Pre-order Bonded Deficit Round Robin for Channel Bonding System*

Yang Zhongzhen[1,2], Wang Jinin[1] , Wang Xianguan[1] and Jiao Chengwen[2]

[1]National Network New Media Engineering Research Center, Institute of Acoustics, Chinese Academy of Sciences Beijing, China
[2]University of Chinese Academy of Sciences Beijing, China
{yangzz , wangjl&wangxg}@dsp.ac.cn jcw880403@163.com

Abstract – For multiple channels packet scheduling in DOCSIS3.0, we propose a new scheduling algorithm, called Preorder Bonded Deficit Round Robin (PB-DRR) scheduling algorithm. The PB-DRR employs the preorder mechanism for Bonded Deficit Round Robin (BDRR) to overcome the large latency problem of BDRR. Theoretical analysis proves that PB-DRR is a latency-rate server and has O(logz) complexity. Simulation results show that PBDRR have more advantages in delay guarantee compared with BDRR.

Index Terms – Bonded Deficit Round Robin (BDRR), DOCSIS3.0, Channel Bonding, Latency-Rate, Packet Scheduler Algorithm

1. INTRODUCTION

To boost up network throughput, the Data over Cable Service Interface Specification (DOCSIS) 3.0 uses channel bonding technology to aggregate multiple physical channels into a virtual high bandwidth channel [1]. DOCSIS ensures the Quality of Service (QoS) by mapping packets into service flows, which is separate queue. Packet scheduling algorithm in cable modem termination system (CMTS) plays a critical role in the performance of DOCSIS3.0 network. However, due to the channel bonding and QoS requirements, the packet scheduler does not only support per-flow queue, but also support distributing packets to a cable modem (CM) over bonded several channels, which bring difficulties for packet scheduling [2].

Frame-based schedulers, such as Deficit Round Robin (DRR), Preorder-DRR, is extremely efficient compared with the sorted-priority schedulers, such as SCFQ, WF2Q [3-5], but have a complexity O(1). Therefore, it is widely studied and deployed in high-speed network. However, due to the round robin fashion, the flow is backlogged at the end, cause a long delay. Multi-Server Deficit Round Robin (MS-DRR) is proposed to use multiple channels to schedule packet but works like a single channel DRR and transmits the packet in the first available channel [6]. However, MS-DRR deems that a packet can be transmitted on all channels. If the channels are not free, the scheduler will be blocked until one of these channels becomes free. For DOCSIS3.0 packet scheduling, SCFQ are also extended from the aspects of weighted fair sharing of the aggregate capacity for intersecting Bonding Groups. But extended SCFQ have high complexity O (logn) as the same as SCFQ, and n is the number of flows [7]. Therefore, it cannot be applied to practice. The OutQ−DRR and BDRR based on the deficit round robin (DRR) are designed for DOCSIS3.0 [2]. Once a packet arrived, OutQ−DRR immediately distributes it to an output queue associated with a particular channel which is selected to transmit the packet. But delay guarantee for multi-channel scheduling cannot be provided because of the simple stripping distribution policy [2]. For BDRR, the channel which a packet will be transmitted is determined until the packet gets the opportunity to transmit. The queuing position is referred to as input queuing. Although BDRR is latency rate server and have low complexity, the high latency problem remains unchanged.

In this paper, a new scheduling algorithm, Pre-order Bonded Deficit Round Robin (PBDRR), is proposed by introducing the pre-order mechanism for BDRR. The pre-order mechanism utilizes priority queues to reorder the packet transmission sequence [8]. Thus the packets distributed to the output are more evenly among flows and latency can be decreased. This also results in a significant low latency bound, while preserving the low complexity of BDRR.

2. BONDED DEFICIT ROUND ROBIN SCHEDULING

The BDRR scheduler model is described in Fig1. Support there are N service flows contending for M channels in DOCSIS3.0 network. Each channel capacity is C_m and each flow n has a pre-assigned weight w_n and corresponding bandwidth rate r_n . Quantum, which is obtained from $Q_n = w_n Q_{\min}$, is the available transport amount of flow n in a round robin cycle. Each flow is associated with a deficit counter DC_n , which indicates the date amount the flow that can be sent in a round. Because of the bonding group, the flows of a particular CM can access only channels within its bonding group, not all channels in the network.The relationship between flows and bonding group is represented by an N×M structure 0-1 matrix δ_{nm} . Only when $\delta_{nm} = 1$, flow n is allowed to transmit in channel n. The bonding group, which flow n belongs to, consists of M_n channels.According to δ_{nm} and rate partitioning operation, $r_{\mathrm{n}}^{\mathrm{m}}$ and w_{n}^{m} is also pre-assigned for each flow n in channel m, $r_{\mathrm{n}}^{\mathrm{m}} = w_n^m r_{\min}^{\mathrm{m}}$. It is clear

* This work is partially supported by National High Technology Research and Development Program of China #2011AA01A102, the National science and technology support plan under Grant# 2012BAH02B01 and the Chinese academy of sciences key deployment project under Grant #KGZD-EW-103-4.

that $\sum_{m=1}^{M} r_{\mathrm{n}}^{\mathrm{m}} = r_n$. For each flow in this channel, a quantum is also given by $Q_n^m = w_n^m Q_{\min}^m$.

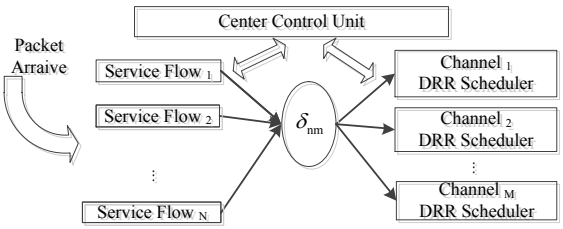

Figure.1 the BDRR scheduling model

As an input queuing algorithm, BDRR keeps one packet queue per flow. In each channel m, there is a separate DRR scheduler with a BackloggedFlows$^{\mathrm{m}}$ list, which is ID list of flows to be serviced. The number of packets in a new arriving flow's queue gradually increases from 0. Meanwhile, DRR scheduler of a relevant channel within corresponding bonding group is notified to add the flow to the tail of BackloggedFlows$^{\mathrm{m}}$, until M_n channels are all notified. According to BackloggedFlows$^{\mathrm{m}}$, channel DRR schedulers traverse all backlogged flows in a round robin order. When visited by DRR, the deficit counter DC_n^m of flow n is increased with its quantum Q_n^m. Then, packets are sent one by one. Accordingly, DC_n^m is decreased by the size of the packets, until 错误!未定义书签。is less than the size of packet. The unused DC_n^m will be still available for the next round. In this way all backlogged flows are serviced. Once the packet number of flow n is less than M_n, the channel scheduler removes the flow n from backloggedFlows$^{\mathrm{m}}$ and DC_n^m is set to 0. When M_n channel schedulers remove the flow n, this flow exits BDRR. The channel scheduler seems separate, but actually they are not independent. They are coordinated by the center control unit, which stores and updates the variables common to all channels, such as notified channels, unused channels, queue status, etc. The common memory pool of the packet queues may be accessed by more than one scheduler. The concurrency visit is also controlled by the center control unit. Due to the input queuing, the select channel function has to go through all channels to find an unused channel under the worst case, which leads to $O(M)$ complexity. Although BDRR is a latency-rate server and the latency is derived, its high average and maximum packet delay remains unchanged because of the inherent drawbacks of the DRR.

3. THE PRE-ORDER BONDED DEFICIT ROUND ROBIN

A. Scheduling module

The proposed pre-order bonded deficit round robin scheduling model is presented in Fig.2. PB-DRR employs a two-level scheduler to serve multi-channel packet scheduling and packets of the same flow n are queued as Service Flow Queue (SFQn) at the entrance of the PB-DRR. The first-level flow scheduler (SFS) is DRR, which distributes packets of SFQn to corresponding channel in round robin fashion. The second-level packet scheduler in each separate channel is priority queue (PQ), which dispatches the packet from the highest queue (PQ1) to the lowest priority queue (PQz) in the physical layer channel. The center control unit is responsible to the coordination of the separate channel scheduler and Quantum updating operation, etc. By introducing the PQ, PBDRR pre-orders the packet transmission sequence in every channel, and the high delay problem of DRR can be overcome.

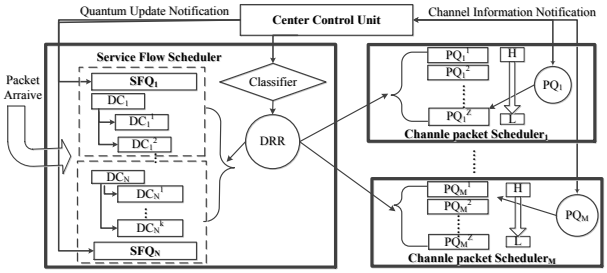

Figure.2 the PBDRR scheduling model

B. Algorithm

A detail description of the implementation of PBDRR is given bellow. Service flow scheduler maintains a BackloggedFlows list, which records the non-empty queues in the current round robin. Each SFQn keeps an ActChList$_n$ for the channels within its bonding group which have been active. Service flow scheduler always serves the head flow of BackloggedFlows ,and sends packets among channels of ActChList$_n$ in turn .Constant and variable definitions are the same as BDRR in Section 2.

Pseudo code of service flow scheduler is given as follows:

ServiceFlowScheduler

Service Flow Queuing Process (SFQP)

```
do while(on arrival of packet p of SFQn)
   Enqueue(p,SFQn);   //Packet p is placed into SFQn
   if  ActChList_n <  PacketNumber(SFQn ) then
   actCh();//Active channels for continuing growing packets
   sendMSG(Classifier, n);  //Inform classifer handle flow n
   endif
endwhile
```

Packet Classify Process(Classifier)

```
while(FOREVER)do
   n=waitMSG();   //wait an channel active signal
   for ∀m  in  ActChList_n
      if  Round_n^m  !=  Round_sys^m  ;   // a new round arrive
         Round_n^m  ←  Round_sys^m ;
         DC_n^m ← MAX( DC_n^m , Quantum_n^m );
      endif
   endfor
   for ∀SF_n  in  BackloggedFlows
```

```
    while( DC_n^m >0 and notEmpty(SFQ_n ))
      pSize← size(SFQ_n .head());
      if(pSize<= DC_n^m ) then
            //Credit is enough to send out a packet
        DC_n^m ← DC_n^m – pSize;
            //Decrease the credit by the packet size
        p← Z- DC_n^m / PQG_p^m ; //compute the priority queue
        Enquene(Dequene(SFQ_n), PQ_p^m );
            //move packet from SFQ_n to PQ_p^m
      if (size( PQG_p^m )=1) then
        MH_Insert(m,p); //Insert PQ_p^m to MH(m)
      endif
    endwhile
  enfor
  if size(SFQn)>0 then //there is unused credit
    Enqueue(n,BackloggedFlows);
    if Size( BackloggedFlows )=1 then SetEvent(EV_actlist);
  else
    if ActChList_n > PacketNumber(SFQn ) then
      sdCh ();// shutdown specific channel for flow n
    endif
  endif
endwhile
```

The classifier in Service Flow Scheduler decides sending packets in this round to the corresponding priority queue PQ_p^m. And PQ_p^m depends on the $ActChList_n$ and the DC_n^m of this packet. Each channel keeps a min heap MH (m), whose root node indicates the highest priory queue. The insert or delete operation for MH (m) is coordinated by the center control unit. In order to ensure at least one packet to be transmitted in its queue, the active or shutdown operation is also controlled by the center control unit.

Pseudo code of packet scheduler is given as follows:

PacketScheduler(m) : **Packet Sending Process(PSP)**

```
while(FOREVER)do
  if MH_Empty(m) then //there are no packet in any PQ_p^m
    Round_sys^m = Round_sys^m +1 ; // a new round start
    EC=WaitEvents(EV_minheap^m, EV_backloggedlist);
//Waiting for new packet arriving and backloggeds update
    if(EV= EV_backloggedlist) then
      do while(BackloggedFlows.size()>1)
        n= Dequeue(BackloggedFlows);
        DC_n^m = DC_n^m + Quantum_n^m ;
            //Accumulate the unused credit last round
        SendMSG(Classifier, n);
      endwhile
    waitEvent(EV_minheap^m);
    endif
  endif
  waitEvent(ServerIdle);
  MH_lock(m); //concurrent control for MH(H)
  if Empty(PQ_MinHeapRoot^m);
    MH_unlock(m);
  Send(Dequeue(PQ_MinHeapRoot^m));
      //set out packet of minimum PQ_p^m
endwhile
```

Each packet scheduler sends packets distributed by flow scheduler. When finishing the round, it updates DC_n^m immediately by $Quantum_n^m$ in the BackloggedFlows for the decisions of next round.

4. PERFORMANCE ANALYSIS

The latency and efficiency, which are the most desirable properties for scheduling discipline, is analyzed below.

A. Delay bound

A general model, called Latency-Rate (LR) servers, is developed for the latency analysis of scheduling algorithms [9]. From the perspective of latency, a flow can obtain upper bounds from the time when it is being backlogged until it is serviced at its guaranteed rate. The LR server is further defined in [10], which is the minimum non-negative constant that satisfies

$$W_n(t_0,t) \geq \max(0, r_n(t-t_0-\theta_n^{scheduler}))$$

Where $W_n(t_0,t)$ is the amount of service received by flow n in the time interval (t_0,t). t_0 is a start of a busy period and t is anytime instance within this busy period . r_n is the reserved rate of service flow n, r_n^m is the reserved rate for flow n in channel m. M_n is the channel set of bonding group for flow n. The priority queue schedulers in each channel are appended to the DRR scheduler. Therefore, PBDRR can also be treated as many single PDRR in each channel.

$$W_n(t_0,t) = \sum_{m\in M_n} W_n^m(t_0,t)$$

$$\geq \sum_{m\in M_n} \max\{0, r_n^m(t-t_0-\theta_n^{PDRR,m})\}$$

$$\geq \sum_{m\in M_n} \max\{0, r_n^m(t-t_0-\max_{m\in M_n}\theta_n^{PDRR,m})\}$$

$$\geq \max\{0, \sum_{m\in M_n} r_n^m(t-t_0-\max_{m\in M_n}\theta_n^{PDRR,m})\}$$

$$=\max\{0, r_n(t-t_0-\max_{m\in Mn}\theta_n^{PDRR,m})\}$$

Then $\theta_n \leq \max_{m\in M_n}\theta_n^{PDRR,m}$

Therefore, PBDRR scheduler belongs to the class of the LR-servers, which has an upper bound on $\max_{m\in M_n}\theta_n^{PDRR,m}$

B. Complexity

Algorithm complexity is determined by operations needed by service flow scheduling and packet scheduling process. For service flow queuing process, each individual packet is inserted to corresponding SFQn, and the active channel is selected randomly. Thus, the complexity is O(1) When a packet redistributed to PQ, as long as $\text{Quantun}_n^m \geq L\max$, more than one packet will be sent out for the queue visited, the complexity is O(1). When SFQn became empty, the flow ID in MH(m) will be deleted. It takes O(logZ) to adjust MH. Thus, the work complexity is O(logZ) with respect to the number of priority queues.

In summary, PB-DRR is a latency-rate server and requires only O(logZ) complexity to process a packet, so it is simple enough to be implemented in hardware.

5. SIMULATION

Simulations are shown in this section to compare the delay performance of PBDRR with BDRR. Simulation program is implemented on OMNET++. There are 4 channels, each channel's capacity is10Mbps, and 16 backlogged flows described in table1. All flows are assumed as constant bit rate (CBR) with fixed packet size. The max packet size is 1518 bytes.

TABLE I Service Flow Description

Flow	Rate	Channel
flow1-flow4	4Mbps	per flow for each channel
flow5-flow8	4Mbps	per flow for each channel
flow9-flow16	1Mbps	2 flow for each channel

The average delay vs. time is shown in Fig. 3. BDRR always has larger delay than PBDRR and suffers high delay jitter. Due to the pre-order, PBDRR has a smaller average delay than BDRR. Both the delay of them has upper limitation，but PBDRR's is lower,because of smaller latency of PBDRR.

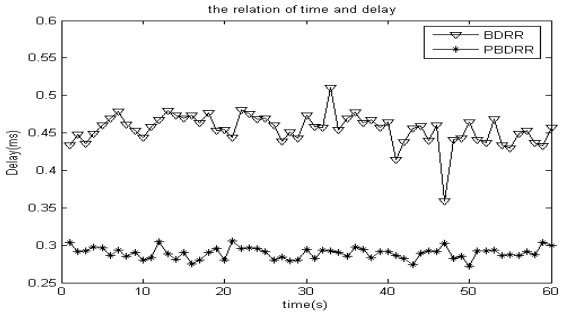

Figure.3 the relation of time and delay

In Fig4, with the increasing of packet size, the delay of both schedulers increases and gap between them (represented by gray colour shading) also increases. While the packet size is large, the the variation is smoother. This is because of large packet in PBDRR performs better than small packet.

Thus, our algorithm could perform better than BDRR by decreasing the large delay.

6 CONCLUSION

This paper presents a novel scheduling algorithm which combines the BDRR with the preorder discipline. Within PB-DRR， we design a scheduler merge all the desirable properties of the BDRR and PDRR algorithms and discards their drawbacks. Theory analysis shows that PB-DRR is a packet scheduling algorithm with O(logZ) complexity and bounded delay. The simulation shows that PB-DRR does not only reduces the scheduling delay time of the flow effectively, but also keeps a lower complexity

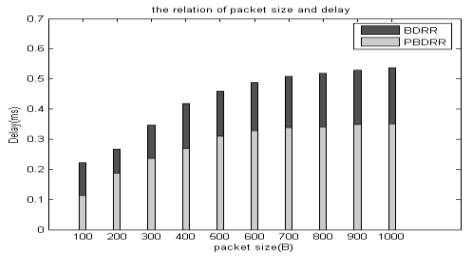

Figure.4 the relation of packet size and delay

Acknowledgment

This work is supported by the National science and technology support plan under Grant No. 2012BAH02B01, the Chinese academy of sciences key deployment project under Grant No.KGZD-EW-103-2, the Chinese academy of sciences key deployment project under Grant No.KGZD-EW-103-4, the National High Technology Research and Development Program of China under Grant No. 2012AA011703..

References

[1]. CableLabs, " Data Over Cable Service Interface Specification (DOCSIS 3.0), ". 2008.

[2]. Nikolova, D. and C. Blondia, " Bonded deficit round robin scheduling for multi-channel networks " ,Computer Networks, vol. 55, no. 15, pp. 3503-3516, 2011.

[3]. Golestani, S.J. "A self-clocked fair queueing scheme for broadband applications". INFOCOM'94. Networking for Global Communications., 13th Proceedings IEEE, pp.636-646, 1994.

[4]. Bennet, J. and H. Zhang. "WF2Q: Worst-case Fair WFQ". Proceedings of IEEE MILCOM1996.

[5]. Shreedhar, M. and G. Varghese. "Efficient fair queueing using deficit round robin". ACM SIGCOMM Computer Communication Review, pp.231-242, 1995.

[6]. Haiming, X. and Y. Jiang, " Analysis of multi-server round Robin scheduling disciplines " ,IEICE transactions on communications, vol. 87, no. 12, pp. 3593-3602, 2004.

[7]. Hong, G., J. Martin, S. Moser, and J. Westall. "Fair Scheduling on Parallel Bonded Channels with Intersecting Bonding Groups". Modeling, Analysis & Simulation of Computer and Telecommunication Systems (MASCOTS), 2012 IEEE 20th International Symposium on, pp.89-98, 2012.

[8]. Tsao, S. and Y. Lin, " Pre-order deficit round robin: a new scheduling algorithm for packet-switched networks " ,Computer Networks, vol. 35, no. 2, pp. 287-305, 2001.

[9]. Stiliadis, D. and A. Varma, " Latency-rate servers: a general model for analysis of traffic scheduling algorithms " ,IEEE/ACM Transactions on Networking (ToN), vol. 6, no. 5, pp. 611-624, 1998.

[10]. Stiliadis, D., Traffic scheduling in packet-switched networks: analysis, design, and implementation. 1996: University of California, Santa Cruz.