

# Research and Application of Scheduling Algorithm for Digital Television Multiplexer

Xiuzhi Yang\*, Jian Chen and Weixing Wang

College of Physics and Information Engineering, Fuzhou University, 3500108, Fuzhou, Fujian, China

Received: 19 Jun. 2013, Revised: 25 Oct. 2013, Accepted: 26 Oct. 2013

Published online: 1 Jan. 2014

**Abstract:** The DTV program multiplexer in system level needs to combine many programs and some message into multi-program TS for transporting in the channel. Because DTV program contains audio and video signal, and the fluctuation of code rate is very large, data packet is easy to lose in the process of combining them into a data stream at a steady rate. Therefore, the research on dispatch algorithm about DTV multiplexer is of vital importance. This paper puts forward a two-level joint dispatching method with monopolizing and sharing buffers, which gets packets from monopolizing buffers respectively, and forms a buffer array supporting priority in sharing buffer, and then outputs at steady rate. Through verify this method has less data loss rate at equivalent serve intension.

**Keywords:** Multiplexer, Two-level Dispatch, Sharing Buffer

## 1 Introduction

In the front end of digital television system, several channels of program signals which are mutual independent from each other, combine together after multiplexer, go through code modulation, and share a channel, and then transmit to households. Under the premise of preserving users' looker-in quality, it hopes to transmit more programs in a channel and use its capacity as far as possible. Because input stream in digital television contains different types of signal such as sound and video signal, program and user information, and so on, the code rate is not constant without a certain rule, and it does not permit to obliterate data excessively in the sending process, which will affect user's looker-in effect. It becomes a necessary problem about how to dispatch packages reasonably in multiplying, and send data out at an even rate as the channel tolerates. Therefore, the research in digital television multiplexer's dispatch algorithm and realization structure is of vital significance [1, 2].

There are several literatures in domestic and foreign about dispatch methods of digital multimedia multiplexer. In literature [3] it mentioned that multiplying technology might require coordinating with encoder, and encoder's compression ratio should be controlled according to channel capacity, but it wasn't convenient to realize in

multiplexer's major applications, such as TV programs' system-level multiplying in local television station even county television station [4]. In literature [5, 6], it used a monopolizing-buffer multiplying method which established a buffer for each input program, but data had a high missing rate although different dispatch algorithms were introduced. In literature [7], qualitative analysis was carried on about cell loss ratio and transmission delay feature separately by the ways of monopolizing and sharing buffer multiplying, and it obtained a conclusion that data missing ratio in sharing cushion method was lower than monopolizing cushion way when transmitting equivalent stream, but the former average latency is longer than the latter. However this article had not solved the problem how to unite multiple channels to one, and there was still a section of distance between analysis result and actual situation.

This article proposes a type of two-level joint dispatching method with monopolizing and sharing buffers. Firstly data packets are taken out at a high speed from each unshared buffers referring to dynamic priority dispatching algorithm in real-time system [8, 9], then sent to the sharing buffer where cushioning array is constituted to support priority, finally output referring to the code rate tolerated by channel, so as to achieve reasonable dispatching goal.

\* Corresponding author e-mail: [yangxz@fzu.edu.cn](mailto:yangxz@fzu.edu.cn)

## 2 Monopolizing-sharing Joint Buffer Model and its Design Realization

Monopolizing-sharing joint buffer model is shown as figure 1.

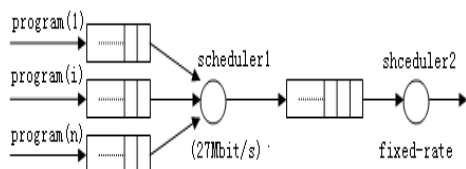


Fig. 1: Monopolizing-sharing buffer model

The goal of first buffer level is to reshape and cushion data from each source, and provide necessary standby period for dispatching data from each input channel to sharing buffer by the form of packet. Because the aim of first cushioning level is to transfer data to the second level buffer, its capacity doesn't need to be so big. The goal of second cushioning level is to provide buffers for data from each channels lining up and waiting to be sent. For smoothing the fluctuation of cell's speed, the cushioning capacity in second level should be set rather large.

In this model, it needs to dispatch data packets for two times, and be completed by the first and second level of dispatch.

**Dispatching algorithm in first level.** The multiplexer transmits stream in the monopolizing buffer to the sharing buffer by enough speed according to LSF (*Least Slack-Time First*) dispatching algorithm. In the case of very small transmitting delay, it can guarantee data from dropping. As following this algorithm will be proved and minimum processing speed is calculated in this level.

LSF dispatching algorithm had already been approved the most superior dispatching algorithm [10]. When and only when  $U < 1$ , the system is able to be controlled, where  $U = \sum_{i=1}^N \frac{C_i}{T_i}$ , when N channels of stream arrive simultaneously, the program's code rate of channel i is  $v_i$ , the arriving cycle is  $T_i = 1/v_i$  ( $T_i$  is invariable),  $C_i$  is the operating time for extracting a data packet from channel i and storing to the following sharing buffer.

In LSF algorithm, task's priority is to dynamically allocate package based on task's idle time. Shorter the idle time is, higher is the priority. Idle time is equal to subtract task execution time from closure time limit.

Supposing reading rate from unshared buffer is  $r'$ ,  $r' = m * M$ , in which  $M$  is channel capacity, namely output rate of multiplexer,  $m$  is a coefficient. Data reading from unshared buffer takes byte as an unit, and is deposited in the following sharing buffer, therefore speed  $r'$  is equal to processor's working speed in real-time

system. The package length of transport stream in digital television is 188 bytes or 204 bytes. We take 188 bytes as an example for analysis, and may obtain each packet's processing time  $C_i$ , where  $C_i = 188/r' = 188/(m * M)$ .

If N channels of programs are simultaneously transmitted,  $U$  may be calculated by considering two kinds of extreme cases:

(1) When each channel is pure video signal with high speed fluctuation, in limiting case cell's average rate is  $M/N$ , the maximum speed does not surpass average speed by 3 times.

For the moment, we can infer the most minimum cycle of data arriving is  $188 * 8 * N / (3 * M)$  and calculate its  $U$  value, also namely the biggest  $U$  value, where

$$U = \sum_{i=1}^N \frac{C_i}{T_i} = \sum_{i=1}^N \frac{188/(m * M)}{188 * 8 * N / (3 * M)} = \frac{3}{8 * M} \quad (1)$$

(2) When the average rate  $\bar{v}_i$  of one input channel is very high, and the fluctuations of other channels are small, it's considered that the capacity of transmitting channel as well as the average rate of SD (*standard definition*) programs in digital television is generally  $6 \sim 8M$ , while HD (*high definition*) is usually  $10 \sim 13M$ . It is supposed to be 1/3 of channel capacity, and its maximum rate is 3 times of average rate, then the minimum cycle of packet arriving is  $188 * 8 * 3 / (3 * M)$ , while code rate of other  $N - 1$  channels are all  $2/3M / (N - 1)$ , so the biggest  $U$  value is:

$$U = \frac{188/(m * M)}{188 * 8 * 3 / (3 * M)} + \sum_{i=2}^N \frac{188/(m * M)}{188 * 8 * (N - 1) / (2/3) * M} = \frac{5}{24 * m} \quad (2)$$

In actual work of DTV system, above limiting case is impossible to appear, namely the actual situation surpasses the above two suppositions.

By integrating Eq. (1) with (2), it can be see that when supposing  $m > 3/8 = 0.375$ , it will satisfy  $U < 1$  and the system can be controlled without data loss.

In actual realization, it's considered that multiplexer's dispatch must take package as a unit, so read-out speed from monopolizing buffer is 27M (*multiplexer's operating frequency*). When channel's transmitting rate is  $36 \sim 48M$ , the corresponding value  $m$  should be in the scope of  $0.75 \sim 0.56$ , which is far greater than the scheduled threshold value, so as to guarantee data from dropping in this level.

Due to introducing buffer, transmission delay is generated. Supposing transmission delay in this level is  $D_1$ , its relationships with buffer capacity  $b$  and output speed  $r'$  may be expressed as:

$$D_1 = b/r' \quad (3)$$

Since data processing speed in this level is quick enough, while  $b$  is generally very small in range of  $512 \sim 1K$  bytes, therefore  $D_1$  is usually quite small in the magnitude of  $\mu s$ , which may be ignored.

**Dispatch algorithm in the second level.** The data packets which read out from  $N$  channels of unshared buffers according to the priority deposit in the sharing buffer provided with FIFO structure, form cushioning formation with priority, and then transmit to channel at given rate.

This algorithm in this level may guarantee the important data inserting at fixed time without losing. By fully using large capacity of buffer, low dropping rate can be realized.

Below we analyze loss situation of data packet in sharing buffer.

Supposing the speed of source  $S_i$  at time  $t$  is  $v_i(t)$ , then the compound speed at which  $N$  sources enter the sharing buffer with a capacity of  $B$  is:

$$V(t) = \sum_{i=1}^N v_i(t) \tag{4}$$

Assuming that the length of package queue in buffer is  $f(t)$ , the probability of data loss at this moment is:

$$\begin{cases} P(t) = 1, f(t) = \text{int}(\frac{B}{188*8}) \\ P(t) = 0, f(t) < \text{int}(\frac{B}{188*8}) \end{cases} \tag{5}$$

Then the quantity of data loss in sharing-buffer multiplexing during the period of  $[0, T]$  is

$$L2 = \int_0^T (\frac{V(t) - M}{188*8}) p(t) dt \tag{6}$$

When buffer queue is not empty (*i.e.*  $f(t) > 0$ ), the time in  $[0, T]$  can be divided into  $n$  numbers of interval  $[x_j, y_j]$  without overlap. As cell loss only possibly occurs in these intervals, therefore counting of data loss in the time interval  $[x_j, y_j]$  is

$$l_j = \int_{x_j}^{y_j} (\frac{V(t) - M}{188*8}) p(t) dt \tag{7}$$

It's supposed that the first time abiding by  $f(t) = \text{int}(\frac{B}{188*8})$  in interval  $[x_j, y_j]$  is  $x'_j$ , and the last time is  $y'_j$ , then[7]

$$l_j = \int_{x'_j}^{y'_j} (\frac{V(t) - M}{188*8}) dt \tag{8}$$

As the size of buffer queue in the time interval  $[x_j, x'_j]$  grows from 0 to  $f(t) = \text{int}(\frac{B}{188*8})$ , so that

$$l_j = \int_{x_j}^{y'_j} \sum_{i=1}^N \frac{v_i(t)}{188*8} dt - (\int_{x_j}^{y'_j} \frac{M}{188*8} dt + \frac{B}{188*8}) \tag{9}$$

Obviously,

$$\begin{aligned} L2 &= \sum_{j=1}^n l_j = \int_0^T \sum_{i=1}^N \frac{v_i(t)}{188*8} dt - (\int_0^T \frac{M}{188*8} dt + \frac{B}{188*8}) \\ &= \int_0^T \sum_{i=1}^N \frac{v_i(t)}{188*8} dt - (\frac{MT + B}{188*8}) \end{aligned} \tag{10}$$

The probability of data loss is:

$$P(L2 > 0) = P[\sum_{i=1}^N \overline{v_i(t)}^T > (M + \frac{B}{T})] \tag{11}$$

In Eq. (11),  $\overline{v_i(t)}^T = \int_0^T v_i(t) dt$  is the mean value of code rate in channel  $k$  during the time interval  $[0, T]$ . If the code rate of input stream awaiting to multiplex has the same statistical property, and obeys Gaussian distribution  $N(\bar{v}_i, \sigma_i^2) (1 \leq i \leq N)$ [5], that is to say,  $\sum_{i=1}^N \bar{v}_i^T$  is obedient to Gaussian distribution with an average value  $\sum_{i=1}^N \bar{v}_i$  and variance  $\sqrt{\sum_{i=1}^N \frac{\sigma_i^2}{T}}$ . It can be seen from above, as a result of rate's addition of each channel, total rate fluctuation arriving at sharing buffer reduces, which has held down the peak value of code rate, so as to ensure total dropping rate by sharing-buffer multiplying far smaller than the design of unshared buffer[6].

It may also be seen from Eq.(11), smaller the sum of rate's mean value is, fewer is the drop probability; while increasing the capacity of sharing buffer  $B$  may reduce dropping probability, but as multiplying time extends gradually, the influence for buffer size to the dropping probability would reduce gradually, therefore  $B$  value should be set suitably.

The maximum transmitting delay for sharing-buffer multiplying is

$$D_2 = B/M \tag{12}$$

As  $B$  usually takes  $16K \sim 64K$ ,  $D_2$  is about 1ms in general. The biggest transmitting delay which MPEG-2 decoder can tolerate is approximately 30ms, therefore this postpone does less affect to decoder.

The above two-level dispatching method has solved the problem of merging multi-channel input stream into one without losing, and realized sharing-buffer multiplying. Its quantity of data loss is smaller than the design of monopolizing buffer, under the situation of similar buffer capacity and bandwidth utilization. In this method, data packet's transmitting delay meets the requirement of decoder.

### 3 Structure Design of Dispatch Module

The implementation of two-level scheduling module is as follows: the first level schedules data is at a large rate (27MHz) in accordance with LSF algorithm, then who

has shorter idle time owns higher priority; the second level constitutes output queue according to priority, and sends packets out at a fixed code rate predetermined. The structure of scheduling module is shown in figure 2.

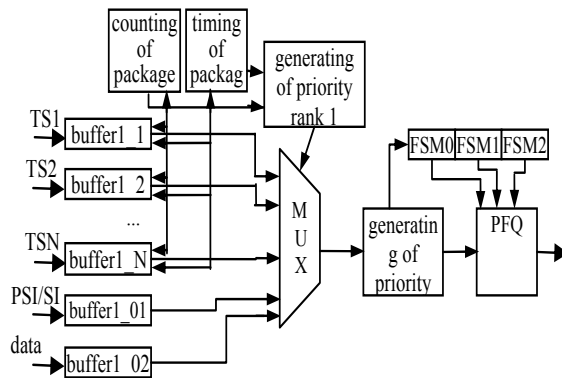


Fig. 2: The structure of scheduling module

**Design of the first dispatch module.** On design for dispatching in the first level, in order to guarantee the principle that whose idle time is shortest will gain the highest priority, two algorithms named first full first out and first in first out are put forward: Who contains most packets has the highest priority, and those carrying with same packets have the same priority; if several input channels have the same priority, first in first out strategy may be imported. The weights of these two algorithms add together, who has the maximum weight will gain prior dispatching opportunity. The strategy of first full first out allocates more sending opportunities to input TS packets with high rate, and assigns fewer opportunities to those with low rate. The strategy of first in first out can firstly dispatch the packets that has stayed longer when they have equivalent packets in buffers. Therefore, properly adoption of these two strategies can reduce transmitting latency of data stream, so as to improve the quality of video playback.

The scheduling module in first level is mainly composed of counting circuit of package number, timing circuit of package arrival and generating circuit of priority rank.

When input rate of each channel is quite uniform, and their sum is equal to or smaller than output rate, each input buffer only needs a capacity of two packages. Taking the uniformity nature of input rate into account, each buffer should leave some redundancy. Because the first-level dispatch operates at a system clock up to 27M, so it doesn't require a large buffer capacity. This design chooses a capacity of four packages, and kinds of actual test show that, lossless scheduling can be achieved

without packet loss in the case that output rate is appropriately set.

**Design of the second dispatch module.** In order to guarantee some important information from losing, scheduling module in this level also establishes priority rank according to data type, PSI/SI information which needs to transmit at fixed time is defined as the highest priority, then the secondary priority is assigned to program data containing PCR which must ensure to be sent, the following priorities are allocated to other video and audio TS packages which don't contain PCR. When new packets arrive, their order should be arranged by the following principle: firstly packets are inserted into queue on the basis of priority, the one with higher priority level would stand at the right side of the lower. If their priorities are equal, they will line up according to arrival order, the one coming later would stand at the left side of the former[11]. When there is no program data in buffer, empty pack (*PID is 1FFF*) should be output.

Every arrival of a new packet will trigger a series of shifting operation. All data with lower priority in buffer will shift left in turn, and new packet will be inserted in proper position. The discarding strategy adopted is that: If the buffer is full, the data packet with lowest priority will be discarded at first.

The internal control logic in second dispatching level is mainly completed by three state machines FSM0, FSM1 and FSM2. FSM0: receives data packets from input port and puts them in the queue. FSM1: shifts data. FSM2: controls data arranged at the rightmost side to output at fixed code rate.

## 4 Experiment Result

Two-level dispatch algorithm with monopolizing-sharing buffer is used in the multiplexer, which is realized on FPGA chip of EP2S90F1020C4 from Altera Corporation, and whose working condition has been tested through stream analyzer, simultaneously monopolizing-sharing buffer multiplying model is constructed by Matlab, as well as simulating test. The test should be carried on in two aspects: bandwidth utilization and data loss probability.

### 4.1 Simulation Test Result by Matlab

In the model of Matlab, there are 9 channels of input program sources, in which one channel simulates PSI/SI information and data packet transmits one time at every 100ms regularly; the rates of other 8 channels with common video and audio data conform to Gaussian distribution. In the monopolizing-buffer multiplying model, the capacity of unshared buffer is  $1KB \sim 4KB$ , using LSF dispatch algorithm; in the monopolizing-sharing buffer multiplying model, the



capacity of unshared buffer is 1KB while shared 8KB ~ 32KB. Through the above two-level dispatch algorithm, testing result is shown in figure 3.

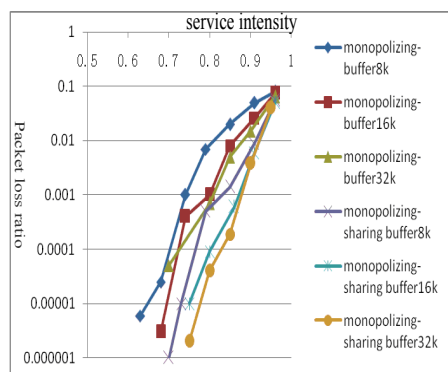


Fig. 3: Performance comparison of two scheduling algorithms

In Fig.3, capacity of unshared buffer is the sum of 8-channel buffers. The ratio about the sum of all input programs' average rate to output rate of multiplexer is called service intensity. As showed in Fig.3, when service intensity is larger than 0.70, two-level dispatching method surpasses one level obviously under the same service intensity; the difference is not so obvious when service intensity is smaller than 0.70. This also explains that two-level dispatching method has higher bandwidth utilization under the same broadcasting quality. As seen from Fig.3, dropping ratio with monopolizing-sharing cushion is also obviously better than monopolizing-buffer multiplying, with same buffer capacity and within a certain range ( $service\ intensity < 0.9$ ); at this moment, addition to buffer capacity is able to reduce cell's losing ratio effectively. But if it surpasses this scope, the result of increasing buffer capacity is not so good. Dropping ratios of these two kinds of multiplying ways also approach gradually. After the service intensity surpasses 0.95, it's unable to improve cell's losing rate through increasing buffer basically.

#### 4.2 Test Result of Multiplexer Using Two-level Dispatch

The TV programs which have been received from satellite are merged by multiplexer using two-level dispatch algorithm with monopolizing-sharing cushion, and the multiplexed data is analyzed by stream analyzer. Although program signal from satellite is used directly, the relationships between service intensity, buffer capacity and data loss probability may also be found on quality. Simulation result is shown in Fig. 4, and test situation is close to Matlab simulation.

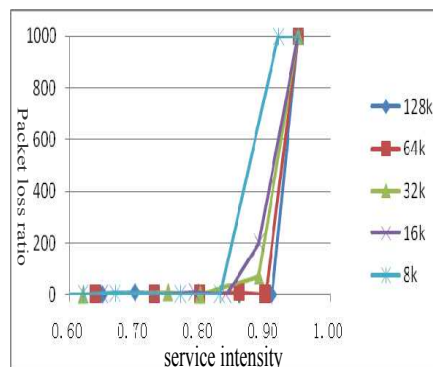


Fig. 4: Testing result of multiplexer using two-level dispatch

## 5 Conclusion

Seen from the simulation result and test situation, in the normal using scope of multiplexer ( $service\ intensity$  between 0.7 ~ 0.9), data loss rate of scheduling algorithm with monopolizing-sharing buffer is obviously shorter than purely unshared-buffer dispatch. When the service intensity is larger than 0.9, their effects may be similar; when the service intensity is larger than 0.95, loss rate of these two methods would rise suddenly. Thus the second-level dispatching algorithm is only suitable for the situation that service intensity is smaller than 0.9. The test result also indicates that with the same buffer capacity, loss rate of dispatching algorithm in the second level is less than the first level. Therefore, under the same service intensity and buffer capacity, the performance of second-level dispatching algorithm is better than unshared-buffer dispatch. In the case of same broadcast quality, the multiplexer with second-level dispatching strategy raises the use factor of channel bandwidth, reduces the amount of buffer's usage, and cuts down designing cost. Meanwhile, it may also be seen from the experimental result, when the buffer capacity is small (e.g. smaller than 16K), through increasing buffer capacity it's possible to relieve data loss rate. When the buffer capacity is larger than 32K, this type of improvement will be not obvious. That is to say, as for buffer capacity, the bigger doesn't mean the better, for example, 32K ~ 64K will be OK.

## Acknowledgement

This paper is supported by significant and special topic funded project of science and technology in Fujian "Key technological research and industrial production of digital television launching system" (2009HZ0003-1) and major project from education department in Fujian "One kind of new H.264 real-time coding system's research in HD digital television" (JK2009001).

## References

- [1] Wang Xingdong, Yu Songyu, Liang Longfei, Design of MPEG 2 Transport Stream Remultiplexer, Journal of Shanghai Jiaotong University, **37**, 824-828 (2003).
- [2] Zhao Zhongwei, Li Shiju, Statistical multiplexing model in the packet switching network and algorithms implemented on CATV network, Journal of Zhejiang University (Engineering Science), **38**, 145-148 (2004).
- [3] D. L. Wang, Joint-rate Controlling Technology in Statistical Multiplexing, Digital Communication World, **12**, 64-66 (2009).
- [4] ISO/IEC International Standard 13818-1, Generic Coding of Moving Picture and Associated Audio System, Recommendation H, 220.0 (1996).
- [5] J. Wen, R. Zhang, X. L. Ru, Effective Bandwidths Under Multiplexing Traffic, Systems Engineering and Electronics, **24**, 84-86 (2002).
- [6] L. Y. Cao, G. L. Yang, J. X. He, An Algorithm of Multiple Media Stream Scheduler Based on Weight. Journal of Hunan Institute of Engineering, **17**, 41-45 (2007).
- [7] Z. Wang, X. B. Leng, L. Z. Fu, G. X. Luan, Statistical Multiplexing Characteristics of VBR MPEG-2 Traffic, Journal of China Institute of Communications, **23**, 61-69 (2002).
- [8] X. L. Bin, Research on the Scheduling Technology of Tasks in Real-Time System, (Institute of Computer Science National University of Defense Technology, Changsha, Hunan, P.R.China, (2004).
- [9] Z. P. Wang, G. Z. Xiong, Study of Real-time Scheduling Algorithms, Journal of UEST of China, **29**, 205-208 (2000).
- [10] Clifford W. Mercer, An Introduction to Real-Time Operating Systems: Scheduling Theory, School of Computer Science; Carnegie Mellon University, (1992).
- [11] Y. H. Yang, X. L. Bin, Y. Q. Zheng, A Priority Buffering Queue Architecture in High-Speed Switch, Computer Engineering and Applications, **39**, 128-131 (2003).



internal and international journals. Her research interests are in the areas of image processing, video frequency information analysis and digital television technology.



research interests is media communication technology.



vision.

**Yang Xiuzhi** is currently Associate Professor of the College of Physics and Information Engineering at Fuzhou University, China, She is also System laboratory director of Fuzhou University DTV Engineering Research Center simultaneity. She has published over 30 papers in

**Chen Jian** received the MS degree in communication and information system from Fuzhou University. She is lecturer in the College of Physics and Information Engineering at Fuzhou University in China, and also a student for her PhD degree in the same specialty. Her

**Wang Weixing** received his PhD degree at Royal Institute of Technology in Sweden in 1997, he is an professor and PhD supervisor in Computer Science and Technology. His main research interests are :Image processing, Pattern recognition and Computer