Adaptive Control for Delay and Throughput in Broadband GEO Satellite Interactive Channels

Ki-Dong Lee, Deock-Gil Oh, and Ho-Jin Lee

Abstract

We develop an efficient real-time method of delay control for non-real-time data in the return link of interactive multimedia satellite networks. Considering multiple delay levels of packets, we mathematically formulate this control problem so that a fair transmission opportunity may be given to satellite interactive terminals. With problem reduction, we promote the computational efficiency in searching for the optimal solution for delay control.

Index Terms

Interactive, satellite, scheduling, delay.

NOMENCLATURE

	Input parameters
$\mathcal{R}$	the set of log-on RCSTs
$\mathcal{S}$	the set of available timeslots per frame
M	the no. of delay classes ($m = 1, \cdots, M$)
$d_{j,m}$	the length of packets of delay class m; measured in timeslots
v_m	the penalty weight factor (per packet) of delay class m
	Control variables
x_{ij}	$\begin{cases} 1, & \text{timeslot } i \text{ is assigned to RCST } j \text{ (for } i \in \mathcal{S},\ j \in \mathcal{R}) \\ 0, & \text{otherwise} \end{cases}$

I. INTRODUCTION

The interactive multimedia satellite (IMS) networking technology will provide a wide variety of mixed digital video broadcasting (DVB) and high-speed digital communication services. The ISM network usually consists of a hub, a geostationary satellite, and a group of satellite interactive terminals (RCSTs). The European standard for the ISM system [1] calls for a return link using multi-frequency time-division multiple access where a frame , a specific time-frequency block (bandwidth: W_f; duration: T_f) in the time-frequency domain, consists of a number of timeslots (bandwidth: W_t; duration: T_t). RCSTs in need of capacity send capacity request (CR) messages to the hub via a satellite, and the hub makes a terminal burst time plan (TBTP). As a result, there is inherent delay which may cause critical degradation of quality-of-service (QoS).

After the recent release of the standard [1], even though several scheduling methods are proposed [3], [4], QoS degradation caused by queueing delay in the interactive satellite return link has not been mathematically addressed.

Under heavy traffic conditions in the return link, each RCST may not obtain all the capacity that it has requested to the hub and, therefore, it may suffer delay (the packet delay levels are classified by *age* measured in frames). If a packet is not sent for a expiration period T_{out}, it is discarded from the queue. The objective of this study is to minimize the packet discard rate under a time-expiration constraint.

Introducing penalty weight factors for M delay classes, we mathematically formulate the objective as a timeslot scheduling problem and develop an exact realtime solution algorithm. We analytically evaluate the optimality of the proposed algorithm and the computational efficiency.

II. PROBLEM FORMULATION

A. The Objective and the Objective Function

Even though each RCST may not obtain all the capacity it has requested because of many RCSTs competing for capacity, the average packet delay among the RCSTs must be controlled to minimize the packet discard because of time-expiration. With the penalty weight factors $\{v_m\}$, long-delayed packets may obtain more capacity than younger packets so that the average packet delays of the respective RCSTs are equally likely. For this, the penalty weight factors must satisfy

$$0 < v_1 < \cdots < v_M. \tag{1}$$

Conceptually, the penalty is defined as the (weighted) number of timeslots which is not assigned to a given RCST generating a CR message.

The authors are with ETRI, Daejeon, 305-350 Korea. (e-mail: kdlee@etri.re.kr)

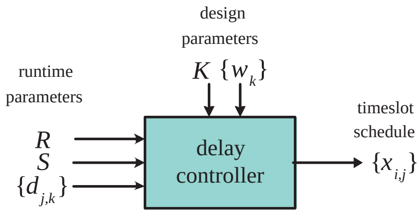

Fig. 1. Controller function diagram.

Let $p_j(A_j)$ denote the penalty of RCST j when A_j timeslots are allocated to the RCST ($A_j = \sum_{i \in \mathcal{S}} x_{ij}$). The penalty function of RCST j is defined as a continuous function such that $p_j(z_{j,M}) = 0$ and

$$p_j(y)\Big|_{A_j}^{A_j+1} = -v_{M-m+1}, \;\; z_{j,m-1} \leq A_j < z_{j,m} \tag{2}$$

(for $m = 1, \cdots, M$), where

$$z_{j,m} = \begin{cases} \sum_{n=1}^{m} d_{j,M-n+1}, & m \geq 1; \\ 0, & m = 0 \end{cases} \quad \forall j \in \mathcal{R}.$$

Then we can rewrite $p_j(\cdot)$ as

$$p_j(A_j) = a_0 + \sum_{m=1}^{M} a_m \cdot |A_j - z_{j,m-1}|, \tag{3}$$

where $a_0 = \sum_{m=1}^{M} (v_n \cdot d_{j,n} - a_n \cdot z_{j,n-1})$ and

$$a_m = \begin{cases} -\frac{1}{2}(v_1 + v_M), & m = 1 \\ -\frac{1}{2}(v_{M-m+1} - v_{M-m+2}), & m > 1 \end{cases}.$$

The overall penalty caused by the log-on RCSTs is given by

$$p(A) = \sum_{j \in \mathcal{R}} p_j(A_j) \tag{4}$$

where $A = (A_1, \cdots, A_M)$. Since the final goal is to find the optimal timeslot assignment $\mathbf{x}$, we rewrite the objective (4) as follows

$$g(\mathbf{x}) = \sum_{j \in \mathcal{R}} p_j \Big(\sum_{i \in \mathcal{S}} x_{ij} \Big). \tag{5}$$

Example: Suppose that an RCST has two 1-age packets and three 2-age packets (one timeslot mapped to one packet), i.e., $(d_{j,1}, d_{j,2}) = (2, 3)$, and $(v_1, v_2) = (1, 2)$. If it obtains zero timeslot, then its penalty is $1 \cdot 2 + 2 \cdot 3 = 8$; else if 4 timeslots, then $1 \cdot 1 + 2 \cdot 0 = 1$; else if 5 timeslots, then zero.

B. Problem Formulation

With $g(\mathbf{x})$ in (5), we formulate the timeslot assignment problem (TAP) as a binary integer programming problem as follows.

(TAP)
Minimize $\quad g(\mathbf{x})$
Subject to

$$\sum_{j \in \mathcal{R}} \sum_{i \in \mathcal{S}} x_{ij} \leq |\mathcal{S}| \tag{6}$$

$$\sum_{j \in \mathcal{R}} x_{ij} \leq 1, \;\; \forall i \in \mathcal{S} \tag{7}$$

$$\forall \;\; x_{ij} \;\in\; \{0, 1\}$$

Constraint (6) means that the total number of timeslots to assign cannot be greater than the number of available timeslots, and constraint (7) means that each timeslot cannot be assigned to more than one RCST simultaneously.

Phase I: Solve (TAP1) - Optimizing Assignment Amount
Find m^* such that $\sum_{j\in\mathcal{R}} z_{j,m^*-1} \le |\mathcal{S}| < \sum_{j\in\mathcal{R}} z_{j,m^*}$;
$d'_{j,M-m^*+1} := \left\lfloor (|\mathcal{S}| - \sum_{j\in\mathcal{R}} z_{j,m^*-1}) \cdot \frac{d_{j,M-m^*+1}}{\sum_{j\in\mathcal{R}} d_{j,M-m^*+1}} \right\rfloor$;
$A^*_j := z_{j,m^*-1} + d'_{j,M-m^*+1}$; // $\forall j \in \mathcal{R}$
Phase II: Solve (TAP2) - Timeslot Scheduling
for($\forall j \in \mathcal{R}$)
for($\mathbf{x}^* := \mathbf{0}$; $A^*_j > 0$; $A^*_j := A^*_j - 1, \mathcal{S} := \mathcal{S} - \{i\}, x^*_{ij} := 1$)
select an element i from $\mathcal{S}$;

Fig. 2. The proposed solution algorithm for (TAP).

III. Solution Method

A. Problem Reduction and Decomposition

Since (TAP) usually has a great number of control variables, a direct solution approach may cause computational inefficiency. To improve computational efficiency, we use a *problem reduction technique* [5]. This idea is incorporated into the following two subproblems: (TAP1) is to optimize the assignment amount for each RCST, and (TAP2) is for timeslot scheduling.

(TAP1)
Minimize $p(A)$
Subject to

$$\sum_{j\in\mathcal{R}} A_j \le |\mathcal{S}| \tag{8}$$

$$A_j \ge 0, \ \ \forall j \in \mathcal{R}$$

(TAP2)
Find $\mathbf{x}$
Subject to

$$\sum_{i\in\mathcal{S}} x_{ij} = A^*_j, \ \ \forall j \in \mathcal{R} \tag{9}$$

$$\sum_{j\in\mathcal{R}} x_{ij} \le 1, \ \ \forall i \in \mathcal{S} \tag{10}$$

$$\forall \ x_{ij} \ \in \ \{0,1\}$$

B. Exact Solution Algorithm

We propose a simple and exact solution algorithm for (TAP) together with several propositions regarding the performance of the proposed algorithm: the *optimality* of resulting solutions and the *computational efficiency*.

Proposition 1: Let $\mathbf{x}^*$ and A^* be the optimal solution of (TAP) and the optimal solution of (TAP1), respectively. Then we have $g(\mathbf{x}^*) \ge p(A^*)$.

Proof: Substitute $\sum_{i\in\mathcal{S}} x_{ij}$ with A_j in (TAP1). Then, the feasible region of (TAP) is a subset of the feasible region of (TAP1) because of constraint (7). Thus, the optimal objective value of (TAP1) cannot be greater than that of (TAP). ■

Proposition 2: If there exists a feasible vector $\mathbf{x}^{**}$ of (TAP2), then $\mathbf{x}^{**}$ is the optimal solution of (TAP).

Proof: If $\mathbf{x}^{**}$ satisfies constraints (9) and (10) of (TAP2), then it also satisfies the constraint (6) of (TAP). The feasible region characterized by (8), (9), and (10) is equal to the feasible region of (TAP). Thus, we have $g(\mathbf{x}^{**}) = g(\mathbf{x}^*)$. This completes the proof. ■

Proposition 3: The feasible set of (TAP2) is non-empty, i.e., there exists at least one vector $\mathbf{x}^*$ that solves (TAP2).

Proof: Suppose that we choose arbitrary A^*_j timeslots for RCST $j \in \mathcal{R}$. Let $\mathbf{x}^*$ denote this assignment. This assignment satisfies (9) and also satisfies (10) because each timeslot is chosen only once. ■

According to Propositions 1–3, a sequential solution procedure of (TAP1) and (TAP2) provides the optimal solution of (TAP). We suggest an efficient and exact algorithm for (TAP1) and (TAP2) as shown in Fig. 2. We optimize the amount of capacity to assign to the respective delay classes of each RCST in Phase I, and then we specify which timeslots are assigned to each RCST in Phase II.

Proposition 4—Solution Optimality: A^* obtained by the Phase I of the proposed algorithm is the optimal solution of (TAP1).

Proof: Let F^I (an integer set) denote the feasible set of (TAP1). Let us consider a relaxed feasible set of F^I, say F (a real number set). Since $F^I \subset F$, there is no better solution than A^* in F^I if there is no feasible direction improving the objective $p(\cdot)$ over F. Consider a feasible direction $\epsilon = (\epsilon_1, \cdots, \epsilon_{|\mathcal{R}|})$ such that $|\epsilon_j| \ll 1$, $\sum_{j\in\mathcal{R}} \epsilon_j = 0$, and $\|\epsilon\| \ne 0$.

TABLE I

PARAMETER VALUES USED IN SIMULATION.

item	value	item	value
W_f	8MHz	T_f	2688000μs
W_t	4MHz	T_t	84μs
$\|\mathcal{S}\|$	$\sim$unif$(32000, 64000)$	$\|\mathcal{R}\|$	$\sim$unif$(1, 256)$
M	10	$d_{j,1}$	$\sim$Pareto, $j \in \mathcal{R}$
α	0.5	T_{out}	53.760s
v_m	$m \ (= 1, \cdots, M)$	—	—

The guard-bandwidth and roll-off factor are not included in W_t.

TABLE II

UPPER BOUND OF ELAPSED TIME (ms).

$\|\mathcal{R}\|$	Phase I	Phase II	Total
1 $\sim$100	3	6	8
101$\sim$200	3	7	9
201$\sim$256	3	7	9

PC: Pentium III 2.0GHz. Source code in C++.

From the Phase I, we have

$$A_j^* = z_{j,m^*-1} + d'_{j,M-m^*+1}, \quad \forall j \in \mathcal{R}. \tag{11}$$

Let $\mathcal{R}_{d+} = \{j|d'_{j,M-m^*+1} > 0\}$, $\mathcal{R}_{d0} = \{j|d'_{j,M-m^*+1} = 0\}$, $\mathcal{R}_{\epsilon-} = \{j|\epsilon_j < 0, \ j \in \mathcal{R}_{d0}\}$, $\mathcal{R}_{\epsilon+} = \{j|\epsilon_j > 0, \ j \in \mathcal{R}_{d0}\}$, and $\mathcal{R}_{\epsilon 0} = \{j|\epsilon_j = 0, \ j \in \mathcal{R}_{d0}\}$. Then we have $\mathcal{R} = \mathcal{R}_{d+} \cup \mathcal{R}_{d0}$, $\mathcal{R}_{d0} = \mathcal{R}_{\epsilon+} \cup \mathcal{R}_{\epsilon 0} \cup \mathcal{R}_{\epsilon-}$, and

$$\begin{aligned}
&\Delta p(A^*) \equiv p(A^*) - p(A^* + \epsilon) \\
&= \sum_{j \in \mathcal{R}_{\epsilon-}} \epsilon_j \cdot v_{M-m^*+2} + \sum_{j \in \mathcal{R}_{\epsilon+}} \epsilon_j \cdot v_{M-m^*+1} \\
&\qquad + \sum_{j \in \mathcal{R}_{\epsilon 0}} \epsilon_j + \sum_{j \in \mathcal{R}_{d+}} \epsilon_j \cdot v_{M-m^*+1} \\
&= \sum_{j \in \mathcal{R}_{\epsilon-}} \epsilon_j \cdot (v_{M-m^*+2} - v_{M-m^*+1}) \\
&\qquad + \sum_{j \in \mathcal{R}-\mathcal{R}_{\epsilon 0}} \epsilon_j \cdot v_{M-m^*+1} + \sum_{j \in \mathcal{R}_{\epsilon 0}} \epsilon_j \\
&= \sum_{j \in \mathcal{R}_{\epsilon-}} \epsilon_j \cdot (v_{M-m^*+2} - v_{M-m^*+1}) \\
&\qquad + \sum_{j \in \mathcal{R}-\mathcal{R}_{\epsilon 0}} \epsilon_j \cdot v_{M-m^*+1} + \sum_{j \in \mathcal{R}_{\epsilon 0}} \epsilon_j \cdot v_{M-m^*+1} \\
&= \sum_{j \in \mathcal{R}_{\epsilon-}} \epsilon_j \cdot (v_{M-m^*+2} - v_{M-m^*+1}) + v_{M-m^*+1} \cdot \sum_{j \in \mathcal{R}} \epsilon_j \\
&= \sum_{j \in \mathcal{R}_{\epsilon-}} \epsilon_j \cdot (v_{M-m^*+2} - v_{M-m^*+1}) \\
&\leq 0 \qquad (\text{“=” holds if and only if } \mathcal{R}_{\epsilon-} = \emptyset).
\end{aligned}$$

Thus, there is no feasible direction to improve the objective at $A = A^*$ over F. Since F^I is a subset of F, A^* is the optimal solution of (TAP1). ■

Proposition 5—Computational Efficiency: The proposed algorithm has the following computational complexities: $O(M \cdot |\mathcal{R}|)$ in Phase I; $O(|\mathcal{S}|)$ in Phase II.

In a practical system, $|\mathcal{R}|$ and M are much smaller than $|\mathcal{S}|$. Proposition 5 implies that the proposed algorithm has one order computational complexity, i.e., $O(|\mathcal{S}|)$, which shows its computational efficiency.

C. Simulation Results and Discussions

We show extensive simulation results of our algorithms using randomly generated demand vectors and other parameters specified in Table I. α is the fraction of packets with a *timeout* constraint, and these packets with delay more than a timeout threshold T_{out} are deleted from the queue. Table II presents the upper bounds of computing times (ms) elapsed in Phase I and Phase II, respectively, which shows the computational efficiency. Fig. 3 shows that the fraction of discarded packets out of the total packets with a time-expiration constraint is reduced by using the proposed delay control method. It is observed that the fraction of discarded packets is reduced up to 32% and 28% under heavy and sparse traffic conditions, respectively, by using the proposed method.

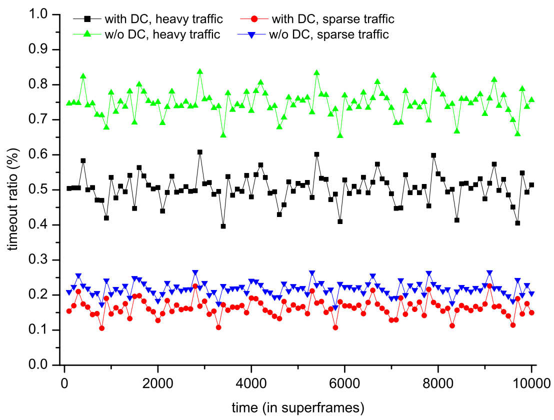

Fig. 3. Timeout ratios (%) with/without delay control (DC): the fraction of deleted packets out of packets with a timeout constraint.

IV. Concluding Remarks

We have developed an efficient method for delay and throughput control in an interactive multimedia satellite network. With this, a transmission opportunity can be provided to the satellite terminals having uneven distribution of capacity demand. To do this, we develop a mathematical optimization formulation together with an efficient exact solution algorithm. Analytical performance evaluation and extensive simulation results showed that the proposed algorithm provides both solution optimality and computational efficiency.

Acknowledgments

This work was a partial result of a dynamic resource management module at the detailed design phase of the mobile broadband satellite access (MoBISAT) system. The first author express thanks to the members of the MoBISAT project for their contribution during interface and integration test phases.

References

[1] ETSI EN 301 790 (V1.2.2): “Digital Video Broadcasting (DVB); Interaction Channel for Satellite Distribution Systems”, 2000.

[2] J. Neale, R. Green, and J. Landovskis, “Interactive Channel for Multimedia Satellite Networks,” *IEEE Commun. Mag.*, vol. 39, no. 3, pp. 192-198, Mar. 2001.

[3] K.-D. Lee, H.-J. Lee, Y.-H. Cho, and D. G. Oh, “Throughput-Maximizing Timeslot Scheduling for Interactive Satellite Multiclass Services,” *IEEE Commun. Letters*, vol. 7, no. 6, pp. 263-265, Jun. 2003.

[4] A. Iuoras, “Advanced scheduling techniques”, *Proc. Ka-band Utilization Conf.*, pp. 397-406, Italy, Sept. 2002.

[5] K. Murty, *Linear and Combinatorial Programming*, Wiley, 1976.

[6] T. Le-Ngoc, V. Leung, P. Takats, and P. Garland, “Interactive Multimedia Satellite Access Communications,” *IEEE Commun. Mag.*, vol. 41, no. 7, pp. 78-85, Jul. 2003.