Adaptive Hierarchical Resource Management for Satellite Channel in Hybrid MANET-Satellite-Internet Network

Nelson X. Liu Xiaoming Zhou John S. Baras
Institute for Systems Research
University of Maryland
College Park, MD 20742
{nliu, xmzhou, baras}@isr.umd.edu

Abstract— **MANETs are often deployed in an infrastructure-less or hostile region where the satellite provides the only link for the MANETs to communicate with the rest part of the world. It faces many challenges to support multiple serviced communications between MANETs and Internet through satellite. In this paper we propose an efficient resource management scheme called AHRM to dynamically allocate bandwidths among multiple MANET users and multiple priority and non-priority services sharing a multi-access satellite channel. It uses a flexible hierarchical structure to exploit the channel utility and resolve contention from two levels. A bandwidth adaptation algorithm is designed to adjust the allocation dynamically in response to traffic and link status changes. The algorithm turns out to be in line with reinforcement learning and is a customized version of it for the practical satellite network setting. Implementation issues are discussed. Simulation results are presented, showing that the scheme can guarantee fast delivery of critical messages in spite of channel contention, and significantly improve the performance of multiple services.**

Keywords-satellite, hybrid network, resource management, adaptive hierarchical scheduling

I. INTRODUCTION

The mobile wireless ad hoc network (MANET) [3] [5] has become very popular in research. It has great potentials in many areas such as environmental observation, military battles, emergency rescue, and scientific investigations. These applications often happen in an infrastructure-less or hostile region where the satellite may provide the only link for the MANET to communicate with other part of world. Furthermore, the MANET data often need to be further relayed through Internet before reaching its home, which we call the service center. Control messages and service data from the service center also need to go through the Internet and satellite to reach the MANET. This scenario is illustrated in figure 1. In this paper we will study the efficient resource management for this hybrid MANET-satellite-Internet environment.

It is clear that the satellite is the single most salient bottleneck in this hybrid network. It largely determines the end-to-end performance between the MANET and the service center. Some features make the satellite link unique as a scarce resource. First, it is the only link that all data have to go through. There is no alternative path to bypass it. Second, the link is so heavily demanded that it is often in contention. Third, it is a long link with very large delay, up to hundreds of milliseconds. Forth, unlike the Internet part, there is much noise on the link. So optimizing the resource management of the satellite link in the hybrid network environment is the most important and the most demanding task for performance improvement.

MANETs as Internet-over-satellite users have more demanding requirements than current commercials users such as those of DirecWay [12]. For example, they may require symmetric satellite communication because there is large-volume traffic in both directions between MANETs and the service center. A MANET may send various types of data such as voice, video, image, and files to the service center, and request various types of data from it. There are often critical messages in the communication, such as control messages, combat commands, emergency alert, and emergency handling instructions. For mobile sensor networks (a kind of MANET), the storage is often quite limited, and sensing data need to be sent out quickly or they will get lost. This requires delay-sensitive services. Our service model for the resource management design assumes 1) each MANET has multiple services, including real-time and elastic services; 2) there are high-priority and low-priority messages. The resource management should allocate bandwidths flexibly for these services and cope with congestion, delay, and variation of link status.

We propose a scheme called the adaptive hierarchical resource management (AHRM). It basically answers the question how much bandwidth every user and every service should be allocated at any moment. It is an abstract design that is independent of particular link layer mechanisms. A particular link layer can follow the guideline and implement the allocation with its own mechanisms. We call the path from the service center to the MANET the forward channel and that from the MANET to the service center the reverse channel. For asymmetric communication such as those used in DirecWay

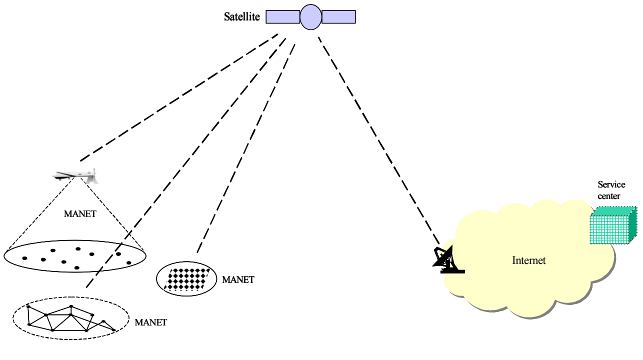

Fig.1. A hybrid network of MANET-satellite-Internet.

This work is supported by DARPA under contract No. N66001-00-C-8063.

system, the scheme suits for the reverse channel. For symmetric communication that allow multi-access mode in both directions, it suits both channels.

II. ADAPTIVE HIERARCHICAL RESOURCE MANAGEMENT

In the AHRM scheme, the resource is managed at two levels explicitly, the user level (level I) and the service level (level II). Each MANET is viewed as a user. Level I allocates bandwidth among users, and level II allocates bandwidth among services within each user. A basic feature of this scheme is that it can guarantee the bandwidth shares of different users and services, which provides the basis for user-differentiated and service-differentiated QoS delivery. High bandwidth users and low bandwidth users, priority services and non-priority services can be well served in this framework. Meanwhile, the hierarchical structure allows the freedom to exploit channel utilization and resolve contention at different granularities. The two levels are two granularities or scales for resource exploitation.

The utility exploitation and the contention resolution are through the dynamic adaptation of the bandwidth shares at both levels. Because of the bursty nature of the traffic, the resource demands of users and services keep varying. The adaptation at user level digs free bandwidth from the users who temporarily have low resource demand and applies it to the users who desperately need it. The adaptation at the service level advances the exploitation among services, and improves the overall quality of all services. This is a repeated optimization up and down the hierarchy and it approximates the maximum utility efficiently. The adaptation algorithm employs reinforcement learning [2] [10]. It is adaptive to both congestion and link status change.

So AHRM uses the hierarchical structure and bandwidth share adaptation to meet the challenge of the resource management. It is a balance of isolation and sharing among multiple users and services. As we will see later, it also supports distributed implementation. We call the whole structure an AHRM scheduler.

III. ALGORITHMS

The AHRM architecture is illustrated in figure 2. At each level there is a "priority queue + weighted fair queue (WFQ)" structure, in which the priority queue handles critical messages and the WFQ manages resources for non-priority services. It should be pointed out that it is possible that physical buffers only exist at level II, and the queues at level I only has logical implication. there. We will discuss more about this later. The operation process is as follows. When a MANET has data ready for transmission, level II sends a request to level I. Level I reviews requests of all MANETs and allocate bandwidths (time slots) for them. These resources are further allocated to services by level II at the time of transmission. We will explain details in this Section.

A. Handling Critical Messages

Critical messages should be let through whenever there is bandwidth. Our policy is, the priority of critical messages is guaranteed consistently throughout the hierarchy. If they are given a high priority within a user but every user shares bandwidth equally, they may be blocked by low priority data of another user. So we set up high priority queues at both levels, which provide a dedicated "channel" for critical messages to go through all levels. When a request is sent to level I from level II, level I allocates resource for all critical messages in the first place wherever they are from. These messages are served in the first-come-first-serve manner by the priority queue at level I. A request from level II to level I includes two fields of information, the total amount of data to be transmitted and how much of it is critical messages.

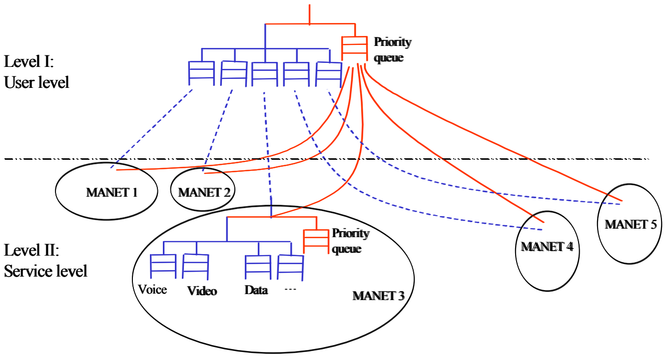

Fig.2. Architecture of the adaptive hierarchical resource management. The illustration involves five MANETs. MANET 3 is used to show the details of level II. Solid lines connecting level II and level I indicate service paths of the critical messages, and dashed lines indicate those of other services.

B. Bandwidth Reservation

Each user initially subscribes a fixed share of bandwidth at level I. A small portion is taken off from all users to amount to a reservation for critical messages. The rest is allocated to each user, which is done by assigning a weight to each user in the WFQ of level I. So the overall allocation plan is as follows:

$$BW_{total} = BW_{res} + \sum_{i} BW_{i}$$

where BW_{total}, BW_{res}, and BW_i represent total bandwidth, reserved bandwidth for critical messages, and bandwidth allocated to user *i*. At level II, each user has the flexibility to allocate its share among services, which is through its own WFQ. However, these allocations are subject to change later. The sole purpose of this step is to keep the initial resources for users and services matching their subscription fees or roughly estimated demands. The shares will be adaptively adjusted with the change of network status.

C. Adaptation

A WFQ can dynamically distribute the excess bandwidth among active users or services by nature. However, this distribution is not most beneficial from the point of view of QoS, because it is merely based on fixed weights and does not reflect instantaneous demands of different users or services. In AHRM the bandwidth share adaptation of two levels is done through dynamical WFQ weight adjustment. At contention due to congestion or link noise, it balances the resource allocation such that overall QoS violation is reduced as much as possible. Even when the channel is not contended, it also "squeezes" some free bandwidth from those who do not need it, and share this among others so as to improve QoS.

1) Quality of Service Criteria

We are interested in four QoS indices, the delay, the loss, the throughput, and the jitter in designing the adaptation algorithm. All services are assumed to be explicitly or implicitly defined by these four dimensions. A service i is assigned a finite buffer size $Q_i = c_i \cdot D_i$ at level II, where c_i and D_i are the initial (or average) bandwidth share and the deadline of the service. So any packets beyond the deadline would be dropped. Even the best-effort data should not be delayed too long in MANETs. The adaptation algorithm is intended to improve the performance at all four dimensions. However, if there is a conflict between the delay and other indices, the absolute delay may be extended as long as it is within the deadline so as to gain better throughput or loss performance.

2) Information Needed

In general, dynamic traffic rate and link status information is needed. To make the algorithm more effective, we also need the instantaneous queue length information. We assume mechanisms collecting these information are available.

3) Weight Adjustment Algorithm

Without loss of generality, below we will describe the adaptation algorithm using level II as an example. It immediately applies to level I by changing "services" to "users". The general idea is as follows. When service i's arrival rate v_i is greater than its bandwidth share (or queue service rate) W_i, there is an incipient danger that the queue increases quickly and the QoS requirement would be violated. We want to prevent this in advance. An efficient way is to adjust W_i in the reverse direction of $W_i - v_i$, i.e.,

$$W_i \leftarrow W_i + \alpha(v_i - W_i) = (1 - \alpha)\, W_i + \alpha v_i, \qquad v_i \geq W_i \qquad (1')$$

where $0 < \alpha < 1$ is a constant. (We have combined the case of $v_i = W_i$ into above formula. In that case the weight does not change). The more v_i is above W_i, the bigger the adjustment is to prevent the violation. Meanwhile, the services with arrival rates less than their bandwidth shares, say service j, reduce their bandwidths, i.e.,

$$W_j \leftarrow W_j - \beta(W_j - v_j) = (1 - \beta) W_j + \beta v_j, \qquad v_j < W_j \qquad (2')$$

where $0 < \beta < 1$ is a constant. The above adjustments are done at each time instance. We notice formulas (1′) and (2′) are of a common form. Although we start to design the adaptation algorithm based on physical rationale, it turns out the above formulation confirms to standard reinforcement learning [2] [10]. W_i or W_j corresponds to the Q-factor, and α or β is the learning rate. Reinforcement learning is known to be an online optimization process that is in many cases equivalent to dynamic programming. We can expect the algorithm to produce good performance.

In spite of its theoretical soundness, the above algorithm needs to be improved in several aspects for our practical network setting. First, it does not consider the effect of queue status. Second, the adjustment should keep the long-term fairness. Third, the free bandwidth saved with (2′) should be balanced with that consumed with (1′). We address these problems by improving the design as follows.

The queue status is more closely related with the QoS violation than the traffic rate. It is possible that the queues size of service i, q_i, is small when $v_i \geq W_i$, and the danger of violation is not urgent. Conversely, if $v_i < W_i$ but q_i exceeds the buffer size of service i, Q_i, the service is already under violation. We consider the impact of the queue status by making α and β in (1′) and (2′) be functions of q_i. Let $\alpha \leftarrow q_i/Q_i$ and $\beta \leftarrow 1 - q_j/Q_j$. Formulas (1′) and (2′) become

$$W_i \leftarrow W_i + (q_i/Q_i)(v_i - W_i), \qquad v_i \geq W_i \qquad (1)$$

$$W_j \leftarrow W_j - (1 - q_j/Q_j)(W_j - v_j), \qquad v_j < W_j \qquad (2)$$

So the queue status modulates the value of the adjustment. In the case of $v_i \geq W_i$, the upward adjustment of (1) is weak when the queue is nearly empty ($q_i/Q_i \approx 0$), and becomes much stronger when the queue is nearly full ($q_i/Q_i \approx 1$). Conversely, in the case of $v_i < W_i$, the downward adjustment of (2) is strong when the queue is nearly empty ($1 - q_i/Q_i \approx 1$), and weak when the queue is nearly full ($1 - q_i/Q_i \approx 0$). By tracing the queue status, the modulations actually make the bandwidth adjustments more smooth because the queue length can not jump suddenly. This contributes to reducing the jitter while adapting the bandwidth share.

To enhance the modulation effect of q_i/Q_i, especially at two extremes of $q_i/Q_i \approx 0$ and $q_i/Q_i \approx 1$, and helps prevent queue overflow, we modify (1) and (2) by replacing q_i/Q_i with a parameter γ that takes value as follows.

$$\gamma = \begin{cases} 1, & q_i/Q_i \geq \eta_H \\ \eta_L + (q_i/Q_i - \eta_L)(1 - \eta_L)/(\eta_H - \eta_L), & \eta_L \leq q_i/Q_i < \eta_H \\ \eta_L, & q_i/Q_i < \eta_L \end{cases}$$

where η_L and η_H are two thresholds satisfying $0 < \eta_L < 1$, $0 < \eta_H < 1$, and $\eta_L < \eta_H$. So γ changes linearly in the section $[\eta_L, \eta_H)$, but takes a non-zero value η_L when $q_i/Q_i \approx 0$ and equals 1 when $q_i/Q_i \approx 1$.

Now we address the second problem. With the iterations of (1) and (2) the bandwidth allocated to a service may drift away from its subscribed share in the long run. Careful enhancements are needed to keep the long-term fairness. As a simple solution we use the subscribed share as a reference point in the formulas, and turn the iterative updates to memory-less updates. Let W_i^0 be the initial bandwidth share of service i. We have the following formulas to replace (1) and (2):

$$W_i \leftarrow W_i^0 + \gamma(v_i - W_i^0), \qquad v_i \geq W_i^0 \qquad (1+)$$

$$W_j \leftarrow W_j^0 + (1-\gamma)(v_j - W_j^0), \qquad v_j < W_j^0 \tag{2+}$$

These formulas are applied when services with $v_i \geq W_i$ and services with $v_j < W_j$ exist simultaneously. If $\forall i \in I$, $v_i \geq W_i$, or $\forall i \in I$, $v_i < W_i$, where I is the set of indices of all services, we let

$$W_i \leftarrow W_i^0 \tag{3+}$$

i.e., every service returns to its subscribed bandwidth. We argue this is fair because when everyone is in trouble, everyone deserves its maximum capability to handle its own crisis (or they are unwilling to share own resources to others); on the contrary, when no one needs additional bandwidth, it is a waste of energy to adjust the initial bandwidth share. It is easy to understand that above algorithm works best in achieving fairness with traffic shapers being present at the access point to the satellite channel, because traffic shapers, such as leaky bucket filters, can ensure that services' long-term traffic rates do not exceed their subscribed shares.

Now we address the third problem. In fact, it is about how to assign weights based on the above bandwidth adjustments. The free-bandwidth consumers should not consume more than the free-bandwidth producers can provide. We implement this by simply sequencing the adjustments. The producers, i.e., services with $v_j < W_j^0$, are first adjusted, which is called step 1, and the consumers, i.e., services with $v_i \geq W_i^0$, follow, which is called step 2. After step 1 we can re-calculate the new weight of *each* producer and the overall weight of all consumers. Denote by w^+ the *overall* weight of all consumers. After step 2, new bandwidth shares of all consumers are available. We normalize these shares *within* the consumer group without considering the producers. Let η_i be the normalized share of consumer i. Then its final weight is $w^+\eta_i$. This way the free bandwidth is exactly distributed among consumers. Of course, when (3+) applies, all weights are reset.

With a channel status monitor, above weight adjustment can be made adaptive to the channel status change. The monitor measures the instantaneous channel capacity c. If c's change exceeds certain threshold Δ, which is called the monitor resolution, an additional adjustment is triggered. This can be viewed as a "re-optimization" of the bandwidth allocation to make most of the new capacity. Schemes for channel monitoring are available in literature, and we will not get into details of it.

D. Implementation issues

An advantage of the AHRM scheme is that it supports flexible implementations, either centralized or distributed. In the centralized implementation, the whole AHRM scheduler is put together and operates at a single point, either on board of satellite or in the gateway on ground. In the distributed implementation, level I runs on board and level II resides on ground. This way the complexity is distributed and the overall reliability of the resource management increases.

We expect the communication between level I and level II would not affect the scheduler's performance much. In the centralized implementation, the "request" from level II and level I is only in logical sense, because level I can directly "see" the queues at level II and get the information it needs. In the distributed implementation, explicit request messages may be necessary but we still can explore efficient mechanisms to avoid the "request-and-wait" delay, which is beyond the scope of this paper.

IV. SIMULATION RESULTS

In the simulations we will evaluate the performance of the AHRM. The network setting is similar with that shown in figure 1. Three MANETs communicate with a service center through a satellite and high-speed Internet. The satellite has asymmetric channels: the bandwidth of the forward broadcast channel is 2Mbps and that of the reverse multi-access channel 384Kbps. The propagation delay from ground to the satellite is 125 ms. The AHRM scheduler sits in the satellite gateway to allocate bandwidth for the reverse channel. Multiple services are included: critical messages, video, voice, and data file transfer. The simulation environment is OPNET.

The scheme will be compared with two commonly used schemes: the plain FIFO scheduling without hierarchical structure and the priority multi-queue scheduling in which each service has a separate queue with a unique priority value. The priority order in the priority scheduling scheme is, critical message > voice > video > data. Some satellite gateway performance optimization techniques including connection splitting and rate-based flow control [11] are adopted in the simulations, which are common for all schemes.

In the simulation, three MANETs send traffic to the service center. All users generate critical messages. In addition, MANET 1 has one voice flow and three video flows, MANET 2 has four voices flows, and MANET 3 has one voice flow and a large file of 1MB. Critical messages are periodical packets with size 2KB each and frequency 1 packet every 3 seconds. Voice traffic and video traffic are generated with exponential and heavy-tailed on-off models, respectively. A voice flow has an average rate of 16Kbps, and a video flow 80Kbps. Noise is added to the channel such that its capacity randomly varies between 0 and 40Kbps below the ideal capacity.

Figures 3 and 4 compare the performance of the three schemes. The three graphs in figure 3 show cumulative distributions of the response time (round-trip delay) of critical messages, and the end-to-end delays of voice and video services. The cumulative distribution implies information on both delay and jitter. In the third graph, for example, if we look at the vertical line of delay = 0.5s, we see that almost 90% of video packets have delays below 0.5s with the AHRM scheme, while only around 10% packets have so low delays with the FIFO and the priority schemes. It is easy to understand that the more a curve is on the left in those graphs, the less delay it means, and a larger horizontal span generally indicates bigger jitters. An ideal performance is a vertical line close to the Y-axis. We see that in all three graphs the curves for the AHRM scheme are closest to the Y-axes and are exactly or nearly

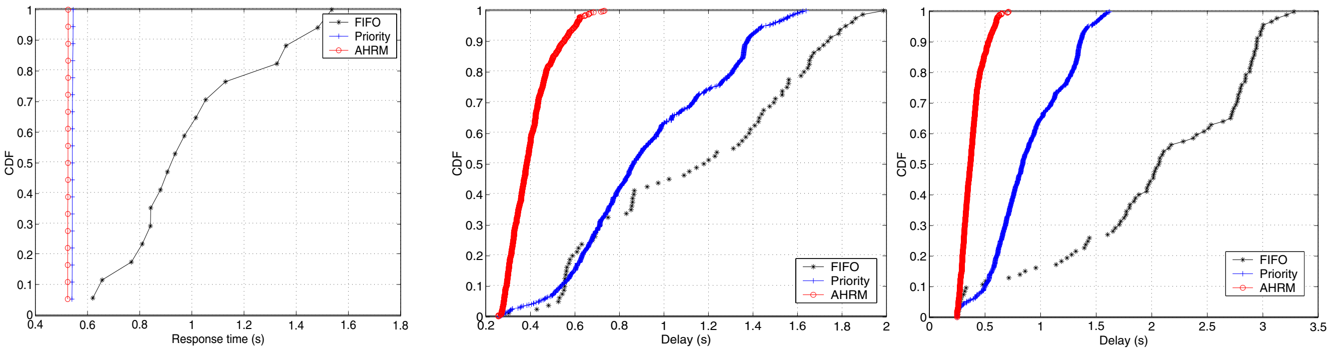

Fig.3. Performance comparison of AHRM with FIFO and priority scheduling schemes. From left to right: results for critical messages, the voice service, and the video service.

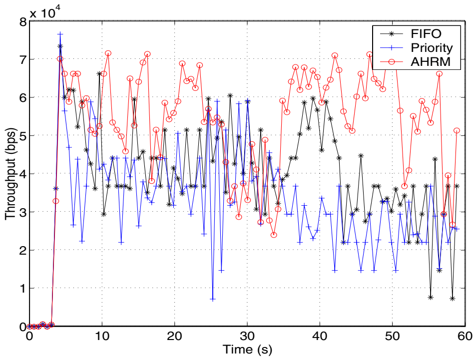

Fig.4. Performance comparison of AHRM with FIFO and priority scheduling schemes for data services.

vertical. Those for the FIFO scheme are on the far right and span widely, and those for the priority scheme are in the middle. Figure 4 compares the throughputs for the data service, which ranks the schemes in decreasing performance order as AHRM > Priority > FIFO. So the ARHM scheme consistently outperforms other two schemes for all services.

V. RELATED WORK

There has been much work on hybrid satellite network resource management, mainly for ATM or IP over satellite. We can only name a few of them as representatives. Paper [6] proposed to use a round robin scheduling scheme in ATM over satellite. Thesis [1] proposed two scheduling strategies to improve the performance of the best-effort traffic over satellite link. Paper [8] gives a DFQ queueing policy to support real-time multimedia services over satellite. Paper [9] evaluates a differentiated service implementation with CBQ link sharing. Paper [7] discusses multiple services over satellite in general. Paper [4] briefly describes an idea of hierarchical fair bandwidth allocation. But it uses a very different approach from us. It has no priorities or adaptations, and is only for best effort connections.

VI. CONCLUSIONS

In this paper we present an adaptive hierarchical resource management scheme for satellite channel in the hybrid MANET-satellite-Internet network. The hierarchical structure enables flexible bandwidth allocation in different levels, and helps exploit the utilization of the satellite link. It supports the distributed implementation, which can improve the reliability and the scalability of the resource management. The weight adaptation algorithm further advances the resource efficiency in finer granularity. It tolerates contention among services and reduces the probability of QoS violation. These mechanisms also help the resource management adapt to the channel status change. Simulation results show that the AHRM scheme can ensure the critical messages to get through in spite of contention, and improve the quality of various services.

REFERENCES

[1] H.O. Awadalla, Resource Management for Multimedia Traffic over ATM Broadband Satellite Networks. Ph.D. Thesis, University of London, March 2000.

[2] D. Bertsekas and J. Tsitsiklis, *Neuro-Dynamic Programming*. Athena Scientific, Belmont, MA, USA, 1996.

[3] S. Corson and J. Macker, Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations. *IETF RFC* 2501, January 1999.

[4] A. Girard, C. Rosenberg, and M. Khemiri, Fairness and Aggregation: A Primal Decomposition Study, Networking 2000, in G. Pujolle et al eds, *Lecture Notes in Computer Science* 1815, Springer-Verlag, May 2000, pp. 667-678.

[5] Z.J. Haas,et al, eds., *IEEE Journal on Selected Areas in Communications – Special Issue on Wireless Ad Hoc Networks*, 17(8), 1999.

[6] A. Hung, M. Montpetit, and G. Kesidis, ATM via Satellite: A Framework and Implementation. *Wireless Networks – Special Issue: Hybrid and Satellite Communication Networks*, 4(2):141-153, 1998.

[7] T. Le-Ngoc, V.C.M. Leung, P. Takats, and P. Garland, Interactive Multimedia Satellite Access Communications, *IEEECommunications Magazine*, 41(7):78- 85, 2003.

[8] H. Le Pocher, V.C.M. Leung, , and D.W. Gillies, Real-Time Multimedia Scheduling Policies For End-To-End Delay Jitter And Loss Guarantees Across ATM Satellite Systems. *IEEE Journal on Selected Areas in Communications*, 17(2):314–25, 1999.

[9] L.S. Ronga, T. Pecorella, R. Fantacci, F. Volpi, Real-Time QoS DiffServ Gateway Implementation for Satellite IP Networks. COST272 TD02-013.

[10] C.J. Watkins, Learning from Delayed Rewards. PhD Thesis, Kings College, Cambridge, England, May 1989.

[11] X. Zhou, N. Liu, and J. Baras, Flow Control at Satellite Gateways, Technical Report ISR TR 2002-37, Institute for Systems Research, University of Maryland, October 2002.

[12] Hughes Network Systems, Inc., URL: http://www.hns.com/.