On Performance of HFC MAC Layer Algorithms.

A thesis submitted in fulfillment
of the requirements for the degree of
Master of Science

by
Igor Polevoy

Supervised by
Prof. Danny Dolev

Institute of Computer Science
The Hebrew University of Jerusalem
Jerusalem, Israel.

September 25, 2000

Acknowledgments.

First, I would like to thank my advisor, Prof. Danny Dolev for his support, guidance and patience. His experience and comprehension made the whole thing possible.
I thank all the people who contributed their time, although their were not ought to do so, in order to help me with this work: Ilya Shnaiderman, Tal Anker, David Breitgand, Lev Khaikovich, my dad and others. Thanks to Osnat Mokryn for introducing me into the theme.
Special thanks to Daniel Lederer for the idea on opening sentence.

Abstract

Although cable television networks were designed to provide data transmission in one direction only – from the central location to the subscribers, a recent upgrade of cable TV infrastructures, namely the creation of combined hybrid fiber-coax (HFC) networks on one hand, and growing demand for interactive services on the other, lead to the introduction of bi-directional transmission over those networks. A number of algorithms that facilitate bi-directional communications over the HFC networks were suggested in the recent years.

The work studies influence of different parameters on performance of the algorithms for the medium access layer of such networks. It investigates how the performance changes if variants of the same algorithm are used, and what is the influence of network architecture parameters, such as number of subscribers or the propagation delay, on performance. It also studies modifications that could be done to the integral parts of the algorithms – bandwidth allocation scheme, frame size adjustment and shows what performance improvement could be gained with their help.

1 Introduction.

Since the dawn of humanity, man has searched for the answer to the fundamental question: "What is the meaning of life?"

The truth is that there is probably nothing more distant from this concept than the greatest invention of the 20-th century - the television. Television's ubiquity is unbelievable: as mentioned by W. Ciciora et al.[28], more people in the United States have televisions than telephones, and cable services are available to 96.7% of television households.

Cable television started off more than 40 years ago as Community Antenna Television (CATV) and was intended to provide TV services to areas in which over-air broadcasting could not be received. Since then it has greatly evolved, and nowadays has comprehensive infrastructures and provides hundreds of channels, including satellite signals retransmission, pay-per-view services, etc. As CATV systems were initially intended to provide broadcast signals in one way - from a central location to subscribers - they were designed and implemented to optimally realize this goal. However, this design was not suitable for other purposes, such as the use of cable television infrastructure for a general-purpose bi-directional communications system.

A recent upgrade of CATV infrastructures, which entailed replacing the part of the coax cable trunk that reaches from the central location towards the stations, with a fiber carrying optical signals, lead to the creation of combined hybrid fiber-coax (HFC) networks. A number of fiber characteristics, such as low cost, increased bandwidth, reliability, higher signal quality and immunity to ingress of interfering signals, made it possible to facilitate two-way communications. The cable system was broken up into a number of smaller systems, each connected through its own fiber cable to the central location.

On the other hand, recent years have witnessed a rapidly increasing demand for high-speed communications, combined with constant growth of the Internet market and the development of interactive video/audio services. This presented an opportunity for cable operators to enter the competition over providing these services to the public, and from the outset, the cable companies held a good position in this competition. They have an infrastructure available that is able to provide enormous bandwidth, and very mature technologies that have been in use for years.

The problem of converting a regular cable television system into a bi-directional communications network raises, besides serious challenges in the field of electrical engineering, the well-known problem of access to a shared medium. There is no separate line from each station to the central location, and in order to enable a station to communicate in the reverse direction (also called upstream), the access of each station to the cable should be efficiently managed. This problem is typically solved at the MAC layer level of the OSI model, using special transmission protocols. The MAC level protocols used in the networking need to be adjusted to the special constraints of HFC networks, inherent to their structure.

The basic HFC structure is discussed in the following introductory section, as are the MAC protocols developed for cable television upstream transmission. Also described are major standards developed to provide interoperability of equipment that enables two-way communications - cable modems, placed at the customer premises, and the head-end (HE), placed at the central location.

The main characteristics of the MAC protocols for HFC networks have been thoroughly studied by now. However, a number of aspects in the structure of these networks and in the protocols themselves that can be expected to influence the systems' performance, have not yet been fully researched, as these studies do not show the extent and form of this influence. The main body of the work investigates these aspects in greater detail.

Two main contention resolution algorithms – p-persistence and ternary tree - were simulated during our study. The influence of changes in the HFC network structure on the performance of the algorithms was examined. Our results demonstrate that decreasing the number of stations linearly improves performance, while there is little improvement gained from decreasing the distance. The ternary tree algorithm with the variable bandwidth allocation proved to be the scheme that improves most when the distance decreases. We also researched bandwidth allocation policies themselves, and will later show that the widely accepted traffic-based bandwidth allocation has some drawbacks. We suggest a simpler collision-based strategy that can solve the problems. We discovered that changing the frame size for the clustered mode ternary tree algorithm does influence the systems' performance. An optimal frame size was found, and we suggest algorithms that, when run at the head-end, adjust the frame size dynamically and keep it close to optimal at every load. The connection between the request access delay jitter, the algorithms' ranging parameter and the distance to the head-end was observed and thoroughly studied. We demonstrate how the jitter improves when the head-end is brought closer to the stations, and conclude that this holds more importance for the p-persistence class of algorithms. Finally, we show that there is a difference between the p-persistence algorithm's standard way of operation in clustered and continuous modes, having compared the two modifications and drawn conclusions regarding their relative performances.

The thesis is designed as follows: Chapter 2 contains an overview of the HFC network architecture, the MAC layer protocols and the standards that were developed for this architecture. Several market statistics conclude the chapter. Chapter 3 presents the simulation framework and the common simulation parameters. Chapter 4 contains results of the comparison between the performances of continuous and clustered mode p-persistence algorithms. In Chapter 5, we study the influence of the distance to the head-end vs. the number of stations on the algorithm's performance. Chapter 6 describes the traffic-based bandwidth allocation scheme and presents an alternative collision-based algorithm. Chapter 7 deals with request access delay jitter and with the way it changes with distance. The influence of frame size on the request access delay and ways of finding the optimal frame size for a given load are described in Chapter 8. Chapter 9 contains some concluding remarks and suggestions for future work.

With the foreseen future development of cable modem services, service providers will eventually have to make decisions regarding possible changes in the protocols' operational parameters and in the network configurations. Ultimately, it may be decided to change the whole HFC hierarchy. Understanding the results presented here should help make fully informed and aware decisions over these issues.

2 Overview and Previous Work.

2.1 The HFC Architecture and Integrated Services.

In the last few years, the Internet market has undergone a tremendous growth, and other digital network services, such as digital audio and video transmission, have matured. Hence, new challenges are posed to the performance and availability of the existing public network services. Recent technological advances in electrical engineering and in computer science have given cable TV companies the option to deviate from their traditional business of broadcasting entertainment and compete over effective provision

of these new services to the community. The deregulation in the communications market, which enables full-scale competition among telephone, data, cable TV, wireless services, etc., is already in force in the US and in other countries, and seems to finally to have arrived in Israel. In this overview chapter we wish to show the position of cable TV companies in this competition, and present the fundamental characteristics of their solution to the challenge of providing integrated services.
Since the first introduction of Cable TV in the late 40's, its networks were developed to enable effective one-way broadcasting. For this reason, the topology of the Cable TV network is characterized by a tree-and-branch structure, the one best suited to this need.

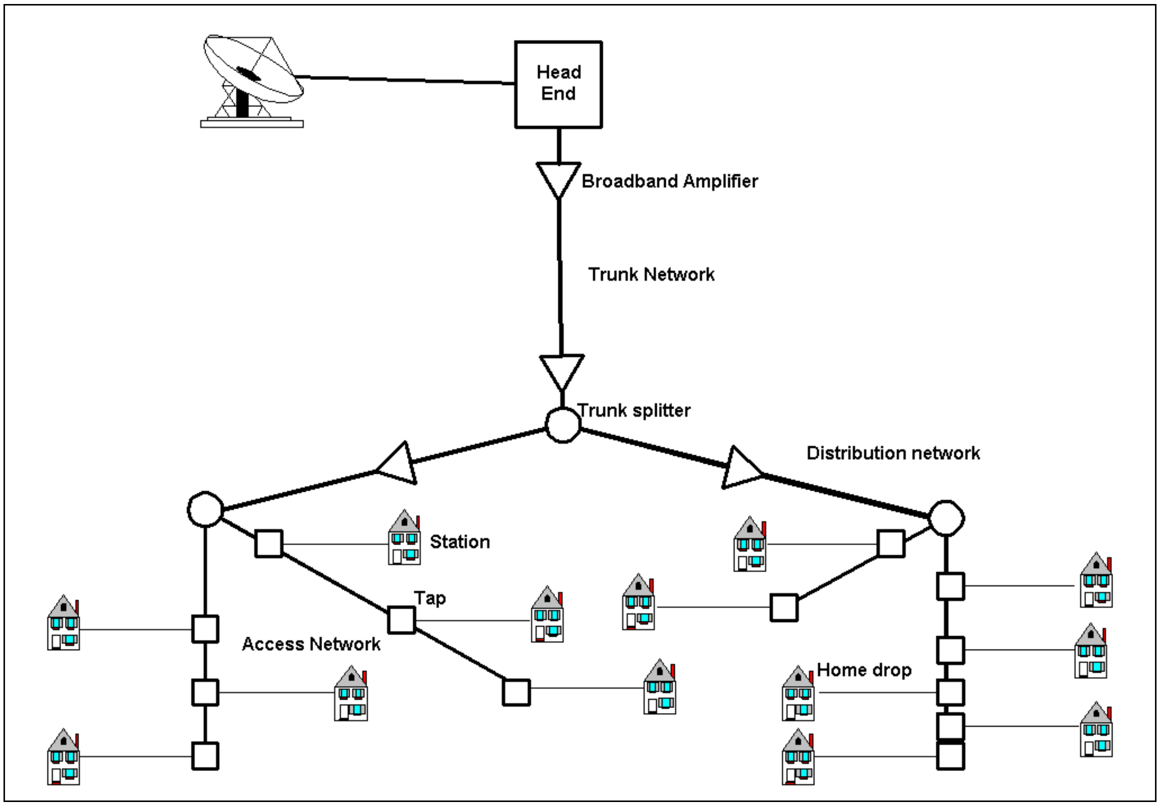

Figure 2.1

At the root of the tree, the head-end controller is located. This broadcasts data received from satellites, or from other outer networks/sources, onto the network. The first part of the cable network, known as the trunk, is shared by all subscribers. The network then divides into separate neighborhood (or distribution) networks, each one serving a cluster of residences. Here, the last-mile (also called access) network consists of drops (taps) – a lower quality cable that connects user devices with the distribution cable. The medium traditionally used for building CATV networks was the coaxial cable, or simply - coax. The coax allows efficient signal transmission over a wide range of frequencies, and due to its insulation layer, is less susceptible than the twisted copper pairs used in telephony to ingress noises that can interfere with the transmitted data. Still, the signals sent over the coax, as over any other wire, are attenuated and can become, over distance, indistinguishable from the noise. The attenuation is proportional to the square root of the signal frequency, if expressed in decibels per unit length. This introduces another important piece of hardware used in CATV networks – broadband amplifiers. The amplifiers can be placed at approximately every 700 m along the trunk, as well as at the points connecting the trunk and distribution networks, and in the distribution network itself. They amplify signals arriving on the incoming cable from the head-end (HE) to the outgoing cable, within a certain range of frequencies. This flow constitutes the so-called "downstream" – typically, a broadcast transmission sent from the HE to the network subscribers. It is clear that in order to be able to provide two-way communications, the ability for upstream transmission must also be built into CATV networks.

In order to achieve this, some part of the frequency spectrum is assigned to the upstream, and the amplifiers in the network are "upgraded", in the sense that they now need to amplify this portion of frequencies in the reverse direction, for the upstream. However, amplifiers cannot distinguish between actual signals and noise. Therefore, if one signal is transmitted in the downstream direction, it is amplified along with the noise in the trunk, and is sent this way to the distribution networks. In the upstream direction, if one subscriber sends a signal, it combines with noise coming from other branches of the tree and is then amplified. In this way, noise accumulates in the upstream. The problem is known as "noise funneling". Very robust signal modulation schemes must be used in order to cope with this problem

In a new kind of architecture for CATV networks, fiber-optic cables are used for the trunk part of the network, and sometimes for the distribution part. Fibers have a number of important qualities, superior to those of coax cables. They attenuate signals less, thereby eliminating the need for amplifiers in the trunk at current distances (up to 80 km), which reduces cost and increases overall robustness in the network. The fiber plant itself is not an expensive medium, therefore it is possible to assign a separate fiber to each node serving a neighborhood, thus significantly reducing the funneling problem. Fibers provide more bandwidth than coax cables (up to approximately one gigahertz). And finally, the system is able to "stack" various signals that are modulated in different ways – analog or digital - and transmit them as they are, without any re-coding, by translating them to higher frequencies. The only expensive component of fiber-optic cables is the opti-electric connectors. Providing a computer (or any other electronic device) with a card that can receive an incoming optical signal and convert it into an electrical one, could cost today thousands of dollars. For this reason, the last-mile network is typically left with coax cables. Such a network is called Hybrid Fiber-Coax (HFC), because both media are used in the same network.

Figure 2.2 is a schematic representation of an HFC network. Optical modulators and equipment for Internet and telephone access have now been added to the head-end. Bi-directional amplifiers with bandwidth filters have replaced the unidirectional ones in the coax part of the network. The conversion of optical signals into electric signals, and vice versa, in the trunk and distribution parts of the network is separated in the fiber nodes. A device for handling upstream transmission has been installed at the customer's premises. Such a device is called Cable Modem (CM). The CM consists mainly of a downstream demodulator, an upstream modulator and a CPU with associated memory, to provide for participation in the MAC layer protocol (and possibly higher layer protocols). Three main formations of CM are possible: 1) Internal CM – when the CM is simply a PCI-bus card inside the customer's computer 2) External CM - a box connected to one or more computers through an Ethernet (10-baseT) 3) Interactive Set-Top Box (STB) – a device connected to the customer's TV set that contains a CM inside.

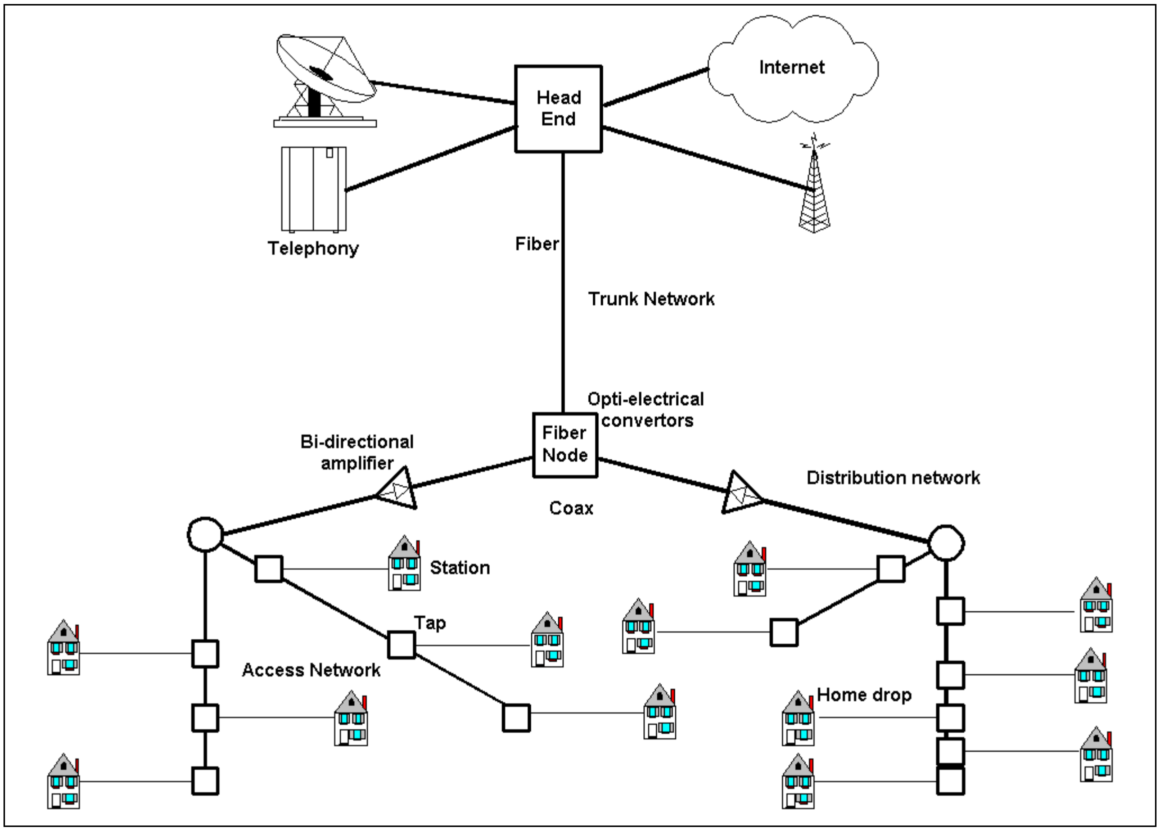

Figure 2.2

A variety of services that pose different demands to bit-rate, delay and QoS can be provided over HFC networks. The following table taken from [1] summarizes some of the popular services.

Broadband Digital Services	Downstream Bandwidth	Upstream Bandwidth
Broadcast Video: *Broadcast TV,* *Enhanced pay per view (per channel).*	*1.5 to 6 Mb/s.*	*None or POTS.*
Interactive Video: *Video on demand, Interactive TV, Interactive games, Information services.*	*64 kb/s* *to* *6 Mb/s.*	*9.6* *to* *64 kb/s.*
Internet: *WWW, FTP, Telnet, etc.,* *Electronic mail, Voice, Broadcast.*	*14.4 kb/s* *to* *10 Mb/s.*	*9.6* *to* *128 kb/s.*
Symmetric Data: *Desktop multimedia, Work at home,* *Video conferencing, Video telephony,* *FAX.*	*64 kb/s* *to* *1.5 Mb/s.*	*64 kb/s* *to* *1.5 Mb/s.*
Small Business and Home: *Internet home page, Internet information server.*	*9.6* *to* *384 kb/s.*	*64 kb/s* *to* *1.5 Mb/s.*

In both the downstream and upstream directions, not all the frequency spectrum is used for a single transmission. Rather, the bandwidth is divided into separate channels, with a guard-band between them in order to keep the signals from interfering with each other. The size of each channel is typically 6MHz (USA) or 8 MHz (Europe). In each channel, bandwidth is allocated and reallocated according to current demands by services. For historic and economical reasons, the most popular are the so-called "sub-split" networks, in which the lower part of the spectrum (5-45 MHz) is assigned to the upstream and the rest to the downstream. The following figure - Figure 2.3 [2] - demonstrates a possible bandwidth allocation for an HFC network and for the services carried over each range.

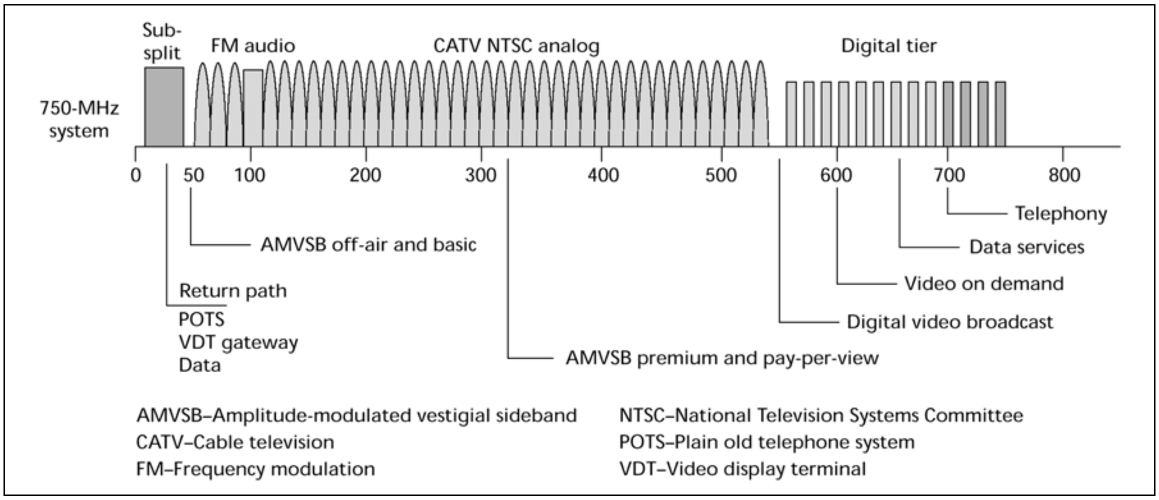

Figure 2.3

From the above, the advantages of HFC networks can be summarized as follows:

- They have a widely deployed networking infrastructure, capable of providing enormous bandwidth.
- No additional equipment is needed to convert the signals into the format accepted by terminal devices. Signals are transmitted at bit-rates and frequencies that are appropriate to the service being delivered.
- Bandwidth is dynamically allocated. The medium is shared and therefore is packet-switched (customers can be added on without having to individually assign extra equipment to each connection).
- There is no need for any kind of "connection" procedure (like dial-in) – the CM is connected the moment it is turned on.

The problems inherent to the HFC infrastructure and topology are:

- Asymmetric upstream/downstream bandwidth.
- Lack of carrier sensing. Due to the tree-and-branch topology and the way the signals are amplified, one station cannot listen to the transmission of another.
- Long propagation delays, comparable to the time of a single upstream burst transmission, that are induced by the relatively large distance between the head-end and the stations.
- Security. Since the medium is shared among many users, authentication and privacy must be provided.

2.2 MAC Level Protocol Basics.

On the MAC level, the HFC network should provide an efficient on-demand upstream bandwidth allocation to active subscribers. In order to achieve this goal, the basic protocol between the HE and the stations is as follows:

The algorithm employs a kind of TDMA. All the time that is available for upstream transmission is divided into fixed-size mini-slot (MS) units. In the downstream, the HE periodically sends a description of the MS allocation for the subsequent time interval, which is called “frame” (or “cluster”). During the frame, there are MS allocated to carry the stations’ requests on a contention basis (CS), and MS reserved to carry actual data, which are assigned by reservation, as a result of previous successful requests. In order to ensure fairness, the duration of the frames must be greater than, or equal to, the delay to the farthest station. The delay consists of the propagation delay, the processing delay and the time needed to send the packet on both sides. Between the frames, some bandwidth is reserved for initialization and maintenance purposes. The stations’ end basic MAC algorithm is:

1. Initialization. When a station becomes active it sends a ranging request to the HE, in a pre-defined MS reserved for this purpose. After receiving the response from the HE, the station computes its propagation delay to the HE (RTD – round trip delay) and adjusts the clock accordingly. A message sent in the upstream should arrive to the HE exactly in the predefined MS, so the station must take the propagation delay into account. The station also acquires a global time reference, periodically sent in the downstream, in order to know the exact MS boundaries.
2. New data arrival. When a new packet arrives at a station, it sends a request to transmit in the subsequent contention slot, which is open for new requests. Recall that the frame description sent by the HE in the downstream gives a full description of the subsequent frame time interval, including information on the position of the contention slots and on the identity of stations that are eligible to transmit in the specific slot. The request usually consists of an amount of data to be sent and the station’s ID. In this way, collisions can only happen to the smaller request messages, and the data itself is transmitted collision-free.
3. Waiting for request feedback. If the request was successfully transmitted, the station will receive a data grant from the HE. The grant will reserve some data slots for the station, or indicate to the station that the request was processed but that the data slots will be allocated in upcoming frames. If additional data arrives at the station during the granted data transmission, it can choose to use “piggybacking”, i.e. to put new requests inside the data slot, in order to avoid contention. If more than one station sends a request in a specific contention slot, a collision will occur.
4. Collision resolution. If the station receives an indication that its request collided, it must retransmit, according to the Collision Resolution Algorithm. Note that the HE can only know that a collision has occurred - it has no additional information regarding the collision multiplicity or the identity of the collided stations. Also, it is common practice not to allow a station that has pending requests to send a new one. However, if new data arrives in the meantime, the station can update the amount of data requested.

In addition, the HE can allocate some separate bandwidth for the contention and isochronous modes, in the upstream. In the contention mode, stations are permitted to send data packets directly, without previously having to request permission. The isochronous mode provides constant bandwidth for the application, during the time it is needed. In this way, QoS demands can be met.

2.3 Previous Work.

The two main parts of the HFC MAC protocol that most influence the system's performance are the bandwidth scheduler, at the head-end, and the Collision Resolution Algorithm (CRA). The CRA can be p-persistence or a splitting tree algorithm.

2.3.1 General CRA's.

2.3.1.1 P-persistence.

This algorithm is based on the well-known ALOHA protocol. Each station transmits its request in the available CS, with probability p. Unlike in traditional ALOHA, this rule applies to new requests as well as to retransmissions. The maximum achievable throughput of ALOHA is 36.7% ($1/e$). The probability of successful transmission for p-persistence is given by $P_{succ} = n * p * (1-p)^{(n-1)}$, in which n is the number of contenders at the beginning of the slot. The system is stabilized when $p = 1/n$, under the Poisson traffic assumption. In order to estimate *n*, which is generally unknown, the pseudo-Bayesian algorithm, suggested by Rivest [10], is used at the HE.

2.3.1.2 N-ary Splitting Tree.

In this group of algorithms, all stations involved in a collision are split into *n* sub-groups. Then, each station that was involved in the collision randomly selects one of these subgroups. The first of the *n* subgroups retransmits in the subsequent available contention slot. All other stations enter waiting mode, until the resolution of the previous subgroup. The collision resolution process can be represented as a tree, in which each collision produces *n* new nodes. It has been shown [11] that the best throughput is achieved with $n = 3$ (ternary tree). The following figure shows an example of the collision resolution process with ternary tree, in which the initial collision multiplicity equals 5. The numbers inside the tree nodes represent the stations that transmitted, and the numbers next to each node show the resolution order.

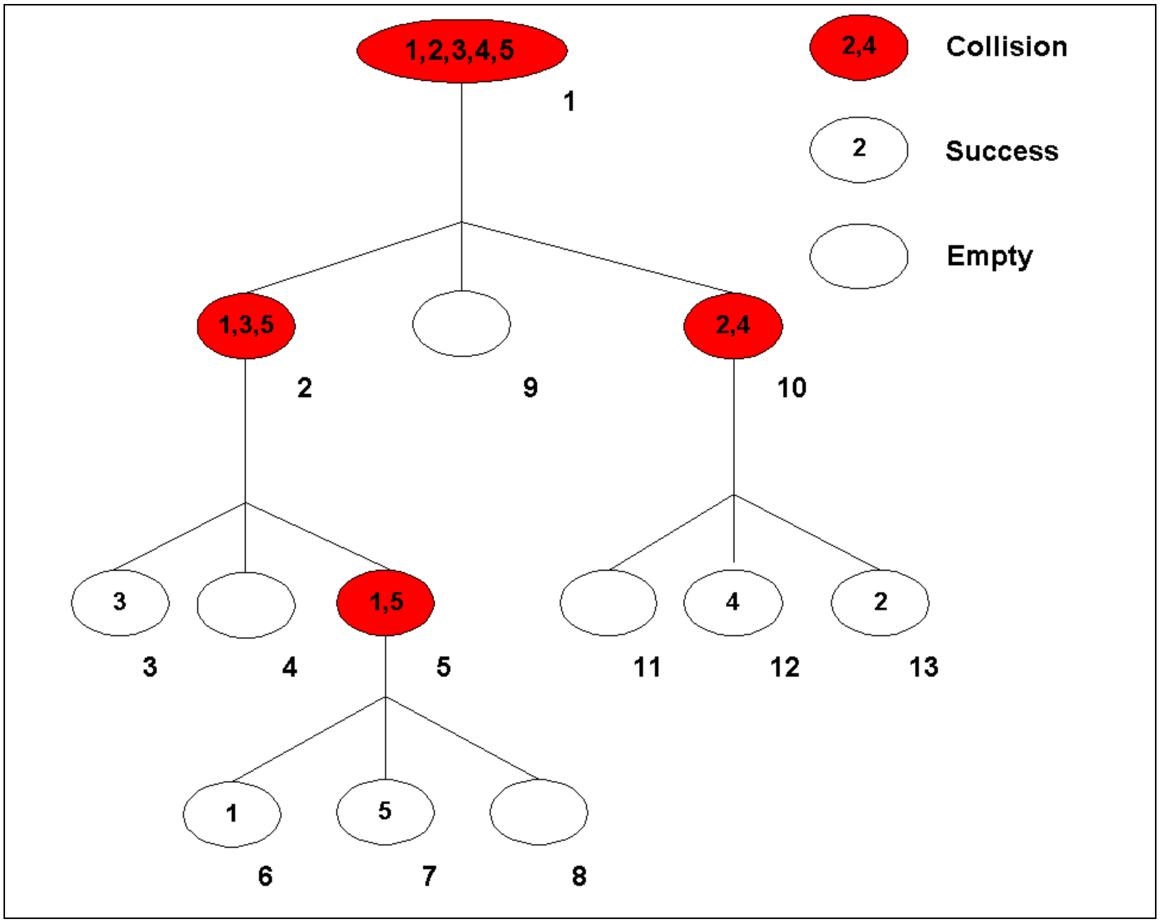

Figure 2.4

The newcomers' transmission policy (FTR – first transmission rule) has a strong influence on the performance of the tree-algorithms. *Blocked access* retains new requests until the current contention is resolved, i.e. newcomers are not allowed in the contention slots used for retransmission. *Free access* allows newcomers to send immediately, in any contention slot. The order of collision resolution in the tree can be LIFO, in which new collisions are resolved first, or FIFO. It is convenient to visualize the n-ary tree algorithm as a stack, in which each level is occupied by a subgroup. Only the stations at level 0 are allowed to transmit. In Figure 2.5 there is an example, taken from [22], of such virtual stacks, showing the differences between blocked/free access and the LIFO (2.5a)/FIFO (2.5b) resolutions in the ternary-tree collision resolution algorithm.

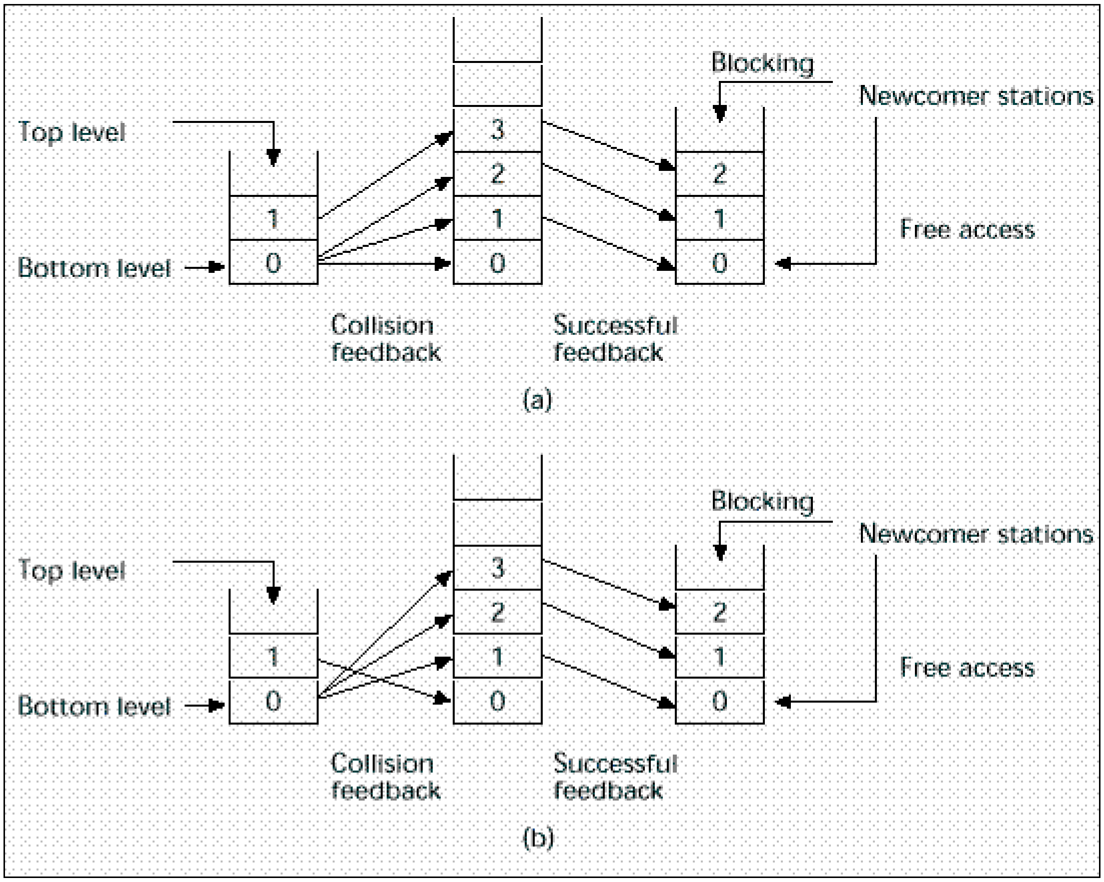

Figure 2.5

2.3.2 The proposed HFC MAC layer algorithms.

Here we wish to present some examples of MAC protocols and their respective collision resolution algorithms, proposed for HFC networks. The first suggested MAC protocols were distributed, in the sense that there was no one central control point and each station made its own decisions according to the protocol.

2.3.2.1 R-ALOHA.

The R-ALOHA scheme (Reservation ALOHA), a modification of Slotted ALOHA, was initially proposed for satellite networks [12]. Each time slot here matches one cell. In case of successful transmission in one slot, the corresponding slot in the subsequent frame is reserved for the station. Stations that have new data check the current frame. Any idle slot will be available in the subsequent frame.

2.3.2.2 XDQRAP.

XDQRAP (Extended Distributed Queuing Random Access Protocol), from the Illinois Institute of Technology [13][14], is a distributed algorithm in which each station maintains queues for transmission of both data and requests. The contention resolution algorithm is tree-based, and short one-cell messages can preempt long data messages. In the upstream channel, a data slot is followed by two (or three) contention slots. All stations must monitor for the feedback from the request transmission and update their data and request queues accordingly. In this way, the "source" station knows when to commence transmission, and the "destination" station knows when to commence reading the message. The head-end remains passive throughout this scheme. The distributed schemes, however, do not use the inherent central control point of the network – the HE. It is more difficult to meet QoS demands with distributed implementations and they are more susceptible to errors. Therefore, many centralized algorithms were also proposed.

2.3.2.3 MLAP.

MLAP (MAC Level Access Protocol) by IBM [15] divides the upstream into frames of variable lengths, which are called blocks. Each block contains a number of contention slots and a number of data slots, and each *data slot encapsulates an ATM cell*. MLAP assumes that the HE scheduler can prioritize transmissions, as the stations can have a number of queues for different data sources, based on priorities, and can send priority information with requests. Stations can also use "piggybacking". The algorithm used to resolve collisions in MLAP is START-n (n-ary Stack Resolution). START-n actually runs a free-access LIFO n-ary tree for each collided slot. It is implemented by using a simple counter at each station to simulate the virtual stack previously described. The collided stations set their counter randomly between 0 and n-1, while non-collided stations that are "waiting on stack" increment their counter by n-1. In case of success or of an idle slot, all participating stations decrement their counter by 1. Each station is allowed to run a number of START-n engines at the same time, so there is no need for them to wait the RTD time until receiving feedback before making a new request. This technique, which is used to compensate for long delays in HFC networks, is commonly known as *interleaving*.

2.3.2.4 ADAPT+.

The ADAPt+ MAC protocol from Bell Labs [16] also relies on centralized control by the HE. The protocol defines frames of fixed sizes, in which the head-end allocates the first regions for isochronous traffic (i.e. telephony) and the rest for available bit rate traffic. In the latter part, bandwidth is available for both request and data transmissions (contention mode) and the rest of the bandwidth is left for the reservation mode. The protocol supports data carriage in ATM cells. No original CRA is proposed in ADAPt+, and the authors suggest using any well-known algorithm.

2.3.2.5 CPR.

CPR (Centralized Priority Reservation), by John Limb and Dolors Sala from the Georgia Institute of Technology [17], uses the HE to manage the request and data channels in the upstream, and the grant and data channels in the downstream. This is achieved with knowledge of the exact delay for each station, and by "sandwiching" a number of contention slots between each data slot in the upstream. In the downstream, there must be exactly the same number of ack/grant mini-slots. After transmitting its request, the station must monitor in the downstream for the ack/grant mini-slot that will appear exactly after the RTD time. If the request was successfully received, it will contain an acknowledgement for the station. The same mini-slot, or a later one, will include in its second part a grant for the station, giving it the right to transmit in a number of data cells. The station will immediately transmit in the subsequent data cells. The p-persistence algorithm is used to resolve collisions.

2.3.2.6 Continuous mode with p-persistence.

The same authors later suggested [18] not to impose a framed structure on the upstream channel. In the beginning, all the mini-slots are open for transmission of contention requests. As the successful requests arrive at the HE, it will reserve slots for data transmission in the upstream. This way the mechanism is self-regulating – at low loads there are plenty of contention slots, and when the load is high and there are not enough CS, the requests cannot be sent and therefore more contention slots will be allocated. The CRA that is used is once again p-persistence, and “piggybacking” is permitted. However, this simple scheme works well only when the propagation delay is short – about 1 MS long - and less so when the distance to the HE is sufficiently long (40 km, in this paper). The problem is that in the latter case, many requests are accumulated during the RTD time, which leads to a long burst of data slots allocated by the HE, therefore there will be even more requests waiting until the end of the data burst, etc. The proposed solution to this problem is to periodically insert a number of contention slots “by force”. These slots are called FMS (forced mini-slots). The amount of FMS is estimated as follows: for short one-cell bursty traffic, there are *e* FMS per data slot, as the maximum throughput of ALOHA is $1/e$, or less at lower loads. In this case, authors suggest using 2 FMS. In general, if “piggybacking” is used and the average request size is *k* cells rather than 1, the proposed formula is $N_{fms} = (1-\lambda_p) * e/k$, in which λ_p is the “piggybacking” arrival rate. We wish to note that the use of FMS reduces the proposed self-regulating continuous mode to a kind of clustered mode scheme with small frames.

2.3.2.7 PCUP.

The last MAC level protocol we would like to mention here is PCUP [19] - Pipelined Cyclic Upstream Protocol. This protocol operates in two modes: *cyclic transmission* mode and *negotiation mode*. In *negotiation* mode, the HE runs a *membership control* algorithm in order to permit the off-line stations to join in. Every 0.5 seconds or more, the HE sends a special *invitation* frame to all the inactive stations. Stations that were inactive since the last membership control become off-line and do not receive the current invitation. Then the HE performs *positioning*. In positioning, a transmission start time is assigned to each station, in a way that neutralizes propagation offsets. Data from different stations arrives in sequence to the HE and further away stations can start transmission before closer stations complete theirs. During positioning, the HE performs ranging and classifies the stations by an ascending order of distance. Accordingly, the transmission commencement time for station i , s_i is computed as $s_i = \sum_{j=1}^{i-1} t_j - \tau_i$, in which t_j is the transmission duration and τ_i the propagation delay. Prioritized bandwidth scheduling (i.e. t_j computation) is performed at the end of the cycle, for two traffic classes: guaranteed and best-effort. Also, the station may be instructed to change its upstream channel. A special frame sent to the station specifies the new transmission frequency. During the cyclic transmission, each station transmits in the allocated interval. The size of the data slot is designed to match the ATM cell. In the last slot, the station sends its buffer status to the HE in order to facilitate the subsequent cycle scheduling. Although interesting, this algorithm appears to have the following problems: a complicated membership control algorithm has to be run by the HE; if a station is considered off-line, it will not have an opportunity to transmit until the following membership control – a 0.5 second delay; if the stations do not have a meaningful propagation offset, e.g. all the stations are concentrated in the last mile of the network, which is the common case, the protocol will lose its ability to “pipeline” the data; and conversely - for the positioning to work properly, the exact place of each splitting point in the network and the delay from each branch to this point have to be known, otherwise the data may collide.

2.4 Standards and Standard Bodies.

2.4.1 Standard Bodies.

In 1994, the IEEE 802.14 Work Group was chartered to create standards for the physical (PHY) and MAC layers of HFC networks. The group set out to provide high-quality services, while taking future technologies and support for QoS into account. However, it failed over a long period of time to produce a final specification. In 1997, a consortium of North American MSO's (abbreviated as MCNS - Multimedia Cable Network Systems) was formed. Within a year they created their own standard for the PHY and MAC layers. The standard is called DOCSIS (Data-Over-Cable Service Interface Specification), and was designed to enable minimal customers' equipment cost, short production and deployment cycle and support for existing technologies. Yet another standard exists - DAVIC/DVB (Digital Audio Video Council/ Digital Video Broadcasting). This standard was developed in Europe and is recognized by a number of leading European MSO's as a preferred technology. The work is now run by DVB and the standard is also known as DVB-RCC and as ETS 300 800.

2.4.2 IEEE 802.14 and DOCSIS key features.

In the following section we draw a short comparison between the DOCSIS and IEEE 802.14 standards. Despite the fact that the IEEE WG failed to produce a market standard, it spent a considerable amount of time selecting the best technology and achieved very interesting results. Also, it is more worthwhile to compare their MAC layer with MCNS, since there are more differences to be found. The following tables illustrate key features of the PHY and MAC layers in the standards [20][21].

PHY layer	IEEE 802.14	DOCSIS
Upstream frequency	5-65 MHz*	5-42 MHz
Downstream frequency	88-860 MHz*	50-850 MHz
Downstream channel size	6,8 MHz	6 MHz
Upstream modulation	QPSK, 16 QAM	QPSK, 16 QAM
Downstream modulation	64 QAM, 256 QAM	64 QAM, 256 QAM
Upstream symbol rate	160-5120 Ksym/sec	160-2560 Ksym/sec
Upstream data rate	320 Kb/s – 20.4 Mb/s	320 Kb/s – 10.2 Mb/s
Downstream data rate	41, 56 Mb/s	27, 36 Mb/s

* The IEEE standard allows different upstream/downstream frequencies. For North America they are 5-42 MHz and 88-860 MHz, for Europe they are 5-65 MHz and 110-862 MHz, and for Japan they are 5-55 MHz and 90-770 MHz, respectively. Also specified are three downstream PHY types, called A, B and C. These types differ in channel size – 6 or 8 MHz - and in error correction methods.

MAC layer	IEEE 802.14	DOCSIS
Carried traffic	ATM cells.	IP packets.
Mini-slot duration	8 bytes + PHY overhead (6 bytes of data and 2 bytes CRC).	16 bytes of data + PHY overhead.
Data Packet (PDU) size	Fixed – ATM cell + overhead. PDU must comprise an integral number of MS.	Variable length, 0-1500 bytes of data payload. PDU's can be concatenated.

Source identification during operation	14 bit LI (Local ID) for station + 10 bit LQ (Local Queue).	14 bit SID (Source ID).
Frame structure	The frame duration is defined by the HE. CS in the frame are allocated in clusters of variable size.	The frame duration is defined by the HE. The frame is composed of separate Information Elements (IE). The IE's define one or more mini-slots, each IE can be allocated for CS.
Access modes	Reservation, Piggybacking and Constant Bit Rate.	Reservation, Piggybacking and Immediate.
Contention resolution	Priority + FIFO for newcomers, N-ary tree for retransmissions. Multiple engines.	Truncated binary exponential backoff + backoff window.
HFC network parameters	Maximal distance to the HE – 80 km. Maximum propagation delay - 200 µsec.	Maximal distance from the station to the HE - 100 miles. Typical distance is 10-15 miles. Maximum propagation delay - 800 µsec. The typical propagation delay is much less.

Following is a more detailed description of the above CRA's.

2.4.2.1 DOCSIS Collision Resolution.

Binary Exponential Backoff. Two numbers, that are an exponents of two, are specified by the HE – Data Backoff Start (DBS) and Data Backoff End (DBE). A station that has a new request uses DBS to randomly select a number within the range of [0, 2^i], i = DBS. The station postpones its transmission request for this number of CS. If a collision occurs, the station increases the range by 2 and selects CS from this wider range, until the DBE range is reached. If a maximum of 16 retries is reached, the data must be discarded (truncated). The Binary Exponential Backoff algorithm is actually used in the Ethernet, when the DBS and DBE are different. If the HE sets DBS = DBE, the result is a variant of the p-persistence scheme, in which the station's probability to transmit is $1/2^i$ in each CS, and it will send with probability 1 during the subsequent 2^i available contention slots.

2.4.2.2 IEEE 802.14 Collision Resolution.

Priority FIFO + N-ary tree. New requests are ordered using a time boundary. Only requests that arrive before the Admission Time Boundary (ATB) are permitted to contend. The ATB is dynamically provided by the HE. In addition, an 8-bit vector $(P_1, P_2, \ldots, P_8)$ is used to impose priorities on newcomer requests. A newcomer of priority k is admissible only when $A(k) = \sum_{n=1}^{k} P_n > 0$ and it will only use contention slots that are allocated for this priority group, as defined by the priority group descriptor field, whose number equals *A(k)*. When the HE detects a collision in a CS, it assigns the collision a number, in ascending order starting from 1, called an RQ (Request Queue). Before the subsequent frame, all stations are notified about the collided mini-slots and the corresponding RQ's. The HE then allocates a group of N contention slots for each RQ, in descending order. Each station that receives a collision feedback knows the RQ corresponding to the collided CS, and retransmits in the contention slots with this RQ. Retransmission is carried out by selecting a random number *k* in [0,N]. If the RQ group size is greater than, or equal to, *k,* the station transmits in the *k*-th slot of the group. If not, it has to wait for the subsequent frame(s), in which the remainder of the RQ group slots should be allocated. Newcomers use RQ == 0, so all the backlogged stations receive their allocations and have an opportunity to send first. The constant default value of N is 3. The IEEE 802.14 retransmission scheme actually represents multiple N-ary trees with LIFO stack ordering running simultaneously.

2.4.3 Performance of the CRA's and their variants.

Many different algorithms and algorithm implementations were reviewed by the IEEE WG during the standartization process, before the above scheme was voted in. The most important performance measures for a CRA, that were also studied during the simulations, are: Mean Access Delay (AD) – the average transmission time for the data from the moment it arrives at the station until it is received at the HE (in some studies – until the data is sent); Mean Request Access Delay (RAD) - the average time from the moment a new request is generated at a station as a result of new data arrival until the grant for this request is received from the HE; Mean Throughput – the average percentage of bandwidth used to carry the data payload in the HFC network.

The conclusions that were drawn are:

- The mean AD for p-persistence is higher than for ternary-tree, starting from a medium (30%) offered global load. As the global load increases, the collision resolution time for p-persistence grows, while for ternary-tree it is constant, due to backlogged stations and the separate contention slots for newcomers. There is no upper bound on the access delay for p-persistence.[22][24]
- The FIFO ordering shows the best performance as the First Transmission Rule. It has slightly lower collision multiplicity and mean access delay than either p-persistence (which was the FTR proposed in the earlier drafts of the IEEE standard) or the free access and blocked access methods. Most importantly, it has a significantly lower access delay variance than the p-persistence scheme, due to the ordering imposed on the requests.[23][24]
- The ternary-tree algorithms with blocking access consistently perform better than the free access, for all global offered loads. An exception is when the maximum request size equals 1, in which case the free access scheme performs slightly better. [24]
- The comparison between the *continuous* and *clustered* modes (clustered - i.e. using frames) of operation showed that ternary tree in clustered mode performs better than both the ternary tree and p-persistence in continuous mode, in terms of mean access delay, collision multiplicity and delay variance, and especially at high loads. This holds true for Poisson and self-similar traffic types. However, there is no advantage to the clustered mode p-persistence over the continuous mode p-persistence.[25]

It should also be noted that the standards do not define the bandwidth allocation and data scheduling algorithms run at the HE, and do not affect interoperability. A study of the standard CRA's, also with very "forgiving" simulation parameters (20 stations and an 8 μsec mini-slot duration, which corresponds to a 7 or 14 Mb/s upstream bit rate) was conducted in [26]. Here it is shown that the IEEE algorithm has a lower RAD at a medium load than MCNS, and the performance is similar at high and low loads. The study used "piggybacking", which shortens the delay at high loads. The bandwidth allocation policy that was used was to allocate 6 CS for newcomers in each frame and 3*C, C being the number of collided slots in the last frame, for retransmissions in the IEEE algorithm. Also, the FTR rule used for the IEEE algorithm was the ideal p-persistence, i.e. the HE could know the number of contenders. One of the conclusions drawn in the paper is that avoiding collisions is more important than resolving them quickly, and that the allocation of mini-slots for newcomers should depend on load, while the allocation of mini-slots for retransmissions should depend on the number of observed collisions. Authors showed that setting a backoff window with DBS = DBE, which is dynamically adjusted with the load, performs better than the Ethernet-like setting, because the former "tries to avoid collisions in the first transmission".

2.5 Market Statistics.

Currently, there are over one million cable modem subscribers in North America. Here are some of the latest market statistics and projections:

1. The largest broadband access providers in the US are Excite@Home and Road Runner, servicing about 70% of the market. Another emerging giant is AOL Time Warner, created as the result of a merger between America On-Line and the Time Warner Company.
2. Time Warner cable systems are avilable to 20 million homes and serve 13 million subscribers, finishing 1999 with 300,000 cable modems installed. Adelphia Communications announced it finished 1999 with 37,495 cable modem customers, up from 15,439 at the end of 1998. Insight Communications reported closing 1999 with 8,300 cable modem customers, out of a total of 542,000 homes targeted as a market for high-speed services. The Canadian MSO Shaw Communications reported finishing its second fiscal quarter, which ended on February 29, with 219,016 @Home cable modem customers and 1.6 million homes marketed, yielding a phenomenal 13.7% penetration rate. United Pan-Europe Communications N.V. (UPC) announced it finished 1999 with 117,925 residential cable modem customers, up from 79,039 at the end of 1998.
3. It is estimated that by the end of the year 2000, there will be approximately 2 million cable modems installed in North America and 2-4 millions in the rest of the world (the estimates differ). Other significant markets will include Canada, Japan and the Netherlands. Currently, there are twice as many cable modem subscribers as rival DSL services.
4. Approximately 11-19% of shipped cable modems are DOCSIS compliant. Cable modem shipments and market share figures are as follows:

   Motorola: 760,000 (44%)
   Nortel: 490,000 (28%)
   Terayon: 145,000 (9%)
   Com21: 140,000 (9%)
   Zenith: 35,000 (2%)
   GI: 30,000 (2%)
   3Com: 25,000 (1.5%)

3 The Simulation Framework.

3.1 Network Configuration.

All the simulations were performed using an extensively modified version of the NIST ATM/HFC Simulator v. 4.0. This short section presents the common simulation parameters. The HFC network is represented by a HE with attached stations. Each station is a logical unit, able to produce its own type of traffic at a given bit rate. Each unit represents a single logical queue in the upstream. All stations commence work at the same time, and produce the same type of traffic in a simulation, unless stated otherwise. The stations are located at a specified distance from the HE. The downstream bandwidth is not considered limiting. In the outward direction, the HE is connected via a 30 Mb/s link to an ATM switch, which is in turn connected to a single server station that serves simply as a sink for the incoming traffic.

3.2 Simulation Parameters.

The following table summarizes the numerical parameters common to all the simulations. Note that they were chosen so as to represent an average HFC network, in terms of performance, in accordance with the IEEE 802.14 WG recommendations and with the simulation parameters presented in most of the other papers regarding HFC networks, which were studied in the course of the work.

Simulation Parameter	Value
Upstream Data Rate	3 Mb/s
Downstream Data Rate	30 Mb/s
Propagation Delay	5 μs/km
Mini-slot Size	16 bytes (42.6667 μs in the upstream)
Mini-slots per Data Slot	4
Max. station request size	32 cells
Interleaves per frame	1
HE processing time	0

3.3 Traffic and Results Representation.

The traffic is carried in ATM cells. It fits in 64 bytes, out of which 48 bytes are data payload, 5 bytes for ATM cell header and another 11 bytes are for PHY/MAC headers and guardband. We chose to generate a short Poisson traffic of 1 cell size in order to stress the system and to study performance for this very bursty traffic type scenario, which is the most problematic.

In order to study the CRA's performance and separate it from the data scheduling delay, the Request Access Delay (RAD) parameter was used in the simulations. The RAD and AD are plotted in mini-slot units on graphs. The global load is plotted in percentage of the upstream bit-rate. The typical "warm-up" time, i.e. the initial period not included in the overall result, is 1 sec. In the clustered mode simulations, the contention slots are located at the beginning of the frame and their number is limited, so that a station sending a request in the last CS will receive feedback before the next frame begins.

4 P-persistence modifications.

4.1 The newcomers' ranging parameter.

As mentioned above, the pseudo-Bayesian estimator is used in the p-persistence CRA in order to estimate the number of contenders. When the estimator is used in the clustered mode, the formula will be $R = \max\left\{\min\left(n_{st}, R_{prev} - n_i - n_s + \frac{n_{col}}{e-2} + \frac{CS_{prev}}{e}\right), CS\right\}$, in which n_{st} is the number of active stations, n_i, n_s, n_{col} are the number of idle, successful and collided contention slots in the previous frame and CS is the number of contention slots in the subsequent frame. R is computed at the HE at the end of the contention region in the current frame, and is sent to the stations so as to arrive before the next frame begins. *R* is an important parameter, used in most of the following simulation algorithms, and has an impact on their performance. It is called the *newcomers' ranging parameter*. Note that in the continuous mode, any station that wishes to send a request decides for each CS independently whether to transmit or not, with probability $P_{send} = 1/R$. If there are a number of consecutive contention slots in the upstream channel, the station is entitled to attempt transmission in each one of them. Then the probability to send in the *i-th* CS will be $P_{send}^{i} = 1/R * (1 - 1/R)^{i-1}$, i.e. it decreases with each consecutive slot. The scheme for the clustered mode is different. Here, the number of CS in the next frame is known in advance. Say, for instance, that there are *k* slots open for newcomers. A station that has received new data before the beginning of the *i-th* CS, $0 \le i \le k-1$, is permitted to try once per frame, with probability $P_{send} = (k-i)/R$. In case of the data arriving before the beginning of the frame CS region, it is just k/R for any contention slot. The following results compare the above two p-persistence implementations for the clustered mode.

4.2 Simulation Results. Multiple vs. One-Choice Schemes.

Here we show three average simulation results, out of a number of simulations that were conducted, that best explain the behavior of the two p-persistence modifications. The duration of each simulation was 20 *sec,* with a 2 *sec* warm-up time, and the global load was increased during simulation time from 10% to 60%. Figure 4.1 represents experiments in which *R* was fixed and equal to the number of contention slots in the frame. For a frame of 35 MS and a CS to DS ratio of 1, this gives $R = 7$. This actually means that any station that wishes to transmit a request, whether it is a newcomer or backlogged, will be able to attempt transmission, with this value of the newcomers' range. It can be seen that the multiple-choice scheme performs better here as the load increases. With the above value of *R* at over a 30% load, the number of contenders in each frame becomes greater than the number of CS available, and all will attempt to transmit in every frame. Then the default p-persistence scheme distributes

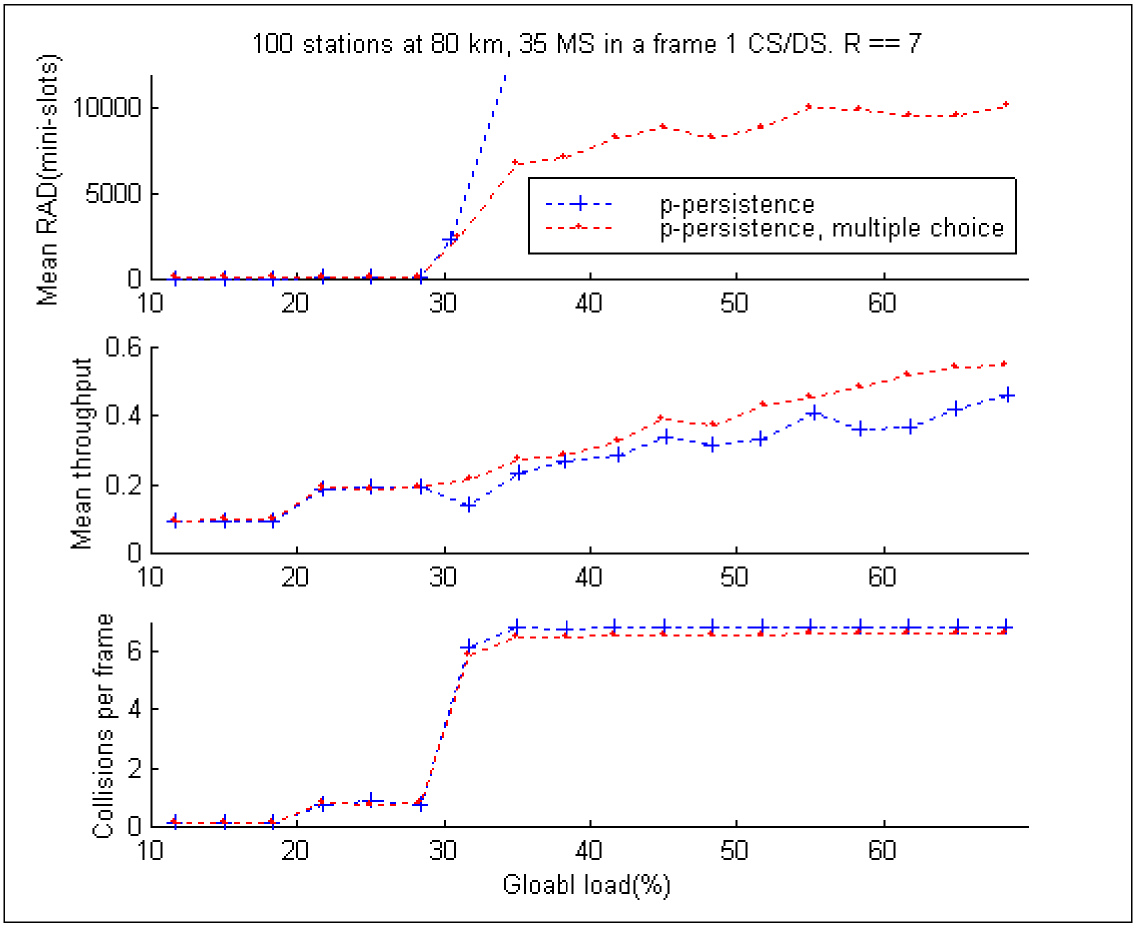

Figure 4.1

the requests equally among all the contention slots, and causes repeated collisions in every contention slot available, which in turn increases RAD and lowers throughput. The multiple-choice scheme, on the other hand, has a greater probability of transmission in the first contention slots, thus the last slots will have fewer collisions and the stations that choose them will succeed sooner. It should be noted that the appropriate graph is not presented here - we actually saw that in the multiple-choice scheme, the amount of collisions decreases as the contention slot index increases in a frame. When R is changed by the HE, according to the pseudo-Bayesian estimator (Figure 4.2), there is no significant difference in the performance of the two algorithms. It is clear that R has a major influence that reduces the differences between the two p-persistence variants. As the load increases, R will also increase, and instead of repeatedly retrying to send their requests, as in the case of $R = 7$, more and more stations will be "held off" transmission. In this situation, there will be no advantage to freeing the last contention slots, as all the slots will be appropriately freed by the increasing ranging parameter value. Moreover, it can be seen that the mean request access delay significantly decreases, in comparison to the simulations with the constant R value. This is due to the fact that more collisions are prevented, which lowers the average RAD - despite the fact that the stations must wait more time before attempting to send a request.

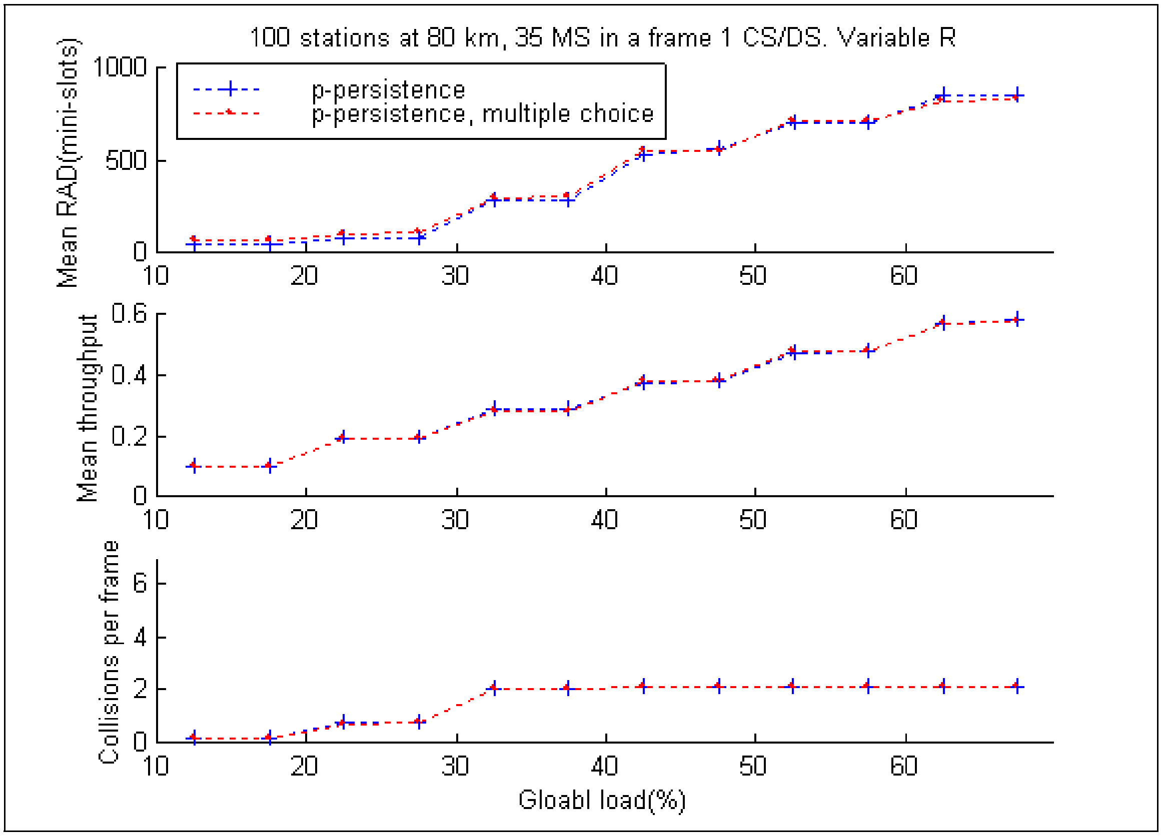

Figure 4.2

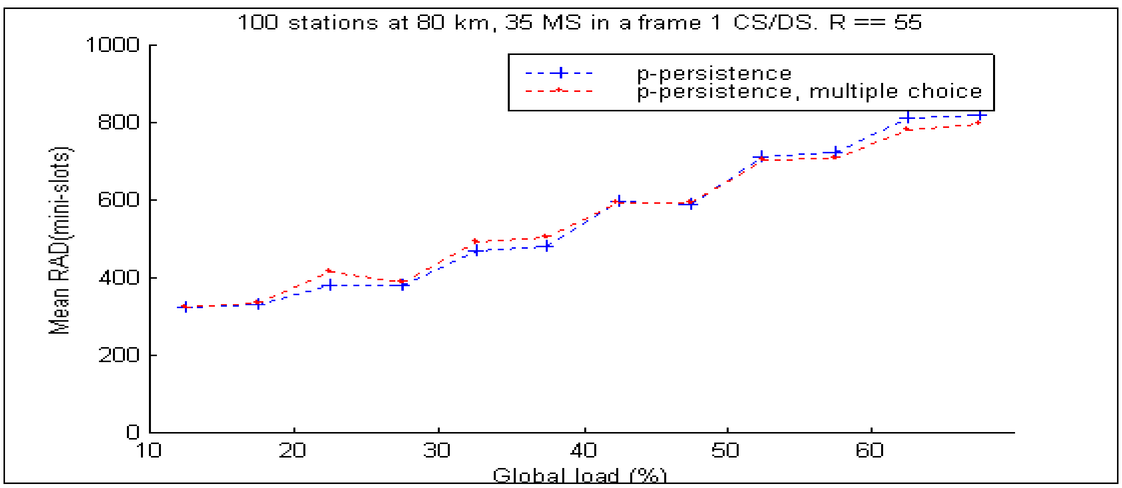

Figure 4.3

Figure 4.3 further proves this point. The figure represents the RAD of the same simulation, while R is constant, but high. The value $R = 55$ is close to the one produced by the pseudo-Bayesian estimator at a 50% global load. The high value of R reduces the differences in transmission probability for different contention slots, in the multiple-choice algorithm. For example, with the given parameters, for the case of $R = 7$, $P^1_{send} = 0.1428$, $P^7_{send} = 0.0567$ in the multiple-choice scheme, and for the case of $R = 55$, we have $P^1_{send} = 0.0181$, $P^7_{send} = 0.0163$ - almost no difference. In addition, it can be seen that at low loads, the multiple-choice scheme does not perform quite as well as the usual one, because it creates more collisions in the first slots, which is unnecessary at a light load, as there are enough CS for all contenders.

4.3 Conclusions.

From the above, we can conclude that the influence of the ranging parameter on all types of the p-persistence algorithm is very strong, and is much more significant than the differences in sending probabilities. Still, at high loads there is an advantage to assigning different transmission probabilities to the different contention slots in a frame, especially when not only the global load on the upstream is high, in terms of bit rate, but also the number of collisions per cluster. Simulations that were conducted with the same parameters as above, but with 200 stations transmitting, instead of 100, showed that the request access delay for the multiple-choice scheme was 1-2 *msec* lower, on average, and 50-80 *msec* maximum, than the one-choice scheme at loads as high as 50%.

5 Impact of the HFC Network Architecture Parameters on Performance.

5.1 Motivation.

As was mentioned in the previous chapters, the delay in the HFC networks is of significant value, due to the networks' topology, and it influences the MAC level algorithms and the networks' performance. It is interesting to see how precisely the distance to the HE changes the delay values. Is it worthwhile to bring the HE closer to the stations, or maybe to place an "intermediate HE" somewhere between the distribution and the access networks? Another parameter in the HFC network architecture that can be changed is the number of stations served by the HE. This parameter's value is typically established by the trade-off between the cost to the service operator of serving a smaller number of stations, and the quality of service available to the network users.

The results in this chapter are drawn from simulations that were conducted in order to study the above two parameters, and show how they influence the different contention resolution algorithms.

5.2 Simulation Results. Distance vs. Number of stations.

Figure 5.1 shows the results for the p-persistence algorithm, and Figure 5.3 shows the results for the IEEE 802.14 algorithm, as described above – FIFO with ATB for newcomers and Ternary Tree for collision resolution. This algorithm will be addressed in the following discussion as Ternary Tree Unblocking (Ternary Tree UB), in order to distinguish it from the Ternary Tree R - with p-persistence access for newcomers based on the ranging parameter - or simply, Ternary Tree.

5.2.1 The Distance Impact for p-persistence.

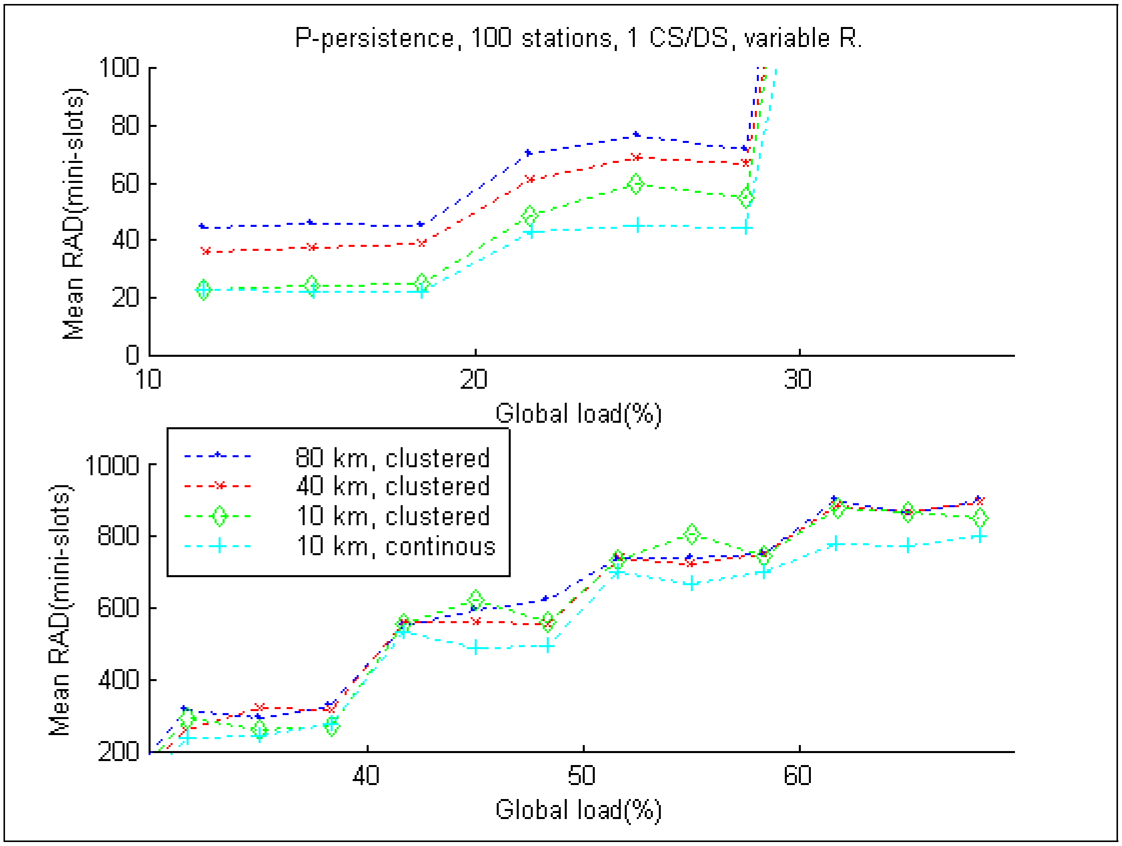

Figure 5.1

The results for p-persistence show that some improvement, although not very significant, is reached when the distance to the HE is decreased. At low global loads, which are less than 30%, the average difference in RAD between the 80 km distance graph and the 10 km distance graph is about 15 mini-slots, for clustered mode, which is a 20-30% percent improvement on the 80 km result. The continuous mode performs better at a very short distance of 10 km, as expected, since the frames here become very small, so there is no reason to impose a framing structure at all. This mode gives a 25 mini-slot (35-55%) improvement, on average, at these loads. At higher loads, the difference for the clustered mode algorithms is even smaller, in terms of RAD percentage. The average is only 50 mini-slots in the best case, while the mean request access delay is as high as 500-800 mini-slots. The continuous mode p-persistence, at a 10 km distance, has a stable difference in mean RAD of 100-120 minis-lots, compared to the clustered mode at an 80 km distance, at high loads. Figure 5.2 shows that the same pattern applies when the number of CS increases to a ratio of 2 CS per DS, despite the fact that the mean RAD is significantly reduced.

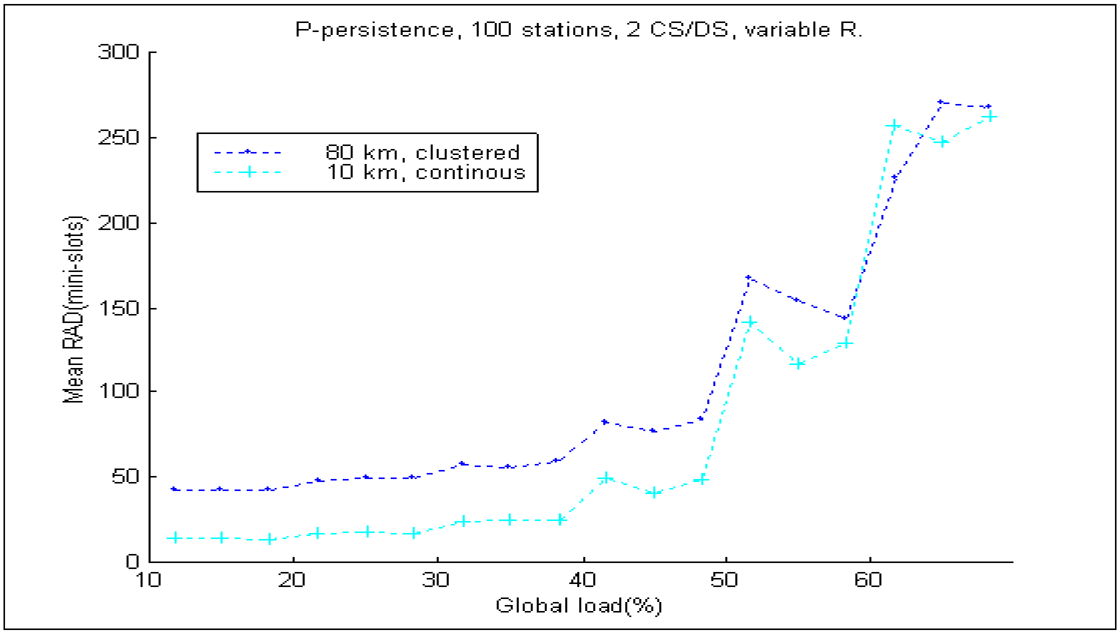

Figure 5.2

5.2.2 The Distance Impact for Ternary Tree.

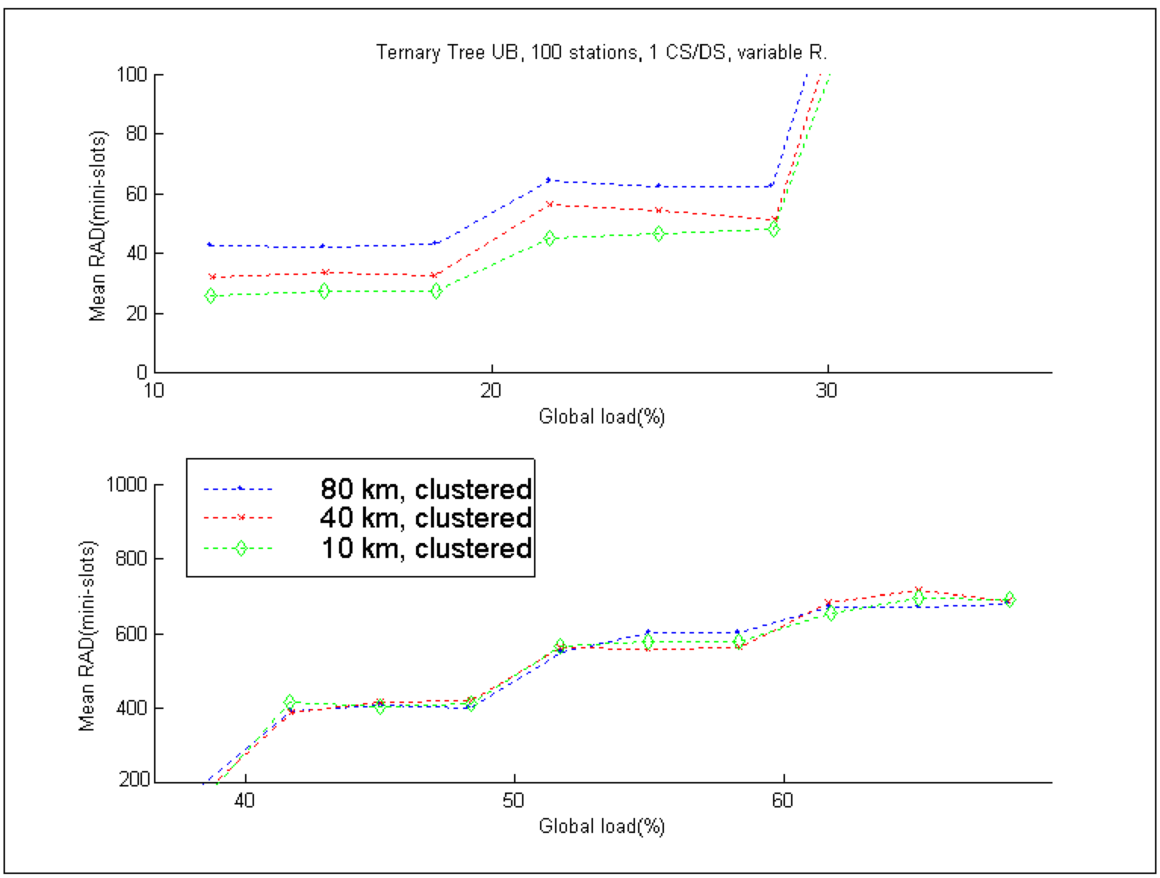

Figure 5.3

The results for Ternary Tree are quite similar. It can be noted that the mean RAD is lower than for p-persistence, as expected, because the newcomers are separated from the backlogged requests and the collided slots are simultaneously resolved here. However, there is no significant improvement when the distance to the HE decreases, especially at high loads.

5.2.3 Conclusions on the Distance Impact for Algorithms with a Fixed Bandwidth Allocation.

We conclude that decreasing the distance to the HE does not improve the CRA's performance, when other simulation parameters are fixed. The only figure that remains variable is the ranging parameter *R*. The decreased distance allows the HE to adjust *R* faster and within shorter time periods, and accordingly, the stations will receive *R* faster in the downstream and its value will be more precise (the ideal situation here is when the feedback is immediate and *R* is updated before the beginning of each mini-slot). P-persistence is more influenced by the value of the ranging parameter than Ternary Tree UB, in which *R* only influences the newcomers' admission, but not the collision resolution. In the pseudo-Bayesian formula, *R* changes by some constant amount according to the last contention slot status. We shall call it δ, $\delta = \begin{cases} \frac{1}{e-2} & collision \\ -1 & idle/success \end{cases}$. If there are *N* contention slots during some round-trip time, R will change during this time like so: $R = R_{prev} + \lambda + \sum_{i=1}^{N} \delta_i$. As the load grows, *R* will grow with it and will become significantly greater than any $\sum_{i=1}^{N} \delta_i$. So, the difference between updating *R* each time by an appropriate δ_i or by the whole sum at the end of the round trip time, becomes insignificant, compared to the value of R. Therefore, not much can be gained by adjusting *R* faster. The CRA's performance here depends more on effective collision resolution and collision prevention than on distance.

5.2.4 Distance Impact on Algorithms with a Variable Bandwidth Allocation.

We know that the bandwidth allocation in the upstream can also be varied, meaning that the HE can dynamically decide, based on some policy, how many mini-slots in the next frame will be assigned to contention and how many to data. It would seem that with such a bandwidth allocation mechanism, more could be gained from decreasing the distance to the HE, because at small distances, frequent changes in the bandwidth allocation should help resolve collisions more efficiently, and prevent wasting excessive bandwidth. Figure 5.4 shows the results of such simulations for the Ternary Tree UB algorithm.

5.2.4.1 A Simple Bandwidth Allocation Algorithm.

A simple bandwidth allocation scheme was used. The number of contention slots in the subsequent frame was computed to maximally satisfy the number of contention slots needed to resolve all the collisions in the previous frame. In the case of Ternary Tree, 3 CS must be allocated for each collision in the previous frame. The maximum number of CS per frame is limited, so all the stations will receive their feedback before the beginning of the subsequent frame. The number of CS is computed first. Then mini-slots that cannot be used for data slots (remaining after the allocated CS are subtracted from the frame size, and the resulting number is divided by the number of mini-slots in one DS) are converted into contention slots. Also, if there is not enough data to be scheduled to fill all the data slots that are available after the CS allocation, these free mini-slots are converted into contention slots. If no collision occurred in the previous frame, the number of contention slots in the subsequent frame will be equal to the number of DS that can be accommodated in the whole frame, but not less than 2.

5.2.4.2 Simulation Graphs Explanation.

The graphs on Figure 5.4 represent simulations conducted with the Ternary Tree UB algorithm, in clustered mode, with 100 stations attached to the HE. The first two graphs represent simulations conducted using the variable bandwidth allocation scheme described in the previous paragraph. The third graph, shown here for comparison purposes, represents a simulation that used fixed bandwidth allocation with a 1 CS/DS ratio, as before. The graphs in the upper part of the picture represent the mean request access delay. The 10 km distance network, with the variable bandwidth allocation, has a significantly lower RAD than the 80 km distance network RAD. The RAD remained lower than 50 mini-slots until the end of the simulation, while the mean RAD for the case of the 80 km distance reached about 210 mini-slots at a 60% load. The RAD for the fixed bandwidth allocation grew much faster than for the variable allocation, starting at 30% of the global load applied. The graphs in the bottom part represent the mean access delay time. This graph clearly shows that both variable schemes behave very reasonably, and begin to exhibit a high access delay only at a 60% load. This is due to the bandwidth allocation scheme, that allocates many mini-slots for contention (at these high loads it will actually have to allocate the maximal allowed number of CS), leaving too little for data, thus sharply increasing the access delay. Note that were piggybacking used, it could significantly relieve the problem of collisions at high loads, since most of the stations have some data to send and could most of the time piggyback their new requests.

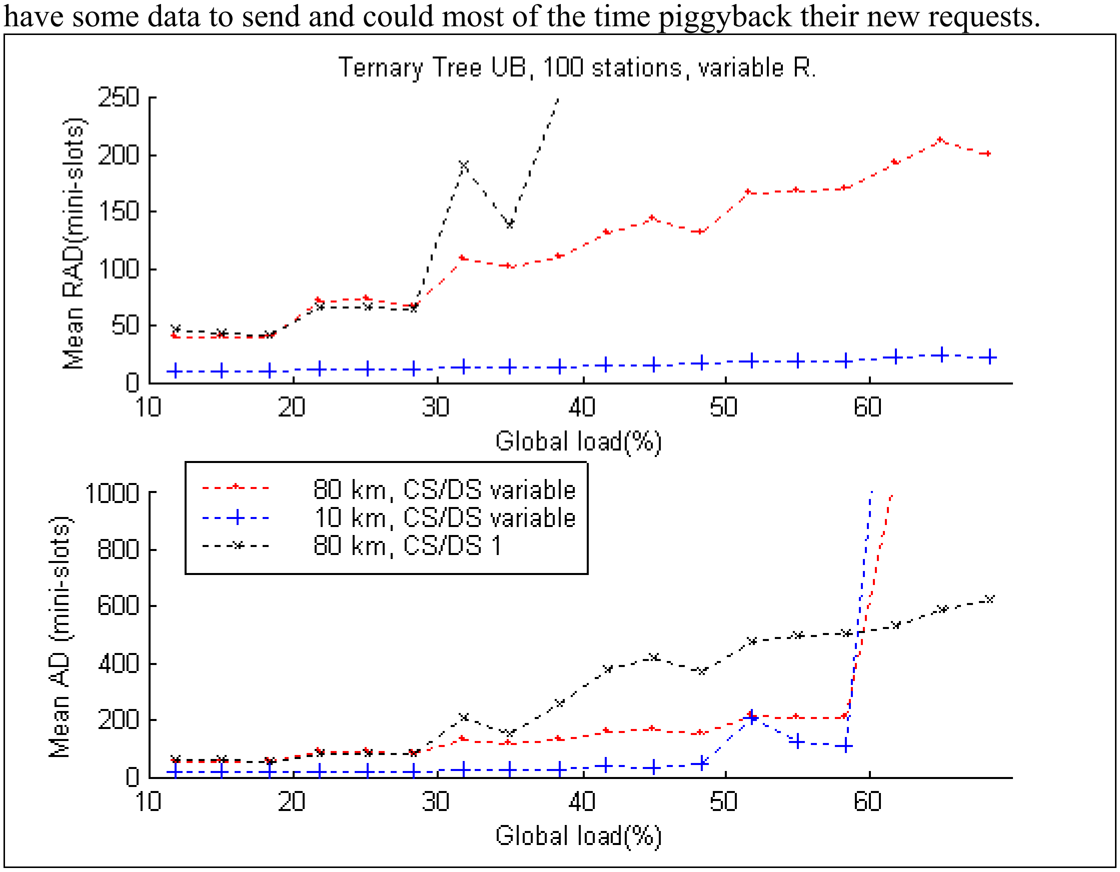

Figure 5.4

5.2.5 Changing the Number of Stations.

The last figure for the chapter helps compare the effect of changing the number of stations with the effect of changing the distance to the HE. The difference is quite obvious: as the load becomes significant, the mean request access delay decreases linearly with the number of stations, for both the p-persistence and the Ternary Tree Unblocking algorithms.

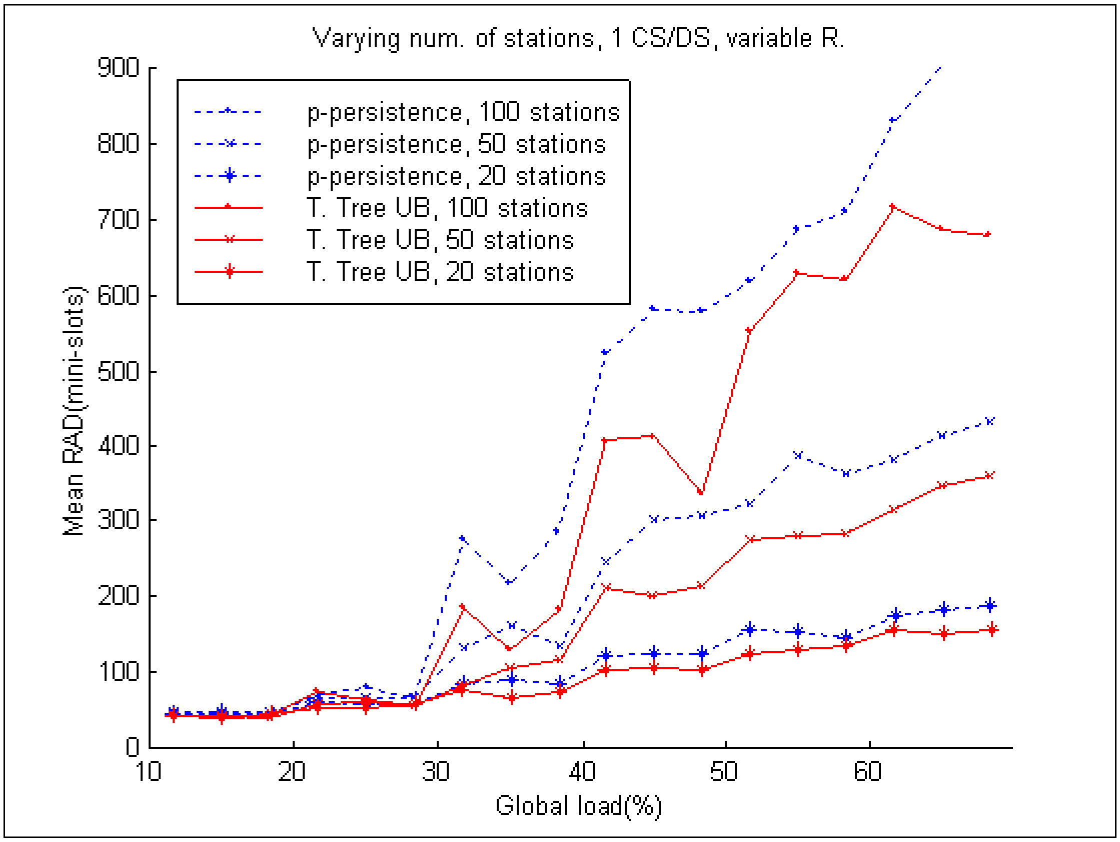

Figure 5.5

5.3 Conclusions.

Apart from upgrading/changing the hardware equipment and cables, the configuration of an HFC network can be altered in two ways – by changing the distance to the head-end, and by varying the number of subscribers attached to the head-end. From the above results we can see that any reduction in the number of subscribers will proportionally improve performance for both considered classes of algorithms. In order to gain some significant improvement from reducing the distance to the head-end, a variable bandwidth allocation, preferably with a splitting tree algorithm, should be used. This is due to the fact that when the bandwidth allocation is fixed, performance improvement cannot be achieved. The small round-trip delay time can only be used in this case for faster updating of the ranging parameter, but this is insignificant in comparison to the delay produced from the large number of collisions.

6 Dynamic Bandwidth Allocation.

6.1 Motivation.

From the previous chapter we concluded, as a side result, that the variable bandwidth allocation scheme performs better than a fixed CS/DS allocation, especially at high loads. This result agrees with conclusions drawn in a number of sources, such as [22][23][24][26]: *"…any fairly simple variable CS/DS ratio scheme gives better results than fixed policies"*[24]. In all the schemes mentioned in the papers referenced above, the HE performs bandwidth allocation based on traffic load estimation. This chapter presents a short description of an algorithm for variable bandwidth allocation proposed by K. Sriram and used, with minor modifications, in most of the previous works. Following that are the results of simulations comparing this algorithm to the scheme described in 5.2.4.1, and a discussion on the drawbacks of the algorithm's implementation.

6.2 Traffic-Based Bandwidth Allocation.

The formula for computing the number of contention slots in a frame was proposed by K. Sriram in [27]. The primary assumption is that the contention mini-slot throughput efficiency is 33%, because the maximal theoretical throughput for a simple CRA is the same as for the stabilized slotted ALOHA – 37%. Then, 100% efficiency should be achieved with a 3/1 CS/DS ratio. If a message consists of more than one data-slot, then for all these data slots, only one successful request can be transmitted to the HE. Denoting the average message size by ℓ, the authors suggest introducing a coefficient $f_m (f_m < 1)$, because the bursty message length is often shorter than average, and the formula for *effective message length* is $l_e = \max\{\lfloor l * f_m \rfloor, 1\}$. Then, denoting N – frame size in data slot units, k – the number of mini-slots in a data slot, and j_x – the initial value of bandwidth to be allocated for contention, we write an equation: $j_x * k = 3 * \dfrac{N - j_x}{\ell_e}$.

From this we receive the formula $j_x = \max\left\{\left\lceil \dfrac{3 * N}{3 + k * \ell_e} \right\rceil, 1\right\}$. If the following condition: $j_x \geq 2$ and $\frac{1}{3}(j_x - 1) * k * \ell_e \geq N - j_x$, which means that the throughput via $j_x - 1$ contention slots is already greater than the number of data packets in the frame, holds true, then the final value is adjusted: $j_v = j_x - 1$. Otherwise $j_v = j_x$. The number of CS in the frame is computed as $CS(j_v) = j_v * k$. Note that the formula first computes the number of contention slots needed, and then the rest is allocated to the data slots. This is also called *mini-slot priority allocation*, as opposed to *data slot priority allocation*.

When the traffic is not purely sporadic, with a message length of 1, the authors in [27] argue that the contention slot efficiency is higher than 33%, since some of the traffic will be isochronous and the bursty component of the traffic will contain a mixture of messages of different lengths. As a result, in this case a 2 CS/DS ratio replaces 3 in the above formula, for j_x.

6.3 Simulation Results. Bandwidth Allocation Schemes Comparison.

In [27], the authors suggest estimating the mean effective message length using a weighted moving average (WMA), computed at the HE as follows: $l_{i+1} = (1 - \alpha) * l_i + r * \alpha$. Here, r is the average request size computed from the requests received during the last frame, and α is the WMA coefficient, typically $\alpha = \frac{1}{16}$. This approach is also implemented in the NIST simulator. The subsequent figure plots the performance parameters for the Ternary Tree UB simulations, with 100 stations, as usual. The frame size is 36 mini-slots. One of the simulations was conducted using K. Sriram's bandwidth allocation, another implemented the simple bandwidth allocation scheme, as described in 5.2.4.1. The ranging parameter R was initially fixed at – R_c=4, and during the simulation, its value was chosen to be no smaller than the number of contention slots available for newcomers - $R = \max(R_c, CS_{newc})$.

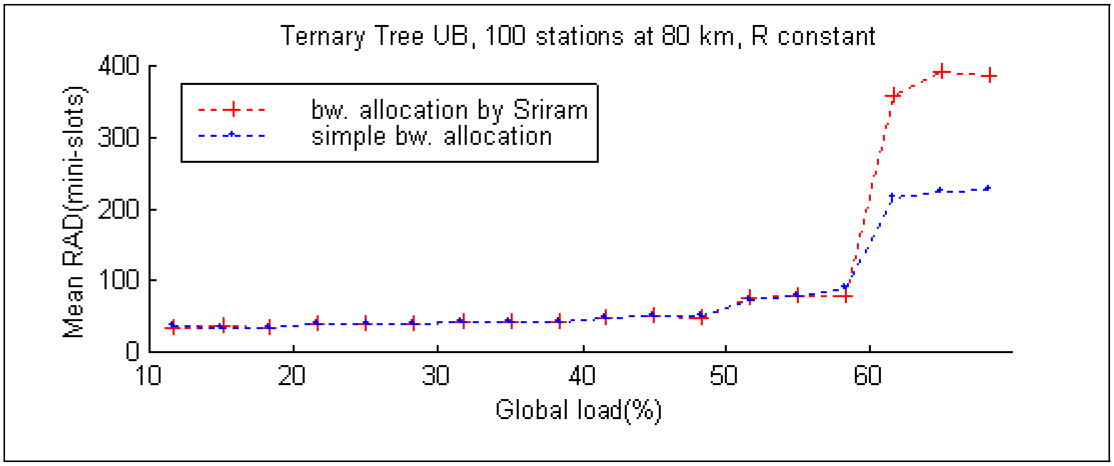

Figure 6.1

It can be seen that at high loads, the traffic-based bandwidth allocation starts to exhibit request access delays twice as high as for the simple bandwidth allocation scheme. We claim that this effect is caused by the problems in the implementation of Sriram's formula. More precisely, the effective message length WMA computed at the HE uses the size of requests, as they arrive from stations. As the global load increases, collisions will occur more frequently, and as result, stations will spend more time retransmitting their requests until they achieve success. The probability of receiving new requests during this waiting period becomes significant at high loads, and these new requests can and will be added to the retransmitted request. This will lead to an increase in the average request size computed at the HE. This, in turn, according to the traffic-based algorithm description given in 6.2, will decrease the number of contention slots allocated by the traffic-based formula, as ℓ_e increases. The following frames will receive less contention bandwidth, so the number of collisions will grow. As a consequence, stations will retransmit more and will request to send even more packets in their rare successful requests. This is a "chain effect", that will increase the RAD until the minimal number of CS per frame, at the HE end, or the maximal number of cells per request at the station's end, is reached.

Note that if the ranging parameter R were variable, it would grow appropriately with the load. The result would be that instead of transmitting their requests repeatedly, the stations would wait longer until they were admitted to try (recall that the admission time boundary depends on R). Again, during this waiting period there is a high probability that new data packets will arrive, so the transmitted request size will grow. In other words, the chain effect still has place with a variable R, simply the waiting period is incurred by the decreasing admission window and not by repeated retransmissions.

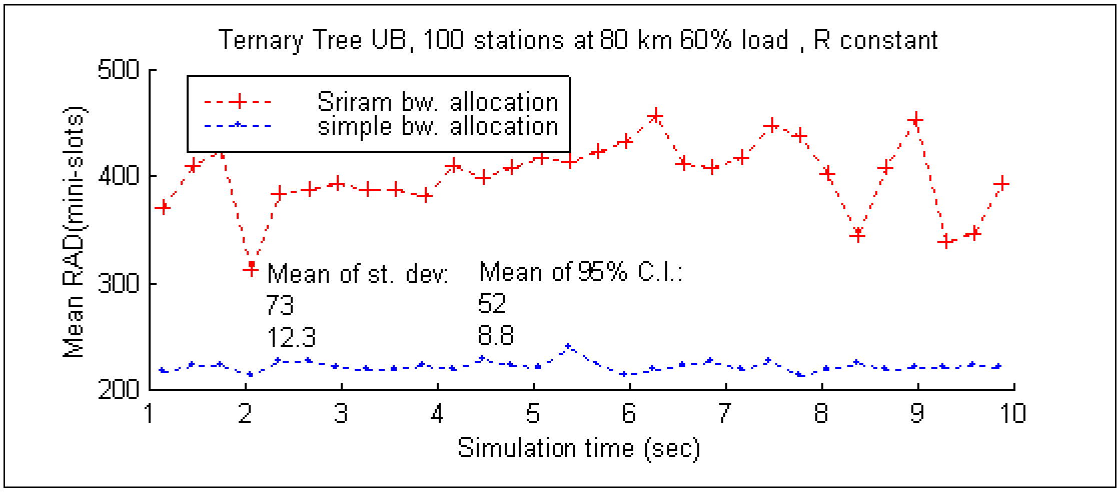

Figure 6.2

A number of simulations with similar parameters, at a 60% load, were conducted, in order to validate the above results. The following graphs each represent a mean of 20 simulations. The simulation length is 10 sec, with a warm-up time of 1 sec.

Figure 6.2 shows average graphs for the request access delay at a 60% load. We show here that the results are statistically different. The 95% Confidence Interval is 52 MS for Sriram's bandwidth allocation simulation results and 9 MS for the simple bandwidth allocation results (which means that the schemes are statistically different). Different values of α (the weight coefficient in the WMA formula) were tried with the traffic-based allocation formula, in order to improve the algorithm's time of reaction to the changes in request sizes. In general, when $\alpha = 0.0625$, the WMA will take 16 rounds to adjust to the request size *r*, and when α is 0.4 it will take only 2.5 rounds. However, as can be seen from the upper picture on Figure 6.3, a high α value did not significantly help improve the request access delay. As the number of Cable Modems attached to the head-end grows, the negative effect, first mentioned in Figure 6.1, rapidly increases. This is shown on the graphs at the bottom picture of Figure 6.3. When the number of stations was changed from 100 to 200, the average request access delay increased from 400 to over 800 mini-slots. For the simple bandwidth allocation scheme with 200 stations, the request access delay grew only by about 100 mini-slots, compared to the 100 stations case. Last are the results for the case of the variable ranging parameter, in which the other simulation parameters are same as in Figure 6.1. On the upper part of Figure 6.4, we see that varying R according to the pseudo-Bayesian estimator effectively reduces the difference between the two bandwidth allocation policies. Surprisingly, if we change the frame size from 36 to 35 mini-slots, the traffic-based scheme shows a significant performance fall off, unlike the simple scheme. If we recall the formula for j_v, the reasons for this sensitivity to frame size become clear: the contention bandwidth allocation is counted in the formula in rather large units of k – the number of mini-slots per data slot. Here $k = 4$. There is a big difference, in terms of influence on performance, between the allocation of 8 contention slots and of 12 contention slots per frame, while the difference for j_v is only 1. Moreover when the frame size does not divide by k without remainder, which is true for a 35 MS frame, the N parameter automatically becomes smaller, e.g. for a 36 MS frame, $N = 9$, but for a 35 MS frame, $N = 8$, and the real difference between the two frame sizes is 1 mini-slot. The smaller value of N inappropriately diminishes the average value of j_v, i.e. the mean number of contention slots available per frame.

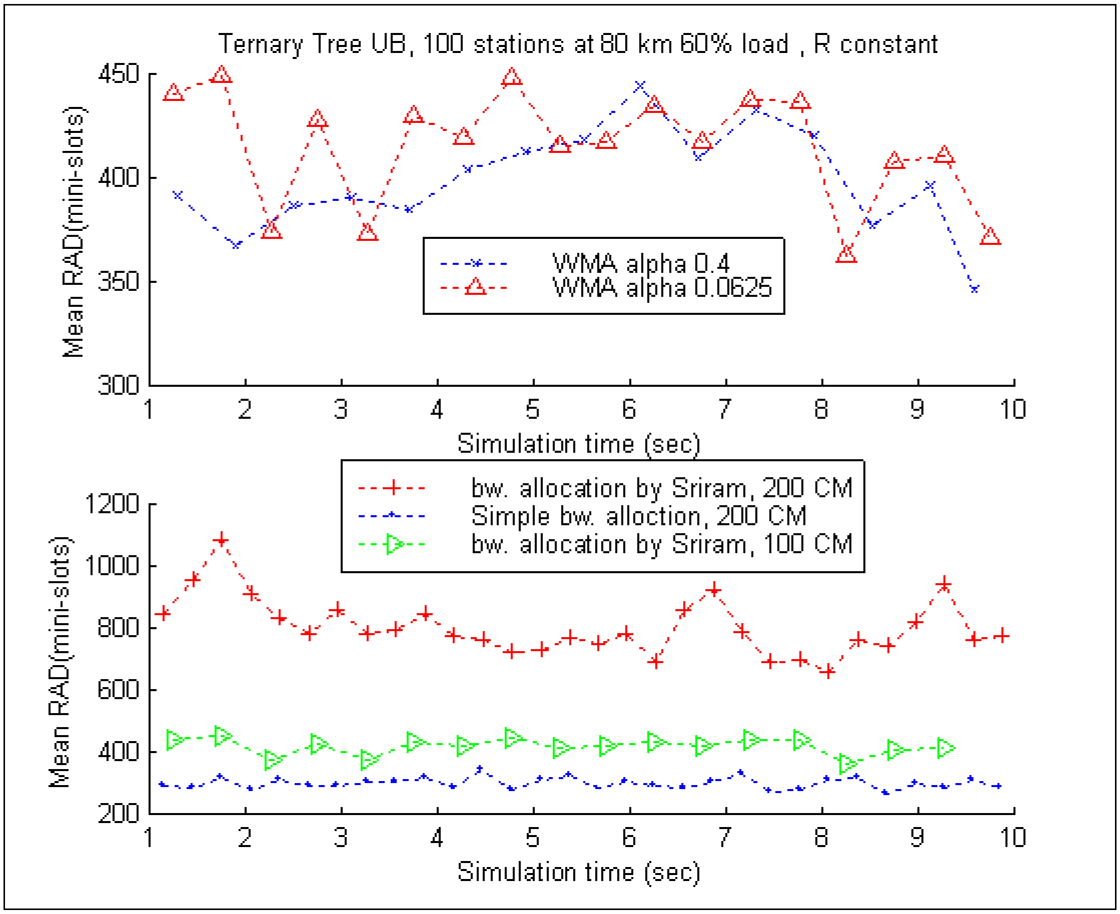
Ternary Tree UB, 100 stations at 80 km 60% load , R constant
Mean RAD(mini-slots)
450
400
350
300
1
2
3
4
5
6
7
8
9
10
Simulation time (sec)
WMA alpha 0.4
WMA alpha 0.0625
bw. allocation by Sriram, 200 CM
Simple bw. alloction, 200 CM
bw. allocation by Sriram, 100 CM
Mean RAD(mini-slots)
1200
1000
800
600
400
200
Simulation time (sec)

Figure 6.3

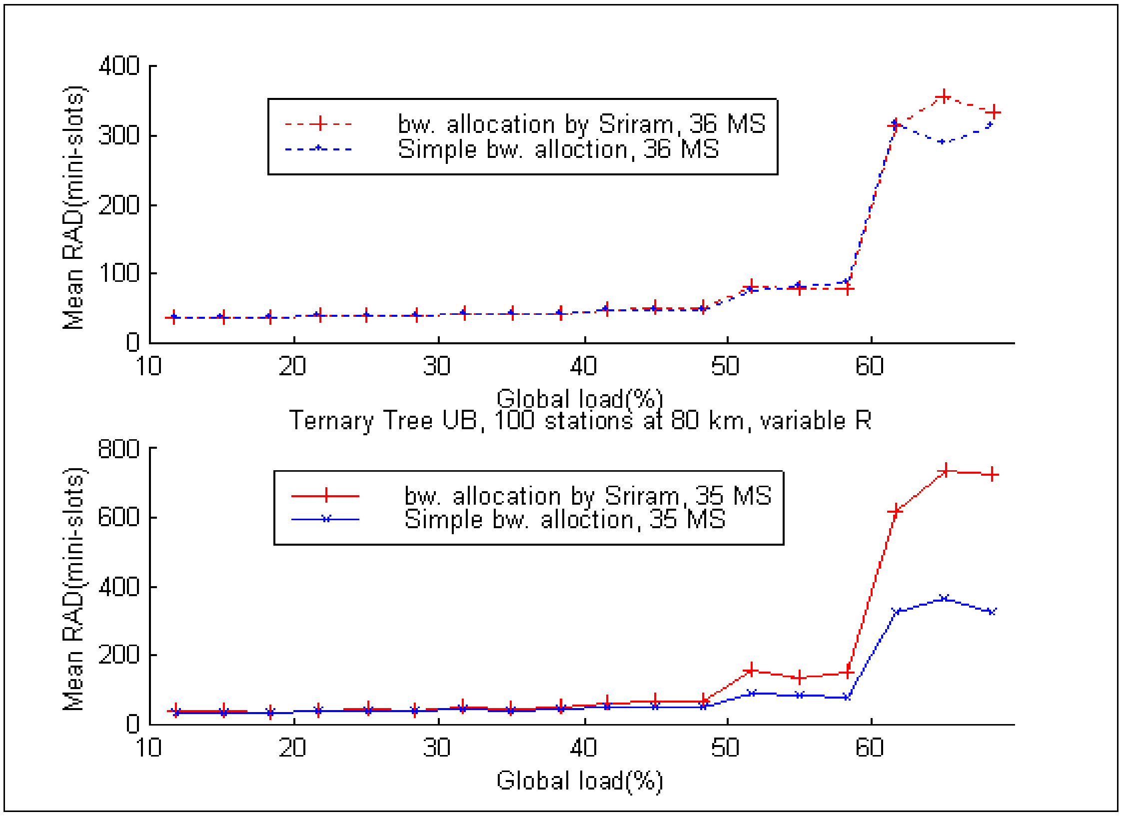

Figure 6.4

6.4 Conclusions.

This chapter included a description of the variable bandwidth allocation method, which is based on traffic estimation, and a comparison of this method with the simple bandwidth allocation scheme, primarily based on the number of collisions. Both schemes improved the performance of the ternary tree CRA, as expected. However, some problems arose regarding the traffic-based implementation. Specifically, it exhibited a poorer performance at high loads, as a result of the proportional dependence between the applied global load and the average message size, as estimated at the head-end. In order to have an exact estimate of the message size at the HE, some additional information has to be included in the station's request. This does not agree with current standards (neither DOCSIS nor IEEE). Another problem is the relatively large allocation units, and consequentially, the excessive sensitivity of the scheme to the protocol parameters. We conclude that the simple collision-based scheme gives an equal or better performance, in terms of access delay and scalability, while having simpler ideology and implementation than the traffic-based scheme, for the class of splitting tree algorithms.

7 Algorithms' Behavior at a Fixed Load.

7.1 Motivation.

In the previous chapters we investigated different algorithm parameters and showed their influence on performance under a dynamically changing global load. A gradual increase of the offered load on the network could have its own (possibly different) influence on the system's behavior. In the next chapter we wish to keep the load fixed, and show the performance of the contention resolution algorithms under a constant load, and the difference, if any, that is caused by changing other HFC topology and MAC protocol parameters, with a fixed load.

7.2 Simulation Results.

7.2.1 Request Access Delay Changes at a Fixed Load.

The simulations described in this chapter were somewhat shorter – 7 sec with a 1 sec warm-up time, since the load remained constant, and this time suffices to demonstrate tendencies in the system. The first figure – Figure 7.1 - demonstrates the influence of changes in the ranging parameter on the request access delay. Actually, the request access delay follows the oscillations of the ranging parameter very closely. The changes in *R* occur around some optimal value for the given load, the number of stations and the CRA. The changes can be explained as follows:
Let us observe the lowest point that the RAD graph reaches at the beginning of some amplitude. Here, *R* is small, and many newcomers are admitted to contend over sending their requests. Successful requests have a short delay, as they have to wait less time before trying, but on the other hand, the number of collisions increases, as there are more contenders (at medium and high loads) than available contention slots. Consequentially, R grows, and the RAD increases with it – so stations will wait more before sending their requests. This process will continue until the RAD reaches the peak point on the graph. Here, the ranging parameter is big enough, so that very few stations with new requests are admitted to try at all. In other words, at this point the probability of an idle or successful contention slot will become higher than $1/e$ $(n_i+n_s>CS/e+n_{col}/e-2)$, so *R* will start to decrease.

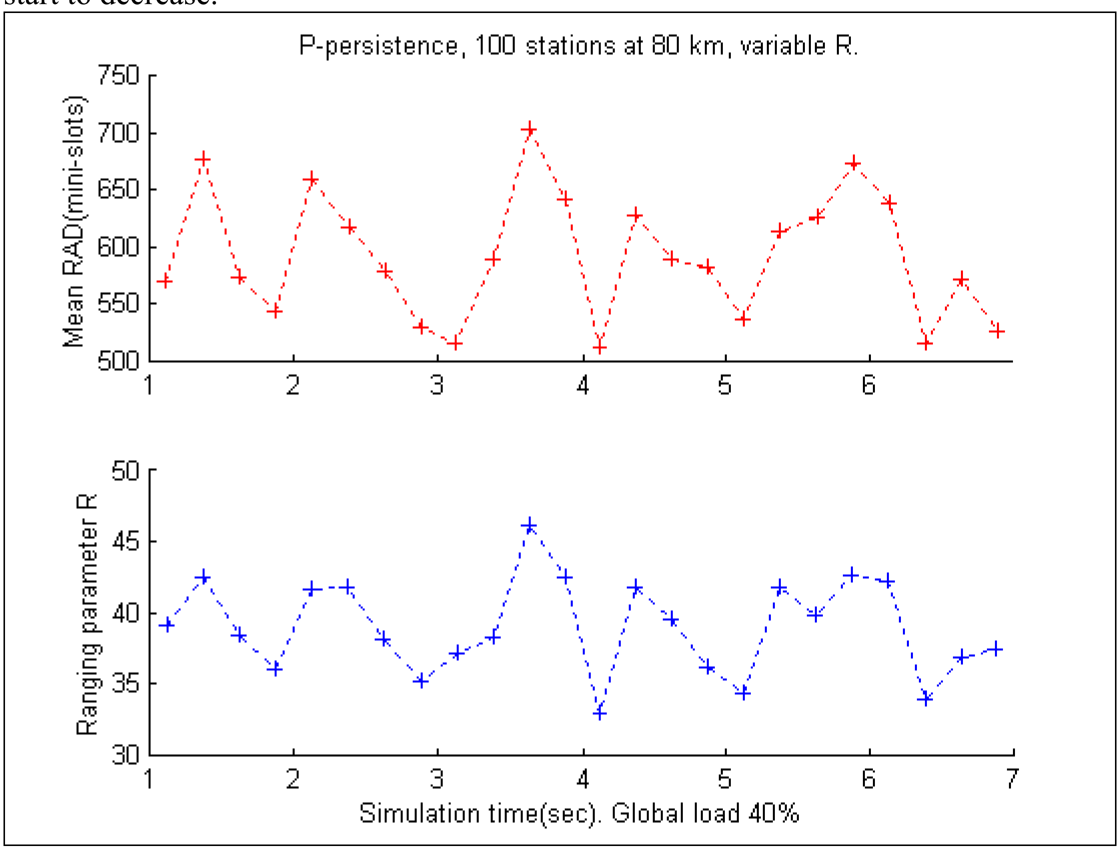

Figure 7.1

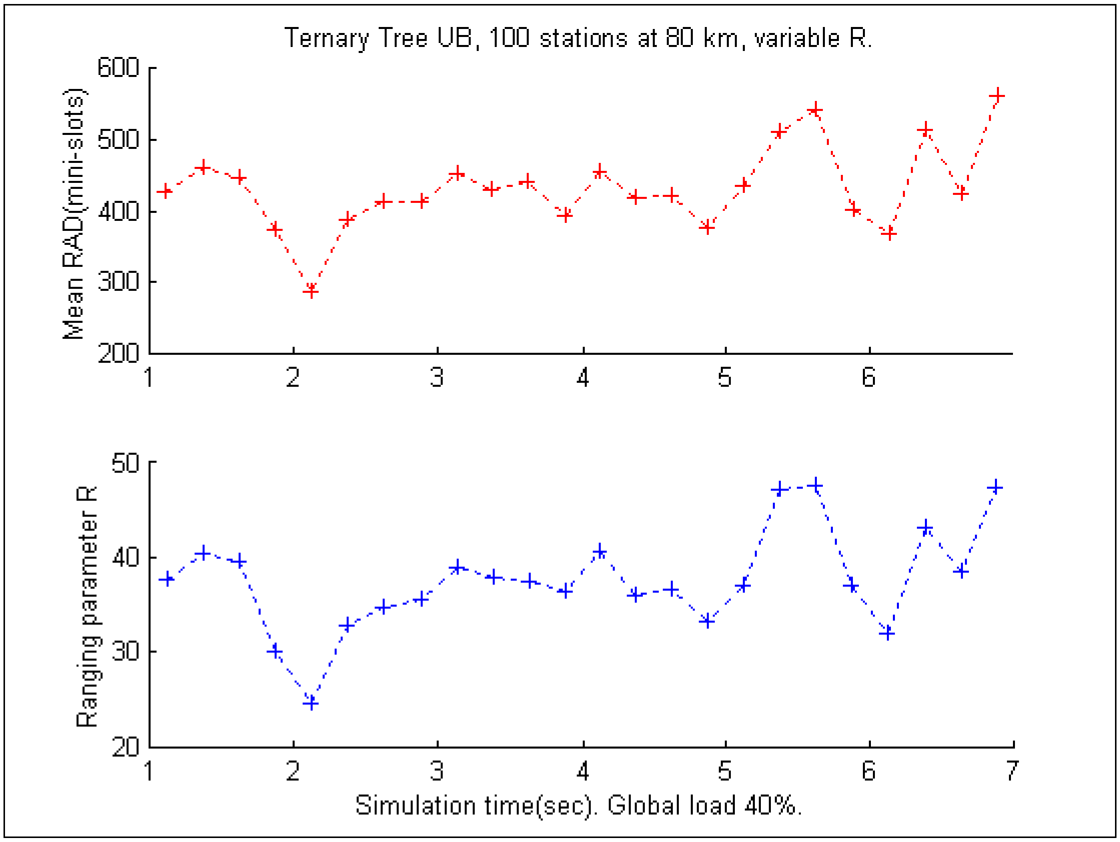

Figure 7.2

The reverse process takes place, until the probability to send the request becomes higher than the probability to refrain from sending, at which point the newcomers and backlogged stations will start trying to send, more collisions will occur than idle/successes, R will increase, and the process will repeat itself. The same is true, as seen from Figure 7.2 for the splitting-tree algorithm, because although it has an admission time boundary instead of a sending probability, this boundary is still based on the ranging parameter value.

7.2.2 Changes in RAD Jitter with Varied Distances.

Based on these primary observations, we further show the results of simulations conducted with a fixed global load and varied distances to the HE. The simulations use a 1 CS/DS ratio for both clustered and continuous modes, unless stated otherwise. Each graph represents a mean of 10 independent simulation runs with given parameters. First, we consider the p-persistence CRA. We can observe from Figure 7.3 that the amplitude of the oscillations becomes smaller with distance. The jitter numbers for each graph shown on the figure prove this. The jitter here is computed as a standard deviation for each graph, in mini-slot units. The first two graphs are for clustered mode p-persistence, and for a much shorter 10 km distance we show the continuos mode p-persistence graph, as we saw previously that it performs better than clustered mode at a 10 km distance. These results proved to be stable over a substantially larger number of separate simulations, so the graph merely presents the trend in visual form. In the above simulations, the frame size for clustered mode was changed along with the distance, so as not to create unnecessary delays at short distances with frames that are too long. The jitter trend results would also remain intact, if we left the frame size fixed, as can be concluded from Figure 7.4. Here, the frame size used is identical for all different distances – 35 mini-slots. The simulation for the 10 km distance was also conducted here in clustered mode. The jitter decreases slightly more with distance than in the previous case, at the expense of a higher average request access delay.

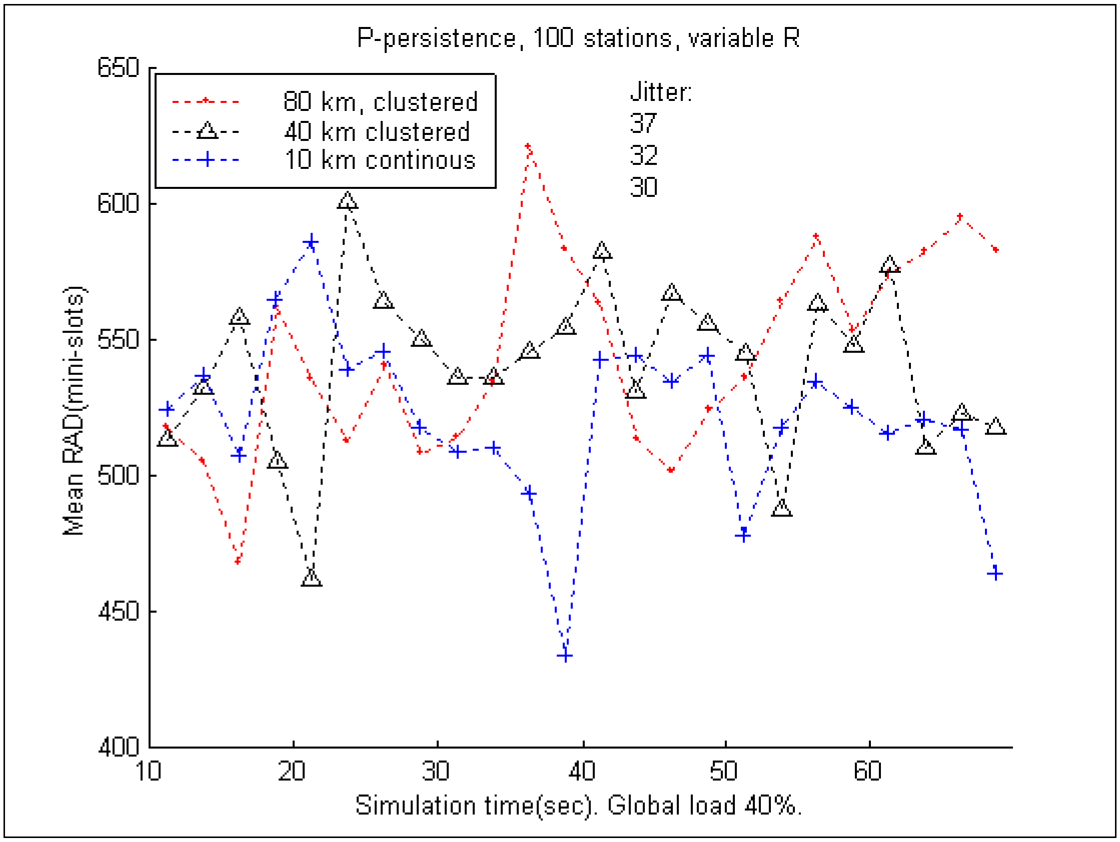

Figure 7.3

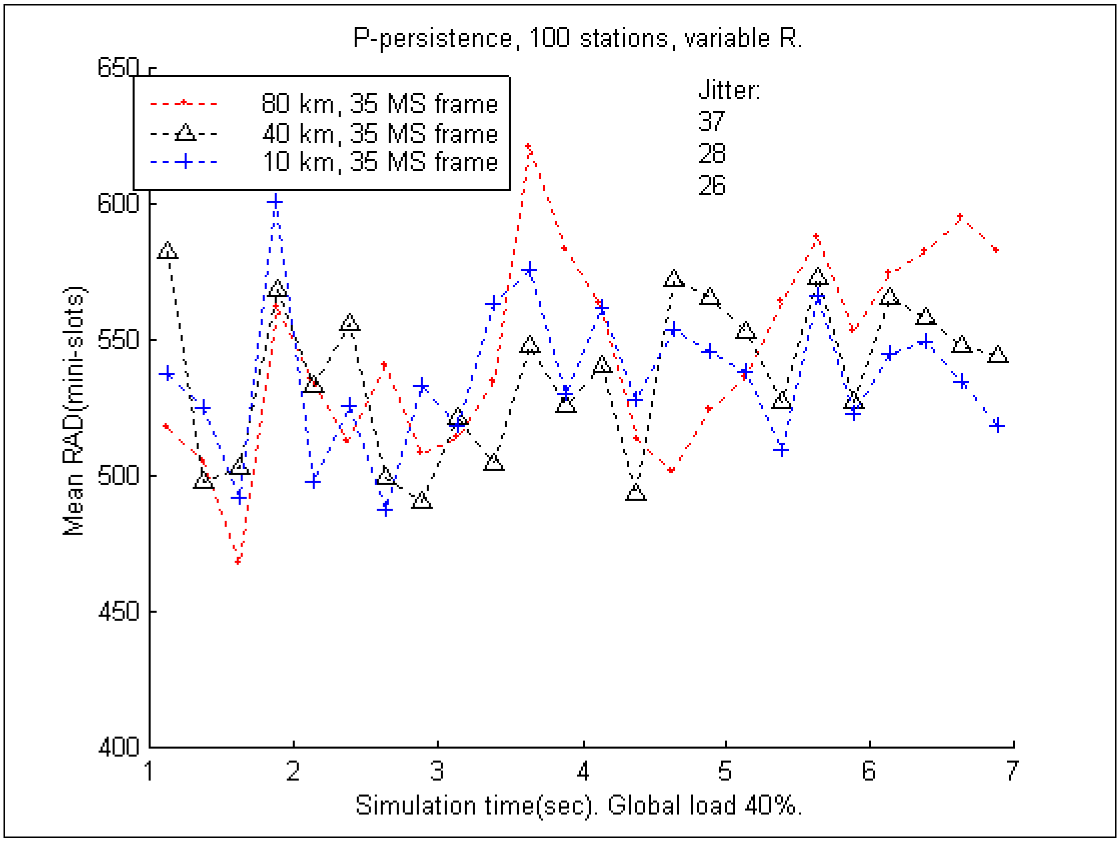

Figure 7.4

There is no such clear pattern for the Ternary Tree Unblocking algorithm. Figure7.5 shows graphs for the case of a 35 mini-slot frame size. The mean RAD is almost identical for all three cases, although it is about 100 MS lower than for the p-persistence case. In terms of jitter, there is no clear advantage (which here means lower jitter) to the case of the 10 km distance to the HE, over the case of the 80 km distance. Other results, with a variable frame size, behave similarly, so we do not present their figures here. Additional simulations that were conducted showed that if the simulation sequences are identical, i.e. if the stations produce messages at the same time and in the same order during the compared runs, and the distance is changed as before, the RAD jitter clearly decreases with distance, for both the p-persistence and ternary tree algorithms, and for all the cases of constant/variable frame size, constant/variable R, etc. The comparison, then, is made only between these single identical sequence runs. This shows that under the exact same conditions, the RAD jitter decreases with distance, but the p-persistence CRA accumulates a tendency of decreasing jitter with distance when the simulation sequences are different, while in the case of ternary tree, different runs cause the tendency to diminish, on average.

This can be understood if we recall that the p-persistence algorithm is purely random in nature, and each run, and moreover each decision made by a station during the run, depends solely on the sending probability *p,* and is not further controlled by the HE (i.e. it is independent of other stations). The sending probability is, in turn, the reverse of the ranging parameter, which mostly depends on the load and number of stations. It is therefore expected that the effects caused by this major parameter accumulate. For the IEEE, the ternary tree algorithm is more centralized, and the parameters through which the centralized control is exhibited or influenced, such as the ATB that orders the newcomers in the FIFO manner, the number of RQ groups that can be accommodated within the frame structure and the number of mini-slots left by the algorithm for the newcomers, have a conclusive significance on the algorithm's performance.

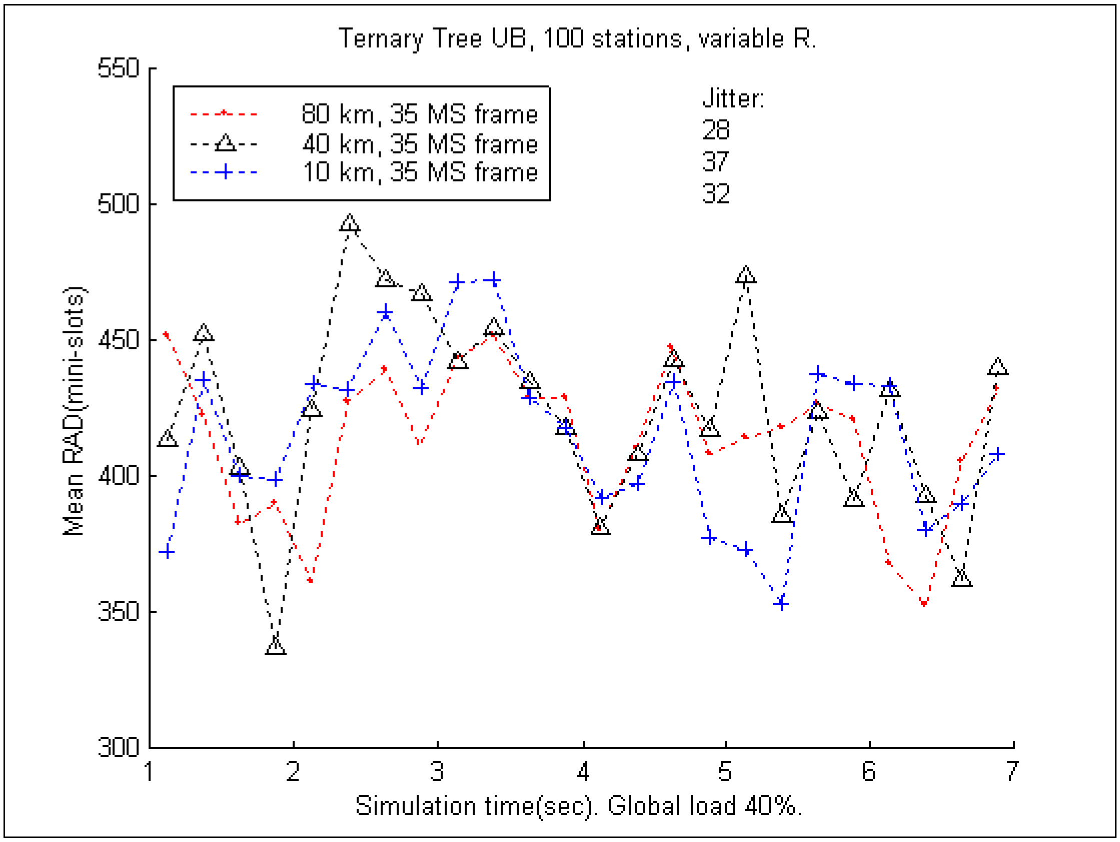

Figure 7.5

7.3 Conclusions.

With a fixed global load, the request access delay oscillates around some average value. The amplitude of the oscillation, or the jitter, in each run, depends on the distance from the HE. The dependence is stronger for the p-persistence class of algorithms, while for the ternary tree class it is overshadowed by the centralized HE control.
The jitter parameter is of particular importance for applications that demand a constant bit-rate/low constant delay, such as voice telephony. The above results suggest that an architecture that brings the head-end closer can help solve the problem of jitter reduction, especially if the CRA used belongs to the p-persistence family.

8 Dynamic Frame Size Allocation.

8.1 Motivation.

Having seen how CRA's behave with a fixed global load, and having concluded that the ternary tree algorithm jitter delay is less affected by "external" parameters, such as the HFC network topology, we wish to examine one of their inner parameters - the frame size. The questions of interest are : does the frame size have any effect on the request access delay? If the answer to the first question is positive - how can we find some way to predict or, better still, dynamically define the optimal frame size, i.e. the one that helps us minimize the RAD?

8.2 Simulation Results.

The following simulations were conducted with the Ternary Tree Unblocking algorithm, using 100 stations. Each run length was 10 sec with 1 sec warm-up time, unless stated otherwise and the ranging parameter is variable.

8.2.1 RAD differences for different frame sizes.

Firstly, several simulations were conducted, in order to measure the dependence of the request access delay on frame size for the Ternary Tree UB algorithm. The offered global load and the distance to the HE are, in general, different for each figure, but are constant throughout each simulation. Once more, each graph represents the mean of the 10 simulations, which differ only in simulation sequence.

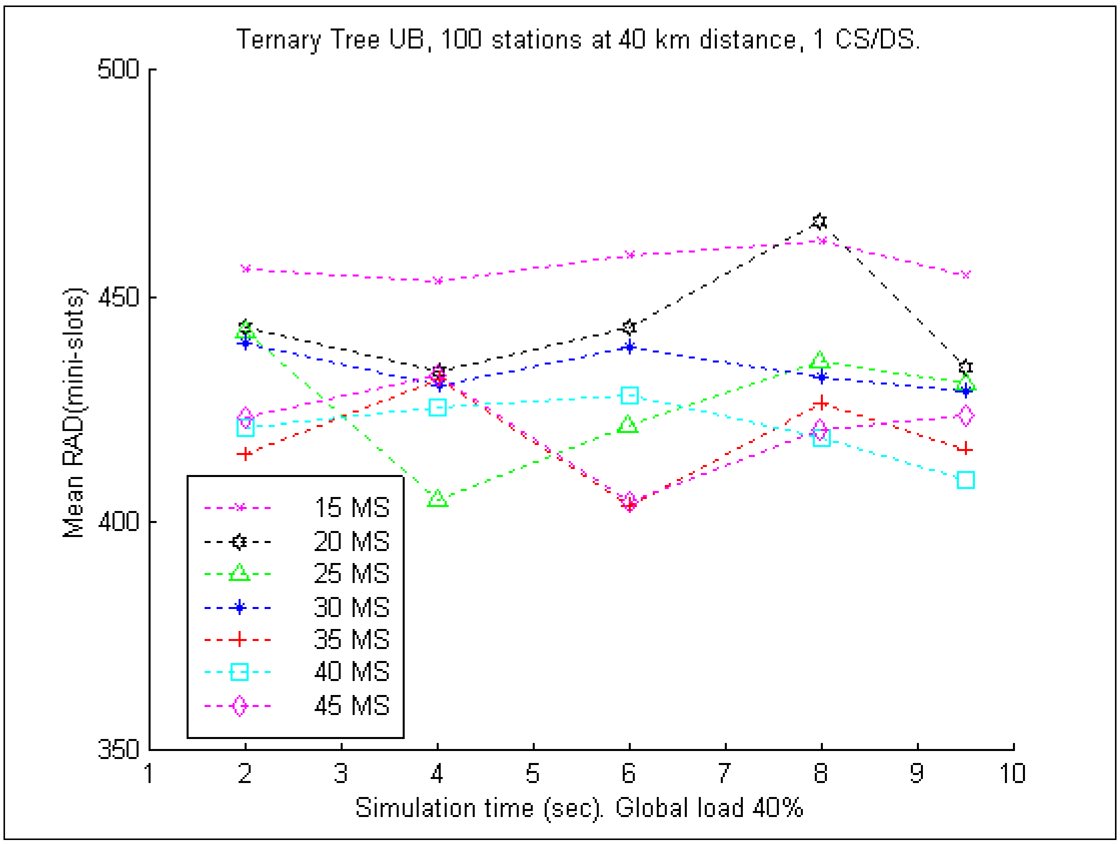

Figure 8.1

We find a tendency of RAD decrease, while the frame size increases in Figure 8.1 and Figure 8.2, i.e. at high offered global loads. Conversely, when the load is light (20%), the best results are achieved with the smallest frame size, of 15 mini-slots, and the RAD clearly increases with the larger frame sizes.

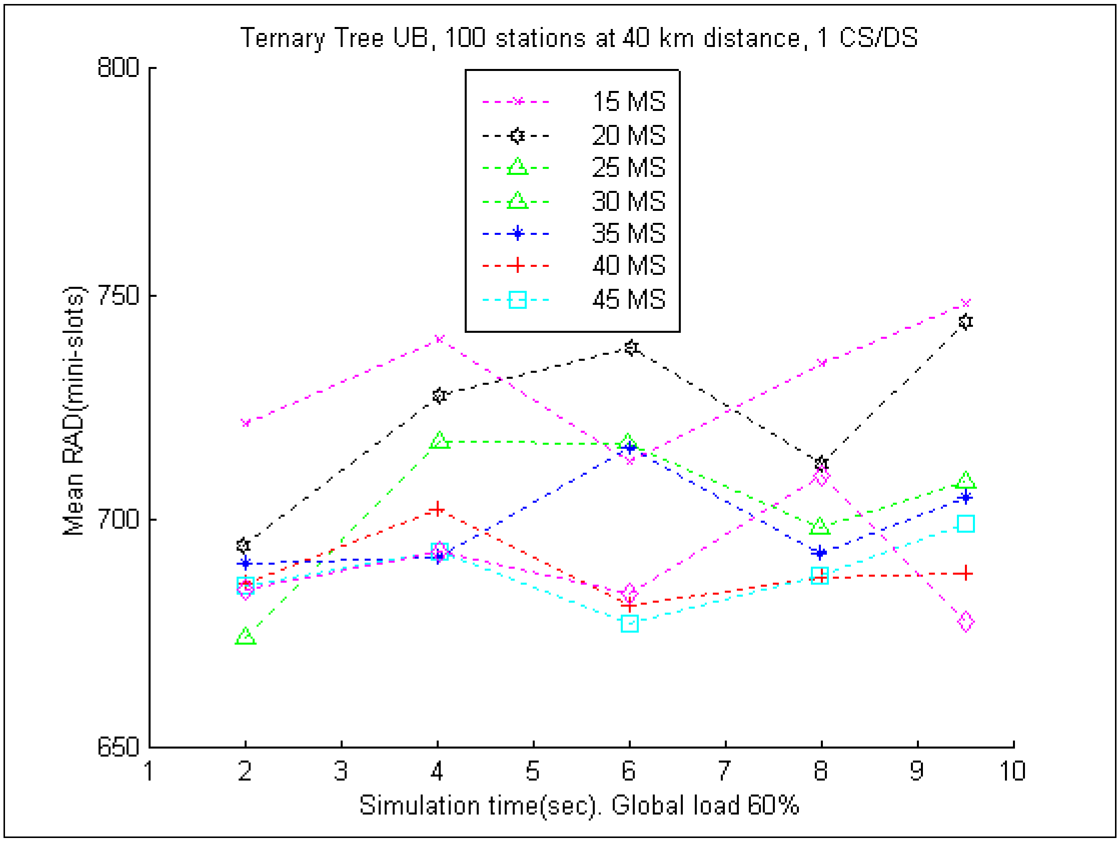

Figure 8.2

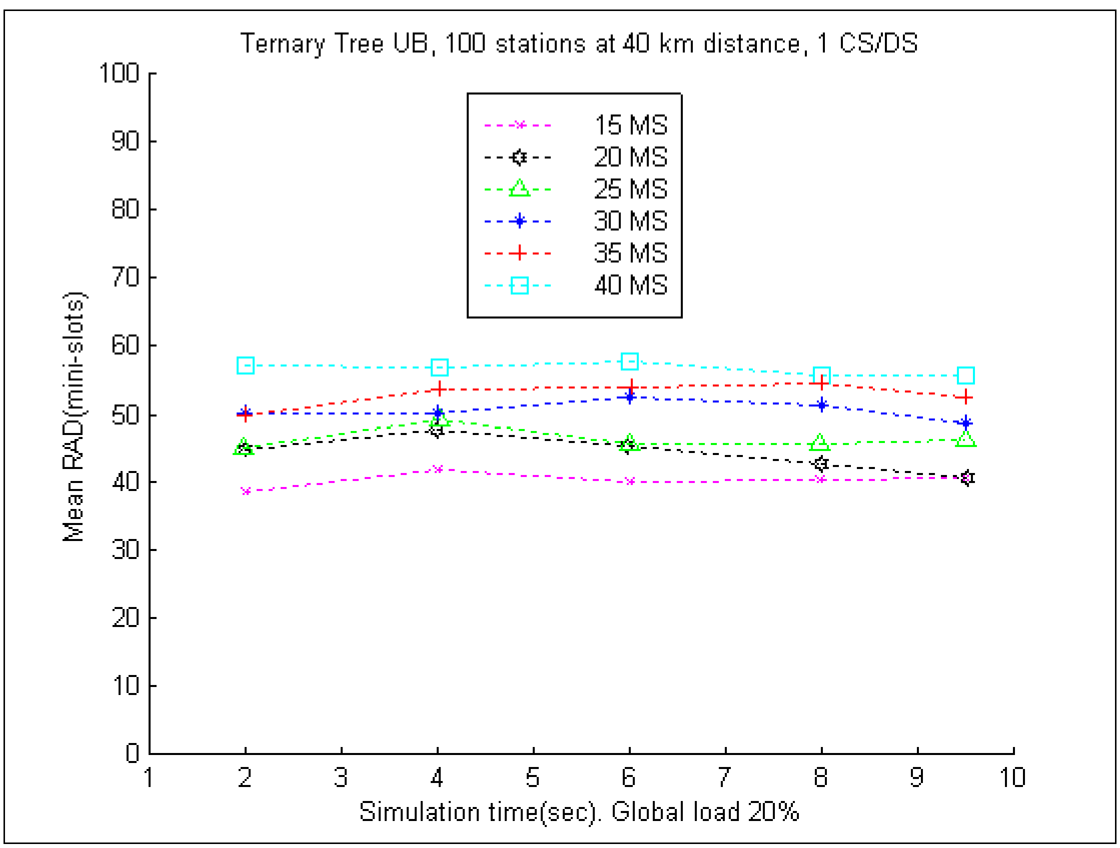

Figure 8.3

8.2.2 Dynamics of RAD changes with frame size.

The above results suggest that the request access delay does depend on the frame size used by the algorithm, and there is actually a frame size optimum for a given network configuration. It appears that the optimum depends on the offered load: the frames smaller than the optimal size are "too small" and introduce unnecessary collisions and newcomer delays, and on the other hand, the frames that are "too big" waste mini-slots and introduce unnecessary delays for newcomers arriving in the middle of a frame. If the above theory is correct, we can expect to see a graph of RAD dependency on frame size that looks like parabola. And indeed, in Figure 8.4, which summarizes the results for the above and some additional simulations, we can clearly see a parabolic curve on the 30% load graph. This is a "normal" global load for the system.

As the system starts to become overloaded at a 40% global load, we can see that the curve has a bigger difference between the maximal and minimal RAD values, in the part of the graph where the RAD drops while the frame size increases. On the contrary, the tail end of the graph, the part after the minimal RAD value was reached, is much flatter. As we move to the 60% load figure, we see that the tail becomes almost a straight line, while the max-min difference grows slightly more. As for the 20% load picture, we can actually only see the tail end of the parabola. The minimum RAD is reached when the frame size is 15 mini-slots. We can not further decrease the frame size here below 15 MS, as this would create frames that are shorter than the round-trip delay time. Also note that the frame size value, near which the optimum is reached, increases with the load. As the load increases, more collisions will occur, therefore increasing the frame size at high loads improves the RAD more significantly. On the other hand, at a high load it becomes more "difficult" to reach the situation in which there are many idle contention slots open for newcomers, and the frame itself becomes too big. We see that with the 60% load, even a frame of 200 mini-slots is not sufficient.

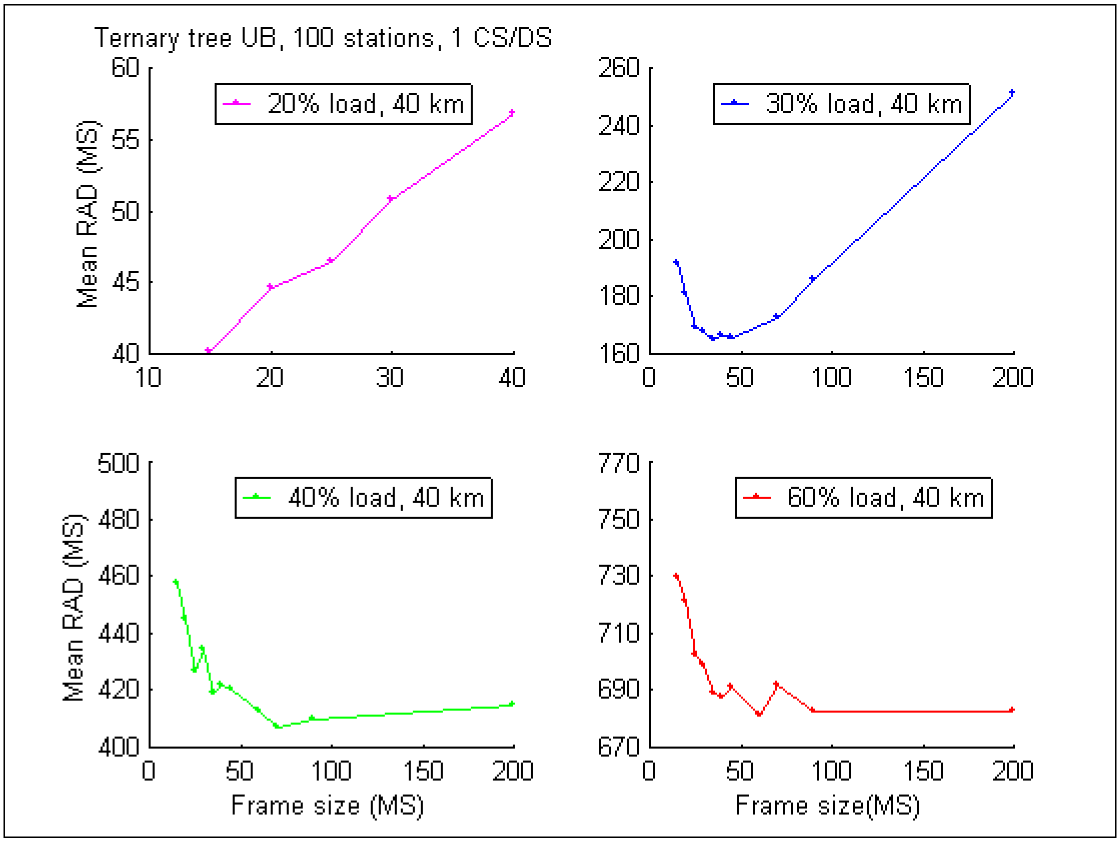

Figure 8.4

If we double the distance to the head-end, from 40 km to 80 km, we will receive the very same picture, as seen in Figure 8.5. The are only two differences: one, that the RAD is higher at a low load of 20%, as a result of the increased RTD time, which is the major contributor to the RAD when the number of collisions is minimal; and two, because we cannot use the small frame size of 15 MS for the 80 km distance, the first frame size used in simulations was 25 MS, so the left side of the 30,40, and 60 % load graphs looks lower than the appropriate graphs in Figure 8.4. We see that the influence of changing the frame size on RAD does not depend on distance, but rather on the offered global load.

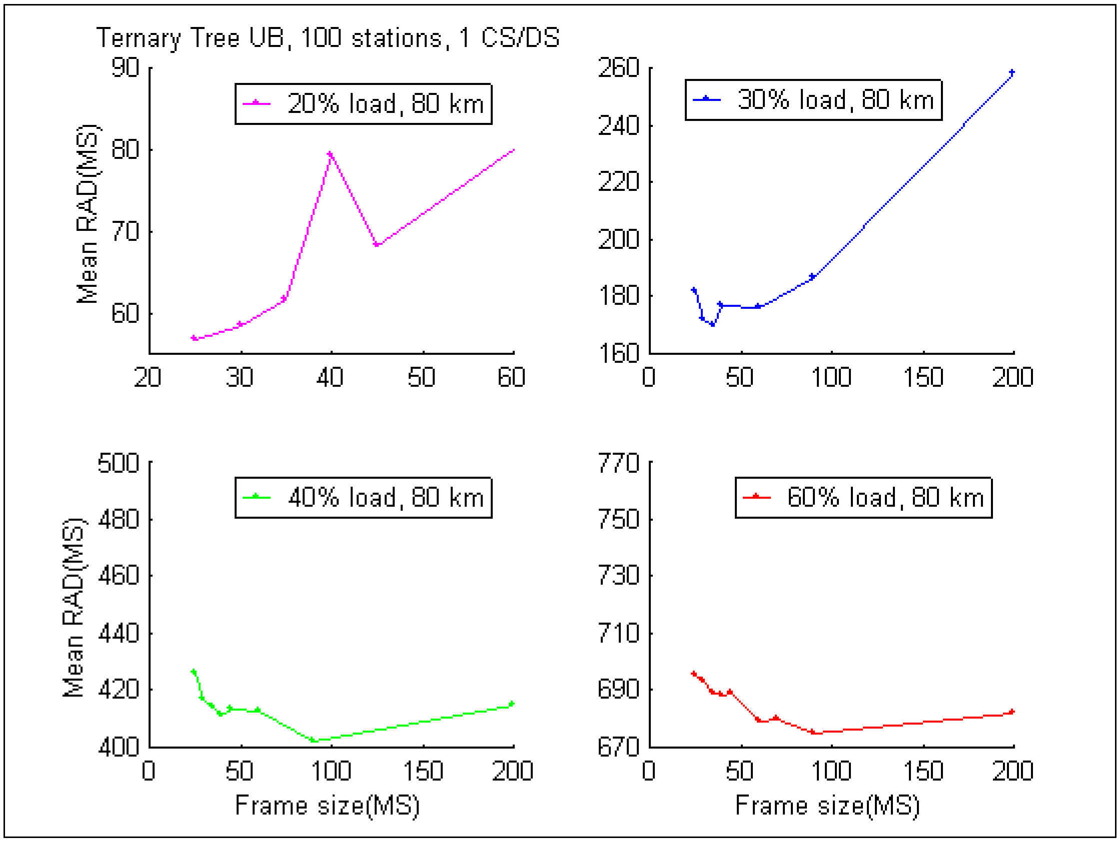

Figure 8.5

8.2.3 RAD and frames with no NMS correlation.

The request access delay comprises a number of components, which can be described as follows: $T_{delay} = t_{arrival} + t_{bound} + t_{CRA} + t_{RTD}$. Here, $t_{arrival}$ is the time from the cell's arrival to the station until the station "sees" a contention slot in which it can try to send a request for the cell. On average, the cells arrive in the middle of the frames, so this time should be equal to half the frame duration. t_{bound} is the time the station will wait until the cell's arrival time becomes greater than, or equal to, the Admission Time Boundary defined by the head-end. t_{CRA} is the delay imposed by the collision resolution, once a collision has occurred. t_{RTD} is a constant delay component, which equals the time needed to transmit the request, plus the processing and feedback transmission time at the HE, plus the delay time for the request to travel to the HE and back to the station. The computation of the ATB made in the IEEE 802.14 algorithm implementation is represented by the following formula: $T_{bound} = T_{prev} + \frac{NMS}{R} * \Delta T$ - assuming that we are using one interleave (which we are). Here, T_{bound} is the bounding time, T_{prev} is the bounding time value at the end of the previous frame, R is the ranging parameter, NMS is the number of mini-slots assigned for newcomers in the next frame, and $\Delta T = T_{cs} - T_{prev}$, T_{cs} is the time equal to the end of the contention slots' region in the following frame. The time boundary is computed by the HE at the end of each frame and is sent to all the stations. Recalling that the newcomer mini-slots have the lowest priority, i.e. are assigned last, after all other RQ's, we can see that if many collisions occur, and all the contention mini-slots are used for $RQ > 0$ - collision resolution, the NMS will equal zero and the T_{bound} will remain unchanged, in such a frame. When the collisions will be resolved and the subsequent frame will have a positive number of newcomer contention slots, which may happen after a number of such frames that had $NMS = 0$ and did not admit newcomers, the T_{prev} will still be equal to the T_{bound} of the first frame, that had no NMS. In other words, the ΔT becomes rather big, which is not desirable, as many newcomers will be admitted due to this, leading to many new collisions. A strong correlation was found between the request access delay and the number of frames that had $NMS = 0$ during the simulation. Figure 8.6 demonstrates this. The percentage of frames with no contention slots (zero NMS) for newcomers, relatively to the total number of frames during the simulation, is plotted on the y-axis.

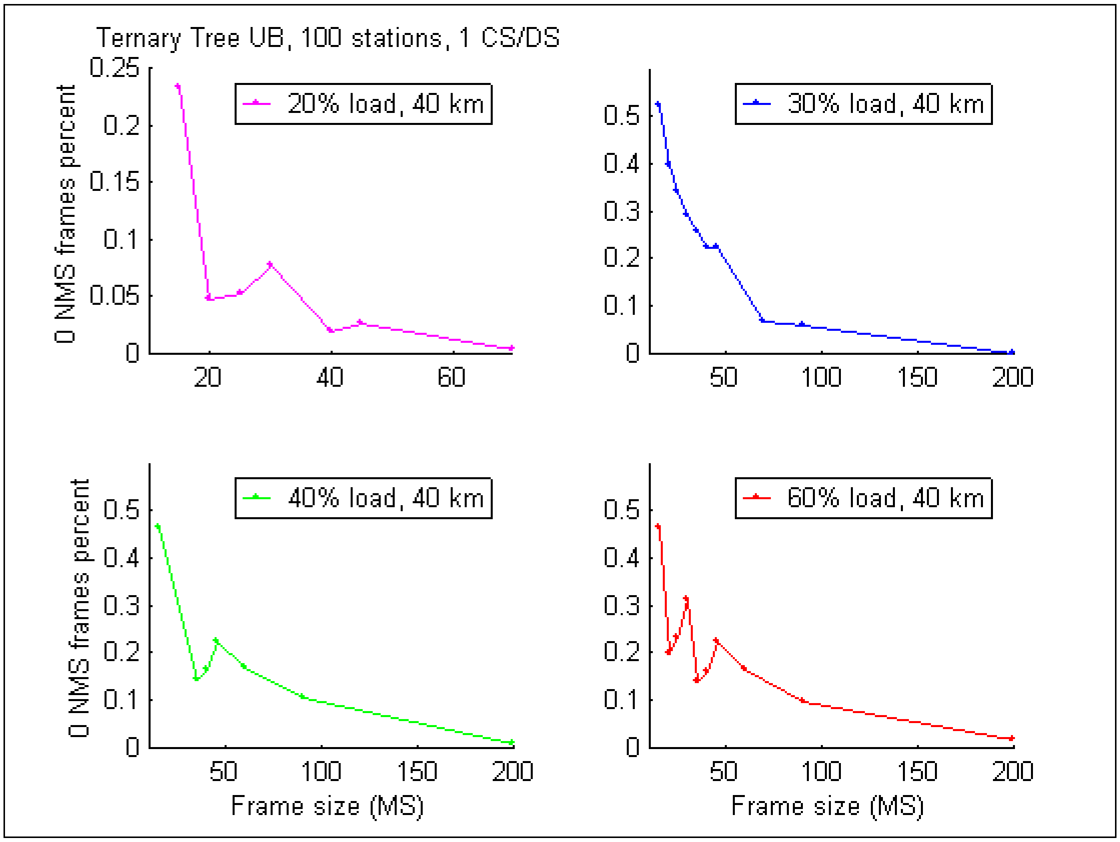

Figure 8.6

From a comparison between Figure 8.6 and Figure 8.4 we can see that the optimum, in terms of request access delay, is near 10%, for the number of frames with no newcomer contention slots. When the percentage of frames with no NMS is greater, the effect of first delaying newcomers and then having a relatively big admission window, as described above, has a negative influence on the mean RAD. If the amount of such frames drops well below 10% of the total amount of frames, this indicates that the frame is too big, and has too many idle contention slots, on average. 10% also proved to be the optimal no-NMS frames value for the 80 km distance in the simulations conducted.

8.2.4 Dynamic frame size adjustment.

Now that the changes in the system's behavior, which result from changes in the frame size, are well understood, we would like to build into the HE a mechanism that will alter the frame size dynamically, in order to improve the system's performance. First, we use the no-NMS frames limit of 10% to dynamically estimate an optimal frame size at the HE. A weighted moving average of no-NMS frames percentage during a window of 50 frames was computed. While the current WMA value was significantly greater than the

10% limit, the HE increased the frame size. The significance of the difference between the WMA and the no-NMS frames limit was controlled by the precision parameter, which was set to 25% from the no-NMS frames limit. When the opposite happened, i.e. when the WMA became significantly smaller than the desired no-NMS frames limit, the head-end started to reduce the frame size. Figure 8.7 presents results for the case of 100 stations at a 40 km distance from the HE. The global load increased during the 20 second simulation from 10% to 60%. The first graph demonstrates the mean request access delay for ternary tree with a fixed frame size of 35 MS, which was found to give the best results for the above algorithm/configuration. The second graph was received when the HE changed the frame size according to the above description. The initial frame size was set to the smallest value - 15 MS, as we assume a small load at the system's startup time. The lower part of the figure shows a graph of the frame size changes during the simulations. We see that after some initial fluctuations, the frame size stabilizes around the optimal value for each given load. From the RAD graphs we see that the dynamic scheme manages to stay close to the minimal RAD values for each load, demonstrated in Figure 8.4. This scheme has a mean advantage of about 50 MS, in terms of RAD, over the fixed-size scheme, at medium and high loads (which cannot be seen very clearly in the figure, due to the big scale needed to represent all the RAD values).

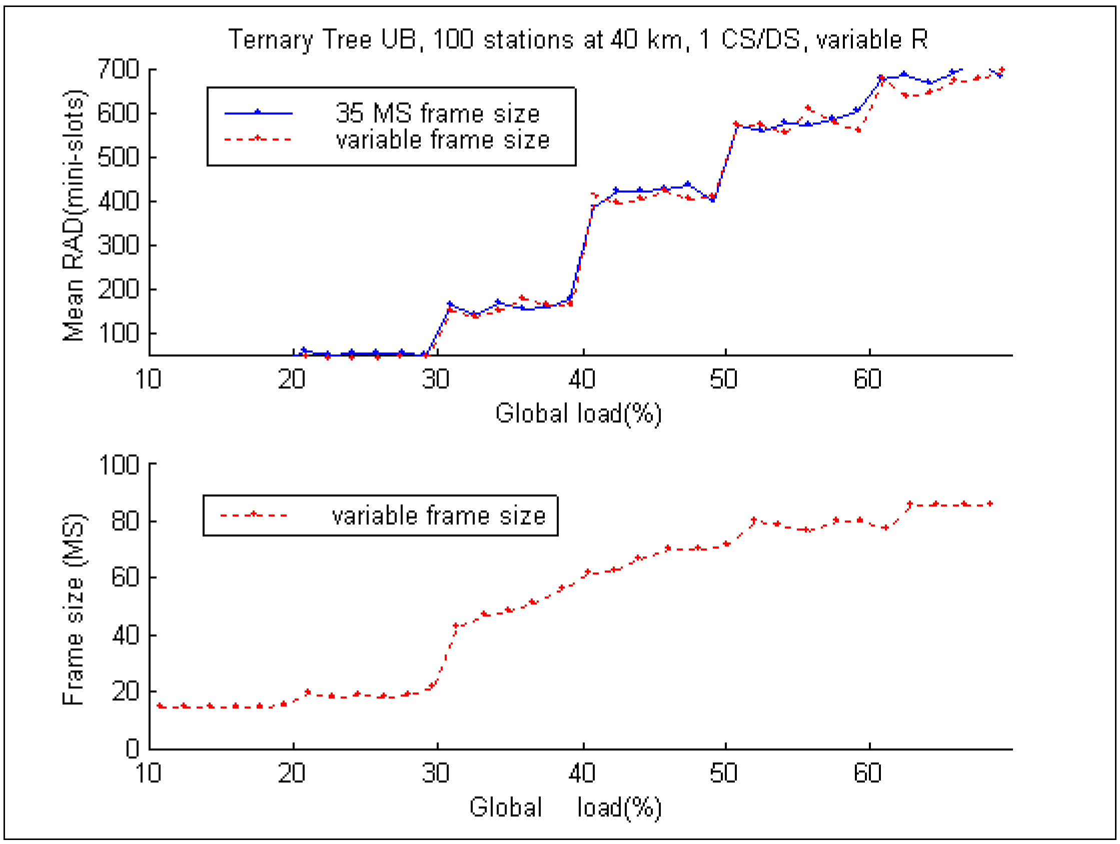

Figure 8.7

The number of no-NMS frames parameter gives the possibility to estimate an optimal frame size for the given load and a CS/DS ratio. There is additional information, available at the HE, which can be used to dynamically evaluate the network's condition – namely, the ranging parameter, which approximates the number of contenders and the weighted moving averages for the number of collided/idle contention slots. One of the advantages of using these parameters is the fact that the desired value of no-NMS frames percentage does not have to be known in advance, and it does not have to be used as the external parameter given to the HE. Here, we can simply ask the HE to maintain the frame size value so that the frame will have enough contention slots (on average) open for newcomers, on the one hand, and not too many idle slots unused by newcomers, on the other hand. The algorithm for the frame size adjustment is:

1. The HE computes WMA's for the collided, idle and total of the contention slots available for newcomers (NMS). It also should keep a value of the ranging parameter *R'* different from the one used for CRA, in that it does not use the amount of contention slots available as a minimal value.
2. The HE uses a 50 frame size window. At the end of the window, the HE will:
   a. Increase the frame size, if the WMA of collided slots exceeds the amount of contention slots in the frame or if the ranging parameter has grown too big - $R' > n_{cs} * 2$ (in which n_{cs} is the amount of contention slots).
   b. Decrease the frame size, if the WMA of idle NMS is greater than $\varepsilon * n_{nms}$ [in which ε is constant and n_{nms} is the weighted moving average of NMS available per frame].

Figure 8.8 shows simulation results for the algorithm. As before, it is compared with the ternary tree, using a constant 35 MS frame size. During the simulations we set $\varepsilon = 0.5$.

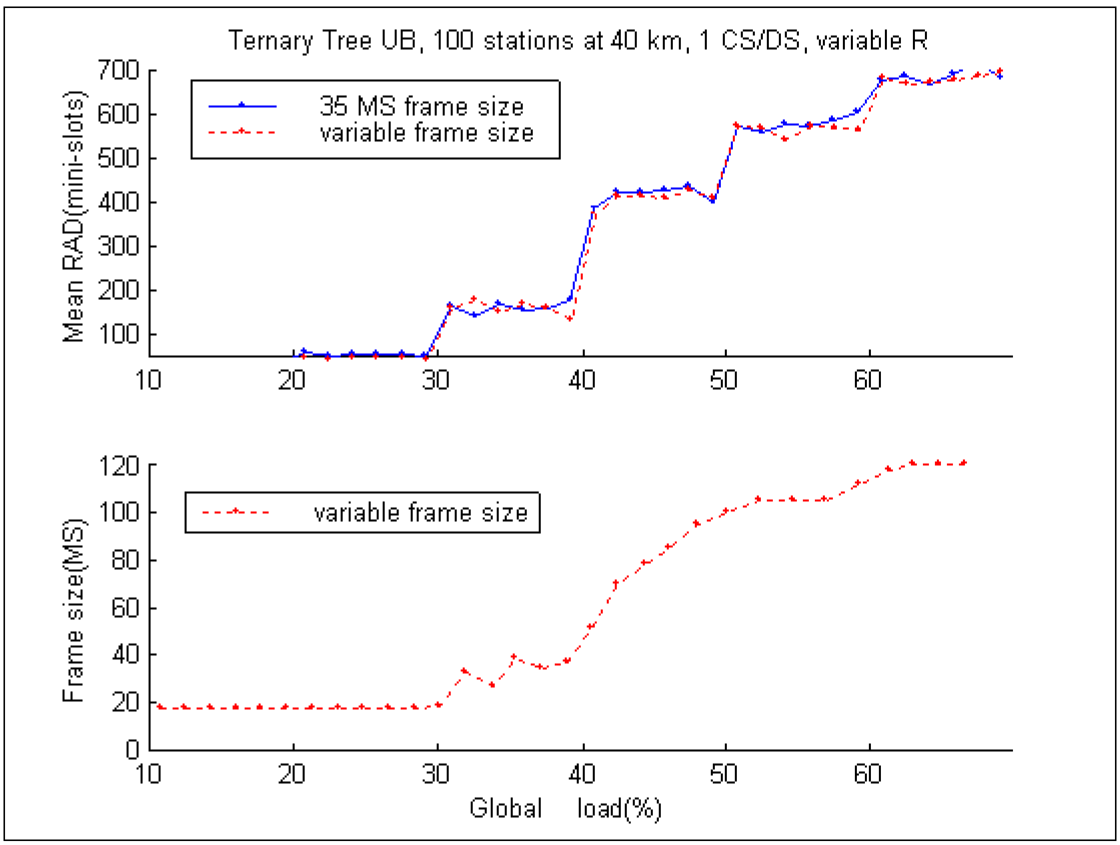

Figure 8.8

The following algorithm is an improvement over the last algorithm for dynamic frame size adjustment described in 8.2.4.
The algorithm computes weighted moving averages for the number of contention mini-slots available for newcomers in each frame, and the *R'* - the modified ranging parameter. It estimates the number of collided newcomer requests using the expectation of number of mini-slots that will contain two or more newcomer requests. The expectation is computed like following:
Let's denote *n* – the number of available contention slots, and *k* – the number of requests. The probability of having *j* requests in a certain mini-slot is $P_j = \binom{k}{j}\left(\frac{1}{n}\right)^j\left(1-\frac{1}{n}\right)^j$. Then the probability of having two or more requests collided in a certain mini-slot will be: $1 - P_0 - P_1 = 1 - \left(1-\frac{1}{n}\right)^k \frac{n-1+k}{k-1}$. An expectation *E* of the number of collided mini-slots is therefore just $E = n * \left[1 - \left(1-\frac{1}{n}\right)^k \frac{n-1+k}{k-1}\right]$[1].

After we have E computed at the HE, using newcomers' contention-slots WMA as *n* and *R'* WMA as *k*, the algorithm compares the number of expected collisions with the current frame size and performs the frame size adjustment:

- If $\lfloor E \rfloor * 3 \geq n_{cs}$ set cluster size equal to $\lfloor E \rfloor * 3 * (cs\_ds\_ratio + ms\_per\_ds)$ mini-slots.
- Else if $\lfloor E \rfloor * 3 < \varepsilon * n_{cs}$ decrease the frame size by $(cs\_ds\_ratio + ms\_per\_ds)$ min-slots.

Here n_{cs} is the number of contention slots per frame, *ms_per_ds* is the number of mini-slots contained in one data slot and *cs_ds_ratio* is the bandwidth allocation parameter.

The algorithm tries to predict systems behavior using probabilistic estimate, and not simply some heuristics. It is able to react faster than the previously described algorithms because it assigns the frame size a value expected to be sufficient, directly. Other algorithms were increasing the frame size gradually. It works better when it's window interval is smaller, i.e. when it recomputes the frame size more frequently, which is not true with other algorithms. Less variables and external parameters are used by this algorithm. The ε external parameter defines the percent of idle contention slots acceptable. Note that increasing this parameter will lead to improved performance, in terms of the request access delay, at high loads, but will degrade it for low loads and vice versa.
Decreasing the frame size is still performed gradually. We prefer to have excessive idle contention slots for some time in this case. The opposite (frame too small)would lead to high collision rate which is much worse. Next figure shows request access delay graph for the ternary tree with the above frame size adjustment algorithm and is compared to the ternary tree with a constant frame size.

[1] Formula for expectation is due to Prof. M. Perlis.

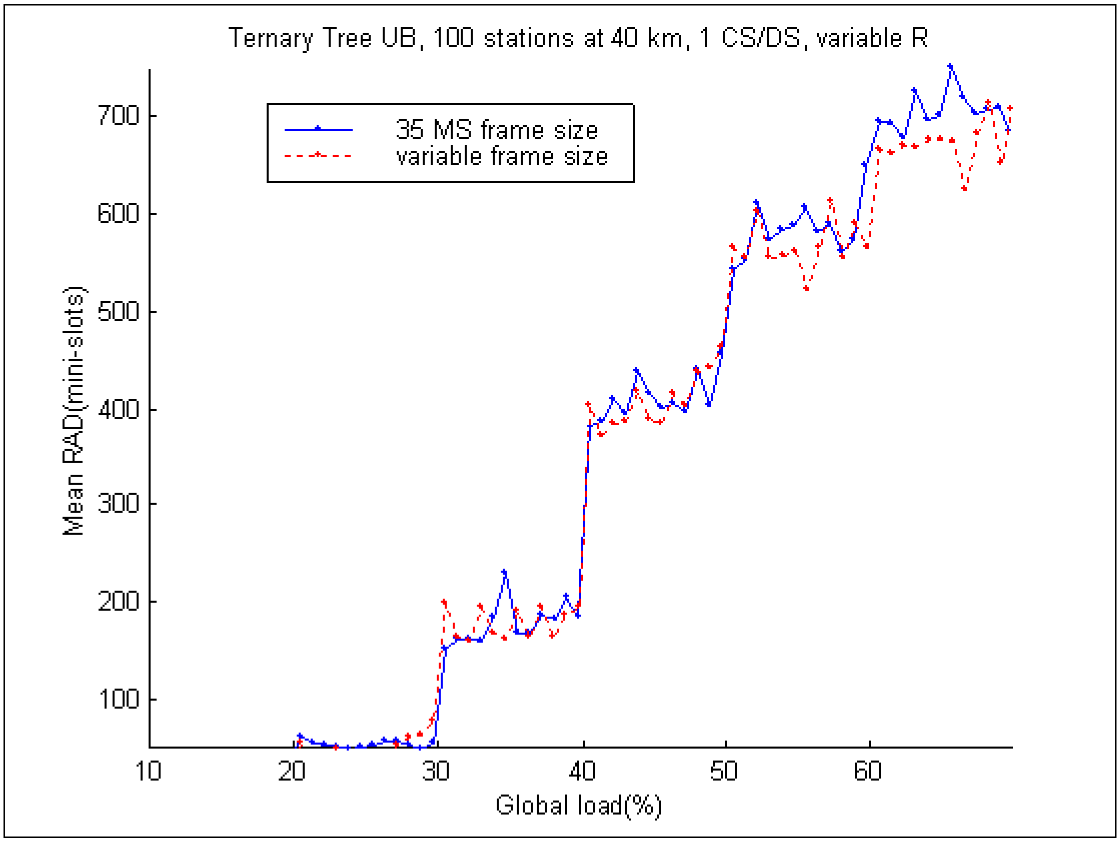

Figure 8.9

8.3 Conclusions.

We have seen that frame size has an influence on the mean request access delay, for the ternary tree CRA. The optimal value of the frame size, i. e. the one that allows for effective contention resolution without wasting contention bandwidth, can be found. The optimum depends on the offered global load and not on the round-trip delay time. Simple heuristic schemes for dynamic frame size adjustment, implemented at the HE, allow keeping the frame size close to optimal at different loads, and so to further reduce the access delay.

9 Summary.

9.1 Summary of results.

The results of the thesis can be subdivided, into the ones related to contention resolution algorithms and the ones regarding the HFC network architecture and configuration.
As for the first group of results, two main algorithms were considered: p-persistence and ternary tree, with the FIFO ordering of newcomers, as described by the last IEEE 802.14 draft. All the results agree with previous studies, in that ternary tree performs better than p-persistence, in terms of both lower delay and smaller delay jitter. In addition, the important role of the ranging parameter in both algorithms was demonstrated. One of the particularly noticeable effects was that when the ranging parameter oscillates near its mean value, the request access delay follows these oscillations closely. Other results suggest that when using the p-persistence algorithm, it make sense to give different

probabilities to contention slots at high loads, and also that the request access delay jitter would decrease if the distance to the head-end would be shortened. The continuous mode p-persistence performs better than the clustered mode, when the round-trip delay is small. The ternary tree algorithm proved to be less susceptible to the jitter vs. distance problem, as it has more centralized control. A simple collision-based scheme for variable bandwidth allocation was showed to perform even better than the traffic-based algorithm by K. Sriram, in the ternary tree case. Using the collision-based bandwidth allocation, the ternary tree algorithm was shown to be able to gain more performance improvement than p-persistence, when the distance to the head-end is shortened.
As for the second group of results, we showed that the systems' performance, in terms of request access delay, improves linearly with a decrease in the number of stations served by the head-end, while bringing the head-end closer to the stations gives little improvement at all. When the variable bandwidth allocation was used, the performance improved more, in the latter case. Still, the improvement is small, when compared with the improvement achieved after reducing the number of stations. Collisions here make the major contribution to the delay, and this cannot be improved by decreasing the reaction time. An additional improvement was achieved after we found that the frame size in the clustered mode has an influence on the request access delay, and the exact pattern of the influence was studied. An algorithm for dynamically maintaining the frame size close to optimal at given load was found to be easy to implement, at the head-end.

9.2 Future work.

There are many other architectural aspects and operational parameters in HFC networks that can be changed.
It would be interesting to examine how the above systems' performance may change if the stations produced a self-similar traffic. It would be of particular interest to receive an input from a realistic web-traffic generator that is able to imitate the behavior of Internet users, or even better - access to real data. Changing the traffic patterns could suggest changing the contention resolution algorithm so that it will allocate contention slots based on the estimation of request transmission probability from a given group of stations.
Another research issue is how exactly the tendencies presented above will change when the number of stations served will increase to thousand(s).
Yet another research direction can be exploiting the possibility of building a two (or more) level HFC system. In such a system, simpler head-ends are located close to the stations on the first level, and they aggregate their bandwidth requests in order to forward them together to the second-level head-end, placed at the root of the tree.

References.

[1] T. Aprille, L. Schwerin, J. Sipes, and N. Stevens, "Interactive Broadband Services and PCS Network Structure", Bell Labs Technical Journal, pp. 11-26, June 1996.

[2] Paul F. Gagen, William E. Pugh, "Hybrid Fiber-Coax Access Networks", Bell Labs Technical Journal, pp. 28-35, June 1996.

[3] S. Kempainen, "Chips and high-speed cable modems enable two-way communications", EDN Magazine, pp. 57-70, May 21, 1998.

[4] C. Bisdikian, K. Maruyama, David I. Seidman and Dimitros N. Serpanos, "Cable Access Beyond the Hype: On residential broad-band data services over HFC networks", IBM Research Report, 7/1 1996.

[5] "Cable Data Modems. A Premier for Non-Technical Readers", CableLabs, April 1996, http://www.cablelabs.com/.

[6] A. Dutta-Roy, "Cable is not just for TV", IEEE Spectrum, pp. 53-59, May 1999.

[7] Thomas J. Aprille, Larry M. Schwerin, James D. Sipes and Nancy S. Stevens, "Interactive Broadband Services and PCS Network Structure", Bell Labs Technical Journal, pp. 11-26, June 1996.

[8] David Gingold, "Integrated Digital Services for Cable Networks", Master Thesis, MIT, September 1996.

[9] Rolf V. Østergaard, "Cable Modem Tutorial", http://www.cable-modems.org/tutorial.

[10] D. Bertsekas, R. Gallager "Data Networks", Prientice Hall, 1992.

[11] P. Mathys, P. Fajolet, "Q-ary Collision Resolution Algorithms in Random-Access Systems with Free or Blocked Channel Access", IEEE Trans. On Information Theory, Vol. IT-31, no. 2, pp. 217-43, March 1985.

[12] S.S. Lam, "Packet Broadcast Networks – A Performance Analysis of the R-ALOHA Protocol", IEEE Trans. Comp., vol. C-29, no. 7, pp. 596-603, July 1980.

[13] W. Xu, G. Campbell, "A Distributed Queuing Random Access Protocol for a Broadcast Channel", DQRAP Research Group Report 90-1, CS Department, IIT.

[14] C. Wu, G. Campbell, "The Extended DQRAP(XDQRAP) algorithm", DQRAP Research Group Report 94-3, Dec. 9, 1994.

[15] C. Bisdikian, B. McNeil, R. Norman and R. Zeiz, "MLAP: A MAC Level Access Protocol for the HFC 802.14 Network", IEEE Communications Magazine, pp. 114-121, March 1996.

[16] B. Doshi, S. Dravida, P. Magill, C. Siller, Jr., and K. Sriram, "A Broadband Multiple Access Protocol for STM, ATM, and Variable Length Data Services on Hybrid Fiber-Coax Networks", Bell Labs Technical Journal, pp. 36-65, Summer 1996.

[17] J. O. Limb, D. Sala, "A Protocol For Efficient Transfer of Data over Hybrid Fiber/Coax Systems", IEEE/ACM Transactions on Networking, Vol. 5, No. 6, pp. 872-881, Dec. 1997

[18] D. Sala, J.O. Limb, S. U. Khaunte, "Adaptive Control Mechanism for Cable Modem MAC Protocol", Proceedings of INFOCOM '98, San Francisco, CA, March 29 - April 2, 1998.

[19] Ying-Dar Lin et al., "PCUP: Pipelined Cyclic Upstream Protocol Over Hybrid Fiber Coax", IEEE Network, Vol. 11, No. 1, pp.24-34, January/February 1997.

[20] IEEE Project 802.14/a, Draft 3, Revision 3, October 1998.

[21] Data-Over-Cable Service Interface Specifications, Radio Frequency Interface Specification, SP-RFIv1.1-I01-990311, CableLabs Inc., 1999.

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

[23] Y-D. Lin, "On IEEE 802.14 Medium Access Control Protocol", IEEE Communications Surveys, vol. 1 no. 1, Fourth Quarter 1998.

[24] N. Golmie, S. Masson, G. Pieris and D. Su, "A MAC Protocol for HFC Networks: Design Issues and Performance Evaluation", National Insititute of Standards and Technology, 1997.

[25] N. Golmie, Y. Saintillan, D. Su, "Continous vs Cluster Mode: Simulation Results", Contribution to the IEEE 802.14 WG, Document no. 802.14/97-76.

[26] Y-D. Lin, C-Y. Huang, W-M. Yin, "Allocation and Scheduling Algorithms for IEEE 802.14 and MCNS in Hybrid Fiber Coaxial Networks", IEEE Transactions on Broadcasting, Vol. 44, No. 4, pp. 427-235, 1998.

[27] K. Sriram, P.D. Malgill, "Enhanced throughput efficiency by use of dynamically variable request mini-slots in MAC protocols for HFC and wireless access networks", Telecommunication System No. 9, pp. 315-333, 1998.

[28] W.Cicoria, J.Farmer, D. Large, "Modern Cable Television Technology", Morgan Kaufman Publishers, Inc., 1999.