Differentiated Services over Cable Networks

Nada Golmie *Frederic Mouveaux* *David H. Su*

NIST – National Institute of Standards and Technology
Gaithersburg, MD 20899
Email: {nada.golmie |frederic.mouveaux|david.su}@nist.gov

Abstract

The current Medium Access Control interface specifications of the Multimedia Cable Network System (MCNS) project describe a best effort access service over cable networks. In this paper, we show that even though these specifications are mainly intended to support IP traffic, there is a significant benefit in the stratification of the cable service offerings. We present a priority system to extend the MCNS MAC protocol so that differentiated services can be supported on Hybrid Fiber Coaxial networks with different levels of performance.

1 Introduction

By taking advantage of an already deployed broadband architecture, cable network operators are in a unique position to compete with several other access technologies such as xDSL, ADSL, and ISDN, etc. However, the key to success for cable modems is the ability to offer more than just best effort service and deliver voice, data and video services in addition to high-speed access.

The first MAC interface specifications to be completed are the Data Over Cable Service Interface Specifications (DOCSIS) [3], started as a project of the Multimedia Cable Network System (MCNS) organization and later adopted as a standard by the Data Standard Subcommittee of the Society of Cable Telecommunications Engineers (SCTE). While guaranteed quality of service was not part of the functional requirement of the standard that was originally designed to provide Internet access to the home, now there is a clear need for supporting differentiated classes of service even for Internet traffic. Furthermore in many markets operators are developing stronger business cases for offering data services to the business community rather than offering Internet access to the home. This change in focus may even favor an ATM-induced solution similar to one provided by the IEEE 802.14 draft specifications [2], or other vendor specific implementations that can provide differentiated quality of service. The need to extend the MCNS DOCSIS specifications for guaranteed quality of service becomes a necessity in order for it to compete against ATM friendly MAC protocol specifications.

The main goal of this paper is to describe a multi-tiered priority system at the MAC layer that could support different negotiated services for higher layer applications over a cable network. We believe that if differentiated service techniques implemented at higher layers are mapped into corresponding ones at the MAC layer, then true differentiated services can be provided end-to-end.

The rest of the paper is structured as follows. In section 2 and 3, we give some general insights on the MAC operation and then describe our scheme. In section 4 we evaluate the proposed priority system and present simulation results. A conclusion is offered in Section 5.

2 MCNS MAC Protocol

In this section we give a brief overview of the MAC protocol as specified by the MCNS DOCSIS standard [3]. The priority scheme described in Section 3 largely depends on the basic operation of the MAC protocol. In multiaccess environments, an extra sublayer, the Medium Access Control (MAC) sublayer, is generally added between the Data Link Layer and the Physical Layer in order to allocate the medium among various nodes. Hybrid Fiber Coaxial (HFC) networks are multiaccess environments characterized by a branch and tree topology. At the root of the tree the headend controls the downstream (one-to-many broadcast) and the upstream (many-to-one shared among all stations) transmission. Thus the main purpose of the HFC MAC protocol is to coordinate the communication between the headend and the stations and control the behavior of users who want to access the network. According to the MCNS specifications, the HFC upstream channel is divided into discrete basic time slots, called minislots. A variable number of minislots are grouped to form a MAC layer frame as shown in Figure 1. The headend determines the frame format by setting the number of data slots (DS) and contention slots (CS) in each frame and then sends this information to the stations on the downstream using an *Upstream Bandwidth Allocation Map* message. Several minislots can be grouped together in order to form a DS that carries a MAC Packet Data Unit, (MPDU) which is assumed to be an IP packet (or ATM cell) plus the MAC layer overhead. The DS are explicitly allocated to a specific station by the headend, using a *DS Grant* contained in the *Allocation Map*. CS fit into one minislot and are used by stations to transmit requests for bandwidth. Since more than one station can transmit a request at the same time, CS are prone to collisions. The headend controls the access to the CS by setting an initial backoff window, or *Data Backoff Start*.

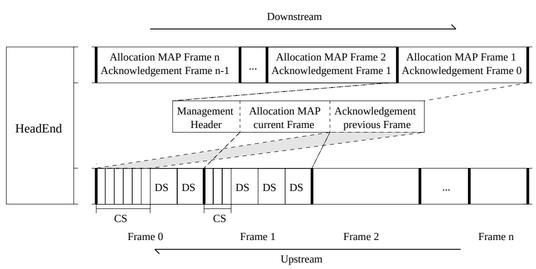


Figure 1: MCNS Request and Grant Mechanism

When a station has data to send it sets its internal backoff window equal to *Data Backoff Start*, as defined in the *Allocation Map*. The station then randomly selects a number within this backoff window. The random value indicates the number of contention transmit opportunities, that the station must defer before transmitting. After a contention transmission, the station waits for either a *Data Grant* or an *Acknowledgement* (*Ack*) in a subsequent *Allocation Map* (whichever comes first). Once either is received the contention resolution process is complete and the station activates its data transmission state upon the arrival of a *Data Grant*. Upon receipt of a station's request (in case of a successful transmission), the headend processes it and assigns a DS to the station by sending a *Data Grant* in the *Allocation Map*. The headend may send an *Ack* to the station in case it needs more time to process the request and send the *Data Grant*. But since multiple stations may attempt to send their request in the same upstream CS, a collision may occur. The station detects the collided slot when it does not find an *Ack* or *Data Grant* for it in the *Allocation MAP* with an *Ack Time* more recent than the time of its transmission. The station must then increase its backoff window by a factor of two as long as it is less than the maximum backoff window set in the *Allocation Map*. The station randomly selects a number within this new window and repeats the contention process described above. After 16 unsuccessful retries the station discards the MPDU.

3 A Priority Mechanism for MCNS MAC Protocol

Previously we have shown that using a preemptive scheduler when allocating bandwidth to stations at the headend is not sufficient to effectively implement priority access for the IEEE 802.14 MAC protocol [4]. This result holds for the MCNS MAC specifications. The problem is mainly the following. During the request in contention process there is no preferential treatment for stations of higher priority; all stations are treated equally. Furthermore, there is no mechanism to separate and resolve collisions in a priority order.

3.1 Priority Scheme Description

Similar to the priority system suggested in [4], we introduce a scheme which offers a multi-tiered access system for the MCNS specifications. However we face additional challenges since we cannot rely on the ternary-tree blocking algorithm to preserve a priority order when resolving collisions [2]. We devise our own algorithm to allocate more CS to each priority level when needed. In our protocol, areas of contention are defined for each priority level. The details of the mechanism are described next.

Priority Allocation MAP: In Figure 2 we suggest a new Allocation MAP format that supports our priority system. Each *Allocation MAP* message carries a priority for each CS allocated on the upstream channel. This insures that requests for upstream bandwidth from different priorities do not mix.

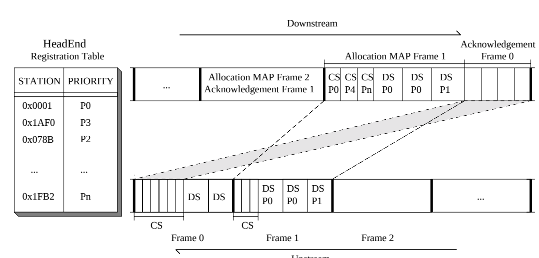


Figure 2: Priority Allocation MAP

Access Control: Priority stations use the CS assigned with their priority for initial access as well as for retransmissions. A station with a new request waits for a group of CS (one or more) with a priority that matches its own priority, and transmits the request with probability 1 within that group (i.e., the station randomly selects a CS slot within that priority window).

Collision Resolution: As described in Section 2, the MCNS specifications use binary exponential backoff in order to resolve collisions. We propose slight modifications to the collision resolution backoff scheme by giving a different backoff value to stations of high priority. The backoff value is set equal to the number of contention slots reserved for high priority stations. This insures that a high priority station retransmits its request in a timely manner in case of collision. All the other priorities use the binary exponential backoff scheme as defined in the MCNS standard.

Priority DS Allocation: A number of allocation schemes have been proposed in order to dynamically adjust the ratio of CS and DS [7]. The algorithm we use [6] is a slight variation of what is proposed by Sriram in [7]. The total number of CS, CS_F, contained in each upstream frame is dynamically adjusted as the headend converts a number of DS, DS_{CS}, into CS according to the following expression:

$$DS_{CS} = \lceil \frac{2 * DS_{F_{max}}}{(2 + m * k)} \rceil \quad (1)$$

where $DS_{F_{max}}$ is the maximum number of DS in a frame, m is the number of minislots that a data slot occupies, and k is the average number of DS's that can be requested at a time. As a result, CS_F can be determined by:

$$CS_F = \begin{cases} CS_{F_{min}} & \text{if } RQ \geq \alpha * (DS_{F_{max}} - DS_{CS}) \\ \min(CS_{F_{min}}, m * DS_{CS}) & \text{else} \end{cases} \quad (2)$$

where RQ is the length of the request queue at the headend, α is a design parameter set to 2.5 and $CS_{F_{min}}$ is the minimum number of CS in the frame. The headend can then satisfy a number of requests for bandwidth by allocating a number of DS, DS_F, according to a priority order using a preemptive round robin scheduling mechanism. Any unused DS resulting from the above allocation are then converted into CS as suggested by Sala in [1].

Priority CS Allocation: Given CS_F per frame, the next step is to compute CS_i, the number of CS to allocate to each priority i in a system with p priorities ($i \in [1..p]$). In such a system p is assumed to be the highest priority ($p > p-1 > p-2 > .. > 1$). We define G_i as the number of guaranteed CS per priority. G_i is set by the network operator in order to provide a minimum service access. Let $A_{i_{min}}$ be the number of CS computed with respect to $CS_{F_{min}}$ and the percentage of priority traffic a_i $\forall$ i, $i \in [1..p]$ according to:

$$A_{i_{min}} = \lceil a_i * CS_{F_{min}} \rceil \quad (3)$$

We note that $\sum_{i=1}^{p} A_{i_{min}}$ needs to be less than $CS_{F_{min}}$ in order for each priority to have at least G_i slots. Thus we compute a correction factor B_i in order to satisfy the following inequality:

$$\sum_{i=1}^{p} G_i < \sum_{i=1}^{p} A_{i_{min}} - \sum_{i=1}^{p} B_i \leq CS_{F_{min}} \quad (4)$$

Then for every i, we define a B_i that allows us to free extra CS from lower priorities. These slots can be used by higher priorities for resolving collisions. Let

$$B_0 = A_p * C_f + [\sum_{i=1}^{p} A_i - CS_{F_{min}}] + G_p - \min(A_p, G_p) \quad (5)$$

$$B_p = 0 \quad (6)$$

then $\forall$ i, $i \in [1..p-1]$ B_i is then given in terms of B_{i-1}:

$$B_i = B_{i-1} - [A_i - \max(A_i - B_{i-1}, G_i)] \quad (7)$$

In order to satisfy inequality (4) we limit the minimum number of CS available in each frame for priority i to Min_i. By combining A_i, B_i and G_i, we obtain:

$$Min_p = \max(A_p, G_p) \quad (8)$$

$$Min_i = \max(A_i - B_{i-1}, G_i) \quad (9)$$

CS_i, which is the number of CS given to class i, is then computed in terms of Min_i and the maximum number of slots to be used in each frame, Max_i, as follows:

$$CS_p = max\{min[\max(A_p, col_p * 2), Min_p], Max_p\} \quad (10)$$

and $\forall$ i, $i \in [1..p-1]$

$$CS_i = max\{min[\max(A_i, col_i * 2), Min_i], Max_i\} \quad (11)$$

where col_i is the number of collisions in the previous frame for priority i, and A_i is the number of slots needed with respect to the priority traffic proportion. A_i is given by:

$$A_i = \lceil a_i * CS_F \rceil \quad (12)$$

and Max_i is defined as:

$$Max_p = CS_F - \sum_{i=1}^{p-1} Min_i \qquad (13)$$

$$Max_i = CS_F - \sum_{j=1}^{i-1} Min_j - \sum_{k=p-(i+1)}^{p} CS_k \qquad (14)$$

3.2 Priority System Implementation

In this section we discuss the necessary adjustments to the MCNS specifications in order to implement our proposed priority scheme. Before getting into the discussion we should point out that our scheme can be easily supported by the standard. We use either reserved fields or user defined element types in order to integrate our algorithm. We identify two main issues in the priority system implementation. The first issue is providing a means for the station to request a priority level during registration. This idea is similar to service negotiation during connection setup in ATM networks. The other issue has to do with marking the CS with different priorities in order to perform the priority CS allocation described in the previous section. Note that DS need not to be marked since they are addressed to a specific station with a known priority. The details of the changes follow.

Station Registration - The station entry into the network occurs in several stages. After the station acquires synchronization of a downstream signal and completes the ranging and power leveling phase, it sends a registration request message (*REG REQ*) to the headend as shown in Figure 3. The MCNS standard supports several user defined classes of *REG REQ* that allows the user to request different types of services. We reserve one class type for requesting a priority service and we use its subclass to determine the priority value (8 bits to support up to 255 priority levels). Based on the priority value contained in the *REG REQ* message the headend can associate a priority with each station *ID* and construct a table mapping similar to the example in Figure 3.

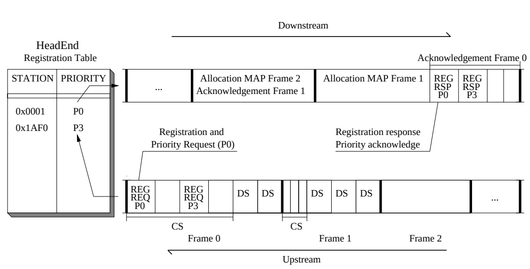


Figure 3: Modified Station Registration

The information contained in the table is used by the headend in order to compute various network operation statistics to support the priority service such as, computing a percentage of priority traffic for the CS allocation.

CS Priority Marking - In this case we use a reserved field in the *Allocation MAP* message, sent by the headend in order to mark the CS with a priority value as shown in Figure 4.

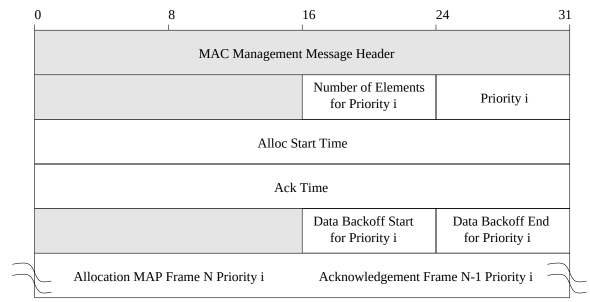


Figure 4: Modified Allocation MAP

Thus for each group of CS the headend specifies a priority, a start time of the group in the next frame (*Alloc Start Time*) and the *Ack* for the CS transmitted in the previous frame (if any). Also we use the fields reserved for the backoff window (*Data Backoff Start* and *Data Backoff End*) to specify the desired backoff value for each priority level. Note that by setting *Data Backoff Start* = *Data Backoff End* we can completely override the binary exponential backoff implemented at the stations by default.

4 Performance Evaluation

In this section we present performance evaluation results from simulating our proposed priority system.

4.1 Simulation Model

We have modified the HFC module of the NIST ATM/HFC Simulator [5] in order to implement our changes to the MCNS MAC protocol specifications. We used the configuration and system parameters for the HFC network shown in Table 1. All simulations, were run for 30 seconds of simulated time and the first 10% of the data was discarded. We present the results from four different simulation experiments that measure the effectiveness of the priority system using the mean access delay, the throughput and the access delay probability distribution.

Note that the mean access delay is the time it takes a packet to reach the headend from the time the packet arrives at the station. A summary of the experiments

	Low Priority		Medium Priority		High Priority		Overall
Experiment	Stations	Load	Stations	Load	Stations	Load	**Load**
Priority Access							
(a) IP Traffic	100	60%†	80	30%†	20	10%†	[5%,85%]
(b) Short IP Traffic	100	60%†	80	30%†	20	10%†	[5%, 85%]
Varying Priority Proportions	40	6.5%‡	80	[55.25%,3.25%]‡	80	[3.25%, 55.25%]‡	65%
Varying Medium Priority	100	20%‡	80	[10%,60%]‡	20	5%‡	[35%, 85%]
Varying High Priority	50	10%‡	50	10%‡	100	[5%,65%]‡	[25%, 85%]

Table 2: Simulation Scenarios
† As a percentage of the overall offered load ‡ As a percentage of the capacity (3 Mbits/s)

Simulation Parameter	**Values**
Distance from nearest/furthest station to headend	25/80 km
Downstream data transmission rate	Not considered limiting
Upstream data transmission rates	3 Mbits/sec
Propagation delay	5 μs/km for coax and fiber
Length of simulation run	30 seconds
Length of run prior to gathering statistics	10% of simulated time
Guard-band and pre-amble	Duration of 5 bytes
Data slot size	in multiple of 16 bytes
CS size	16 bytes
DS MAC Overhead	16 bytes
Frame size	36 mini-slots
Number of CS per Frame	Variable
Number of CS Guarantee per priority i (G_i)	$G_H = 2$; G_L=G_M = 1
Number of CS minimum per frame ($CS_{F_{min}}$)	8 CS
Roundtrip	1 Frame
Maximum request size	32 DS
Headend processing delay	0 ms

Table 1: Simulation Parameters

Message Size (bytes)	64	128	256	512	1024	1518
Probability	0.6	0.06	0.04	0.02	0.25	0.03

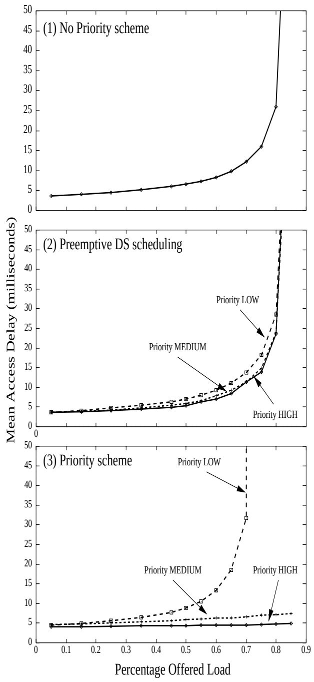
Table 3: IP Traffic: Message Size Distribution

is shown in Table 2. In all simulations we use three priority levels (high, medium and low). Unless mentioned otherwise the traffic type used is generated according to an IP message size distribution as shown in Table 3. The message interarrival time is exponentially distributed with mean $T = \frac{1}{\lambda}$ where λ varies according to the offered load. This traffic type is referred to as IP traffic in the remainder of the paper. In some cases (for example Experiment 1) we also use sources generating short IP messages (fixed to 64 bytes) according to a Poisson distribution with a mean arrival rate of λ, referred to as short IP traffic.

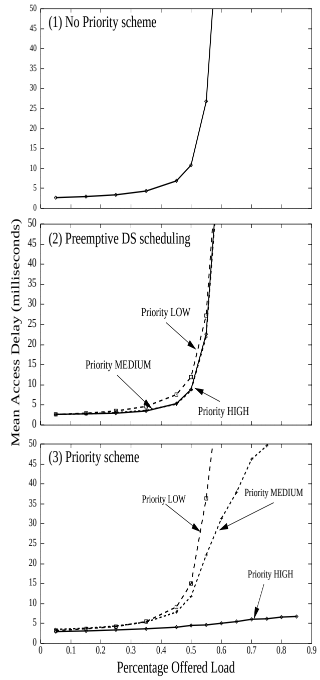
4.2 Experiment 1

The main objective in Experiment 1 is to show that a priority system is needed for the MCNS MAC access and to evaluate the performance of our priority scheme. We first plot the mean access delay for a system that does not implement any priorities (Figure 5(a)(1)) conforming to the current specifications [3]. This constitute a reference case to our study. Then we plot the mean delay using three priorities that we label low, medium and high in a system that supports only preemptive DS scheduling at the headend (Figure 5(a)(2)). In this case the headend preserves the priority ordering when it receives a successful request from the station. However it does not guarantee any priority access to transmit the request or resolve the collisions. There are 20 high priority stations, 100 low priority and 80 medium priority stations which contribute respectively 10%, 60%, and 30% of the offered load.

The effectiveness of our priority system is shown in Figure 5(a)(3) by comparing its performance with the previous cases. While the low priority mean access delay curve takes off a little earlier ($\sim$ 70% of 3Mbits/s in Figure 5(a)(3) as opposed to $\sim$ 80% of 3Mbits/s in Figure 5(a)(1) and (2)), we observe a relatively flat delay for both the medium and high priorities. The high priority mean delay remains almost constant even at high loads because our priority scheme allocates more contention slots for the medium and high priority levels. The low priority traffic in this case is treated as the best effort service and gets the remaining contention slots. Figure 5(b) shows the same results but with the short IP traffic type. We use this type of traffic in order to stress the contention access and demonstrate the adaptiveness of our priority allocation scheme. The maximum theoretical throughput bound for the delay in Figure 5(b)(1) is slightly lower than in Figure 5(a)(1) (delay curve takes off at $\sim$ $55\% of 3$Mbits/s). This is due to the extra overhead needed for short IP packets (overhead of 16 bytes for every 64 bytes while multiple 64 bytes need one MAC overhead for the IP traffic defined in Table 3).



(a) IP Traffic



(b) Short IP Traffic

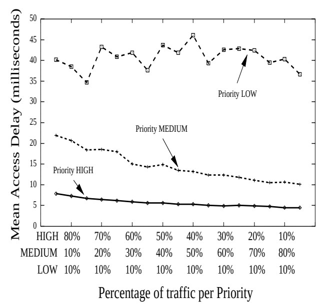
Figure 5: Experiment 1: Priority Access

In Figure 5(b)(3) we are still able to get almost constant mean delays for the high priority at high loads. The delays for the medium priority are not as good as with IP traffic (Figure 5(a)(3)). This is mainly due to the bursty nature of the short IP packet distribution. Overall our priority scheme maintains a stable performance regardless of the traffic type used.

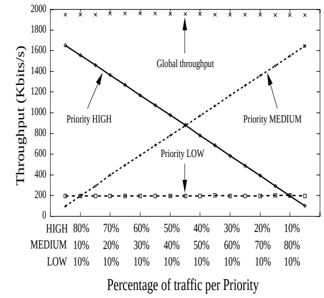
4.3 Experiment 2

To test the stability of our priority system and to confirm that the performance for each priority level does not depend on the amount of traffic it transmits, we fix the offered load to 65% of the upstream channel and vary traffic proportions among the various priorities. We keep the low priority at a constant percentage of 6.5% of the channel capacity (i.e. 10% of the offered load) and decrease the high priority from 55.25% to 3.25% of the capacity while increasing the medium from 3.25% to 55.25%.

The results of this experiments are given in Figure 6(a) for the mean access delay and Figure 6(b) for the throughput. In Figure 6(a) as the high priority traffic percentage is decreased, we observe a slight decrease in the access delay for the high priority ($\sim$ 2ms). The decrease in delay is more significant for the medium priority although the medium percentage of traffic is increased ($\sim$ 12ms). This phenomenon is mainly due to the decrease in the high priority traffic that requires



(a) Mean Access Delay

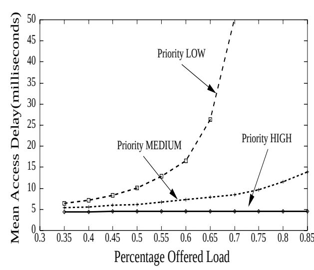


(b) Throughput

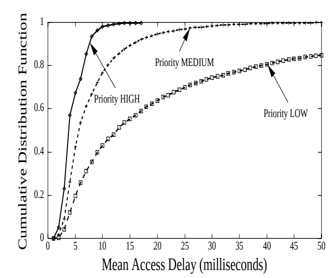
Figure 6: Experiment 2: Varying Priority Traffic Proportions for 65% of Channel Capacity

less CS reserved for the high priority in order to efficiently resolve collisions. Note that by design our scheme always favors the high priority traffic in order to keep the access delay constant. Figure 6(b) shows the throughput for the different priority levels. Each priority level is getting its share of the bandwidth according to its load (195Kbit/s constant for the low priority and 1950 Kbits/s for the overall throughput). The results of this experiment prove that our allocation scheme is robust and does not depend on a given priority load.

4.4 Experiment 3



(a) Mean Access Delay

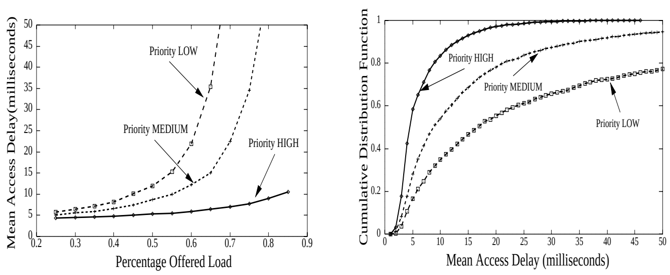


(b) Probability that Mean Access Delay $< x$ ms Offered Load= 65% of Capacity

Figure 7: Experiment 3: Varying Medium Priority

Experiment 3 shows the effect of varying the load from a particular priority level (in this case the medium priority) on the high priority. We use 20 high priority stations which contribute 5% of the channel capacity to the offered load and 100 low priority stations which represent 20% of the load. 80 stations are used for the medium priority which load is varied between 10% and 45% of the capacity. Figure 7(a) illustrates the mean access delay. As the medium priority traffic increases the headend allocates more CS for the medium priority contention and less for the low priority stations. This causes the mean delay of the low priority traffic to take off starting from 65%. In Figure 7(b) we plot the cumulative distribution function of the access delay for 65% of the capacity. This measurement helps us determine the variability of the access delay for a given load. We observe that for the high priority, the probability the access delay is less than 10 ms is almost 1, while it is close to 0.8 and 0.4 for the medium and low priority respectively. This result demonstrates the efficiency of our priority scheme in terms of providing low delay variations for the high priority level. This may prove to be critical for higher layer applications requiring low delay variance in addition to low end-to-end delay.

4.5 Experiment 4



(a) Mean Access Delay

(b) Probability that Mean Access Delay $< x$ ms, Offered Load= 65% of Capacity

Figure 8: Experiment 4: Varying High Priority

Experiment 4 shows the effect of varying the load of the high priority traffic. 50 low priority stations and 50 medium priority stations that contribute 20% of the channel capacity (10% each level). 100 high priority stations are varied between 5% and 65% of the capacity. Figure 8(a) depicts the mean access delay at different loads for all three priority levels. In order to keep the access delay low for the high priorities, less contention slots are allocated to the medium priorities. This results in significantly higher delays for the medium: 25ms at 70% of capacity as opposed to 10ms at the same load in Experiment 3, Figure 7(a). At 85% load there is small increase ($\sim$ 5ms) in delay for the high priority. However it is rather unlikely that a network could be operating with such a high percentage (65% of upstream channel) of high priority traffic. Figure 8(b) gives the cumulative distribution function of the access delay at 65%. In this case since the high priority traffic contributes more to the overall load the probability that the access delay is less than 10ms is almost 0.8 as opposed to 1 in Experiment 3 (Figure 7(b)). However we note that the tail of the cumulative distribution function for the high priority access delay is still finite and converges to 1 around $\sim$ 25ms. This is clearly the price to pay in the case there are more customers requiring services with stringent delay constraints.

5 Concluding Remarks

In this paper we presented a priority scheme for the MCNS MAC protocol. We provided a generalized algorithm and identified the changes needed in the MCNS specifications in order to implement it. We demonstrated the performance of our mechanism using a total of three priority levels. We used simulation results to prove that our proposed mechanism is robust and adapts well to various network conditions and traffic types. Efficiency is obtained by using priority ordering during contention access. Finally, the results obtained clearly show the advantages of using such a priority system that can differentiate between traffic classes where distinct levels of performance are desired.

References

[1] D. Sala, J. Limb, and S. Khaunte. Adaptive Control Mechanism for Cable Modems MAC Protocols. In *Proceedings of Infocom'98*, pages 1392–1399, San Fransisco, CA, March 1998.

[2] IEEE 802.14 Working Group. Media Access and Control. IEEE Std 802.14, Draft 3 Revision 1, IEEE 802.14 Working Group, April 1998.

[3] MCNS Holdings L.P. Data-Over-Cable Service Interface Specifications, Radio Frequency Interface Specification. SP-RFI-I02-9710008, October 1997.

[4] M. Corner, N. Golmie, J. Liebeherr, and D. Su. A Priority Scheme for the IEEE 802.14 MAC Protocol for Hybrid Fiber-Coax Networks. In *Proceedings of Infocom'98*, pages 1400–1407, San Fransisco, CA, March 1998.

[5] N. Golmie, F. Mouveaux, Y. Saintillan, A. Koenig, D. Su. The NIST ATM Network Simulator: Operation and Programming Version 4.0. National Institute of Standards and Technology, Internal Report 5703R4, November 1998.

[6] N. Golmie, Y. Saintillan and D. Su. A Review of Contention Resolution Algorithms for IEEE 802.14 Networks. In *IEEE Communications Surveys*, http://www.comsoc.org/pubs/surveys, First Quarter, 1999.

[7] K. Sriram. Performance of MAC Protocols for Broadband HFC and Wireless Access Networks. *Advances in Performance Analysis*, Vol.1(1):1–37, 1998.