PAPER

Two-Phase Minislot Scheduling Algorithm for HFC QoS Services Provisioning

Wei-Ming YIN[†], **Chia-Jen WU**[††], *and* **Ying-Dar LIN**[†††], *Nonmembers*

SUMMARY Data-Over-Cable Service Interface Specifications v1.1 (DOCSIS v1.1), developed for data transmissions over Hybrid Fiber Coaxial (HFC) networks, defines five upstream services for supporting per-flow Quality of Services (QoS). The cable modem termination system (CMTS) must periodically grant upstream transmission opportunities to the QoS flows based on their QoS parameters. However, packets may violate QoS requirements when several flows demand the same interval for transmission. This study proposes a two-phase, i.e., the scheduling sequence determination phase and the minislot assignment phase, minislot scheduling algorithm to reduce the QoS violation rate. In the scheduling sequence determination phase, the flow whose packets are most unlikely to violate QoS is scheduled first. Then, in the minislot assignment phase, the scheduler allocates to a flow the available interval where the likelihood of packet violation is minimum. Simulation results demonstrate that our scheduling algorithm can reduce the QoS violation rate by 80–35% over that of the first-come-first-serve-random-selection algorithm. It increases the utilization by 25% as well. The two-phase minislot scheduling algorithm can work within the DOCSIS v1.1 framework.

key words: *HFC, DOCSIS, upstream, scheduling, QoS*

1. Introduction

CableLabs proposed Data-Over-Cable Services Interface Specifications (DOCSIS) [1] to deploy a high-speed packet-based communications system on Community Antenna Television (CATV) infrastructures. The intended services permit transparent bi-directional transfer of Internet Protocol (IP) traffic over a complete-coaxial or hybrid fiber coaxial (HFC) access network. The topology of the HFC network is a point-to-multipoint, tree-and-branch access network in the downstream direction, but a multipoint-to-point bus-like access network in the upstream direction, as shown in Fig. 1. Being subject to collisions, the shared upstream channel requires an efficient collision avoidance and resolution scheme. Collisions may occur upstream since upstream is a multiple-access media and free accessible to the cable modems (CMs).

Recently, the number of real time applications has been growing, such as voice over IP (VoIP) and teleconference. These applications have critical network transmit delay constraints, implying that a packet would be discarded if not received within the tolerated interval. Therefore, time-critical applications should not use the traditional best effort (BE) service that guarantees neither bandwidth nor access delay. CableLabs defines five upstream services in DOCSIS v1.1: Unsolicited Grant Service (UGS), Unsolicited Grant Service with Activity Detection (UGS-AD), Real-Time Polling Service (rtPS), Non-Real-Time Polling Service (nrtPS), and Best Effort (BE) Service. The CMTS must periodically provide data grants or unicast request opportunities to the flows. When the demands of multiple flows overlap the upstream, the CMTS may be forced to drop some packets, delay or advance their transmission time. This situation is called QoS violation. However, DOCSIS leaves the CMTS scheduling algorithm to be designed by vendors instead of specifying it in the standards. Integrating DiffServ and IntServ into DOCSIS v1.0 framework can be found in [2] and [3], respectively. However, none focuses on DOCSIS v1.1 framework. A novel two-phase minislot scheduling algorithm which considers flow scheduling sequence and minislots assignment is proposed herein to meet the QoS requirements and reduce the QoS violation rate. Simulation results herein confirm that the two-phase minislot scheduling algorithm outperforms the first-come-first-serve-random-select (FCFS-RS) in many aspects.

The remainder of this paper is organized as follows. Section 2 and Sect. 3 present the DOCSIS media access control (MAC) protocols and upstream QoS services, respectively. Then, Sect. 4 introduces the two-phase minislot scheduling algorithm. Subsequently, the simulation results are analyzed in Sect. 5. Finally, Sect. 6 gives a conclusion.

2. DOCSIS MAC Protocol

2.1 Minislot

In the DOCSIS MAC layer, the upstream transmission time-line is divided into fixed-length intervals, minislots, by time division multiple access (TDMA) technologies, as illustrated in Fig. 2. The minislot is the unit of granularity for upstream transmission opportunities. Each minislot is labeled with an integer identifier

---

Manuscript received November 29, 2000.
Manuscript revised June 6, 2001.

[†] The author is with Fastlink Communication Corporation, Taipei, 114, Taiwan.

[††] The author is with ET Home Company, Taipei, 114, Taiwan.

[†††] The author is with the Department of Computer and Information Science, National Chiao Tung University, Hsinchu, 300, Taiwan.

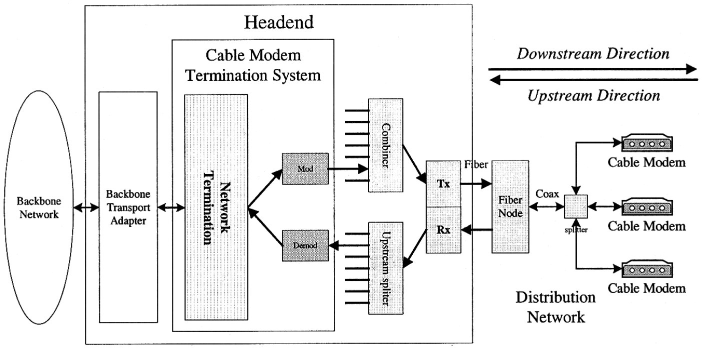

Fig. 1 The topology of an HFC network.

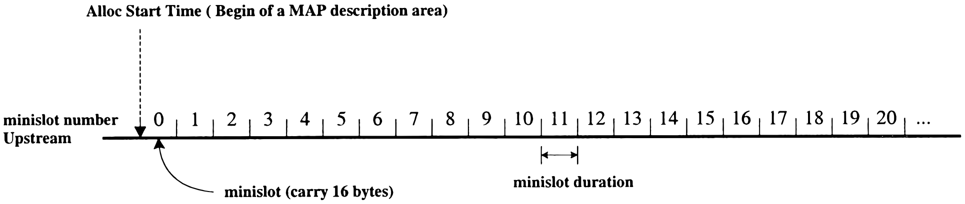

Fig. 2 Upstream minislots.

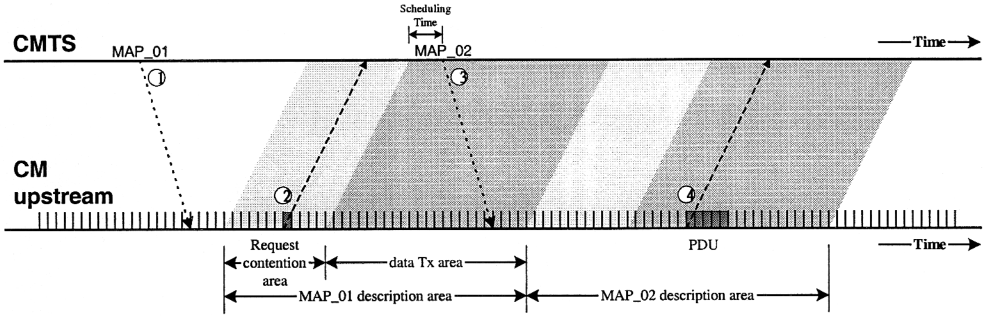

Fig. 3 Normal operation of the DOCSIS MAC protocol.

called minislot number that represents the offset related to the beginning of a MAP description area. The size of each minislot is expected to carry a request PDU, 16 bytes, and the duration of the minislot is the power of two multiples of 6.25 μs.

2.2 Service Flow

A service flow, represented by a Service ID (SID), is mapped to a service class and a virtual queue inside the CM. A CM obtains its SIDs corresponding to the service for which it negotiates with the CMTS during registration or dynamic service establishment. Multiple service flows of single CM are possible if the CM requires several types of service. Whenever the CMTS schedules upstream transmission, it considers each service flow rather than each CM. Therefore, packets from different service flows would acquire different QoS treatments even if they come from the same CM.

2.3 MAC Operation

The use of upstream minislots is centrally controlled by the CMTS to reduce bandwidth wastage due to collisions. Therefore, CMs must transmit small request protocol data units (PDUs), which are subject to collisions, to notify the CMTS when data are backlogged in their virtual queues. In the normal operation, some of the upstream minislots are described as request minislots, and the other minislots are data minislots, presented in Fig. 3. The request minislots, which form the request contention area, are opened for contending access by CMs. A CM may randomly select one minislot in the request contention area to send its bandwidth request. After collecting all the requests, the CMTS has sufficient information about the bandwidth requests of CMs. Then, the CMTS, by running the scheduling algorithm, assigns an appropriate number of data minislots to accommodate the bandwidth re-

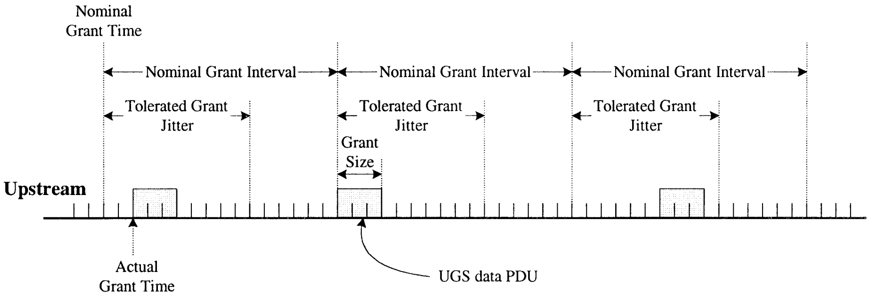

Fig. 4 UGS service.

quests and informs CMs of the scheduling results by an MAP message, which describes the usage of upstream bandwidth. The upstream bandwidth management adopted by DOCSIS is achieved by periodically broadcasting the bandwidth allocation MAP messages on the downstream channels. Therefore, the MAP messages can inform CMs of the bandwidth assignment and let them transmit their data PDUs on the collision-free data minislots.

Notably, collisions may occur when two or more CMs place their request PDUs in the same minislot. CMs can only realize whether of not the request is successful by reading the next MAP message, because they cannot listen directly to the upstream. The collided requests should be retransmitted until they are either successfully received by the CMTS or timed out. A good contention resolution algorithm is responsible for reducing the data access delay, which is the time between the data PDU being generated and successfully transmitted to the CMTS. DOCSIS v1.1 adopts the truncated binary exponential back off [1], [4], [5], like CSMA-CD [6] in Ethernet, as the mandatory method of contention resolution. The major advantages of this algorithm are simplicity, fairness, and efficiency. However, no guarantee of data access delay exists because of unknown contention resolution time. Performance of collision resolution algorithms over HFC networks is further studied in [7], [8].

For real time applications, data is useless if it does not arrive at the receiver in time. To reduce the access delay, the request contention process should be shortened or even bypassed. Therefore, DOCSIS defines five upstream services that can obtain bandwidth without request contention. These services are introduced below.

3. DOCSIS v1.1 Upstream Scheduling Services

Five upstream scheduling services designed for real time or high bit rate applications exist, including UGS, UGS-AD, rtPS, nrtPS, and BE. Except for BE, these services avoid request contention by an unsolicited grant or polling. Unsolicited grants, opportunities for collision-free data transmission automatically issued by the CMTS, allow CMs to transmit their PDUs without bandwidth requests. The polling service provides collision-free request opportunities, termed unicast requests. The bandwidth demands in unicast requests inform the CMTS so that access delay in polling is guaranteed.

3.1 Unsolicited Grant Service (UGS)

UGS is defined to support CBR data transmission, such as audio streams, over the upstream channels. The CMTS provides fixed size data grants at periodic intervals to the UGS flows after the CM specifies the QoS parameters during registration. Since the bandwidth is reserved without request contention, the UGS can guarantee both bandwidth and data access delay. Four QoS parameters are mandatory for the UGS. As displayed in Fig. 4, the *Nominal Grant Interval* is specified according to the interval between packets being generated of the CBR flow. Hence, whenever a packet is generated, a corresponding data grant is made for it. The *Unsolicited Grant Size* represents packet size, and the *Tolerated Grant Jitter* is the maximum tolerated data access delay. The CMTS should grant data transmission at a nominal grant time, but it can defer the actual grant time within the *Tolerated Grant Jitter* if necessary. If the flow specifies a small *Tolerated Grant Jitter*, the data reassemble buffer may be minimized and the performance of the real time application is enhanced. However, this flow risks losing data because it increases the scheduling constraints on the CMTS. If the CMTS cannot find any available transmission time, the CM discards the data PDU. Therefore, specifying the Tolerate Grant Jitter is a trade off. This issue is discussed in Sect. 4 based on our simulation results.

3.2 Real Time Polling Service (rtPS)

For variable bit rate (VBR) traffics, such as MPEG streams, the CMTS cannot estimate the bandwidth demands because of the burst nature of VBR flows. Therefore, this traffic is more appropriate for real-time Polling Service (rtPS) than for UGS. In this service, the CMTS provides periodic opportunities for unicast request, namely polling, for the specific CM, and the

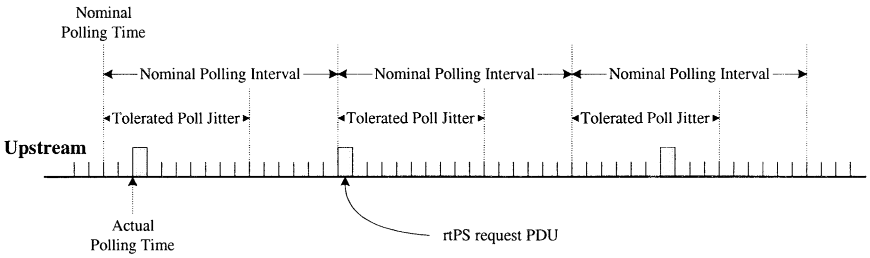

Fig. 5 rtPS service.

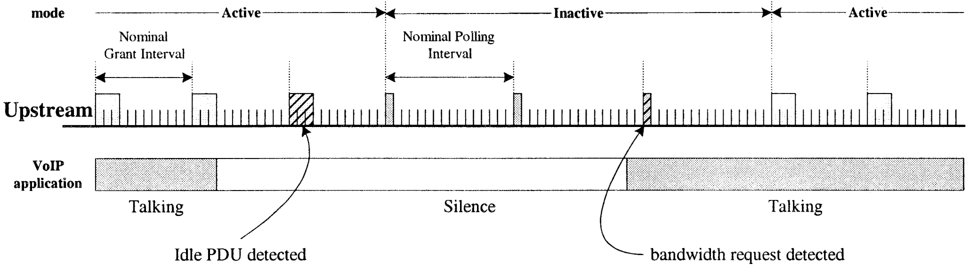

Fig. 6 UGS-AD mode switching.

CM can report the bandwidth demands through these opportunities. Notably, the unicast request is collision-free, so the request is guaranteed to be received by the CMTS in time. As shown in Fig. 5, the parameters of rtPS are very similar to those of UGS. *Nominal Polling Interval* specifies the interval of unicast request opportunities, while *Tolerated Poll Jitter* specifies the maximum deferment of a request. Because the request size is invariably 16 bytes, and only one outstanding request is allowed for one service flow, there is not necessary to define polling size and polling per interval as in UGS.

3.3 Unsolicited Grant Service with Activity Detection (UGS-AD)

For some audio compression technologies such as ITU G.728 [9], silence suppression is supported, and so there is no data during silence. UGS is not very suitable for this traffic because the upstream bandwidth is largely wasted if the CMTS grants data transmission during the silent period. To reduce bandwidth wastage of such flows, DOCSIS defines Unsolicited Grant Service with Activity Detection (UGS-AD), which is with a combination of UGS and rtPS. Figure 6 indicates that, during conversation the service is in active mode and so the CMTS periodically grants fixed size data transmission opportunities to the CM. Meanwhile, during the intervals of silence, there is no data to transmit, and thus the CMTS detects some data grants are unused and switches the flow status into inactive. During the inactive period, the CMTS polls the CM by periodically giving unicast requests, like rtPS. As soon as request PDU is detected from the CM, the CMTS realizes that the conversation has resumed and switches the flow status back to active. The activity detection algorithm is flexible to vender design because it is not involved in communication. Notably, the request is only used for activity detection in UGS-AD. Therefore, the CMTS neglects the bandwidth demands in the request PDU and resumes allocating *Grant Size* grants to the CM.

3.4 Non Real Time Polling Service (nrtPS)

Non real time polling service (nrtPS) closely resembles rtPS, but has a longer polling interval, around 1 sec or more. Therefore, nrtPS flow can use both unicast request opportunities and broadcast request opportunities. The default value of the *Nominal Polling Interval* is supplied by the CMTS, but the CM can specify an alternative value. This service is designed for high-speed data transmission, such as the high bit rate file transfer protocol (FTP).

3.5 Best Effort (BE)

The conventional service provided in the previous version of DOCSIS is best effort (BE). In this service, the CM generally uses contention request opportunities for bandwidth demands. The BE flow can still acquire unicast request opportunities, but only infrequently because the CMTS polls BE flows only when the load of the HFC network is relatively light. BE flows can also directly transmit their data PDUs in the request/data contention area. In conclusion, BE flows have no QoS guarantee, and so are only suitable for conventional applications such as Telnet or the World Wide Web (www).

Table 1 lists significant QoS parameters for each

Table 1 QoS parameters for DOCSIS upstream scheduling services.

	UGS	**rtPS**	**UGS-AD**	**nrtPS**	**BE**
Nominal Grant Interval	**mandatory**	optional	**mandatory**	N/A	N/A
Tolerated Grant Jitter	**optional CMTS-specific**	optional	**optional CMTS-specific**	N/A	N/A
Unsolicited Grant Size	**mandatory**	N/A	**mandatory**	N/A	N/A
Grants per interval	**mandatory**	optional	**mandatory**	N/A	N/A
Nominal Polling Interval	N/A	**mandatory**	**optional**	**optional CMTS-specific**	N/A
Tolerated Poll Jitter	N/A	**optional CMTS-specific**	**optional CMTS-specific**	N/A	N/A

variety of service. In summary, three upstream services exist that are suitable for real time applications, UGS, rtPS, and UGS-AD. More precisely, UGS is suitable for CBR traffic, rtPS is suitable for VBR traffic, and UGS-AD is suitable for CBR traffic with the On-off model. Two services exist, that are suitable for non-real-time applications, nrtPS and BE. NrtPS provides more request opportunities, and so that it is suitable for high bit rate applications. Meanwhile, the BE service is used for general applications. The following section introduces how the scheduling algorithm deals with various types of flows.

4. Two-Phase Minislot Scheduling Algorithm

Whenever the request contention area ends in a MAP description area as depicted in Fig. 3, the headend collects successful requests and then runs scheduling algorithm to assign data minislots. Thereafter, the headend constructs a MAP to describe the usage of next MAP description area.

4.1 Motivation

To meet QoS requirements for time critical flows, the minislot scheduler should consider bandwidth requirements as well as time constraints. The time constraints could be mapped into time intervals, i.e., *satisfying regions*, where the packets must be transmitted. If the CMTS arranges data grants within corresponding satisfying regions, the scheduling results would meet the QoS requirements. However, the QoS requirements may be violated owing to overlapped satisfying regions, a phenomenon when several flows simultaneously demand the overlapped area. Therefore, the minislot cost mechanism is proposed herein to estimate the probability of QoS violation, and the estimated cost is used for minislot assignment to reduce QoS violations.

4.2 Satisfying Region

Since different flows use different QoS parameter sets to describe their bandwidth requirements, a unified QoS parameter set is required to achieve simplicity and a manageable scale in scheduling bandwidth. Therefore, the unified QoS parameter set, measured in minislots, is defined herein to describe a time critical flow. The unified QoS parameter set employed herein contains four parameters: I, J, S, and G indicate the PDU generating interval in minislots, the maximum tolerated jitter in minislots, the PDU size in minislots, and the number of PDUs being sent in a satisfying region, respectively. For most time critical flows, fix-sized PDUs are periodically generated so that the bandwidth requirement of a QoS flow, BW, can be represented as

$$BW = \frac{MPS}{I} \times (S \times \text{minislot\_size} \times 8), \tag{1}$$

where MPS denotes the number of minislots per second and the minislot_size is 16 bytes in the DOCSIS system. The access delay of a PDU, D, is guaranteed to be less than J as

$$D \leq J \times \text{minislot\_duration}, \tag{2}$$

where minislot_duration is the power of two multiples of $6.25\,\mu$s in the DOCSIS system. Meanwhile, the scheduler should grant G PDUs in a satisfying region. The satisfying regions of a QoS flow are specified by I, J, and S. Notably, the CMTS should assign the starting minislot number, SMN, of the first satisfying region so that the successive satisfying regions can be located.

Figure 7 shows the relationship of the DOCSIS parameters and the satisfying regions for UGS or active UGS-AD flows. For example, $SMN = 2$ and $J = 16$ reveal that minislot$_2$ to minislot$_{17}$ are within the satisfying region. Meanwhile, $S = 4$ implies that the data PDU requires 4 minislots for transmission, and $G = 1$ means only one data PDU exists per satisfying region. The following formulas provide the translations between the DOCSIS parameters and the satisfying regions:

$$I = \left\lceil \frac{\text{Nominal Grant Interval}}{\text{minislot duration}} \right\rceil,$$

$$S = \left\lceil \frac{\text{Unsolicited Grant Size}}{\text{minislot size}} \right\rceil,$$

$$J = \left\lceil \frac{\text{Tolerated Grant Jitter}}{\text{minislot duration}} \right\rceil + S, \quad \text{and}$$

$$G = \text{Grants per Interval}.$$

For rtPS or inactive UGS-AD, since a request PDU is the size of a single minislot, S is invariably one. Furthermore, G is also invariably one because the CMTS can only poll once per interval. Figure 8 illustrates the relationships between the DOCSIS rtPS parameters and satisfying regions. The unified parameters can

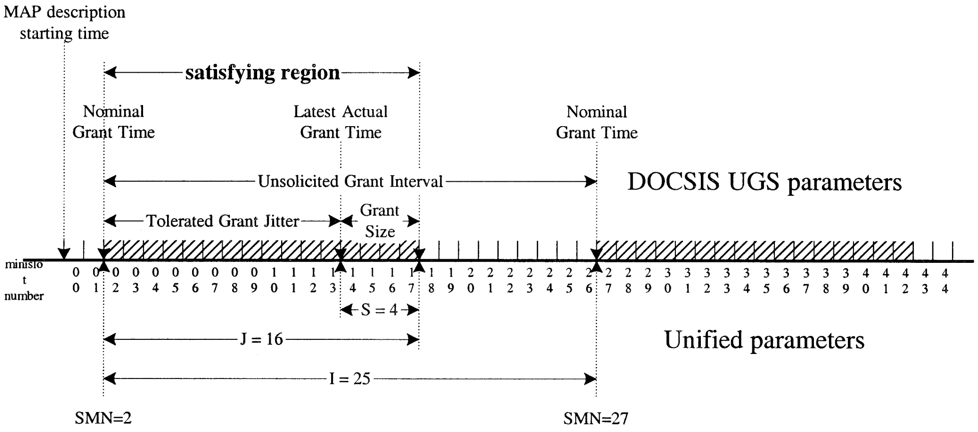

Fig. 7 The relationship between DOCSIS UGS parameters and unified parameters.

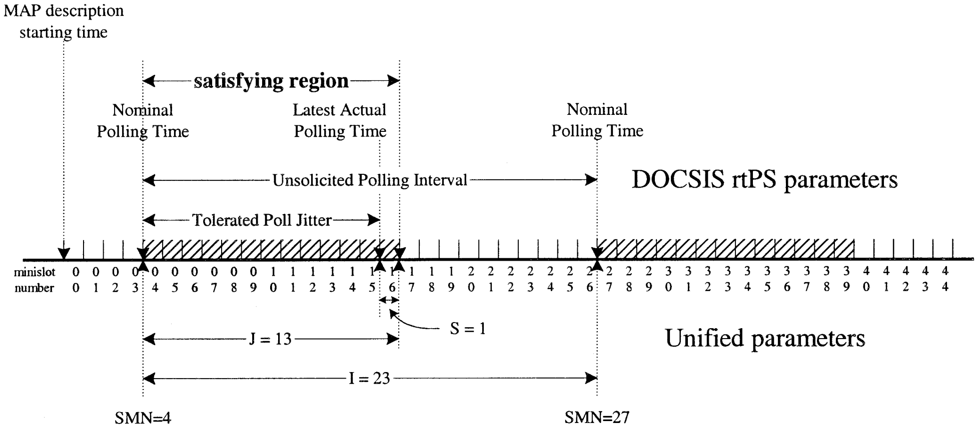

Fig. 8 The relationship between DOCSIS rtPS parameters and unified parameters.

be derived as:

$$I = \left\lceil \frac{\text{Nominal Polling Interval}}{\text{minislot duration}} \right\rceil,$$

$$J = \left\lceil \frac{\text{Tolerated Poll Jitter}}{\text{minislot duration}} \right\rceil + 1,$$

$$s = 1, \quad \text{and}$$

$$G = 1.$$

Notably, two sets of parameters exist for UGS-AD, one for the active mode and the other for the inactive mode. These parameters must be switched when the flow switches to the other mode. Meanwhile, no limitation on PDU transmission delay exists for nrtPS and BE flows, and thus these flows have no satisfying region.

4.3 QoS Violation

Overlapping of satisfying regions implies that the bandwidth demands from different flows may be conflicting. Since hundreds of active flows exist, more than one flow may demand the same minislots. For example in Fig. 9, the satisfying regions among flow_A, flow_B, and flow_D overlap at minislot_4, minislot_5, and minislot_6. The scheduler should decide which flow to prioritize. Suppose minislot_4 to minislot_7 are allocated to flow_A, then the scheduler can find no available interval for flow_D. The PDU of flow_D must be discarded, a situation called QoS violation. If a flow suffers serious QoS violations, considerable quantities of data can be lost. Obviously, the later the scheduler serves a flow, the fewer transmission chances are provided. Hence, the policies for determining the flow scheduling sequence and assigning minislots are crucial in reducing QoS violation rate.

A two-phase minislot scheduling algorithm is proposed herein to meet the QoS requirements of each flow as well as reduce the QoS violation rate. The scheduling algorithm comprises three parts: minislot cost setup, scheduling sequence determination phase, and minislot assignment phase. The following three sections detail the algorithm.

4.4 Minislot Cost Setup

To run the scheduling algorithm, the cost of each minis-

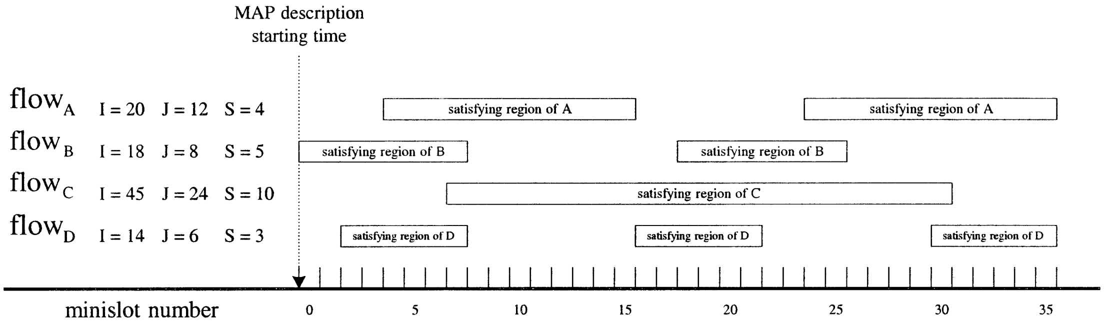

Fig. 9 Overlapping of satisfying regions.

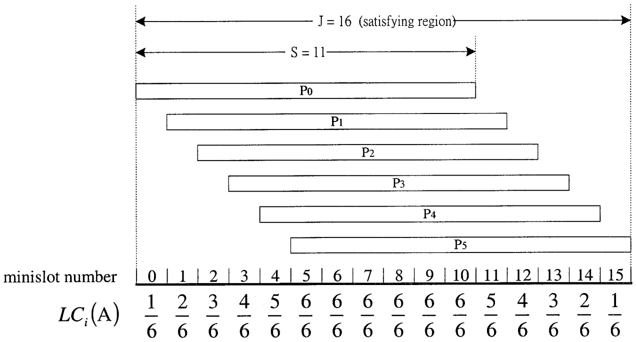

Fig. 10 Local cost example.

lot must be evaluated. Cost setup comprises two steps, local cost setup and global cost setup. The *local cost*, $LC_i(k)$, is the probability of minislot$_i$ being occupied by flow$_k$ and is evaluated flow-by-flow as

$$LC_i(k) = \begin{cases} \frac{\min(i - SMN_k + 1,\ \min(S_k, J_k - S_k + 1))}{J_k - S_k + 1}, \\ \quad \text{if } 0 \le i - SMN_k < \frac{J_k}{2} \\ \frac{\min(SMN_k + J_k - i,\ \min(S_k, J_k - S_k + 1))}{J_k - S_k + 1}, \\ \quad \text{if } \frac{J_k}{2} \le i - SMN_k < J_k \\ 0, \quad \text{otherwise} \end{cases} \tag{3}$$

where SMN_k represents the starting minislot of the satisfying region of flow$_k$, J_k denotes the size of satisfying region, and S_k is the grant size. For example, consider a UGS flow$_A$ with parameters $I = 200$, $J = 16$, $S = 11$, and $G = 1$, as shown in Fig. 10. In this satisfying region, the scheduler has six possible choices labeled from P_0 to P_5. Since minislot$_{13}$ can be assigned to P_3, P_4, and P_5, its local cost, $LC_{13}(A)$, is 3/6. Higher $LC_i(k)$ implies that minislot$_i$ is more needed by flow$_k$.

To measure how strong a minislot is needed by all flows, the *global cost* of minislot$_i$, GC_i, is defined as the *maximal* local cost among flows demanding minislot$_i$, as

$$GC_i = \max_{k \in F}(LC_i(k)), \tag{4}$$

where F is the set of flows demanding minislot$_i$. For example from Fig. 11, the global cost of minislot$_{12}$ is

$$GC_{12} = \max_{k \in \{A,B,C\}}(LC_{12}(k)) = LC_{12}(A) = \frac{6}{6},$$

which means this minislot will be occupied by a flow with probability 1. Higher global cost implies that the corresponding minislot is more likely to be occupied.

4.5 Scheduling Sequence Determination Phase

Since each flow has its bandwidth and QoS requirements, different scheduling sequences lead to different minislots allocation, which results in diverse performance in terms of the QoS violation rate and minislot utilization. Therefore, the scheduler should consider traffic priority and QoS violation rate when determining the scheduling sequence.

The QoS in DOCSIS v1.1 can be divided into three classes. The first class of QoS provides not only bandwidth but also access delay guarantees, such as UGS and UGS-AD. The second class of QoS, rtPS is most typical, provides request access delay guarantee only. The third class of QoS, nrtPS and BE, provides no guarantee, but does provide additional unicast request opportunities under a light network load. Obviously, the first class services should have a higher priority than the second class services, and the second class is higher than the third. Restated, the scheduler should serve flows according to their priorities.

The scheduler determines the scheduling sequence of flows with the same priorities by defining a sequence estimator for each flow that reflects the probability of QoS violation. The *sequence estimator* of flow$_k$, α_k, can be calculated as

$$\alpha_k = \frac{\sum\limits_{i \in SR_k} GC_i}{|SR_k|}, \tag{5}$$

where SR_k indicates the set of minislots within the satisfying region of flow$_k$. The sequence estimator of a flow can be viewed as how strong the minislots of its satisfying region are needed by all flows. If a flow with a higher sequence estimator is scheduled earlier, it might immediately block many other flows that also need the sim-

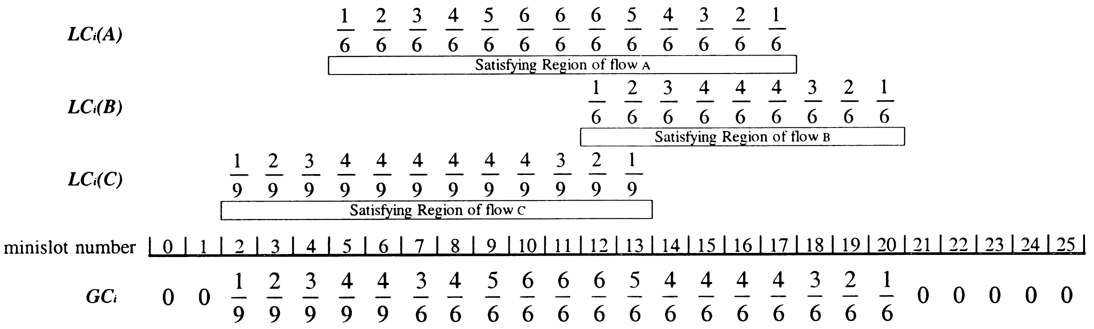

Fig. 11 Global cost example.

ilar set of minislots. Therefore, it has better to schedule flows in the increasing order of sequence estimator. For example, in Fig. 11, the sequence estimator of flow A is (4/9+4/9+3/6+3/6+4/6+5/6+1+1+1+5/6 +4/6+4/6+4/6+4/6)/13=0.722. Similarly, the sequence estimators of flow B and C are 0.611 and 0.616, respectively. Therefore, the scheduling sequence is B-C-A.

4.6 Minislot Assignment Phase

For each flow, the scheduler should select an available interval of S minislot, in which none of the minislots has been occupied, in the flow's satisfying regions. The *assignment estimator* of an available interval of flow$_k$, $\beta_{k,m}$, is calculated to measure the probability of having QoS violation in this interval and is defined as

$$\beta_{k,m} = \frac{\sum_{i=m}^{m+S_k-1} GC_i}{S_k}, \tag{6}$$

where m denotes the starting minislot number of the interval. A high assignment estimator indicates that there is high demand of that interval. Consequently, allocating that interval first might block other flow's demand. Accordingly, the minislot assignment policy used herein is to select an available interval with *minimal* assignment estimator in the corresponding satisfying region, which leaves as many transmission opportunities as possible for other flows. Therefore, the QoS violation rate could be reduced. For example, in Fig. 11, for assigning minislots to flow B, there are six choices. The first assignment estimator is (1+5/6+4/6+4/6)/4 = 0.792, and the others are 0.708, 0.667, 0.625, 0.542, and 0.417, respectively. Therefore, the scheduler assigns the first four consecutive minislots, i.e. minislot$_{12}$ to minislot$_{15}$, to flow B.

4.7 Algorithm Summary

The two-phase minislot scheduling algorithm comprises minislot cost setup and two scheduling phases, the scheduling sequence determination phase and the minislots assignment phase, as shown in Fig. 12. In minislot cost setup, the scheduler first calculates the local cost of minislots for each QoS flow, and then determines the global cost of each minislot by selecting the maximal local cost. In scheduling the sequence determination phase, the scheduler evaluates the sequence estimator of each flow via the global costs. The scheduler determines flow sequence according to the increasing order of sequence estimator. Finally, in the minislot assignment phase, for each flow, the scheduler allocates the interval with the minimal assignment estimator that is also derived from the global costs and writes the scheduling results in the DOCSIS downstream MAP message.

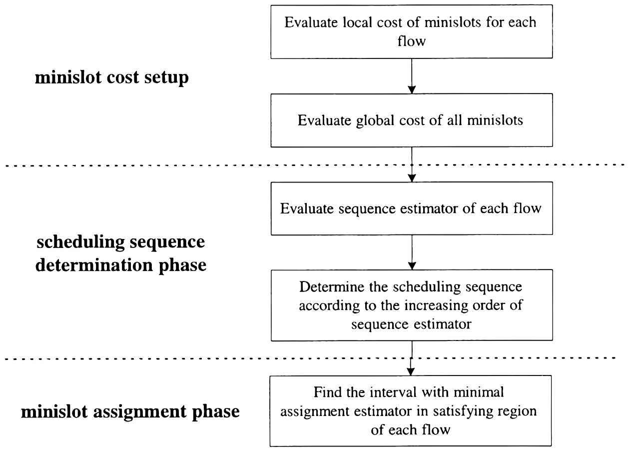

Fig. 12 Two-phase minislot scheduling algorithm.

The time complexity of the proposed algorithm is proportional to the number of flows and the total size of overlapping areas. We might pre-construct a table indexed by (J, S) in which each entry is a local costs list to the corresponding satisfying region. The time to obtain the local costs could thus be reduced by looking up that table. Most of scheduling time is to calculate the assignment estimators. It is possible to calculate all assignment estimators in a satisfying region with referring each minislot only once. Therefore, the time complexity could be significantly reduced.

5. Simulation Results

5.1 Models

Network model

The two-phase minislot scheduling algorithm is compared with the First-Come-First-Serve-Random-Select (FCFS-RS) algorithm with respect to violation rate and minislot utilization. FCFS-RS applies FCFS discipline in the scheduling sequence determination phase. Meanwhile, in the minislot assignment phase, the FCFS-RS scheduler randomly selects an available interval in the satisfying region for each QoS flow. Determining which phase improves the performance most is also significant. Therefore, FCFS-RS is also compared with PhaseI-RS and FCFS-PhaseII. PhaseI-RS follows the policy in the two-phase minislot scheduling algorithm to determine the scheduling sequence, and randomly selects the available interval. On the other hand, FCFS-PhaseII applies FCFS scheduling sequence and adopts the minislot assignment policy in the two-phase minislot scheduling algorithm. Finally, the effects of QoS parameters, including packet inter-arrival time and tolerated jitter, on the two-phase minislot scheduling algorithm proposed herein are discussed. Packets violating QoS requirements are considered “dropped” in our simulation. Table 2 lists the parameters of the simulation environment where the scheduling range stands for the size of the MAP description area.

Traffic model

The comparisons made herein focus only on the time critical flows, i.e. UGS flows and rtPS flows. In addition, our simulation does not consider piggyback requests. Table 3 lists the sources of UGS flows. Since UGS is suitable for CBR flows, such as VoIP applications, speech audio streams are used herein as the sources of UGS flows. The duration of a communication link is exponentially distributed [10]. The average active duration is 180 seconds, and the average inactive duration is 600 seconds. The simulation involved four CODECs for telephony quality of audio streams, and which CODEC is adopted by a newly activated flow is randomly determined. Therefore, the pattern of satisfying regions combination is near randomly distributed. The packet size is limited to between 64 bytes and 512 bytes so that the range of packet inter-arrival time in milliseconed can be derived from

$$\text{packet inter–arrival time} = \frac{1000 \times 8 \times \text{packet size}}{\text{bit rate}}. \tag{7}$$

Table 2 Simulation environment.

parameter	value
upstream channel capacity	10.24 Mbps
minislot size	16 bytes
minislot duration	12.5 μs
minislots/second	80,000
scheduling range	2,000 minislots
number of CMs	2,000

5.2 Numerical Results

Delay and delay jitter

Note that the scheduler allocates minislots every grant interval to a VoIP flow. If the scheduler fails to allocate minislots in a satisfying region of that flow, the corresponding data will be dropped since it is absolute. Therefore, for any transmitted data, the delay is bounded under this kind of scheduling algorithms. Regarding delay jitter, since the considering scheduling algorithms assign available minislots to a flow within its satisfying region or drop the data if there is no available minislots, the delay jitter certainly meets the QoS requirement. Consequently, these four scheduling algorithms lead to close performance in terms of delay and delay jitter. However, they are differentiated in terms of the QoS violation rate and minislot utilization.

QoS violation rate

The QoS violation rate is defined as the amount of dropped packets to the total amount of bandwidth requirements. Figure 13 presents the QoS violation rate with respect to QoS load. It reveals that when the QoS load is 1, the QoS violation rate of the FCFS-RS algorithm is approximately 40.4%, while that of the two-phase minislot scheduling algorithm is about 26.4%. A higher violation rate implies that the QoS flows suffer from more serious data loss. The violation rate of the two-phase minislot scheduling algorithm is lower than that of other algorithms because the scheduler assigns, based on sequence and assignment estimators, minislots that are most unlikely demanded flow-by-flow to reduce the QoS violation rate as much as possible. If we change the tolerated jitter from 0.5–5 ms to 0.5–10 ms, the violation rate is significantly reduced to only 3%. This is because larger tolerated jitter results in larger satisfying region; the scheduler thereby has more flexibility to assign minislots. In other words, if minislots are allowed to be assigned earlier or later to the satisfying

Table 3 UGS flow parameters.

CODEC	bit rate	packet arrival interval (ms)	tolerated jitter (ms)	mean active duration (s)	mean inactive duration (s)
G711	64kbps	12.5 ~ 60	0.5 ~ 5	180	600
G721	32kbps	20 ~ 100	0.5 ~ 5	180	600
G722	56kbps	20 ~ 60	0.5 ~ 5	180	600
G728	16kbps	40 ~ 125	0.5 ~ 5	180	600

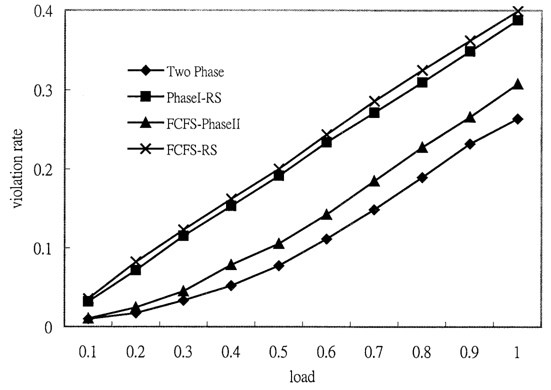

Fig. 13 QoS violation rate.

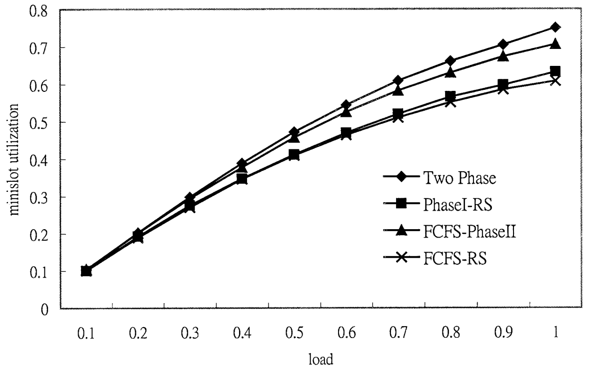

Fig. 14 Minislot utilization.

region, the violation rate could be improved.

Minislot utilization

Figure 14 shows the minislot utilization of each scheme with respect to load. It indicates that when the load is 1, the minislot utilization of the two-phase minislot scheduling algorithm is about 75% while that of the FCFS-RS is about 60.8%. From previous subsection, higher QoS violation rate, leads to more dropped packets, which will directly cause lower minislot utilization. On the other hand, lower QoS violation rate results in higher minislot utilization. Therefore, our proposed scheme outperforms other schemes.

Phase improvement

To compare each scheduling phase with FCFS-RS, the improvement for scheme S, γ_s, is calculated as

$$\gamma_s = \frac{VR_s - VR_{FCFS-RS}}{VR_{FCFS-RS}}, \tag{8}$$

where VR_s denotes the violation rate of scheme S. Figure 15 illustrates the improvement rate of each scheme under various QoS loads. The maximum improvement rate of the two-phase minislot scheduling algorithm is about 78.9% when the QoS load is between 0.2. Notably, the major improvement comes from the minislot assignment phase. For example, Fig. 15 displays that when the QoS traffic load is 1, the improvement rate of the scheduling sequence phase is only 4.6%, while that of the minislot assignment phase is around 23.8%. Since the mean size of satisfying regions, 2.3 ms (about 184 minislots), is much larger than the mean grant size, 280 bytes (about 18 minislots), it renders many minislot assignment choices that neutralize the effect of scheduling sequence. However, assigning right minislots leaves more available minislots to other flows. This is why the major improvement comes from the minislot assignment phase.

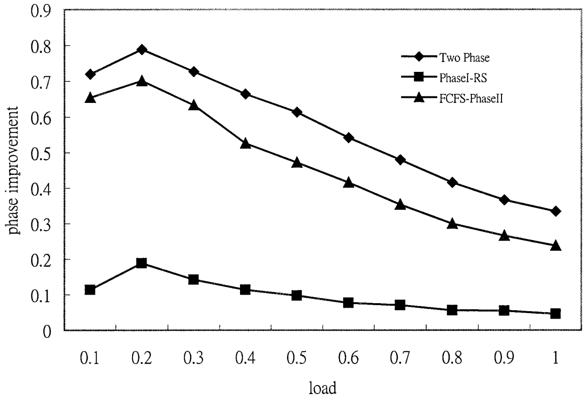

Fig. 15 Phase improvement.

Packet inter-arrival time

For a UGS flow, the *Nominal Grant Interval* is specified as the packet inter-arrival time. The influence of packet inter-arrival time on the scheduling results is interesting. In this experiment, the QoS traffic load is fixed at 0.8 and the packet inter-arrival time of the UGS flows is controlled from 10 ms to 100 ms. Figure 16 displays that a flow with a longer packet inter-arrival time suffers from a higher QoS violation rate. From the figure, when the packet inter-arrival time is 10 ms, the average QoS violation rate of the G.711 flow is 11.4%. However, the QoS violation rate increases to 34.7% when the packet inter-arrival time is 100 ms. This phenomenon occurs because the packet size increases with packet inter-arrival time and larger packet sizes reduces flexibility in assigning minislots. Therefore, we recommend that the packet inter-arrival time of a UGS flow should be minimized; however, it renders higher computing complexity since the number of satisfying regions increases. More consecutive minislots probably leads to higher assignment estimator; therefore, given the same inter-arrival time, G.711 flow undergoes higher violation rate than G.728 flow owing to more consecutive minislots needed.

Tolerated jitter

The tolerated jitter is an essential performance parameter of real time applications, especially for real time audio. For example in telephony, the tolerated jitter should not exceed 24 ms. In the two-phase minislot scheduling algorithm presented herein, the tolerated jitter affects the size of the satisfying region. Flows with smaller jitter acquire fewer transmission opportunities

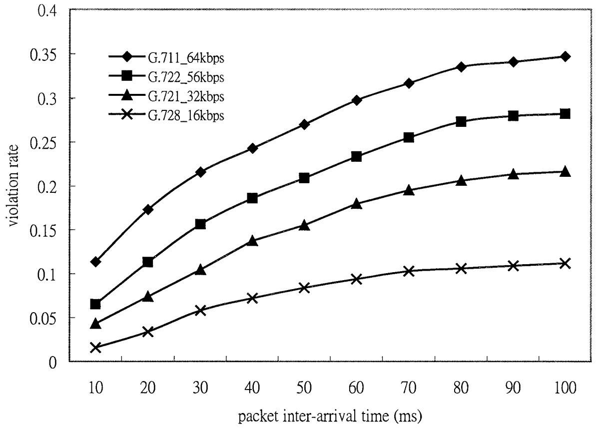

Fig. 16 Violation rate of UGS flows under different packet inter-arrival times.

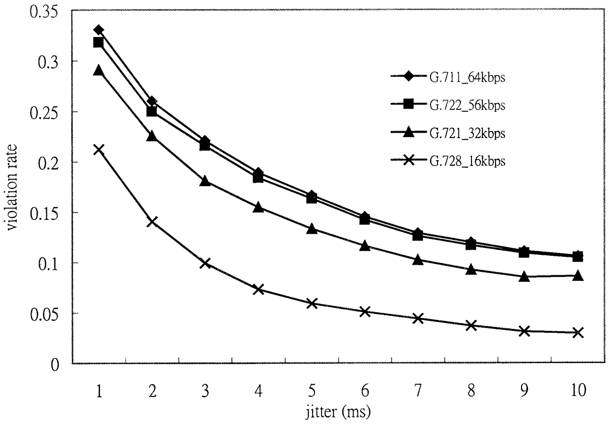

Fig. 17 Violation rate of UGS flows with different tolerated jitter.

since the satisfying region is relatively small. In this experiment, the tolerated jitter is controlled from 1 ms to 10 ms. Figure 17 displays that flow with a larger tolerated jitter has a lower QoS violation rate because the larger satisfying region provides more minislots assignment choices. Therefore, to reduce the QoS violation rate, we believe that tolerated jitter should be maximized. However, it also brings about higher computing complexity since the size of satisfying region is larger.

6. Conclusions

This study introduces the operation of MAC protocol and upstream QoS services in DOCSIS v1.1. To ensure simplicity and a manageable scale, the unified QoS parameter set, which can be transformed from DOCSIS QoS parameters, was proposed to describe the QoS requirements of flows. Meanwhile, the satisfying region is derived from the unified QoS parameter set to indicate the QoS guaranteed transmission time interval for each flow. The local cost and global cost of each minislot are derived to estimate the probability of having QoS violation in the minislot. A novel two-phase minislot scheduling algorithm presented herein, which includes the scheduling sequence determination phase and the minislot assignment phase, is based on the satisfying region so that the scheduling results are guaranteed to meet the QoS requirements of each flow. To reduce the QoS violation rate, the scheduling sequence policy is to first schedule the flow with the minimal sequence estimator derived from Eq. (5), while the minislot assignment policy is to select the available interval with minimal assignment estimator obtained from Eq. (6).

Simulation results demonstrate that the two-phase minislot scheduling algorithm outperforms the FCFS-RS scheduling algorithm in terms of QoS violation rate and minislot utilization. Additionally, three observations are made as follows: First, minislot assignment is more important than scheduling sequence. In other words, once a well-designed minislot assignment scheme is applied, the QoS violation rate is significantly reduced regardless of scheduling sequence. Second, the packet inter-arrival time of a UGS flow should be decreased to reduce the violation rate; however, it renders higher computing complexity since the number of satisfying regions increases. Third, the tolerated jitter of a UGS flow could be increased, if allowed, to reduce the violation rate; however, it also brings about higher computing complexity since the size of satisfying region is larger.

Regarding time complexity, the proposed algorithm needs more computing power owing to the calculation for costs and estimators than FCFS-RS does. However, failing to take minislot popularity into account, the FCFS-RS leads to higher violation rate and inefficient bandwidth usage. We briefly introduce mechanisms to reduce the time complexity of two-phase scheduling algorithm. The detailed and further enhancing work is on going.

How to locate the initial satisfying region for a new flow is not discussed in this study. Theoretically, if the scheduler properly assigns the SMN parameter, the offset of the initial satisfying region which is now randomly selected, for each flow to evenly interleave satisfying regions of various flows, the overlapping of satisfying regions would be decreased and the QoS violation rate therefore might be reduced.

References

[1] Cable Television Laboratories, Data-Over-Cable Service Interface Specifications—Radio Frequency Interface Specification v1.1, July 1999.
[2] N. Golmie, F. Mouvaux, and D. Su, "Differentiated services over cable networks," Proc. GLOBECOM99, Rio de Janeiro, Brazil, Dec. 1999.
[3] R. Rabbat and K.Y. Sun, "QoS support for integrated services over CATV," IEEE Commun., vol.37, no.1, pp.64–68, Jan. 1999.
[4] Y.D. Lin, C.Y. Huang, and W.M. Yin, "Allocation and scheduling algorithms for IEEE802.14 and MCNS in hybrid

fiber coaxial networks," IEEE Trans. Broadcast., vol.44, no.4, pp.427–435, Dec. 1998.

[5] N. Golmie, F. Mouveaux, and D. Su, A Comparison of MAC Protocols for Hybrid Fiber Coax Networks: IEEE802.14 vs. MCNS, IEEE Commun., April 1999.

[6] D.E. Comer, Internetworking with TCP/IP, 2nd ed., pp.23, 484, Prentice Hall, 1991.

[7] Y.D. Lin, W.M. Yin, and C.Y. Huang, "An investigation into HFC MAC protocols: Mechanisms, implementations, and research issues," IEEE Commun. Surveys, vol.3, no.3, 3rd Quarter 2000, http://www.comsoc.org/pubs/surveys/.

[8] W.M. Yin and Y.D. Lin, "Statistically optimized minislot allocation for initial and collision resolution in hybrid fiber coaxial networks," IEEE J. Sel. Areas Commun., vol.18, no.9, Sept. 2000.

[9] F. Fluckiger, Understanding networked multimedia, Prentice Hall, 1995.

[10] D. Gross and C.M. Harris, Fundamentals of Queueing Theory, 3rd ed., Wiley Interscience, 1998.

Wei-Ming Yin received B.A., M.S., and Ph.D. in Computer and Information Science from National Chiao Tung University in 1995, 1997, and 2001, respectively. He worked for Computer and Communications Laboratory of Industrial Technology Research Institute as a software engineer. Currently, he serves as a senior engineer in Fastlink Communication Corporation, which targets on WCDMA chipset design. His research interests include protocol design and analysis of residential networks, bandwidth scheduling over QoS guaranteed networks, and UMTS protocol engineering.

Chia-Jen Wu received B.A. in Computer Science and Information Engineering from National Chiao Tung University in 1998 and received M.S. in Computer and Information Science from the same department in 2000. He works for ET home company as a system engineer. His research interests include protocol design and analysis of CATV networks, and bandwidth scheduling over QoS guaranteed networks.

Ying-Dar Lin was born in Hsi-Lo, South Taiwan, in 1965. He received the Bachelor's degree in Computer Science and Information Engineering from National Taiwan University in 1988, and the M.S. and Ph.D. degrees in Computer Science from the University of California, Los Angeles in 1990 and 1993, respectively. At UCLA Computer Science Department, he worked as a Research Assistant from 1989 to 1993 and worked as a Teaching Assistant from 1991 to 1992. In the summers of 1987 and 1991, he was a technical staff member in IBM Taiwan and Bell Communications Research, respectively. He joined the faculty of the Department of Computer and Information Science at National Chiao Tung University in August 1993 and is Professor since 1999. His research interests include design, analysis, and implementation of network protocols and algorithms, wire-speed switching and routing, quality of services, and intranet services. He has authored two books. Dr. Lin is a member of ACM and IEEE.