Guaranteed Quality-of-Service Wireless Medium Access by Packet-by-Packet Generalized Processor Sharing Algorithm

Chia-Sheng Chang, Kwang-Cheng Chen
Institute of Communications Engineering, College of Electrical Engineering
National Taiwan University, Taipei, Taiwan 10617

Abstract— **To serve traffic with different characteristics and service requirements in multimedia wireless packet networks, we propose a multiaccess methodology PGPS/RAP with guaranteed quality-of-service (QoS) to serve constant-bit-rate (CBR), variable-bit-rate (VBR), and available bit rate (ABR) traffic sources. This multiaccess methodology without any specific frame concept can guarantee worst-case delay of CBR and VBR traffic sources and therefore can guarantee the QoS of delay-sensitive, jitter-sensitive CBR and VBR traffic sources. The derived delay bound of a specific traffic only depends on its own traffic parameters. This impressive characteristic benefits bandwidth allocation and consequently results in a simple call admission control policy. Other important issues, including a simple link control mechanism for CBR and VBR over unreliable channels, are also suggested.**

I. Introduction

State-of-the-art wireless networks are designed to carry multimedia traffic such as voice, audio, video, image, in addition to traditional data traffic. To integrate all the traffic together, the quality-of-service (QoS) for all kinds of traffic must be satisfied. As the primary difficulty in end-to-end QoS lies in less reliable wireless access, a guaranteed QoS medium access is obviously the most important issue we need to consider. Traditional multiaccess schemes with integrated services are usually TDMA based approaches with frame structures [2]-[3], while the optimal frame structure to meet general QoS demands is still an open problem. Another scheme using priority scheduling and polling technique was proposed in [6]. However, this priority property prevents fair sharing of bandwidth among traffic sources with similar QoS demands. An effective access along with appropriate bandwidth allocation is still very much needed in multimedia wireless networks.

For the purpose of bandwidth allocation, PGPS algorithm is well known for its flexibility, efficient sharing of resource. We investigate PGPS scheduling algorithm from both principle and implementation points of view. It can be shown that only the head packet information is essential to the scheduling decisions and this characteristic is critical to the applicability of PGPS in a distributed environment, e.g., a wireless channel.

Consequently, we propose a medium access methodology which combines both packet-by-packet generalized processor sharing (PGPS) scheduling algorithm and random addressed polling (RAP) multiaccess protocol to provide access services for constant bit rate (CBR), variable bit rate (VBR) and available bit rate (ABR) traffic sources. This centralized methodology operates without any specific frame structure/concept and uses weighting factor concept to share bandwidth among all traffic sources fairly and efficiently. If the weighting factors of PGPS are assigned according to our proposal, PGPS/RAP can be shown to guarantee worst-case delay of CBR and VBR traffic sources and thus provides guaranteed QoS for delay-sensitive CBR and VBR traffic in multimedia wireless networks.

II. Modified Packet-by-Packet Generalized Process Sharing Algorithm

We first give the definition of a GPS (generalized processor sharing) processor [1].

Definition 1: Suppose there are n first-come-first-serve queues with incoming load packets and a server with rate r bits per second. Each queue has a weighting factor ϕ_i and each packet has a packet-size field (size in bits). If we assume: 1) the processor can serve multiple queues simultaneously. 2) every packet is infinitely divisible like fluid. Then a GPS processor is a work-conserving server with the following property: Suppose $B(t)$ denotes the set of non-empty queues at time t. Then the GPS processor serves the i^{th} queue with rate:

$$r_i(t) = \begin{cases} r \times \frac{\phi_i}{\sum_{j \in B(t)} \phi_j} & \text{if } i \in B(t) \\ 0 & \text{otherwise} \end{cases}$$

Consequently a PGPS scheduler is defined as follows [1]:

Definition 2: The server is work-conserving and suppose that the server becomes free at time t, it picks the first packet that would complete service in the GPS simulation as if no additional packets were to arrive after time t.

PGPS scheduler, a packet-by-packet transmission scheme, is an approximation to GPS processor. An inequality in [1] quantitatively justifies the difference between the output process of GPS processor and that of PGPS scheduler $\hat{F}_p \leq F_p + L_{max}/r$, where r (bps) is the service rate of the server, L_{max}(bits) is the maximum packet length, $\hat{F}_p$ and F_p are the time instants at which packet p departs under PGPS and GPS, respectively.

A. A PGPS Implementation with Minimum Information

PGPS scheduling algorithm was originally proposed to operate in a centralized environment. However, in a multiac-

This research was supported by the National Science Council of Taiwan, R.O.C. under the contract NSC-86-2221-E-007-035.

0-7803-4788-9/98/$10.00 © 1998 IEEE.

cess environment like wireless channel, these queues are distributed and the instant information of each queue is not available. To use PGPS as the scheduling algorithm in a multiaccess environment, it is critical to determine the minimum sufficient information required for PGPS to operate. If we call the first packet of a queue as the "head packet", a non-transparent queue is a queue we can see only its head packet (if there is any). With these definitions, we now give the following proposition.(The proof is in [11].)

Proposition 1: The arrival and packet size information of all head packets is the minimum amount of information needed to make the scheduling order decision satisfying the definition of PGPS scheduling algorithm.

Parekh and Gallager proposed an implementation of PGPS scheduler called "virtual time implementation" in [1], which requires the server being aware of the occurrences of every arrival event and its the packet size. Hence this implementation uses all causal arrival and packet size information and clearly can not work in the non-transparent queue situation. Therefore we shall modify the original "virtual time implementation" of PGPS such that it can operate with the minimum information specified in Proposition 1. If the server can get all causal arrival information, the basic implementation idea is: whenever the server becomes free, it first runs GPS simulation from the beginning of this busy period to the *current time*, denoted by t_{cur}. After that, it *predicts* which packet would complete service first in the GPS simulation as if no additional packets were to arrive afterward. If the queues are non-transparent, problem arises when some head packet's GPS departure time is earlier than its PGPS departure time. In this case, the server has no idea if there is any packet queued behind the head packet due to the head packet's mask. Consequently the server can not know whether this queue is non-empty under GPS after the head packet's GPS departure time. Hence this GPS departure time is the farthest time that the GPS simulation can run, and the server is forced to make scheduling decision even though the GPS simulation can not run to the current time. To overcome this difficulty, we introduce a new time index called *simulation time* in addition to *current time* in this modified implementation. *Simulation time*, denoted by t_{sim}, represents the farthest time to which the GPS simulation can run with "available" arrival and packet size information. Before we state the details of this modified PGPS implementation, we give the definition of virtual time and some other notations as follows [1].

Each arrival and departure from the GPS server is called an *event*. Let t_j be the time at which the j^{th} event occurs relative to the beginning of a busy period. And t_1, the time of first arrival of a busy period, is set to 0. B_j is the set of non-empty queues under GPS model between (t_{j-1}, t_j). Denote a_i^k the arrival time of the k^{th} packet of queue i. Each new packet arrival is marked its arrival time and then appended to the tail of queue i. L_i^k is the length of the k^{th} packet of queue i. Then virtual time, $V(t)$, is defined recursively as[1]: $V(0) = 0$ and for $\tau \leq t_j - t_{j-1}, j = 2, 3, \ldots$

$$V(t_{j-1} + \tau) = V(t_{j-1}) + \frac{\tau}{\sum_{i \in B_j} \phi_i} \cdot r \qquad (1)$$

With the definition of virtual time, we have following notations [1]: S_i^k is the virtual time corresponding to service-beginning time of the k^{th} packet of queue i. F_i^k is the virtual time corresponding to service-finishing time of the k^{th} packet of queue i. $F_i^0 = 0$ for all i and [1]

$$S_i^k = \max\{F_i^{k-1}, V(a_i^k)\} \quad \text{and} \quad F_i^k = S_i^k + \frac{L_i^k}{\phi_i} \qquad (2)$$

In addition, we say that a finishing virtual time F_i^k is valid at time t if $F_i^k > V(t)$. Otherwise, it is invalid.

Given the notations and definitions listed above, we now state this modified implementation: The first scheduling decision is to serve the queue at which the first packet of this busy period arrives. After the service is done, then

begin: the server updates $V(t)$ by using (1) from $t = t_{sim}$ until any of the following two conditions occurs:

1. $V(t)$ reaches a valid F_i^k for some i. The server sets t_{sim} to this reach time and then checks if this packet (the k^{th} packet of queue i) has already departed (i.e. PGPS departure time earlier than GPS departure time).

(a) If this is the case, the server just discards F_i^k and goes back to *begin*.

(b) Otherwise, this packet is the one to serve next. The server takes it out from queue i and now can see if there is any packet queued behind. In this case, the packet queued behind is the $k + 1^{th}$ packet of queue i and the server sets F_i^{k+1} by (2). When the service of the selected packet is done, the server goes back to *begin*.

2. t reaches t_{cur}. The sever sets $t_{sim} = t_{cur}$. In this case, the server does own sufficient information to run the GPS simulation to *current time*. Hence the server chooses the queue with a head packet marked the minimum finishing virtual time F_i^k. The server takes it out from queue i and checks if there is any packet queued behind. In this case, the packet queued behind is the $k + 1^{th}$ packet of queue i and by (2) the server sets: $F_i^{k+1} = \max\{F_i^k, V(a_i^{k+1})\} + \frac{L_i^{k+1}}{\phi_i} = F_i^k + \frac{L_i^{k+1}}{\phi_i}$. When the service of the selected packet is done, the server goes back to *begin*. ■

This modified version differs from its original version in that:

1. The finishing virtual time is determined when a packet becomes a head packet. Nevertheless, in the original version, the finishing virtual time is determined at the packet arrival time.

2. A new time index t_{sim} is introduced to track the progress of the GPS simulation.

We therefore propose a modified "virtual time implementation of PGPS" which requires only the head packets information to operate. This modification makes PGPS algorithm applicable in more general environments.

III. Combining PGPS Algorithm and RAP Protocol

A. System Model

Consider the following multiaccess communication model: There are n nodes, each node has a queue of packets to be transmitted and the multiaccess channel is a common server (denote C as the channel capacity). In addition, there is a special node called the coordinator where these n nodes use this multiaccess channel to communicate with it. Under this model we try to serve three kinds of source: constant bit rate (CBR), variable bit rate (VBR), and available bit rate (ABR) sources.

We further assume: 1) Propagation delay is negligible compared to a packet transmission time. 2) All packets generated from a source are of the same size. 3) The packets of all ABR traffic sources are of the same size but the packet of CBR or VBR traffic sources may be of different size. 4) There exists a fixed parameter L_{max}, which is a size upper bound of all packets.

A CBR traffic source is characterized by four parameters $(r_i^c, L_i^c, T_i^c, D_i^c)$, where r_i^c (bps) is the rate of this source, L_i^c is the size of packets of this source, T_i^c is the packet interarrival time, D_i^c is the maximum tolerable delay measured from the arrival time of a packet to its departure time from the i^{th} CBR local queue, and $r_i^c = \frac{L_i^c}{T_i^c}$.

A VBR traffic source is characterized by five parameters $(r_i^v, L_i^v, T_i^v, Q_i^v, D_i^v)$, where r_i^v (bps) is the peak rate of this source, L_i^v is the size of packets of this source, T_i^v is the minimum packet interarrival time, Q_i^v is the query packet size of this source (explained in III-C), D_i^v is the maximum tolerable delay measured from the arrival time of a packet to its departure time from the i^{th} VBR local queue, and $r_i^v = \frac{L_i^v}{T_i^v}$.

For ABR source, we assume that if two or more ABR sources send their packets in a transmission interval, then there is a collision. The packet is correctly received for only one packet in the multiaccess channel.

B. RAP Multiaccess Protocol

In the following we state the operation of simplified RAP protocol which will be used later and then define a few notations which will be referenced in subsequent subsections. A complete operation of RAP can be found in [5]. Its generalization to represent a very wide-range multiaccess protocols can be found in [10].

Step 1 The base station broadcasts a [READY] message to all mobile nodes in its coverage area.

Step 2 Each active mobile node that intends to transmit generates a random number from the set $\{0, 1, 2, \ldots, p-1\}$ and simultaneously transmit their random numbers with orthogonal signals, e.g. by different frequencies or time slots.

Step 3 The base station polls each active random number one by one. The mobile nodes who sent the polled random number transmit packet to the base station. Collision is possible since there might be two mobile nodes sending the same random number.

Step 4 If the base station successfully/unsuccessfully receives the packet from any mobile node, it sends a positive/negative acknowledge ([PACK]/[NACK]) right away before polling the next one(s). Then go back to *Step 1*.

From the point of view of ABR nodes, the coordinator is just like a RAP base station. ABR nodes wait for [READY] signal to up-transmit random addresses and wait for the coordinator to poll each random address. The coordinator and ABR nodes are all aware of the following parameters: p is the number of random addresses, and L_p^a is the fixed ABR packet size. P_u^a is the duration of time required for the coordinator to detect random addresses. L_u^a is the equivalent packet size with transmission time P_u^a in a channel with capacity C bits per second, i.e. $L_u^a \equiv P_u^a C$.

C. PGPS/RAP Multiaccess for CBR and VBR Sources

Considering the multiaccess model in III-A, all packet queues are distributed and the coordinator can not get instantaneous information of the state of each queue. To overcome this difficulty, we propose the concept of "presumed arrival patterns". These arrival patterns are determined from traffic parameters, which are passed to the coordinator at the call setup time. To be more specific, instead of inputing the actual, instant arrival information of user queues, we generate permits according to presumed arrival patterns into the scheduler at the coordinator.

For $(r_i^c, L_i^c, T_i^c, D_i^c)$ CBR source, a grant permit with size L_i^c is generated every T_i^c seconds at the coordinator.

For $(r_i^v, L_i^v, T_i^v, Q_i^v, D_i^v)$ VBR source, a query permit with size Q_i^v is generated every NT_i^v seconds at the coordinator, where N is the maximum positive integer such that $D_i^v \geq NT_i^v + \frac{NT_i^v(L_i^v+Q_i^v)}{NL_i^v+Q_i^v} + \frac{L_{max}}{C}$ (Figure 1).

Whenever a query permit is chosen to serve by the coordinator, the following tasks are performed: 1) the coordinator queries the i^{th} VBR user the number of packet arrivals at the i^{th} VBR local queue in the n^{th} counting interval, $[(n-1)NT_i^v, nNT_i^v)$. 2) the i^{th} VBR user receives this query and replies the number immediately.

Denote P_i^v the duration of the above query operation. The common channel is in use during the query operation, and we can regard this operation as transmitting a packet with equivalent size Q_i^v, where $Q_i^v \equiv CP_i^v$.

Depending on the result of the query operation, we input grant permits with length L_i^v between query permits. Denote n the query result, $0 \leq n \leq N$. If $n = 0$, no grant permits are input after this query permit. Otherwise, if $n > 0$, a grant permit with length L_i^v is generated $NT_i^v Q_i^v/(NL_i^v + Q_i^v)$ seconds after this query permit. Then another is generated every $NT_i^v L_i^v/(NL_i^v + Q_i^v)$ seconds till all n grant permits are generated.

Unlike CBR and VBR nodes, each has its own corresponding presumed arrival pattern. There is only one presumed arrival pattern for *all* ABR nodes at the coordinator and these permits are accumulated in a queue called the "ABR queue". In addition, this ABR queue is always full and the packet size field of a permit depends on its predecessors. In Figure 2, for example, the first is an RA permit with size L_u^a, which represents an up-transmission random address period (UTRA period for short). If two random addresses are detected active, there follows two polling permits with size L_p^a. Then another RA permit follows and the process repeats. Therefore the size field of each permit is determined from its predecessor recursively.

We can observe an important characteristic of these permit generating mechanisms: There are some cases in which the arrival time field and packet size field of a permit depends on the "execution result" of its predecessors. For example, if a query result is 0, there will be another query permit (with size Q_i^v) arriving NT_i^v seconds later. But if the query result is a positive integer, the next will be a grant permit (with size L_i^v) arriving only $NT_i^v Q_i^v/(NL_i^v + Q_i^v)$ seconds later. Therefore we can regard the queues containing permits as non-transparent queues. Only after a permit is chosen and the service operation is done, the scheduler can get the arrival time and packet size information of its successor. We have proved in Section II that by definition, PGPS can operate in non-transparent queue situation and a practical implementation is proposed. Consequently, the PGPS/RAP multiaccess is stated as follows:

(i) Given the permit generating mechanisms stated above, set the weighting factor ϕ_i^c of the i^{th} CBR source to r_i^c; set the weighting factor ϕ_i^v of the i^{th} VBR source to $r_i^v \times (1 + Q_i^v/NL_i^v)$, and set the weighting factor ϕ^a of the ABR queue to $C - \sum_i \phi_i^c - \sum_i \phi_i^v$.

(ii) Use a PGPS scheduler to select a permit queue and then get one permit from the selected queue.

(ii.1) if the permit is for the i^{th} CBR user, the coordinator notifies the i^{th} CBR user to transmit a packet.

(ii.2) if the permit is for the i^{th} VBR user, and

(ii.2.1) it is a query permit, the coordinator queries the i^{th} VBR user.

(ii.2.2) it is a grant permit, the coordinator notifies the i^{th} VBR user to transmit a packet.

(ii.3) if the permit is from the ABR queue, and

(ii.3.1) it is a RA permit, the coordinator asks all ABR users to up-transmit random addresses.

(ii.3.2) it is a polling permit, the coordinator polls the next un-polled random address.

(iii) Determine the arrival time and packet size fields of the next permit of the selected queue. Goto *(ii)*.

It can be shown that for CBR and VBR nodes the packet delay is bounded under PGPS/RAP multiaccess. We present the bounds in two propositions, while proofs are in [11].

Proposition 2 (CBR delay bound) Denote a_p the arrival time of a packet p of the i^{th} CBR source and $\hat{F}_p$ its departure time from the i^{th} CBR local queue. Under PGPS/RAP multiaccess, we have:

$$\hat{F}_p - a_p \leq \frac{2L_i^c}{r_i^c} + \frac{L_{max}}{C} \tag{3}$$

Proposition 3 (VBR delay bound) Denote a_p the arrival time of a packet p of the i^{th} VBR source and $\hat{F}_p$ its departure time from the i^{th} VBR local queue. Under PGPS/RAP multiaccess, suppose a query permit is generated every NT_i^v seconds, we have:

$$\hat{F}_p - a_p \leq NT_i^v + \frac{NT_i^v(L_i^v + Q_i^v)}{NL_i^v + Q_i^v} + \frac{L_{max}}{C} \tag{4}$$

Note the VBR delay bound in Proposition 3 is an increasing function of N. If a VBR source has larger tolerable delay, a larger N can be selected to save the bandwidth consumed by query operations for VBR sources.

D. Call Admission Control

Whenever a new user decides to request a CBR or VBR call, it must first join the ABR users and make a call setup request packet via the ABR service. Once the transmission is successful (no collision), it waits for "call established" signal from the coordinator. If it does not receive the signal for a certain period of time, it knows this request has been rejected by the coordinator.

In the critical load situation, i.e. CBR and VBR occupies all bandwidth of the multiaccess channel, a minimal amount of bandwidth $R_d \ll C$ must be reserved for ABR sources to ensure the normal operation of PGPS/RAP under critical load. Suppose that the call setup request packet contains the parameters of its traffic source, and in the below all traffic parameters of this new user will be subscripted *new*. The call admission control policies are:

1) A CBR call shall be accepted if the delay requirement can be satisfied, i.e. $D_{new}^c \geq \frac{2L_{new}^c}{r_{new}^c} + \frac{L_{max}}{C}$ and there is still sufficient bandwidth for ABR sources, i.e. $C - (\sum_i \phi_i^c + \sum_i \phi_i^v + r_{new}^c) \geq R_d$.

2) If the source is VBR, the conditions are very similar to that of the CBR case. According to the parameters of this VBR source, the coordinator tries to find a positive integer N satisfying both delay requirement and bandwidth requirement. That is, this VBR call will be accepted if there exists a positive integer N such that: $D_{new}^v \geq NT_{new}^v + \frac{NT_{new}^v(L_{new}^v + Q_{new}^v)}{NL_{new}^v + Q_{new}^v} + \frac{L_{max}}{C}$ and $C - \left(\sum_i \phi_i^c + \sum_i \phi_i^v + r_{new}^v \left(1 + \frac{Q_{new}^v}{NL_{new}^v}\right)\right) \geq R_d$. If there are more than one positive integers satisfying these two conditions, N is set to be the largest one.

E. Simulation Results

In this subsection, we give a simulation result of PGPS/RAP multiaccess with ten CBR and seven VBR traffic sources in a wireless channel with capacity 10Mbps. The VBR source 1 is the output of a four-rate-level packet generator. The generator stays in each rate level for 0.015 seconds, and randomly decides the next rate level with equal probabilities. The four rate levels are: 0, 180kbps, 270kbps, 540kbps. Other VBR sources are outputs from ON-OFF coders. For instance, VBR source 2 generates a packet of size 390 bits every 0.001 second with probability 0.6. The CBR and VBR traffic sources take up 98% of the channel capacity and the remaining 2% are used for ABR service, with the parameter $p = 5$, $L_p^a = 1000$, $L_u^a = 200$. The details of all traffic sources are listed in Table I. The simulation maximum delays v.s. the theoretical delay bounds are listed in Table II. Although only 2% of total bandwidth is originally allocated to ABR traffic, approximately 10% of time are flexibly used to execute RAP to provide ABR service. ABR traffic indeed takes advantage of the surplus bandwidth when VBR source are in their lower-rate period. It demonstrates the advantage of our frame-less bandwidth allocation capability in multimedia (radio) networks.

From the result the proposed delay bounds of CBR and VBR traffic sources are justified and all our proposed methods pretty well achieve the role of multimedia radio access in wireless channels.

IV. Conclusions

In this paper we proposed PGPS/RAP methodology as an effective solution for guaranteed QoS medium access of future multimedia networks. From the environment assumption, PGPS/RAP is applicable to all multiaccess networks with short delay, especially wireless networks due to the nature of RAP protocol. Except for the constrain of maximum packet length, PGPS/RAP places no other restrictions on the packet size of CBR or VBR sources as a pretty universal approach. Therefore CBR or VBR sources even with large divergence in rates and packet sizes can be effectively integrated together with ABR traffic. In addition, due to the inherent independent performance guarantees of PGPS scheduler, delay bounds of CBR and VBR users are functions of only their own traffic parameters and system-wide parameters, and totally independent of the parameters of other traffic sources. This impressive characteristic benefits the ease of bandwidth allocation under PGPS/RAP and consequently results in call admission control policies with relatively low complexity. With the proposed simple retransmission policy in [11], PGPS/RAP still functions to meet multimedia networking requirement even in unreliable channels. Another improvement of PGPS/RAP in long propagation delay channel is also discussed in [12].

The VBR query operation can be seen as an information fetching operation to investigate the rate of VBR sources. Since the VBR delay bound is shown to be an increasing function of N (the ratio of a counting interval to T_i^v), we can easily determine the optimum length of a counting interval of the i^{th} VBR source such that this choice maintains the delay in a tolerable range and occupies the least overhead bandwidth at the same time. Therefore the information fetching of PGPS/RAP in a multiaccess channel is performed in a very efficient manner. Discarding the concept of frame structure, our proposed methodology provides extra flexibility in multimedia networking. Consequently, it is obvious that PGPS/RAP methodology is extremely potential to act as a complete solution for the medium access of future multimedia wireless networks.

References

[1] A. K. Parekh, R. G. Gallager, "A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Single-Node Case," *IEEE/ACM Transactions on Networking*, Vol. 1, No. 3, pp. 344-357, June 1993.

[2] P. Narasimhan, R. D. Yates, "A New Protocol for Integration of Voice and Data over PRMA," *IEEE Journal on Selected Area in Communications*, vol. 14, no. 4, May 1996.

[3] G. Anastasi, D. Grillo, L. Lenzini, "An Access Protocol for Speech/Data/Video Integration in TDMA-Based Advanced Mobile Systems," *IEEE Journal on Selected Areas in Comm.*, Vol. 15, No. 8, 1997.

[4] Kwang-Cheng Chen, "Medium Access control of Wireless LANs for Mobile Computing," *IEEE Network*, Sep/Oct, 1994.

[5] Kwang-Cheng Chen, Cheng-Hua Lee, "RAP—A Novel Medium Access Control Protocol for Wireless Data Networks," *Proc. IEEE Globecom'93*, Houston, 1993.

[6] Cheng-Shang Chang, Kwang-Cheng Chen, Ming-Young You, Jin-Fu Chang, "Guaranteed Quality-of-Service Wireless Access to ATM," *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 1, pp. 106-118, 1997.

[7] A. Demers, S. Keshav, and S. Shenkar, "Analysis and simulation of a fair queueing algorithm," *Internet. Res. and Exper.*, vol. 1, 1990.

[8] Y. K. Sun, K. C. Chen, D. C. Twu, "Multiple Access in Packet Switching Networks: Part I—Generalized Tree Protocol," submitted for publication.

[9] D. C. Twu, K. C. Chen, Y. K. Sun, "Multiple Access in Packet Switching Networks: Part II—Stability Properties," submitted for publication.

[10] K. C. Chen, "Multiple Access for Wireless Packet Networks," Multiple Access, Mobility, and Traffic for Personal Comm. Workshop, Melboune, Dec. 1997.

[11] C. S. Chang, K. C. Chen "Guaranteed Quality-of-Service Medium Access by Packet-by-Packet Generalized Processor Sharing Algorithm: Part I—Multimedia Radio," submitted for publication.

[12] C. S. Chang, K. C. Chen "Guaranteed Quality-of-Service Medium Access by Packet-by-Packet Generalized Processor Sharing Algorithm: Part II—Broadband Access," submitted for publication.

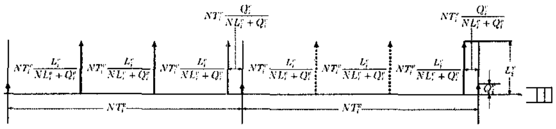
$NT_i^v \frac{Q_i^v}{NL_i^v+Q_i^v}$
$NT_i^v \frac{Q_i^v}{NL_i^v+Q_i^v}$
$NT_i^v \frac{L_i^v}{NL_i^v+Q_i^v}$
$NT_i^v \frac{L_i^v}{NL_i^v+Q_i^v}$
$NT_i^v \frac{L_i^v}{NL_i^v+Q_i^v}$
$NT_i^v \frac{L_i^v}{NL_i^v+Q_i^v}$
$NT_i^v \frac{L_i^v}{NL_i^v+Q_i^v}$
$NT_i^v \frac{L_i^v}{NL_i^v+Q_i^v}$
L_i^v
Q_i^v
NT_i^v
NT_i^v

Fig. 1. VBR presumed arrival pattern.

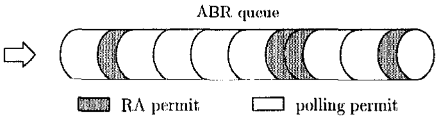

Fig. 2. ABR presumed arrival pattern.

call	r_i^c, r_i^v (bps)	L_i^c,(L_i^v, Q_i^v) (bits)	pon	ϕ_i^c, ϕ_i^v	NT_i^v (sec.)
CBR1	2000000	500	N/A	2000000	N/A
CBR2	1500000	900	N/A	1500000	N/A
CBR3	1000000	600	N/A	1000000	N/A
CBR4	850000	170	N/A	850000	N/A
CBR5	750000	240	N/A	750000	N/A
CBR6	600000	480	N/A	600000	N/A
CBR7	500000	450	N/A	500000	N/A
CBR8	250000	200	N/A	250000	N/A
CBR9	250000	100	N/A	250000	N/A
CBR10	100000	700	N/A	100000	N/A
VBR1	540000	(270,30)	N/A	600000	0.0005
VBR2	390000	(390,30)	0.6	400000	0.003
VBR3	280000	(350,50)	0.8	300000	0.0025
VBR4	280000	(490,70)	0.5	300000	0.0035
VBR5	180000	(450,50)	0.7	200000	0.0025
VBR6	96000	(240,40)	0.2	100000	0.01
VBR7	90000	(360,40)	0.4	100000	0.004

TABLE I

Simulation traffic sources.

call	max. delay (sec.)	bound (sec.)	call	max. delay (sec.)	bound (sec.)
CBR1	0.000447	0.0006	CBR10	0.01097	0.0141
CBR2	0.001058	0.0013	VBR1	0.000792	0.0011
CBR3	0.001057	0.0013	VBR2	0.003717	0.00415
CBR4	0.000371	0.0005	VBR3	0.003527	0.00393
CBR5	0.000559	0.00074	VBR4	0.004659	0.00546
CBR6	0.00142	0.0017	VBR5	0.004241	0.0051
CBR7	0.001579	0.0019	VBR6	0.01181	0.0129
CBR8	0.001368	0.0017	VBR7	0.006402	0.0081
CBR9	0.000657	0.0009			

TABLE II

Simulation results v.s. theoretical bounds in (3) (4).