

Homenet: A Broadband Voice/Data/Video Network on CATV Systems

By M. HATAMIAN and E. G. BOWEN*

(Manuscript received April 3, 1984)

Homenet is a broadband distributed communication system that supports data, real-time digitized voice, and analog video on a single cable in a CATV type of network. The distance limitation problem encountered in local area networking schemes is eliminated by dividing the large CATV net into smaller "homenets." This feature makes the network suitable for a large number of users located in a relatively wide geographic scope. This paper describes the implementation of a small experimental version of this system in hardware. More attention is given to the protocol processing hardware, which implements a protocol based on collision detection called Movable Slot Time Division Multiplexing (MSTDM). The MSTDM protocol guarantees the continuity of voice signals received at the user station. Problems such as clock synchronization and confusion of data and voice packets are addressed, and solutions are given. Presently, an experimental network composed of five user nodes in two different frequency nets is operational. An interactive video retrieval service implemented in the network is described as an example of the type of user services (other than data/voice/one-way video) that can be offered at the main head end of the system.

I. INTRODUCTION

The concept of Local Area Networks (LANs) is a well-developed one in the field of computing and data communication.¹ These networks are used for sharing computing resources, and for communicat-

* Authors are employees of AT&T Bell Laboratories.

Copyright © 1985 AT&T. Photo reproduction for noncommercial use is permitted without payment of royalty provided that each reproduction is done without alteration and that the Journal reference and copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free by computer-based and other information-service systems without further permission. Permission to reproduce or republish any other portion of this paper must be obtained from the Editor.

ing data among a number of users in a limited geographic scope. Adding voice and video capability to these networks makes them very attractive for the office information systems of the future. Furthermore, if the distance limitation and the constraint of limited geographic scope are removed, then such networks become potential candidates for the home information systems of the future, provided that the cost of user's terminal equipment is minimal. Such networks should more appropriately be called Metropolitan Area Networks (MANs) rather than LANs. Solving the distance limitation problem can also increase the attractiveness of the office information systems; office branches located at distant locations can become part of the network and can share information.

Homenet is a broadband data/voice/video communication system, first proposed in Ref. 1, which satisfies all the above requirements for MANs. The system combines frequency and time multiplexing, and supports the communication of data, real-time digitized voice, and analog TV signals on a single cable in a cable TV (CATV) type of network. This paper describes the hardware implementation of the homenet and some of its features. Presently, a fully working testbed composed of five user stations connected in two nets is in operation.

Sections II and III describe the homenet and the communication protocol. Section IV gives a detailed description of the hardware implementation of the system and its features. More attention is given to the protocol processing hardware, which is essentially the intelligent node of a distributed packet switching network.

II. WHAT IS A HOMENET?

Homenet is a broadband communication system based on a combination of frequency and time multiplexing, and distributed packet switching techniques. The system supports the communication of data, digitized voice, and one-way analog TV signals on a cable in a CATV type of network. Since it is basically a distributed switching network, all the switching functions are performed at the user's terminal equipment and there is no central switching involved. Reference 2 describes the system in detail.

A relatively large community of users is divided into small geographic regions and each region, called a homenet (or a net for short), is assigned a 6-MHz frequency band. Users within each frequency band can receive signals from any other net by tuning their receivers to the frequency of that net. This tuning is performed automatically by a signaling scheme. All users transmit their data and digitized voice signals on a single transmit frequency, F_0 , which is then translated to

its homenet frequency and propagated throughout the network in such a way that all the users in all nets are able to receive it. Before data are transmitted, users in each frequency band have to contend for the channel using the protocol described in the next section of this paper.

The overall operation of the homenet is shown in Fig. 1. Suppose that user No. N in net 3 wants to transmit a packet to user No. 1 in net 2. First, user No. N contends for the channel in net 3 and once access to the channel is gained, it transmits its data on frequency F_0 to the nominal head end of net 3 (H_3). At this point, frequency F_0 is translated to two different frequency bands, F_3 and F_{r3} . Frequency F_3 is transmitted downstream to the users in net 3 and all the nets following net 3 (in this example there are no nets following net 3). Frequency F_{r3} is transmitted upstream to the main head end of the network (H_1) at net 1, where it is translated back to F_3 and sent to the users in net 1 and net 2. Now user No. 1 in net 2, with its tuner listening to frequency F_3 , can receive the packet by demodulating the signal from F_3 to baseband and searching for its address in the address area of the packet. Other than the frequency translation operations, each nominal head end H_i is equipped with two notch filters—one for frequency band F_i , which stops the signal translated from F_{r_i} to F_i at the head end; and one for F_0 , which allows all the nets to use the same transmit frequency F_0 without interfering with their adjacent nets.

The above scheme is certainly not the only possible way of using

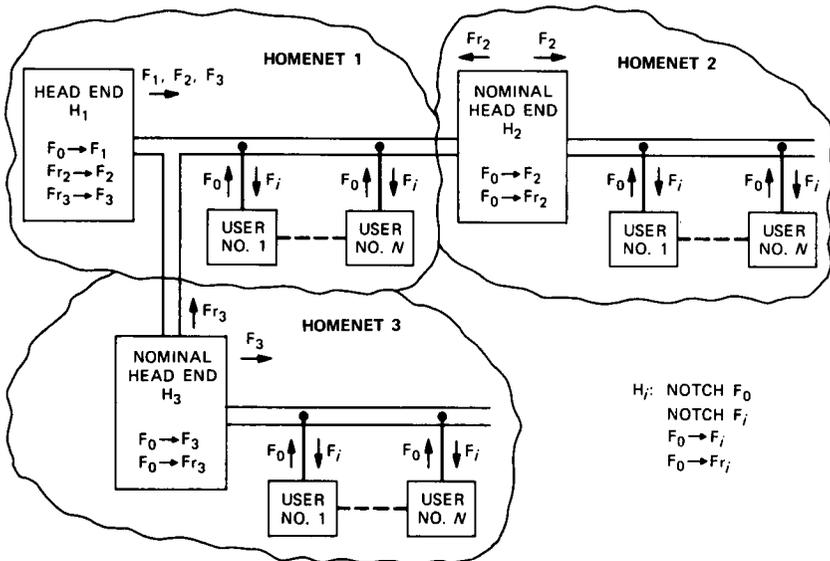


Fig. 1—Frequency assignment in homenet.

the frequency bands;³ however, it is the one that requires the least amount of hardware for each user station and also is compatible with currently installed midsplit CATV networks.

Establishing a communication network like homenet for a large community of users of the size accommodated by homenet would not be possible by direct extension of LAN techniques (i.e., increasing the cable length and the bit rate). One of the major limitations of the local area networks is the cable length constraint, which forces the users to be located close to each other to prevent large propagation delays. *Ethernet*,* one of the best-known local area networks,⁴ is limited to a cable length of about 2.5 kilometers operating at 10 Mb/s; increasing the bit rate or the cable length results in an appreciable reduction in the efficiency of the system. In homenet this distance limitation problem is solved by using different frequency bands and grouping the users that are located close together into one frequency net.⁵ This way, the users in each net have to contend for transmission rights only among themselves, and do not have to worry about transmitters in other frequency nets; they can still listen to all other frequency nets, so a complete connection between all users in all nets exists. This broadband technique increases the size of the network (i.e., the length of the cable) and at the same time reduces propagation delays, which are very important, especially for access strategies that use collision detection schemes such as CSMA/CD. Each net can operate at low bit rates but the total throughput of the network can go up to several hundred Mb/s, depending on the number of frequency bands used in the system.

III. COMMUNICATION PROTOCOL

The communication protocol used in each net of the homenet system is called Movable Slot Time Division Multiplexing (MSTDM)⁶—a variation of the CSMA/CD technique used in *Ethernet*. This protocol guarantees the continuity of the voice signals received at each user station, a task that no other currently available protocol based on collision detection can handle. This protocol is described in the following section.

Integration of packetized data and voice in a local area network requires an upper limit on the voice packet delays to ensure that the voice receiver does not run out of samples before new voice samples arrive. This requirement in turn guarantees a glitch-free, continuous speech signal at the output of the voice receiver. None of the currently available protocols satisfy the above requirement and hence are not

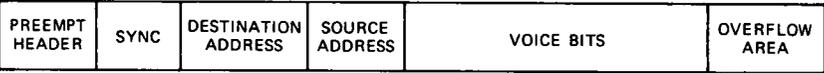
* Trademark of Xerox Corporation.

suitable for integration of data and digitized voice. MSTDM protocol places an upper bound on the voice packet delays; it guarantees the continuity of the reconstructed voice signal and also guarantees that once access to the channel is gained, no two voice sources can collide. MSTDM takes advantage of the periodicity of the voice packets; it also requires that the size of the data packets be smaller than the voice packets. A detailed treatment of this protocol can be found in Ref. 6. A description of its operation is given below.

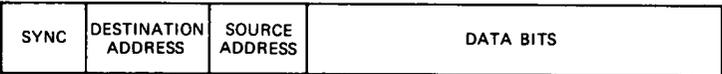
In MSTDM a distinction is drawn between the first packet from a voice source and all the following voice packets from that source (called the secondary voice packets in this paper). The first voice packet and the data packets are treated the same way as in CSMA/CD. They check the channel busy signal to monitor the status of the channel, and once the channel is idle, they start transmitting. They listen to the channel while it is transmitting to make sure that no collision occurs, and if there is a collision, then the colliding sources stop their transmission and try to access the channel after a period of time defined by a retry strategy. This procedure stays the same for all data packets; however, for the voice sources the procedure is different. Once the first packet of voice successfully acquires the channel, then the following packets from that voice source get transmitted (when they are ready for transmission) as soon as the channel becomes idle; they do not listen to the channel for collision during transmission. If a collision occurs between these secondary voice packets and any other packet, the other transmitter is forced to stop its transmission and the secondary voice packets override.

Figures 2a and b show the voice and data packet formats, respectively. When there is a collision between a secondary voice packet and a data packet, the preempt portion of the voice packet, which does not contain any information, allows enough time for the data source to detect collision and stop its transmitter before the sync bits from the secondary voice packet appear on the channel.

After a voice source transmits a packet, it schedules its next trans-



(a)



(b)

Fig. 2—Packet formats for (a) voice packet, and (b) data packet.

mission for T seconds later. Assuming the rather unlikely situation where all the packets from a particular voice source find the channel idle when they are ready to be transmitted, then, for that particular voice source, the channel looks exactly like a TDM system with reserved time slots that are T seconds apart. However, it is quite likely that when the voice source is ready for its next transmission, a data source or another voice source is in the middle of transmitting its packet. In this case the voice transmitter has to wait for the channel to become idle. Obviously, in this situation the voice packet will be delayed and this delay causes the time slot for the voice source to move back in time. Therefore, for the voice sources the system looks like a TDM channel in which the time slots are not fixed in time and are free to move; hence the name movable slot TDM.

Reference 6 proves that if the voice packet delay is less than a packet transmission time (the upper bound on the voice packet delay in MSTDM to which we previously referred), then voice sources will never collide and no voice samples will be lost. To satisfy this requirement data packets are constrained to be shorter than voice packets.

The voice samples arriving during the voice packet delay time are stored in the overflow portion of the voice buffer (see Fig. 2a), and are transmitted along with the rest of the packet when the channel becomes available. The overflow area is always transmitted even if it does not contain any voice samples (i.e., the case when the voice packet is not delayed). This contributes to the proof of the fact that the voice sources never collide in MSTDM once they successfully transmit their first packet.² Obviously the overflow area of the voice buffer should be long enough to accommodate the voice samples that arrive during the voice packet delay, which is less than a packet transmission time. Since the transmission clock rate is much higher than the voice sampling rate, the size of the overflow area need not be larger than a few bits.

IV. HARDWARE IMPLEMENTATION

A small version of the homenet network described in previous sections has been built-in hardware for experimental purposes and feasibility studies. We currently have a fully working testbed composed of five user stations running in two frequency nets. Except for the protocol processing hardware, which is the most important part of a user station, all the components used in the system (such as frequency translators, taps, splitters, channel selectors, cable, etc.) are similar to those used by the CATV industry.

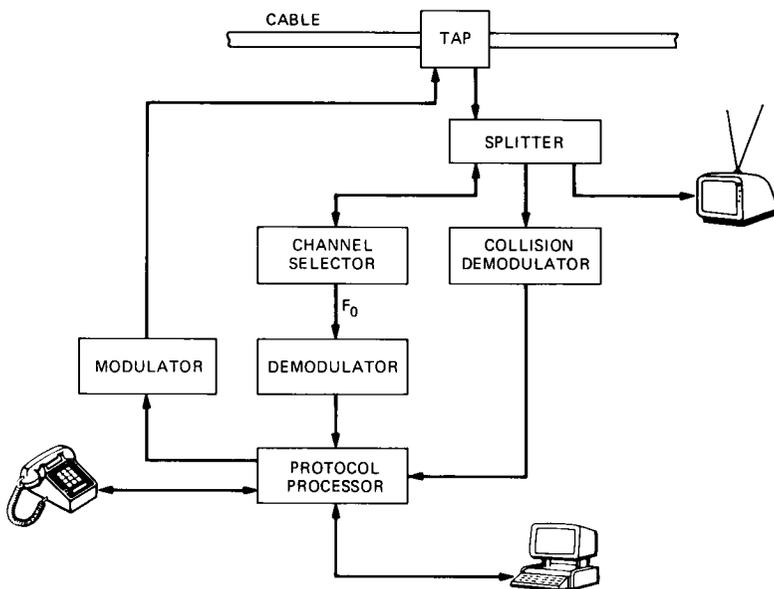


Fig. 3—Block diagram of user node hardware.

4.1 User node

Each user in homenet requires some hardware to interface his or her equipment to the communication cable. The user-node hardware can be thought of as a black box with one end connected to a cable and the other end to the user's voice source (normally a phone set), data source, and TV set. Figure 3 shows various components of the user-node hardware. The signal picked up from the cable through the tap is split and distributed in three ways. One line is connected directly to a TV set, another line feeds a collision demodulator whose function will be described later in this paper, and the third line is connected to a channel selector. A signaling scheme controls the channel selector and sets it to the frequency of the net that the user wants to hear. This frequency band is converted to a common Intermediate Frequency (IF) band F_0 and fed to a demodulator, which translates the information contained in band F_0 to a digital bit stream. This bit stream is then processed by the protocol processor and, if the information is destined for the user's address, it will be appropriately sent to either the data or the voice section. For the purpose of transmission, once the channel is accessed, the packets prepared by the protocol processor are modulated to the frequency band F_0 using a modulator and sent to the user's nominal head end (through the tap and over the

cable) to be distributed throughout the network, as we described in Section II.

Except for the protocol processor, which is special-purpose hardware, the rest of the user node components are commercially available items. The modulators and demodulators are tuned to a center frequency of 43.4 MHz (F_0 in current system).

4.2 Protocol processor

The protocol processor is essentially the intelligent part of the user node hardware. It is responsible for digitization and packetization of voice signals, packetization of data, and most importantly, implementation of the MSTDM protocol.

The processor is divided into two main sections, transmitter and receiver. Each section has two separate circuits, one for voice and another for data. Following is a detailed description of the operation of these circuits.

4.2.1 Transmitter

The voice and data sections of the protocol processor's transmitter operate independently of each other, with their own dedicated buffers. In terms of the ordering of the sync and address fields, the packet formats are fixed and are as shown in Fig. 2. However, the position of the sync word, the number of sync words, the length of the voice preempt header, and the length of the packet can be arbitrarily set by the user to conform to a net standard. The packet length can be set to any number of bits fewer than 4096.

4.2.1.1 Voice. Before we describe the operation of the voice transmitter, we should discuss the structure of the voice buffer. As we mentioned before, the first voice packet is treated the same as data packets in terms of accessing the channel. The moment that the voice signal is activated, the voice transmitter starts filling the voice buffers and at the same time makes a request for transmission. When transmission right is granted, the number of collected voice samples may not be enough to fill the whole buffer, and as a result a number of empty locations (noise bits) will remain at the end of the buffer. Now, if the buffer is transmitted from beginning to end, then the empty area will cause a quiet interval (or a glitch) in the voice signal between the first and the second voice packets. In applications where a Time Assignment Speech Interpolation (TASI) mode of operation is desired, this effect can cause serious distortion in the reconstructed speech signal. However, if the empty portion of the first voice packet is transmitted before the actual voice bits, then the quiet interval will not be in the middle of the voice signal and will not create any difficulty. This procedure is illustrated in Fig. 4. The buffer is shown in this figure as

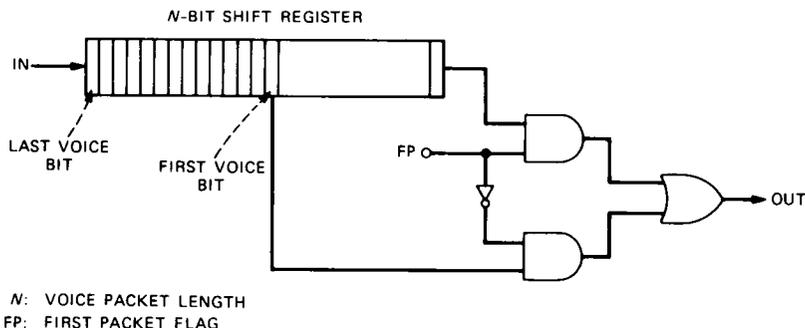


Fig. 4—Voice buffer operation.

a long First-In First-Out (FIFO) shift register; a First Packet flag signal (FP in Fig. 4) indicates whether the bits would be read out from the end of the FIFO register or from its head position (i.e., the first voice bit). In Very Large-Scale Integration (VLSI) design, the implementation of such a buffer is rather simple considering the regularity of its structure. In Transistor-Transistor Logic (TTL) design, we used RAMs as buffers, and counters and latch registers to keep track of the first voice bit position, last voice bit position, and the length of the unused portion of the buffer.

Figure 5 is a block diagram of the voice transmitter. To be able to handle the voice signal in real time, two buffers operating in ping-pong mode are needed; one buffer is being filled with voice bits while the other is being transmitted. The combination of RAM and counter blocks in Fig. 5 represent a FIFO register which, in conjunction with the first packet handler circuit, operates in a manner described above. The input voice signal from the voice source is digitized and converted to a serial bit stream by a 64-kb/s μ -law codec chip (8-kHz sampling rate, 8 b/sample). A multiplexer at the input of the ping-pong buffer controls the distribution of the input bits, clock signals (voice clock and transmission clock), and memory write pulses (R/W) to the buffers. The buffer that is supposed to be transmitted receives the transmission clock and no write pulses; the buffer that is being filled with the voice bits receives the voice clock (64 kHz), the voice bits, and the memory write pulses.

Before trying to establish a voice connection, the Timing and Control Circuit (TCC) first clears the buffers and then writes the header information (i.e., sync, destination and source addresses) into both ping and pong buffers through the input multiplexer. At the time the first packet of voice begins to be formed in the ping buffer, a request for transmission is made by TCC to the Transmit Request Circuit (XRC). Once XRC receives the request, it starts monitoring

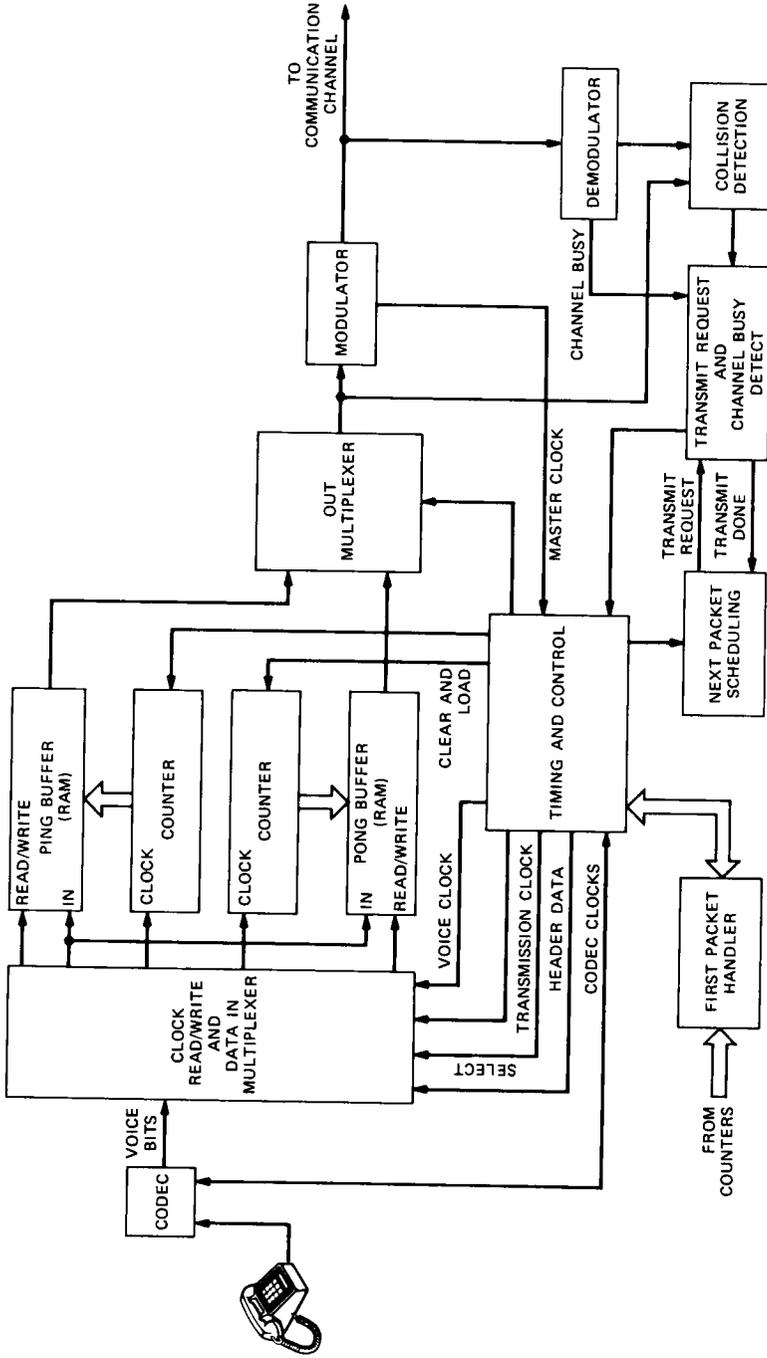


Fig. 5—Block diagram of the voice transmitter.

the channel busy signal provided by the demodulator, and the moment the channel becomes idle XRC sends an acknowledgment to TCC. Upon receiving this acknowledgment, TCC switches the ping-pong mode (by changing the multiplexer control signals) and sets a transmission flip-flop. Now the pong buffer starts receiving the voice bits and the ping buffer, which contains the first packet of voice, begins to be transmitted on the cable under the control of the first packet handler circuit, through the output multiplexer and the Frequency Shift Keying (FSK) modulator. If no collision is reported by the Collision Detection Circuit (CDC), then the transmission continues until the last bit of the voice packet is sent, at which time TCC clears the transmit flip-flop and sends a signal to the Next Packet Scheduling (NPS) circuit. NPS schedules a transmission request for T seconds later. T is switch selectable and is set by the user, depending on the length of the voice packet. During this T seconds, voice samples are being stored in the pong buffer, and the ping buffer, which has already transmitted its data, is waiting to be switched. After T seconds, NPS sends a transmission request to XRC and when transmission right is granted, TCC again sets the transmit flip-flop and switches the ping-pong mode. Now, the input voice samples are routed to the ping buffer and the secondary voice packet contained in the pong buffer begins its transmission. The first packet handler and the collision detection circuits are disabled at this time, because the first packet has been successfully transmitted and there is no need to check for collisions (see the description of MSTDM protocol in Section III). When the second packet is transmitted, NPS receives a transmit done signal and again schedules a request for T seconds later, and the procedure cycles until the user decides to end the voice connection.

If, during the transmission of the first packet, a collision is detected by CDC, then the transmit flip-flop is cleared, the buffer pointer is reset to the beginning, and TCC tries to make a request again. At present, the only retry strategy built into the protocol processor is on the basis of a random retry. The operation of the collision detection circuit is described in a separate section following the discussion of the data transmitter.

4.2.1.2 Data. Figure 6 is a block diagram of the data transmitter. This circuit is considerably less complex than the voice transmitter. It does not require the ping-pong buffering mode and there is no difference between the way the first packet and the following ones are handled; this eliminates the need for the first packet handler circuit.

Before establishing a data connection, the TCC writes the header information into the buffer, and then it sets the transmitter ready flip-flop. Now the transmitter buffer is ready to accept data bits. The data are written in the buffer and once the packet is formed and the

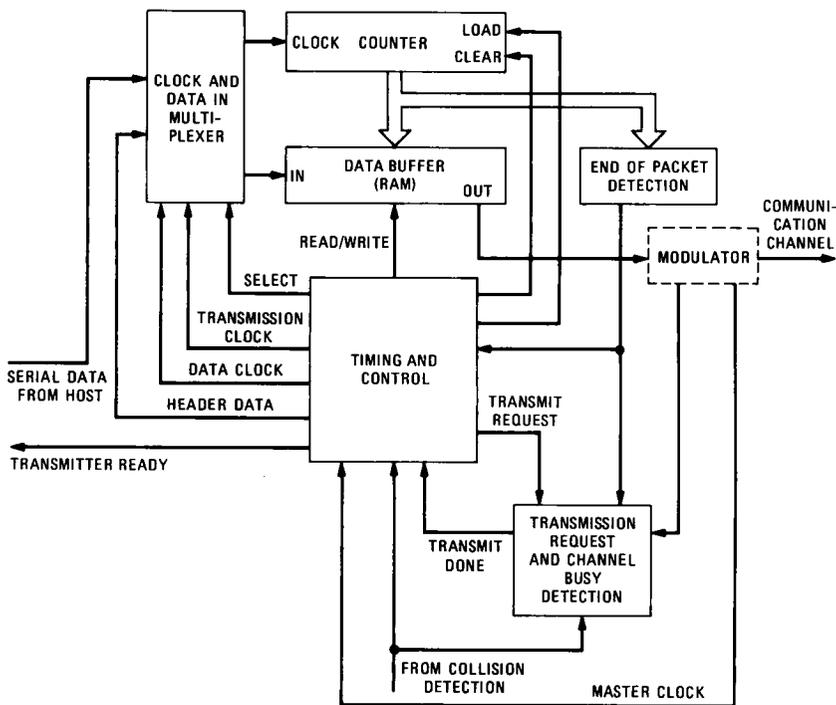


Fig. 6—Block diagram of the data transmitter.

buffer is full, the End of Packet Detection (EPD) circuit sends a signal to TCC, which resets the transmitter ready flip-flop. The transmitter is then ready to access the channel and data sources should not try to send any new data for packetization. At the same time, TCC sends a transmission request signal to the XRC. When the channel becomes idle, XRC sets a transmit flip-flop and the transmission of the packet begins. Once the last bit of the packet is transmitted, XRC generates a transmit done signal, which sets the transmitter ready flip-flop; the buffer then becomes available for packetizing data bits, and the cycle starts again.

In case of a collision with another packet, the same procedure used for the first packet of voice is used. The collision detection circuit is shared between the voice and data transmitter.

4.2.1.3 Collision detection. In homenet, collisions can be detected in the digital domain by comparing every single bit of a packet before and after its transmission. Relying on just the amplitude of the signal on the line at each transmitter for detecting interference from other transmitters can result in missing collisions that are caused by transmitters that are distant from each other. In homenet, however, all the

transmitters in each net, say net H_i , send their signal, on frequency F_0 , to the nominal head end. There it is translated to frequency band F_i and returned to all sites. By inserting proper attenuators on the transmitters, we arrange to have the same amplitude for all the signals received at the nominal head end from different transmitters. As a result, the signals that are returned in frequency F_i at each site have the same relative amplitude, and there is no chance of missing collisions between distant transmitters.

The collision detection circuit has a dedicated demodulator that is always listening to the frequency of the net that the transmitter is in (i.e., F_i for net H_i). This demodulator is referred to as the collision demodulator in the block diagram of Fig. 3. Notice that detecting collision in frequency F_0 rather than F_i does not have any advantage over baseband, and would not solve the amplitude problem.

It is obvious that the process of modulating the bits to frequency F_0 , transmitting to the nominal head end, translating to frequency band F_i , and transmitting them back will introduce some location-dependent delay between the bits that leave the transmitter and the ones that are received by the collision demodulator. The collision detection circuit corrects for this delay by inserting an equal delay on the transmitted bits before they are compared. This is accomplished by an adjustable delay line on one input of the CDC, which is adjusted only once depending on the user's distance from the nominal head end.

4.2.2 Receiver

Much like the transmitter, the receiver is also divided into two almost independent sections, voice and data. In the transmitter, the collision detection circuit was shared between the voice and data sections. In the receiver, there is no need for collision detection; however, there is one circuit that is shared between the two sections, and that is the Sync and Address Detection (SAD) circuit.

4.2.2.1 Voice. Figure 7 shows the block diagram of the voice receiver. The operation of the ping and pong buffers in the receiver is similar to the voice transmitter. When one is receiving the input bit stream at the channel rate, the other is playing the previously received voice packet into a codec at the voice bit rate (64 kb/s). The switching of the ping-pong mode is done by the TCC. The input bit stream is first demodulated from band F_i to baseband and directly fed to the input of the buffers, the SAD, and the end of packet detection circuit. Right after the transition of the channel busy signal from an idle to a busy state, SAD starts looking for a sync word. The sync words for data and voice differ in their most significant bit. If the detected sync is a voice sync, then the receiver starts looking for either another sync

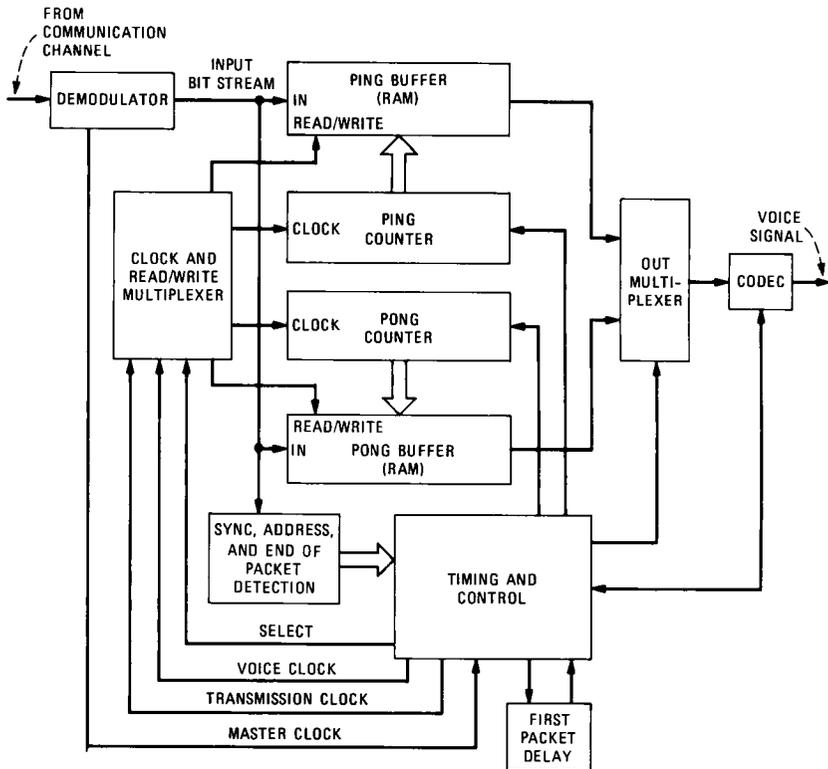


Fig. 7—Block diagram of the voice receiver.

word or its address. The sync words can be repeated consecutively in the header area of the packet as many times as desired. The destination address should always be immediately after a sync word.

If the sync word indicates a voice packet and the destination address field of the packet is matched with the receiver's address, then SAD first stores the following field (i.e., the source field) as the address of the source, and sends a signal to TCC indicating the beginning of the voice bits. TCC then sets a receive flip-flop and applies the transmission clock to the ping buffer. The voice bits begin to be stored in this buffer until an end of packet is detected by SAD, at which time TCC resets the receive flip-flop and switches the ping-pong mode. The end of packet detection is not done based on just the packet length. The voice transmitter always transmits a fixed number of bits for each packet, as required by MSTDM protocol. However, not all of these bits are actual voice bits; the bits in the overflow area may not be useful information. Therefore the end of the voice samples in the packet must be marked. This is done at the transmitter by placing a

flag byte at the end of the voice samples with all eight bits set to "1." When the transmitter's ping-pong mode is switched, the value of the counter for the buffer that was receiving the voice bits is saved; when the transmission of this buffer begins, the control circuit monitors the buffer counter and when it is equal to the saved value, TCC forces the output bits to a high state until the buffer counters reach the packet length. Therefore the unused portion of the voice packet is transmitted as "all ones." At the receiver, SAD circuit searches for an "all one" flag byte and sends an end of packet signal to TCC, as mentioned above. The codec used for digitizing the voice signal does not use the "all one" level in its code. Therefore, no voice samples will be coded as all ones to cause a false end of packet detection at the receiver.

Once the end of packet is detected and the ping-pong mode is switched by TCC, the voice samples stored in, say, the ping buffer are played back to the codec to reconstruct the voice signal. The pong buffer is idle at this time and is waiting to receive the next voice packet, which will arrive sometime during the playback process.

In MSTDM protocol, the voice packets can exercise a bounded delay less than one packet transmission time. This delay may cause the receiver buffer to run out of voice samples before the next packet arrives, and may also cause a distortion in the speech signal. To alleviate this problem, a small delay is inserted before the beginning of the playback process for the first packet of voice (only the first packet). This task is accomplished by the First Packet Delay (FPD) circuit at the receiver (see Fig. 7). After the first packet is played back, TCC deactivates this circuit. The FPD circuit is also used for recovering from a distortion problem created by timing discrepancies between user nodes. This will be described in detail in a later section on clock synchronization.

4.2.2.2 Data. Figure 8 shows a block diagram of the data receiver. This circuit is the simplest section of the protocol processor. The sync and address detection circuit is shared between the voice and data sections. When a data sync pattern followed by the receiver's address is detected in the input bit stream, TCC first resets a data ready flip-flop, indicating that the receiver buffer is being filled with the incoming packet bits. The receive clock is then applied to the buffer counter and data bits are stored in the buffer until an end of packet signal is generated by the EPD circuit. The end of packet is simply detected by comparing the value of the buffer counter with the length of the data packet. This signal sets the data ready flip-flop, indicating that the receiver buffer contains valid data bits and is ready to transfer those to the host system. At this time, if the host ready signal is high, TCC applies the data clock to the buffer counter and the data bits are transferred to the host system. Once this transfer is made, the circuit

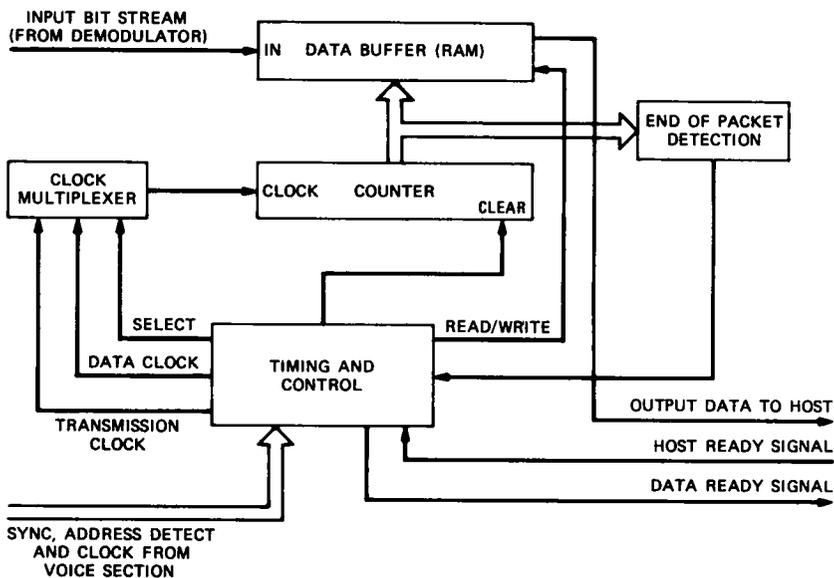


Fig. 8—Block diagram of the data receiver.

is reset and TCC waits for the next data sync detect signal, at which time the cycle starts again.

The performance of both the data receiver and transmitter can be somewhat improved using a ping-pong buffering scheme as in the voice section. However, since this is not a necessity, we did not choose to implement it in our experimental system. This option can always be easily incorporated into the system if needed.

4.2.3 Clock synchronization

When a communication link, either data or voice, is established between two sites, it is obvious that for proper operation the receiver's clock pulses should be synchronized with the incoming data (i.e., synchronized with the transmitter's clock). In homenet, each user station (user-node hardware) has its own crystal oscillator generating a 16-MHz master clock signal. All the clock pulses used by the protocol processor are derived from this master clock, and are synchronized with the incoming data using the transitions of the input bit stream. The synchronization circuit, which is part of the demodulator board, uses a very simple digital technique similar to the clock recovery circuits used with nonreturn to zero data streams.

For the purpose of synchronization, the receiver requires at least two or three transitions in the incoming bit stream before any useful data can be picked up from the line. For the voice packets the required

transitions can be placed in the preempt portion of the packet; for the data packets a 4-bit preempt header is added to the beginning of the packet format shown in Fig. 2b.

Due to different operating conditions the frequency of the crystal oscillators at two ends of a communication link can be slightly different, thus creating a minor timing discrepancy. This small frequency difference does not create any difficulty in receiving the information bits because the bit values are read into the receiver buffer in the middle of clock pulses, and a small drift can be tolerated. However, due to the periodic nature of the voice sources and their real-time requirement, the timing discrepancy affects the voice section of the protocol processor in two ways, as described below.

First, we consider the effect of timing discrepancy on the next packet scheduling time. The situation is illustrated in the timing diagram shown in Fig. 9 for two arbitrary voice sources that have successfully transmitted their first packet and reserved a movable time slot on the channel. Voice source No. 1 schedules its next transmission for T seconds after it transmits the current packet (i.e., T seconds after the falling edge of TRANSMIT signal in Fig. 9). After T seconds, the NPS circuit generates a TRANSMIT REQUEST pulse and the voice source is guaranteed to have access to the channel within δ seconds, where δ is between zero and a maximum of one packet transmission time. Voice source No. 2 operates in exactly the same way except that, owing to the slight timing discrepancies between the two sources, the next packet scheduling time for this source will be $T + \epsilon$ rather than T . This can cause the TRANSMIT REQUEST pulse and the time slot for voice source No. 2 to drift very slowly in time with respect to voice source No. 1 (see Fig. 9). The drifting continues until the time slots for both sources are adjacent to each

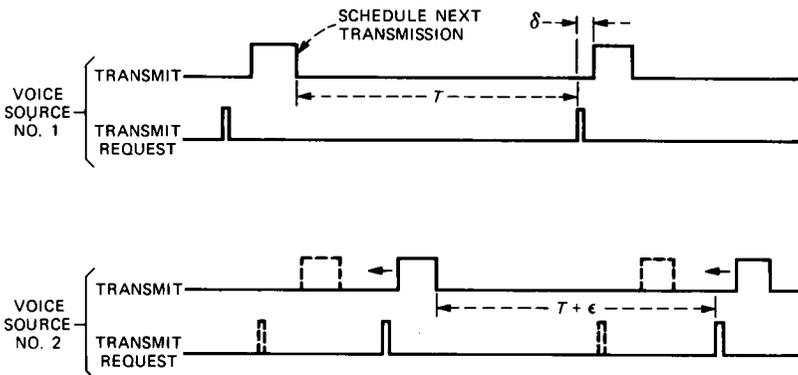


Fig. 9—Effect of timing discrepancy on next packet scheduling.

other. If there are a number of voice sources on the line, all the time slots gradually move until they are adjacent to each other. Notice that this gradual movement of the slots is different from the movement dictated by MSTDM protocol, which can vary depending on the data traffic. If the data traffic is light and the voice sources continue to stay on the channel for a long time, then, from the above discussion, all the time slots will eventually be packed next to each other. The timing discrepancy in this case does not introduce any difficulty; as a matter of fact, it creates a favorable situation.

With respect to the rate at which the voice samples are generated at the source and used at the destination, the timing discrepancy can create an undesirable situation. If the receiver clock is slower than the transmitter, then there will be a time when both ping and pong buffers at the receiver will be full when a new packet arrives. If the receiver clock is faster than the transmitter, then the receiver will eventually run out of samples before a new packet arrives. These situations cause a distortion in the voice signal, which cannot be recovered from for some time. In the receiver hardware, both of the above situations cause the ping-pong switching command to be generated at the same time with transmission. To solve the problem, when such a condition is detected, TCC activates the first packet delay circuit (see Fig. 7), which inserts a delay in the playback of only the next packet and causes the ping-pong switching command not to overlap with the transmission. The consequence is that the current voice packet is lost, but the receiver goes back into a normal undistorted operation mode. In our system, which uses a conventional oscillator and crystal type, this situation occurs about every 20 minutes. In other words, if a conversation lasts for a long time, then every 20 minutes one packet of voice (i.e., a few milliseconds of voice signal) will be lost. There are more expensive solutions that totally eliminate the problem; however, losing a few milliseconds out of 20 minutes (12×10^5 ms) is of no significance at all, and more expensive solutions are not justified. Besides, using better crystal types can increase the 20-minute period to well over an hour.

4.2.4 Packet confusion

In the case of collision between a secondary voice packet and a data packet, a confusion between packets can occur at the receiver. Consider the following situation: a data source is transmitting when it detects a collision with a voice packet after the sync word has been transmitted (i.e., in the address area of the packet). The data transmitter stops, the voice transmitter continues with its transmission, and the line will look something like the following:

DATA SYNC	GARBAGE	VOICE SYNC	VOICE DESTINATION ADDRESS	VOICE SOURCE ADDRESS	VOICE BITS
-----------	---------	------------	---------------------------	----------------------	------------

In the above display the collision occurred in the garbage area, which includes the voice preempt header as well.

Let us consider what happens at the receivers. All the receivers detect the data sync and start looking for the destination address. One particular receiver whose address is equal to the first byte of the garbage area is going to detect a match, and it will erroneously start storing the following bits as a data packet. Obviously this situation is very undesirable. We can solve the problem by looking at the length of the received packet. A false data packet like the above will certainly be longer than a normal data packet. If, at the time that the end of packet pulse is detected (based on the packet length as described before) the receiver is still busy with the incoming bit stream, then the received data packet must be a false one and should therefore be discarded.

The receiver that is supposed to receive the voice packet first synchronizes on the data sync; however, since it does not find its address following the data sync, it resets itself and starts searching for a combination of sync and destination address again. It should be noted that this problem occurs during the preempt header so that the data transmitter stops and the voice sync and address are undisturbed. Therefore, we do not have to worry about the voice packet; it will get to where it is supposed to. Of course, in the highly unlikely situation of the voice destination address being exactly the same as what is found in the garbage area, the above assumption will not be true and the voice packet will be lost.

4.3 Head end

Aside from the frequency translation and filtering operations, a variety of user services can be incorporated in the homenet's main head end (head end H_1 in Fig. 1). Our experimental system presently supports two services, an interactive video disc and a service for establishing voice links with sources outside the network. Figure 10 shows the block diagram of the main head end. A user station identical to the ones used at the user nodes is dedicated to this head end.

A head-end processor, which, depending on the application and the network requirements, can be anything from a small microprocessor to a large computer system, controls all the services. A user can control a video disc located at the head end by sending commands to the head-end processor over the cable using its data transmitter. These com-

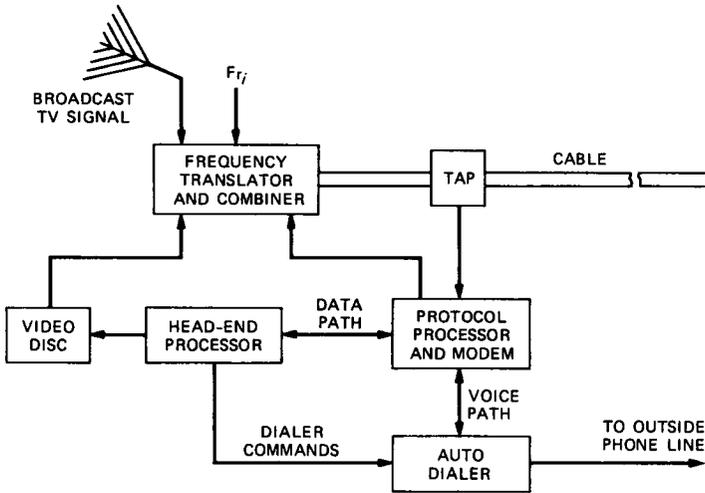


Fig. 10—Block diagram of homenet's main head end.

mands are interpreted by the head-end processor and the proper signals are sent to the video disc through a serial port.

To establish a voice link with sources outside the network, a user sends a data packet to the head-end processor giving the number to be called; the head-end processor then sends the proper commands to an autodialing system, which connects the voice path of the protocol processor to the outside line. Work on this autodialing option, as well as on other services such as call processing, file transfer, and higher-level communication protocols, is currently in progress.

V. CONCLUSION

This paper described the hardware implementation of a network for simultaneous communication of data, digitized voice, and analog video in a CATV system. The network uses a broadband approach to solve the distance limitation and delay problems suffered in local area networks. As a result, a considerably large community of users can be supported by the network. The problem of guaranteeing a continuous nondistorted speech signal at the receiver is solved using a variation of CSMA/CD protocol called movable slot time division multiplexing. This protocol places an upper bound on the maximum delay that can be experienced by a voice packet.

At the present time, an experimental network of five user nodes in two frequency bands is fully operational. The protocol processing hardware was described in detail. This processor is built with standard TTL components. It was shown that, owing to the method used by

the protocol processor for scheduling packet transmission times and compensating for timing discrepancy, clock synchronization is not a major problem.

REFERENCES

1. W. R. Franta and I. Chlamtac, *Local Networks*, Lexington, MA: Lexington Books, 1981.
2. N. F. Maxemchuk and A. N. Netravali, "A Multifrequency Multiaccess System for Local Access," *ICC 83*, Boston, MA.
3. A. N. Netravali and N. F. Maxemchuk, private communication.
4. R. M. Metcalfe and D. R. Boggs, "Ethernet: Distributed Packet Switching for Local Computer Networks," *Commun. ACM*, 19, No. 7 (July 1976), pp. 395-404.
5. A. N. Netravali and Z. L. Budrikis, unpublished work.
6. N. F. Maxemchuk, "A Variation on CSMA/CD That Yields Movable TDM Slots in Integrated Voice/Data Local Networks," *B.S.T.J.*, 61, No. 7 (September 1983), pp. 1527-50.

AUTHORS

Mehdi Hatamian, B.S.E.E., 1977, Ary-Mehr University of Technology, Tehran; M.S. and Ph.D. (Electrical Engineering), University of Michigan, Ann Arbor, in 1978 and 1982, respectively; AT&T Bell Laboratories, 1982—. From 1979 to 1982 Mr. Hatamian was a Research Assistant on a NASA project working on the design and development of hardware and software for one of the Space Shuttle experiments. His research interests are in real-time signal processing, circuit design, image processing, and VLSI design. Member, IEEE, Sigma Xi.

Edward G. Bowen, A.S. (Electrical Engineering), 1963, Vermont Technical College; B.S. (Electrical Engineering), 1971, Newark College of Engineering; AT&T Bell Laboratories, 1963—. At AT&T Bell Laboratories, Mr. Bowen is an Associate Member of the Technical Staff. His principal interests are coding of video signals.