A Priority Scheme for the IEEE 802.14 MAC Protocol for Hybrid Fiber-Coax Networks

Mark D. Corner, *Student Member, IEEE*, Jörg Liebeherr, *Member, IEEE*, Nada Golmie, *Member, IEEE*, Chatschik Bisdikian, *Senior Member, IEEE*, and David H. Su, *Member, IEEE*

***Abstract*—In order to support quality-of-service (QoS) for real-time data communications such as voice, video and interactive services, multiaccess networks must provide an effective priority mechanism. The context of this work is the IEEE 802.14 standard for hybrid fiber coaxial (HFC) networks which has a shared upstream channel for transmissions from stations to the headend. This work presents a multilevel priority collision resolution scheme, which separates and resolves collisions between stations in a priority order, thereby, achieving the capability for preemptive priorities. We present a set of simulation scenarios which show the robustness and efficiency of the scheme, such as its ability to isolate higher priority traffic from lower priorities and to provide quick access to high-priority requests. In March 1998, a framework for handling priorities in the collision resolution process, which adopts a semantics similar to the semantics of our scheme, was included in the 802.14 standard.**

***Index Terms*—Local area networks, quality-of-service.**

I. Introduction

EXISTING community cable television systems are evolving into bidirectional hybrid fiber coaxial (HFC) networks [18], [22] that can support interactive broadband applications, including video-on-demand, tele-conferencing, telephony, and Internet access. HFC is only one among several competing residential broadband access technologies, including digital subscriber line (xDSL), fiber to the home (FTTH), fiber to the curb (FTTC), fiber to the building (FTTB), local multipoint distribution service (LMDS), and wireless in the loop (WITL) [8], [10], [17], [22]. HFC networks are attractive as they can take advantage of the installed residential coax network's extensive coverage area. In comparison to xDSL, which takes advantage of installed telephone lines, HFC networks have significantly higher transmission capacity.

The residential cable network architecture uses a hierarchical tree-and-branch topology with as many as 2000 subscribers attached at the leaves of the tree. The coaxial wire portion of the network extends from a fiber-optic interconnected node to the subscribers' homes (see Fig. 1). The fiber node has a fiber connection to the so-called *headend*, which terminates the HFC network. All data coming from the subscribers is directed to the headend.

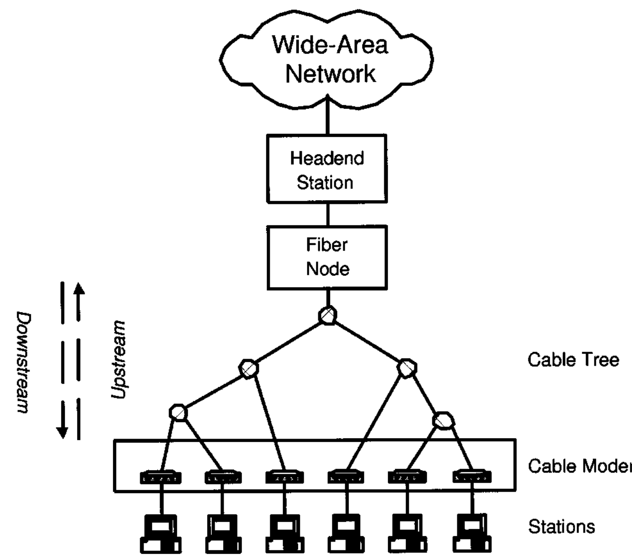


Fig. 1. HFC architecture.

The frequency spectrum on the coax wire portion of the network is divided into a downstream region and an upstream region. The downstream spectrum typically ranges from 50 to 750 MHz, divided into channels of fixed width, e.g., 6 MHz in North America, and 8 MHz in Europe. The upstream spectrum is in the range from 5 to 40 MHz with variable size channels typically from 1 to 3 MHz. At any time, a subscriber transmits only on one upstream channel and receives data only on one downstream channel. Data rates on the channels are approximately 3 Mbps and 30 Mbps in the upstream and downstream directions, respectively. Synchronization at the physical layer ensures that all subscribers have a common time reference.

Each upstream channel is a multi-access channel, and collisions occur when multiple subscribers, henceforth called *stations*, transmit simultaneously on the channel. All downstream channels are collision-free. Access to the upstream channel is a two-step process. If a station wants to transmit on the upstream channel, it first sends a transmission request to the headend. If more than one station transmits a request at the same time, the requests collide, and a collision resolution protocol (CRP) is activated to ensure successful retransmission of the request. If a station transmits a successful request, the second step begins.

Manuscript received June 8, 1998; revised January 26, 1999; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor T. Todd. The work of J. Liebeherr was supported in part by the National Science Foundation under Grant NCR-9624106 (CAREER) and Grant ANI-9730103.

M. D. Corner is with the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48109 USA (e-mail: mcorner@eecs.umid.edu).

J. Liebeherr is with the Department of Computer Science, University of Virginia, Charlottesville, VA 22903 USA (e-mail: jorg@cs.virginia.edu).

N. Golmie and D. H. Su are with the National Institute of Standards and Technology, Gaithersburg, MD 20899 USA (e-mail: nada.golmie@nist.gov; david.su@nist.gov).

C. Bisdikian is with IBM T. J. Watson Research Center, Hawthorne, NY 10532 USA (e-mail: bisdik@us.ibm.com).

Publisher Item Identifier S 1063-6692(00)03321-5.

The headend acknowledges the successful request, schedules a time slot on the upstream channel for data transmission, and sends the station a *grant message* to inform the station when it can transmit. Since the grant message never allocates the same time slot to more than one station, all transmission of data is collision-free. The IEEE 802.14 working group (WG) [6], [19] is currently standardizing the multiple access control (MAC) protocol for communication on an upstream channel. The MAC protocol is based on an n-ary stack resolution algorithm [3], [15].

In this paper, we investigate the ability of the 802.14 MAC protocol to provide priority access to stations. An effective priority system is needed to provide quality-of-service (QoS) in HFC applications and services such as voice, video, and interactive data services [5], [13]. While, from the outset, the capability to support priority-based data transmissions was present in the IEEE 802.14 draft standard, priorities for contention-prone transmission requests were not considered. In this paper we will demonstrate that the absence of priority support during the collision resolution process has a negative impact on the effectiveness of the priority scheme. It will be shown that, in order to provide effective handling of priority traffic on a reservation-based system, like the n-ary stack resolution algorithm, one needs to support priorities throughout all transmission phases, including both the request phase and the actual data transmission phase. We present a scheme that can support priorities during contention resolution for tree-search (stack) contention-resolution algorithms. It is worth noting here that the 802.14 WG has accommodated a framework for the handling of priorities of contention-prone request transmissions. The framework in the draft standard enables our priority scheme by simply changing the syntax of our scheme, without changing its semantics.

The remainder of the paper is organized as follows. In Section II we discuss the 802.14 MAC protocol without priorities. In Section III we show why a priority scheme which does not differentiate priorities during the collision resolution process is not effective, and we describe a new MAC level priority scheme. In Section IV we present a set of simulation scenarios that show the performance of our priority scheme. In Section V we offer some conclusions. In the Appendix, we describe the priority mechanism adopted by the 802.14 WG, in the terms of our priority scheme.

II. The 802.14 MAC Contention Resolution Protocol

In this section we review the operation of the IEEE 802.14 MAC protocol. Since our priority mechanism, to be described in Section III, is developed within the the context of the 802.14 MAC [19], an understanding of the protocol is essential for the description of our priority system.

A. MAC Operation

An HFC upstream channel (see Fig. 2) is divided into discrete time slots, called minislots. The headend designates some of the minislots as contention slots (CS) and some as data slots (DS). Contention slots, which are one minislot long, are used to transmit requests for bandwidth. Data slots, which are several minislots long, are used to transmit data. Only contention slots

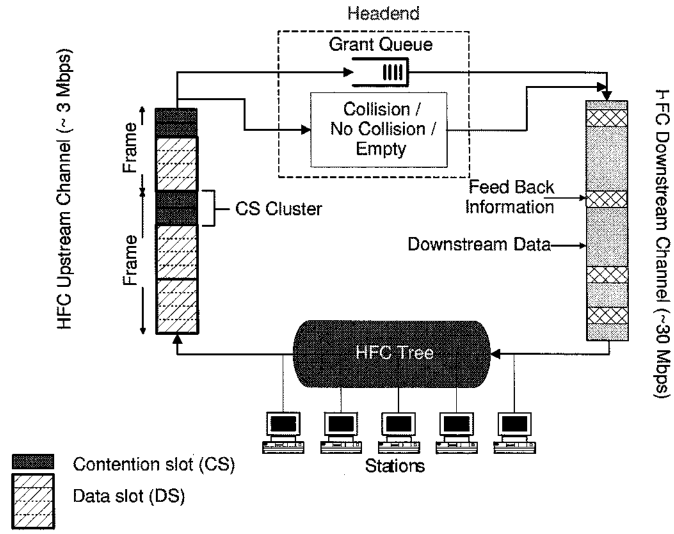


Fig. 2. Media access control in HFC networks.

are prone to collisions, which occur when more than one station attempts to transmit a request in the same slot. Data slots are explicitly allocated to a specific station by the headend and are collision-free. The headend controls the use of contention slots by assigning request queue values (RQ values) to each contention slot.

The headend's control of the MAC protocol uses a logical grouping of several CS and several DS into a *frame* (see Fig. 2). The set of CS in a frame, called a *CS cluster*, are located at the beginning of the frame. After the headend has completely received a CS cluster, it uses the downstream channel to send feedback on each CS to the stations. The feedback indicates whether a CS was empty, successful, or contained a collision. Also, the feedback contains, for each CS, an RQ value assigned by the headend.

For subscriber stations, the 802.14 MAC protocol specifies a multi-step procedure for gaining access to the upstream channel. A station with a new request for bandwidth, called a *newcomer* station, transmits a request for bandwidth using a so-called *first transmission rule (FTR)* [2]. The FTR specifies that the station must wait for a group of contention slots with $\mathrm{RQ} = 0$ (A slot with $\mathrm{RQ} = 0$ is called a *newcomer slot*). The station then picks a number, p, between 0 and a range parameter R. If the CS cluster has more than p slots with $\mathrm{RQ} = 0$, the station transmits the request in the pth slot. Otherwise, the station waits for the next cluster of newcomer slots and tries again. The range of the initial backoff, R, is used to avoid a physical layer complication, called laser clipping, which occurs when a large number of stations transmit in the same slot [7], [9]. Clearly, the backoff also reduces the likelihood of a collision in the case of multiple newcomer stations.

If two or more stations transmit requests in the same contention slot, the headend executes a CRP, which, in the case of the 802.14 MAC, is a blocking ternary-tree algorithm [3], [4], [15].[1] *"Blocking"* refers to the restriction that newcomer stations may not transmit in CS designated by the headend for the

[1]The protocol is actually a variable n-*ary stack algorithm*. The default value of n is 3, and for simplicity of presentation, we will use the default value.

resolution of collisions [2]. "*Ternary tree*" refers to a three-way splitting of each collision. A complete review of this and other collision resolution schemes can be found in [2], [20].

The headend maintains a tree data structure, called *collision tree*, to maintain state information on the collision resolution process.[2] [3] When a frame arrives at the headend, the headend performs the following operations for each collision in the frame.

It assigns an RQ number to the collision, set to one larger than the currently highest RQ value. This RQ value will be assigned to all stations involved in the collision. Then the headend adds three nodes to the collision tree (*ternary split*) and labels the nodes with the collision's RQ value. The labeling of the nodes in the collision tree is used to obtain the RQ assignment for slots in the next CS cluster. The headend sends feedback to the stations on the status (=*empty, no collision, collision*) of the slots, the RQ values assigned to stations, and the RQ values of the slots in the next CS cluster. The combination of RQ values assigned to stations and to slots in the next CS cluster allows the headend to control access to the CS. The process is described in further detail in an example in Section II-B. We refer to [16] for a complete description of the protocol.

B. Collision Resolution Example

Fig. 3 shows an example of the collision resolution process for a network with nine stations, labeled A through I. On the left-hand side of Fig. 3(a) we depict a frame that arrives at the headend. The frame contains 7 contention slots and an unspecified number of data slots.[4] Each contention slot is marked with an RQ value and the labels of stations that attempt to transmit in this slot. Initially, all slots are set to $\mathrm{RQ} = 0$, meaning that all slots are available to newcomer stations. If no station transmits in a slot, the slot is labeled "—". A slot contains a collision if it is marked with more than one letter.

In Frame 1, shown in Fig. 3(a), stations A and B both send a request in the first slot, thus, causing a collision. Station C makes a successful request in the third slot, and stations D, E, F, G collide in the sixth contention slot. Starting at the last collision, each collision is assigned an RQ value. The RQ value assigned is one larger than the currently highest RQ value. Thus, $\mathrm{RQ} = 1$ is assigned to the second collision, and $\mathrm{RQ} = 2$ is assigned to the first collision. The right-hand side of Fig. 3(a) depicts the collision tree after the frame arrived at the headend (We assume that the collision tree is empty before the frame arrives). For each collision, a group of three nodes has been added to the collision tree, and the nodes are labeled with the RQ value of the collision.

When the headend sends feedback for the CS cluster in Frame 1 on the downstream channel, the feedback contains the RQ values assigned to the collisions. Thus, stations A and B will be assigned $\mathrm{RQ} = 2$, and D, E, F, and G are assigned $\mathrm{RQ} = 1$.

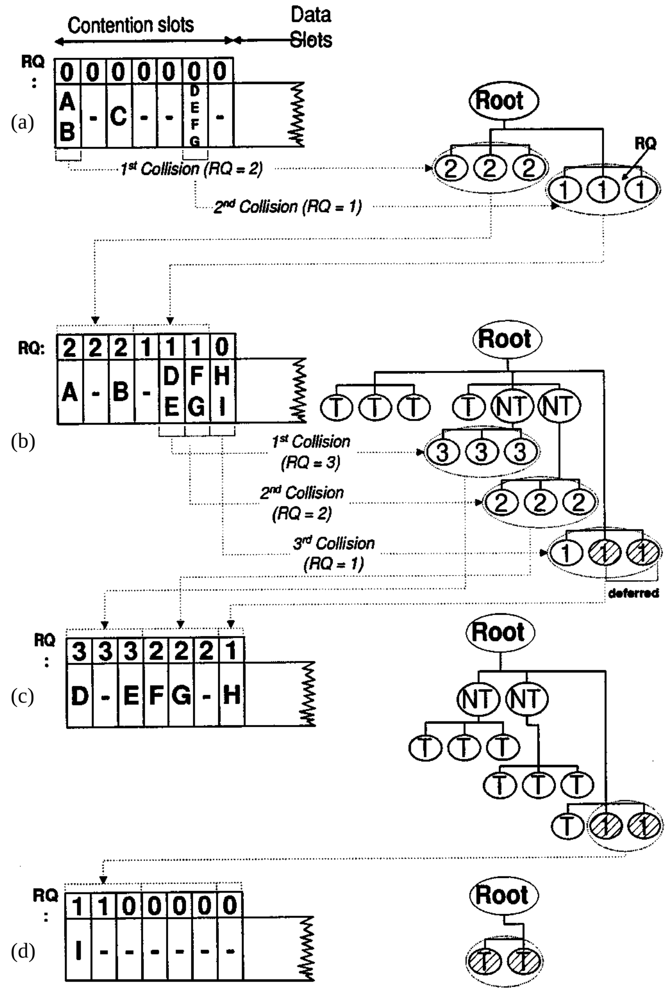


Fig. 3. Ternary tree collision resolution. (a) Arrival of Frame 1 at headend and resulting collision tree. (b) Arrival of Frame 2 at headend and resulting collision tree. (c) Arrival of Frame 3 at headend and resulting collision tree. (d) Arrival of Frame 4 at headend and resulting collision tree.

After building and labeling the collision tree, the headend uses the collision tree to assign RQ values to the slots of the CS cluster in the next frame, Frame 2. The first slot in Frame 2 receives the RQ value of the leftmost leaf node in the collision tree, the second slot is given the RQ value of the second leaf node, and so forth. Fig. 3(b) shows the result of the RQ value assignment for Frame 2. The first three slots are assigned $\mathrm{RQ} = 2$, slots 4–6 are assigned $\mathrm{RQ} = 1$. The remaining slots, one in Fig. 3(b), are assigned $\mathrm{RQ} = 0$.

The 802.14 MAC protocol enforces that a station can only transmit in a contention slot if the RQ value of the slot matches its own RQ value. (During contention resolution after a collision, a station can use a contention slot with an RQ value equal to or less than its RQ value.) If several slots match the RQ value, the station makes a random selection. In Frame 2, shown in Fig. 3(b), stations A and B could select any of the slots with $\mathrm{RQ} = 2$. Here, they select the first and third slot, respectively. Stations D and E, both with $\mathrm{RQ} = 1$, both transmit and collide in the fifth slot, and F and G collide in the sixth slot. The

[2]We emphasize that this tree is a data structure used by the headend to assign RQ values, and is *not* related to the tree-and-branch topology of the cable system.

[3]As an alternative to a tree, the collision resolution can be performed with a stack data structure. One can show that collision resolution with tree and stack are isomorphic to each other [15].

[4]The number of contention slots in a frame is not specified by the 802.14 MAC. An algorithm for determining this number can be found in [12] and [21].

seventh slot is still open for newcomer stations $(\mathrm{RQ}=0)$ and newcomer stations H and I collide in it.

The RQ values assigned to the collisions are $\mathrm{RQ}=3$ for the first collision, $\mathrm{RQ}=2$ for the second collision, and $\mathrm{RQ}=1$ for the third collision. Thus, when the headend sends feedback for Frame 2, stations D and E will be set to $\mathrm{RQ}=3$, stations F and G will be set to $\mathrm{RQ}=2$, and stations H and I will be set to $\mathrm{RQ}=1$.

The right-hand side of Fig. 3(b) depicts the collision tree after Frame 2 has arrived at the headend. Any leaf that corresponds to a slot which does not contain a collision is considered *terminated*, labeled with a "T", and eliminated from the tree. Leaf nodes that contain a collision are considered *not terminated*, labeled with "NT", and obtain three children nodes. If a collision occurred in a slot with $\mathrm{RQ}=0$, three leaf nodes are split from the root.

The RQ values for the CS in Frame 3 are assigned according to the labels of the leaf nodes in the collision tree: slots 1–3 are assigned $\mathrm{RQ}=3$, slots 4–6 are assigned $\mathrm{RQ}=2$, and the remaining slot is assigned $\mathrm{RQ}=1$ [see left picture in Fig. 3(c)]. Note that the number of not terminated leaf nodes $(=9)$ is larger than the number of available contention slots $(=7)$. Thus, two slots with $\mathrm{RQ}=1$ which do not fit into the next frame will be deferred until Frame 4.

The left-hand side of Fig. 3(c) shows that Frame 3 has no collision. Thus, all nodes of the collision tree, except the two nodes with $\mathrm{RQ}=1$ for the deferred slots, are terminated. To accommodate the deferred slots, two contention slots with $\mathrm{RQ}=1$ are allocated in Frame 4 [see Fig. 3(d)]. The remaining contention slots in Frame 4 are set to $\mathrm{RQ}=0$, meaning that they are available to newcomer stations. In Fig. 3(d), one of the slots with $\mathrm{RQ}=1$ is randomly selected by station I to transmit its request. After Frame 4, all leaf nodes are terminated, hence, all collisions are resolved. In the subsequent frame (not shown), all CS slots will be labeled with $\mathrm{RQ}=0$.

III. MULTIPRIORITY ACCESS SCHEME FOR HFC NETWORKS

The IEEE 802.14 MAC protocol provides three possible places that can be used to implement a priority scheme:

1) use priorities for the FTR;
2) use priorities in the CRP;
3) use priority scheduling at the headend when granting transmission of data slots.

Originally, the 802.14 MAC only used priority scheduling at the headend. We first motivate the need for a better priority system by showing that headend scheduling alone is not sufficient. Then we propose our solution, which employs prioritization also in the FTR and the CRP. Throughout this work we maintain priority scheduling at the headend.

A. Motivation for a Priority System

In earlier versions of the IEEE 802.14 draft specifications, stations indicate the priority of their traffic type through a queue identifier (QI) field in the contention slot. The headend uses a priority scheduler for the transmission queue of grant messages for those stations which have indicated a high-priority in the QI field. With priority scheduling, a station which has transmitted a successful priority request can gain faster access to the channel.

There are two fundamental shortcomings with this scheme. The first shortcoming is that the FTR treats all stations equally, regardless of their priority. Therefore, a high-priority request may be blocked due to an ongoing collision resolution of low-priority requests. The problem can be illustrated in the example shown in Fig. 3. Recall that each collision splits across three CS in the next frame. Consider a situation as in Frame 3, where all contention slots are used for resolving collisions, and, therefore, no newcomer slots with $\mathrm{RQ}=0$ are available. In such a situation, the absence of newcomer slots blocks the station with a high priority from transmitting the request.

The second problem with the priority scheme results from the fact that the MAC does not resolve collisions in a priority order. This problem, too, can be illustrated in Fig. 3. Consider Frame 3 and suppose that station D is requesting bandwidth for high-priority traffic, while station E is requesting bandwidth for low-priority traffic. Here, station D has to send its request in contention with the low-priority station E. Since there is no consideration of priorities during collision resolution, it may happen that the request by the low-priority request from station E is completed before the high-priority request from station D.

The lack of priority support for contention-prone transmission requests in the 802.14 MAC is reflected in the MAC delays of stations in the HFC network.[5] To demonstrate this, we present a simulation experiment.[6] In the simulated scenario, three groups of stations are present, each with a different priority level. We use the convention that a lower priority index indicates a lower priority. The three groups are comprised of the following stations and loads: a group of 20 priority-2 stations with an aggregate load of 5% of the upstream capacity, a group of 80 priority-1 stations with a load that is varied from 0–32%, and a group which consists of 100 priority-0 stations with an aggregate load of 20%. In the simulation experiment, we measure the 95th percentile of the MAC delay.

In Fig. 4, we show the MAC delays for the 802.14 MAC which uses a priority scheduler at the headend, but which does not use priorities for the FTR and CRP. Fig. 4 clearly shows that the MAC delays increase for all priorities as the priority-1 stations increase their load. Higher priorities do no receive smaller MAC delays.

For comparison, we show the same simulation scenario using our priority system (presented in the next subsection). Fig. 5 shows that our scheme separates traffic from different priority levels. As the load from priority-1 traffic is increased to more than $>$15% of the upstream capacity, the delay of priority-0 traffic increases drastically, while the delays for priority-1 and priority-2 traffic is controlled. When the load of priority-1 traffic is further increased, beyond $>$30% of the upstream capacity, the priority-1 traffic is practically preempted, without affecting the delay of priority-2 traffic.

[5] *MAC delay* is the time from the arrival of data to the station until the successful transmission of the data. We assume that data that arrives at the MAC layer of a station is small enough to fit into a single data slot. The MAC delay includes the waiting time for a newcomer slot, delays due to collision resolution, scheduling delay of the grant message at the headend, and transmission delay of the data slot.

[6] The experiment is similar to the simulations in Section IV. Refer to Section IV for a complete discussion of the simulation parameters and the simulated network.

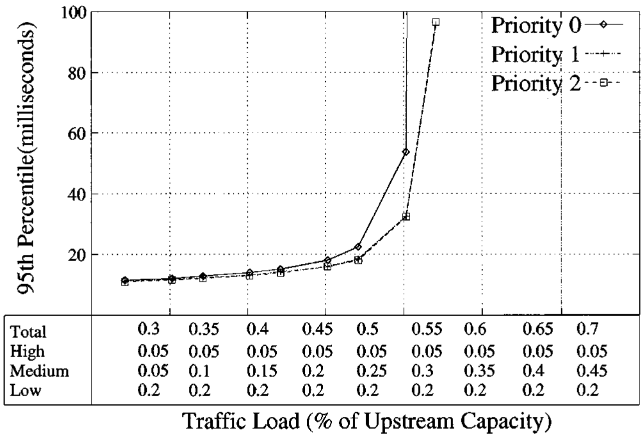


Fig. 4. The 95% percentile of MAC delays (without our priority MAC protocol).

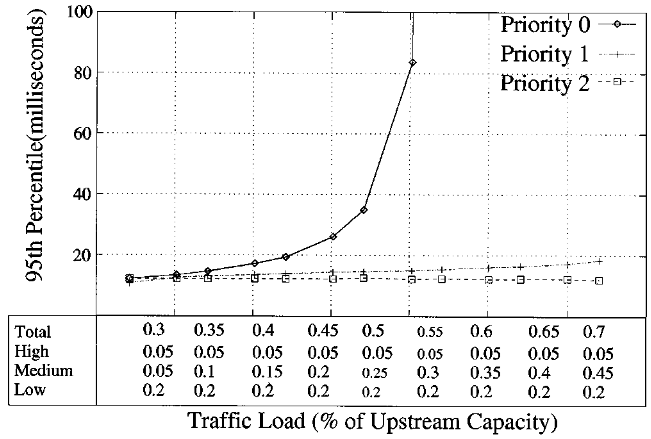


Fig. 5. The 95% percentile of MAC delays (with our priority MAC protocol).

Remark: We want to point out that the outcome of the experiment in Figs. 4 and 5 is dependent on the relative capacities of the reservation channel and the data channel. If the reservation channel capacity is large, a backlog will build at the headend, rather than in the stations. In such a case, the headend scheduling algorithm has a larger impact on the effectiveness of traffic prioritization. Note that in the depicted scenario, the reservation channel capacity is small relative to the capacity of the data channel.

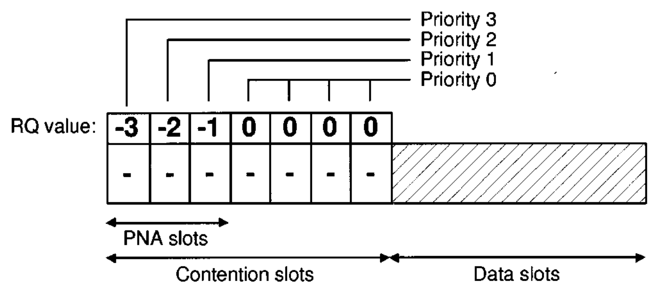


Fig. 6. Priority frame layout.

B. Priority Protocol Description

We propose a priority scheme which addresses the problems of the 802.14 MAC pointed out in the previous subsection. By allowing high-priority stations to bypass the blocking feature of the CRP and by separating the first transmission rule and collision resolution for different priorities, we show that contention can be confined to the set of stations in the same priority level.

New Frame Format: In Fig. 6 we suggest a new frame format. Several contention slots at the beginning of the frame are converted for exclusive use by priority stations. Each of these contention slots, referred to as a *Priority Newcomer Access (PNA)* slots, correspond to a single priority level. The headend identifies a PNA slot with a negative RQ value, where the RQ value $-N$ is reserved for priority level N.[7] We assume that a larger priority index indicates a higher priority, with "0"denoting the lowest priority. For example, an RQ value of -3 indicates that the slot is reserved for priority level 3. With the PNA slots, we can ensure that each priority level (other than level "0") can send transmission requests to the headend without interference from other priorities. Thus, higher priority stations are never completely blocked from transmitting a request for bandwidth.

New First Transmission Rule: The 802.14 MAC uses a blocking FTR to prevent newcomer stations from interfering with an ongoing collision resolution. Newcomer CS are designated with an RQ value set to $\mathrm{RQ} = 0$. Using parameter R, stations distribute their requests randomly over the number of available newcomer CS.

[7]In the 802.14 draft, RQ values are represented by an 8-bit integer. If the number is interpreted as a two's complement quantity, we can still support more than 100 $(=2^7)$ priorities.

We define a new FTR which enforces that stations with priorities >0 must use their PNA slot for all newcomer station access. The original FTR is used only by stations of the lowest priority $(=0)$. With the new FTR, stations with higher priority requests can immediately transmit requests for bandwidth in the PNA slots. A station with a new request waits for a PNA slot with a priority that matches its own priority, and transmits the request with probability 1.

Note that the new FTR always runs into a collision if two or more stations from the same priority level (>0) transmit a request in the same frame. Seemingly, this is a disadvantage as the necessary collision resolution increases the MAC delay. In reality, however, a high-priority collision is a fast method to signal to the headend that more contention slots are needed for exclusive use by high priorities. Additionally, in Section IV we show that the availability of additional PNA slots does not lead to improved delay performance for high-priority traffic.

Separate Collision Resolution for Each Priority: We completely separate the collision resolution at each priority level. Newcomer stations with priority requests (>0) transmit requests only in slots which exactly match their priority level. Therefore, the headend knows that all stations involved in a particular collision are of the same priority level. In case of a collision at a certain priority level, the headend attempts to allocate three slots in the next frame for each collided slot; each one of these slots is reserved exclusively for requests from the same priority as the collision. Hence, requests only collide with other requests from the same priority.

Contention Slot Allocation: The number of contention slots available in a frame may not be sufficient to accommodate all the slots needed for ongoing collision resolution and newcomer access. We have seen such a scenario in the single-priority scenario in Fig. 3(c), where two needed contention slots could not be accommodated and had to be deferred to a later frame. In our prioritized slot allocation scheme, the headend follows a priority order when determining which slots are allocated in the next frame, and which are allocated in a later frame. Given that P is the highest priority, the order is as follows: (1) Collision resolution slots for priority stations at level P, (2) PNA slot for level P, (3) Collision resolution for level $P-1$, (4) PNA for level $P-1$, and so on. Any leftover slots are allocated with $\mathrm{RQ}=0$, to be used by lowest priority newcomer stations. This ordering gives highest priority to the collision resolution of the highest priority level. If the number of contention slots is not sufficient, lower priority collision resolution slots are preempted, and deferred to later frames.

Note that the complexity of the described priority algorithm is very small. A newcomer of a certain priority class simply uses the slots designated for the priority of its class. Also note that the presented priority collision resolution algorithm is orthogonal to the selection of the priority scheduling algorithm at the headend. Our priority algorithm does not restrict the selection of the priority scheduling algorithm at the headend.

The scheme as described here integrates priority slots directly into the 802.14 frame format. The use of an extra slot to indicate high-priority traffic was first proposed for XDQRAP [14], however, not in the context of the ternary tree algorithm of the 802.14 MAC. As opposed to the fixed frame format found in XDQRAP, the flexible frame size of the 802.14 standard allows our protocol to allocate more contention slots to each priority level when needed.

C. Example Priority Collision Resolution

We demonstrate the collision resolution process using our priority scheme with an example. The scenario is depicted in Fig. 7. We assume four priority levels, where "3" is the highest priority, and "0" is the lowest priority. We use seven stations, labeled A through G, with

priority level 3 for stations A and B,
priority level 1 for station C, and
priority level 0 for stations D, E, F, and G.

Each frame consists of seven contention slots and an unspecified number of data slots. Assuming that the system has been idle long enough, so no previous collision is currently being resolved, the headend will set the RQ values in the priority frame as shown in the left picture of Fig. 7. Recall that a negative RQ value $-P$ designates the contention slot as a PNA slot of priority level P. The first three contention slots with RQ values $-3, -2, -1$ are PNA slots for priority levels 3, 2, and 1, respectively. The remaining slots are assigned a priority level of 0.

In Frame 1, in Fig. 7(a), newcomer stations A and B both transmit requests at priority level 3. Thus, they transmit their first request in the PNA slot with $\mathrm{RQ}=-3$, causing a collision. Station C transmits a successful request for priority 1 traffic in the PNA slot with $\mathrm{RQ}=-1$. Stations D, E, F, and G all transmit a request for priority level 0 in the same slot with $\mathrm{RQ}=0$. On

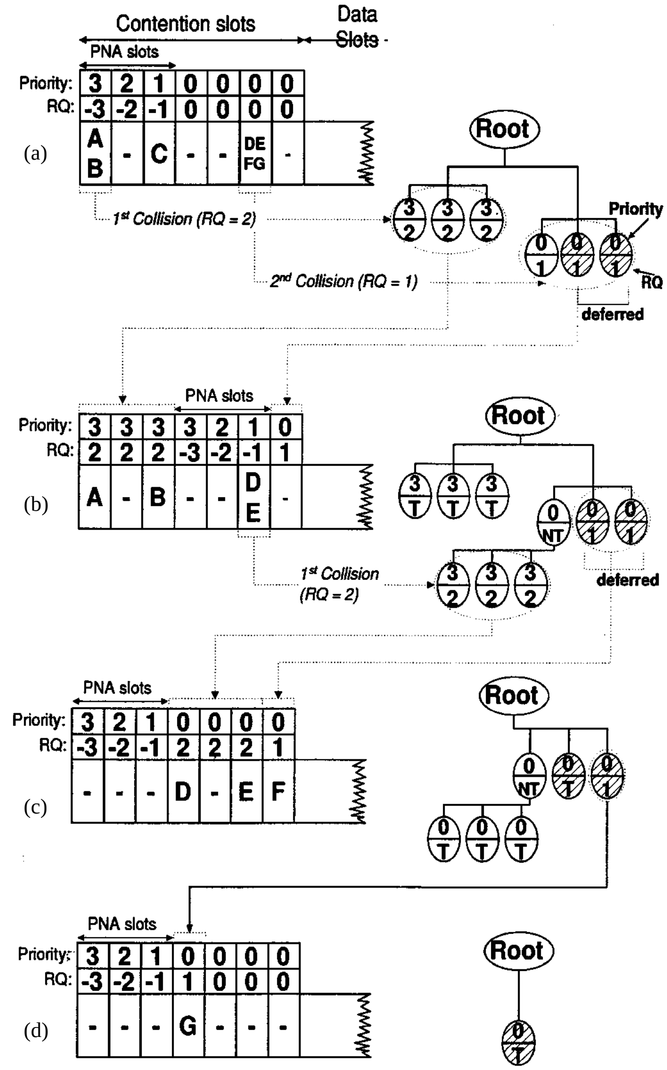


Fig. 7. Priority collision resolution. (a) Arrival of Frame 1 at headend and resulting collision tree. (b) Arrival of Frame 2 at headend and resulting collision tree. (c) Arrival of Frame 3 at headend and resulting collision tree. (d) Arrival of Frame 4 at headend and resulting collision tree.

the right-hand side of Fig. 7(a), we show the collision tree for Frame 1. For each collision, three new nodes are created, and the nodes are labeled with an priority index and an RQ value. The priority index of a node is identical to the priority index of the slot where the collision occurred. The RQ values are set as in the uni-priority case, that is, the RQ value is incremented for each collision.

In the second frame [Fig. 7(b)] we see how the headend assigns RQ values and priorities to the next CS cluster. Each collision is split across three slots, each with the same priority as the collided slot they are generated from. The PNA slots for each priority level are still allocated to provide newcomers from higher priorities access to a request slot. Note that there are not enough slots in the CS cluster to accommodate all slots needed for collision resolution. Thus, two slots, corresponding to the two rightmost leaf nodes in Fig. 7(b), will be deferred to a later frame.

In Frame 2, stations A and B select different contention slots with $\mathrm{RQ}=2$. Stations D and E both transmit in the priority-0 slot with $\mathrm{RQ}=1$, and experience another collision. Stations

F and G have randomly selected one of the deferred slots, and, therefore, do not retransmit their request in Frame 2. The collision tree for Frame 2 shows that the collision with $\mathrm{RQ} = 2$ is resolved and is labeled as terminated. Due to the collision of stations D and E, three more nodes are created in the tree. Since the collision with $\mathrm{RQ} = 1$ is still not resolved, the nodes are labeled $\mathrm{RQ} = 2$.

The priority and RQ value assignment for Frame 3, shown in Fig. 7(c), is directly obtained from the collision tree. Due to the PNA slots, only 4 slots are available for the 5 nodes in the collision tree. Thus, the rightmost leaf node in Fig. 7(c) is again deferred to the next frame. The retransmission of requests in Frame 3, by stations, D, E, and F shows that there is no collision. As a result, all nodes, except the deferred node, are labeled as terminated. In the last frame, shown in Fig. 7(d), all stations complete their requests and the system returns to the idle state.

The IEEE 802.14 WG followed a proposal by [1] and adopted a general framework for a priority collision resolutiow scheme, which is now part of the draft standard [19]. The standardized framework uses similar semantics for resolving priority collisions as the scheme just presented; but it takes a more general form and uses a different syntax. For example, the standardized scheme does not use RQ values to designate the priority of CS in the system. Rather, it uses CS/DS allocation messages to designate certain newcomer CS to one of eight priority levels. Also, the adopted version does not give guidelines for allocating priority newcomer slots. (Our scheme allocates them in every frame.) The standardized version is discussed in the Appendix.

TABLE I
SIMULATION PARAMETERS

Simulation Parameter	Values
Distance from nearest/furthest station to headend	25/80 km
Downstream data transmission rate	Not considered limiting
Upstream data transmission rates (only one upstream channel is used)	3 Mbps
Propagation delay	5 μs/km for coax and fiber
Length of simulation run	10 sec
Length of run prior to gathering statistics	10% of simulated time
Guard-band and pre-amble between transmissions from different stations	Duration of 5 bytes
Data slot size	64 bytes
Payload in a data slot	48 bytes
CS size	16 bytes
DS/CS size ratio	4:1
Frame size	52 minislots
Size of CS cluster	Fixed 18 slots
Round trip	1 Frame
Maximum request size	32 data slots
Headend processing delay	1 ms

IV. PERFORMANCE EVALUATION

We have built a simulation program to evaluate the performance of the priority system. The simulation program was created as an HFC module for the NIST ATM simulator [11]. We used the configuration and system parameters for the HFC network shown in Table I.

We present the results from six different simulation experiments that measure the effectiveness of the priority system using average request delay in Experiments 1–4, and transient throughputs in Experiment 5–6. The request delay is the time it takes a transmission request to successfully reach the headend from the time the request arrives at the station. Different from the MAC delay, as defined in footnote 5, the request delay does not measure delays that are incurred after the successful transmission of a request, i.e., scheduling delay of the headend and transmission time of data slots are not included.

In all simulations, the maximum number of priority levels is set to three. Priority 0 is the lowest priority level, and priority 2 is the highest priority level. In each experiment we have groups of stations which transmit at a given priority level. A summary of the number of stations in each priority group, and the load from each priority group, expressed in percentage of the upstream capacity, is shown in Table II. Traffic arrivals to a station are following a Poisson process and each arrival requires a single data slot. If multiple data slots arrive before a station can transmit a request, a station will issue a request for multiple data slots.

- In Experiments 1 and 2, we show the impact of increasing the load of one priority on the request delays of the other priorities. Experiment 1 varies the priority-1 load, and Experiment 2 varies the priority-2 load. As the traffic from a particular priority is increased, traffic from lower priorities is expected to be preempted. At the same time, high-priority traffic should not be affected.
- In Experiment 3, we attempt to quantify the overhead caused by the allocation of PNA slots in each frame in a network with only one priority level. We compare the request delay for low-priority traffic in a system with PNA slots to a system without PNA slots. Since no high-priority traffic is present, this experiment evaluates the overhead due to the priority system.
- In Experiment 4, we evaluate the bandwidth that should be reserved for priority newcomer stations. We verify that our selection of only one PNA slot per frame is sufficient. Note that priority-1 and priority-2 stations are given only one newcomer slot, the PNA slot, while priority-0 stations are given the remaining slots with $\mathrm{RQ} = 0$ in the frame.
- In Experiment 5, we evaluate how fast our priority scheme can preempt lower priority traffic when higher priority traffic becomes active. We also verify that the priority system is fair within the same priority level.
- In Experiment 6, we show how the preemptive priority scheme can be relaxed by using a different scheduling algorithm in the grant queue of the headend. Specifically, we demonstrate the implementation of a rate-proportional bandwidth sharing scheme which allocates bandwidth to priority levels in a specified ratio.

A. Experiment 1: Varying Priority-1 Load

Our goal is to investigate the impact of increasing the load of a particular priority level on the other priority levels. In the ex-

TABLE II
SIMULATION SCENARIOS (THE LOAD IS EXPRESSED IN PERCENTAGE OF THE UPSTREAM CAPACITY)
†Continuous backlog ‡50 in Group 1 and 50 in Group 2

		Priority 0 (low priority)		Priority 1 (medium priority)		Priority 2 (high priority)	
#	Experiment	Stations	Load	Stations	Load	Stations	Load
1	Vary Priority-1 Traffic	100	20%	80	[10%,45%]	20	5%
2	Vary Priority-2 Traffic	50	12.5%	50	12.5%	100	[10%,45%]
3	Protocol Scheme Overhead	80	[5%,45%]	0	0	0	0
4	Varying Number of PNA Slots	50	[2.5%,35%]	50	[2.5%,35%]	0	0
5	Transient Throughput	50	100%†	100‡	100%†	50	100%†
6	Transient Throughput	50	100%†	100‡	100%†	50	100%†

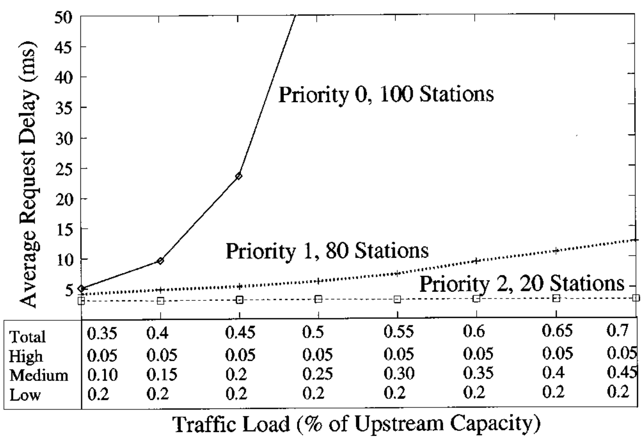


Fig. 8. Experiment 1: Varying priority-1 load.

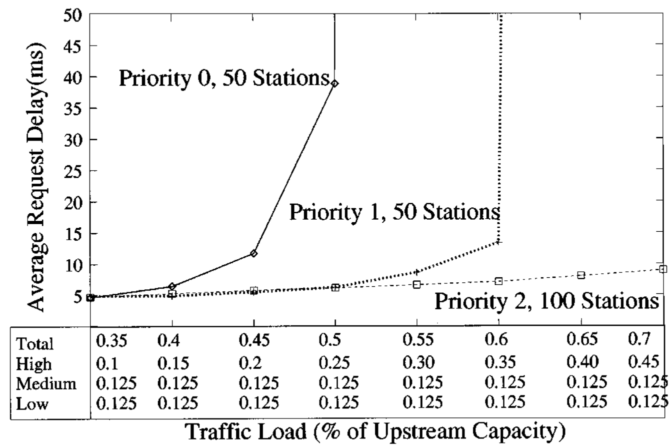


Fig. 9. Experiment 2: Varying priority-2 load.

periment, a total of three priority levels are used. There are 20 priority-2 stations which contribute 5% of the channel capacity to the load and 100 priority-0 stations which transmit a total load of 20% of the upstream capacity. 80 priority-1 stations are introduced to the system generating a load that is varied from 10% to 45%. In Fig. 8, we plot the request delay versus load for each priority level. We observe that, as the priority-1 traffic increases, the delay of the priority-0 stations rises sharply, while it increases moderately for the priority-1 traffic. Since during collision resolution, the headend allocates more contention slots for the priority-1 contention and less for the priority-0 stations, the delay of priority-0 traffic increases while the delay of priority-1 traffic remains relatively flat. As the priority-1 traffic is further increased, the delay for priority-1 stations rises. The delays of high-priority stations, however, remain nearly constant at a low level.

B. Experiment 2: Varying Priority-2 Load

Similar to Experiment 1, we vary the load of stations from one priority level. Here, we show the effect of varying the load of high-priority traffic on the other traffic classes. The groups of priority-0 and priority-1 stations each consist of 50 stations, generating each a traffic load equal to 12.5% of the upstream capacity. The third group of stations consists of 100 high-priority stations, generating a load varied from 10% to 45% of the capacity. Fig. 9 shows that, as the load from high-priority stations increases, the priority-0 stations are delayed. When the load is increased further, the request delays for priority-1 stations increases sharply, yet, the priority-2 stations receive low request delays.

C. Experiment 3: Overhead of the Priority Scheme

If no high-priority traffic is present, the presence of PNA slots in each frame is consuming bandwidth that cannot be used by the lowest priority class. In this experiment, we quantify the system overhead in a system that sends all traffic at the same (lowest) priority level. We compare two cases. In the first case, the PNA slots are not present and the priority-0 stations can use the entire range of contention slots. In the second case, three contention slots in each frame are marked as PNA slots for higher priorities. We plot the average request delay versus traffic load in Fig. 10. The figure shows that the reserved PNA slots cause only a slight increase in request delay.

D. Experiment 4: Overhead of Low Load Performance

In Experiment 4, we compare the performance of traffic from priorities 0 and 1 when we vary the number of PNA slots. Note that for newcomer stations from higher priorities, only one PNA slot per frame is available. In contrast, the lowest priority level can use all slots with $\mathrm{RQ} = 0$ for newcomer stations. At low system loads, the lack of slots for newcomer stations may cause request delays for high-priority stations to be larger than the delays of low-priority stations. In Fig. 11 we show the result of an experiment with 50 priority-0 stations and 50 priority-1 stations. We observe that the priority-1 traffic has slightly higher

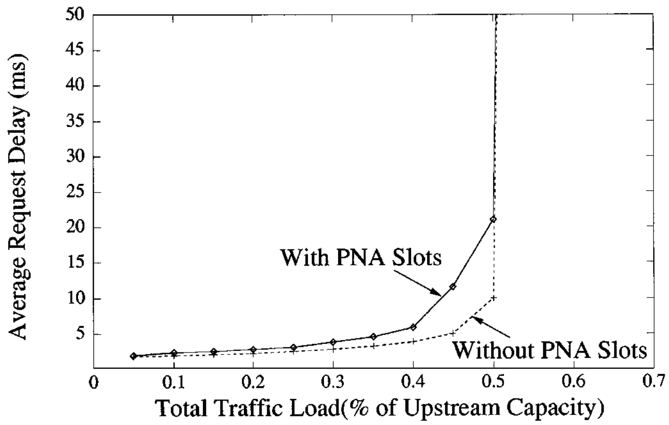


Fig. 10. Experiment 3: PNA overhead-average request delay.

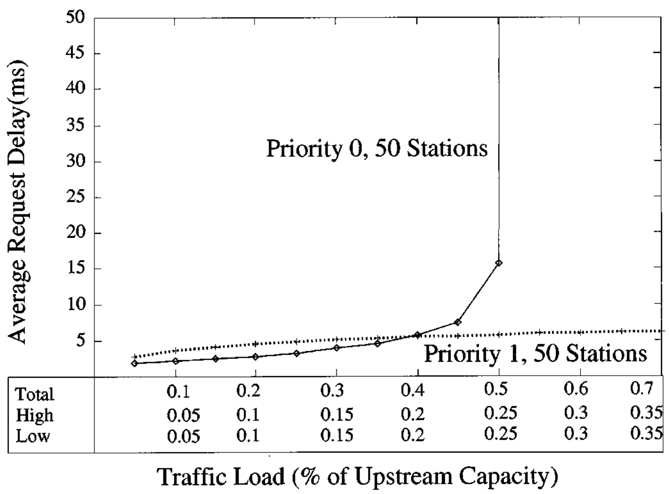


Fig. 11. Experiment 4: Low load performance—1 PNA for priority-1.

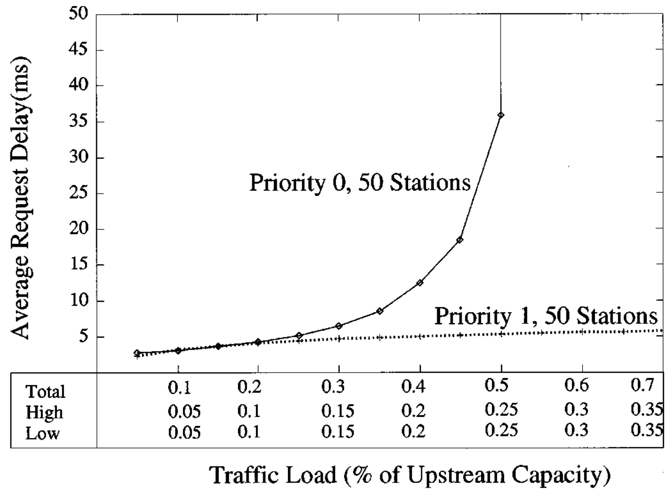


Fig. 12. Experiment 4: Low load performance—5 PNA's for priority-1.

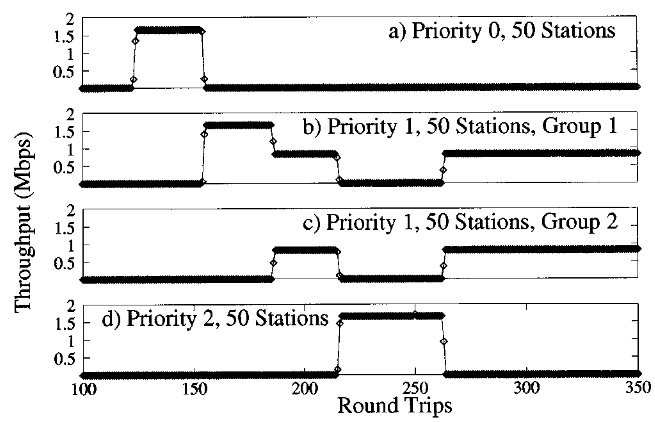


Fig. 13. Experiment 5: Transient throughput.

request delays than the priority-0 traffic (about 1 ms between 5% and 45% load). This can be attributed to the fact that priority-1 newcomer stations are confined to only one PNA slot, while the remaining contention slots are used by the priority-0 traffic. Since at low loads, collisions are infrequent, the request delay is mostly comprised of the time to transmit the first request. At higher loads (above 45%), the request delay is mostly attributed to collision resolution. For comparison, in Fig. 12 we show the same simulations where we allocate five PNA slots to priority-1 traffic in each frame. With this modification, the higher priority traffic never has higher delays than the lower priority traffic. However, with multiple PNA slots per priority, more capacity is wasted if no high-priority traffic is used (since the PNA slots cannot be used by the lowest priority). Ideally, one would dynamically adjust the number of PNA slots; but a comparison of Figs. 11 and 12 shows that the potential benefit of such an adaptive scheme is rather small.

E. Experiment 5: Transient Throughput

In Experiment 5, we show the transient performance of the protocol. In this experiment, we measure the throughput attained by station groups per roundtrip delay (=160 km). We assume that each station group is permanently backlogged, that is, there is always a station in the group which has an outstanding request. The experiment measures the throughput values over a total length of 350 round-trips.

Initially, the upstream channel is idle. After 122 round-trip delays, a group of priority-0 stations begins to transmit, occupying the entire available bandwidth [see Fig. 13(a)]. Note that, even though the upstream bandwidth is 3 Mbps, the usable bandwidth is less than 1.7 Mbps. At time 152, a group of 50 priority-1 stations begins to transmit. As shown in Fig. 13(b), the priority-1 stations completely preempt priority-0 traffic within 1–2 round-trip delays. After 183 roundtrip delays, a second group of priority-1 stations begins to transmit. A comparison of Fig. 13(b) and (c) shows that the two groups of priority-1 stations divide the bandwidth evenly. At time 213, a group of 50 priority-2 stations becomes active. Fig. 13(d) shows that all lower priority traffic is preempted within 1–2 round-trip delays. When the high-priority traffic ceases transmission, after 264 round-trip times, the priority-1 stations again grab the available bandwidth. The experiment illustrates that the priority scheme reacts fast to changes in the traffic load. Low priority traffic is completely preempted within one or two round-trip delays.

F. Experiment 6: Rate–Proportional Bandwidth Sharing

The objective of this paper is the presentation of a MAC protocol for HFC networks with preemptive priorities, that is, lower

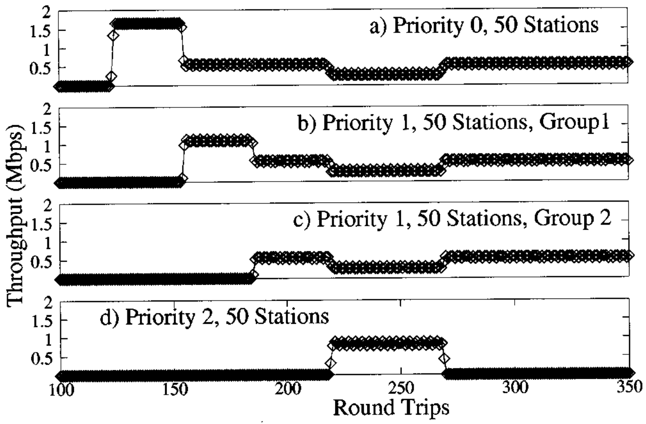


Fig. 14. Experiment 6: Transient throughput with rate-proportional priorities.

priority traffic is completely preempted if the demand for bandwidth from high-priority traffic is sufficiently high.

In this experiment, we demonstrate that our MAC protocol can also implement less stringent priority schemes. We have implemented a rate-proportional bandwidth sharing scheme, where bandwidth is allocated to different priorities in a fixed ratio. Specifically, the ratio of bandwidth allocated to Priorities 2, 1, and 0, is set to 3:2:1. This means that, under conditions of high network load, priority-2 traffic receives three times as much bandwidth as priority-1 traffic, and priority-1 traffic receives twice the bandwidth allocated to priority 0.

We wish to emphasize that the collision resolution protocol in Experiment 6 is the same as in the previous experiments. The only modification for this experiment consists in a different scheduling algorithm for the grant queue at the headend. Specifically, we replaced the static priority scheduler at the headend with a rate-proportional scheduling algorithm.

The results of the experiment are summarized in Fig. 14. As in Experiment 5, time is measured in round trip delays. At time 122, a group of 50 priority-0 station takes the whole channel for transmission. At time 152, a group of priority-1 stations starts transmission. The figure illustrates that bandwidth is shared in a ratio of 2:1, that is, on the average, two priority-1 packets are sent for each priority-0 packet. At time 183, a second group of priority-1 stations starts to transmit. As before, the priority-1 stations share the bandwidth with the priority-0 stations in a 2:1 ratio. At the same time, the two groups of priority-1 stations split the bandwidth available to priority-1 traffic evenly. At time 213, a group of 50 priority-2 stations becomes active. Fig. 14 shows that the different priorities share the bandwidth in a 3:2:1 ratio. At time 264, the priority-2 stations become inactive. As a result, the priority-1 and priority-0 stations again divide the bandwidth in a 2:1 ratio.

V. Concluding Remarks

In this paper we have shown that the implementation of an preemptive priority scheme for transmissions on the upstream channel in an HFC network requires support of priorities throughout all phases of the transmission. We showed that the lack of priority support during the collision resolution process has a negative impact on the effectiveness of a priority scheme. We proposed a priority scheme which provided handling of priorities during both the request phase and the actual data transmission phase. Specifically, we presented a priority scheme which supports priorities during contention resolution for the ternary-tree contention-resolution algorithm employed by the IEEE 802.14 MAC protocol. The 802.14 WG has accommodated a framework for the handling of priorities of contention-prone request transmissions, which has a similar semantics as the scheme presented in this paper. In fact, our scheme can be exactly implemented in the framework of the standard (see the Appendix).

While the priority scheme presented in this paper was developed within the context of the IEEE 802.14 standardization effort, the proposed algorithm is applicable to a general setting of multiaccess protocols with a contention/reservation approach [3].

Appendix
IEEE 802.14 Standard Priority System

The priority scheme, as presented in this paper, was first developed in July 1997. In November 1997, a proposal was made to the IEEE 802.14 WG [1] for enabling priorities during collision resolution. In April 1998, this proposal has since been incorporated into the 802.14 draft standard. It is important to note that the standardized version only provides a framework for supporting priorities, so that vendors can implement a priority scheme, and does not explicitly state guidelines for a specific scheme.

The standardized priority mechanism uses semantics for resolving collisions of priority transmission requests similar to the semantics of our scheme. In particular, our priority scheme from Section III, which was shown to be able to preempt all low-priority traffic, can be implemented within the framework of the standard.

Next we discuss some of the key differences between the standardized priority framework and the scheme presented earlier in this paper.

One difference between the scheme in Section III and the standardized version is the method used to allocate newcomer slots to stations. Recall, that our scheme used special request slots, so-called PNA slots, for high-priority newcomer stations, and used slots with $\mathrm{RQ} = 0$ for access by newcomer stations from the lowest priority. The standardized version does not have dedicated PNA slots. Rather, the headend explicitly designates for each frame which newcomer slots can be used by a given priority. Another difference from the scheme presented in Section III, is that newcomer slots, if so designated by the headend, can be used by more than more than one priority. (This, however, may have an adverse effect on the effectiveness of the priority scheme.)

As in Section III, the standard uses the downstream feedback messages from the headend, so-called request minislot allocation elements (RMAE), to designate RQ values and priorities to the slots in the next upstream CS cluster. However, different from the scheme in Section III, the feedback also contains the priority assignment for newcomer slots in the next CS cluster. The format of an RMAE message is shown in Fig. 15; only

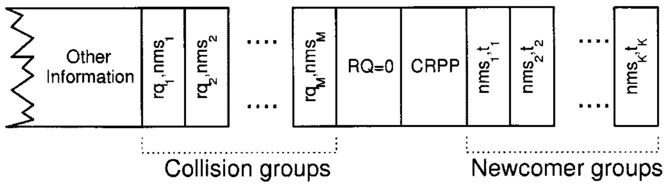


Fig. 15. Request minislot allocation element. Collision groups: One group for each value RQ>0 for which slots are allocated in the next CS cluster. Newcomer groups: One group for each bit set to "1" in bitfield CRPP.

fields that are relevant to the priority scheme are shown. The message informs the stations of the number of CS of each RQ value and provides information about priority newcomer slots. Each RMAE message contains a number of so-called *collision groups*. There is one collision group for each value $\mathrm{RQ} > 0$ for which slots are allocated in the next frame. Recall that $\mathrm{RQ} > 0$ implies that there was a collision in an earlier frame. The i-th collision group is given by a tuple $(\mathrm{rq}_i, \mathrm{nms}_i)$, with the interpretation that the next frame will have nms_i (Number of Minislots) fields designated as $\mathrm{RQ} = \mathrm{rq}_i$.

In [1], newcomer stations from all priorities must use slots with $\mathrm{RQ} = 0$ for sending a request. (Again, in Section III, we used PNA slots for high-priority newcomer stations.) The assignment of slots with $\mathrm{RQ} = 0$ to priorities is done as follows. Each RMAE message has a CRPP field that dictates which priorities can transmit their request. A newcomer station with a priority-k message will be allowed to transmit a request in the next frame only if CRPP has a "1" in bit position k or higher.[8]

For example, if $k = 5$, then any of the following 8-bit CRPP fields will allow transmission of a priority-5 request:

12 345 678	12 345 678	12 345 678
xxxx1xxx	xxx10xxx	xx100xxx

12 345 678	12 345 678
x1000xxx	10 000xxx

where x's are either "0" or "1."

For each bit position in the CRPP field with a "1," the RMAE message has a so-called *newcomer group*, where each group is represented by a pair (nms, t). For the i-th "1" in the CRPP field, say in the qth bit, the pair (nms_i, t_i) indicates that the next CS cluster has nms_i slots for use by priorities q or higher. Parameter t_i is a time boundary which indicates that only newcomers generated at time t_i or earlier are admitted in the nms_i slots.[9]

Example: Consider a station with priority $k = 5$ that sees an RMAE message with the following CRPP and newcomer groups:

CRPP field

...	*RQ*	10101000	nms_1	nms_2	nms_3
	0		t_1	t_2	t_3

Note that 3 bits are set in the CRPP field which correspond to priorities 1, 3, and 5 (1 being the lowest priority). Therefore, there are 3 newcomer groups following the CRPP field. Our priority-5 station can only use the nms_3 minislots in the third group (with time boundary parameter t_3). The slots in the first group are reserved for priorities 1 and 2, and the slots in the second group are reserved for priorities 3 and 4. Priorities 6, 7, and 8 will use the same newcomer group information as priority 5, i.e., $(\mathrm{nms}_3 : t_3)$.

Each priority level can only use one newcomer group. For priority-k, this is the group that corresponds to the *rightmost* "1" at or to the left of bit-position k. For example, if the CRPP is set to 00 001 001, all newcomer stations with priorities 5 or higher are admitted. The slots that correspond to bit-position $k = 5$ can be used by priorities 5, 6, 7, and the slots that correspond to bit-position $k = 8$ are for the exclusive use by priority 8. Thus, in the next frame, a priority-5 newcomer station can use the slots for position $k = 5$, but not the slots for $k = 8$.

With this mechanism, a large set of priority schemes can be implemented. To achieve a preemptive priority scheme, i.e., a scheme which completely preempts low-priority traffic from the network if the traffic load from high-priority traffic is sufficiently high, the headend must ensure that collisions are resolved in a priority order. This can be done if the headend always assigns higher RQ values to higher priority collisions, and if the headend resolves collisions in the order of RQ values.

It is not hard to see that our priority scheme from Section III can be implemented in this framework if, in addition to collision resolution in priority order, the following options are selected:

- The CRPP field is set to 1 111 111. This ensures that newcomer slots never contain collisions from different priorities. (If the next frame does not have a sufficient number of CS, the CRPP is modified appropriately.)
- For each but the lowest priority, the NMS value of the newcomer group is set to $\mathrm{nms}_k = 1$, that is, there is exactly one newcomer slots for each priority (other than the lowest priority). This slot replaces the PNA slots shown in Fig. 6. For the lowest priority level, we set nms_k to the remaining number of contention slots in the next frame.
- For all but the lowest priority k, the time boundary parameter is set to $t_k = \infty$, that is, a newcomer station from priority k selects the available slot with probability 1. For the lowest priority, we set nms_k to the remaining number of contention slots and $t_k = \infty$.

[8]A higher priority index indicates a higher priority level.

[9]To guarantee orderly and fair admission of newcomer stations, the time boundary value t_i has substituted the backoff parameter R (see Section II) in the latest draft standard [19].

REFERENCES

[1] C. Bisdikian, "Enhancing the collision resolution algorithm comments to d2R2," IEEE 802.14 Working Group, Contrib. IEEE 802.14/97-121, Nov. 1997.

[2] ——, "A review of random access algorithms," presented at the Int. Workshop on Mobile Communications, Thessaloniki, Greece, Sept. 1996.

[3] C. Bisdikian, B. McNeil, R. Zeisz, and R. Norman, "MLAP: A MAC level access protocol for the HFC 802.14 network," *IEEE Commun. Mag.*, vol. 34, pp. 114–121, Mar. 1996.

[4] J. Capetanakis, "Tree algorithm for packet broadcasting channel," *IEEE Trans. Inform. Theory*, vol. IT-25, pp. 505–515, Sept. 1979.

[5] M. Corner. (1998) Quality of service issues in hybrid fiber-coax networks, M.S. thesis. Univ. Virginia, Charlottesville. [Online] Available: http://www.eecs.umich.edu/mcorner/papers.html

[6] J. Eng and J. Mollenauer, "IEEE project 802.14: Standards for digital convergence," *IEEE Commun. Mag.*, vol. 33, pp. 20–23, Apr. 1995.

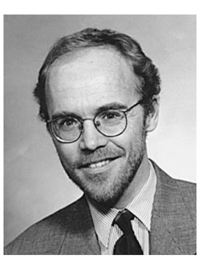
[7] B. Burroughs *et al.*, "Collision sensitivity due to laser clipping," IEEE 802.14 Working Group, Contrib. IEEE 802.14/96-160, July 1996.

[8] C. Bisdikian *et al.*, "Cable access beyond the hype: On residential broadband data services over HFC networks," *IEEE Commun. Mag.*, vol. 34, pp. 128–135, Nov. 1996.
[9] N. Golmie *et al.*, "New packet admittance policies for IEEE 802.14 MAC protocol," presented at the Proc. 4th Int. Conf. Optical Communications and Networks, Paris, France, Jan. 1997.
[10] M. Gagnaire, "An overview of broad-band access technologies," *Proc. IEEE*, vol. 85, pp. 1958–1972, Dec. 1997.
[11] N. Golmie, M. Corner, Y. Saintillan, A. Koenig, and D. Su, "The NIST ATM network simulator: Operation and programming," Nat. Inst. Stand. Technol., Intern. Rep. 5703R1, Mar. 1998.
[12] N. Golmie, Y. Saintillan, and D. Su, "A review of contention resolution algorithms for IEEE 802.14 networks," *IEEE Commun. Surveys*, First Quarter 1999.
[13] N. Golmie, M. Corner, J. Liebeherr, and D. Su, "Improving the effectiveness of ATM traffic control over hybrid fiber-coax networks," presented at the IEEE Globecom'97, Phoenix, AZ, Nov. 1997.
[14] H. Lin and G. Campbell, "PDQRAP—Prioritized distributed queueing random access protocol," Comput. Sci. Dept., Illinois Inst. Technol., Chicago, Tech. Rep. 93-2, Mar. 1994.
[15] P. Mathys and P. Flajolet, "Q-ary collision-resolution algortithms in random-access systems with free or blocked channel access," *IEEE Trans. Inform. Theory*, vol. IT–31, pp. 217–243, Mar. 1985.
[16] B. McNeil, D. Lin, and C. Bisdikian, "Implementation overview of tree-based algorithm," IEEE 802.14 Working Group, Contrib. IEEE 802.14/96-246, Nov. 1996.
[17] A. Paff, "Hybrid fiber/coax in the public telecommunications infastructure," *IEEE Commun. Mag.*, vol. 33, no. 4, pp. 40–45, Apr. 1995.
[18] S. Ramanathan and R. Gusella, "Toward management systems for emerging hybrid fiber-coax access networks," *IEEE Network*, vol. 9, pp. 58–68, Sept./Oct. 1995.
[19] IEEE Project 802.14/a draft 3 revision 3: Cable-TV access method and physical layer specification, I. Reede, M. Brandt, and J. Karaoguz, Eds., Unapproved IEEE 802.14 Project Draft Specification, Oct. 1998.
[20] R. Rom and M. Sidi, *Multiple Access Protocols: Performance and Analysis*. Berlin, Germany: Springer-Verlag, 1990.
[21] K. Sriram and P. Magill, "Enhanced throughput efficiency by use of dynamically variable request minislots in MAC protocols in HFC and wireless access networks," *Telecommun. Syst.*, vol. 9, no. 3/4, pp. 315–333, 1998.
[22] M. Vecchi, "Broadband networks and services: Architecture and control," *IEEE Commun. Mag.*, vol. 33, pp. 24–32, Aug. 1995.

Mark D. Corner (S'95) received the B.S. and M.S. degrees in electrical engineering at the University of Virginia, Charlottesville. He is currently pursuing the Ph.D. degree in electrical engineering at the University of Michigan, Ann Arbor.

His interests include MAC layers for broadband networking, wireless networking and multimedia transmission.

Mr. Corner is a member of Eta Kappa Nu and Tau Beta Pi.



Jörg Liebeherr (S'88–M'92) received the Dipl.-Inf. degree from the University of Erlangen, Germany, in 1988 and the Ph.D. degree in computer science from the Georgia Institute of Technology, Atlanta, in 1991.

In 1992, he was a Postdoctoral Fellow at the University of California, Berkeley. From 1992 to 1997 and since 1998, he has been with the Department of Computer Science, University of Virginia, Charlottesville, where he is currently an Associate Professor. From 1997 to 1998, he was an Associate Professor in the Department of Electrical Engineering, Polytechnic University, Brooklyn, NY. His research interests are QoS networks, multicast communications, and multimedia systems. He currently serves on the editorial boards of *Computer Communications, Cluster Computing*, and *Real-Time Systems Journal*.

Dr. Liebeherr is a member of ACM (Sigcomm, Sigmultimedia, Sigmetrics). He currently serves as Editor-in-Chief of *IEEE Network* and also serves on the editorial board of IEEE/ACM TRANSACTIONS ON NETWORKING.

Nada Golmie (M'96) received the B.S. degree in computer engineering from the University of Toledo, Toledo, OH, in 1992, and the M.S. degree in electrical and computer engineering from Syracuse University, Syracuse, NY, in 1993.

Since 1993, she has been a Research Engineer in the High-Speed Networks Technologies group, National Institute of Standards and Technology (NIST), Gaithersburg, MD. Her research interests include modeling and performance evaluation of network protocols, media access control, and quality-of-service for ATM, IP, HFC, WDM, and wireless network technologies.

Ms. Golmie is a member of ACM.

Chatschik Bisdikian (SM'94) received the Diploma in electrical engineering from the Polytechnic School, Aristotle University, Thessaloniki, Greece, in 1983, and the M.Sc. and Ph.D. degrees, both in electrical engineering, from the University of Connecticut, Storrs, in 1985 and 1988, respectively.

Since 1989, he has been a Research Staff Member with the IBM T. J. Watson Research Center, Yorktown Heights, NY. His research interests include short-range wireless communication networks, service discovery, spontaneous networking, multimedia and broadband communications, HFC networks and cable modems, modeling and performance evaluation of high speed networks, and computer system modeling and analysis. He has published over 75 technical papers on these and other related areas, and holds two patents. He has been actively involved with the development of the Bluetooth specification, a short-range RF-based personal area network technology, and is currently working for the Bluetooth standardization in the IEEE 802.15 working group. He has recently contributed heavily in the development of the IEEE 802.14 draft standard for HFC CaTV systems, with special emphasis on its dynamically adjustable, priority-enabled contention resolution algorithm.

Dr. Bisdikian was a 1995 finalist for the Eta Kappa Nu Outstanding Young Electrical Engineer. He has served on the editorial boards of *IEEE Network* (since January 1999), IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS Special Wireless Series (since August 1999), and *Telecommunication Systems* (September 1993 to August 1996). He served on the program committees and chaired sessions in many conferences (INFOCOM, GLOBECOM, ICC, etc.). He is currently serving as a vice-chair of the IEEE 802.15.1 task group. He has been listed in *Who's Who in Science and Technology*, is a voting member of the IEEE802.15 standards committee, and is a member of Eta Kappa Nu and Phi Kappa Phi.

David H. Su (M'96) received the Ph.D. degree in computer science from The Ohio State University, Columbus.

He is the manager of the High Speed Network Technologies group in the Information Technology Laboratory, National Institute of Standards and Technology (NIST), Gaithersburg, MD. His main research interests are in modeling, testing, and performance measurement of network protocols. He has been involved in modeling and evaluation of protocol being developed by standardization organizations including protocols for ATM, HFC, and wireless personal area networks. He has also participated in the development of standard conformance test suites for testing of X.25, ISDN, FDDI, and ATM network protocols.

Dr. Su is a member of ACM.