

Energy Optimization in Mobile TV Broadcast Networks

Mohamed Hefeeda and Cheng-Hsin Hsu
School of Computing Science
Simon Fraser University
Surrey, BC V3T 0A3, Canada

Abstract

We consider energy optimization in mobile TV networks in which a base station broadcasts multiple TV channels to mobile receivers over a common wireless medium. In these systems, the base station broadcasts TV channels in bursts with bit rates much higher than the encoding rates of the video streams. Thus, mobile devices can receive a burst of traffic and then turn off their radio frequency circuits till the next burst in order to save energy. To achieve this energy saving, the base station must carefully construct the burst schedule for all TV channels. In this paper, we formulate the problem of determining the optimal burst schedules for broadcasting multiple TV channels. We show that this problem is NP-complete for channels with arbitrary bit rates. We then propose a practical simplification of the general problem, which allows TV channels to have different bit rates, but at power of two increments. Using this simplification, we propose and analyze an optimal burst scheduling algorithm. We implement our algorithm in a mobile TV testbed and we present empirical results to demonstrate its efficiency and optimality.

1. Introduction

Mobile TV is expected to be a huge market in the near future. People are already using it in parts of Europe and Asia to watch TV shows and games on small hand-held devices while traveling [3]. A common issue in all mobile wireless devices is the limited energy supply since they are battery powered. Thus, minimizing the energy consumption in mobile TV systems becomes a critical problem for the success and wide adoption of such systems. We consider energy optimization in mobile TV systems in which a base station concurrently broadcasts multiple digital TV channels to mobile receivers. Examples of such systems include networks based on the Digital Video Broadcast-Handheld (DVB-H) standard [4, 6]. In these systems, the base station

broadcasts TV channels in *bursts* with bit rates much higher than the encoding rates of the video streams. Thus, mobile devices can receive a burst of traffic and then turn off their radio frequency (RF) circuits till the next burst. This is referred to as *time slicing* [6]. A burst transmission schedule needs to be computed in order to specify the burst start time and size for the time slicing module. Computing the burst transmission schedule for multiple TV channels is complex, because the schedule must: (i) ensure that receivers for any TV channel have no buffer overflow or underflow instances, and (ii) have no burst conflicts, which occur when two or more bursts intersect with each other in time.

Current practices of computing burst schedules are rather ad hoc. For example, the heuristic method proposed in the DVB-H standard documents [4, pp. 66] provides schedules for only one TV channel. This heuristic simply allocates a new burst only after the data of its preceding burst is consumed by the player at the receiver. This cannot be generalized to multiple TV channels with *different* bit rates, because the computed schedule may have burst conflicts and may result in buffer under/over instances. Thus, many mobile TV deployments have to resort to encoding all TV channels at the same bit rate in order to use the heuristic burst scheduling technique in the standard. This is clearly inefficient and may yield large quality variations among TV channels carrying different types of multimedia content, which is the typical case in practice. For example, encoding a high-motion soccer game requires much higher bit rate than encoding a low-motion talk show. If we encode all TV channels at the same high bit rate, some channels may unnecessarily be allocated more bandwidth than they require and this extra bandwidth yields only marginal or no visual quality improvement. Thus, the expensive wireless bandwidth of the broadcast network could be wasted. On the other hand, if we encode all TV channels at the same low or moderate bit rate, not all channels will have the same visual quality, which could be annoying to users of a *commercial* service. To the best of our knowledge, there exist no systematic ways in the literature that fully address the

burst scheduling problem, despite the tremendous potential of mobile TV systems.

In this paper, we formulate the burst scheduling problem in mobile TV systems. We, unfortunately, show that this problem is NP-complete for TV channels with arbitrary bit rates. We then propose a practical simplification of the general problem, which allows TV channels to be classified into multiple classes and each class has a different bit rate. The bit rate of class c , r_c , can take any value in the form of $r_c = 2^i \times r_1$, where $i \in \{0, 1, 2, 3, \dots\}$, and r_1 is the bit rate of the lowest class. r_1 can take any arbitrary bit rate. For example, the bit rates 800, 400, 200, and 100 kbps could make four different classes for encoding sports events, movies, low motion episodes, and talk shows, respectively. This classification of TV channels also enables the operators of mobile TV networks to offer differentiated services: higher bit rate classes can broadcast premium services for higher fees. This service differentiation is not possible with the current burst scheduling schemes. Using the above simplification, we develop an optimal (in terms of energy consumption) burst scheduling algorithm, which is quite efficient: its time complexity is $O(S \log S)$, where S is the total number of TV channels. We present empirical results from a mobile TV testbed to validate our theoretical analysis and to analyze the performance of our algorithm.

Related Work. A few previous works have addressed energy saving in mobile TV systems. The authors of [7] and [4] estimate the effectiveness of the time slicing technique for given burst schedules. These two works do not solve the burst scheduling problem, they only compute achieved energy saving for *pre-determined* burst schedules. In contrast, we formulate and solve the burst scheduling problem for arbitrary channel bit rates. The authors of [8] consider mobile TV receivers with an auxiliary short range wireless interface and construct a cooperative network among several receivers over this wireless network. Mobile TV receivers share received IP packets using this short range network, so that each receiver only receives a small fraction of IP packets directly from the broadcast network. This allows these receivers to reduce the frequency of turning on their RF circuits. The proposal in [8] is orthogonal and complementary to our work, as it resides in the mobile devices themselves and tries to achieve additional energy saving on top of that achieved by time slicing. In contrast, our algorithm would be implemented in the base station broadcasting TV channels to mobile receivers.

2. Problem Statement and Solution

We consider mobile TV systems in which a base station concurrently broadcasts S digital TV channels to clients with handheld devices over a wireless medium with bandwidth R kbps. Each TV channel s , $1 \leq s \leq S$, has a bit

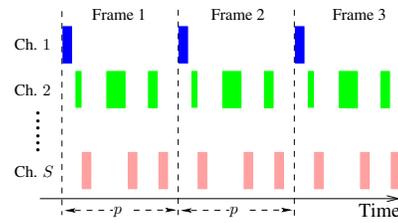


Figure 1. The burst scheduling problem.

rate r_s kbps. The base station broadcasts each TV channel in *bursts* at bit rate R kbps. After receiving a burst of data, the RF receivers are switched off till the time of the next burst, which is computed by the base station and included in the header fields of the current burst. Notice that the RF receivers of mobile TV receivers must be open slightly before the burst time, because it takes some time to wake up and synchronize the circuitry before it can start receiving data. This time is called the overhead duration and is denoted by T_o . With current technology, T_o is in the range of 50–250 msec [4, 6]. The energy saving achieved by mobile devices receiving TV channel s is denoted by γ_s and it is calculated as the ratio of time the channel is in off mode to the total time [4, 7]. We define the system-wide energy saving metric over all TV channels as $\gamma = (\sum_{s=1}^S \gamma_s) / S$. The energy saving and the burst scheduling itself are typically performed on a recurring time window called a frame, which is a system parameter and is denoted by p . Fig. 1 illustrates a simple example for the burst scheduling problem. Notice that the bursts have different sizes, are disjoint in time, and are repeated in successive frames.

Problem 1 (Burst Scheduling in Mobile TV Systems)

Given S TV channels of different bit rates to be simultaneously broadcast to mobile receivers. Each TV channel is broadcast as bursts of data to save the energy of mobile devices. Find the optimal transmission schedule for bursts of all TV channels to maximize the system-wide energy saving metric γ . The transmission schedule must specify the number of bursts for each TV channel in a frame as well as the start and end times for each burst. The schedule cannot have burst collisions, which happen when two or more bursts have nonempty intersection in time. In addition, the schedule must ensure that there are no receiver buffer violations for any channel. A buffer violation occurs when the receiver either has no data in the buffer to play out (buffer underflow), or has no space to store data during a burst transmission (buffer overflow).

Theorem 1 The burst scheduling problem stated in Problem 1 is NP-complete.

Proof Sketch. We show that the problem of minimizing the energy consumption (Problem 1) is the same as the prob-

P2OPT

```

1.  Compute optimal frame length  $p^*$ 
2.  Allocate a leaf node for every channel  $s$ 
3.  Push all leaf nodes to a priority queue  $P$  with key  $r_s/r_1$ 
4.  while true{
5.    let  $m_1 = \text{pop\_min}(P)$ ,  $m_2 = \text{pop\_min}(P)$ ;
6.    if  $m_2$  is null or  $m_1.\text{key} < m_2.\text{key}$  { // no sibling
7.      if  $m_2$  is not null push( $P$ ,  $m_2$ ); // return  $m_2$  to  $P$ 
8.      Allocate dummy node  $m_2$ ,  $m_2.\text{key} = m_1.\text{key}$ 
9.    }
10.   Allocate an internal node  $n$ 
11.   let  $n.\text{left} = m_1$ ,  $n.\text{right} = m_2$ ;
12.   let  $n.\text{key} = m_1.\text{key} + m_2.\text{key}$ ;
13.   push( $P$ ,  $n$ ); // insert this internal node
14.   if  $n.\text{key} \geq R/r_1$  break;
15.  }
16.  if  $|P| > 1$  return  $\emptyset$ ; // no feasible schedule
17.  let  $\mathbb{T} = \emptyset$ ; // start composing schedule
18.  Traverse the tree from root down, annotate each node
  with an offset with the value of the reverse bit pattern from
  root till this node
19.  foreach leaf node  $l$  {
20.    for  $i = 0$  to  $l.\text{key} - 1$  {
21.       $\text{start} = \text{offset} + i \times (R/r_1)/l.\text{key}$ ;
22.      insertBurst( $\mathbb{T}$ ,  $\text{start} \times p^*r_1/R$ ,  $l.\text{ch}$ );
23.    }
24.  }
25.  return  $\mathbb{T}$ ;

```

Figure 2. Burst scheduling algorithm.

lem of minimizing the total number of bursts in each frame. Then, we reduce the NP-complete problem of task sequencing with release times and deadlines [5, pp. 236] to it.

Burst Scheduling Algorithm. We propose an optimal algorithm for the burst scheduling problem when the bit rate of the TV channels s , $1 \leq s \leq S$, is given by $r_s = 2^i \times r_1$, where $i \in \{0, 1, 2, 3, \dots\}$, and r_1 can be any arbitrary bit rate. As mentioned in Sec. 1, the TV channels can be divided into classes, where each class contains similar-type multimedia content. Without loss of generality, we assume that the bit rates of the S channels are ordered such that $r_1 \leq r_2 \leq \dots \leq r_S$. If otherwise, a re-labeling of the bit rates is applied. We also assume that the bandwidth of the wireless medium satisfies $R = 2^k \times r_1$, where $k \in \{1, 2, 3, \dots\}$. We present in Fig. 2 an optimal algorithm for solving the burst scheduling problem in this case.

The basic idea of our algorithm is as follows. The algorithm first computes the optimal value for the frame length p^* . It then divides p^* into bursts of equal size p^*r_1 bits.

Thus, there are $(p^*R)/(p^*r_1) = R/r_1$ bursts in each frame. Then, each TV channel is allocated a number of bursts proportional to its bit rate. That is, TV channel s , $1 < s \leq S$, is allocated r_s/r_1 bursts and TV channel 1 is allocated only one burst in a each frame. Then, bursts of TV channel s are equally spaced within the frame, with inter-burst distance of $p^*/(r_s/r_1)$ sec. This ensures that there is no underflow instances in the receiver buffer, because the consumption rate of the data in the buffer for TV channel s is r_s bps and the burst size is p^*r_1 bits. Also the size of each burst is up to the receiver buffer size, which ensures that there is no buffer overflow instances. Finally, bursts of different TV channels are arranged such that they do not intersect in time, that is, the resulting schedule is conflict free.

To achieve the above steps in a systematic way, the pseudo code in Fig. 2 works as follows. It builds a binary tree bottom-up. Leaf nodes representing TV channels are created first, where the leaf node of TV channel s is annotated with the value r_s/r_1 . The algorithm uses this value as the key, and inserts all leaf nodes into a priority queue. This priority queue is implemented as a binary heap to efficiently find the node with the smallest key. The algorithm then repeatedly merges the two nodes that have the least key values into a new internal node. This new internal node has a key value equivalent to the sum of the key values of its children. This is done by popping the smallest two values from the heap, and then pushing the newly created node into it. The merging of nodes continues till the tree has a height of $\log R/r_1$. The last merged node becomes the root of the binary tree. Note that if the wireless medium is fully utilized by the TV channels, i.e., $\sum_{s=1}^S r_s = R$, the computed bursts of the different TV channels will completely fill the frame p . If otherwise (i.e., $< 100\%$ utilization), the wireless medium will have to be idle during some periods within the frame. The algorithm represents these idle periods as dummy nodes in the tree.

Once the binary tree is created, the algorithm constructs the burst schedule. It allocates to each TV a number of bursts that is equal to its key value. In order to ensure conflict-free schedule, the algorithm computes the start time for each burst as follows. For each leaf node representing a TV channel, the algorithm traverses the tree top-down. During the traversal, each node is assigned the reverse bit pattern from the root to this node, where the right branch has the bit 1 and the left branch has the bit 0. The bit pattern for a leaf node encodes the number of bursts and their start times for the TV channel corresponding to that node. For example, in a tree of depth 3, the bit pattern 010 means that the TV channel is assigned only the second burst in a frame of eight bursts. For leaf nodes at levels less than the depth of the tree, the bit pattern is padded with one or more 'x' to the left. For example, if a leaf node has the bit pattern x01 in a tree of depth 3, this means that the node is at

level 2 from the root. It also means that this node should be assigned the two bursts: 001 and 101. Notice that the first burst assigned to any TV channel is equal to the numeric value of its bit pattern, with all 'x' bits set to zero. The algorithm computes this value and refers to it as the offset. The algorithm then computes successive bursts relative to this offset.

Theorem 2 *The burst scheduling algorithm P2OPT in Fig. 2 returns a conflict-free schedule with no buffer under/over flow instances, if one exists. And it has a worst-case time complexity of $O(S \log S)$, where S is the number of TV channels.*

Proof Sketch. For the correctness part, we prove two issues. First, we prove that if P2OPT returns a schedule, this schedule is conflict-free schedule with no buffer under/over flow instances. We do this by analyzing the construction steps of the binary tree and showing that the bit patterns assigned to leaf nodes lead to no burst collision, enough number of bursts allocated to each channel, and correct inter-burst durations for each channel. Second, if P2OPT fails to return a schedule, there exists no feasible schedule for the given TV channels. For time complexity, P2OPT can be efficiently implemented using a binary heap, which can be initialized in time $O(S)$. Notice that we have at most $\log R/r_1$ dummy leaf nodes because there is at most one dummy leaf node for each level. The while loop in lines 4–15 iterates at most $O(S + \log R/r_1) = O(S)$ times, because $\log R/r_1$ can be considered as a constant for practical encoding bit rates. Since each iteration takes $O(\log S)$ steps, the while loop takes $O(S \log S)$ steps. Constructing the burst schedule in lines 17–24 takes $O(S)$ steps, since the tree has up to $2S$ nodes. Thus, the time complexity of P2OPT is $O(S \log S)$.

Theorem 3 *The frame duration $p^* = b/r_1$, where b is the receiver buffer size, computed by P2OPT maximizes the average energy saving γ over all TV channels.*

Proof Sketch. We use the fact that we established in the proof of Theorem 1: minimizing energy consumption is the same as minimizing the total number of bursts in each frame. Then, we show that any $p < b/r_1$ results in higher energy consumption. Then, we show that if there is another optimal schedule with $p > b/r_1$, we can find an equivalent optimal schedule with length equal to b/r_1 .

3. Evaluation using Mobile TV Testbed

We have set up a testbed for DVB-H networks [4, 6], which provides a realistic platform for analyzing our burst scheduling algorithm. The testbed has two parts: base station and receivers. The base station is a Linux box in which we installed the RF signal modulator available from [1].

This modulator implements the physical layer of the protocol stack and transmits DVB-H standard compliant signals via an indoor antenna. We have implemented an IP encapsulator for DVB-H systems, which encapsulates IP packets of video streams into transport streams. We have implemented our P2OPT burst scheduling algorithm in this IP encapsulator. We use the Nokia N92 and N96 devices as the TV receiver to assess the visual quality of videos. We also use the DVB-H Analyzer available from [2] to collect and analyze detailed information on the RF signals and DVB-H channels.

We configure the modulator to use a 5 MHz radio channel with Quadrature Phase-Shift Keying (QPSK) modulation. This leads to 5.445 Mbps air medium bandwidth according to the DVB-H standard [4]. We concurrently broadcast 9 TV channels using our P2OPT algorithm for 10 minutes. These TV channels are classified into 4 classes: 2 channels at 64 kbps, 3 channels at 256 kbps, 2 channels at 512 kbps, and 2 channels at 1024 kbps. The receiver buffer size is 1 Mb. For each of these TV channels, we set up a streaming server on the base station to send 1-kB IP packets at the specified bit rate. To conduct statistically meaningful performance analysis, we collect detailed event logs from the base station. The logs contain the start and end times (in msec) of broadcasting every burst of data and its size.

We develop software utilities to analyze the logs for three performance metrics: bit rate, time spacing between successive bursts, and energy saving. We compute the bit rate for each TV channel by considering the start times of two consecutive bursts and the burst size. We use the bit rate to verify that our burst scheduling algorithm leads to no buffer under/overflow instances. We compute time spacing between bursts by first sorting bursts of all TV channels based on their start times. Then, we sequentially compute the time spacing between the start time of a burst and the end time of its immediate, previous, burst. We use the time spacing to verify that there are no burst conflicts, as a positive time spacing indicates bursts do not intersect with each other. We compute the energy saving for each TV channel as the ratio between the RF circuit on time and off time. We set the overhead duration $T_o = 100$ msec.

Due to space limitations, we present only a sample of our results. We first validate the correctness of our P2OPT algorithm from the actual testbed implementation. Fig. 3(a) demonstrates the correctness of the P2OPT algorithm, where we plot the cumulative data received by receivers of a sample channel as the time progresses (other results are similar). We also show the consumed data with the time. To account for the worst case, we assume that the receiver starts consuming (playing back) the video data immediately after receiving a burst. The figure clearly shows that there are: (i) no buffer underflow instances as the consumption line never crosses the stair-case curve represent-

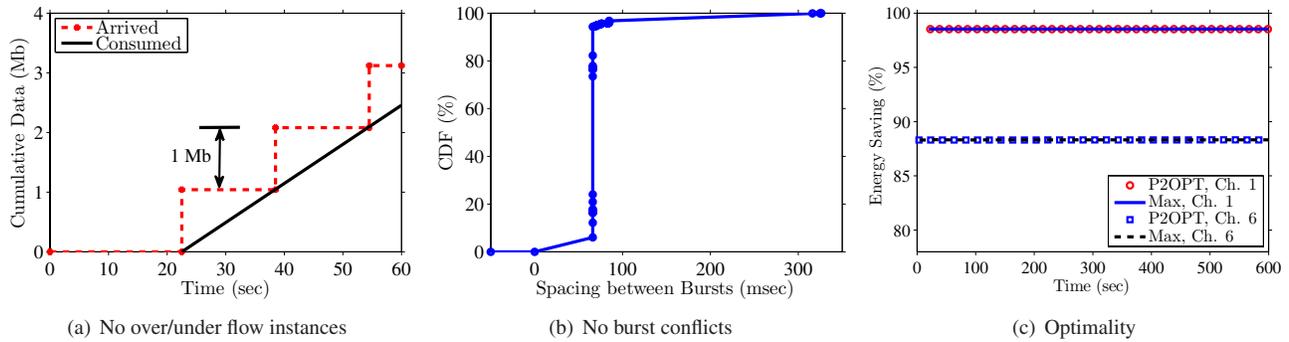


Figure 3. Empirical validation of the correctness and optimality of the P2OPT algorithm.

ing arrived data, and (ii) no buffer overflow instances as the distance between the data arrival and consumption curves never exceeds the buffer size (1 Mb). In order to show that there are no burst conflicts, we plot the cumulative distribution function (CDF) of the time spacing between successive bursts in Fig. 3(b). This CDF curve is computed from all bursts of all TV channels broadcast during the experiment period (10 minutes). Negative time spacing would indicate that two bursts are intersecting in time, i.e., burst conflict. This figure clearly shows that our P2OPT algorithm results in no burst conflicts.

Finally, we verify that the energy saving achieved by our P2OPT algorithm is indeed optimal. To do this, we compute the absolute maximum energy saving that can be achieved by any algorithm for a given TV channel. We compute this maximum by making the base station broadcast only the given TV channel. In this case, the base station easily maximizes the energy saving by allocating the largest burst that can fill the receiver's buffer. The RF circuit of the receiver is then turned off till the data of this burst is consumed. We repeat this experiment nine times; once for each considered TV channel, and we compute the maximum possible energy saving. We then run our algorithm to compute the burst schedule for the nine TV channels, and we make the base station broadcast all of them *concurrently*. We compute the energy saving for each TV channel in case. Sample results for channels 1 and 6 are presented in Fig. 3(c); all other results are similar. The figure confirms the optimality of the P2OPT algorithm in terms of energy saving.

4. Conclusions

In this paper, we formulated the burst scheduling problem in mobile TV systems. We showed that the general burst scheduling problem for multiple TV channels with arbitrary bit rates is NP-complete. We proposed a practical variation of the general problem that enables the use of different classes for video streams, where the encoding

bit rates can take on power of two increments. Multiple video classes with various bit rates enable network operators to define their own objectives such as quality equalization, quality maximization, and even a general utility function to support premium services for higher fees. We proposed and analyzed an optimal burst scheduling algorithm that runs in $O(S \log S)$ time, where S is the number of TV channels. We implemented the proposed algorithm in a real testbed for one of the major mobile TV networks: DVB-H [6]. We analyzed several logs from the testbed and empirically showed that our algorithm yields optimal energy savings, and produces valid burst schedules with no buffer under/overflow instances and no burst conflicts.

References

- [1] Dektec DTA-110T PCI modulator, 2008. <http://www.dektec.com/Products/DTA-110T/>.
- [2] Divi Catch RF-T/H transport stream analyzer, 2008. <http://www.enensys.com/>.
- [3] Digital Video Broadcasting - Handheld (DVB-H) home page, 2008. <http://www.dvb-h.org/>.
- [4] Digital Video Broadcasting (DVB); DVB-H implementation guidelines. European Telecommunications Standards Institute (ETSI) Standard EN 102 377 Ver. 1.3.1, May 2007.
- [5] M. Garey and D. Johnson. *Computers and Intractability: A Guide to The Theory of NP-completeness*. W. H. Freeman, 1979.
- [6] M. Kornfeld and G. May. DVB-H and IP Datacast – broadcast to handheld devices. *IEEE Transactions on Broadcasting*, 53(1):161–170, March 2007.
- [7] X. Yang, Y. Song, T. Owens, J. Cosmas, and T. Itagaki. Performance analysis of time slicing in DVB-H. In *Proc. of Joint IST Workshop on Mobile Future and Symposium on Trends in Communications (SympoTIC'04)*, pages 183–186, Bratislava, Slovakia, October 2004.
- [8] Q. Zhang, F. Fitzek, and M. Katz. Cooperative power saving strategies for IP-services supported over DVB-H networks. In *Proc. of IEEE Wireless Communications and Networking Conference (WCNC'07)*, pages 4107–4111, Hong Kong, China, March 2007.