A/PROV

PROVISIONAL APPLICATION COVER SHEET

JC514 U.S. PTO 09/22/97

This is a request for filing a PROVISIONAL APPLICATION under 37 CFR 1.53(b)(2).

Docket Number	2810-052PRO	Type a plus sign (+) inside this box →	+

INVENTOR(s)/APPLICANT(s)

LAST NAME	FIRST NAME	MIDDLE INITIAL	RESIDENCE (CITY AND EITHER STATE OR FOREIGN COUNTRY)
HUMPLEMAN	Richard	J.	Fremont, California
WOLFF	Robert	M.	Fremont, California
HARMS	Gordon	K.	San Jose, California
Deacon	Michael		Campbell, California
Chandiramani	Omi		Santa Clara, California

TITLE OF THE INVENTION (280 characters max)

IMPROVED HOME NETWORK, BROWSER BASED, COMMAND AND CONTROL

CORRESPONDENCE ADDRESS

LOWE, PRICE, LEBLANC & BECKER
99 Canal Center Plaza, Suite 300
Alexandria, VA 22314
(703) 684-1111

STATE	Virginia	ZIP CODE	22314	COUNTRY	USA

ENCLOSED APPLICATION PARTS *(check all that apply)*

[X] Specification with Drawings *Number of pages* [76] [ ] Small Entity Statement
[ ] Drawings *Number of sheets* [ ] [ ] Other (specify): _____

METHOD OF PAYMENT *(check one)*

[ ] A check or money order is enclosed to cover the Provisional filing fees [X] The Commissioner is hereby authorized to charge filing fees and credit Deposit Account Number: 12-2237	PROVISIONAL FILING FEE AMOUNT ($) 150.00	

The invention was made by an agency of the United States Government or under a contract with an agency of the United States Government.

[X] No.

[ ] Yes, the name of the U.S. Government agency and the Government contract number are: _____

[ ] Additional inventors are being named on separately numbered sheets attached hereto.

Respectfully submitted,

LOWE, PRICE, LEBLANC & BECKER

Thomas A. Jolly
Registration No. 39,241

99 Canal Center Plaza, Suite 300
Alexandria, Virginia 22314
(703) 684-1111 TAJ:ka
Date: September 22, 1997

...ate under 37 CFR 1.10

EM445691253US 9-22-97
Express Mail Mailing Number Date of Deposit

I hereby certify that this application is being deposited with the United States Postal Service Express Mail Post Office to Addressee service under 37 CFR 1.10 on the date indicated above and is addressed to the Assistant Commissioner for Patents Washington, DC 20231

Kim Arca
(Printed name of person mailing application)

Kim Arca
(Signature of person mailing application)

Cover Page for

Improved Home Network, Browser Based, Command and Control

The present invention relates to networks, and more particularly to networks that are used to connect and control a plurality of devices within a home or business setting.

This provisional application includes this cover page and pages 1 through 29 of a home network, browser based, command and control system specification which describes methods and arrangements in accordance with one embodiment of the present invention for controlling a network. This provisional application further includes Appendices A through D which are disclose further embodiments and/or refinements to the home network, browser based, command and control system.

List of enclosed pages {total pages, with drawings therein, 76}:

Patent Disclosure for Improved Home Network, browser based, command and control: CLO-HTML/HTTP/IP/IEEE1394.

Enabling DTV to render GUI's from and control CE devices with Command Language Optional (CLO).

1.0 Background

A couple of goals: 1) Stretch the standard use of HTML to its absolute limits prior to making extensions to that standard. 2) Put as few constraints on the target systems as possible. There are instances where dictating behavior is required, but outside of this, dictating behavior shall be limited as much as possible.

The heart of the "power of the web" is made up of two primary things: 1) Hyperlinking gives us the ability to get from anywhere to anywhere with one simple "reference" click. 2) Servers (and authors) are able to present new information, a new user interface, and many custom features to the masses who have web browsers as long as they stay inside the HTML spec boundries. This is a simple statement, but a very powerful notion. Historically, users have had to get "new" software on their computers in order to have these new "experiences". Never has there been a spec which is both simple and flexible which allows the world to choose one web browser piece of software to experience so much diverse material. Also, with the advent of server-side custom components, the user can be presented with simple HTML which activates complex server-side actions without the user knowing "what's behind the curtain".

A key in understanding why these points are so important is to realize that this means each device is almost completelyresponsibile for its own actions and this makes it almost fully independent of all other devices. This is really the idea of"encapsulation".

As long as each device on the network has HTML files to describe their GUI and as long as they use HTTP protocol to transfer those files, then any "client" device that understands how to "web-browse" and render HTML will be able to use the device with the human-interface GUI. There are caveats to this such as automation and macro capabilities which do complicate things, but the power of the web does indeed lend itself very well to the home theater for such devices. The reason this works is that if a new "unknown" device is invented with a new function such as "instant video rental", all the new device has to do is present the HTML code which implements a button whose label is "Instant Video Rental" and when the user presses this button, the action taken is to "submit" or "post" information back to the originating

Inventor's: ROBERT WOLFF, KEVIN HARMS, MICHAEL DEACON, RICHARD HUMPLEMAN

Inventor's Signature ..... Date 6/13/97

Inventor's Signature ..... Date 6/16/97

Witness Name ..... Signature ..... Date .....

Witness Name ..... Signature ..... Date .....

device. The device itself figures out which button was pressed and takes the appropriate action to carry out the action. The client system has no knowledge of how this happens and does not need to know (nor does it want to).

2.0 Outline

The Home Theater is considered to be the model for digital equipment throughout the whole home, see figure 1. This shows a collection of entertainment sources, audio, video and data networked together by a digital home network and one or more Digital TV[1] sink or display devices. The DTV provides the human interface for the whole system via GUI and the display.

There cannot be only one network to satisfy all needs and so the diagram shows bridges to a Home Automation network and a 1394 network that supports IEC1883 protocols. Another example (not shown) is an Ethernet Home Office Network. One of the reasons for choosing IP protocols is that it is a fully mature routeable inter-networking protocol that allows different networks in and outside the home to inter-operate.

See Home-Network Protocol Proxy Appendix.

The problem faced by the DTV designer is how to control the potential myriad of devices while keeping the unit simple, current and generic.

The brute-force way is to build a DTV with knowledge of all the devices and GUI user interfaces for them. In addition one could develop a command set for the digital interface to enable remote control. One problem with this approach is that given the development rate of new devices it is impossible to keep the DTV GUI and Interface/Network command set from being complex and obsoleted.

The approach chosen here is for the DTV to be a rendering Browser and bring in the Character of the device the user wishes to control, see figure 2. The device is represented by an 'html' (hyper text mark-up language) file kept in a accessible directory of the device. The 'html' file is an ascii text file with details of the device and infomation that enables a browser to present it graphically.
In addition to 'bringing in' the html GUI to the DTV there is a return capability back to the device making the mechanism 2 way. The user can view the rendered html GUI and control the device by 'clicking' buttons and form fill.

The DTV browser accesses, using http protocol, the devices html file and renders it to create the devices GUI and present it to the user. The DTV can do this for any device but doesn't know what the

---

1.See appendix for definition of DTV

Inventor's: ROBERT WOLFF, KEVIN HARMS, MICHAEL DEACON, RICHARD HUMPLEMAN

Inventor's Signature ..... Date 6/13/97

Inventor's Signature ..... Date 6/16/97

Witness Name ..... Signature ..... Date .....

Witness Name ..... Signature ..... Date .....

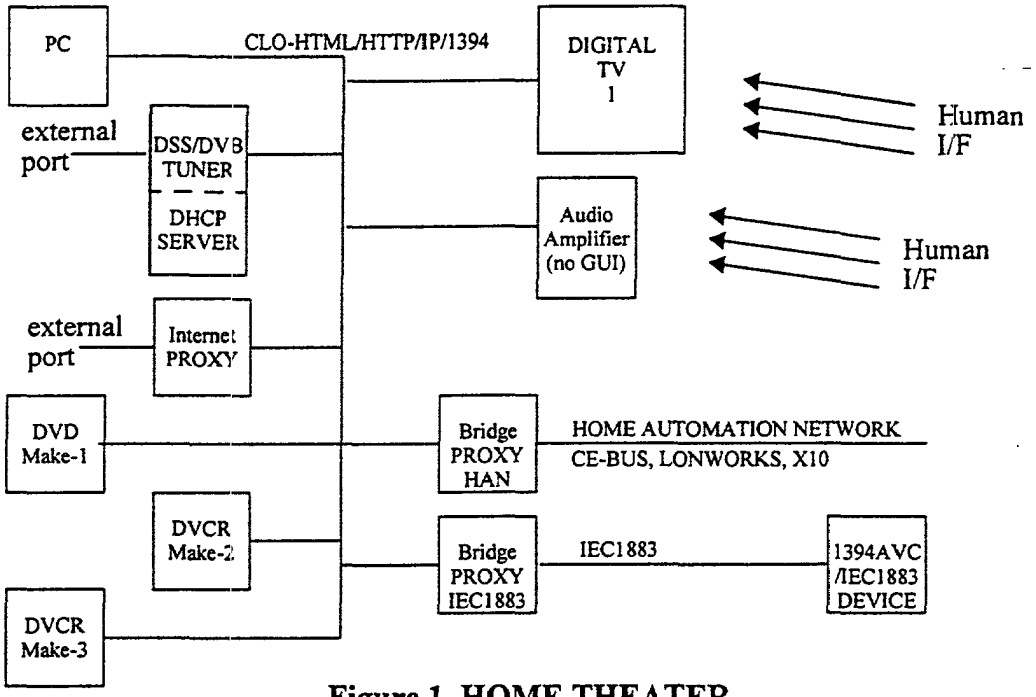


Figure 1, HOME THEATER

device type is -it is up to the human intelligence to read the GUI. This mechanism is used ordinarily for accessing information by computers communicating on the Internet and World-Wide-Web (www). In the case here, for device control, the html file represents the character of the device whereas in the www case the html file represents information.

The device behaves like a server and the DTV behaves like a client. The client accesses the server initially to control it and later perhaps to receive it's video program stream. Typically the server has multiple html files in a file directory structure. In this case they may be partly device specific html and partly dynamically generated html from the media installed or online. The user is unaware of the physical source of the html GUI's eg some reside local to the DTV for it's own control to enable the switch between the overlay or window for the control browser and the overlay or window for the video program on the DTV.

For simple cases the DTV is completely generic and there is no need for an interface command set. Moreover the system is compatible with the Internet protocols so may be controlled from a computer outside the home running a browser just as well as the home DTV.

As with the Internet case, the html file access use the http protocol that runs on the TCP transport protocol and IP network layer protocol. TCP provide a reliable delivery mechanism and IP the routeable addressing mechanism. The data transfers of audio and video program material are started by the html mechanism but run on the digital interface using hardware streams outside of the client server http/IP network based system.

Inventor's: ROBERT WOLFF, KEVIN HARMS, MICHAEL DEACON, RICHARD HUMPLEMAN

Inventor's Signature..... Date 6/13/97

Inventor's Signature..... Date 6/16/97

Witness Name..... Signature..... Date.....

Witness Name..... Signature..... Date.....

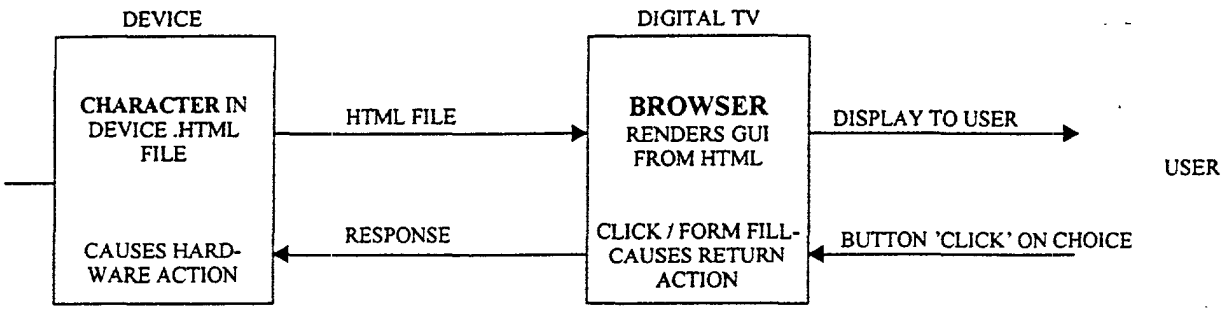


Figure 2, HTML 2 Way Mechanism

There are 2 cases where the html GUI command and control method isn't sufficient:-
2) Machine and/or automatic control (see later).
1) A client with no display capability (no GUI) -illustrated by the Audio Amplifier device. This is considered to be a low order problem as there are other ways to get-around this problem.

For these cases it is appropriate to have commands, buried in the html code, that are readable and writeable by software see later.

Inventor's: ROBERT WOLFF, KEVIN HARMS, MICHAEL DEACON, RICHARD HUMPLEMAN

Inventor's Signature..... Date..6/13/97.....

Inventor's Signature..... Date..6/16/97.....

Witness Name..... Signature..... Date.....

Witness Name..... Signature..... Date.....

3.0 Architecture

3.1 Model

For simplicity the architecture is described with 3 devices: DTV (Digital TV), DVCR (Digital Video Cassette Recorder) and DSS-NIU (DirecTV Satellite System-Network Interface Unit), see figure 3.

The DTV is a Client as described above (client DTV unit also has to have server capability to enable access to the local hardware, see later). See appendix for more detail on the client model.

The DSS-NIU is a mini-server (limited capacity) unit that can output a video program. The DSS-NIU is the video program tuner (satellite) separated into it's own unit called Network Interface Unit. Logically this makes sense as both DVCR and DTV can now make use of the NIU capability. See appendix for more detail on the mini-server models.

The DVCR is a mini-server unit as regards control. When recording (from DSS) the DVCR receives data and seems to be a client but this is a data transfer outside of the realm of client/server model.

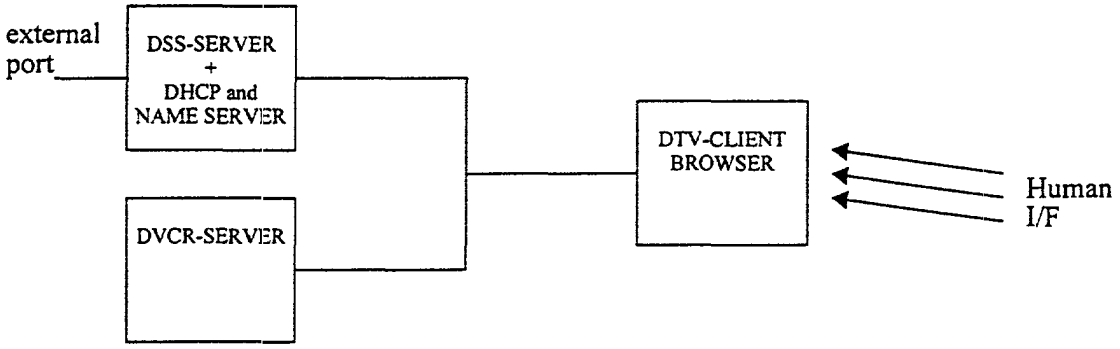


Figure 3, Architecture Devices

3.2 Device and Media discovery

The IP network protocol has to be automatically supported by the system. The unit least likely to be replicated is nominated to be the DHCP server (Dynamic Host Configuration Protocol) -in this system the DSS Server (though can be installed in any device of the home network). This is required in IPV4 (IP

Inventor's: ROBERT WOLFF, KEVIN HARMS, MICHAEL DEACON, RICHARD HUMPLEMAN

Inventor's Signature..... Date 6/13/97

Inventor's Signature..... Date 6/16/97

Witness Name..... Signature..... Date.....

Witness Name..... Signature..... Date.....

network protocol version 4) to enable the network to work automatically and not require fixed IP addresses to be set up by the user. With future IPV6 this requirement will go away.

See also appendix : Self Populating HTTP Server.

As the DTV and DVCR power up they run DHCP client software that broadcasts on the network for the DHCP server. Once the DHCP server assigns IP addresses and names it updates the DSS-PC resident, ***devices.html*** file with a hot link to each of the devices top-level html page.. This becomes the key file for user on the network to access devices, see figure 4. (ALL **FILE NAMES** GIVEN ARE ABRITRARY UNLESS OTHERWISE STATED).

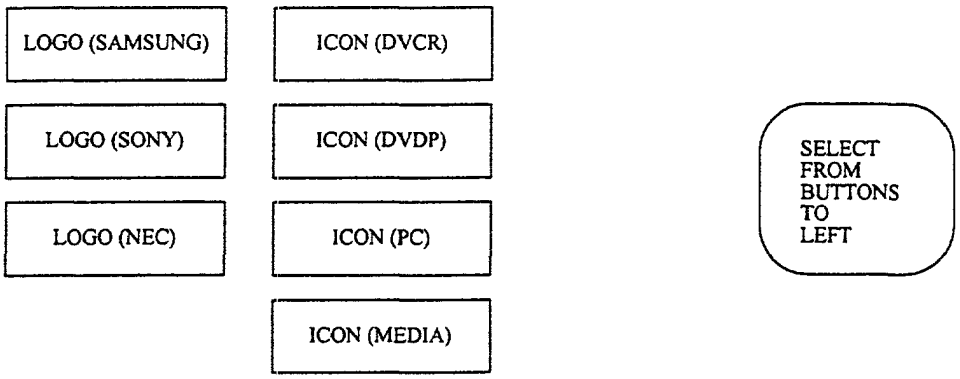


Figure 4, Devices GUI -html

When the DTV powers up the browser runs and displays the DTV top-level default html page which is the local user GUI ***file:///C\dtv/user.html.*** This file is like the front panel control or remote control of the TV. One of the buttons is "devices on home network". This accesses the URL ***http:// dhcp_server.xxx/devices.html.*** (the actual access hot link would not specify the filename.html and would therfore access the machine_name/default.html). This must be a standard URL (hot link) for the home network the details of which (IP address and machine name) are completed by DHCP(client). For the case of the NIU, one physical unit which incorporates the DHCP server and DSS-NIU, there are 2 separate default files accessed by different IP addresses ie machine names.

There is a case for obtaining a standard, top-level-domain 'dot extention' (machine.xxx) from InterNIC for the home network to clearly identify all local hot links and save unnecessary external internet access.

The devices.html file accessed contains entries (buttons) for all available devices in the system. In this model it contains 3 entries (dtv, dvcr and dss). The user can now access all the devices, see devices.html GUI figure 4.

A devices ***user.html*** GUI contains access to a structure of html files for additional purposes eg set-up of device (adjusting brightness levels etc); selecting program material (TV channel); making a profile or

Inventor's: ROBERT WOLFF, KEVIN HARMS, MICHAEL DEACON, RICHARD HUMPLEMAN

Inventor's Signature..... Date 6/13/97

Inventor's Signature..... Date 6/16/97

Witness Name..... Signature..... Date.....

Witness Name..... Signature..... Date.....

macro; checking status; automatic control etc, see figure 5.

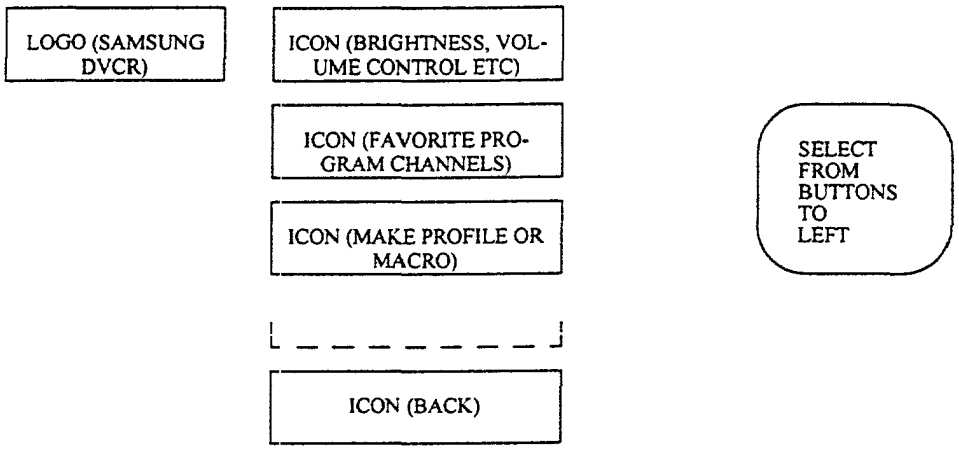


Figure 5, DVCR example User GUI -html

These human driven control actions take place entirely at the device using the GUI also from the device so an industry standardised command and control set isn't necessary. However, standard ascii command set descriptors can be useful for scenarios involving machine driven (automatic) control.

Note the use of LOGO and ICON graphic (GIF) files. These allow the graphical company logo and icon device descriptor to be used. The GIF file names, sizes and compression type can be standardised to:- ***logo.gif, size: 120x40*** and ***icon.gif, size: 120x90.***

GIF compression is chosen because it is lossless, easy to render, and supports transparency and animation (against is the limit of 256 colors).

3.2.1 Device Page

The Device page will list all of the devices on the Home Network with a link to each of the devices' top level HTML pages. The icon and logo image files will all be of a standard size so the list will can be arranged in an orderly manner. Preferably the device's logo would be displayed directly above the device's icon. The logo can act as a link to the device manufacturer's home page if so desired and the icon will be a link to the devce's top level HTML page. The images can be arranged in any manner desired by the NIU Server manufacturer, from as simple as a row and column configuration to a network topology diagram if possible. The NIU Server manufacturer might even allow the user to rearrange the images as they see fit and provide them with an additional text line below each device where the user can enter their own name or description for each device. For instance the user might be allowed to group devices by their location in the home with a name for each location (this kind of feature is entirely implementor driven and is not required).

Inventor's: ROBERT WOLFF, KEVIN HARMS, MICHAEL DEACON, RICHARD HUMPLEMAN

Inventor's Signature ..... Date 6/13/97

Inventor's Signature ..... Date 6/16/97

Witness Name ..... Signature ..... Date .....

Witness Name ..... Signature ..... Date .....

3.2.2 Logo Image Files

A Logo image file is a file containing an image that represents the manufacturer of the device. It would typically contain an image with the name and logo of the manufacturer of the device. In order for the NIU Server to locate the file it must use the name logo.gif The file must also be of a standard size, 120 x 40 pixels, so the list of devices will have a neat, uniform look. Several variations of the logo file may reside on a device with a link to the desired file. The link can be updated over time or based on certain criteria of the manufacturer's choosing. The image may also be animated.

3.2.3 Icon Image Files

An Icon image file is a file containing an image that represents the type of device and potentially its state. It would typically contain an image with a picture of the device or a symbol that represents the type of device. In addition, a model number might be included at the bottom of the image to assist in identification of the device on the Home Network. Several variations of the Icon file may reside on the device with each one representing a potential state. A link to one of the images would represent the current state of the device. To represent the various device states, the manufacturer has the choice of using a variety of symbols, colors, or even animation. (List some graphical examples below). The link may be updated over time or based on certain criteria of the manufacturer's choosing to indicate a change in state. Possible state values may be On, Off, Playing, Stopped, Recording, Rewinding, Forwarding, Searching, Media Inserted, or Media Absent.

The purpose of the Icon image is to provide immediate device state information feedback to the user. In addition, since the Icon images are retrieved from all devices whenever the device list is displayed, there is an immediate indication of the accessibility of all devices on the network. In order for the NIU Server to locate the file it must use the name icon.gif The image must also be of a standard size, 120 x 90 pixels, so the list of devices will have a neat, uniform look. Note that it is up to the device to decide which of it's many ICON's to substitute when asked for ***icon.gif***.

3.2.4 Location of files after discovery phase

Figure 6 is a summary showing the location of files after the device discovery phase. Here each device now has a ***user.html*** file and the DHCP/Name Server has the ***devices.html.*** In an actual implementation these would both be named ***default.html***.

Inventor's: ROBERT WOLFF, KEVIN HARMS, MICHAEL DEACON, RICHARD HUMPLEMAN

Inventor's Signature ..... Date 6/13/97

Inventor's Signature ..... Date 6/16/97

Witness Name ..... Signature ..... Date .....

Witness Name ..... Signature ..... Date .....

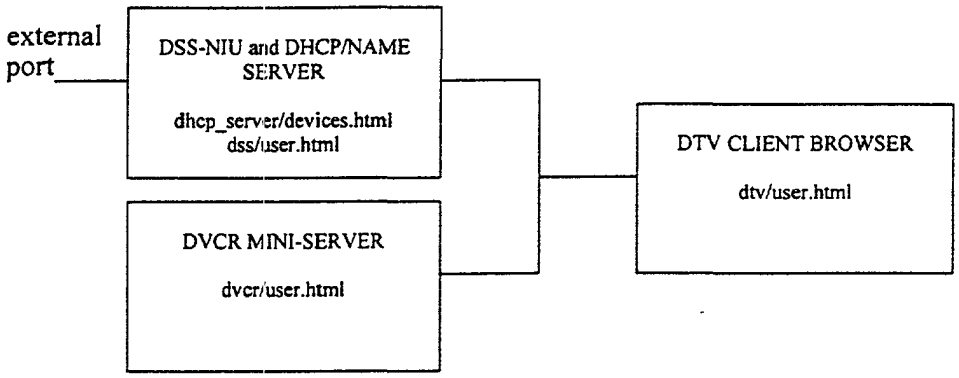


Figure 6, Files Location after Device discovery and selection

3.3 Device set-up and control actions

Use button 'clicks' and form fills to run programs and scripts on the device to make control actions. This is local and proprietary to the device -not performed remotely and therefore doesn't require any standardised 1394 command set. See figure 7 for location of the file and program components.

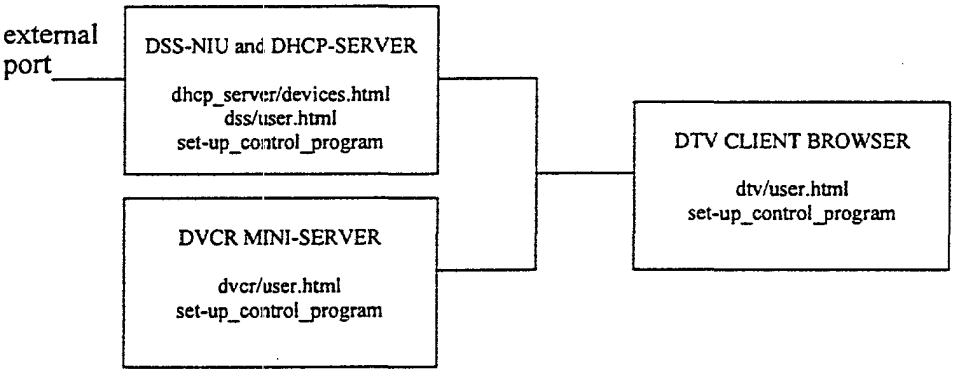


Figure 7, Files/Programs Location after Device set-up and control program

An example is given. The user may wish to change the brightness. On the User html GUI page the user can click on the brightness button. This may bring up another GUI with bright and dim buttons. If the

Inventor's: ROBERT WOLFF, KEVIN HARMS, MICHAEL DEACON, RICHARD HUMPLEMAN

Inventor's Signature..... Date.....

Inventor's Signature..... Date 6/16/97

Witness Name..... Signature..... Date.....

Witness Name..... Signature..... Date.....

user clicks one of these the http server will cause a set brightness control program to run and make the control hardware action. For action local to the DTV the DTV needs server capability -something to interpret the post actions from the browser.

All home network DTV devices need server capability to be able to post actions to control their local hardware. This is important to understand. You can have a browser to pick up local html files and render them to a GUI but this doesn't invoke the http server. Clicking on a button must involve an http access to the local machine name or IP address to invoke the local http server to respond which in turn invokes the local device (eg brightness) control program.

3.4 Program selection

Here additional html files are available to represent the programs audio and video material available for the server device to source. They may be represented directly on the user GUI or down a level. They are represented as ***dss-channel.html*** and ***videofile.html*** for the dss-niu and dvcr respectively. These html files are special as they are not at all static. The device updates the html file based on the dss EPG (Electronic Program Guide) and in the case of the dvcr the tape present in the machine. A program must exist as a go-between the source material content and the html file GUI available to the user. This is called the Dynamic Content_Control_Program. See figure 8 for the location of files and programs.

3.5 Making a Profile or macro

In order to reduce several, often used steps to a few easy steps, the use of macros as shortcuts is beneficial. Macro's are also used to store user Profiles or preferences. A macro is a sequence of commands that is saved in memory and is easily retrieved and executed at will. A macro is created by saving the commands that would normally be executed during a sequence of button pushes or actions by a user within the user interface. The macro is given a name so that it may be easily retrieved at a later time and executed. When the macro is executed it executes the sequence of actions in the macro just as if the user were selecting buttons or performing actions from within the user interface.

A macro is made and stored on the Server for which it is created. Because of the difficulties overcoming possible conflicts and deadlocks with other devices, a macro's scope is limited to the Server on which it is created. That is, it can only execute actions that pertain to the Server on which it was created. Therefore, when a macro is created it must be limited to commands that pertain only to the Server. Profiles/ macros across multiple machines are to be tackled at a later time.

If the macro feature exists then the ***user.html*** contains a profile/macro button for generation on the server. Clicking this button starts the profile or macro recording by the macro generation program.The macro generation program monitors and saves all subsequent hotlinks accessed ('html issues') and 'return actions' for later replay. Ultimately the program results in the creation of another button the named profile or macro with hot-link: ***cgi_bin/macro_for_user.html.***.

Inventor's: ROBERT WOLFF, KEVIN HARMS, MICHAEL DEACON, RICHARD HUMPLEMAN

Inventor's Signature..... Date 6/13/97

Inventor's Signature..... Date 6/16/97

Witness Name..... Signature..... Date.....

Witness Name..... Signature..... Date.....

For more detail on macro generation see apendix.

3.6 Checking status

After a client (html fetch) and server (return-action) handshake the http spec server normally returns a status response code to indicate return-action good or not good (eg 200 returned indicates good and 400 or 300 returned indicates no-good). A bad response initiates a fetch of the ***status.html*** GUI this can include ***icon.gif*** files that indicate graphically what the problem is. See figure 8 for the location of files and programs.Subsequently there can be a 'follow-up' action to access STATUS.HTML files generated and resident on the server with further information and suggested corrective action, see figure 9.

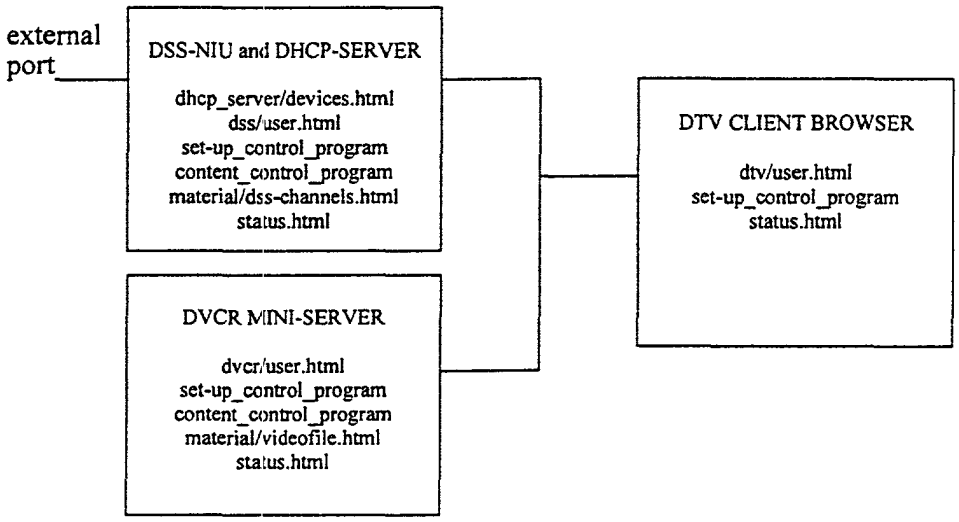


Figure 8, Files/Programs Location after Program selection and status

Inventor's: ROBERT WOLFF, KEVIN HARMS, MICHAEL DEACON, RICHARD HUMPLEMAN

Inventor's Signature..... Date.....

Inventor's Signature..... Date.....

Witness Name..... Signature..... Date.....

Witness Name..... Signature..... Date.....

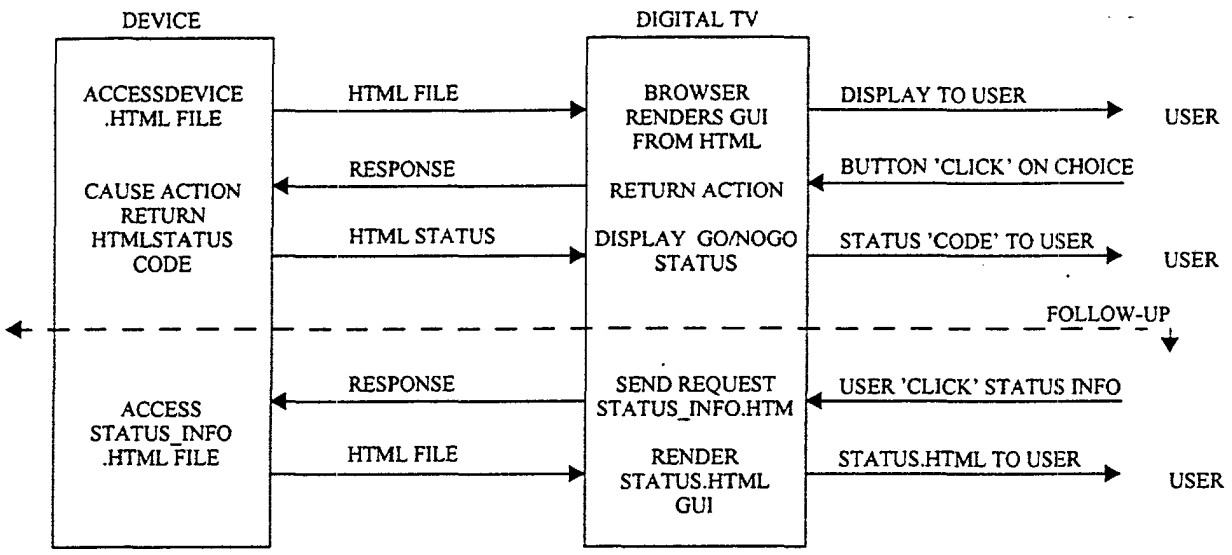


Figure 9, Status code response and Follow-up status access

3.6.1 Follow-Up status checking

One of the inherent challenges associated with a digital network attached to consumer devices is the problem of multiple near-simultaneous access effecting the device. An example of this is that one NIU device could be setting the VCR's clock while someone at a DTV is telling the VCR to play a video while someone at a PC (at work possibly) is telling the VCR to "record channel 7 at 8pm tonight for 1 hour". Each of these activities has a status associated with the action. In the case of "atomic" operations, the status returned is either OK or NOT OK and that is all. In other operations such as rewinding a tape, the initial status may come back as "OK", but a status regarding how far along the rewind is or just if it has completed rewinding is needed via a status page. Another non-atomic example is a more complex one where the VCR has been set to record later tonight, but the user (now at work) wishes to change that setting or delete it altogether. This section describes an innovative way to handle these types of situations through multiple "status pages".

When a client makes a connection to an http device, the client's IP address is given to the device so that the device knows where to send the requested information (HTML files usually). The idea here is to use that IP address as a unique identifier for making custom status files on the device for each client. So, in the above example, there would be three custom status files for "status.NIU_IPaddress.htm", "status.VCR_IPaddress.htm", and "status.DTV_IPaddress.htm". This gives each of those clients the ability to get status from the device which pertains ONLY to their client and no others. A generic "status.htm" file would contain hotlinks to each of the custom status pages as well as some other general status about the

Inventor's: ROBERT WOLFF, KEVIN HARMS, MICHAEL DEACON, RICHARD HUMPLEMAN

Inventor's Signature..... Date 6/13/97

Inventor's Signature..... Date 6/16/97

Witness Name..... Signature..... Date.....

Witness Name..... Signature..... Date.....

device (no tape inserted, 3 record_times currently set, low battery (if appropriate), power reset 20 minutes ago, etc.) So, to be clear though, remember that each device has an actual IP address such as 192.3.27.14 and so making it unique.

3.7 CLO-on-HTML/HTTP/IP/1394

(CLO -Optional Command Language)
An industry standard command set on top of html is useful for Automation and to a lesser extent GUI-less client devices eg Audio Amplifer. This allows software to review html files for content and respond to make control actions. The command language is optional. Home devices and Digital TV work just fine with HTML/HTTP/IP/1394.

A front-panel button press on the Audio Amplifier (involving an external device) is used for GUI control even though no GUI is displayed. Here the external device html is accessed and a parser reviews the html and select keywords. A control program selects the response return depending on the function required.

Hot-links are standardised as commands eg source_select.html, increment_volume.html, bass_level.html, treble_level.html. The device can be now operated locally or remotely and can control other devices.

3.7.1 Automatic Control (eg One touch record)

Automatic control of the home network devices the layout of a set of necessary "control commands" for automation. This list, see table 1, is not an all-encompassing list of commands but an effort is made to make it as complete of a list as possible. This makes the implementors' job easier and clearer. Remember that many devices have no need to implement these commands. Only devices which have such functions or have a need to control other devices which have these functions have the need to implement specific functions from the list.

Table 1: One Touch Record

Name	# Fields	# Buttons	Description
Time_Set	1	1 (set)	hhmmss (Local time)
Record_Time	4	1 (set)	ch#, time(hhmmss), len, mode?
Record_Time_Delete	2	1 (delete)	ch#, time(hhmmss)

Inventor's: ROBERT WOLFF, KEVIN HARMS, MICHAEL DEACON, RICHARD HUMPLEMAN

Inventor's Signature..... Date 6/13/97

Inventor's Signature..... Date 6/16/97

Witness Name..... Signature..... Date.....

Witness Name..... Signature..... Date.....

Table 1: One Touch Record

Name	# Fields	# Buttons	Description
Mode_Select	1	1 (set)	For SP, LP, SLP modes. Should this be a drop-down instead?
Stop			
FF			
Rew			
Play			
Record			

The home network model with separate NIU, VCR and DTV requires some automatic remote control must take place across the network. Simple actions involving 2 devices eg dtv and a server are controlled by the simple html GUI mechanism described above. It is a simple action to select the dvcr GUI and play a tape installed to the dtv, however, setting the dvcr to record involves 3 devices the dss-niu, dvcr and the dtv where the control information entry takes place. One could set-up the dss-niu and then go and set-up the dvcr this would be 2 simple actions. However it is thought that the user would want an automatic one touch record system available.

One touch record takesplace at the dss GUI where a selection is made for a future recording. Somehow the information must be transfered to the dvcr automatically. This is done by the dss server accessing the dvcr GUI automatically and filling in the record information and returning it back to the dvcr.

This action involves an html GUI based command-set as a program in the dss-niu server must be able to scan the dvcr GUI for recognisable key words eg "RECORD_TIME" to enable it to fill in the time. This program needs to know it is accessing the dvcr GUI. Prior to this section the dtv accesses GUI's under human control and the machine had no knowldege of the device.

The One Touch Record (OTR) program is triggered by the server observing a 'record_program' set in the dss GUI. The OTR accesses the dvcr GUI transfers information from the dss GUI to the dvcr GUI and returns the dvcr form to set it to record, see figure 10.

Inventor's: ROBERT WOLFF, KEVIN HARMS, MICHAEL DEACON, RICHARD HUMPLEMAN

Inventor's Signature..... Date 6/13/97

Inventor's Signature..... Date 6/16/97

Witness Name..... Signature..... Date.....

Witness Name..... Signature..... Date.....

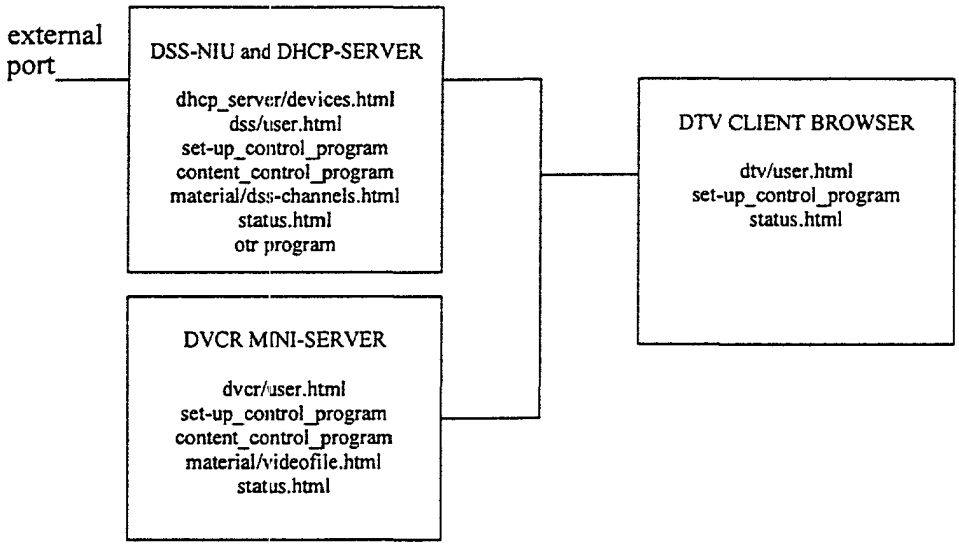


Figure 10. Files/Programs Location with OTR program

3.7.1.1 One Touch Record -How it works

See appendix (B-D) and section 5.0 of this specification.

Inventor's: ROBERT WOLFF, KEVIN HARMS, MICHAEL DEACON, RICHARD HUMPLEMAN

Inventor's Signature ..... Date 6/12/97

Inventor's Signature ..... Date 6/16/97

Witness Name ..... Signature ..... Date .....

Witness Name ..... Signature ..... Date .....

4.0 Conclusions

The industry is currently looking a number of approaches to solving the home-network/Digital TV Interface problem.

4.1 Approaches

All approaches use IEC1883 for transport of isochronous data streams across IEEE1394 or special hardware control of 1394 isochronous streams for non-MPEG2 transport. There are 3 different approaches to controlling the program material streams as outlined below.

4.1.1 Hardware-level command and control (1349AVC/IEC1883/IEEE1394)

This approach without network layer addressing (eg IP) is limited in scope to local cluster control and data flow. The system is fixed to 1394 physical layer and a detailed hardware level command and control specification must be standardised (and is under development) for all devices to use.

Further functional and interconnect expansion can be done with proxy devices converting from network/application layer command and control to the AVC/1883 type command and control.

The approach doesn't work well for a Home Theater DTV which is expected to control everything. A complex GUI would have to be accompanied by the detailed command set for every device making it difficult to design, expensive and quickly obselete.

4.1.2 CAL/IP/IEEE1394

This approach uses the command language CAL on the network layer IP so is much more general and flexible than 3.1.1 and not restricted just to 1394.

The method seems not to solve the problem of DTV GUI availability for all current devices and obelescence regarding future devices and relies on remote control over the network layer IP.

4.1.3 CLO-html/http/ip/1394 (SIPHOT approach)

HTML/HTTP neatly solves the GUI problem by making the DTV a rendering browser and no interface command set is needed for human control of home network devices. An Optional Command Language (CLO) can be used for automatic machine control of devices (rather than human control). This takes the form of specific ascii commands on HTML/HTTP.

One device is nominated to have knowledge of the home network devices to which all devices go initially for device or service selection.

Inventor's: ROBERT WOLFF, KEVIN HARMS, MICHAEL DEACON, RICHARD HUMPLEMAN

Inventor's Signature..... Date 6/13/97

Inventor's Signature..... Date 6/16/97

Witness Name..... Signature..... Date.....

Witness Name..... Signature..... Date.....

5.0 Additional Information

5.1 DIGITAL TV (DTV) definition

The Digital TV (DTV) is an display device (eg like a television CRT) and a box of electronics (STE -Set-top-electronics) containing the home network interface unit (NIU) eg IEEE 1394 digital interface, Digital video/audio decompression, D>A conversion, microprocessor controller to run control software, HTTP/IP protocol, browser software etc, see figure 11.

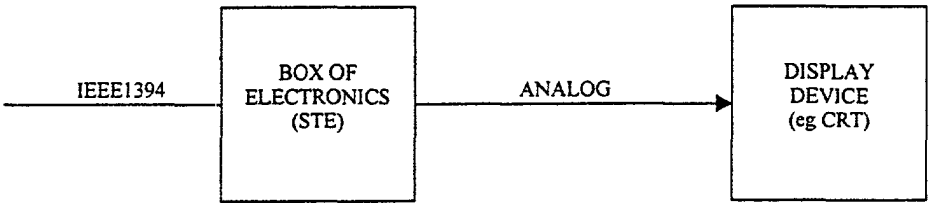


Figure 11, DTV definition

Inventor's: ROBERT WOLFF, KEVIN HARMS, MICHAEL DEACON, RICHARD HUMPLEMAN

Inventor's Signature..... Date 6/13/97

Inventor's Signature..... Date 6/16/97

Witness Name..... Signature..... Date.....

Witness Name..... Signature..... Date.....

5.2 ONE TOUCH RECORD (how it works)

The way this automation works is through the use of HTML forms and the ability for systems to automatically "simulate" user-input of values into form fields and the clicking of a "set" button. An example would be helpful here. Let us take the example of setting a VCR to record a program from 8pm-9pm on Friday night June 6, 1997 on channel 7. In the "normal" HTML GUI, the VCR would produce HTML code which would allow the user to put this information into the form and then click on a button to set this program. A very simple non-graphical version of this GUI would look something like table 2.

Table 2: VCR AUTOMATIC RECORDING SET-UP

	Please fill out forms and click on set
CHANNEL	
DATE (mm/dd/yy)	
START TIME (hhmm)	
START AM/PM	
	SET

This basic form gives all the information necessary to set the VCR to record Friday night. HTTP protocol makes our automated process simple. If each "command" has a unique name, then we implement our automation by doing the following:

Call the "action" /command/<commandname> (in this case /command/record_time) and each field gets a standard name. Then when the automated device wishes to automatically setup a record time as above, it simply prepares the following "POST" command.

```
POST /command/record_time HTTP/1.1
Content-Type: text/plain
Content-Length: 47

channel=07&date=06/06/97&starttime=0800&ampm=pm
```

Generically stated, commands/methods are handled like this:

```
POST /command/standard_commandname HTTP/1.1
Content-Type: text/plain
```

Inventor's: ROBERT WOLFF, KEVIN HARMS, MICHAEL DEACON, RICHARD HUMPLEMAN

Inventor's Signature..... Date..6/13/97.....

Inventor's Signature..... Date..6/16/97.....

Witness Name..... Signature..... Date.....

Witness Name..... Signature..... Date.....

Content-Length: calculated_argumentlength_below

object1=value&object2=value[&objectn=value....]

This simple bit of text is pushed back to the device which is to be set for recording. Then the target device responds with some status information regarding whether or not the requested record time was OK/ valid or not. The status issues are dealt with in a separate section of the document. One of the interesting points to note here is that the "automatOR" device does not have to request any documents in order to program the VCR for recording tomorrow night. As long as the automatOR device knows which command they wish to invoke, they simply gather the information necessary to carry out the command (object=value arguments) and then issue the approriate http POST command and wait for an http status response back.

Indeed this does mean that we carry the initial burden of defining a fair number of commands for general use in order to cover a large percentage of the needs of the device community. This, however, is an acceptable burden as it is not too difficult to define these commands.

Inventor's: ROBERT WOLFF, KEVIN HARMS, MICHAEL DEACON, RICHARD HUMPLEMAN

Inventor's Signature..... Date 6/13/97

Inventor's Signature..... Date 6/16/97

Witness Name..... Signature..... Date.....

Witness Name..... Signature..... Date.....

5.3 DVCR Mini-Server Model

The DVCR-PC functions as a mini-http server, see Figures 12.

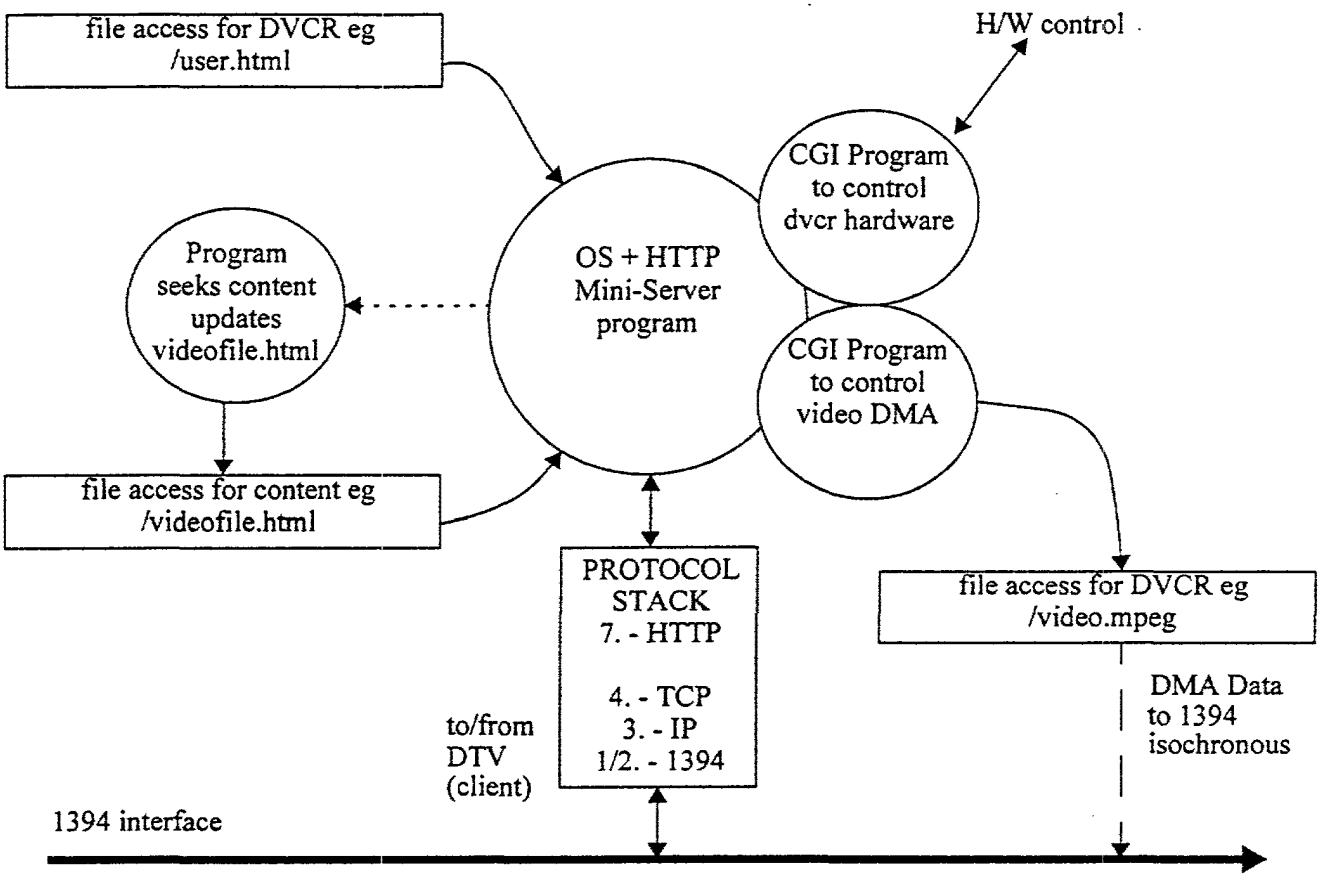


Figure 12. DVCR MINI-SERVER MODEL

5.3.1 DVCR-PC Features

PC andOS with HTTP server capability and TCP/IP PROTOCOL Stack.
HTML user GUI for Samsung DVCR (user.html) and video content on the tape/media (videofile.html)
Accessible MPEG Transport file(s)
MPEG Tpt D(DMA) output to 1394 isochronous from remote http command using CGI-BIN program
Update of content html (videofile.html) by program that can be started on command or by media insertion.

Inventor's: ROBERT WOLFF, KEVIN HARMS, MICHAEL DEACON, RICHARD HUMPLEMAN

Inventor's Signature..... Date 6/13/97

Inventor's Signature..... Date 6/16/97

Witness Name..... Signature..... Date.....

Witness Name..... Signature..... Date.....

5.4 DSS/DVB and DHCP-server Mini-server Model

The DSS-NIU functions as a mini-http server, see Figures 13. The DHCP-server is also shown here though this may reside with any unit on the network.

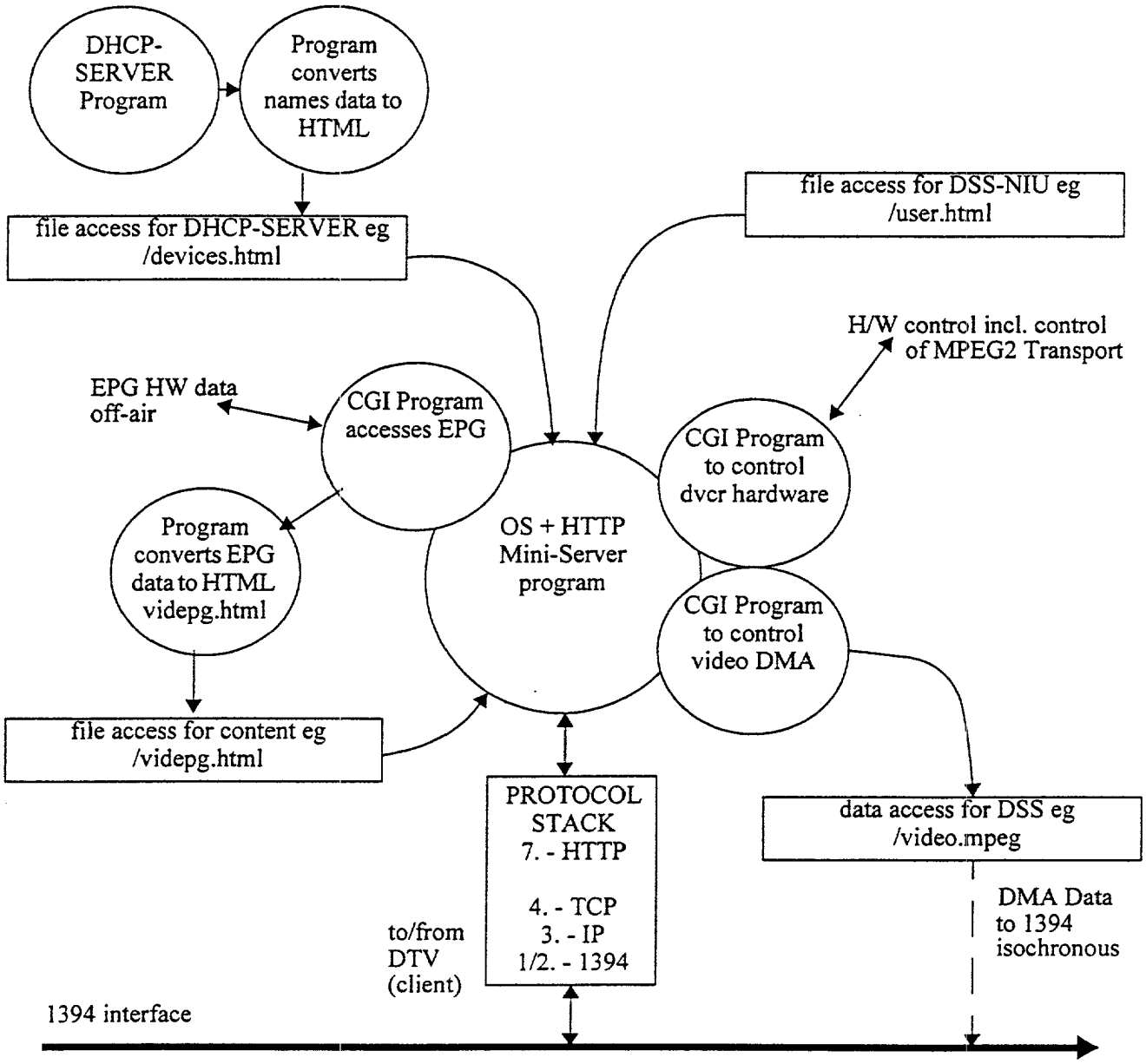


Figure 13.. DSS-NIU MINI-SERVER MODEL

Inventor's: ROBERT WOLFF, KEVIN HARMS, MICHAEL DEACON, RICHARD HUMPLEMAN

Inventor's Signature..... Date 6/13/97

Inventor's Signature..... Date 6/16/97

Witness Name..... Signature..... Date.....

Witness Name..... Signature..... Date.....

5.4.1 DSS/DVB-NIU, DHCP-Server and Internet Features

PC andOS with HTTP server capability and TCP/IP PROTOCOL Stack.
CGI-BIN program to access the hardware

CGI-BIN program to access the off-air EPG hardware and system
Program to convert accessed DSS/DVB EPG data to html GUI form.
Program to take program specific EPG data and convert to html form (eg DISNEY channels only GUI)
File access for videpg.html (video EPG data GUI)
(EPG=Electronic Program Guide).

CGI-BIN program to access the MPEG-2 transport hardware
DSS/MPEG (transport) off-air video program data output to 1394 isochronous from http command (to MPEG2-Transport-over-1394 spec eg IEC1883) possibly using DMA.

Executables to access the DSS/DVB h/w from http command
File access for the dss/dvb-user.html -the GUI shipped with the device for device control.

5.4.1.1 For DHCP server function

DHCP (IP address discovery/assignment) Program
HTML GUI converter/generator of devices present (devices.html)
File access for devices.html

5.4.1.2 For Internet Access Function

Internet access Proxy,
Internet Firewall

Inventor's: ROBERT WOLFF, KEVIN HARMS, MICHAEL DEACON, RICHARD HUMPLEMAN

Inventor's Signature..... Date 6/13/97

Inventor's Signature..... Date 6/16/97

Witness Name..... Signature..... Date.....

Witness Name..... Signature..... Date.....

5.5 DTV Client Model

The DTV client browser human interface is shown in Figure 14. The browser renders GUI (graphical user interfaces) but does not source them. In the demo SIPHOT the source is the Mini-Server (NIU or DVCR). The Browser has 'hot-links' that result in a new GUI and executables can be trigerred to run on the server or client.

Hot links beginning http:// access the mini-server. If the link is also to cgi-bin (or *.asp) then any executable referenced in the html script will be executed on the server. Hot links beginning 'file' are accessed on the client DTV only. Hot links to DTV(self) that are required to perform hardware action, must address the DTV server by having a bonafidi HTTP hot link to the DTV (self). This of course requires that the DTV also has HTTP server capability.

Inventor's: ROBERT WOLFF, KEVIN HARMS, MICHAEL DEACON, RICHARD HUMPLEMAN

Inventor's Signature..... Date 6/13/97

Inventor's Signature..... Date 6/16/97

Witness Name..... Signature..... Date.....

Witness Name..... Signature..... Date.....

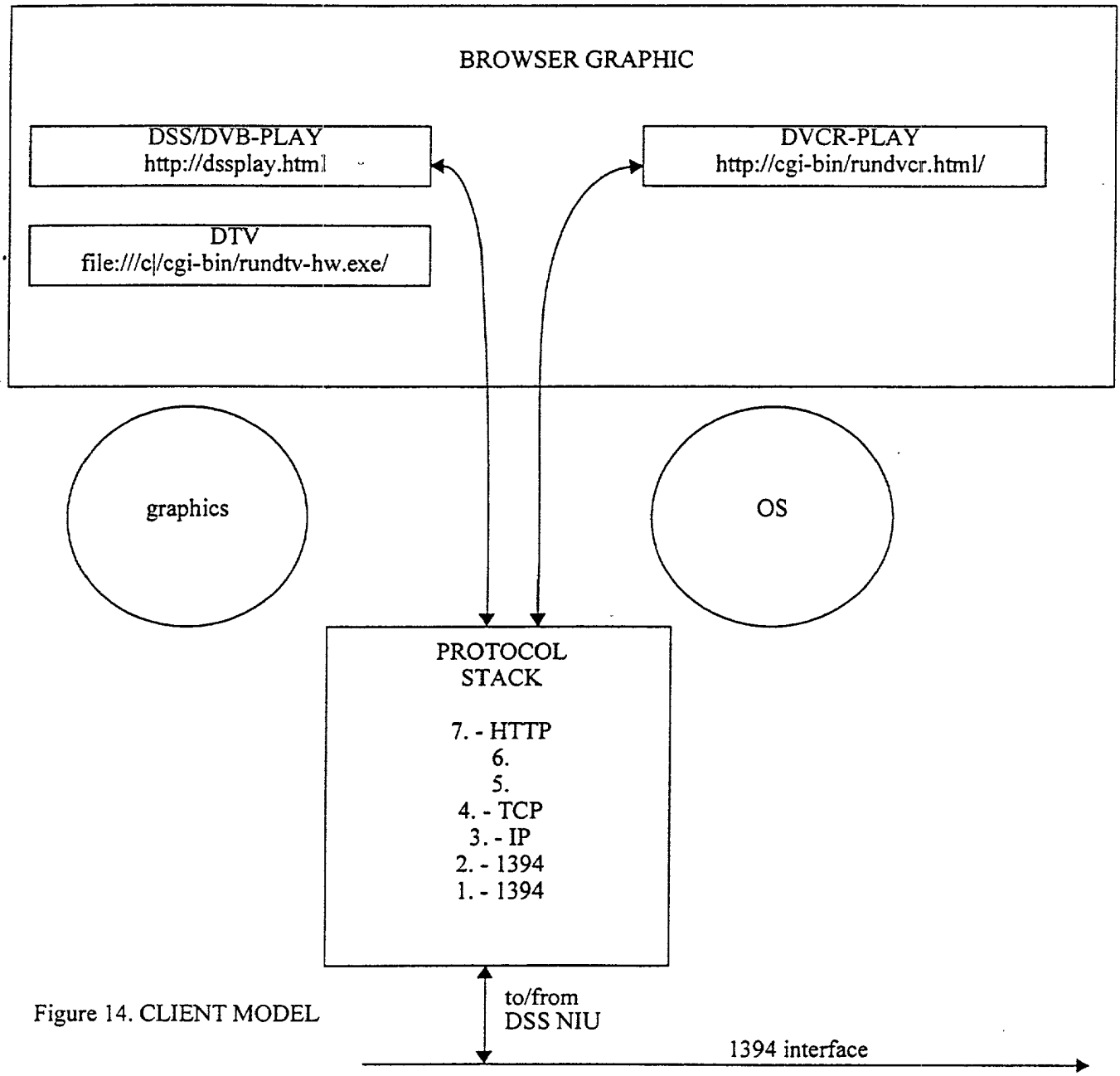


Figure 14. CLIENT MODEL

Inventor's: ROBERT WOLFF, KEVIN HARMS, MICHAEL DEACON, RICHARD HUMPLEMAN

Inventor's Signature..... Date 6/13/97

Inventor's Signature..... Date 6/16/97

Witness Name..... Signature..... Date.....

Witness Name..... Signature..... Date.....

5.6 Self Populating HTTP Server

5.6.1 Problem

From a consumer-viewpoint, all network devices should "simply work" by plugging the device into a network 'outlet' and turning the device on. It should be immediately useable etc. The desire from the user would be to have all network devices have logical names and be able to be used in a singular user interface which is both consistent and easy to use. In addition, Power-users also would like to be able to control, configure, and monitor the network in a consistent and easy-to-use fashion.

5.6.2 Solution

An HTTP server is at its roots simply a collection of files in a heirarchical storage 'tree'. Users gain access to various parts of the 'tree' by browsing through hyperlinks or by being pointed to a specific location in the tree from som external source. At the "root" of the tree, there is a general "welcome" page (home-page user interface). The branches of the tree are usually configured, created, and managed by one or more "webmasters".

In this solution, the custom-programmed webserver self-modifies its tree by creating branches for each device on the network as they are discovered and registered with the central naming authority (DHCP server). This requires cooperation between the self-modifying http server and the DHCP server. After discovering a new device, the HTTP server queries the new device for its capabilities and managability specifics. It then builds a new sub-tree-branch for this new device. Then, the "welcome" page is automatically updated with this new device information "hotlink".

At this point, the user is able to "browse" through the network of devices from one central HTTP server and is able to control, configure, and monitor them since this central HTTP server has intimate knowledge of each device on the network.

One of the more innovative pieces to this self-populating tree is that it has the ability to begin categorizing and indexing available (and unavailable) media for the home. By this, we mean that the server knows at any given time (via polling for status), which devices have which media inserted. For instance, in an advanced (expensive) home setup with multiple AV clusters, Dad puts a DVD movie in one of the DVD players in the bedroom and later that evening, he does not have to remember which DVD player it was inserted into nor if it was DVD, VCR, DVCR, etc. Using the self-populating tree, Dad finds the title he was interested in and it would play regardless of its physical location. This includes the case where Dad's son physically moves the DVD movie to another location for some unknown reason. Such a system requires devices to have insertion and removal notification to the DHCP server in order for it to keep track of which devices have what media. This indexing and categorization technique is also available to older media as well given a little "help". For instance, audio CDROM disks each have a unique ID number which can be associated in a database to the actual artist and title via a database. Even track titles, lyrics, etc. can be obtained by databases. It is our belief that with such a system (and standard) in place that the record label companies will provide the consumer with web-access to such data. This makes the 200+ CD juke-boxes much more compelling to purchase as the user gets to choose their songs to play via the television set or

Inventor's: ROBERT WOLFF, KEVIN HARMS, MICHAEL DEACON, RICHARD HUMPLEMAN

Inventor's Signature..... Date 6/13/97

Inventor's Signature..... Date 6/16/97

Witness Name..... Signature..... Date.....

Witness Name..... Signature..... Date.....

PC.

Welcome to your Home Network
Buttons for Known devices are listed below:

TV
D-VCR
DVD
Den-Computer
Kids-Computer
DSS

Network Configuration/Testing
Network Monitoring

Inventor's: ROBERT WOLFF, KEVIN HARMS, MICHAEL DEACON, RICHARD HUMPLEMAN

Inventor's Signature R Humpleman Kevin Harms Date 6/13/97

Inventor's Signature Robt M. Wolff Michael S Deacon Date 6/16/97

Witness Name..... Signature..... Date.....

Witness Name..... Signature..... Date.....

5.7 (HIPP) Home-network Intranet Protocol Proxy

Home Network browser-browser is another area of intellectual property. First the problems at hand will be presentedfollowed by the solution which contains the IP.

5.7.1 Problem

Due to the uncertainties involved in network standards (or lack thereof), it is unclear which protocol or protocols will become the accepted or most-accepted standard for command and control of consumer electronics devices. There are many proposals recently. The 1394ta has its AVC (Audio/Video Command language) protocol running over IEC1883, http/HTML is another, and CEMA's R4.1 CAL over IP on 1394 is yet another (with or without UDP even). A problem with multiple "standards" could plague the industry for quite some time due to the lack of interoperability it creates. The following idea helps to illeviate this problem.

5.7.2 Solution

Imagine a box which understands IEC1883 protocol on one side and on the other side understands HTTP/HTML and TCP/IP. With such a box and a bit of "help" on the inside, this box can act as a "bridge" or "proxy" for information between two disparate protocols on the same network. "On the same network" does not even imply that it is on the same physical medium as oen of the more useful proxy boxes would proxy between 1394 and X10 systems or other home automation systems. This means that a PC running a web-browser indirectly accesses devices which only understand IEC1883 protocol by using this proxy device as a "translator" of sorts. The user of a web browser is able to graphically "browse" through available resources and devices and also control them. This can be done without the device having any knowledge of HTTP/HTML due to the proxy device.

The implementation of the "box" is simply one or more physical network interfaces with a microcontroller or potentially a full CPU inside which understands both protocols of interest. On the HTTP/HTML side, it generates HTML pages dynamically based upon what is happening on the "other" network.

Inventor's: ROBERT WOLFF, KEVIN HARMS, MICHAEL DEACON, RICHARD HUMPLEMAN

Inventor's Signature..... Date 6/13/97

Inventor's Signature..... Date 6/16/97

Witness Name..... Signature..... Date.....

Witness Name..... Signature..... Date.....

5.8 Using Macros for Shortcuts and Profiles

5.8.1 Purpose

Users will often go through the same sequence of steps during the setup or adjustment of a device, or several users will have different requirements or preferences. If the sequence of steps are repeated often enough or user requirements or preferences differ enough, the use of macros becomes a significant benefit. The use of macros will reduce an awkward sequence of numerous steps to a few easy keystrokes that are easily duplicated at will. Several users may have preferences for brightness, tint, and contrast on a TV for example, or they may have different requirements for their video source.

5.8.2 Functionality

A macro is a sequence of commands that is saved in memory and can be easily retrieved and executed at any time, as often as desired. The macro is created by saving the actions that would normally be executed during user interaction within the User Interface, such as button pushes, data entry, or cursor movement. The macro is given a name so that it may be easily retrieved at a later time and executed. When the macro is executed it executes the sequence of actions in the macro just as if the user were performing the actions from within the user interface, see figure 15.

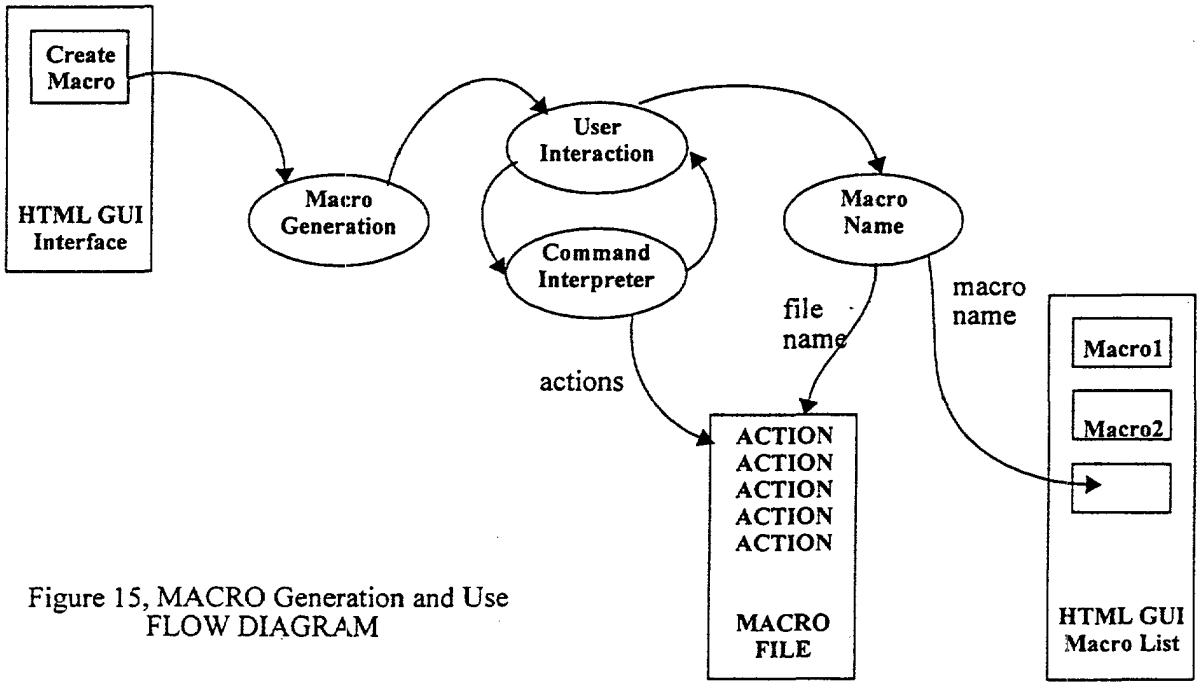


Figure 15, MACRO Generation and Use FLOW DIAGRAM

Inventor's: ROBERT WOLFF, KEVIN HARMS, MICHAEL DEACON, RICHARD HUMPLEMAN

Inventor's Signature..... Date 6/13/97

Inventor's Signature..... Date 6/16/97

Witness Name..... Signature..... Date.....

Witness Name..... Signature..... Date.....

A macro is stored on the device for which it is created. Because of the difficulties overcoming possible conflicts and deadlocks with other devices, a macro's scope must be limited to the device on which it is created. That is, it can only execute local commands. Therefore, during macro creation, only local commands can be enabled. This is a task that is left up to the developer of the macro generation routine.

5.8.3 Is Simultaneous Setup and Control Possible?

Various devices from different manufacturers may exist on a Home Network simultaneously. In order to facilitate convenient setup and control of several devices in tandem, macros may be used. When the macro is executed it executes the sequence of actions in the macro just as if the user were selecting buttons or performing actions from within the user interface. A macro is not limited to storing actions from one user interface, but can be used to store actions from a sequence of menus and various user interfaces.

In a Home Network environment the situation can be made even more complicated by a proliferation of devices that require simultaneous control and by devices that are under the control of several other devices or users (OOPS, Deadlock). (Need to resolve possible conflicts and deadlocks. Macro #1 is going to record from the DSS and gets to a point where it needs access to the VCR, but the VCR is being used by macro #2 that is recording Hawaii Five-0 from the DVD. So while macro #1 waits, macro #2 stops recording from DVD and tries to record from DSS, but macro #1 has control of the DSS.) *

For example, setting up a VCR to record from a DSS. One could leave the DSS on all the time and set the VCR to record at a particular time, but that's an awkward solution. The DSS not only consumes more power and experiences unneeded wear when left on for extended periods, but may be inadvertently switched to a different channel or innocently turned off before the VCR is able to record the desired program.

* - Any failure to acquire resources teminates the macro.

Inventor's: ROBERT WOLFF, KEVIN HARMS, MICHAEL DEACON, RICHARD HUMPLEMAN

Inventor's Signature ..... Date 6/13/97

Inventor's Signature ..... Date 6/16/97

Witness Name ..... Signature ..... Date .....

Witness Name ..... Signature ..... Date .....

Appendix A

Addition to
Draft Specification

A- Page 1

1. Introduction

1.1 Overview

This document proposes a new scheme for Home Network (or Home Theater)based on the popular web browsing technology. All devices which conform to this architecture take the responsibility of controlling all of their own functions(or services it provides) individually and so makes it easy for each device to be in harmony with other devices.

The scheme, described herein, is derived from the internet architecture which adopts client and server model. All devices that provides an additional service to the existing IP/IEEE1394 network is called "Server". And the device that provides display and user interface to consumer is called "Client". Consumer interacts with the client device but services are provided through the web pages in the server.

This is the first approach to transfer Graphic User Interface between devices in home network. No other proposal , till now, provided a method for transferring graphics. In this proposal, all the sources of GUI information(.HTML file) reside in the server device. So every manufacturer can represent its own user interface and graphics. This is one of the many merits that this architecture provides. Another merit is that the client device(user interface to consumer) does not have to send any control command to the server device to which it is accessing (strictly speaking, there are some exception on this.). The client device shows all the functions that the server to which it is accessing can serve and it sends the information only about the response of user, not the control command.

1.2 Normative References

- IEEE std 1394-1995, Standard for a High Performance Serial Bus
- ISO/IEC 13213-1994, Control and Status Register (CSR) Architecture for Microcomputer Buses, Information technology Microprocessor systems.
- IEC1883, Specifications of Digital Interfaces for Consumer Electronics Audio/Video Equipment, HD Digital VCR Conference

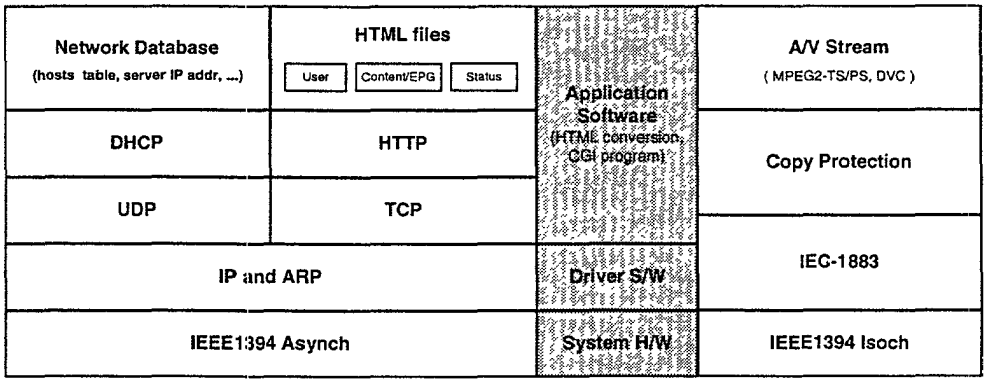
1.3 Definitions, Symbols, and Abbreviations

For this proposal, the following definitions or abbreviations apply.

Client = HTTP client device or process
Server = HTTP server device or process
Configuration Manager = DHCP server which is selected as the result of DHCP contention (In this proposal we use both terms : DHCP Server and Configuration Manager. However it will be clearly defined in the next version of this proposal.)
IP = Internet Protocol
ARP = Address Resolution Protocol
DHCP = Dynamic Host Configuration Protocol
SIPHOT = Samsung IP based HOme Theater

2. Summary Description

Following diagram depicts the protocol stack that we are proposing.



Protocol Stack

Actually, there are three columns in the above figure : Network Management, User Control and A/V Data Flow.

The leftmost column shows the flow of "**Network Management**". The protocol stack for this flow is Network DB/DHCP/UDP/IP&ARP/IEEE1394. Network Management has the role of configuring the network, assigning configuration parameters to each device and providing a directory services to users. The key layer for this column is DHCP(Dynamic Host Configuration Protocol). The protocol is used for several purposes. The first purpose of adopting this protocol is to assign each device an IP address automatically. The IP layer is common to all data link networks as far as concerning to asynchronous control flow. The IP address is the most important part of IP network resources. The second purpose of using it is to provide a directory service to users by collecting information about the network configuration during the network initialization (or device discovery) process. Using the result of this collected information, it is possible for users to control all devices in a well-defined way.

The second column from the left side shows the flow of "**User Control**". The protocol stack for this flow is HTML DB/HTTP/TCP/IP&ARP/IEEE1394. HTML files and the response-action of users are transferred through this flow. The side which provides HTML files is called server and the side which provides user action is called client. This is the way which is already used in Web browsing technology. The most important feature of this Client - Server architecture is that it has object-oriented mechanism. The server is an object and the server provides the way of how to use its own services in HTML format(it is called "method" in object-oriented language). This architecture alleviate the client from the burden of knowing all the functions of the server to which it is accessing.

The rightmost column shows the flow of real "**A/V data stream**". The protocol stack for this flow is AV stream/Copy Protection/IEC1883/IEEE1394. After receiving the command from user or from devices, the device starts to send or receive A/V stream using the procedure described in the IEC1883.

3. Architecture

3.1 IEEE 1394 Generics

3.1.1 Cable Physical Layer

All cable physical layer implementations compliant with this standard shall meet performance criteria specified in the IEEE1394 standard.

3.1.2 Link Layer

All link layer implementations compliant with this standard shall meet performance criteria specified in the IEEE1394 standard.

3.1.3 Transaction Layer

All transaction layer implementations compliant with this standard shall meet performance criteria specified in the IEEE1394 standard.

3.1.4 Serial Bus Management

All serial bus management layer implementations compliant with this standard shall meet performance criteria specified in the IEEE1394 standard. In addition, several Control and Status Registers and Configuration ROM's which will be defined in the future version of this specification shall be implemented.

3.2 TCP/IP and Others

3.2.1 IP and ARP over IEEE 1394

IP and ARP, network layer protocols, are required in this proposal. Currently a working group in IETF is discussing on the transmission of IP & ARP diagram over IEEE1394. That result can be adopted as the protocols in this layer (Several CSR and Configuration ROM will be required.). The future version of this specification will include which portion of it can be adopted and which portion of it is not required for this specification.

3.2.2 Transport Layer

TCP and UDP, transport layer protocols, are required in this proposal. All of the graphic user interfaces are transferred by TCP. However, UDP is also required for the network configuration process.

3.2.3 Security Layer

This layer is required for the both of isochronous and asynchronous transfer. For the isochronous transfer, the result from CPWG/DTDG can be used to prevent user from recording received stream iteratively. For the asynchronous transfer, there is nothing defined yet. However there should be some kind of mechanism for authentication. This will be defined in the future version of this specification.

3.2.4 Application Layer

Two protocols, HTTP and DHCP, are required for this layer. Any other protocols are optional. HTTP server process is required in the case of server and HTTP client process is required in the case of client. All server devices should have DHCP client process and all client devices should have DHCP server process. In addition, the device which has the DHCP server capability should implement DHCP server arbitration protocol which will be

defined in the future version of this specification. The server device which has the capability of the Internet connection should have the DHCP server capability.

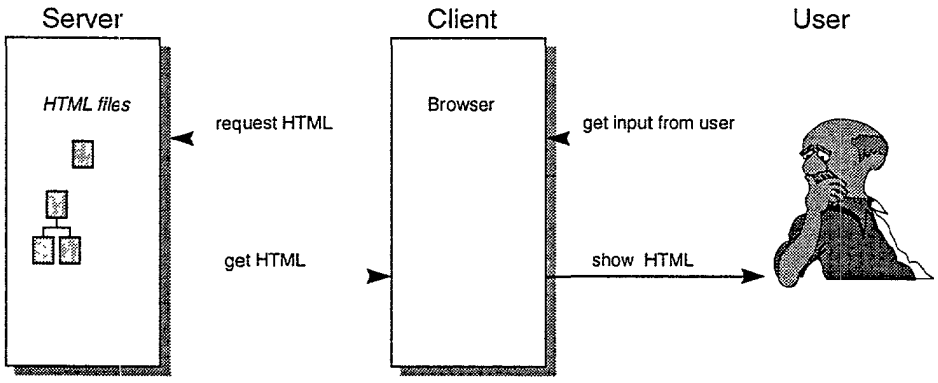
All server device, if it has the content guide, should implement "Content Guide to HTML" conversion software. The content guide means EPG or media content guide. How to convert the content into HTML is vendor dependent.

There is one thing more in this layer: the Database. The database includes HTML files, network database and A/V stream. These are the target objects to be transferred between devices or information sources for them.

4. Device Classification

4.1 Overall System Model

There are normally two types of system : Client and Server. A device which provides a service or functionality to users and provides its control method in HTML form is called "server". A device which accesses the server HTML page, shows the content to users and transfers the response information from users is called "client". Internet DTV is a good example of client device. DVCR and DBS set top box are examples of servers. The following diagram shows the concept of server and client.

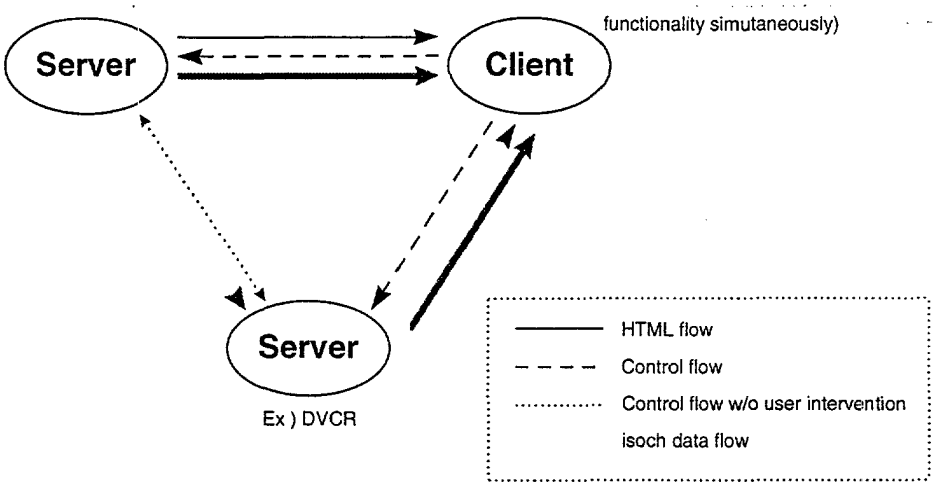


Client and Server

However during the home theater configuration phase, there exists another device called a "Configuration Manager". Actually, it is a DHCP server. It allocates a private IP address (in contrast to public IP address. Note: In IPv6 environment, all devices in home will have a unique IP address) to each device connected to the home theater and assigns a unique name for each. The configuration manager collects information about the network configuration and generates a HTML file which contains hyperlinks to the top HTML page of each device.

To gain a service from each device connected into the home theater, users access the HTML page of the server and click on the corresponding button which they want to be served. For example, users can click a button ,"play" of the DVCR to see video tape installed in the DVCR. Then the DVCR runs a process that drives a capstan motor and transmits A/V streams through the digital interface attached. One way to implement this interaction is to use a CGI program.

Following is another diagram that shows the relationship between server and client and shows the flow of data.



Control and Data flow

Because each server provides its service to users(watching client device) through HTML which is located in it, there is no need to send a standardized control command from one device to another. The only exception is the transaction between server devices without the intervention of a human. In this case, there should be some kind of mechanism to transfer control between them. This is called CLO(command language optional) in this proposal.

4.2 Client Display Model

Requirements for Client Display Device

- Display Capable
- DHCP Server Capable
- DHCP Server Arbitration Algorithm
- Always powered on for networking services
- HTML GUI (Optional icon and logo image files)
- Browser – HTML Rendering
- HTTP Client
- HTTP Server for Local H/W Control
- Local H/W Control Program
- MPEG-2 Transport Stream (TS) over 1394
- MPEG-2 audio/video graphics
- Hosts Table
- Device Page Generator (Uses Hosts Table)
- Isochronous Bit Stream Selection Page (Need way of displaying MPEG bit streams from non-IP devices) – Generate HTML page of all isochronous bit streams on network for user selection.

4.3 Server Model

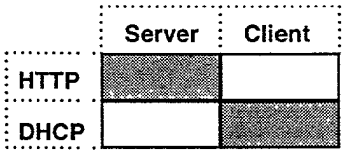
(1) Common Requirements for Server Device

- HTTP Server
- HTML GUI (Optional icon and logo image files)
- Local H/W Control Program
- Content Guide (optional)
- Content Guide to HTML Conversion Program (if Content Guide is present)
- MPEG-2 Transport Stream (TS) over 1394 (Optional for MPEG-2 devices)
- Macro capability (optional, need command set)

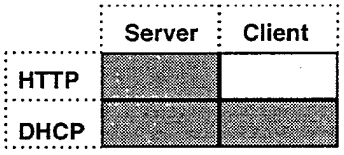
(2) Additional Requirements for Internet-NIU Device (OPTIONAL)

- DHCP Server (higher priority than Display Devices)
- Device Page Generator (Uses Hosts Table)
- Internet Proxy Server
- Gateway Capability
- Firewall

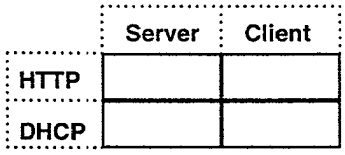
Following diagram shows the protocol requirements for the 3 types of devices.



(a) server



(b) Internet-NIU server

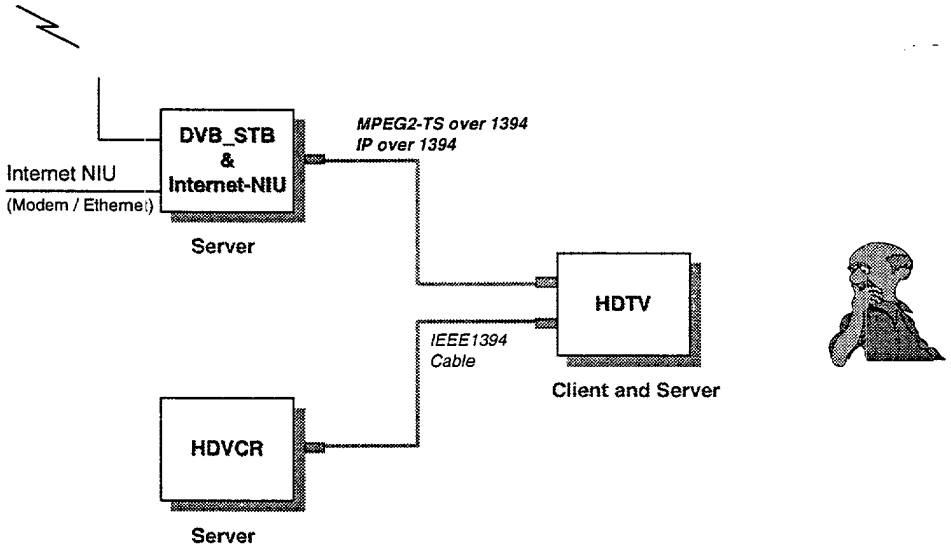


(c) client / display

Protocol Requirements for 3 types of devices

4.4 Example Home Theater

Here is an example home theater. Three devices - HDTV, DVB_STB with Internet NIU and HDVCR - construct a simple home theater.



Example Home Theater

The HDTV is a client device and it also has the server capability for its own functions (ex: brightness/sharpness/.. control, screen configuration and so on).

The protocol stack for HDTV may be as follows.

< HDTV Diagram >

The DVB_STB with Internet NIU is a server device and it is also a configuration manager(DHCP server and Name server). The protocol stack for DVB_STB with Internet NIU may be as follows.

< STB and Internet-NIU Diagram >

The HDVCR is a server device which has the capability of transmitting and receiving an A/V transport stream. The protocol stack for HDVCR may be as follows.

< HDVCR Diagram >

See the following sections for more details.

5. Network Management

Network Management performs several important functions for the network, such as network configuration, generation of devices link page, IP address assignment and the supply of some additional utility for maintaining the network.

5.1 DHCP Server Contention

Sometimes there may be several DHCP servers in home theater. In the case of multiple DHCP servers, there should be some kind of arbitration protocol(or DHCP server to DHCP server communication protocol) between DHCP servers and there should be only one active DHCP server in home theater.

Without this algorithm, there can be a situation that there are several groups of devices and in each group one DHCP server manages all devices belonging to that group. As the result, two devices which belongs to different group cannot communicate each other. To avoid this situation, it is needed to introduce some kind of contention algorithm or some kind of DHCP server to DHCP server communication protocol.

This algorithm will be added in the future version of this proposal.

5.2 IP Address Translation(IPv4 environment only)

With the proliferation of TCP/IP technology worldwide, today's Internet has confronted with several problems: shortage of IP address, too much overhead on router, etc. Several ways of solving this kind of problems have been suggested by many internet engineers and they are working on it. The long term solution to these problems can be found in the widespread deployment of IP Next Generation(IPng or IPv6). However, while the Internet community waits for IPng, IPv4 will need to be patched and modified so that the Internet can continue to provide the universal connectivity we have come to expect.

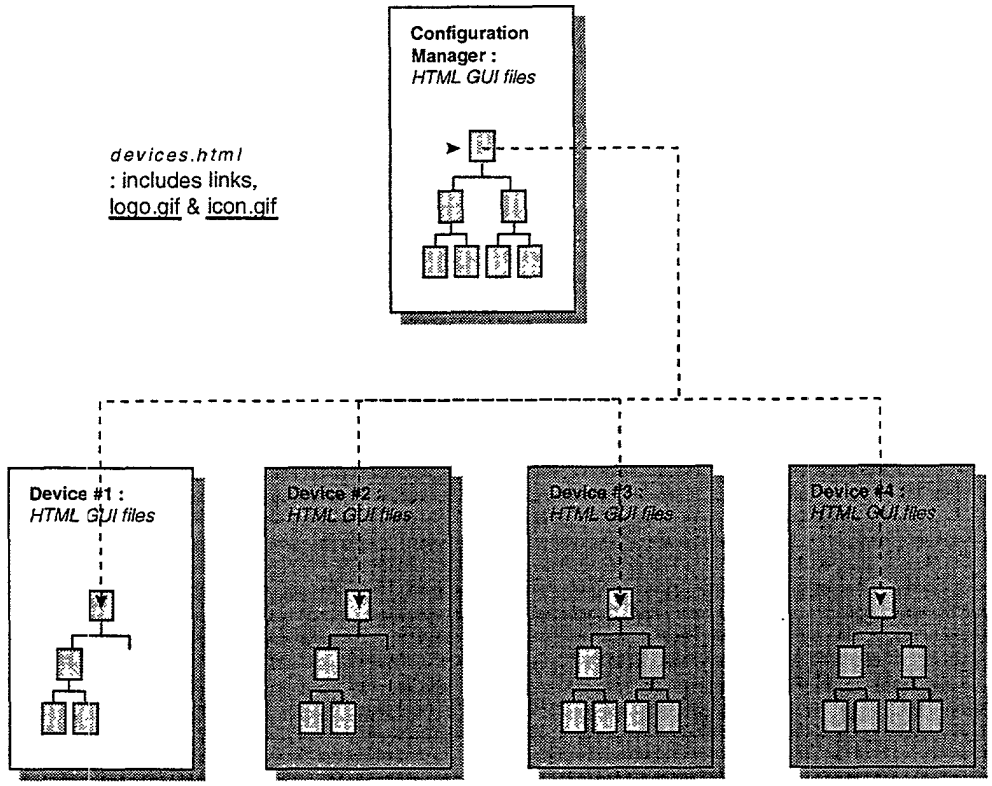
This section can be filled in by the co-work with many Internet experts.

5.3 Device Discovery

5.3.1 Device Discovery Process

The Configuration Manager (DHCP Server) generates a list of all devices on the Home Network during the Device Discovery stage and assigns each an IP address. These results are collected within a file along with the name of each device. These results are then used to create a list of all devices on the Home Network that can be accessed by the user. This list is in the form of a HTML file and is located somewhere within the DHCP Server's HTML GUI file structure(DHCP/devices.html). A program is used to create this HTML file, converting each line in the device list(It is hosts file.) to a hyperlink and adding it to the HTML file. In the figure below, the list is placed at the top level page of the DHCP Server's GUI. This list is known as the Device Link page. It is up to the manufacturer of the DHCP Server to generate a Device Link page that is user friendly and logically located within the DHCP Server's GUI so that it can be easily accessed by the Home Network user. It is also up to the DHCP Server manufacturer to create a link that is easily understood so

the user can easily assess to which device a particular link refers. To assist in this endeavor, each device may optionally contain a logo.gif and icon.gif file. These images are used as links to the device manufacturer's Web page and the device's top level HTML GUI page, respectively. These images are placed on the Device Link page in a manner that makes it easy to access all the devices on the Home Network.



Device Link Page

5.3.2 Device Information File

Each device will contain a file called device.txt specifying the pertinent attributes of the device (This information may already be available in some other form defined by another standards organization such as 1394TA's specification for Self Describing Devices, in which case we will remove this requirement and use the latter form) Each attribute will be represented by a name and value within the file. This information is available to other devices on the network in order to facilitate interoperability. The DHCP Server, for example, may refer to this file during Device Discovery in order to correctly set communication parameters for the device. The attributes are also used to supplement the image files. The information required to implement this specification is listed in the table below. Other fields may be added to address the needs of other specifications.

Name	Length	Description
DeviceManufacturer	20 chars	Device manufacturer's name
DeviceName	30 chars	
DeviceModel	20 chars	Device model number

ManufacturerURL	40 chars	Device manufacturer's home page URL
DeviceType	20 chars	Device category

5.3.3 Logo Image Files

A logo image file is a file containing an image of the device manufacturer's name and logo. In order for the DHCP Server to locate the logo file it is always given the name logo.gif. The image is also a standard size, 120 x 40 pixels, so the list of devices will have a neat, uniform look. Several variations of the logo file can reside on a device with a link to the desired logo file. The link can be updated over time or based on certain criteria of the manufacturer's choosing. The image may also be animated as is the Samsung logo in the example below.

5.3.4 Icon Image Files

An Icon image file is a file containing an image representing the type of device and its state. It typically contains an image with a picture of the device or a symbol representing the type of device. In addition, a model number may be included at the bottom of the image to assist in identification of the particular device on the Home Network. Several variations of the Icon file may reside on the device with each one representing a potential state. A link to one of the images represents the current state of the device. To represent the various device states, the manufacturer has the choice of using a variety of symbols, colors, or even animation. (List some graphical examples below). The link is be updated over time or based on certain criteria of the manufacturer's choosing to indicate a change in state. Example state values are On, Off, Playing, Stopped, Recording, Rewinding, Forwarding, Searching, Media Inserted, and Media Absent. The purpose of the Icon image is to provide immediate device state information to the user. In addition, since the Icon images are retrieved from all devices whenever the device list is displayed, there is an immediate indication of the accessibility of all devices on the network. In order for the DHCP Server to locate the icon file it is always given the name icon.gif. The image is also a standard size, 120 x 90 pixels, so the list of devices will have a neat, uniform look.

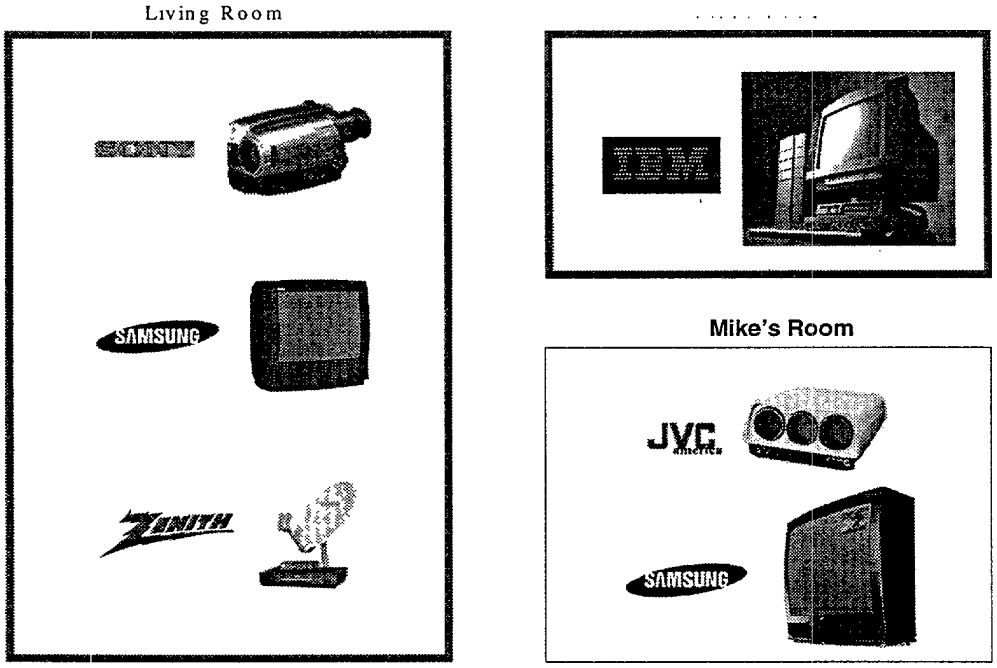
Default Icon Image Files

The DHCP server may contain a set of default icon images representing the various types of devices that can be connected to the Home Network. If an icon image is not present on a particular device the DHCP Server may use one of its default icon images or use the DeviceName attribute to create a hyperlink for the device. The DHCP Server knows the type of device it is referring to by looking at the DeviceType field within the device.txt attribute file located on the device. The DeviceType field specifies the device category such as VCR, DVD, DTV, Camcorder, PC, Security System, etc. for the particular device (We need to locate or specify, a list of device categories).

5.3.5 Device Page

The Device page lists all of the devices on the Home Network with a link to each of the devices' top level HTML pages. The icon and logo image files are a standard size so the list can be arranged in an orderly manner. The device manufacturer's logo is displayed along

with the device's icon. The logo can act as a hyperlink to the device manufacturer's home page if so desired and the icon is a hyperlink to the device's top level HTML page. If they are not available a standard device icon or the string from the DeviceName attribute field may be displayed in their place and used as the hyperlink. The images are arranged in any manner desired by the DHCP Server manufacturer, from a basic row and column configuration to a network topology diagram if possible. The DHCP Server manufacturer may even allow the user to arrange the images in groups according to their placement in the home and provide the user with an additional text lines to describe the groupings and/or devices. For instance, the user may be allowed to group devices by room with a name for each room. The following is an example of this style of Device page:



Example Device Link Page

6. User Control

6.1 Device Control

This section deals with directly controlling a remote device through the use of an HTML GUI that is downloaded directly from the device.

6.1.1 Introduction (Informative)

When the World Wide Web technology was first introduced, it was HTML-based, static and web server-driven. Users at work with web browser could only select what they wanted to see in the next time. In a sense, it supported one-way communication only. However, today's web pages support dynamic, two-way communication. There are many kinds of technology which make it possible. They are CGI program, Java, JavaScript and so on.

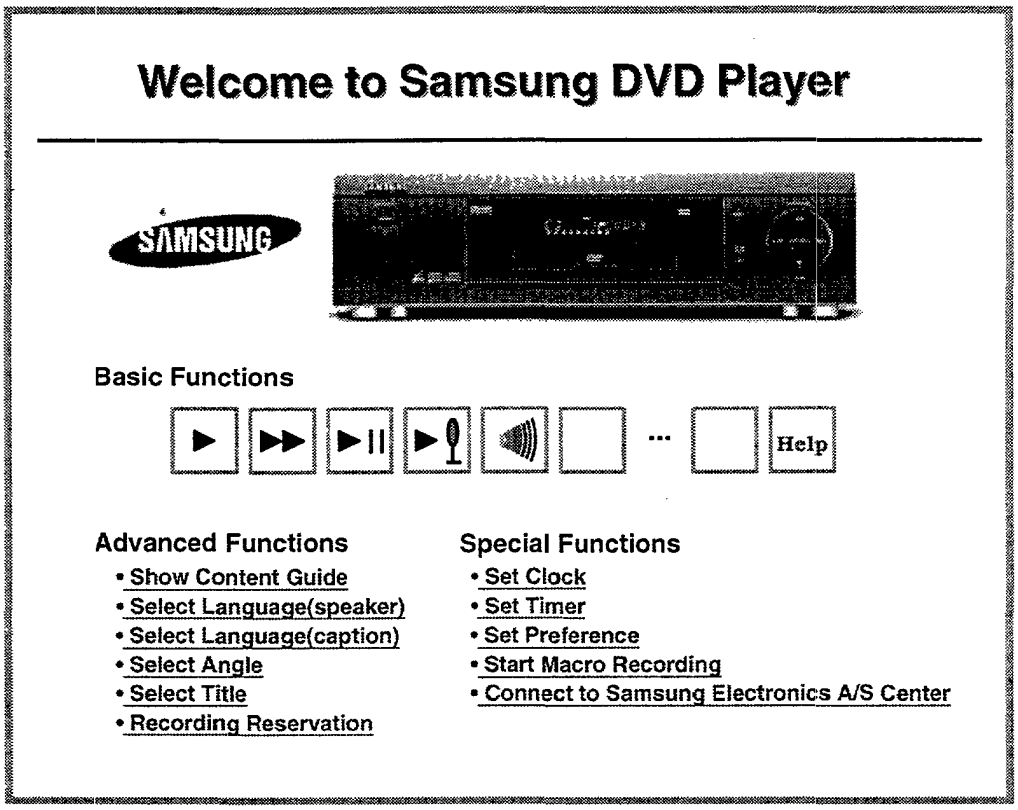
The HTML files that are supplied by server devices can be implemented in various ways. Therefore the manufacturer of those devices can represent their own characteristics and show their own design skill. However some restrictions will be specified in the future version of this proposal. This is inevitable because of the browser capability. If the manufacturer prepares several versions of his HTML files, he can use advanced technology for the better display.

Actually there are two kinds of situations when a user controls devices through web browser. If the service which is requested by the user can be completed by the device to which the user is accessing, there is no problem. One example for it is to click on the button, "play video", which is embedded in the HTML file which is supplied by DVCR. This kind of service will be executed immediately. However there is another situation which cannot be executed without the co-work with other devices. One example for it is to make a reservation for recording a certain program. In this case the server device, DVCR, cannot record a program without the help of Set Top Box (or DTV which has a tuner module). There can be many solutions. One solution for it is to use control command such as AV/C or CAL. The other solution is to generate a notification message to the user : "Now I am ready to record a program. However I can't complete that job by myself. Please Select a tuner device and make a channel selection." There can be another idea that we can use. For example, it is possible for us to define another standardized data structure for each device. So any device can access that data structure of the other device without the intervention of human. However it is not quite a practical idea. It is another big job as you see the case of AV/C command language (Actually we don't have to define whole portion of command set. All we have to do is to define reduced version of command set/data structure which is associated with multiple devices. However it is not a small job, because we have to consider all the cases that can occur. Adopting that kind of data structure will also slow down the communication speed than using already developed, standardized command language). Therefore it is more desirable to use an already developed command language.

Though this proposal may take advantage of some of the features of a command language such as AV/C or CAL, once one is selected for use on Home Theater it is not dependent on any one of them.

6.1.2 Example Control Scenarios

Following picture shows an example top level page of DVD server device.



Example top level page of DVD server device

This section will be added in the future.

(1) Video Playback
(2) Recording Reservation
(3) Selection from Program Guide

6.2 Converting Program Content to HTML
6.2.1 EPG to HTML
6.2.2 Audio Content to HTML
6.2.3 Video Content to HTML

6.3 Macros
6.3.1 Purpose of Macros

Users will often go through the same sequence of steps during the setup or adjustment of a device, or several users will have different requirements or preferences for settings. If the sequence of steps are repeated often enough or user requirements or preferences differ enough, the use of macros becomes a significant benefit. A macro is a sequence of commands that is saved in memory and can be easily retrieved and executed, as often as

desired. The use of macros will reduce an awkward sequence of numerous steps to a few easy keystrokes that are easily duplicated at will. Several users may have preferences for brightness, tint, and contrast on a TV for example, or they may have different requirements for their video source. Macros allow a way for individual users to save their own personal set of preferences for a particular device and then to quickly and easily re-establish those same settings at any time in the future.

6.3.2 Types of Macros

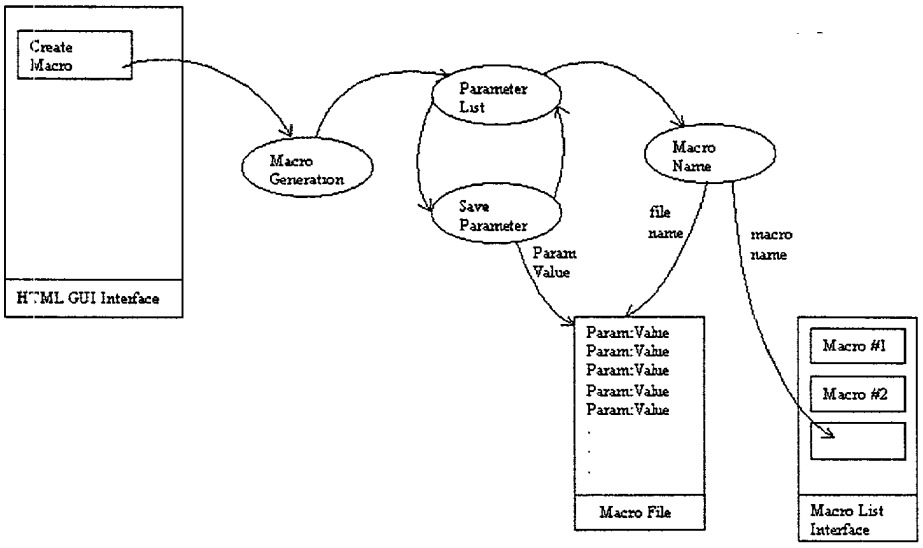
Macros can be generated in two ways depending on the design of the device software and user interface:

- Save the actual values of device parameters (Preset type)
- Save a sequence of button selections and actions specified by the user (Player Piano type)

Depending on the design of the software and the implementation of the user interface it may be possible to implement one or both types. The Presets type can be used whenever parameter values can be queried and set. The Player Piano type can be used when parameter values are entered through the user interface or when a given sequence of user actions will always return the same result. The Player Piano type cannot be used if the result of a given sequence of user actions is dependent on conditions prior to the sequence being executed.

(1) Preset Macros

The Preset macro is created by saving the current values for certain device parameters. When creating the macro, the parameters may be either specified by the user or may be some or all of the configurable device parameters. Likewise, the values may be those entered by the user or the current device settings for the selected parameters, depending on how the manufacturer chooses to design the macro user interface. The macro is given a name so that it may be easily retrieved at a later time and executed. When the macro is executed it issues the appropriate commands to set the chosen parameters to their assigned values.

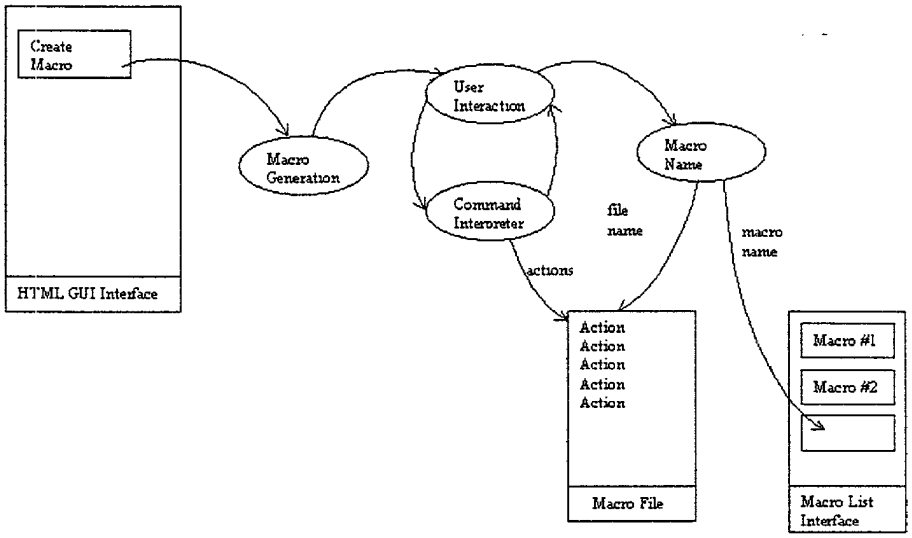


Preset Macros

A macro is stored on the device for which it is created. Because of the difficulties overcoming possible conflicts and deadlocks with other devices, a macro's scope must be limited to the device on which it is created. Therefore, during macro creation, only parameters for the local device can be saved; the macro cannot set parameters for remote devices.

(2) Player Piano Macros

The Player Piano macro is created by saving the actions that would normally be executed during user interaction within the User Interface, such as button pushes, data entry, or cursor movement. The macro is given a name so that it may be easily retrieved at a later time and executed. When the macro is executed it executes the sequence of actions in the macro just as if the user were performing the actions from within the user interface.



Player Piano Macros

(3) Simultaneous Setup and Control

Various devices from different manufacturers may exist on a Home Network simultaneously. In order to facilitate convenient setup and control of several devices in tandem, macros may be used. When the macro is executed it executes the sequence of actions in the macro just as if the user were selecting buttons or performing actions from within the user interface. A macro is not limited to storing actions from one user interface, but can be used to store actions from a sequence of menus and various user interfaces.

In a Home Network environment the situation can be made even more complicated by a proliferation of devices that require simultaneous control and by devices that are under the control of several other devices or users (OOPS, Deadlock). (Need to resolve possible conflicts and deadlocks. Macro #1 is going to record from the DSS and gets to a point where it needs access to the VCR, but the VCR is being used by macro #2 that is recording Hawaii Five-0 from the DVD. So while macro #1 waits, macro #2 stops recording from DVD and tries to record from DSS, but macro #1 has control of the DSS.)

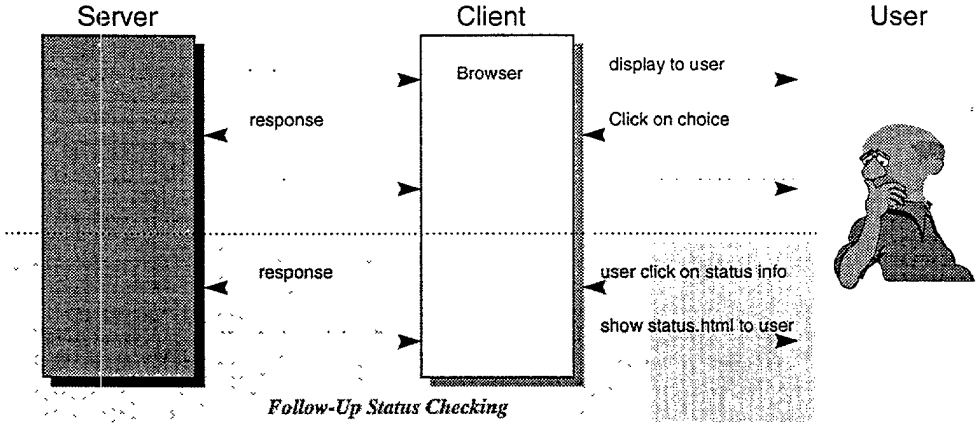
For example, setting up a VCR to record from a DSS. One could leave the DSS on all the time and set the VCR to record at a particular time, but that is an awkward solution. The DSS not only consumes more power and experiences unneeded wear when left on for extended periods, but may be inadvertently switched to a different channel or innocently turned off before the VCR is able to record the desired program

6.4 HTML Command Status

6.4.1 Status Checking

After a client(html file fetch) and server(return-action) handshake, the http server normally return a status response code to indicate return action successful or not (The server returns the response code of 200 when return action was successful. It returns 400 or 300 when

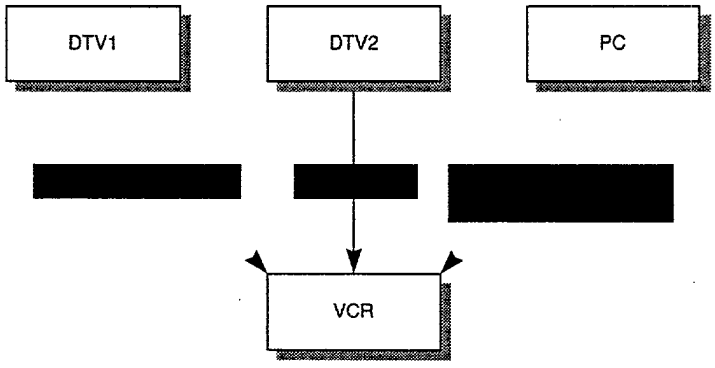
failed.). A bad response initiates a fetch of another status.html file which can include icon.gif files that indicate graphically what the problem is. In this case, status.html file is generated by the server and located in the server. It can also include the suggested corrective action. The following diagram illustrates this mechanism.



. Status code response and Follow-up status access

6.4.2 Follow-Up Status Checking

One of the inherent challenges associated with a digital network which consists of consumer device is the problem of multiple, simultaneous accesses to one device. This is something like the critical section problem in the field of parallel computing. An example of this is that one DTV(DTV1)can be setting the VCR's clock while someone at another DTV(DTV2) is telling the VCR to play a video while someone at a PC(at work possibly) is telling the VCR to "record channel 7 from 8 PM tonight for 1 hour".



. Multiple, simultaneous accesses to one device.

Each of this activities has a status associated with the action. In the case of "atomic" operations, the status returned is either OK or NOT OK and that is all. However in other cases such as rewinding a tape, the initial status may come back as "OK", but a status regarding how far along the rewind is or just if it has completed rewinding is needed via a

status page. Another non-atomic example is a more complex one where the VCR has been set to record later tonight, but the user(not at work) wishes to change the setting or delete it altogether. This section describes an innovative way to handle these types of situations through multiple "status pages".

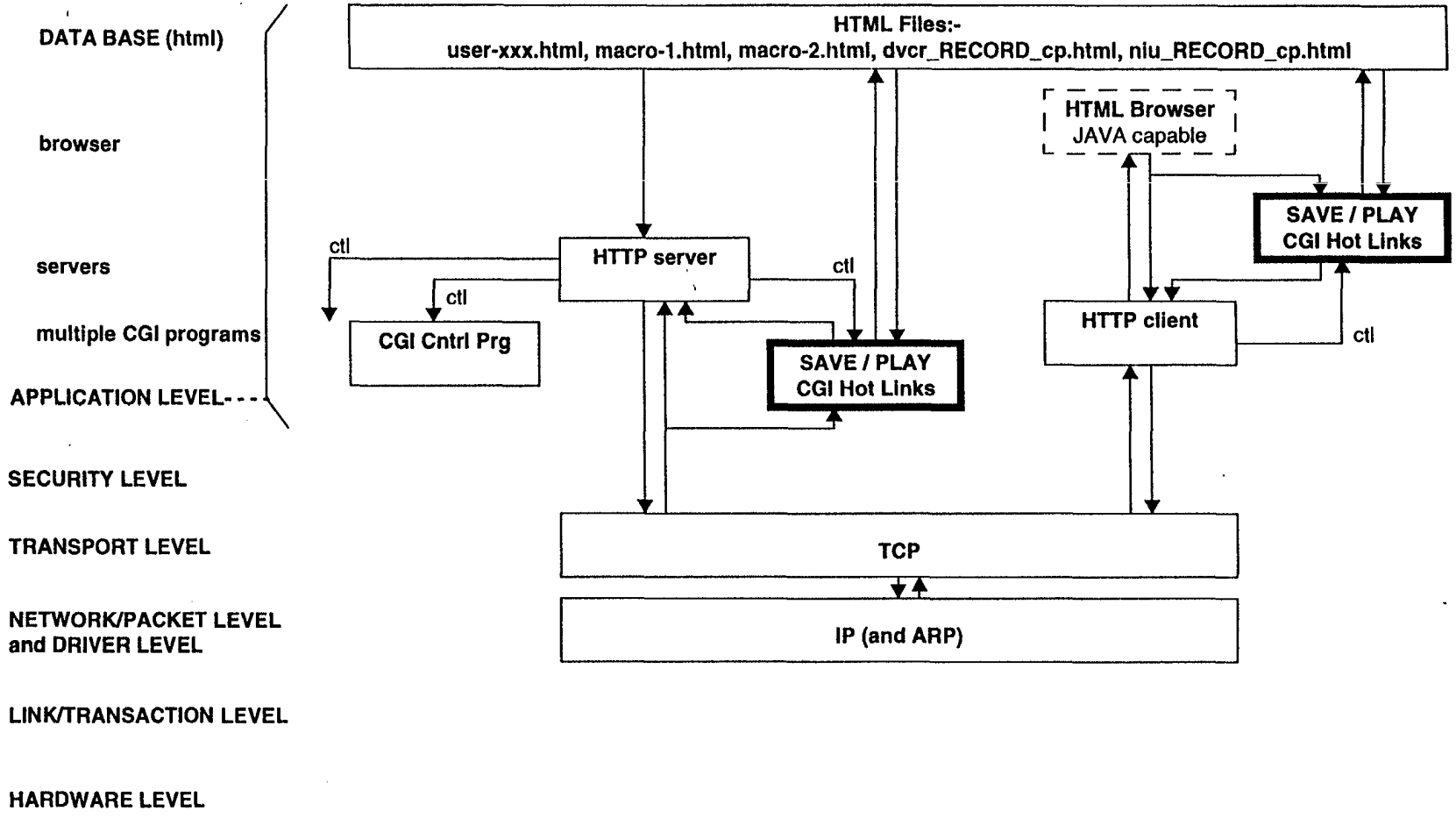
When a client makes a connection to an HTTP server, the client's IP address is given to the device so that the server knows where to send the requested information(HTML files usually). The idea here is to use that IP address as a unique identifier for making custom status files on the server for each client. So, in the above example, there would be three custom status files for "status.DTV1_IPaddress.html", "status.DTV2_IPaddress.html" and "status.PC_IPaddress.html". This gives each of those clients the ability to get their own status file from the common server. A generic "status.html" file would contain hotlinks to each of the custom status page as well as some other general status information about the device(no tape inserted, 3 record times currently set, low battery).

7. Future Extensions

All of the followings will be added in the future version of this proposal.

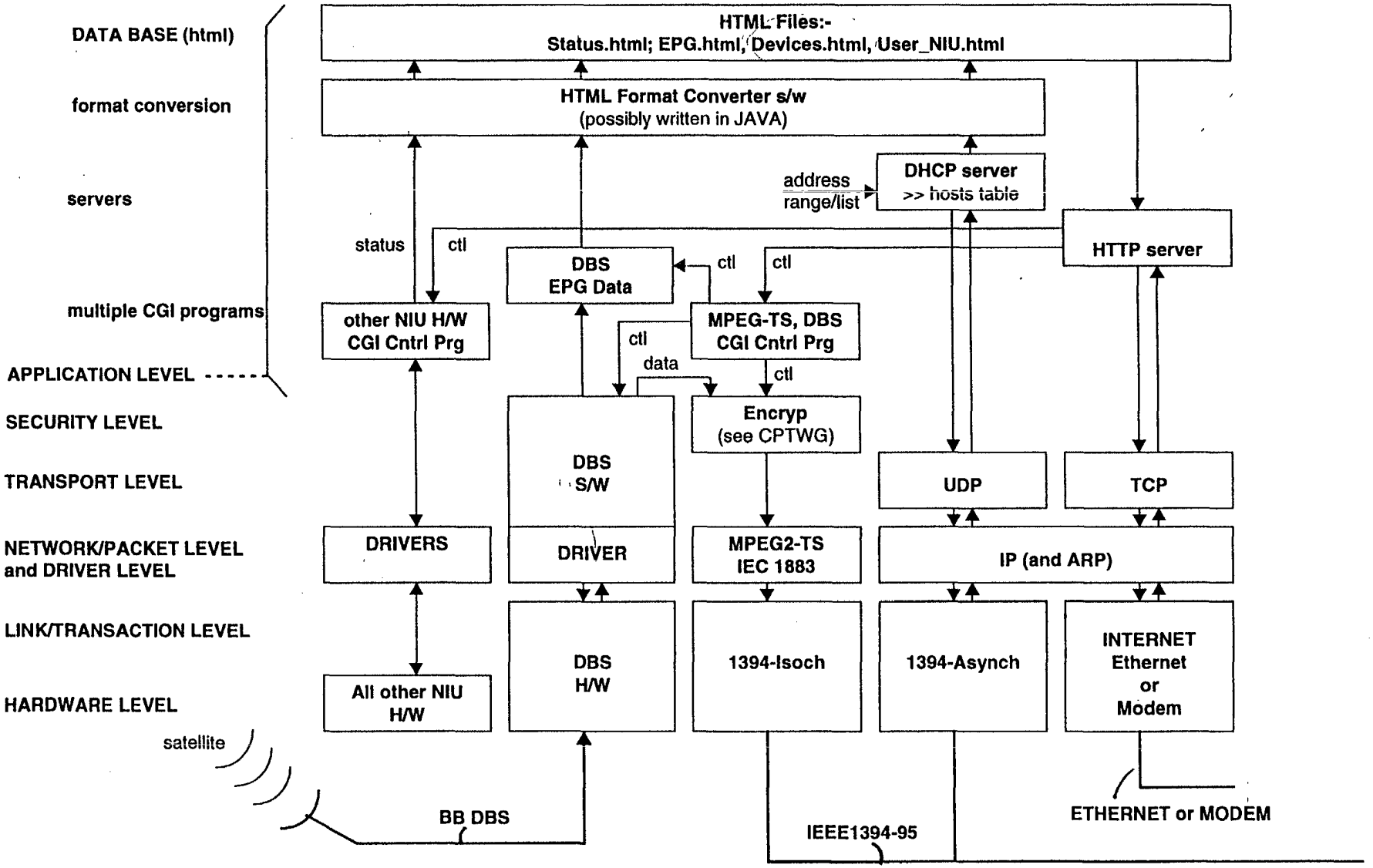
7.1 Software Application Upgrades
7.2 Browser Plug-ins
7.3 HTML GUI Upgrades
7.4 Protocol Revisions

Addition to Protocol Stack for Automatic Control Program (CP) or Macro. Shows Server-side Macro/CP and Client-side Macro/CP.



A - Page 23

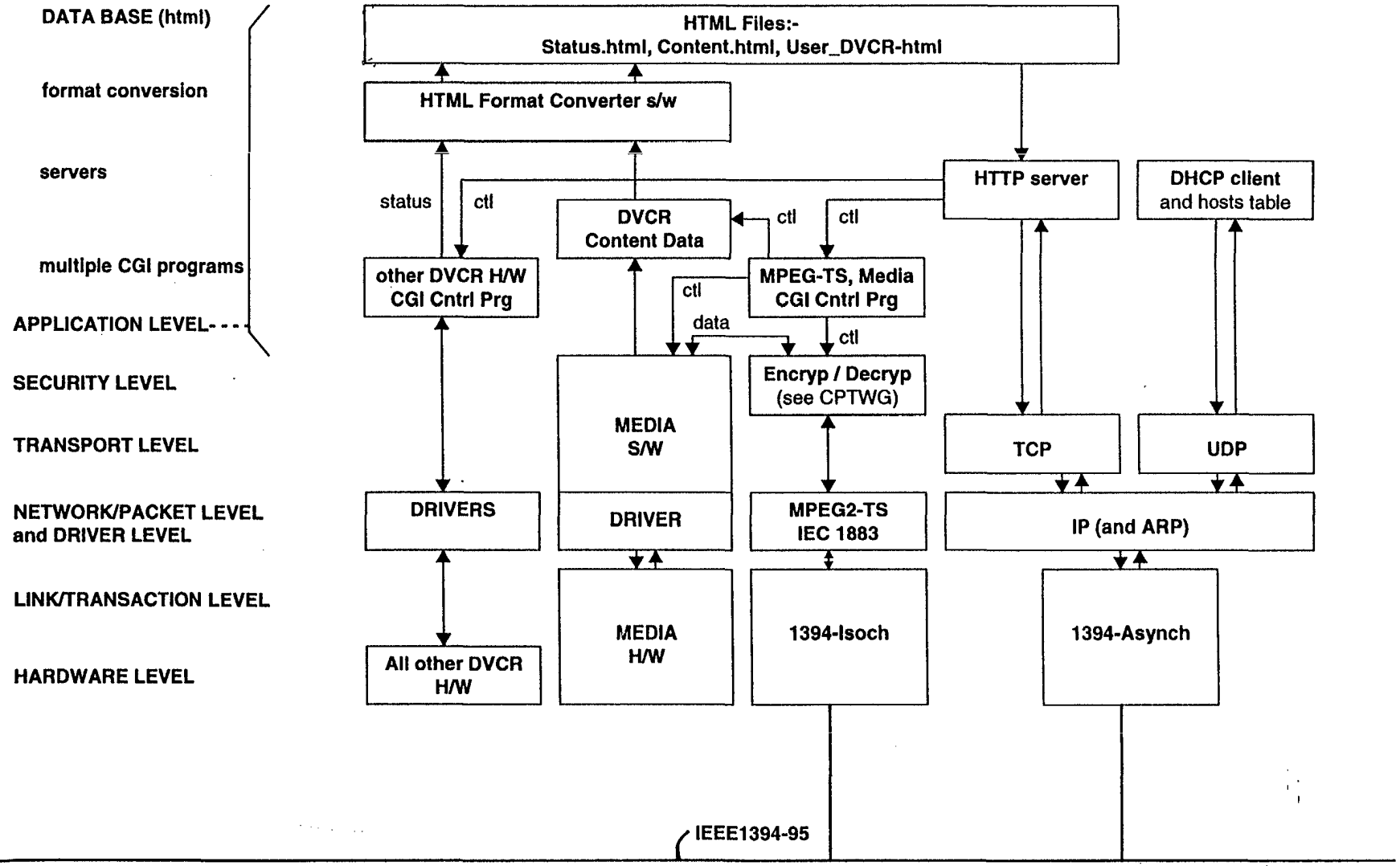
PROTOCOL STACK FOR NIU-PC in SIPHOT (server)



A-24

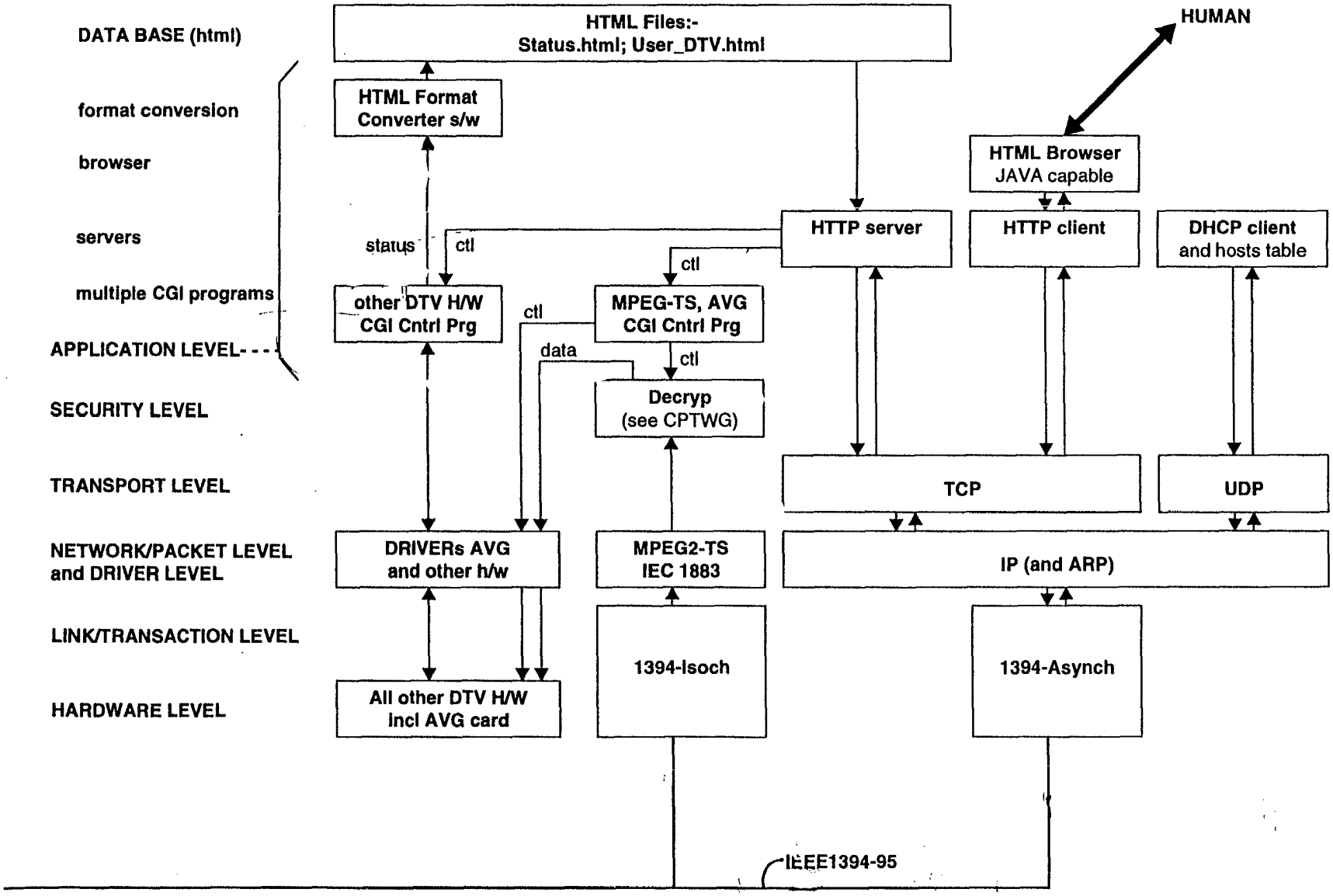
A-Page 24

PROTOCOL STACK for DVCR-PC in SIPHOT (server)



A-25

PROTOCOL STACK for DTV-PC in SIPHOT (client)



A - page 26

Appendix B

Addition to
Draft Specification

1. Introduction

1.1 Overview

This document proposes a new scheme for Home Network (or Home Theater) based on popular web browsing technology. All devices that conform to this architecture take the responsibility of controlling all of their own functions or services and so make it easy for manufacturers to implement new features independent of other manufacturers. It also allows devices to remain autonomous, not having to rely on command compatibility with other devices for user control.

The scheme, described herein, is derived from the Internet, which adopts a multiple client/multiple server model. All devices that provide an additional service to the existing IEEE1394 network are called "Server". Devices that provide display capability and a browser for user control are called "Client". The consumer interacts with the client device, but services are provided through the web pages on the server.

This is the first approach to transfer a Graphic User Interface between devices in home network. No other proposal, till now, provided a method for transferring graphics. In this proposal, all sources of GUI information, in the form of HTML files, reside on server devices. So every manufacturer can represent its own user interface and graphics, and provide their own control mechanisms. This is one of the many features that this architecture provides. Another feature is that the client device does not have to send any control commands to the server device to which it is accessing. The client device shows all the functions that the server currently being accessed can serve and it sends responses from user input back to the server. Control commands are not required for direct user interaction, although they are required for automatic or peer-to-peer interaction. This proposal is also independent of the command language used for automatic control.

The following diagram depicts the protocol stack that we are proposing.

<table>
<tr><td>Network Database
(hosts table, server IP addr, ...)</td><td>HTML files
User | Content/EPG | Status</td><td rowspan="3">Application Software
(HTML conversion, CGI program)</td><td>A/V Stream
(MPEG2-TS/PS, DVC)</td></tr>
<tr><td>DHCP</td><td>HTTP</td><td rowspan="2">Copy Protection</td></tr>
<tr><td>UDP</td><td>TCP</td></tr>
<tr><td colspan="2">IP and ARP</td><td>Driver S/W</td><td rowspan="2">IEC-1883</td></tr>
<tr><td colspan="2">IEEE1394 Asynch</td><td>System H/W</td></tr>
</table>

Protocol Stack

Actually, there are three columns in the above figure: Network Management, User Control and A/V Data Flow.

The leftmost column shows the flow of "**Network Management**". The protocol stack for this flow is Network DB/DHCP/UDP/IP&ARP/IEEE1394. Network Management has the role of configuring the network, assigning configuration parameters to each device and providing a directory of services to users. The key layer for this column is DHCP (Dynamic Host Configuration Protocol). The protocol is used for several purposes. The first purpose of adopting this protocol is to assign each device an IP address automatically. The IP layer is common to all data link networks as far as concerning to asynchronous control flow. The IP address is the most important part of IP network resources. The second purpose of using it is to provide a directory service to users by collecting information about the network configuration during the network initialization (or device discovery) process. Using the result of this collected information it is possible for users to control all devices in a well-defined way.

The second column from the left side shows the flow of "**User Control**". The protocol stack for this flow is HTML DB/HTTP/TCP/IP&ARP/IEEE1394. HTML files and the response-action of users are transferred through this flow. The side that provides HTML files is called server and the side that provides user action is called client. This is the method already used in Web browsing technology. The most important feature of this Client - Server architecture is that it is object-oriented. The service that is provided by a server is an object and the server provides the control mechanisms of its own services in HTML format. This architecture will alleviate the client from the burden of knowing all the functions of the server to which it is accessing.

The rightmost column shows the flow of real "**A/V data stream**". The protocol stack for this flow is AV stream/Copy Protection/IEC1883/IEEE1394. After receiving a command from the user or from other devices, the device starts to send or receive A/V streams using the procedure described in the IEC1883.

2. Architecture

2.1 IEEE 1394 Generics

2.1.1 Cable Physical Layer

All cable physical layer implementations compliant with this standard shall meet performance criteria specified in the IEEE1394 standard.

2.1.2 Link Layer

All link layer implementations compliant with this standard shall meet performance criteria specified in the IEEE1394 standard.

2.1.3 Transaction Layer

All transaction layer implementations compliant with this standard shall meet performance criteria specified in the IEEE1394 standard.

2.1.4 Serial Bus Management

All serial bus management layer implementations compliant with this standard shall meet performance criteria specified in the IEEE1394 standard. In addition, several Control and Status Registers and Configuration ROM's which will be defined in the future version of this specification shall be implemented.

2.2 TCP/IP and Others

2.2.1 IP and ARP over IEEE 1394

IP and ARP, network layer protocols, are required in this proposal. Currently a working group in IETF is discussing on the transmission of IP & ARP diagram over IEEE1394. That result can be adopted as the protocols in this layer. The future version of this specification will include which portion of it can be adopted and which portion of it is not required for this specification.

2.2.2 Transport Layer

TCP and UDP, transport layer protocols, are required in this proposal. All of the graphic user interfaces are transferred by TCP. However, UDP is also required for the network configuration process.

2.2.3 Security Layer

This layer is required for the both of isochronous and asynchronous transfer. For the isochronous transfer, the result from CPWG/DTDG can be used to prevent users from recording received stream iteratively. For the asynchronous transfer, there is nothing defined yet. However there should be some kind of mechanism for authentication. This will be defined in the future version of this specification.

2.2.4 Application Layer

Two protocols, HTTP and DHCP, are required for this layer. Any other protocols are optional. HTTP server process is required in the case of server and HTTP client process is required in the case of client. All server devices should have DHCP client process and all client devices should have DHCP server process. In addition, the device that has the DHCP

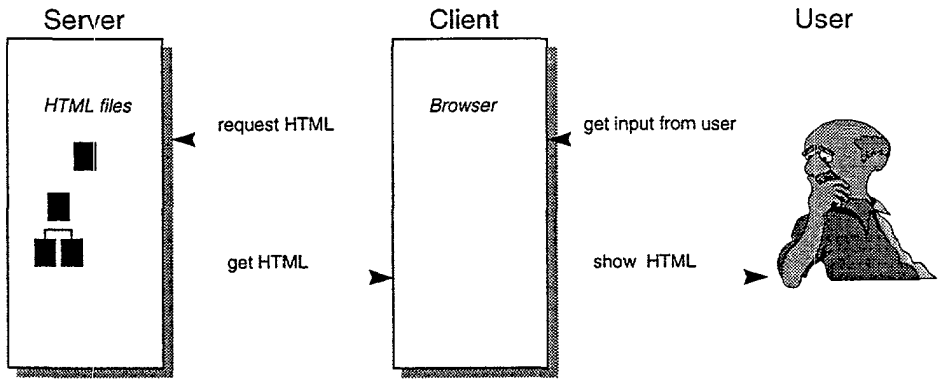
server capability should implement a DHCP server arbitration protocol that will be defined in the future version of this specification. The server device that has the capability of the Internet connection should have DHCP server capability.

A server device, if it has a content guide, should implement "Content Guide to HTML" conversion software. The Content Guide refers to EPG or media content guide. How to convert the content into HTML is vendor dependent.

3. Device Classification

3.1 Overall System Model

There are normally two types of systems: a Client and a Server. A device that provides a service or functionality to users and provides its control method in HTML form is called "server". A device that accesses the server HTML page, shows the content to users, and transfers the response information from users, is called "client". Internet DTV is a good example of a client device. DVCR and DBS set top boxes are examples of servers. The following diagram shows the concept of server and client.

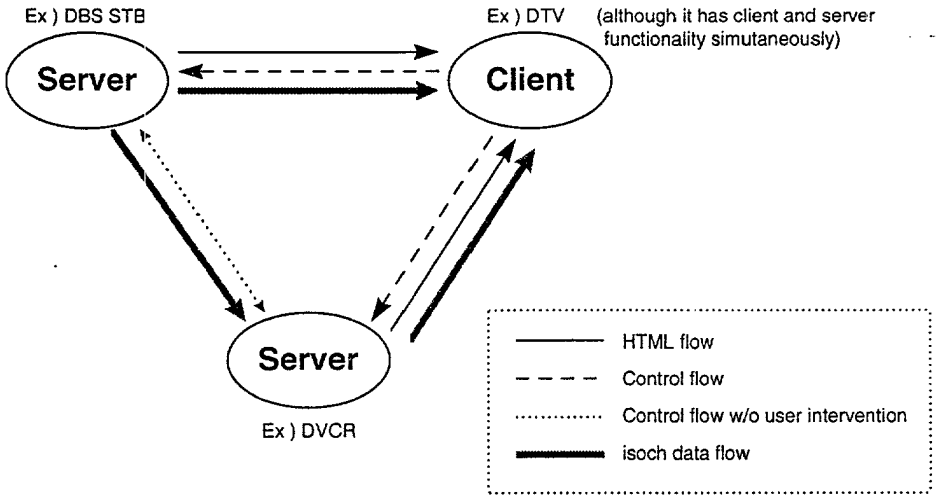


Client and Server

However during the home theater configuration phase, there exists another device called a "Configuration Manager". Actually, it is a DHCP server. It allocates a private IP address (in contrast to public IP address. Note: In IPv6 environment, all devices in home will have a unique IP address) to each device connected to the home theater and assigns a unique name to each. The Configuration Manager collects information about the network configuration and generates a HTML file which contains hyperlinks to the top HTML page of each device.

To gain a service from each device connected into the home theater, users access the HTML page of the server and click on the corresponding button that they want to operate. For example, users can click a "play" button of the DVCR to see a video tape installed in the DVCR. Then the DVCR runs a process that drives a capstan motor and transmits A/V streams through the digital interface attached. One way to implement this interaction is to use a CGI program that is triggered from the HTML page.

Following is another diagram that shows the relationship between server and client and shows the flow of data.



Control and Data flow

Because each server provides its service to users through a HTML page displayed on a browser, there is no need to send a control command from one device to another for human interaction. The only exception is the transaction between server devices without the intervention of a human. In this case, there should be some kind of mechanism to transfer control between them. We refer to this as command language optional (CLO) in this proposal.

3.2 Client Display Model

Following is a list of requirements for a Client Display Device:

- Display Capable
- DHCP Server Capable
- DHCP Server Arbitration Algorithm
- Always powered on for networking services
- HTML GUI (Optional icon and logo image files)
- Browser – HTML Rendering
- HTTP Client
- Local H/W Control Program
- MPEG-2 Transport Stream (TS) over 1394
- MPEG-2 audio/video graphics
- Hosts Table
- Device Page Generator (Uses Hosts Table)
- Isochronous Bit Stream Selection Page – Generate HTML page of all isochronous bit streams on network for user selection.

3.3 Server Model

Following is a list of requirements for Server Devices:

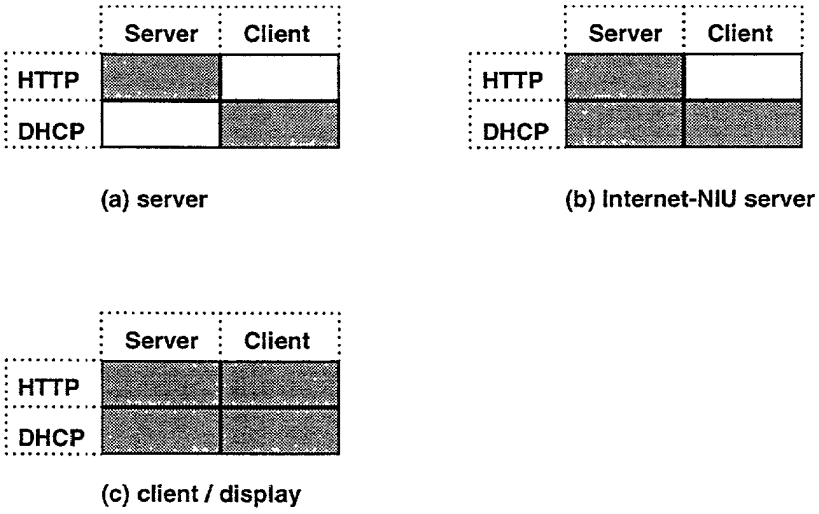
- HTTP Server
- HTML GUI (Optional icon and logo image files)
- Local H/W Control Program

- Content Guide (optional)
- Content Guide to HTML Conversion Program (if Content Guide is present)
- MPEG-2 Transport Stream (TS) over 1394 (Optional for MPEG-2 devices)
- Macro capability (optional, need command set)

Additional Requirements for Internet-NIU Device (OPTIONAL)

- DHCP Server (higher priority than Display Devices)
- Device Page Generator (Uses Hosts Table)
- Internet Proxy Server
- Gateway Capability
- Firewall

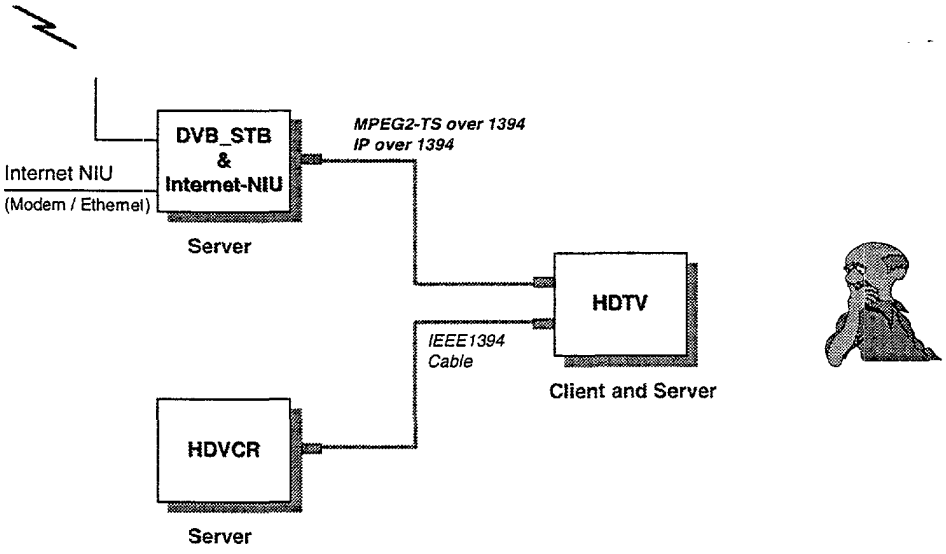
The following diagram shows the protocol requirements for the 3 types of devices.



Protocol Requirements for 3 types of devices

3.4 Example Home Theater

Here is an example home theater. Three devices - HDTV, DVB_STB with Internet NIU and HDVCR - constitute a simple home theater.



Example Home Theater

4. Network Management

Network Management performs several important functions for the network, such as network configuration, generation of device link page, IP address assignment, and the supply of some additional utilities for maintaining the network.

4.1 DHCP Server Contention

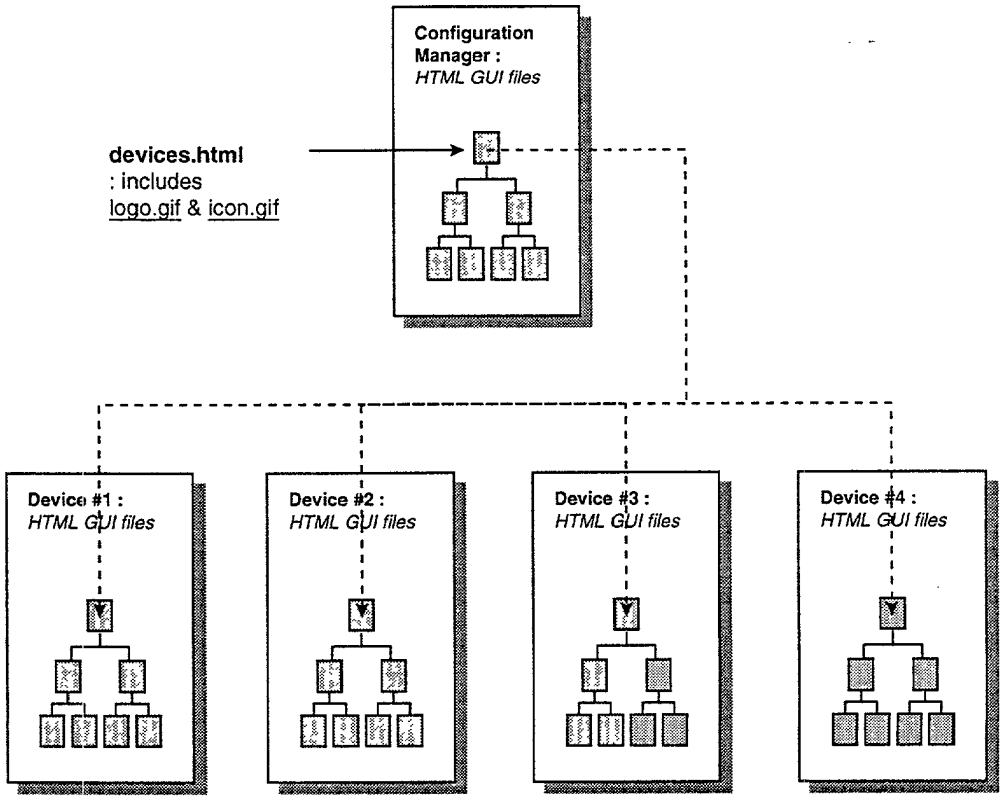
Sometimes there may be several DHCP servers on the Home Network. In the case of multiple DHCP servers, there needs to be some kind of arbitration protocol (or communication protocol) between DHCP servers, and this protocol should result in only one active DHCP server on the Home Network

Without this algorithm, a situation can result in which there are multiple groups of devices and in each group one DHCP server manages all devices belonging to that group. As a result, two devices belonging in different groups cannot communicate with each other because they have two different DHCP servers and therefore are not aware of each other's existence. To avoid this situation, some kind of contention algorithm or DHCP server to DHCP server communication protocol is needed.

4.2 Device Discovery

4.2.1 Device Discovery Process

The Configuration Manager (DHCP Server) generates a list of all devices on the Home Network during the Device Discovery stage and assigns each an IP address. The addresses are collected within a file along with the name of each device. This table is then used to create a list of all devices on the Home Network that can be accessed by the user. This list is in the form of an HTML file and is located somewhere within the DHCP Server's HTML GUI file structure. A program is used to create this HTML file, converting each line in the device list to a hyperlink and adding it to the HTML file. In the figure below, the list is placed at the top-level page of the DHCP Server's GUI. It is up to the manufacturer of the DHCP Server to generate a device link page that is user friendly and logically located within the DHCP Server's GUI so that it can be easily accessed by the Home Network user. It is also up to the DHCP Server manufacturer to create a link that is easily understood so the user can easily assess which device a particular link refers. To assist in this endeavor, each device may optionally contain a logo.gif and icon.gif file. These images are used as links to the device manufacturer's Web page and the device's top level HTML GUI page, respectively. These images are placed on the device link page in a manner that makes it easy to access all the devices on the Home Network.



Device Link Page

4.2.2 Logo Image File

A logo image file is a file containing an image of the device manufacturer's logo and name. Several variations of the logo file can reside on a device with a link to the desired logo file. The link can be updated over time or based on certain criteria of the manufacturer's choosing. The image may also be animated.

4.2.3 Icon Image Files

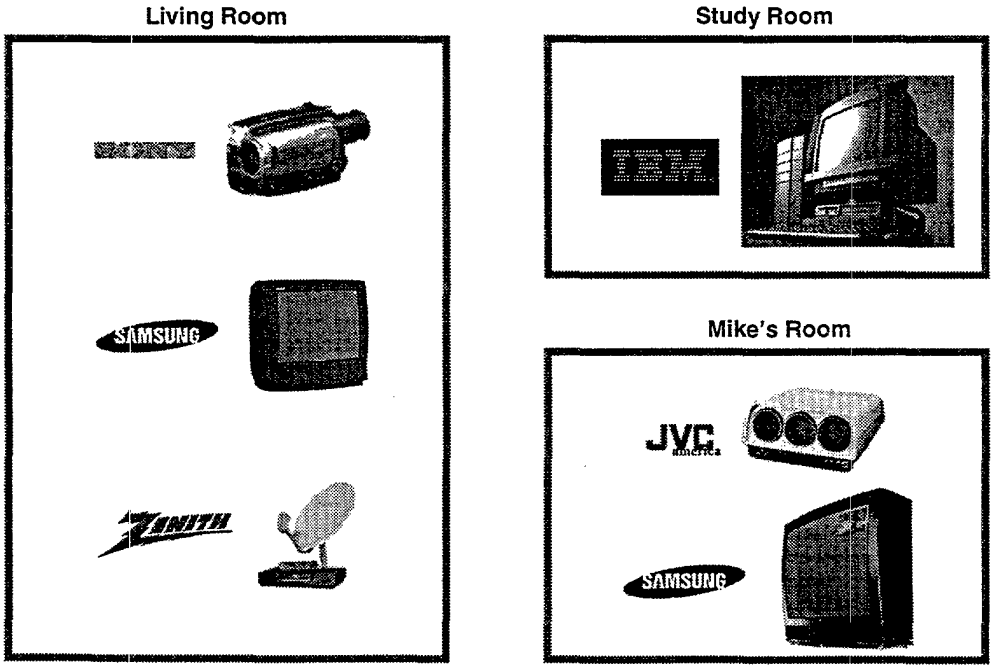
An Icon image file is a file containing a graphical representation of the device and its state. It can be either a picture of the device or a symbol representing the device. In addition, a model number may be included at the bottom of the image to assist in identification of the particular device. Several variations of the Icon file may reside on the device with each one representing a state of the device. A link to one of the images represents the current state of the device. To represent the various device states, the manufacturer has the choice of using a variety of symbols, colors, or even animation. The link to the image is updated over time or based on certain criteria of the manufacturer's choosing to indicate a change in state. Example state values are On, Off, Playing, Stopped, Recording, Rewinding, Forwarding, Searching, Media Inserted, and Media Absent. The purpose of the Icon image is to provide immediate device state information to the user. In addition, since the Icon images are retrieved from all devices whenever the device list is displayed, there is an immediate indication of the accessibility of all devices on the network. The absence of a device indicates that the device has been disconnected from the network.

Default Icon Image Files

The NIU Server may contain a set of default icon images representing the various types of devices that can be connected to the Home Network. If an icon image is not present on a particular device the NIU Server may use one of its default icon images or use the DeviceName attribute to create a hyperlink for the device.

4.2.4 Device Page

The Device page lists all of the devices on the Home Network with a link to each of the devices' top level HTML pages. The icon and logo image files are a standard size so the list can be arranged in an orderly manner. The device manufacturer's logo is displayed along with the device's icon. The logo can act as a hyperlink to the device manufacturer's home page if so desired and the icon is a hyperlink to the devce's top level HTML page. If they are not available a standard device icon or the string from the DeviceName attribute field may be displayed in their place and used as the hyperlink. The images are arranged in any manner desired by the NIU Server manufacturer, from a basic row and column configuration to a network topology diagram if possible. The NIU Server manufacturer may even allow the user to arrange the images in groups according to their placement in the home and provide the user with an additional text lines to describe the groupings and/or devices. For instance, the user may be allowed to group devices by room with a name for each room. The following is an example of this style of Device page:



Example Device Link Page

5. User Control

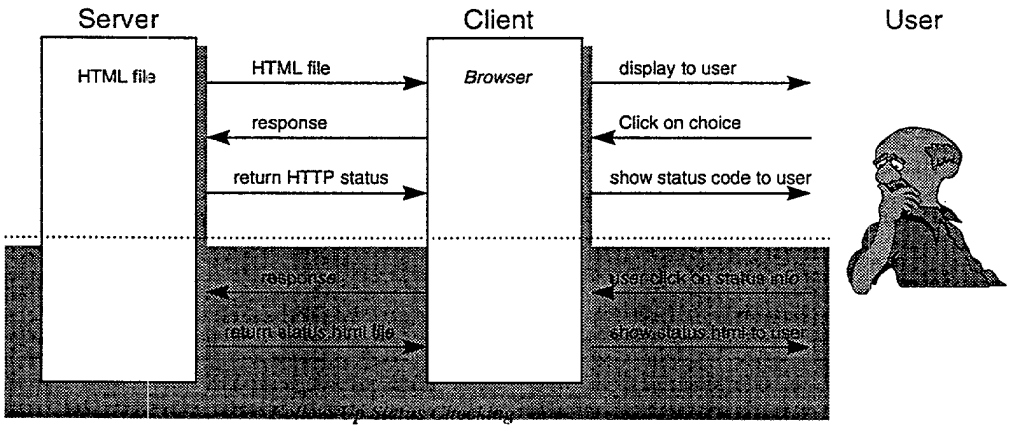
5.1 Macros

Users will often go through the same sequence of steps during the setup or adjustment of a device, or several users will have different requirements or preferences for settings. If the sequence of steps is repeated often enough or user requirements or preferences differ enough, the use of macros becomes a significant benefit. A macro is a sequence of commands that is saved and can be easily retrieved and executed, as often as desired. The use of macros can reduce an awkward sequence of numerous steps to a few easy keystrokes that are easily duplicated at will. Several users may have preferences for brightness, tint, and contrast on a DTV for example. Macros allow a way for individual users to save their own personal set of preferences for a particular device and then to quickly and easily re-establish those same settings at any time in the future. The macro application may be implemented as an applet that intercepts and stores certain commands instead of executing them.

5.2 HTML Command Status

5.2.1 Status Checking

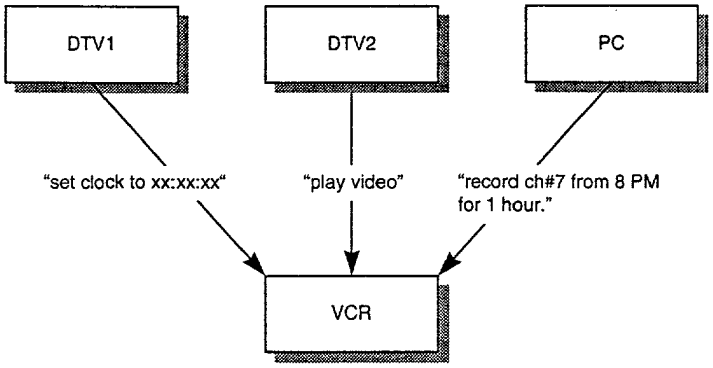
After a client (fetch html file) and server (return action) handshake, the http server normally returns a status response code to indicate whether the return action was successful or not (The server returns a response code of 200 when the return action was successful and 400 or 300 when failed). A bad response initiates a fetch of another status.html file that can include icon.gif files that indicate graphically what the problem is. In this case, status.html file is generated by the server and located on the server. It can also include the suggested corrective action. The following diagram illustrates this mechanism.



Status code response and Follow-up status access

5.2.2 Follow-up Status Checking

One of the inherent challenges associated with a digital network which consists of consumer devices is the problem of multiple, simultaneous accesses to one device. This is something like the critical section problem in the field of parallel computing. An example of this is that one DTV (DTV1) can be setting the VCR's clock, someone at another DTV (DTV2) is telling the VCR to play a video, and someone at a PC (at work possibly) is telling the VCR to "record channel 7 from 8 PM tonight for 1 hour".



Multiple, simultaneous accesses to one device.

Each of these activities has a status associated with the action. In the case of "atomic" operations, the status returned is either OK or NOT OK and that is all. However in other cases such as rewinding a tape, the initial status may come back as "OK", but a status regarding how far along the rewind is or just if it has completed rewinding is needed via a status page. Another non-atomic example is a more complex one where the VCR has been set to record later tonight, but the user (not at work) wishes to change the setting or delete it altogether. This section describes an innovative way to handle these types of situations through multiple "status pages".

When a client makes a connection to an HTTP server, the client's IP address is given to the device so that the server knows where to send the requested information (HTML files usually). The idea here is to use that IP address as a unique identifier for making custom status files on the server for each client. So, in the above example, there would be three custom status files for "status.DTV1_IPaddress.html", "status.DTV2_IPaddress.html" and "status.PC_IPaddress.html". This gives each of those clients the ability to get their own status file from the common server. A generic "status.html" file would contain hotlinks to each of the custom status page as well as some other general status information about the device (no tape inserted, 3 record times currently set, low battery).

Appendix C

Home Network Session Manager Specification

The Home Network Session Manager is the primary interface between a user and the home network. The tool is capable of accessing and controlling every service on the network and should be available on every client device. Although the immediate need is for the session manager to handle VCRs, TVs, and DSS boxes, it is designed to be general enough to handle many different kinds of electronic devices.

Functionality

The critical function of the session manager is to enable the user to start one of the services available on the home network. A service could be starting a movie, programming a DSS box, or any such function that makes use of the devices available in the network.

Each device will have one or more *capabilities*. For example, a VCR can output a video signal, and a PC can accept and output video, audio, or data inputs. Capabilities are either source-like or sink-like. A television has the sink-like capability of "Accepting Video Streams". Each source-like capability will have a *complementing* sink-like capability that is compatible with it. The "Output Video" capability in one device is complemented by an "Accept Video" capability in another device. Each capability will have a certain set of data specifications. For example, when a VCR outputting a video signal, the signal is being broadcast on a particular channel. This channel number and other details about the signal form part of the data specifications. For a device to capture this video signal, it will need to access these data specifications. Some sort of data exchange between the source and the sink is critical for the service to successfully start. The actual task of the Session Manager is to hook up a source capability with a sink capability.

Every device will embed a controlling application. This application will handle *all* communication between the session manager and the device itself. The control application is will be a device specific pocket of Java code that is capable of communicating with, and controlling the device in question. This isolates all the control implementation details within an application that is supplied by the device vendor. The control application will also enable vendors to provide their own look and feel to their devices. This control application will need to comply with certain specifications to be defined in order for it to be able to communicate with the session manager. Minimally, it should provide an interface for the device it needs to control within certain graphical limitations as defined by the session manager. It should also be able to advertise all the device capabilities and their individual data specifications when queried by the session manager.

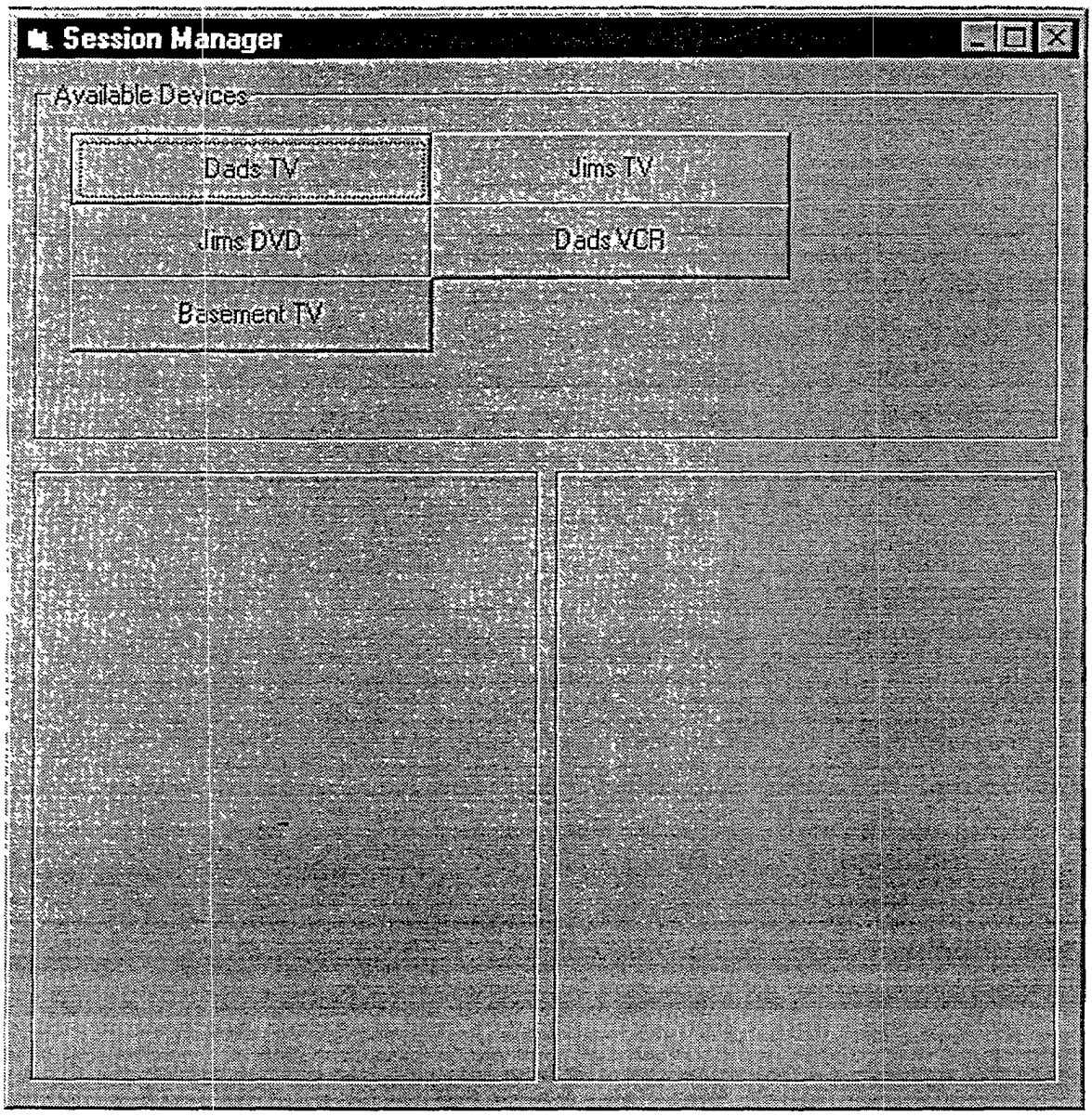
When the Session Manager starts up, the first task it does is to probe the home network to detect all the devices. This probe will return a list of all devices on the network, and their current status (in use, powered down, not in use, etc). All available devices will be presented to the user in the form of a list of icons with some descriptive text for each device.

The user then selects a device to use. Several things happen at this point. First, the session manager contacts the selected device and obtains its control application. It displays the interface of the control application, which will provide controls to the device. Then, the session manager will query the device (via the control application) for all its capabilities. All other devices are then queried to find their capabilities. Devices in the network that have complementing capabilities to the capabilities of the device selected are highlighted. The user should be able to select a source first or a sink. The selection order should not matter. Then the user selects one of the highlighted devices to act as the source/sink for the first device. The control application for the second device is brought up. A common example of this would be a user selecting a VCR, and then selecting a TV on which to view a video. The user needs to control both devices from one simple interface. Once a service is initiated from the source device, the session manager will handle the data exchange between the two control applications if necessary to enable the sink device, and the service will start.

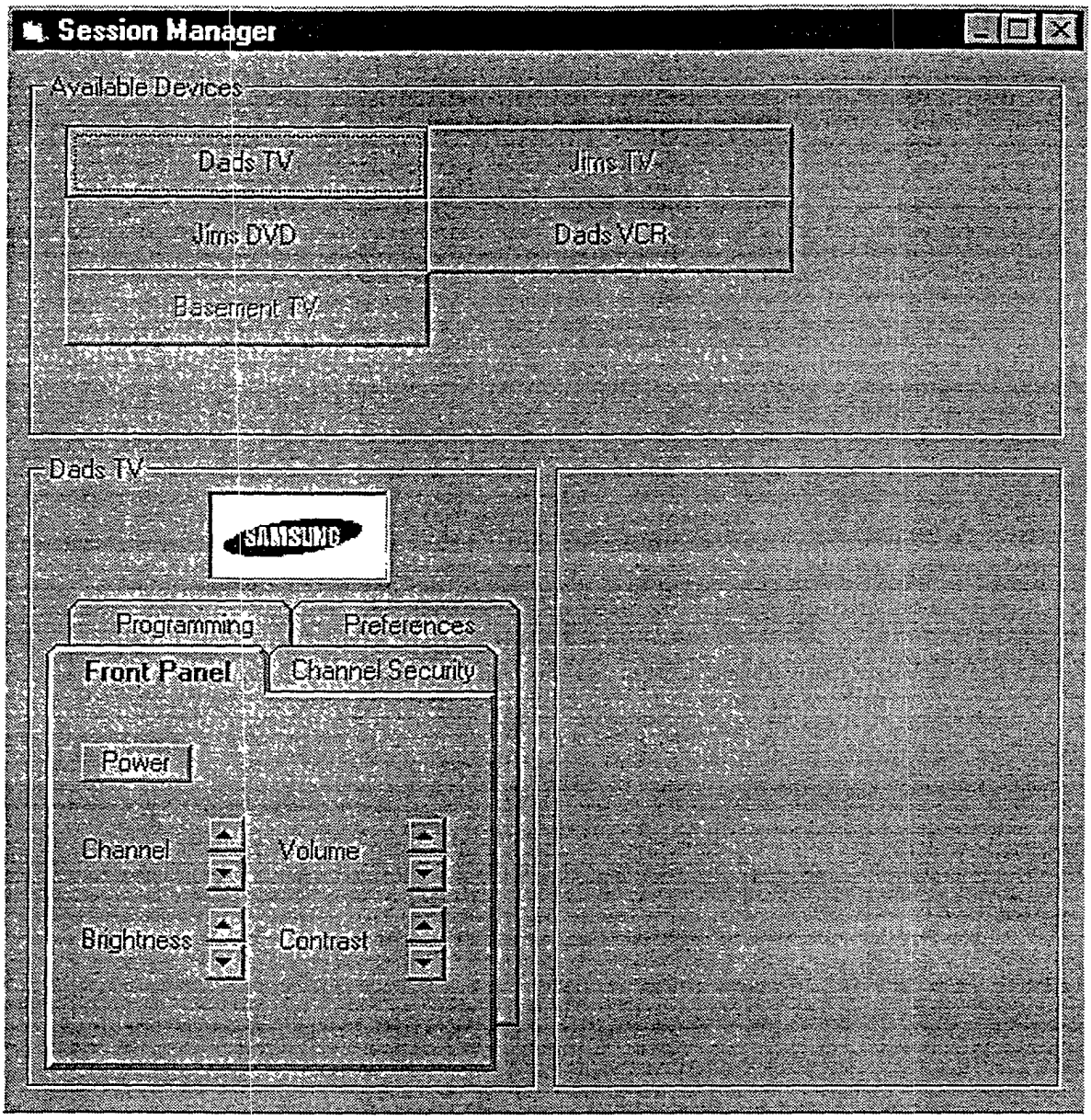
C-1

There are a couple of special cases in this scenario. Its is possible that the source and the sink of a particular session are in one device. In this case the user can simply start operating the control application interface brought up by the device manager for the device. No further selection is necessary.

A second scenario is when the user has selected a source and a sink but it is still not clear which service the user is trying to initiate. Such a situation occurs when there is more than one complementing pair of capabilities between the two selected devices. An example of this would be if a user selected a PC and a PC. The user may want to stream either video, audio or data from one PC to another. In such a scenario the session manager will present the user with a list of services possible between the two devices. The user can then select which service to start between the two devices.

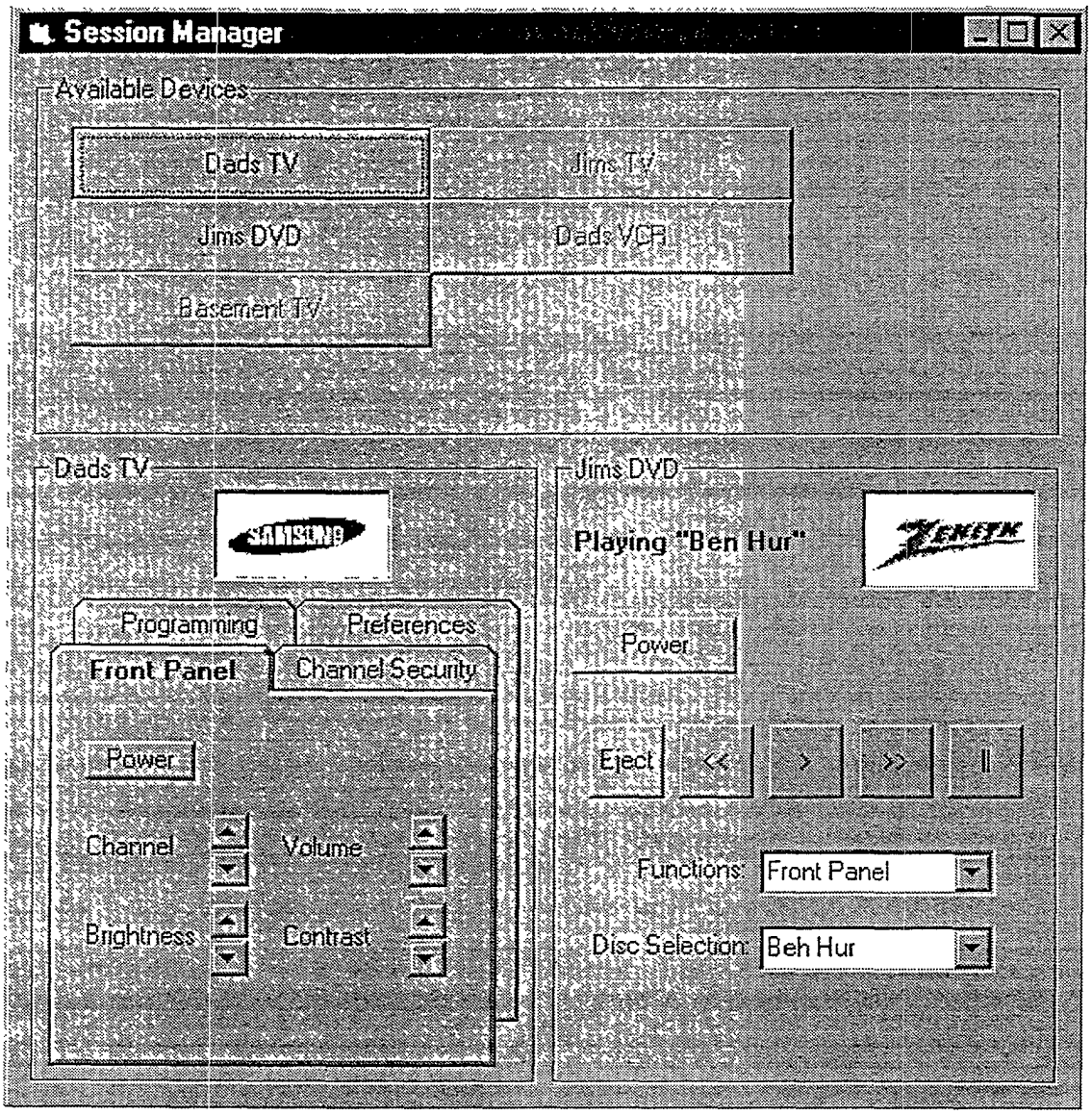


Session manager detects all devices in the Home Network



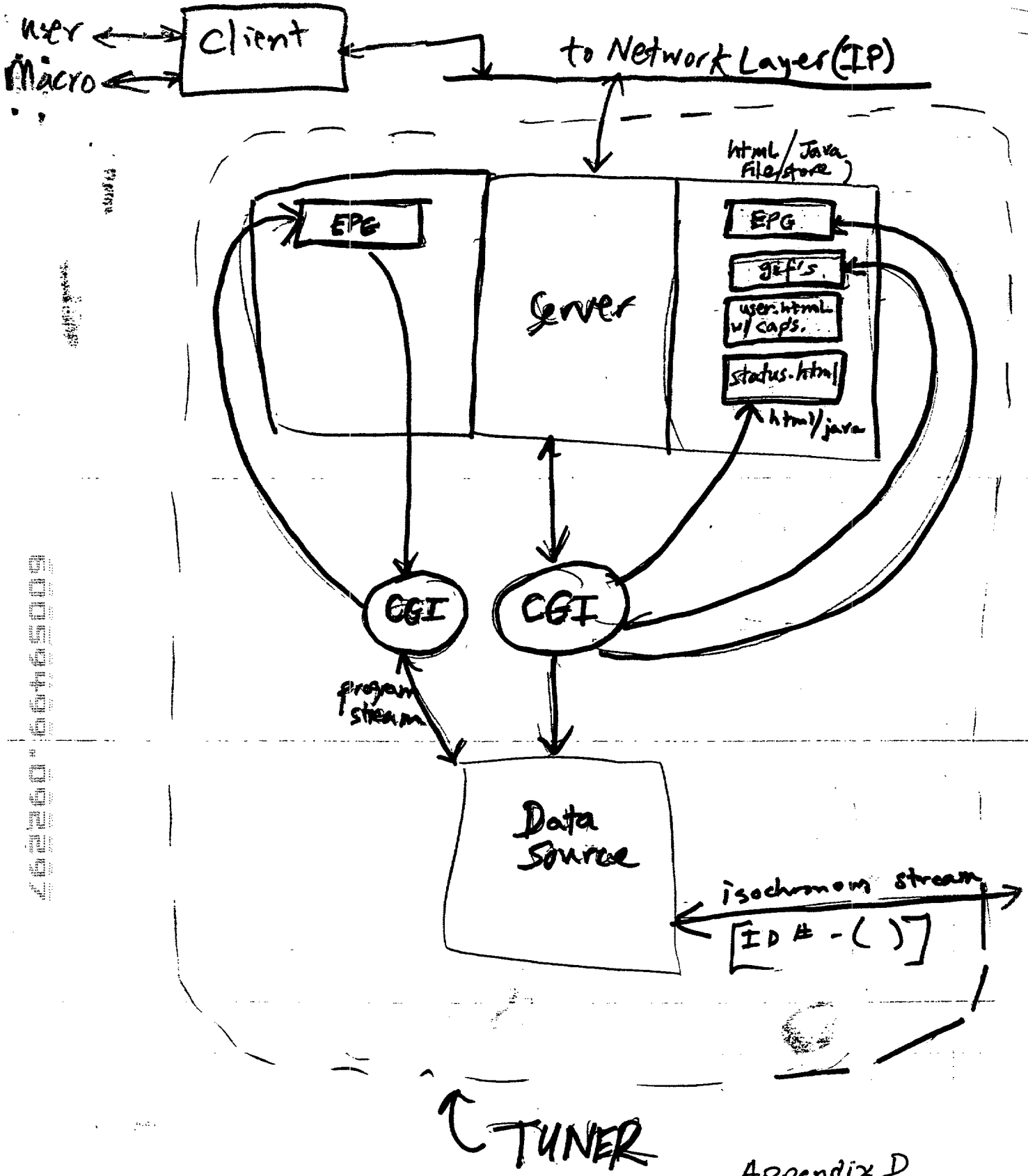
First device selected, device control application retrieved from device

C-4



Source and sink selected, control applications for both devices are available to user

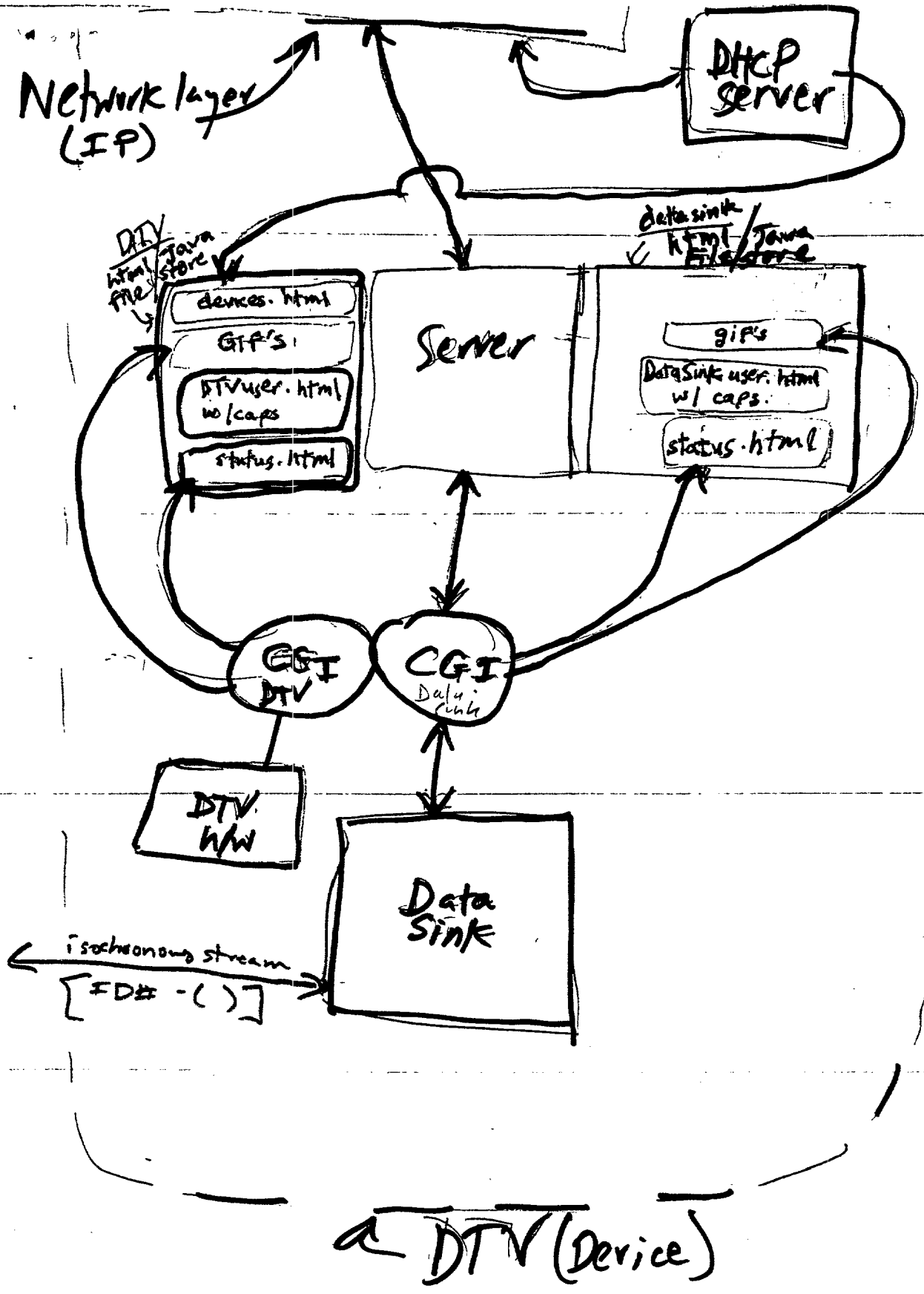
C-5



Appendix D

Additional Disclosure of exemplary tuner and device, in accordance with one embodiment of the present invention

D-1


Network layer
(IP)
DHCP server
DTV
html file
Java store
devices.html
GIF's
DTVuser.html w/ caps
status.html
Server
data sink
html
Java
gif's
DataSink user.html w/ caps.
status.html
CGI
DTV
CGI
DTV hw
Data Sink
isochronous stream
[FD# - ( )]
DTV (Device)

D-2