Breaking Open the Set Top Box

David Banks, Anthony Wiley, Nicolas Catania,
Alistair Coles, Duncan Smith, Simon Baynham,
Eric Deliot, Rod Chidzey
Home Communications Department
Network Technology Department
HP Laboratories Bristol

HPL-97-139
November, 1997

gateway,
multi-service,
home, network,
IEEE1394, web,
MPEG

In this paper we describe the work being done in the area of home networks and gateways. This work is based in the idea of breaking open the set top box by physically separating the access network specific functions (for example demodulation and error correction) from the application specific functions (for example MPEG-2 video decoding). The access network specific functions reside in an access network gateway that can be shared by many end user devices. The first section of the paper presents the philosophy behind this approach. The end user devices and the access network gateways must be interconnected by a high bandwidth network which can offer a bounded delay service for delay sensitive traffic. We are advocating the use of IEEE1394 for this network, and the next section of the paper gives a brief introduction to this technology. We then describe a prototype Digital Video Broadcasting Satellite (DVB-S) compliant gateway that we have built. This gateway could be used, for example, by a PC for receiving a data service or by a digital TV for receiving an MPEG-2 video service. When combined with a suitable return path, the prototype could also be used to deliver bi-directional interactive services. A control architecture is then presented which uses a PC application to provide a web based user interface to the system. Finally, we provide details of our work on extending the reach of IEEE1394 and its standardization status.

Breaking open the set top box

David Banks, Anthony Wiley, Nicolas Catania, Alistair Coles,
Duncan Smith, Simon Baynham, Eric Deliot and Rod Chidzey

Hewlett-Packard Laboratories, Bristol, United Kingdom

ABSTRACT

In this paper we describe the work being done at HP Labs Bristol in the area of home networks and gateways. This work is based on the idea of breaking open the set top box by physically separating the access network specific functions (for example demodulation and error correction) from the application specific functions (for example MPEG-2 video decoding). The access network specific functions reside in an access network gateway that can be shared by many end user devices. The first section of the paper presents the philosophy behind this approach. The end user devices and the access network gateways must be interconnected by a high bandwidth network which can offer a bounded delay service for delay sensitive traffic. We are advocating the use of IEEE1394 for this network, and the next section of the paper gives a brief introduction to this technology. We then describe a prototype Digital Video Broadcasting Satellite (DVB-S) compliant gateway that we have built. This gateway could be used, for example, by a PC for receiving a data service or by a digital TV for receiving an MPEG-2 video service. When combined with a suitable return path, the prototype could also be used to deliver bidirectional interactive services. A control architecture is then presented which uses a PC application to provide a web based user interface to the system. Finally, we provide details of our work on extending the reach of IEEE1394 and its standardization status.

Keywords: gateway, multi-service, home, network, IEEE1394, web, MPEG.

1. INTRODUCTION

Many of todays access networks, used for the delivery of broadcast and interactive services to consumers, are currently in the process of becoming all digital infrastructures. Typical access networks, which are undergoing such changes, include direct broadcast satellite, terrestrial, cable and telco lines. The primary motivation for network operators to migrate to digital is bandwidth efficiency and its associated increase in service provision and service quality. For consumers, this will ultimately lead to them having a more diverse choice of access network providers, each providing an increasingly similar set of broadcast and interactive services.

To ensure that consumers experience continuity during the transitional period from analogue to digital, new set top boxes and modems will need to be developed and deployed. Initially, the primary function of these devices will be to ensure backward compatibility with existing devices such as TVs and PCs. However, there is also an opportunity to completely review how both broadcast and interactive services are terminated at, and distributed around, the consumer's premises. This paper presents an architecture where the functionality found in a conventional set top box has been partitioned and physically distributed. This provides a more flexible and cost effective means for interfacing to a wide range of access networks and their services.

The first section of this paper outlines the philosophy behind the distributed set top approach. The distributed set top architecture is reliant upon a suitable means of interconnecting the various components that make up the system. We are advocating the use of IEEE1394 for this purpose and present a brief introduction to IEEE1394. We then describe a prototype distributed set top system that we have developed based on a Digital Video Broadcasting Satellite (DVB-S[1]) compliant receiver system. This gateway could be used, for example, by a PC for receiving a data service or by a digital TV for receiving an MPEG-2 video service. When combined with a suitable return path, the prototype could also be used to deliver bidirectional interactive services. A control architecture is then presented which uses a web based control application to drive the system. Finally, we

Further author information -
David Banks: dmb@hplb.hpl.hp.com
Anthony Wiley: ajw@hplb.hpl.hp.com,
Nicolas Catania: niko@hplb.hpl.hp.com
Alistair Coles: anc@hplb.hpl.hp.com

provide details of our work on extending the reach of IEEE1394 and its standardization status.

2. DISTRIBUTED SET TOP BOX ARCHITECTURE

Analogue set top boxes have been used to interconnect TVs to access networks such as cable in North America and satellite in Europe. The primary purpose of these devices is to re-modulate signals to a frequency band that can be received on conventional TVs. A number of additional functions also tend to be supported and these include de-scrambling of subscription services (conditional access) and the ability to run electronic program guides.

In a digital set top box, the way in which a channel is extracted from an access network is very much dependent on the transmission media of the access network. The specific functions involved in channel extraction are tuning, demodulation and forward error correction. These functions are collectively termed *channel decoding*. The digital set top box functions that are independent of an access network, such as the de-scrambling, video decoding and electronic program guide, are collectively termed *source decoding*. The digital video encoding standard which has been widely adopted is MPEG-2 so in a digital set top box the video decoder is an MPEG-2 decoder. This is illustrated in Figure 1.

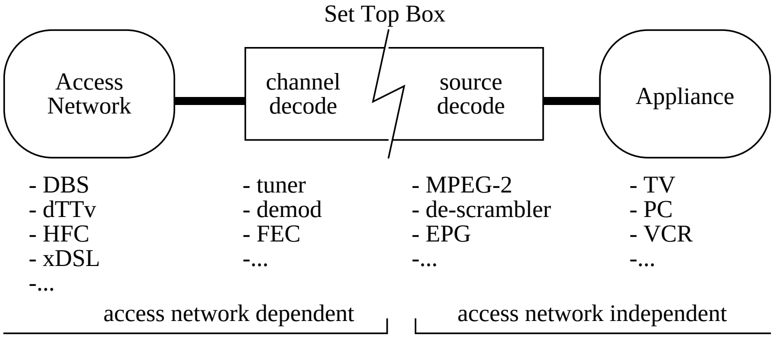


Figure 1 - An integrated digital set top box

The architecture we propose in this paper involves physically separating the channel decoding (access network dependent functions) from the source decoding (access network independent functions), see Figure 2. We will use the term access network gateway throughout the rest of this paper when referring to a physically distinct channel decoder. Achieving this separation requires a suitable means of interconnection and we are advocating the use of the IEEE 1394 High Performance Serial Bus (FireWire) for this purpose. 1394 is capable of supporting both delay sensitive and non-real time traffic through its isochronous and asynchronous services. When combined with its relatively low cost and topological flexibility, 1394 also looks attractive as an in-home backbone network. However, its current limited reach of 4.5m between repeaters is a barrier to its acceptance in this role. Later in this paper, we will describe the work and standardization efforts we are undertaking to extend the reach of 1394 to 50m at 100Mbps.

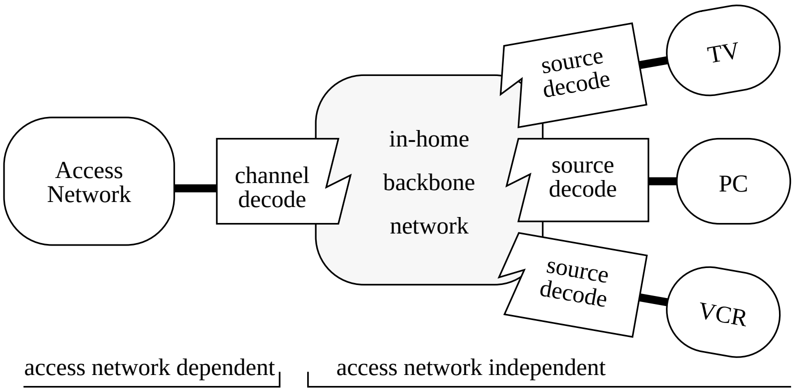


Figure 2 - Distributed set top box philosophy

The distributed set top architecture becomes more compelling when multiple devices, interconnected by a 1394 cluster/ backbone network, can access an access network gateway simultaneously since the cost of the access network gateway is then effectively shared between each of these devices. To ensure that an adequate range of programs is made available to these devices, an access network gateway needs to support the simultaneous processing of multiple channels. Conversely, one (or more) devices may require access to more than one access network and a similar reasoning may be applied, where multiple access network gateways provide data for one (or more) source decoder(s), see Figure 3.

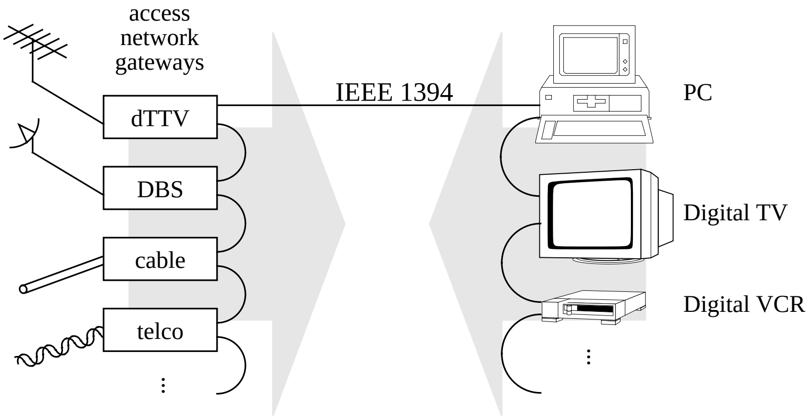


Figure 3 - An appliance may access multiple access networks and/ or multiple appliances may access a single access network

Another advantage of the distributed set top approach is that isolating the access network specifics from the source decoding enables the possibility for the access network to evolve with minimal disruption to the rest of the consumers infrastructure. Consumers also benefit from being able to replace discrete source decoders (supporting existing analogue devices) with digital appliances which will have integrated source decoding in the future. Although we expect source decoding functions to be fully integrated in future digital appliances, this is unlikely to be the case in general for the channel decoding functions. This is because each manufacturer would then have a separate product variant for each access network, which would be costly.

3. IEEE1394 OVERVIEW

IEEE Std 1394-1995[2] (IEEE Standard for a High Performance Serial Bus), also known as Serial Bus, FireWire or simply 1394, defines a high speed serial bus that is increasingly becoming the de-facto standard for interconnecting digital consumer devices[3]. It is unique in that it has strong support from both the computer and consumer electronics industries and it is viewed by many as the logical link between devices from these two worlds. In this section we present a brief review of the capabilities of 1394.

The best way to describe 1394 is to start at the physical layer and work upwards. The physical layer within each device acts as a mini repeater with typical devices having three ports. Ad-hoc topologies can be built up by linking devices together directly without the need for dedicated hubs. There are some restrictions on the topologies that can be created; cables between devices are limited to 4.5m, there can be at most 63 devices, there can be at most 16 hops between any pair of devices, and there can not be any loops. The cable itself consists of two shielded twisted pairs and a power and ground pair. Each physical layer acts as a bit level repeater between its ports. The main purpose of the power is to keep all the physical layers powered up, even if a particular device is powered off. Data transfer rates are currently defined at 100Mbps, 200Mbps and 400Mbps, and different speed capability devices can be mixed on the same bus.

When a device is added to or removed from the bus, a bus reset occurs. The bus then enters into a self-configuration process to determine the new topology and to allocate unique physical IDs to each device. In addition, certain nodes will be elected to perform specific functions: cycle master, isochronous resource manager and bus manager.

The link layer provides for two different types of data transfer: asynchronous and isochronous.

Asynchronous packets carry a 64 bit address, which is broken down into three parts: *bus_ID* (10 bits), *physical_ID* (6 bits) and *unit_address* (48 bits). The *bus_ID* identifies the destination bus (up to 1023 buses can be bridged together). The *physical_ID* identifies the device on the bus (up to 63 devices per bus). The *unit_address* identifies an address within a particular device. All non-broadcast packets are acknowledged by the destination, which allows the next layer up (the transaction layer) to implement a reliable data delivery service.

Isochronous packets are different in a number of respects. They are not addressed to individual nodes, instead they contain a 6 bit *channel_ID* and are broadcast out to all nodes. Each device on the bus decides whether to keep the packet based on the *channel_ID*. Isochronous packets are never acknowledged. In order to send isochronous data, a device must first reserve a channel number and some bandwidth. It does this through the *channels_available* and *bandwidth_available* registers held by the isochronous resource manager. The total amount of bandwidth that can be reserved for isochronous data is limited to 80% of the bus capacity, leaving at least 20% for asynchronous data.

The way in which asynchronous and isochronous packets share the bus is particularly simple and elegant. The device elected to be the cycle master is responsible for generating a *cycle start* packet every 125µs. Following this, isochronous arbitration occurs and each node in turn is permitted to send its isochronous packet(s). This is guaranteed to complete within a bounded time dependent on the total bandwidth reserved on the bus. After this, the remainder of the cycle is available for asynchronous packets. An asynchronous arbitration scheme is used which is based on the idea of a fairness interval. At the start of the fairness interval, each node that has asynchronous packets to send arbitrates for the bus and is allowed to send one packet. When all such nodes have done this, the bus becomes idle and this gap marks the beginning of a new fairness interval. The fairness interval is not fixed, but depends on the number of nodes contending and the size of the packets they have to send. The fairness interval may span multiple isochronous cycles.

4. PROTOTYPE DISTRIBUTED SET TOP

In this section we first introduce some of the issues that arise when carrying MPEG-2 over 1394, since this is central to our architecture. We do this with reference to the work of IEC61883[4], which specifies a standard encapsulation format for MPEG-2 transport stream (TS) packets when carried in 1394 isochronous packets. We then go on to describe our implementation of a prototype DVB-S access network gateway and a simple MPEG-2 source decoder, in which we have implemented these ideas.

4.1 MPEG-2 and timing jitter

The first consideration when transporting MPEG-2 over 1394 is that the MPEG-2 transport stream specification[5] places very tight bounds on the timing jitter that can be tolerated when delivering TS packets to the decoder. This restriction is there because of the need to synchronise a 27MHz reference clock in the decoder to the one used by the encoder. This serves two purposes. First, it allows the decoder and encoder to share a common timing reference, which is necessary to re-synchronize the video and audio elementary streams for presentation to the viewer. Second, it allows the decoder to generate an PAL/NTSC colour sub-carrier. The latter of these imposes the most stringent demands, since even slight instability of the colour sub-carrier results in colour smearing and distortion.[6]

There are two different sources of timing jitter in our system:

1. The access network gateway can filter the TS packets based on the program identifier (PID) value in the packet header. This allows only the program(s) of interest to be selected for forwarding out onto the 1394 bus, which reduces the bandwidth requirements. This rate reduction unavoidably introduces jitter.

2. The 1394 bus is a shared medium rather than a point to point link. A particular packet may experience a delay in gaining access to the bus because it is in use by some other device.

The mechanism used to compensate for jitter involves appending a time stamp to each TS packet. This is done early in the access network gateway, before any jitter has been introduced. The time stamp is based on the current 1394 cycle time (a timing reference shared by all nodes on the bus). The actual value used for the time stamp is the current cycle time plus a fixed offset. This offset must be greater than the maximum delay the TS packet will incur before reaching the source decoder. At the source decoder, the TS packet is held in a buffer until the bus cycle time exceeds the time stamp, and is then released to the MPEG-2 decoder. This results in a fixed delay between the access network and the MPEG-2 decoder.

The combination of the 4 byte time stamp and the 188 byte TS packet is referred to as a *source packet*.

4.2 Mapping source packets to 1394 isochronous packets

The second consideration when transporting MPEG-2 over 1394 is how to map source packets to 1394 isochronous packets. The obvious approach is to map each complete source packet into a single 1394 isochronous packet. This turns out to be inflexible since it fixes the bandwidth that must be reserved at 12 Mbps (192 bytes per 125µs isochronous cycle). If the transport stream is at a lower rate than this, bandwidth is wasted. If it is at a higher rate, the scheme breaks down.

A better approach is to first break the source packet into smaller blocks, and then choose the number of these blocks to carry in each isochronous packet based on the bandwidth of the transport stream. This is essentially what the Common Isochronous Packet (CIP) encapsulation defined in IEC61883 allows. The 192 byte source packet is broken into 24 byte data blocks. An isochronous packet is allowed to contain 1, 2, 4, 8 or an integer multiple of 8 data blocks. One isochronous packet is sent every isochronous cycle. This results in many more bandwidth possibilities, starting at 1.5Mbps, and represents a good compromise between simplicity and flexibility.

4.3 DVB-S access network gateway

We have developed a prototype DVB-S access network gateway that allows an MPEG-2 transport stream (or part of that stream) received from a satellite transponder to be broadcast onto a 1394 bus. The gateway is able to handle both conventional video/audio programs and broadcast data. A block diagram of the gateway is illustrated in Figure 4.

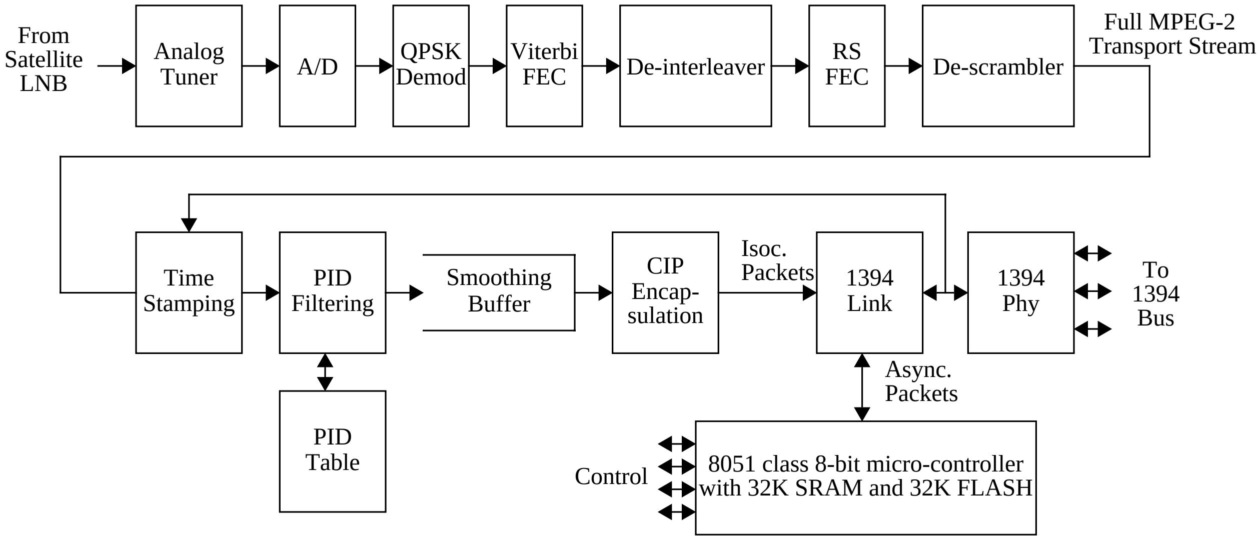


Figure 4 - Access network gateway block diagram

The front end of the gateway performs tuning to a single satellite transponder, followed by analog to digital conversion, QPSK demodulation and forward error correction. The QPSK demodulator can cope with symbol rates between 20 Mbaud and 30 Mbaud. The output of this stage is a transport stream with a typical data rate of 38Mbps (27.5Mbaud, rate 3/4 FEC).

The next stage appends a time stamp to the TS packet, based on the value of the 1394 cycle time plus a programmable offset. This turns out not to be trivial, because we have used an off the shelf 1394 link layer device. The problem here is that although a copy of the 1394 cycle time is maintained by the 1394 link layer device in a register, the delay in reading this register is not constant. This would introduce unacceptable jitter in the time stamps. To work around this we implemented hardware to recognise the cycle start packet directly as it passes across the phy/link interface and to extract the current cycle time from this.

The transport stream at this point contains several programs, not all of which may be of interest. The next stage in the gateway looks up each TS packet's PID in a table to determine whether to drop or keep the packet. Since the PID field is 13 bits and only a single bit is needed for the decision, 8K bits of RAM are required.

If the TS packet is to be kept, it is written into the smoothing buffer. The purpose of the smoothing buffer is to accommodate short term fluctuations in the average rate of TS packets after filtering, and to cope with any delays in gaining access to the shared 1394 bus. Both of these values are well bounded, and the smoothing buffer turns out to be quite small in practice. In our prototype we used a discrete 32Kx9 synchronous FIFO.

Following the smoothing buffer is hardware to fragment the TS packets into small blocks and to add the necessary CIP headers. This block is programmable to allow the number of blocks per isochronous packet to be varied depending on the bandwidth requirements of the filtered stream.

The time stamping, filtering, smoothing buffer and CIP encapsulation are implemented in a single Xilinx FPGA with an external SRAM and FIFO.

The final parts of the gateway are the 1394 link and physical layers. These are implemented using off the shelf components.

To allow the components of the gateway to be controlled, a simple 8051 class 8-bit micro-controller system was implemented with 32K of SRAM and 32K of FLASH. We deliberately chose a very low end micro-controller to demonstrate that the protocols to control the gateway are neither complicated nor expensive to implement. The micro-controller handles all asynchronous traffic and implements the 1394 transaction layer and a basic register command set that allows the gateway to be controlled remotely over the 1394 bus. The other main task it performs is to continuously monitor the performance of the tuner and demodulator and to compensate for any drifting (which would be mainly due to variations in the satellite dish’s LNB local oscillator).

4.4 MPEG-2 source decoder

The other device in our system is an MPEG-2 source decoder. A block diagram of this is illustrated in Figure 5.

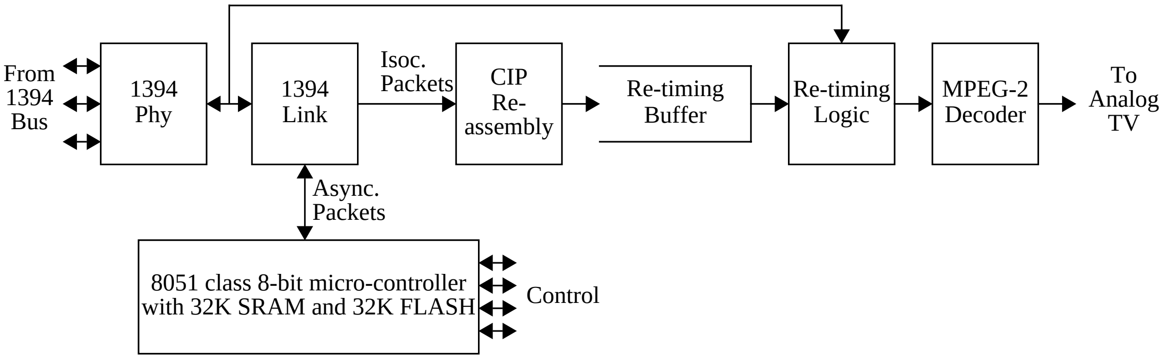


Figure 5 - Source decoder block diagram

Operation of the MPEG-2 source decoder is somewhat simpler than the access network gateway. Isochronous packets from the 1394 bus, containing CIP fragments, are first reassembled into complete source packets. These are held in a re-timing buffer until the current 1394 cycle time exceeds the time stamp on the front of the source packet. When this happens, the time stamp is stripped off and the remainder of the packet is passed to the MPEG-2 decoder. Care is needed to ensure that late or corrupted time stamps do not result in the re-timing logic stalling for long periods.

The prototype MPEG-2 source decoder we have built uses many of the same components as the access network gateway, and in fact the same board level design performs both functions. Instead of implementing our own MPEG-2 decoder block, we have made use of a demonstration board available from one of the many MPEG-2 decoder chip vendors.

5. CONTROL ARCHITECTURE

In this section we explain how the hardware previously described is controlled using a simple command set built on top of 1394 asynchronous transactions. These allow, for example, the gateway to be tuned to a given satellite transponder and a particular set of PIDs to be selected. We go on to describe a PC based control application that uses these commands to dynamically generate a set of web pages. The user interface to the system is through a conventional web browser. The web pages contain the names of the programs currently transmitting and the user can perform program selection by simply clicking on the program name. The complete system is illustrated in Figure 6.

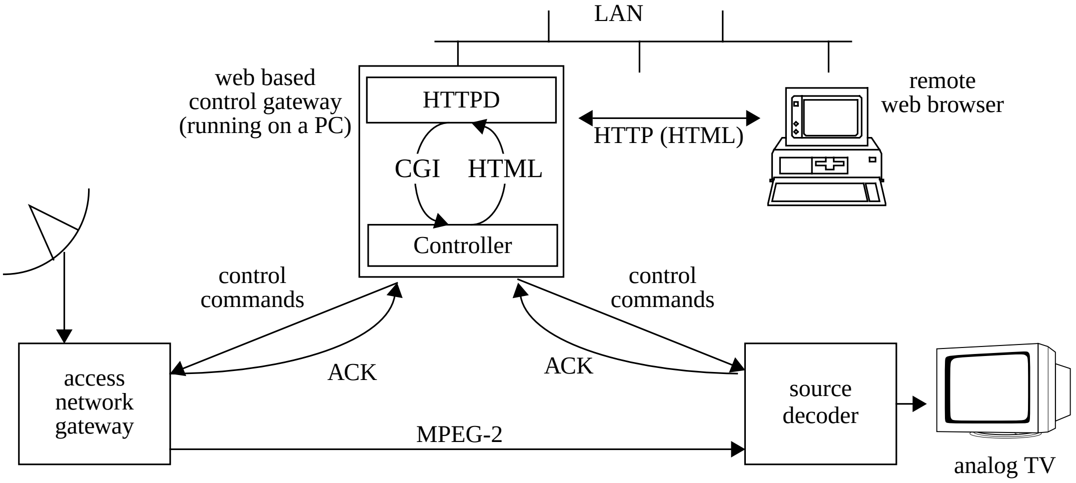


Figure 6 - Web based control model

5.1 Control command set

We have implemented a simple control protocol that allows the access network gateway and source decoder to be controlled from a remote 1394 device (a PC in this case). The 1394 standard specifies an area in a device's 1394 address space that is reserved for device specific registers. We use 1394 write transactions to specific addresses in this region to implement control commands. The address of the write transaction indicates the action to be performed, and the data to be written represents the parameter(s) for the action. The 1394 transaction layer provides for the acknowledgement of transactions. If a command is completed successfully, then an `ACK_complete` is returned. If a command fails an `ACK_data_error` is returned and the controller infers that the last operation has failed. The commands implemented by the access network gateway and the source decoder, together with their associated parameters, are listed in Table 1.

command	*parameters*
`set_isoch_channel`	isochronous channel number
`set_packet_size`	number of blocks per isochronous packet
`tune`	frequency, LNB polarisation, symbol rate
`PID_enable`	PID number to enable
`PID_disable`	PID number to disable
`clear_all_PIDs`	none
`enable_all_PIDs`	none
`set_mpeg_PIDs`	audio PID, video PID, PCR PID

Table 1 - 1394 command set

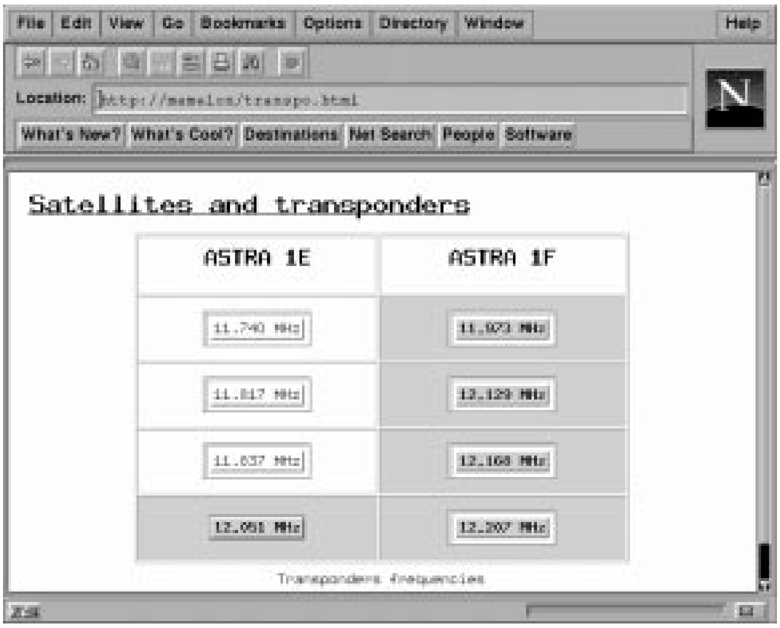
The `set_isoch_channel` and `set_packet_size` commands are common to both the access network gateway and source decoder, and should be used consistently. The `set_isoch_channel` command controls the 1394 isochronous channel number used by the access network gateway to broadcast TS packets on to the 1394 bus. The `set_packet_size` command controls how TS packets are fragmented prior to this. It should be set according to the peak expected bandwidth for the filtered transport stream.

There are several commands implemented only in the access network gateway. The `tune` command allows the gateway to be tuned to a given frequency, LNB polarisation and symbol rate. If this command is successful, there will be a valid transport stream at the input of the PID filter. The various PID related commands allow programs not of interest to be filtered out. `PID_enable` and `PID_disable` select or de-select a single PID at the time. The `clear_all_PIDs` command disables all the programs. Following this, there will be no isochronous traffic on the 1394 bus. The `enable_all_PIDs` command does the opposite and, following this, the full transport stream will be broadcast on to the 1394 bus.

The only source decoder specific command is `set_mpeg_PIDs`. This command allows the PID values of the audio and video elementary streams to be passed to the MPEG-2 decoder. It also identifies the PID of the stream containing MPEG-2 program clock reference (PCR) time stamps. This is needed by the MPEG-2 decoder in order to synchronize the audio and video streams.

5.2 PC based control application

The control application is a piece of software running on a PC that generates a set of web pages which enable a remote web browser to control the system. The opening web page presents the user with a list of (statically configured) satellite transponder frequencies. An example of the transponder selection web page is illustrated in Figure 7.



Location: http://mamelon/transpo.html

Satellites and transponders

ASTRA 1E	ASTRA 1F
11.740 MHz	11.973 MHz
11.817 MHz	12.129 MHz
11.837 MHz	12.168 MHz
12.051 MHz	12.207 MHz

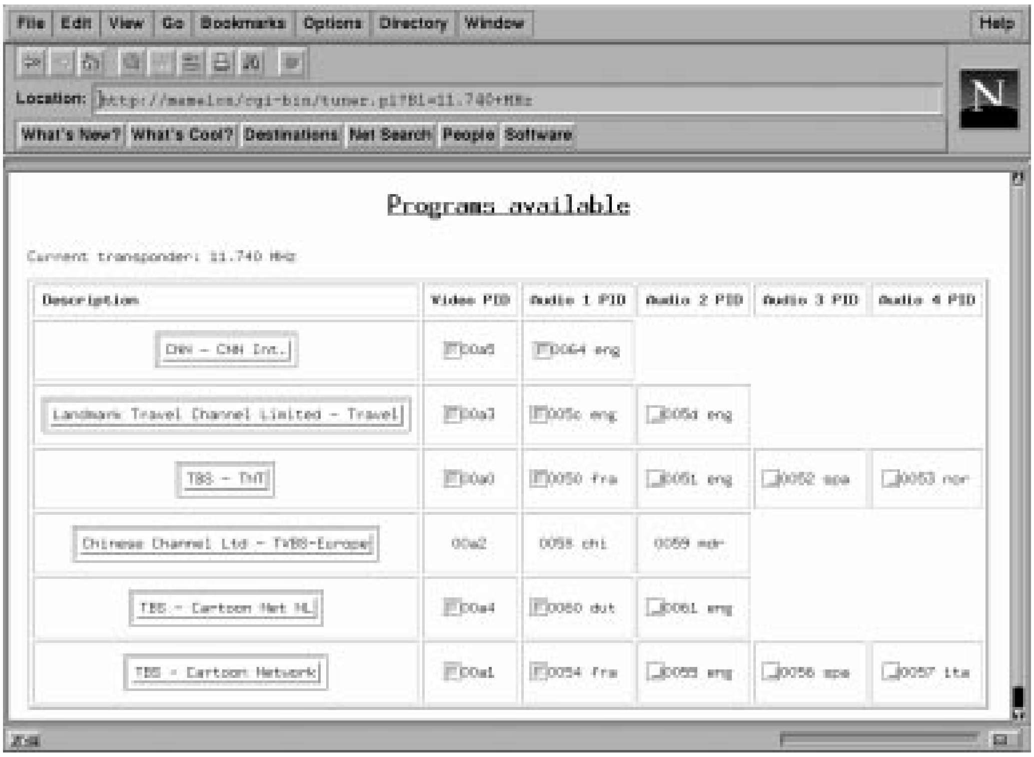
Transponders frequencies

Figure 7 - Transponder selection web page

To select a particular transponder, the user clicks on the corresponding button. The sequence of events following this is:

1. The remote browser sends a HTTP request packet to the HTTP daemon on the server indicating the button pressed.
2. The HTTP daemon uses the common gateway interface (CGI) mechanism to invoke a script to perform the following:
    3. The `tune` command is sent to the access network gateway, which sets the tuner to the requested frequency.
    4. An `enable_PID` command is sent to the gateway to select the DVB program association table (PAT[7]).
    5. An `enable_PID` command is sent to the gateway to select the DVB service description table (SDT[7]).
    6. An `enable_PID` command is sent to the gateway to select the program map table (PMT[7]) for each program in the PAT.
    7. The `clear_all_PIDs` command is sent to the gateway.
    8. The control application uses the information in each of these tables to construct the next web page.
9. The new web page is returned to the browser in a HTTP response packet.

The web page returned to the user contains a list of the program names (taken from the SDT) together with information about the available audio and video streams for each program. Where there are multiple audio streams available in different languages, the web page lists these. An example of the program selection web page is illustrated in Figure 8.



Location: http://mamelon/cgi-bin/tuner.pl?Bl=11.740+MHz

Programs available

Current transponder: 11.740 MHz

Description	Video PID	Audio 1 PID	Audio 2 PID	Audio 3 PID	Audio 4 PID
CNN - CNN Int.	☑ 00a5	☑ 0064 eng			
Landmark Travel Channel Limited - Travel	☑ 00a3	☑ 005c eng	□ 005d eng		
TBS - TNT	☑ 00a0	☑ 0050 fra	□ 0051 eng	□ 0052 spa	□ 0053 nor
Chinese Channel Ltd - TVBS-Europe	00a2	0058 chi	0059 mdr		
TBS - Cartoon Net NL	☑ 00a4	☑ 0060 dut	□ 0061 eng		
TBS - Cartoon Network	☑ 00a1	☑ 0054 fra	□ 0055 eng	□ 0056 spa	□ 0057 ita

Figure 8 - Program selection web page

The user now selects the appropriate audio and video streams, and then clicks on the channel name. The sequence of events following this is:

1. The remote browser sends a HTTP request packet to the HTTP daemon on the server indicating the button pressed.
2. The HTTP daemon uses the common gateway interface (CGI) mechanism to invoke a script to perform the following:
    3. An `enable_PID` command is sent to the access network gateway to enable the audio and video streams.
    4. The `set_mpeg_PIDs` command is sent to the source decoder to start reception.
    5. The control application updates the web page to indicate the channel selected.
6. The new web page is returned to the browser in a HTTP response packet.

The access network gateway is now broadcasting the MPEG-2 video and audio streams onto the 1394 bus. These are being decoded by the source decoder and displayed on the television screen. The user may now either select another program, or back up to the transponder selection page. The whole process of program selection takes approximately 400ms.

6. EXTENDING THE REACH OF IEEE1394

The 1394 standard defines a high speed serial bus for operation over shielded cables up to 4.5m in length. This relatively short cable length is a significant impediment to the application of 1394 as an in-home network, and specifically as a technology to link geographically dispersed components of a distributed set top box. To overcome this limitation, a prototype 1394 extender has been constructed which allows 1394 nodes to be separated by at least 50m. The objectives for the extender were similar to the requirements established by the VESA Home Network committee[8]: a minimum distance of 50m at the 1394 base rate (100Mbps) using low cost, amateur installable cables.

Clearly the most expedient solution would be to increase the existing 1394 cable length whilst maintaining the electrical signalling interface. This is not desirable for a number of reasons:

1. The 1394 signalling method does not allow for increased cable lengths. 1394 uses differential signalling on two twisted pairs: one pair carries the data while the other pair carries an encoded version of the clock. Increased cable attenuation and propagation skew between the twisted pairs used for data and clock would degrade these signals to the point of failure.

2. The 1394 physical layer electrical interface is DC coupled. This is necessary because 1394 data and control signals contain DC offsets which must be propagated to the far end of a cable. As a result, care must be taken in equipment design to ensure that sufficient electrical isolation is provided between physically separated devices. Indeed, the 1394 standard recommends that this be achieved through the use of transformer or capacitor coupling which electrically isolate the PHY and LINK circuits (creating an obstacle to the integration of these functions). As the physical separation of devices increases, electrical isolation becomes more important and should be achieved through the use of AC coupled signalling channels as is the common practice in twisted pair Local Area Networks.

3. The shielded 1394 cable is currently factory assembled with connectors, and does not lend itself to amateur installation and termination.

A block diagram of the extender is shown in Figure 9. The extender is designed to operate over either category 5 unshielded twisted pair (UTP5) cables or plastic optical fiber (POF) cables, the choice being made by installation of an appropriate transceiver device. UTP5 is commonly used for commercial LANs (e.g. 100BaseT) and is easily terminated using RJ-45 connectors. POF is a low cost alternative to glass optical fiber and is also easy to install and terminate. Both of these media are capable of transmission speeds in excess of 100Mbps over the required cable length of 50m.

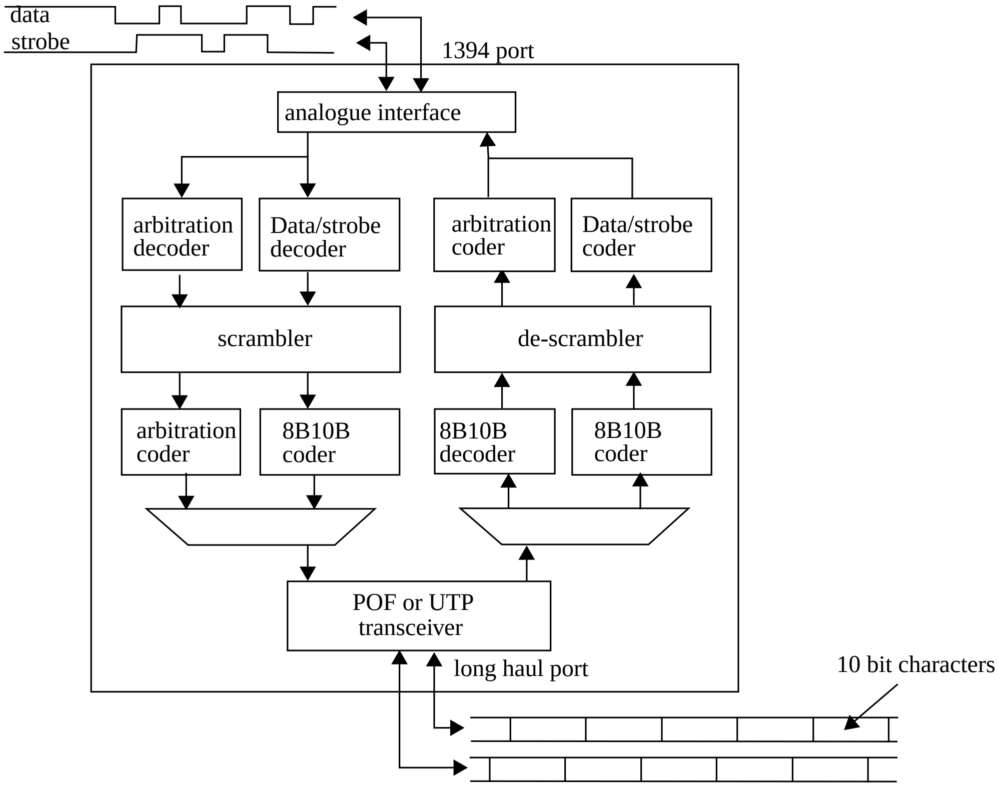


Figure 9 - Extended FireWire repeater block diagram

Data and control signals are received at the 1394 port and mapped to a continuous information stream on either a single plastic fiber or a single twisted pair (the transmit channel) at the long haul port. Information received on the second fiber or twisted pair (the receive channel) is used to replicate the correct 1394 data or control states on the 1394 port.

The 1394 port consists of two bidirectional interfaces with the two 1394 twisted pair channels. During 1394 arbitration, the arbitration decoder examines the state of these 1394 interfaces and decodes the received arbitration state. The arbitration state is then scrambled and coded as a 10 bit character. When a data packet is received on the 1394 port, the data is sampled one byte at a time, and each byte is scrambled and coded as a 10 bit character, using an 8B10B block code. The 8B10B block code used for data is the same as that specified in the Fibre Channel standard[9]. For arbitration states, a new set of 10 bit characters has been invented. This combination of data coding and arbitration state coding has been designed to achieve the following objectives:

1. The resultant stream of 10 bit characters is DC balanced (i.e. it contains an almost equal number of 1s and 0s) and therefore suitable for transmission over AC coupled channels such as UTP5 with transformers, or POF.
2. The 10 bit characters used for data are unique from those used for arbitration and separated by a Hamming distance of 2. This provides some protection against errored data characters being mistaken for arbitration signals.
3. Regular transitions are guaranteed in the stream of 10 bit characters, which allows easier recovery of a clock in the receiving long haul port.
4. When combined with the scrambler, the spectrum of the data and arbitration signals is conducive to satisfying EMC regulations, which are particularly stringent in domestic environments.

A standard approach to extended 1394 links is now being defined by the IEEE 1394B working group[10]. The current proposal is similar to the prototype described here, using the same coding technique for data and arbitration signals. However, the standard will also define longer distance 1394 links at higher rates up to 3.2Gbps. It is expected that at the higher speeds, long haul links will utilize either glass fiber or new, higher performance, graded index POF.

7. ISSUES AND SCOPE FOR FURTHER WORK

The access network gateway prototype we have developed is limited to receiving a single channel (transponder) at a time. At the start of our development, we did consider the options for building a multi-channel prototype. We were prevented from doing this by restrictions in the 1394 link layer chips available at the time, whose intended application was in peripheral devices such as digital camcorders. Since these only require one real time stream, much of the context associated with that stream (packet size, isochronous channel number, etc.) is programmed by a micro-controller into registers. It is not feasible to update these as rapidly as would be required (every 125µs) in a multi channel gateway. Although more recent silicon has more functionality (including support for IEC61883 time stamping and CIP encapsulation), it has still been developed with single channel applications in mind.

One of the limitations of our control protocol is that it does not provide any mechanism to aid sharing of the gateway between multiple devices. Since we completed this work, the 1394 trade association’s audio/video control protocol (AV/C)[11] has been expanded to include a command set for controlling remote tuners. Rather than enhance our own protocol, it would seem sensible to implement this emerging standard. If we did this, then we would also adopt the standard connection management model, based on the idea of logical plugs, that is defined in IEC61883.

The control application we have written can be thought of as a basic electronic program guide (EPG). Another area of possible future work would be to expand this into a full EPG, whilst still retaining the web based interface. Although there are some differences between the DVB and ATSC standards for service information[7,12], an application could be written to handle both. In our prototype system, the control application runs on a PC. In a real system this would probably move on to either the access network gateway or the source decoder.

Whilst we continue to support the efforts to develop a standard long distance 1394 physical layer, we are also interested in wireless backbones for consumers who choose not to install wires between rooms. The current radio technologies which seem plausible for this are unlikely to support bandwidths of 100 Mbps, so it will not be possible to link two 1394 clusters with a simple repeater. The alternative is to develop a distributed 1394 bridge, and this is something we are considering.

REFERENCES

1. European Telecommunications Standards Institute, “Framing Structure, channel coding and modulation for satellite systems”, ETS 300 421, European Telecommunications Standards Institute, 1994.

2. IEEE Computer Society, “IEEE Standard for a High Performance Serial Bus”, IEEE Std 1394-1995, Institute of Electrical

and Electronic Engineers, New York, 1996.

3. Ingrid J. Wickelgren, “The facts about FireWire”, IEEE Spectrum, pp19-25, April 1997.

4. International Electrotechnical Commission Work Group SC100C, “Digital Interface for Consumer Electronic Audio/Video Equipment”, Draft IEC Std 61883-1997, International Electrotechnical Commission, Geneva, expected 1997.

5. International Organisation for Standardization, “Generic Coding of Moving Picture and Associated Audio Information: Systems”, ISO/IEC DIS 13818-1, International Organisation for Standardization, Geneva, 1994.

6. Giovanni F. Andreotti, Giampaolo Michieletto, Luigi Mori, and Alberto Profumo, “Clock recovery and reconstruction of PAL pictures for MPEG coded streams transported over ATM networks”, IEEE Transactions on Circuits and Systems for Video Technology, vol.5, no.6, pp 508-514, December 1995.

7. European Telecommunications Standards Institute, “Specification for Service Information (SI) in Digital Video Broadcasting (DVB) Systems”, Draft ETS 300 468, European Telecommunications Standards Institute, 1995.

8. Joel A. DiGirolamo, “The VESA Home Network: A White Paper”, available from the VESA web site (http://www.vesa.org/vhn_wp.PDF).

9. American National Standards Institute, “Fibre Channel Physical and Signalling Interface (FC-PH)”, ANSI X3.230-1994, American National Standards Institute, 1994.

10. IEEE P1394B (Gibabit 1394) Project web site (http://www.fireflyinc.com/p1394b/index.html).

11. 1394 Trade Association Audio Video Working Group, AV/C Digital Interface Command Set, available from the 1394TA web site (http://www.1394ta.org/abouttech/specifications/techspec.html).

12. Advanced Television Standards Committee, “Program Guide for Digital Television”, ATSC A/55, Advanced Television Standards Committee, 1996.