A New Network Architecture with Intelligent Node (IN) to Enhance IEEE 802.14 HFC Networks

Shiann-Tsong Sheu and Meng-Hong Chen
Department of Electrical Engineering, TamKang University
Tamsui, Taipei Hsien, Taiwan 251, Republic of China
E-mail : stsheu@ee.tku.edu.tw

Abstract- In the hybrid fiber/coax (HFC) architecture, over several hundreds subscribers in CATV (Community Antenna TV) network may cause serious collisions. In this paper, we propose a new network architecture which using an Intelligent Node (IN) to stand for a group of subscribers to request the demand resources. The Intelligent Node has the ability to reduce the collision probability as well as the collision resolving period. The simulation results show that the proposed architecture in terms of throughput, buffer delay, and fairness outperforms the standard architecture.

1. INTRODUCTION

Recently, people use computers to connect the internet service at home. To support this, many solutions have been proposed, for example, the telephone modem, ISDN, ADSL, and CATV network. However, the bandwidth requirement of multimedia services, such as Video on Demand, Video Conferencing and Distance Learning, is far beyond the solutions of telephone modem and ISDN. Although ADSL provides downstream bandwidth up to 8 Mbps (to the user) and upstream bandwidth 1 Mbps (depending upon line length, loop and line conditions provide), it is still not apposite to provide multimedia services because of the point to point network architecture. Recently, many benefits of CATV network have been discussed.

CATV (Community Antenna TV) networks are traditionally one-way and broadcasting infrastructure for residential area TV distribution. With the population of internet services, cable services providers are interested in providing these internet services. Thus, two standard committees, IEEE 802.14 [1] and Multimedia Cable Network System (MCNS) [2] were formed to prescribe solutions about how to provide different kinds of services over the CATV networks.

The hybrid fiber/coax (HFC) architecture, in which a fiber is used to transport multiplexed signals to a group of 500 to 2000 subscribers, has become the standard in CATV industry. The HFC architecture is considered as a bi-directional broadband communication infrastructure, as shown in *Fig.1*. A group of 500 to 2000 subscribers are served by a fiber that comes from the headend controller (HC) to a fiber node (FN). Moreover, signals are transmitted electrically from FN to home by coaxial cable through some amplifiers and splitters. Stations attached to the cable transmit and receive signals over different frequencies, upstream and downstream channels, respectively. The HC divides the upstream channel into numbers of minislots and allocate to stations for requesting and transferring information. That is, the first HC allocates a lot of minislots for contention-based reservation and informs stations to send reservation message in these slots if they have data to transmit. Because users can not listen to the upstream channel, collisions are unable to be detected by users. The collision detection is done by the HC. This implies that traditional CSMA/CD protocol is not suitable in CATV network. Once the HC obtains the reservation result, it will inform stations when to transmit data or contend again. Because of the long propagation delay in HFC network, the throughput will be degraded due to the collision resolution mechanisms. Thus, some collision resolution mechanisms have been proposed and schemes like n-ary tree were considered in the standard. The simulation result from [3] shows that the ternary (n=3) tree algorithm achieves the shortest collision resolution interval and the binary (n=2) tree algorithm is close behind.

The operations of the n-ary tree algorithm is briefly described as follows: (1)HC allocates some Request Minislots (RMS) in the upstream channel and inform stations to send their request information into these minislots. (2)Stations having data to transmit will randomly select a minislot and transmit its request message on it. (3)After the HC collects all RMS, it will obtain the contention result. If there is a collision, after a round-trip propagation delay, HC will respectively allocate a number of n minislots to stations which colliding in the same minislot. (4)HC will repeat step 3 until all the collisions are resolved. (5)Finally, HC will send a Data Minislots (DMS) Grant Message informing the corresponding Data Minislots that stations can transmit.

Take *Fig.2* for example, at the first time, the HC allocates 4 minislots to stations for contention. Stations B and C contended in the first minislot and stations D and E contended in the third minislot, respectively. Only station A successes in the fourth minislot. After a round-trip propagation delay, the HC allocates three minislots (in this case, n=3) to stations B and C. In this case, assume no collision occurs. After then, another three minislots will be allocated to stations D and E. Also, we assume no collision occurs in this round. As soon as all stations are successful, the HC will allocates the data minislots to these five stations. Stations can transmit its data in their respective data minislots.

In HFC architecture, it is not difficult to imagine that 500-2000 subscribers in the network may cause excessive collisions. In this paper, we propose a new network

0-7803-6283-7/00/$10.00 © 2000 IEEE

architecture which using an Intelligent Node (IN) to reduce collision resolving period. The rest of the paper is organized as follows. The proposed network architecture is introduced in Section 2. The media access protocol is described in Section 3. In Section 4, the simulation model and simulation results are presented. The conclusions are given in Section 5.

2. Network Architecture

Traditionally, users want to transmit data should send the request onto the RMS to contend. This is the major drawback of CATV network to support thousands of subscribers. That is, the performance will be degraded with the increase of subscribers. In the proposed network architecture, we place some Intelligent Nodes (INs) in the traditional HFC network as shown in *Fig. 3*. The IN is the agent of a group of users. If there is any user(s) wants to transmit data, the IN will substitute for sending a single request message with the summed bandwidth onto the RMSs. When active users under the corresponding IN is more than one, the collision probability will be decreased than traditional HFC network because that the number of contending users is decreased. After contenting, the IN will inform the user when to transmit data.

In practical network, the IN can be placed in building. Users in the building entrusts the IN to contend the resource. It is very feasible by using the INs because we do not modify the traditional architecture. User who uses the traditional equipment needs not change if he still uses the traditional solution to request resource. Only the HC and the user equipment using the IN need modify slightly. The modified protocols for our network structure will be described in detail in the next section.

3. Media Access Control Protocols

There are three important features in the HFC network:

- Tree-and-branch topology
- Long propagation delay
- Asymmetric upstream and downstream

Intuitively, the designed protocols for such network are more complicated than general networks.

3.1 Intelligent Node (IN) Protocol

When an IN powers on, it must acquire a downstream channel. If the downstream channel does not contain data stream, the station should select another downstream channel.

After acquiring a downstream channel, the procedure of timing acquiring and ranging is the most important step in the initial state which determines the round-trip correction (RTC) parameter. *Fig. 4* illustrates the steps of timing acquiring and ranging of IN and STA. The HC shall send Signature messages regularly which carry the value of the timebase in the HC. When the IN receives the first Signature message, it shall set the station clock to this value and start the clock right away. As each Signature message is received, the IN compares the value with its clock and the rate of the IN clock is adjusted up to down to decrease the error. Timing is acquiring by this method. Besides, the HC periodically invites newcomers by sending ranging invitation message through the downstream channel. The IN sends back a ranging response at the target minislot time. When the IN is not ranged, the response might be received by the HC before or after the target minislot time. The HC will calculate the arrival time error and send back to the IN. The IN shall adjust according to the feedback information. When IN is ranged, the ranging procedure is finished. After these procedures, the IN will derive the RTC parameter.

After performing the procedure of timing acquiring and ranging, INs must register to the HC to get the ID and some upstream parameters. After then, it enters into the Agent state. In the Agent state, INs shall broadcast the signature and ranging invitation message through the downstream channel periodically. At the same time, the IN would wait for the Request Minislot Grant Message from the downstream channel sending by the HC. After receiving this message, the IN would also broadcast a Request Minislot Grant Message to its downstream stations. We note that this message is different from the message that HC sent. The IN Request Minislot Grant Message specifies the number and location of request minislots and contention minislots. Each request minislot is reserved for individual station, which has successfully joined (associated) the IN. The contention minislots are reserved for new coming stations, which are still in the initial state and try to join the IN (this state machine of station will be described later). After observing these request minislots, the IN will obtain the exact number of requests among these downstream stations. If there is any station waiting to transmit data, the IN will wait for the RMSs, which are indicated by the Grant Message from HC. Then IN randomly selects a RMS and puts the request information on it. The request message contains the number of requests that the IN substitutes for contending. After transmitting the request message, the IN would wait for the feedback from HC. The feedback message informs users the contention result. If a collision occurred, IN will back to the state and wait for another RMS Grant Message. Otherwise, the IN will receive the Data Minislots Grant message and then inform stations when to transmit data.

3.2 Station (STA) Protocol

For each station, there are two different states: convention state and agent state. The former is used for stations which have no IN as its agent. These stations must contend resources by themself. The latter state is used for stations which use an IN as their agent for contending.

When the station is powered on, it will listen to the downstream channel to acquire the RMS Grant Message which is sent by IN. If there is no Grant Message, it will enter into the convention state. Otherwise, it will randomly select a contention slot to join the IN. Since it is possible that more than one station want to join to an IN simultaneously, such contention must be solved. This is why the IN Request Minislot Grant Message allocates a number of contention minislots. After contending successfully, the station will enter into the agent state.

3.2.1 CONVENTION STATE

In this state, the operating steps are like the IN state machine. The difference is that station only contending by itself when it has data to transmit. As soon as the station requests successfully, it waits for the DMS Grant Message from HC to obtain the timing to transmit data and transmits data in the proper DMS. If piggyback is enabled, the station sets one flag in its packet when the buffer is not empty. The HC will reserved a DMS for the station to transmit in the next period without contending again.

3.2.2 AGENT STATE

After entering agent state, station performs three procedures: channel acquiring, timing acquiring and ranging from IN. After ranging, the station would acquire the RTC parameter. We note that it is different from the value of IN. And then the station registers to the HC. When it has data to transmit, it waits for the RMS Grant Message from IN and sends request in its unique RMS. After this step, the station waits for the DMS Grant Message from IN and transmits data in assigned DMS. The piggyback method in this state is the same as the convention state.

5. PERFORMANCE MEASUREMENT AND THE SIMULATION MODELS

5.1 PERFORMANCE MEASUREMENT

In our simulation, we compare the architecture with IN to the traditional architecture. In both architectures, we use ternary tree algorithm as the contention resolution mechanism. Assume the simulation time for each simulation run is *ST* minislots. The performances of the two architectures are evaluated in terms of the following metrics:

Average Buffer Delay = the average time (in minislots) from the packet arrives at the station to the time it is transmitted during the simulation time *ST*.

Throughput = the ratio of the minislots which is used for packet transmission to the simulation time *ST*.

Let **Unfairness Packets (UP)** denotes the number of packets which is transmitted faster than all the packets which arrive earlier than the transmitted packet in the network. Let TP denotes the number of transmitted packets during the simulation time *ST*. Therefore, the **Network Fairness (NF)** is defined as

$$NF = 1 - \frac{UP}{TP}.$$

A high performance network would require having a low average buffer delay, high throughput and high network fairness.

5.2 SIMULATION MODELS

The simulation is implemented by the *C* programming language on a *Pentium-based PC*. The packet arrival rate of each station is Poisson distribution [4] with a mean λ and the packet length is an exponential distribution [4] with a mean of *L*. Therefore, if the number of stations in the network is *N*, the *Network Load* (*NL*) for the network is given by

$$NL = \lambda \times L \times N .$$

In the first simulation model, we disable the piggyback function. We compare the two network architectures by the three metrics under different *NL* and round-trip propagation delay (in minislots). In the second model, we enable the piggyback function and compare the performance as before. In these two models, the simulation time *ST* is equal to 100000 minislots and the RMSs, which the HC first allocates for contention is 10 minislots. We randomly distributed the stations to 10 INs in new network architecture.

5.3 SIMULATION RESULTS

5.3.1 WITHOUT PIGGYBACK FUNCTION

Fig. 5 shows the obtained throughput of the two network architectures. The x-axis is the network load (*NL*) and the y-axis is the throughput. In this figure, we investigate the throughput under different propagation delay (delay) and the number of stations in the network (n). In traditional network, the throughput is obviously affected by the number of users in the network as well as the propagation delay. With the increase of users, the collision times will increase and many times of round-trip propagation are needed. As a result, the transmission resource is being wasted and the throughput is unacceptable. Contrarily, in the network with IN, only 10 INs substitute to contend the resource, the number of collisions is decreased and the network throughput is high. Although the propagation delay will affect the result, the effect is slight. We note that the obtained throughput by proposed network architecture with n=30 and 50 are almost the same when delay=0. As delay=20, the difference is still very small even when the *NL* is heavy. This is because that no matter how many active users in the network, only maximal 10 INs will contend the resource. As long as one of IN requests successfully, a group of users will utilize the resource immediately. In consequent, the network throughput will be better. On the contrary, the throughput is significantly affected by the number of users in the traditional network. As n=50 and delay=0, the throughput is degraded when *NL*=0.85. And as delay=20, the throughput is obviously only when the *NL*=0.65. From this figure, we conclude that the throughput of the proposed network architecture with IN performs much better than traditional architecture.

Fig. 6 shows the average buffer delay obtained by two network architectures. The x-axis is the *NL* and the y-axis is the average buffer delay. As before, we investigate how the propagation delay (delay) and the number of stations (n) in the network influence the average buffer delay. From this figure, we find that the average buffer delay increasing conspicuously in the traditional network. When delay=0, the average buffer delay of the traditional network architecture starts to increase extremely as *NL*=0.8. But the network architecture with IN starts to increase as *NL*=0.9 but the slope is moderate. Similar results will be obtained when delay=20 except that the difference is obvious between these two architectures.

The network fairness of the two network architectures is shown in *Fig. 7*. From this figure, we can see that the proposed network architecture with IN also performs better than the traditional network under different kinds of conditions. This is because that the number of contending users is decreased.

5.3.2 WITH PIGGYBACK FUNCTION

Fig. 8 shows the throughput of the two network architectures without piggyback function. We can see that the result is almost the same because that even when the *NL* is large, users only contend one times and transmit continuously. There is no difference in two network architectures. Moreover, the throughput will maintain very high when the *NL* is large. This implies that the proposed network architecture will perform well only when the users access network in the fashion of short and burst. We note that the obtained throughput of proposed network architecture is still slightly better than traditional network. *Fig. 9* shows the average buffer delay obtained by these two network architectures. The results of both network architectures with piggyback function are better than that of network without piggyback function. The slope is more moderate in two network architectures. We can see that the major influence is the propagation delay.

Fig.10 shows the network fairness of these two network architectures. The results obtained by both network architecture with piggyback function are also better than that of networks without piggyback function, but the difference is not apparent. The fairness derived by network with IN is still better than the traditional network.

6. CONCLUSIONS

In this paper, we proposed a new network architecture with Intelligent Node (IN) to improve the performance of the HFC network when the number of users is excess. The network architecture is very feasible to improve the traditional networks. We also designed the network protocols for the IN and stations. Simulation results shown that no matter what kind of conditions the network is, the performance of proposed network architecture is always better than that of the traditional HFC network.

REFERENCES

[1] IEEE Std 802.14 Draft3 R2 Media Access and Control, August 1998

[2] Cable Television Laboratory Inc. ,"Data-over-Cable Service Interface Specification: Radio Frequency Interface Specification (draft 2 revision 2, status: interim)," Oct.8, 1997

[3] Chatschik Bisdikian, "A Review of Random Access Algorithm," contribution to the IEEE 802.14 WG, no. IEEE 802.14 -96/019, January 1996

[4] D. Gross and C.M. Harris, "Fundamentals of Queueing Theory," Second Edition, Wiley Series in Probability and Mathematical Statistics, 1985.

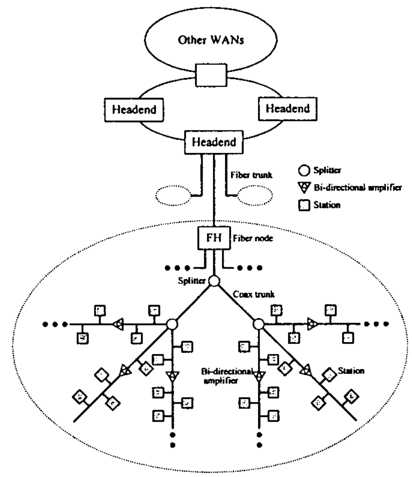

Figure 1. Traditional HFC network.

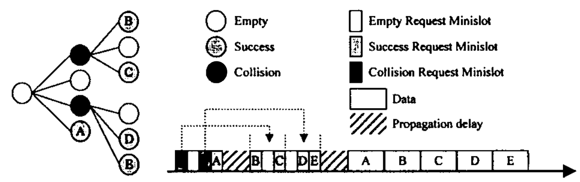

Figure 2. The n-ary tree algorithm (n=3).

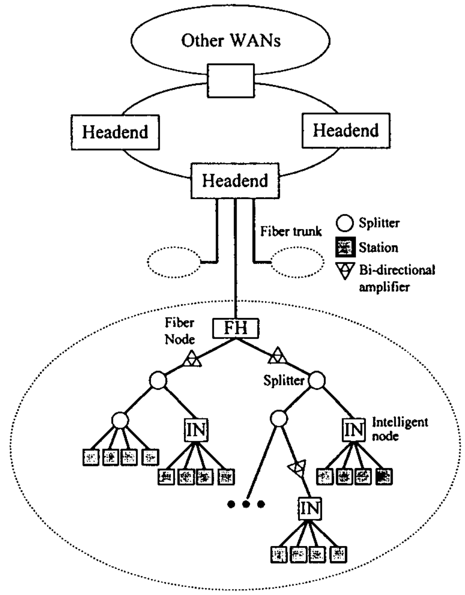

Figure 3. Using INs in the traditional HFC network

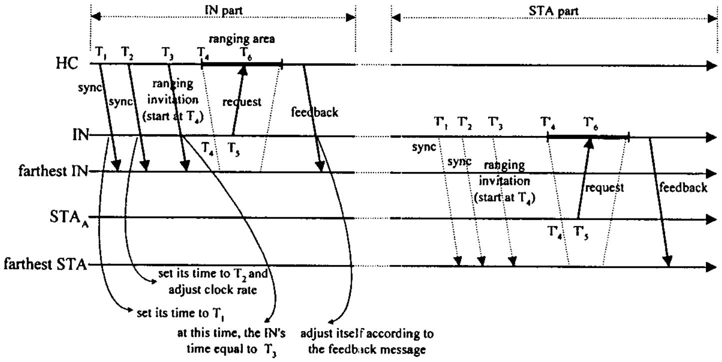

Figure 4. Timing acquiring and Ranging.

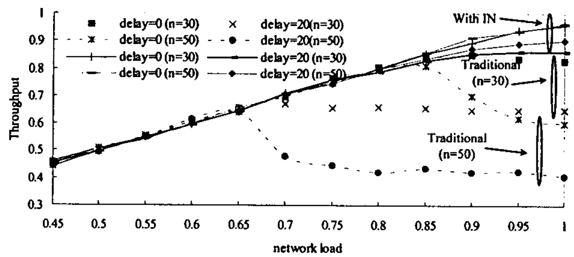

Figure 5. Comparisons of the throughput obtained by traditional and proposed architectures without piggyback.

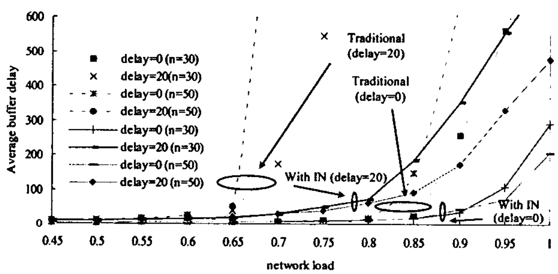

Figure 6. Comparisons of the average buffer delay obtained by traditional and proposed architecture without piggyback.

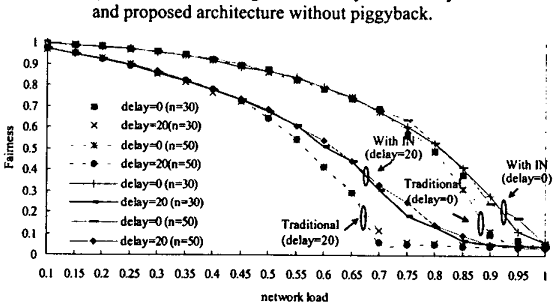

Figure 7. Comparisons of the fairness obtained by traditional and proposed architecture without piggyback.

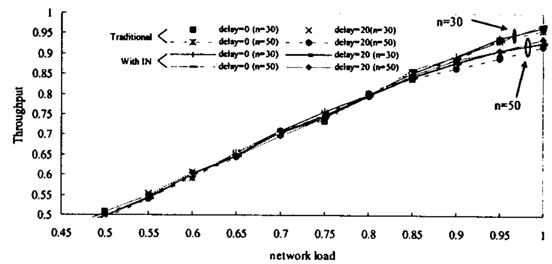

Figure 8. Comparisons of the throughput obtained by traditional and proposed architectures with piggyback.

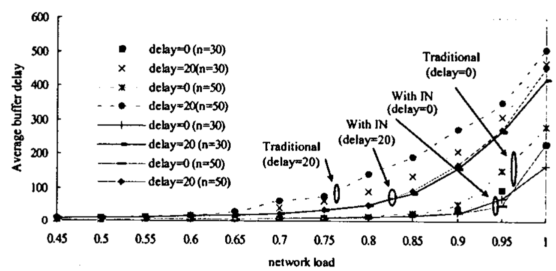

Figure 9. Comparisons of the average buffer delay obtained by traditional and proposed architecture with piggyback.

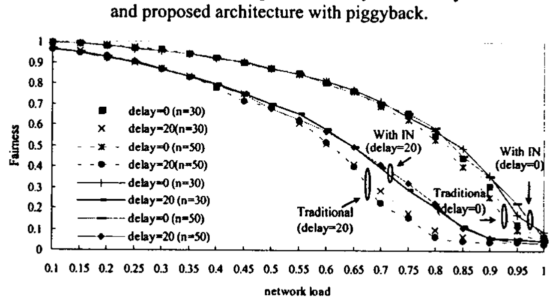

Figure 10. Comparisons of the fairness obtained by traditional and proposed architecture with piggyback.