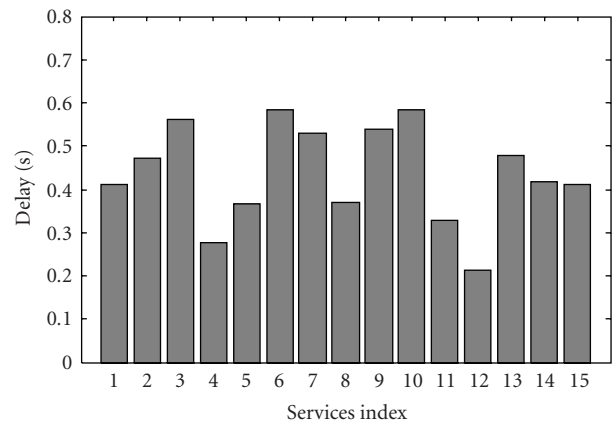FIGURE 9: Buffering delay for 15 multiplexed services in DetMux.

FIGURE 11: Buffering delay for 15 multiplexed services in DetMux, bit streams have different buffering constraints.

To show the performance of service order compensation, another set of simulations was run for a number of 15 services in which the values of δ_n for each service were computed according to the proposed method in Section 3.5 for $\lambda = 0.30$. The average buffering delays in 300 repetitions for DetMux and StatMux are presented in Figures 9 and 10, respectively. The results show that unequal values for δ_n computed by the proposed method provide similar performance over all services independent of the service index (i.e., order). Moreover, the results show that the overall buffering delay decreased from 0.54 seconds in DetMux to 0.15 seconds in StatMux. Further simulations using different bit streams and different values for λ show that the service order compensation method performs very well for a wide variety of possible bit streams. The bit streams used were different in term of required buffering delay in a range of (0.4–2) seconds, in term of σ/μ in the range of (0.08–0.16), and in terms of Hurst exponent in the range of (0.2–0.45). The ranges above are selected based on collected statistics from real video bit streams. Different values for λ were tested in the range of (0.20–0.40). A number of 15 services were multiplexed in the simulations that are close to the upper bound in DVB-H application.

It is worth to note that buffering delay is a major part of the channel changing delay in DVB-H, and therefore, a reduction of buffering delay due to StatMux also leads to a reduction of the channel changing delay.

In the previous simulations, the multiplexed services had similar statistical properties in terms of variations in bit rate or buffering constraint. To evaluate the performance of StatMux in a general case in which the bit streams have different statistical properties another set of simulations was run. A number of 15 bit streams were multiplexed which have different buffering constraints. The order of services was selected randomly without any preferences while the service order effect compensation method was used. Simulation was repeated 300 times. The average buffering delay over the 300 repetitions for DetMux and StatMux are depicted in Figures 11 and 12, respectively. As the results show, all services have a considerably lower delay in StatMux than in DetMux. Furthermore, the results show that unlike DetMux in which the required delays are very different, in StatMux the required delays are very close to each other. The services that have more variations in their bit rate utilize more reduction in
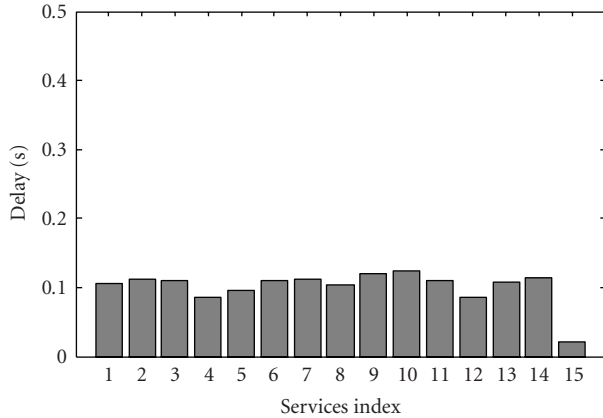
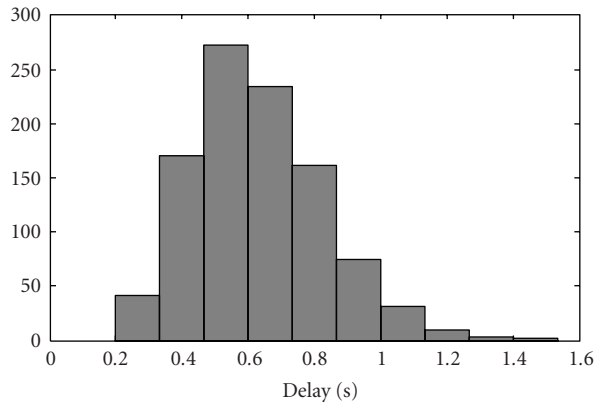FIGURE 12: Buffering delay in StatMux for the bit streams used in Figure 11.

FIGURE 14: Histogram of minimum buffering delay in StatMux for the service used in Figure 13.

FIGURE 13: Histogram of minimum buffering delay for one service in DetMux.
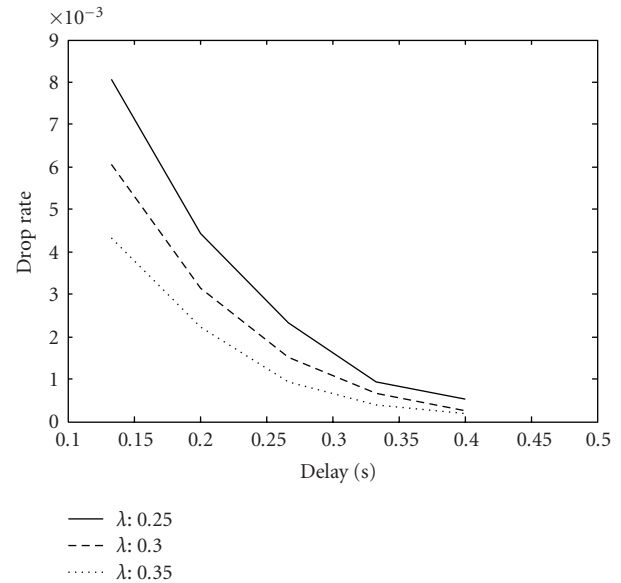
FIGURE 15: Data drop rate versus buffering delay for multiplexed services by 3 different values for λ.

delay such that after multiplexing they need almost similar buffering delays.

So far, the performance of StatMux was evaluated based on delay reduction in average over a number of simulations. However, in practice it is required to signal the exact value of the initial buffering period rather than the expected value to the receiver. From a signaling point of view, the distribution of minimum buffering delay around the average value is more important than the average value itself. To evaluate the performance of StatMux from this point of view, a set of simulations on various bit streams with different multiplexing parameters was run. Simulations were run on 8 services and repeated 1000 times and then the histograms of minimum buffering delays for the multiplexed services were compared between StatMux and DetMux. Two sample histograms for DetMux and StatMux are depicted in Figures 13 and 14, respectively. The histograms are related to the 4th service in multiplexing order while the other services have almost similar histograms. As shown in the figures, the histogram 5 of the buffering delay in StatMux is much narrower (i.e., a smaller variance than DetMux) than DetMux in such a way that signaling a buffering delay, which is a little higher than the average value, can provide
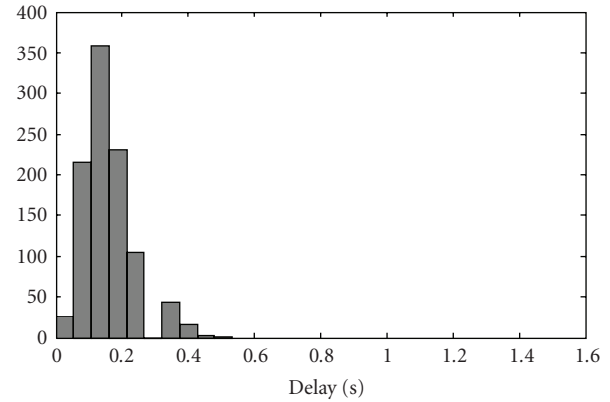
a perfect buffering with zero data drop rate. However, in practice it is possible to accept a limited drop rate to decrease the buffering delay.

To evaluate the performance of the proposed StatMux method in conjunction with data drop rate, another set of simulation was run to measure the data drop rate as a function of buffering delay. Figure 15 shows a sample simulation results in which the data drop rate is depicted as a function of buffering delay for 3 different values of λ, 0.25, 0.30, and 0.35. These results are provided for bit streams with $\sigma/\mu = 0.15$, and $H = 0.29$. Comparing to real video bit streams, these values correspond to bit streams with a high degree of variation in bit rate that is the worst case from the data drop rate point of view. As the results show, even for such bit streams the data drop rate is reasonably small for practically small buffering delays.
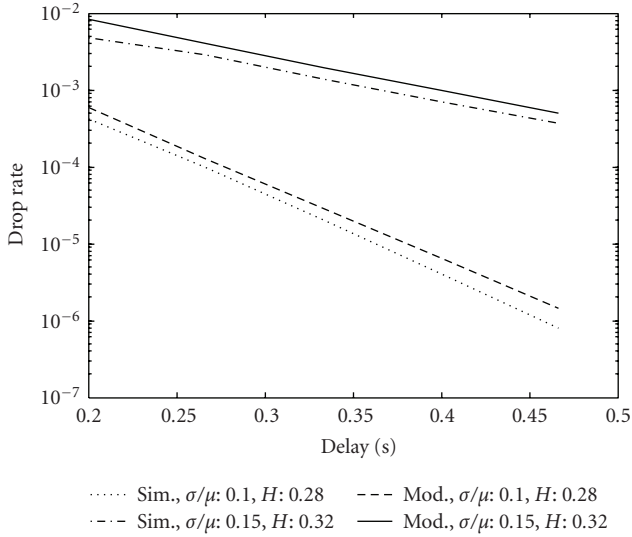
FIGURE 16: Data drop rate versus buffering delay estimated by the model (Mod) and measured by simulation (Sim) for 2 different groups of bit streams in terms of Hurst exponent and σ/μ.

FIGURE 18: Data drop rate versus buffering delay estimated by the model (Mod) and measured by simulation (Sim) for bit streams with different values for Hurst exponent.
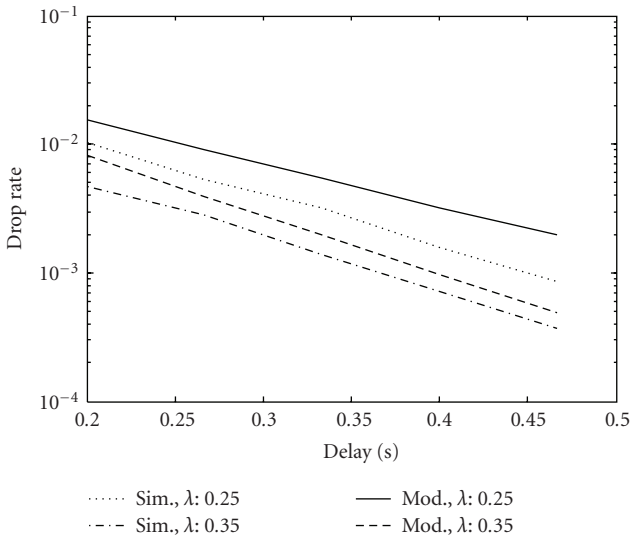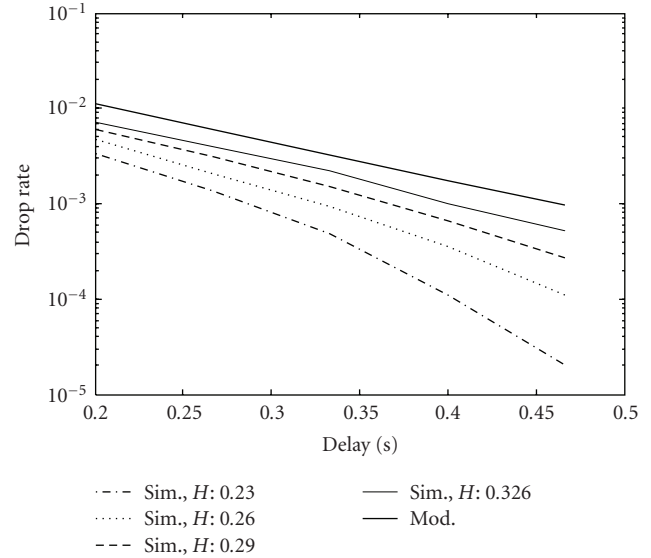

FIGURE 17: Data drop rate versus buffering delay estimated by the model (Mod) and measured by simulation (Sim) multiplexed by 2 different values for λ.

A performance model was provided for the proposed StatMux method in Section 3.9. The parameter β that defines the relationship between the effective bandwidth and the special constrained VBR bandwidth in the proposed StatMux method should be estimated experimentally. To estimate a value for β a set of simulations was run such that in each simulation the multiplexed bit streams have a different value for σ/μ. According to the simulation results a value of 0.6 for β provides a good accuracy for the performance model. Sample simulation results are presented in Figures 16–18. Figure 16 depicts the data drop rate as a function of buffering delay estimated by the model (Mod), and also

measured by simulations (Sim) for two groups of bit streams. The bit streams in each group have a different value for σ/μ (0.10, 0.15) and Hurst exponent ($H : 0.28$, $H : 0.32$). As shown in the plots, the model performs rather accurately. Figure 17 compares the data drop rate estimated by the model against the simulation results while the bit streams are multiplexed by two different values for λ, 0.25 and 0.35. In Figure 18 the multiplexed bit streams have a similar value of σ/μ but they have different values of the Hurst exponent. As expected the bit streams with a higher Hurst exponent have a higher drop rate. Although the Hurst exponent is not included as a parameter in the model, still the estimation error is reasonably small for practical values of the Hurst exponent. Note that the model is extracted based on the lower bound for the FBM model that corresponds to the Hurst exponent $H = 0.5$.

The relationship between StatMux and video quality can be considered from two points of views. First, when StatMux is used, the video bit streams can be encoded with a higher quality at the expense of more variations in the bit rate. Then, the delay resulting from the variations in the bit rate can be effectively reduced by StatMux. Sample simulation results assessing video quality with respect to delay are presented in [5]. Second, when StatMux is used, a higher performance is achieved at the expense of a small data drop rate. Please note that, according to the proposed multiplexing method, the dropped data are video frames that can be selected smartly based on the random access points by the IP encapsulator such that subsequent frames can be reconstructed perfectly without any degradation in quality. In other words, the measured data drop rate means the dropped frames before transmission which is different from data drop rate resulted by transmission error. According to the simulation results (e.g., Figure 18), the proposed method

provides a good performance with a data drop rate in the order of 10^{-3} which is much smaller compared to a data drop rate in order of 10^{-2} resulting from a typical constant bit rate video rate control algorithm that drops some video frames to control the bit rate.

5. Conclusions

A novel statistical multiplexing method was proposed for Mobile TV over DVB-H. The proposed statistical multiplexing method is implemented in conjunction with the time-sliced transmission scheme by the IP encapsulator in a DVB-H network. The proposed method solves the essential problem of delta-t calculation at the expense of a very small increase in power consumption of a DVB-H receiver. According to the experimental and analytical results, when comparing with deterministic multiplexing, the proposed statistical multiplexing method can decrease the end-to-end delay of a DVB-H network considerably without any data drop while the allocated bandwidth is used efficiently. The end-to-end delay decreases even further when a slight data drop rate is permitted. An analytical model was provided that estimates the performance of the proposed statistical multiplexing method assessing data drop rate and delay. The model was successfully validated through extensive simulations.

Acknowledgment

References

[1] ETSI, "Digital Video Broadcasting (DVB): Transmission Systems for Handheld Terminals," *ETSI Standard*, EN 302 304 V1.1.1, 2004.

[2] G. Faria, J. A. Henriksson, E. Stare, and P. Talmola, "DVB-H: digital broadcast services to handheld devices," *Proceedings of the IEEE*, vol. 94, no. 1, pp. 194–209, 2006.

[3] ETSI, "DVB-H Implementation Guidelines," *ETSI Standard*, TR 102377, July 2005.

[4] T. V. Lakshman, A. Ortega, and A. R. Reibman, "VBR video: tradeoffs and potentials," *Proceedings of the IEEE*, vol. 86, no. 5, pp. 952–972, 1998.

[5] M. Rezaei, I. Bouazizi, V. K. M. Vadakital, and M. Gabbouj, "Optimal channel changing delay for mobile TV over DVB-H," in *Proceedings of the IEEE International Conference on Portable Information Devices (PORTABLE '07)*, Orlando, Fla, USA, March 2007.

[6] M. Garrett and W. Willinger, "Analysis, modeling and generation of self-similar VBR video traffic," in *Proceedings of the Conference on Communications Architectures, Protocols and Applications (SIGCOMM '94)*, pp. 269–280, London, UK, August 1994.

[7] J. Beran, R. Sherman, M. S. Taqqu, and W. Willinger, "Long-range dependence in variable-bit-rate video traffic," *IEEE Transactions on Communications*, vol. 43, no. 2–4, part 3, pp. 1566–1579, 1995.

[8] M. Krunz and S. K. Tripathi, "On the characterization of VBR MPEG streams," *Performance Evaluation Review*, vol. 25, no. 1, pp. 192–202, 1997.

[9] M. R. Izquierdo and D. S. Reeves, "A survey of statistical source models for variable-bit-rate compressed video," *Multimedia Systems*, vol. 7, no. 3, pp. 199–213, 1999.

[10] U. K. Sarkar, S. Ramakrishnan, and D. Sarkar, "Modeling full-length video using Markov-modulated Gamma-based framework," *IEEE/ACM Transactions on Networking*, vol. 11, no. 4, pp. 638–649, 2003.

[11] M. Dai, D. Loguinov, and H. Radha, "A hybrid wavelet famework for modeling VBR video traffic," in *Proceedings of the IEEE International Conference on Image Processing (ICIP '04)*, 2004.

[12] M. Rezaei, I. Bouazizi, and M. Gabbouj, "A modle for controlled VBR video traffic," in *Proceedings of the IEEE International Conference on Signal Processing and Communications (ICSPC '07)*, pp. 1455–1458, Dubai, UAE, November 2007.

[13] B. Maglaris, D. Anastassiou, P. Sen, G. Karlsson, and J. D. Robbins, "Performance models of statistical multiplexing in packet video communications," *IEEE Transactions on Communications*, vol. 36, no. 7, pp. 834–844, 1988.

[14] H. Suzuki, T. Murase, S. Sato, and T. Takeuchi, "A burst traffic control strategy for ATM networks," in *Proceedings of the IEEE Global Telecommunications Conference and Exhibition*, vol. 2, pp. 874–878, December 1990.

[15] J. M. Hyman, A. A. Lazar, and G. Pacifici, "Real-time scheduling with quality of service constraints," *IEEE Journal on Selected Areas in Communications*, vol. 9, no. 7, pp. 1052–1063, 1991.

[16] R. J. Gibbens and P. J. Hunt, "Effective bandwidths for the multi-type UAS channel," *Queueing Systems*, vol. 9, no. 1-2, pp. 17–27, 1991.

[17] R. Guerin, H. Ahmadi, and M. Naghshineh, "Equivalent capacity and its application to bandwidth allocation in high-speed networks," *IEEE Journal on Selected Areas in Communications*, vol. 9, no. 7, pp. 968–981, 1991.

[18] A. Weiss, "Introduction to large deviations for communication networks," *IEEE Journal on Selected Areas in Communications*, vol. 13, no. 6, pp. 938–952, 1995.

[19] A. I. Elwalid and D. Mitra, "Effective bandwidth of general Markovian traffic sources and admission control of high speed networks," *IEEE/ACM Transactions on Networking*, vol. 1, no. 3, pp. 329–343, 1993.

[20] I. Norros, "On the use of fractional Brownian motion in the theory of connectionless networks," *IEEE Journal on Selected Areas in Communications*, vol. 13, no. 6, pp. 953–962, 1995.

[21] C.-S. Chang and J. A. Thomas, "Effective bandwidth in high-speed digital networks," *IEEE Journal on Selected Areas in Communications*, vol. 13, no. 6, pp. 1091–1099, 1995.

[22] W. Zhou, S. Ramakrishnan, D. Sarkar, and U. K. Sarkar, "Bandwidth estimation for multiplexed videos using MMG-based single video traffic model," in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM '03)*, vol. 6, pp. 3564–3568, December 2003.

[23] D. P. Heyman, A. Tabatabai, and T. V. Lakshman, "Statistical analysis and simulation study of video teleconference traffic in ATM networks," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 2, no. 1, pp. 49–59, 1992.

[24] F. Kelly, "Notes on effective bandwidth," in *Stochastic Networks: Theory and Applications*, Oxford University Press, Oxford, UK, 1996.

[25] T. Cheng and Y. Wu, "The calculation of effective bandwidth for video traffic on networks," in *Proceedings of the International Conference on Communications, Circuits and Systems*, vol. 1, pp. 104–108, May 2005.

[26] D. M. Cohen and D. P. Heyman, "Performance modeling of video teleconferencing in ATM networks," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 3, no. 6, pp. 408–420, 1993.

[27] A. F. Dagiuklas and M. Ghanbari, "Effect of video synchronization in an ATM multiplexer," *Electronics Letters*, vol. 30, no. 9, pp. 683–685, 1994.

[28] J. Feng, K.-T. Lo, and H. Mehrpour, "Simulation analysis on statistical multiplexing of MPEG video sources," in *IEEE International Conference on Communications (ICC '97)*, vol. 1, pp. 540–544, Montreal, Canada, June 1997.

[29] Z.-L. Zhang, J. Kurose, J. D. Salehi, and D. Towsley, "Smoothing, statistical multiplexing, and call admission control for stored video," *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 6, pp. 1148–1166, 1997.

[30] M. Krunz and S. K. Tripathi, "Impact of video scheduling on bandwidth allocation for multiplexed MPEG streams," *Multimedia Systems*, vol. 5, no. 6, pp. 347–357, 1997.

[31] P. Cuenca, B. Caminero, A. Garrido, F. Quiles, and L. Orozco-Barbosa, "QoS and statistical multiplexing performance of VBR MPEG-2 video sources over ATM networks," in *Proceedings of the IEEE Canadian Conference on Electrical and Computer Engineering*, vol. 1, pp. 33–36, May 1998.

[32] J. D. Salehi, Z.-L. Zhang, J. Kurose, and D. Towsley, "Supporting stored video: reducing rate variability and end-to-end resource requirements through optimal smoothing," *IEEE/ACM Transactions on Networking*, vol. 6, no. 4, pp. 397–410, 1998.

[33] S. H. Kang and A. Zakhor, "Effective bandwidth based scheduling for streaming media," *IEEE Transactions on Multimedia*, vol. 7, no. 6, pp. 1139–1148, 2005.

[34] ETSI, "IP Datacast over DVB-H: Content Delivery Protocols (CDP)," *ETSI standard*, TS 102 472 V1.2.1, December 2006.

[35] M. Rezaei, M. M. Hannuksela, and M. Gabbouj, "Tune-in time reduction in video streaming over DVB-H," *IEEE Transactions on Broadcasting*, vol. 53, no. 1, pp. 320–328, 2007.

[36] M. Rezaei, M. Gabbouj, and I. Bouazizi, "Delay constrained fuzzy rate control for video streaming over DVB-H," in *Proceedings of the International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIHMSP '06)*, pp. 223–227, Pasadena, Calif, USA, December 2006.

[37] H. E. Hurst, R. P. Black, and Y. M. Simaika, *Long-Term Storage: An Experimental Study*, Constable, London, UK, 1965.

[38] R. Coelho and L. Decreusefond, "Fractional Brownian motion usage for smoothed video traffic bandwidth allocation," in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM '97)*, vol. 2, pp. 1027–1030, November 1997.

[39] M. Rezaei, I. Bouazizi, and M. Gabbouj, "Generating antipersistent VBR video traffic," in *Proceedings of the Picture Coding Symposium (PCS '07)*, Lisbon, Portugal, November 2007.

[40] M. Rezaei, I. Bouazizi, and M. Gabbouj, "Statistical time-frequency multiplexing of HD video traffic in DVB-T2," *International Journal of Digital Multimedia Broadcasting*, vol. 2009, Article ID 186960, 12 pages, 2009.

[41] M. Rezaei, M. M. Hannuksela, and M. Gabbouj, "Semi-fuzzy rate controller for variable bit rate video," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 5, pp. 633–644, 2008.

[42] C. Courcoubetis and R. Weber, "Effective bandwidths for stationary sources," *Probability in Engineering and Information Sciences*, vol. 9, pp. 285–296, 1995.

[43] E. W. Knightly and N. B. Shroff, "Admission control for statistical QoS: theory and practice," *IEEE Network*, vol. 13, no. 2, pp. 20–29, 1999.

Special Issue on
Wireless Network Algorithms, Systems, and Applications

Call for Papers

Recent advances in wireless communications and computing technologies have paved the way for the proliferation of ubiquitous infrastructure and ad hoc wireless networks, enabling a broad range of applications ranging from protection of critical infrastructure to protection of wireless communications, from environment monitoring to health care, and from conducting business to improving quality of life. The need to deal with the complexity and ramifications of the fast-growing number of mobile users and services intensifies the interest in the development of fundamental principles, novel algorithmic approaches, rigorous and repeatable design methodologies, and systematic evaluation frameworks for the next-generation wireless networks.

The proposed special issue solicits technical papers that describe previously unpublished research work, visionary approaches, and future research directions dealing with effective and efficient algorithm design and analysis, reliable and secure system development and implementations, experimental study and test bed validation, as well as new application exploration in wireless networks. Topics of interest include, but are not limited to, the following:

- Wireless networks for cyber-physical systems (transportation, health care, civil infrastructure, etc.)
- Theoretical frameworks and efficient algorithm design
- PHY/MAC/Routing protocols
- Application and design of wireless ad hoc and sensor networks
- Experimental test-beds, models, and case studies

Call-for-papers for the proposed special issue will be distributed to wireless network communities. We will also encourage authors of WASA'09 (The Fourth International Annual Conference on Wireless Algorithms, Systems and Applications) to submit their work to this special issue. WASA'09 addresses similar problems of the special issue, including the research and development efforts of various issues in the area of algorithms, systems, and applications for the current and next-generation infrastructure and ad hoc wireless networks.

Before submission authors should carefully read over the journal's Author Guidelines, which are located at http://www.hindawi.com/journals/wcn/guidelines.html. Prospective authors should submit an electronic copy of their complete manuscript through the journal Manuscript Tracking System at http://mts.hindawi.com/, according to the following timetable:

Manuscript Due	September 1, 2009
First Round of Reviews	December 1, 2009
Publication Date	March 1, 2010

Lead Guest Editor

Benyuan Liu, Department of Computer Science, University of Massachusetts Lowell, Lowell, MA 01854, USA; bliu@cs.uml.edu

Guest Editors

Azer Bestavros, Department of Computer Science, Boston University, Boston, MA 02215, USA; best@bu.edu

Jie Wang, Department of Computer Science, University of Massachusetts Lowell, Lowell, MA 01854, USA; wang@cs.uml.edu

Ding-Zhu Du, Department of Computer Science, University of Texas at Dallas, Dallas, TX 75083, USA; dzdu@utdallas.edu

Special Issue on
Theoretical and Algorithmic Foundations of Wireless Ad Hoc and Sensor Networks

Call for Papers

This special issue is devoted to distributed algorithms and theoretical methods in the context of wireless ad hoc and sensor networks. Recent research in mobile ad hoc networks and wireless sensor networks raises a number of interesting, and difficult, theoretical and algorithmic issues. While much work has been done in protocol and system design, simulation, and experimental study for wireless ad hoc and sensor networks, the theoretical research, however, falls short of the expectation of the future networking deployment. The needs to push the theoretical research forward for a deeper understanding about wireless ad hoc and sensor networking and to foster cooperation among networking researchers and theoreticians establish the motivation behind this special issue.

The objective of this special issue is to gather recent advances in the areas of wireless ad hoc and sensor networks, with a focus on theoretical and algorithmic aspect. In particular, it will concentrate on distributed algorithms, randomized algorithms, analysis and modeling, optimizations, and theoretical methods in design and analysis of networking protocol (at link layer or network layer) for wireless ad hoc and sensor networks. Specific topics for this special issue dedicated to theoretical and algorithmic foundations include but are not limited to:

- Channel assignment and management
- Distributed and localized algorithms
- Dynamic and random networks
- Dynamic graph algorithms
- Energy conservation methods
- Localization and location tracking
- Mechanism design and game theory
- Modeling and complexity analysis
- Routing, multicast, and broadcast
- Scheduling and synchronization
- Throughput optimization and capacity

Before submission, authors should carefully read over the journal's Author Guidelines, which are located at http://www.hindawi.com/journals/wcn/guidelines.html. Prospective authors should submit an electronic copy of their complete manuscript through the journal Manuscript Tracking System at http://mts.hindawi.com/, according to the following timetable:

Manuscript Due	November 1, 2009
First Round of Reviews	February 1, 2010
Publication Date	May 1, 2010

Lead Guest Editor

Yu Wang, Department of Computer Science, University of North Carolina at Charlotte, Charlotte, NC 28223, USA; yu.wang@uncc.edu

Guest Editors

Xinbing Wang, Department of Electronic Engineering, Shanghai Jiaotong University, Shanghai 200240, China; xwang8@sjtu.edu.cn

Ping Ji, Department of Mathematics and Computer Science, John Jay College of Criminal Justice, City University of New York, New York, NY 10019, USA; pji@jjay.cuny.edu

Peter Müller, IBM Zurich Research Laboratory, 8803 Rüschlikon, Switzerland; pmu@zurich.ibm.com

Xiaohua Jia, Department of Computer Science, City University of Hong Kong, Kowloon, Hong Kong; csjia@cityu.edu.hk

Athanasios Vasilakos, Department of Computer and Telecommunications Engineering, University of Western Macedonia, Greece; vasilako@ath.forthnet.gr

Special Issue on

Multicamera Information Processing: Acquisition, Collaboration, Interpretation, and Production

Call for Papers

Video sensors have gained in resolution, quality, cost-efficiency, and ease of use during the last decade, thus fostering the deployment of rich acquisition settings, to cheaply and effectively capture scenes at high spatiotemporal resolution, in multiple locations and directions. By providing extended and redundant coverage, multicamera imaging provides a practical approach to support robust scene interpretation, integrated situation awareness, as well as rich interactive and immersive experience in many different areas of industry, health-care, education, and entertainment. Tools and algorithms that aim to recognize high-level semantic concepts and their spatiotemporal and causal relations directly depend on the robustness and reliability of the underlying detection and tracking methods. These tasks related to scene interpretation have a strong impact on many real-life applications and are also fundamental to understand how to render a scene, for example, in a sport event summarization context or while browsing multiview video surveillance content. Finally, multiview imaging allows for immersive visualization by adapting rendered images to display capabilities and/or viewer requests. The goal of this special issue is to present the recent theoretical and practical advances that take advantage of multiview processing to improve 3D scene monitoring, immersive rendering, and (semi-)automatic content creation. Topics of interest include, but are not limited to:

- Acquisition of multiview and 3D images
- Multicamera information fusion
- Automated extraction of calibration or geometry information
- Distributed scene representation and communication
- Depth estimates and arbitrary view synthesis
- Multiview object detection and tracking
- Multiview video stream events/activities mining
- Multiview event detection and recognition
- Assistance to interactive video browsing in a distributed surveillance camera network
- Immersive rendering, and 3D scene virtual navigation

- Automatic and/or personalized summarization of sports events
- Plants or impaired people monitoring applications
- Advanced application case studies

Before submission authors should carefully read over the journal's Author Guidelines, which are located at http://www.hindawi.com/journals/ivp/guidelines.html. Prospective authors should submit an electronic copy of their complete manuscript through the journal Manuscript Tracking System at http://mts.hindawi.com/ according to the following timetable:

Manuscript Due	December 1, 2009
First Round of Reviews	March 1, 2010
Publication Date	June 1, 2010

Lead Guest Editor

Christophe De Vleeschouwer, UCL, Louvain-la-Neuve, Belgium; christophe.devleeschouwer@uclouvain.be

Guest Editors

Andrea Cavallaro, Queen Mary, University of London, London, UK; andrea.cavallaro@elec.qmul.ac.uk

Pascal Frossard, EPFL, Lausanne, Switzerland; pascal.frossard@epfl.ch

Li-Qun Xu, British Telecommunications PLC, London, UK; li-qun.xu@bt.com

Peter Tu, GE Global Research, Niskayuna, NY, USA; tu@crd.ge.com