中華民國八十六年全國計算機會議

互動式有線電視網路的媒介存取協定

On IEEE 802.14 Medium Access Control Protocols

林盈達 Ying-Dar Lin
李俊弘 Chun-Hong Lee

國立交通大學資訊科學系
Department of Computer and Information Science
National Chiao Tung University

摘要

本篇論文提供一完整 "使用現有單向有線電視網路以提供高頻寬互動式服務"的調查報告。首先我們介紹有線電視網路中的混合式光纖同軸電纜架構，並討論在該網路上提供雙向通訊所會遭遇之問題。接著我們討論設計上行頻道多重存取協定上的幾個問題 1)把資料傳送過程與碰撞解析過程同時進行以達有效使用頻寬; 2)碰撞解析; 3)同步化。我們提出自己的方法來取得來回延遲修正值。也會介紹 IEEE 802.14 所採用的碰撞解析方法。

Abstract

In this article, we present a comprehensive survey on using existing CATV networks to provide high bandwidth interactive services. We first introduce the IEEE 802.14 HFC architecture of CATV networks and discuss the problems in providing two way communication over HFC. Then, we discuss the important issues in MAC protocol design, including 1) overlapping the collision resolution and data transmission to achieve best usage of bandwidth; 2) resolving collisions; 3) resolving the synchronization problem, in which we present a solution to get the RTC(Round Trip Correction) parameter. Resolutions adopted by IEEE 802.14 committee are illustrated.

1. Introduction

The information revolution has changed the way we work and is changing the way we live. We can see it in the near future that the multimedia interactive services, such as video-on-demand, high quality video phone, home shopping and financial transactions, will be available to residences. People can easily get information at home.

The CATV industry was established by providing low cost distribution of broadcast video signals to subscribers. The HFC (Hybrid Fiber Coax) architecture[1,2] has become standard in the CATV industry and provides up to 750 MHz of bandwidth to the subscribers in the forward (broadcast) direction. Because the architecture does not contain switching elements in the distribution plant, and only requires optical-to-electrical and electrical-to-optical conversion, amplification, and power splitting, the plant cost is relatively low.

The HFC architecture is also being considered as a bi-directional broadband communication infrastructure, with the 5 to 40 MHz portion of the spectrum available in the distribution plant for transmission from the subscribers to the headend/central office. The cost for adding telecommunication services over the existing HFC networks are relatively low. Several standard associations such as DAVIC, IEEE 802.14, National Cable Television Associations, and Interactive Multimedia Association were formed to seek the total solutions for interactive multimedia services over backbone and community networks.[1,2]

1.1. Hybrid Fiber/Coax (HFC) Network Architecture

In a modern CATV plant employing a HFC "tree and branch" architecture, stations attached to a cable network transmit and receive information using separate frequencies. There is a downstream channel (from the headend to the stations) and an upstream channel (from the stations to the headend).

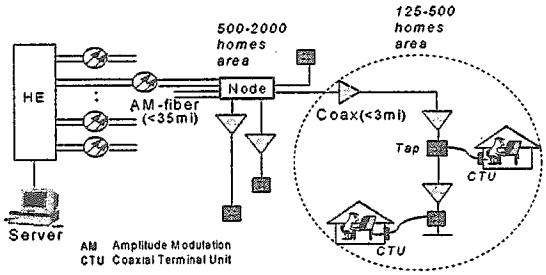
HFC, using analog access technology for CATV community network, is gradually becoming the standard for some cable and telephone companies. Digital transmission is attained by modulating the packetized digital information onto analog RF (Radio Frequency) carriers, via QAM and QPSK. [1,3]

The frequencies from 100 MHz up to the upper

frequency limit supported by the cable plant(i.e., 750 MHz) are allocated for downstream transmission. The sub-split band, i.e., frequencies between 5-42 MHz are allocated for upstream transmission.

HFC has the following important features.

- tree and branch topology
- large propagation delay
- asymmetric upstream and downstream
- user distribution -- Most of the users are distributed on the last few miles of the network.

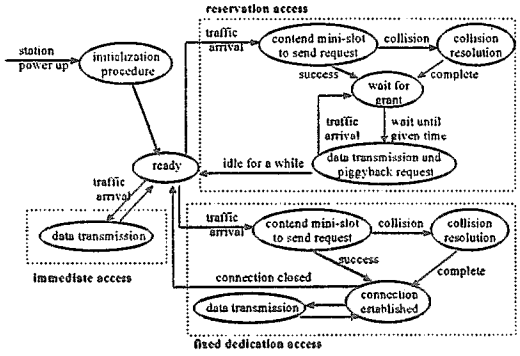


■ *figure 1.1 The architecture of HFC.*

Fig. 1-1 represents a piece of a HFC system. Clusters of homes, 500 to 2000 subscribers, are served by a fiber that comes from the headend to a fiber node. These signals are distributed to homes by electrical signal within the serving area of a fiber node via an amplified tree-and-branch feeder cables, perhaps as short as 3 miles in total length. Each branch affords 125 to 500 subscribers, the stations can not listen directly to the upstream transmissions from other stations; hence, they are incapable of detecting collisions and ultimately coordinating their transmission all by themselves. With multiple access technology, upstream channel lets all subscribers within a branch share the available reverse bandwidth in 5-40 MHz and a single fiber carries the signal back to the headend.[3]

2. Protocol Overview

In the HFC network, part of the downstream channel bandwidth will be used to broadcast control and feedback information as required for MAC protocol implementation of the upstream channel.



■ *figure 2.1 station state diagram of HFC upstream protocol*

Figure 2.1 is a state diagram of stations. In this section, we first discuss three basic medium access modes shown in the state diagram and how they work. The layout of upstream channel is also presented.

2.1. Three Types of Data Transfer

Most of the HFC upstream channel protocols support three access modes, namely, reservation access mode, contention access mode and fixed dedicated access mode.

2.1.1.Reservation Access Mode (Request/Grant)

The reservation access mode provides the ability to dynamically assign bandwidth in a per-request manner. As its name implies, a station has to request to tell the headend how much bandwidth it needs. The headend then reserves bandwidth for this station according to its allocation algorithm.

It is very inefficient if a collision occurs when a station issues a bandwidth request. Mini-slot and piggyback idea is introduced to improve the request operation efficiency. Using mini-slot can reduce the wastage caused by collision and piggyback can reduce request frequency.

Mini-slots

Contention access is not very efficient because of collisions. The mini-slot idea is thus introduced. Mini-slots are used by the stations to send requests. Assume one normal slot size is k times of the mini-slot size. If while using a normal slot a collision occurs, k times of bandwidth is wasted than using a mini-slot.

Piggyback Request for Subsequent Transmissions

If a station is not a newcomer and the headend has already allocated bandwidth for it, it can piggyback its next bandwidth request at the end of its data transmission. The station does not have to compete for the mini-slot again to send a request. This mechanism makes things easier. Only the first transmission of a burst has to compete for the mini-slots.

2.1.2.Fixed Dedicated Access Mode

This access mode intends to support CBR (Constant Bit Rate) applications. Stations have to signal the headend to establish connections prior to data transmission. A station that wants to set up a connection first competes for a mini-slot to send its request. If the headend grant this request, it assigns a position in the upstream channel periodically and informs the station

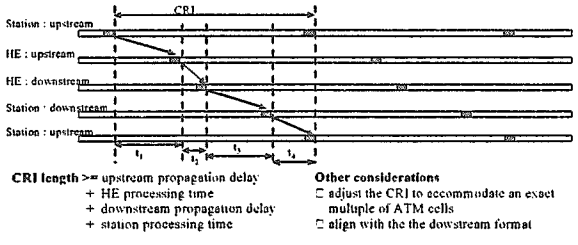
which time slot and the amount of data to transmit. A station that has established a CBR connection does not need to request for a permission to transmit each time.[4,5,6,8]

2.1.3.Contention Access Mode

The headend may provide unallocated upstream bandwidth to contention access mode. When the network is completely idle, in some protocol designs that all of the upstream bandwidth is set to contention access mode. Usually the bandwidth is divided into small size units for small packet transmissions and short-lived applications, for example, when a button is pressed in the interactive TV application. It can also be used for retransmission and station power-on registration. Common agreement reached by 802.14 committee does not support this access mode. That means no data can be sent without sending the requests first, expect the piggybacked requests.

2.2. Allocation of Slots and Mini-slots

A CRI (Contention Resolution Interval) is defined as the maximum time required for all nodes on the network to detect whether a contention transmission was successful or collided with others. As shown in figure 2.2, the station first randomly chooses a mini-slot in the cluster and sends its request. The CRI length must be larger than the sum of t_1 ,t_2 ,t_3 and t_4 so that in the best case the station can retransmit its request immediately.

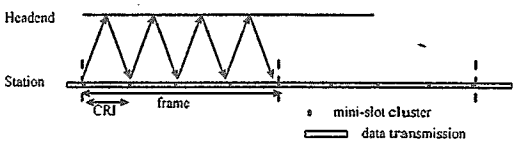


■ *figure 2.2 CRI size consideration*

Having the CRI idea, we can image that if the station sends a request in upstream channel and does not transmit anything while waiting for the acknowledgment. The upstream is idle most of the time. That means most of a CRI time is wasted. In the following sections, we are going to discuss two upstream frame layout methods, namely, concurrent method and concurrent method with interleaving.

2.2.1.Concurrent Method

Concurrent method overlaps the collision resolution and data transmission. When a station waits for the acknowledgment of a previous request, at the same time upstream data transmission is under way. The data transmission part and request mini-slot part are mixed, as shown in figure 2.3. In this method, a frame contains several clusters of mini-slot and data transmission parts. Each cluster of mini-slots is followed by a CRI length data transmission part. After sending a request in cluster *j*, a station knows whether its request is successfully transmitted or not before cluster *j+1* comes and retransmit its request if possible.

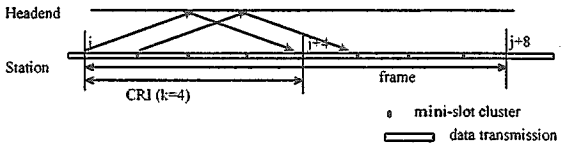


■ *figure 2.3 illustrating concurrent methods*

2.2.2.Concurrent Method with Interleaving

Concurrent method with interleaving has multiple contention resolution engines. Each contention resolution engine operates independently (i.e. interleaving). In concurrent method we discussed above, there is only one mini-slot cluster per CRI. With interleaving, we can have more than one mini-slot cluster per CRI as shown in figure 2.4. For a network with bigger size, the CRI becomes larger and there are more subscribers. We can use the interleaving skill to improve collision resolution efficiency.

If there are *k* clusters of mini-slots in a CRI, the interleaving factor for the network becomes *k*. There are *k* contention resolution engines running in parallel. For example, cluster {1, k+1, 2k+1,...} use the same collision resolution engine. After sending a request in cluster *j*, a station can know whether its request is successfully transmitted or not before cluster *j+k* and retransmit its request if possible.



■ *figure 2.4 illustrating concurrent method with interleaving*

3. Collision Resolution

Collision resolution is an important issue for the HFC upstream protocol. There are two places, in the initialization (power up) of new stations (asynchronized collision resolution) or during the first transmission of a station in the reservation access mode and the fixed dedicated access mode (synchronized collision resolution), that we may need collision resolution. Before examining the collision resolution algorithms, we describe these two circumstances:

- System initialization procedure

The headend has to recognize all active stations connected to the system before providing service to these stations. Once the registration request sent by stations is recognized by the headend, a dialog would take place between the station and the headend to perform the full set of registration functions including ranging and parameter setting.

- Request / Grant procedure

  In reservation access mode, when a station has data to transmit, it first contends a request mini-slot. When the request is successfully transmitted to the headend, the headend replies an acknowledgment in downstream. The acknowledgment only tells the station that its request is successfully received by the headend. The station has to wait another grant message sent from headend which contains transmission start time(Si) and the transmission duration(Ti). With these information, the station knows when to transmit and how long it can transmit.

3.1. Collision Resolution Algorithms

The theory of collision resolution (random access) algorithms for medium access in computer communication networks has matured for well over a quarter of a century. Many algorithms has been proposed, studied and analyzed.[5,6,7] In this section, we will introduce two algorithms adapted by IEEE 802.14 committee to the HFC network.

3.1.1.P-persistence

In p-persistence, the headend chooses a value p ($0<p<=1$) and sends it to all stations. Before a station attempts a transmission, a number is randomly generated between zero and one. If this value is smaller than p, the station transmit, otherwise the station waits. One feature of this algorithm is that whether or not to transmit is independent of previous attempts. No state is stored in the station and the headend. The optimal p value has been proved to be $1/N$, where the N is the number of station involved.

3.1.2.N-ary Tree Based Algorithm

In this algorithm, each station has a counter. The counter value means how many frames you have to let pass before your next transmission. If the counter equals to zero, the station is permitted to transmit in the next frames. Depending on the result of the contention slot, the counter of each station is adjusted as follows: [8,9]

- if collision :
  the station is involved in this contention slot → counter = random[0,n-1]
  the station is not involved → counter = counter + (n - 1)
- if no collision (successful transmission or idle) :
  the station is involved in this contention slot → done
  the station is not involved → counter = counter - 1

The n-ary tree based algorithm can be divided in to the following two types:

blocked-access Some implementation blocks newcomers to join the collision resolution. That means newcomers are not allowed to transmit their requests until the current collision is resolved. This is usually referred to as **tree-search algorithms.**

free-access In non-blocking implementation, each newcomer has a counter value zero. So it can join the collision resolution any time, even when the collision resolution is on-going. Usually referred to as **stack algorithm**

This algorithm has been proven that it is more efficient and the simulation result tells us the same thing.[3,4] So there are many proposed protocols use this algorithm.[15,16,17]

3.1.3.Summary

After presenting these algorithms, we would like to give the definition of the **stability** of a network operating under a particular collision resolution algorithm. Let λ and μ denote the arrival and departure rates of packets to and from the network, respectively. The positive number Λ will be called the throughput of the collision resolution algorithm, while the arrival rates ranges between 0 and Λ.

> A network operating under a collision resolution algorithm is called stable if and only if there exists a positive number Λ such that for all λ that $\lambda <= \Lambda$, $\mu = \lambda$.(i.e., the rates are maintained, for all $\lambda <= \Lambda$, where $\Lambda > 0$.)

The instability problem of ALOHA based algorithm (binary exponential back-off and p-persistence) has long been documented. It has been proven analytically and verified through simulation studies that pure acknowledgment based ALOHA algorithms are unstable. No matter at what rate new traffic is generated, if the system operates for a sufficiently long time it will eventually be overwhelmed by collisions, and nothing will be transmitted. That means they posses no stability region, i.e., $\Lambda = 0$. ALOHA based algorithms can be improved to be stable. [9].

In many studies, it has been proven that the splitting algorithms(tree walk and n-ary tree based) are stable and can achieve throughput much higher than the stable ALOHA. The simulation result borrowed from [9] shows that the ternary(n=3) tree based algorithm achieves shortest collision resolution interval and binary(n=2) tree based algorithm is also close behind. ALOHA based algorithms take the longest time to resolve the collisions. IEEE 802.14 committee has the consensus that the contention resolution scheme should be based on the ternary tree-search algorithm.

3.2. Sequential and Parallel Implementation

CATV network has a large propagation delay. When a station competes to request, it has to wait a round trip feedback time before knowing whether its request is successful or collided. In CATV network, feedback time is of the order of 800 μs(when network length is 80 km).

Let us take a look at the implementation issues. Among the IEEE 802.14 proposals one can identify two main classes of implementations, namely sequential implementation and parallel implementation. Assume we use n-ary tree base algorithm. (Most proposed protocols use this algorithm.)

If a collision occurs at time t_0, then at t_0 + CRI each station involved in this collision sets its counter randomly between zero and n-1. Now we use the same way we described in n-ary tree based algorithm to adjust the counter value, but two different ways to allocate mini-slot clusters:

- Simple sequential implementation

  Mini-slot clusters are arranged at {t_0, t_0 + CRI, t_0 + 2*CRI, ..., t_0 + n*CRI, ... }. Each mini-slot cluster has only one mini-slot.
- Simple parallel implementation

  Mini-slot clusters are arranged at {t_0, t_0 + CRI, t_0 + 2*CRI, ..., t_0 + n*CRI, ... }. Each mini-slot clusters has three mini-slot if collision resolution is under way and only has one mini-slot if all collision is resolved.

It is obviously that the sequential implementation is simpler and intuitive, on the other hand parallel implementation can achieve shorter collision resolution time. There are also many other possible parallel implementations, but some of them lead to non-trivial slot allocation. IEEE 802.14 committee does not put the mini-slot allocation algorithm in its standard discussion.

Recall the idea of mini-slot cluster idea we mentioned in section 2. Following is a simple summary table. Note that the concurrent method have a mini-slot cluster per CRI, i.e. k=1. If we use sequential implementation, each mini-slot cluster has only one mini-slot.

	Sequential	Parallel
Concurrent (k=1)	One cluster per CRI One mini-slot per cluster	one cluster per CRI more than one mini-slot per CRI
Concurrent with interleaving (k>1)	More than one cluster per CRI One mini-slot per cluster	More than one cluster per CRI More than one mini-slot per cluster

■ *table 3.1 Summary of sequential/parallel concurrent mini-slot cluster implementation w/ and w/o interleaving*

3.3. Collision Resolution Mechanism

The IEEE 802.14's collision resolution algorithm is a combination of p-persistence based and 3-ary tree based collision resolution algorithm. See[9,10].

Mini-slot Allocation Algorithm

To support variable number of mini-slot per cluster, the headend has a mini-slot allocation algorithm decides the number of mini-slots in each cluster. The standard does not mention the detail of the mini-slot allocation algorithm.

RQ Value

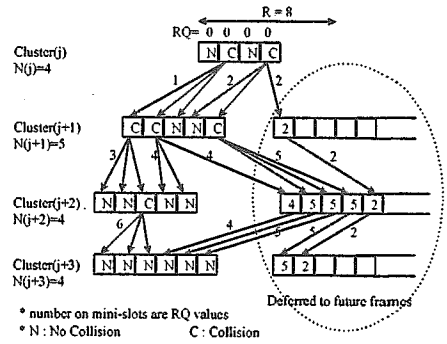
The RQ (Resolution Queue) value is a number provided by the headend and distributed to the stations indicating which collision groups are being resolved. From a station's point of view, it saves the RQ value provided in the headend acknowledgment message. When the next cluster of mini-slot comes, it checks the RQ values to see if it is the same as the saved RQ value. If it is, the station will compete for this mini-slot group or if it is not the same it will continue to wait.

First Transmission Rule (P-persistence Based)

The headend assigns a network parameter R to every station. The number of mini-slot in cluster(j) is N(j) and N(j)<=R. The available cluster means that no collision resolution is under way now, that is RQ = 0. When a station wants to make a request in an available cluster(j), it randomly generates a number i between zero and R. If i is between zero and N(j), the station can send request in mini-slot I, else the station retry in next available cluster. This first transmission rule is called "soft blocking".

Implementation

IEEE 802.14 employees the 3-ary tree based algorithm because a good deal of research has been done that indicates an optimum number of mini-slots that should be made available for splitting is three. Following is an example.



■ *figure 3.1 Collision resolution concept*

As shown in figure 3.1, newcomers randomly pick a slot position *i* between 1 and 8(R=8). The transmission is success for those stations with *i* equal to 0 or 2 and collision occurs for those stations with *i* equal to 1 or 3. Those stations with *i* equal to 4, 5, 6 and 7 do not transmit anything in this cluster but competes for the next cluster. For those stations with *i* equal to 1(it is a collision in figure 3.1), the headend provides a acknowledgment with RQ = 1. These stations save this RQ value and when the

next cluster of mini-slot comes, it checks the RQ values to see if it is the same as the saved RQ value. If it is, the station will compete for this mini-slot group or if it is not the same it will continue to wait.

The simulation result shows this solution can achieve a better performance than pure 3-ary tree based algorithm and p-persistence algorithm. See [11,12].

4. Synchronization

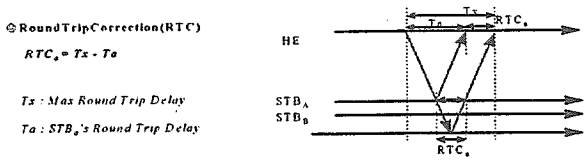
The HFC network should have two level of synchronization. Physical level synchronization can align the signals in bit level and MAC level synchronization can align the bit stream in packet level. This section we discuss the MAC level synchronization.

4.1. Compensating Network Delay

The HFC environment poses a challenging problem to the network designer since propagation delay can be much larger than the transmission time for packets or cells. To overcome this, every station needs to have the following information.

- a global timing reference
- its round trip delay correction

Once these are known, every station can then precisely transmit data in a given time slot and the headend can assign the slots or bandwidth as needed. With these two pieces of information, we can avoid collisions due to different relative locations of stations. Figure 4.1 shows this scheme. $station_A$ and $station_B$ are located at different distances from the headend. Their round trip propagation delays are *Ta* and *Tb*. The network's maximum propagation delay is *Tx*. RTC_A is *Tx-Ta* and RTC_B is *Tx-Tb*. When the headend transmits a sync message in the downstream channel, $station_A$ and $station_B$ will not see the sync message at the same time and their recognition of start of a frame will not be the same. After the initialization procedure, each station gets its own RTC. It knows that the RTC time after they see the sync message is the start of the frame.[13]



■ *figure 4.1 compensating network delay*

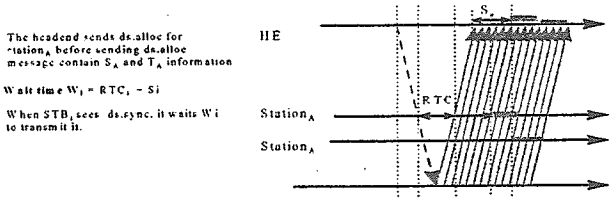
Global Timing Reference

The headend must transmit a sync message in the downstream channel periodically. The exact interval is non-critical but is assumed to be on the order of several to tens of milliseconds. Usually we think sync message as the start of a frame.

Round Trip Delay Compensating (RTC)

During the initialization procedure, each station must perform an operation to get its RTC. We call this procedure ranging or station positioning. We propose our solution to get RTC parameter in the next section.

As illustrated in figure 4.2, the headend schedules $station_A$ to transmit data in this frame, starting at S_A and continuously transmitting for T_A time. It means that after $station_A$ see the sync message, it waits RTC_A time until the start of the frame and then waits extra S_A time until the time slot reserved for it. After that, it continuously transmits T_A time.



■ *figure 4.2 an example of data transmission with RTC synchronization*

4.2. Ranging

Ranging and competing mini-slot in reservation access mode in HFC upstream channel protocol both require contention resolution processes. The main difference between ranging and request in reservation access mode is that the ranging process is non-slotted. The propose of ranging process is to get RTC parameter. Of course no RTC parameter is known when ranging is underway.

Non-slotted vs. Slotted

In non-slotted case a station transmits its registration request at time t_1. Assume that the time required to completely transmit the registration request packet is L. Successful transmission occurs when both of the following two conditions are true:

- No other station transmits their request during (t_1 - L, t_1). This transmission will not collide with other station's unfinished transmission.
- No other station transmits their request during (t_1,t_1 + L). This transmission will not be collided by other station.

It looks like that the transmission time of registration request packet becomes 2*L, instead of L. That is because the collided transmission will not overlap completely.

In slotted case a successful transmission only requires no other transmission at the same time. If collision occurs, the collided transmission overlap completely.

Non-slotted Contention Resolution

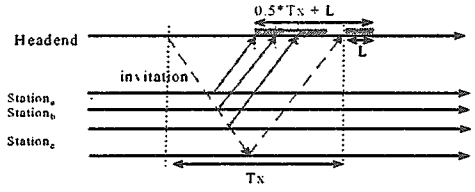
Assume that the maximum round trip propagation delay of the network is Tx and all stations are distributed over the last 50% of the network, i.e. the front half of the cable has no station attached to it. Further assume that the first station reads the headend downstream message

earlier than the last station by 0.25*Tx.

Original Idea : The headend periodically invites newcomers by sending invitation messages through the downstream channel. If $station_i$ wants to register itself to the headend, it transmits its registration request immediately when it reads the headend downstream invitation. As shown in figure 4.3, $station_a$ collides with $station_b$ and $station_b$ collides with $station_c$. In this method, a collision occurs when two stations which may be on the same branch or different branches of the network, have close positions relative to the headend. That means the difference of their round trip propagation delay is within L(i.e. $| T_a - T_b | < L$).

Assume that $station_i$(the first station in the network)is positioned in middle of the network and its propagation delay P_i. is equal to 0.25*Tx. $Station_j$(the farest station in the network) is positioned on the end of the network and its propagation delay P_j is equal to 0.5*Tx. If the headend sends the invitation message at time t_0, the $station_i$ reads this message at t_0+0.25*Tx and its transmission arrives the headend at time t_0+0.25*Tx+P_i(i.e. t_0+0.5*Tx). The last station may reads this message at t_0+0.5*Tx and its transmission arrives the headend at t_0+0.5*Tx+P_j. Its last bit arrives the headend at t_0+0.5*Tx+P_j+L(i.e. t_0+Tx+L). The overhead of this registration process is 0.5*Tx + L per frame.

This method uses each station's position difference to avoid collisions. However, the CATV network has the characteristics that most users are distributed on the last few miles of the network. This causes most subscribers close to each other. Because users usually are located in a neighborhood. Increasing user population in the community causes high subscribers density per mile over the network. The result will be so many collisions that this method will not function satisfactorily.



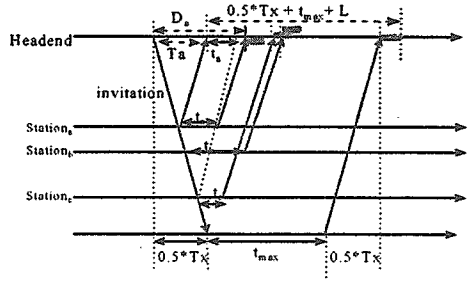
■ *figure 4.3 non-slotted contention resolution : Method 1*

Improved method : If $station_i$ wants to register itself to the HE, it randomly waits t_i time and then transmits its registration request. The random waiting time t_i should be within the maximum waiting time t_{max}. If a collision occurs, it waits for the next invitation. The t_{max} information can be contained in the headend invitation message or the headend broadcasts t_{max} information in the downstream when t_{max} changes.

In figure 4.4, $station_a$ successfully transmits its registration request while $station_b$ and $station_c$ collide with each other. Collision occurs when two station have the difference of the sum of their round trip propagation delay and random waiting time is within L(i.e. $| (t_i + Ti) - (t_j + Tj) | < L$).

Assume that $station_i$(the first station in the network)is positioned in middle of the network and its propagation delay P_i. is equal to 0.25*Tx. $Station_j$(the farest station in the network) is positioned on the end of the network and its propagation delay P_j is equal to 0.5*Tx. If the headend sends the invitation message at time t_0, the $station_i$ reads this message at t_0+0.25*Tx and its transmission arrives the headend at time t_0+0.25*Tx+P_i+t_i(i.e. t_0+0.5*Tx+t_i). The last station may reads this message at t_0+0.5*Tx and its transmission arrives the headend at t_0+0.5*Tx+P_j+t_j. Its last bit arrives the headend at t_0+0.5*Tx+P_j+L(i.e. t_0+Tx+L+t_j). We can calculate the overhead of the registration process with t_i equals to zero and t_j equals to t_{max}. The overhead of this registration process is 0.5*Tx +t_{max}+ L per frame.

In this method if the subscriber density per mile increases, all the system has to do is adjusting the t_{max} parameter. Here we use the relative propagation delay and the random waiting time to avoid collisions. Another implementation alternative is that we can adjust the t_{max} according to the current collision condition. If the previous frame has collisions, we can use larger t_{max} for the next frame. This can achieve smaller contention resolution time.



■ *figure 4.4 non-slotted contention resolution : Method 2*

Station$_i$'s random waiting time t_i is contained in its request. When the request is successfully received by the headend, the headend can correctly calculate the RTC_i value using the following two formula:

(1) RTC_i = Tx - Ti (as mentioned in section 4.1. See figure 4.1.)
Tx : Network maximum round trip delay
Ti : Station$_i$'s round trip delay

(2) Ti = Di - ti (as shown in figure 4.4.)
Di : time difference between the headend sends out invitation and the headend receives the station$_i$'s request.

The headend first uses formula (2) to calculate Ti. Then it uses formula (1) to get RTC_i. After correctly calculating RTC_i value, the headend sends RTC_i parameter to station$_i$ and a dialog would take place between station$_i$ and the headend to perform the full set of registration functions.

In MLAP/MLMP proposed by IBM, a similar ranging algorithm is introduced[18]. In that method, however, the headend needs to timestamp the invitation message.

5. Conclusion and Future Works

We have a complete survey study on upstream channel protocol in HFC network. We identified the three important issues: 1) the upstream frame layout; 2) the collision resolution mechanism; and 3) the synchronization problem. Also we have proposed our ranging method to get RTC parameter. We compared several protocols and introduced the IEEE 802.14's solution.

We have the conclusion that an efficient HFC upstream protocol should have the following properties.

- support reservation access and fixed dedicated access
- provide concurrent data transmission and collision resolution (concurrent method discussed in section 2)
- with a high performance collision resolution mechanism, support variable number of mini-slots (3-ary tree based collision resolution algorithm and parallel implementation discussed in section 3)
- support ATM cell and variable length packet

The IEEE 802.14 standard will be released soon. There are several problems that the standard does not and will not cover. Our future work can be continued in these directions: 1) bandwidth allocation algorithm - When the headend receives the bandwidth requests, how does it schedule the stations' data transmission? ; 2) the mini-slot allocation algorithm - In parallel implementation, how many mini-slots should be there in a mini-slot cluster?

6. References

[1] Charles A. Eldering, Nageen Himayat, and Floyd M. Gardner, "CATV Return Path Characterization for Reliable Communications", IEEE Communication magazine, August 1995.

[2] Y.-D. Lin and C.-J. Wu, "NII Community Network : Interactive CATV", Proc. Int'l. Conf. Comp. Sys. Tech. For Industrial Apps. (CSIA), Hsinchu, Taiwan, Apr. 1996.

[3] Y.-D. Lin, C.-J. Wu, and W.-M. Yin "PCUP : Pipelined Cyclic Upstream Protocol Over HFC", IEEE Network magazine, vol 11, No1, January /February 1997, pp. 24-34.

[4] IEEE 802.14 draft,"802.14MAC/V1.1", December 31,1996.

[5] IEEE 802.14 -96/112R2, "MAC convergence Proposal", September 17, 1996.

[6] SCTE Data Standards Subcommittee Working Group E contribution, "The Current Status of Convergenced 802.14 MAC Proposal", December 2 1996.

[7] Andrew S. Tanenbaum: Computer Networks, third ed., pp. 246-266, Prentice-Hall, 1996.

[8] IEEE 802.14 -96/217, "An example of a MAC based on Convergence Agreements(CMAC)", September 10, 1996.

[9] IEEE 802.14 -96/019, "A Review of Random Access Algorithm", January 1996.

[10] IEEE 802.14 -96/012, "Comparison of Algorithms for Station Power-up in an HFC Network", January 1996.

[11] IEEE 802.14 -96/246, "Implementation Overview of Tree-based Algorithm", Nov 11, 1996.

[12] IEEE 802.14 -96/244, "The Tree-based Algorithm with soft blocking", Nov 4, 1996.

[13] IEEE 802.14, "Framed Pipeline Polling for Cable TV Network", March 1, 1995.

[14] IEEE 802.14-96/076, "Phase 1 Evaluation Result of Framed Pipeline Polling", March 4, 1996.

[15] IEEE 802.14-95/068, "A Proposal to Use XDQRAP for 802.14", July 12, 1995.

[16] IEEE 802.14-95/157, "Formal Proposal for 802.14 MAC Protocol (Part 2 of 1): MLMP (MAC Level Management Protocol)", November 7, 1995.

[17] IEEE 802.14-95/158, "Formal Proposal for 802.14 MAC Protocol (Part 2 of 2): MLAP (MAC Level Access Protocol)", November 7, 1995.

[18] IEEE 802.14-95/144, "Formal MAC Layer Proposal : Adaptive Random Access Protocol for CATV", October 23,1995.

[19] IEEE 802.14 -96/019, "Preliminary Simulation Results of the MAC Protocol Proposals", May 1996.