

VESA Home Networks: RFI for Network Protocol, Network Management, and Device Directory Services.

Response from 4Links, prepared by Paul Walker

1. Overview

The RFI asks an excellent set of questions, which are independent of the choice of any particular network technology. The opportunities for networks in the home are sufficiently important that it should be worth-while to find good answers to these questions and then design a network which optimises the implementation of those answers. In this context, the assumption of using 1394 for the backbone connection must be seen as deciding the solution before the requirements have been set or the problem has been defined.

In particular, security is of fundamental importance in the home network, and candidate technologies which present a weak set of rules which can not be enforced are tantamount to putting a notice on the front door saying "Thieves are requested not to enter these premises", and are unacceptable in the context of the home network

1.1 *JavaPP and simplified URLs*

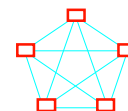
The proposed protocols for configuration and for network routing are based on Java together with a special class library, the combination of which we will call JavaPP, and on a physical routing scheme which bears great similarity to email addresses and URLs.

1.1.1 Configuration with JavaPP

An important aspect of the home network, as highlighted by questions in the RFI, is configuration for Plug & Play. The RFI quotes different mechanisms used by 1394 and CEBus, and points out that while these are valid within an entirely 1394 or entirely CEBus system, they are not generally useful for a heterogeneous network. A further problem with memory based CSRs is that they are fundamentally insecure. One P&P system that was designed for heterogeneous systems is the OpenBoot pioneered by SUN. This is based on small FORTH programs that are interpreted by a variety of different computers. While OpenBoot is probably the best currently available mechanism, FORTH is not a secure language and it is not ideal for use in distributed systems.

The Java language, also from SUN, is secure, and is designed for use in distributed systems.

We therefore propose a configuration mechanism like OpenBoot, but using Java instead of FORTH. As well as the advantage of security, other advantages are the small code size of Java Byte codes, the fact that an applet may be executed either locally or remotely, the fact that as yet undreamed of protocols can be defined, so the configuration can include details of functionality as well as the parameters of a predefined functionality, and the fact that there are already far more Java programmers than FORTH programmers.



Even for configuration, concurrency and indeterminacy can create difficulties for programming in Java as it stands. A special class library has therefore been developed which provides high-level primitives for thread communication and synchronization. These primitives upgrade Java from C A R Hoare's Monitors (1972) to Hoare's Communicating Sequential Processes (CSP) (1978), giving explicit control of non-determinacy and considerably easing the task of programming concurrent and distributed applications. Note that by using this special class library, the language itself is unchanged. For convenience, and for the purposes of this response, we will call the combination of language and class library "JavaPP". The special class library has been developed by researchers at the universities of Oxford (UK), Kent (UK) and Twente (Netherlands). Details will be presented at the forthcoming WoTUG conference at Twente, 13-16 April 1997.

JavaPP enables the definition and instantiation of threaded Java objects that communicate *only* via well understood and theoretically sound CSP synchronization primitives. Primarily, these are channel-based with strongly-checked user-defined protocols sitting on top of them. There is a formal semantics (the traces/failures/divergences model of CSP) to support reasoning about networks constructed from such objects that allows, for example, analysis for deadlock, livelock and starvation problems in the design. But the best thing about it is that these semantics are WYSIWIG, so that system complexity is scaleable and that we get no nasty surprises from even informal reasoning. This is not the case with raw Java (monitor-based) semantics.

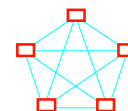
JavaPP objects give us real "Plug-and-Play" components that are simpler to use even than Sun's recent JavaBeans concept. Standing on the shoulders of CSP, we get a simple model of communicating components that precisely reflects the needs for home networks and rests secure on a mature theoretical foundation.

There may, of course, be great advantages in using JavaPP much more widely in the network than for configuration and management. One could conceive of light-switches or TVs or intruder detectors as themselves being JavaPP objects, controlled by aPplets. Java may not, however, be ideal for all applications, and so in this proposal the use of JavaPP is restricted to the configuration and management. Channels are set between end-nodes which need to communicate, and the protocol for the actual communication can be optimized for the particular end-nodes and what they have to communicate.

1.1.2 Network Routing Protocol based on simplified email addresses/URLs

For the network routing protocol, we propose to use a header with Path Identifiers and Channel Identifiers like ATM's VPI and VCI, with the exception that within the bounds of the home network they do not need to be virtual. We also use them very much like URLs or email addresses, except perhaps they are best read from right to left. For example the author's email address, paul@walker.demon.co.uk can be seen as first coming to the UK. Once in the UK, we have no further need of the .uk, so we discard it and find my ISP, demon.co. Having found that, we discard the demon.co and look for walker which finds the machine from which I'm writing this. Within this PC there can be a number of channels, but the one I use most of the time is my name, paul, and so the mail reaches me.

As the paths and channels in the home do not need to be virtual, the email address/URL can be a simple sequence of bytes, each of which selects the physical output port of a routing switch and is then discarded as the packet goes through the routing switch. The route, using the email address form, 10@157.23.14.1 might go from the keyboard on the microwave oven to the living-room thermostat: The packet would go out from the microwave's port 1 which connects to the kitchen hub, from port 14 of the kitchen hub to the closet hub, from port 23 of the closet hub to the CEbus bridge,



and to port 157 on the CEBus which is the living-room thermostat, for which channel 10 says "set temperature" to the temperature sent as the payload of the packet.

Use of the email address/URL in this way greatly simplifies the routing switch hubs used for the packets. The mechanism clearly scales, because the header can be as long as needed. In some cases, a group of routing switch hubs can be considered as a single hub, of up to 128 or more ports, and the header byte would only be stripped on exit from the group of routing switch hubs. (Such a group of routing switch hubs might also be used for a string or daisy-chain of nodes, so that the header does not need to be stripped at each node on the daisy-chain). So, within the confines of the home network the protocol scales and does not use an excessive number of bytes for the routing information.

A special case of the routing protocol is for isochronous, multicast, traffic which dispenses with the path header and simply uses the channel header. The routing switch hubs contain small look-up tables to indicate which output ports are addressed by the channel, and the packet is sent to all the output ports selected by that channel.

2. Questions not asked

In answering these questions it has become even more apparent to the author that bus topologies are seriously flawed in providing adequate answers to these questions. This is particularly the case when the bus exposes memory addresses, both of its operation and its configuration and status registers, to access and corruption from any other node on the network. An important question that should be asked is therefore whether such solutions are in any way compatible with the requirements of the home network.

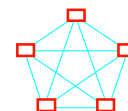
A number of questions concern the internetworking of a high-speed network with a low speed network such as CEBus. It is in fact possible that the costs of the proposed network solution are lower than the costs of nodes on CEBus, but it is recognized that some connection with powerline CEBus may be required, and that the CEBus might therefore be used more widely than just in its powerline version. An appendix to this response gives a possible way that might be used for power distribution in the high-speed network, together with CEBus carried on the same wires and connected via single twisted pair stubs to other equipments around the room.

3. Information Requested

This RFI covers three components of the VHN home network architecture: the internetworking protocol; the device directory service; and the network management service. Each section describes the component and presents an initial set of questions designed to elicit the characteristics and requirements of suitable components for each area. We request that responses to each section of the RFI include answers to as many questions as possible. It would be even more useful if the responses suggest a suitable candidate component solution for each section.

3.1 Network Protocol

To ensure full interoperability among the many networks and devices in the home, a common network protocol is needed. This network protocol will ensure that higher layer protocols can be specified independent of the underlying networks and their link-layer protocols. It also enables incorporation of new communication technologies into the home network without perturbing the existing networks and devices. Therefore the choice of an internetworking protocol is very important for enabling as yet unforeseen uses of the home network.



3.1.1 Candidate network protocols

1. Which network protocols will be suitable for the home network?

The proposed protocol is to encapsulate packets, with a header formatted like an email address or URL to indicate the physical path to be taken through the network, and with a single character terminator. There is no restriction on what can be encapsulated and no restriction placed by the protocol on packet length, although individual end-nodes may place restrictions on the packet lengths they can send and/or receive. In some cases, where the packet to be encapsulated already includes a header, the proposed email address-like header may be substituted for the original header (and resubstituted if necessary at the other end).

2. On what platforms should these network protocols be available?

The purpose of an exceptionally simple but flexible protocol is that it can be implemented for an exceptional variety of platforms. Current implementations exist for PCI, and chips and macrocells are available from SGS-THOMSON, Matra-MHS, and 4Links

3. What application and transport protocols are important to support above the network layer?

Any that are needed. It is suggested that the use of JavaPP may considerably ease the design and implementation of these protocols.

3.1.2 Efficiency

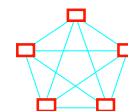
1. What is the acceptable overhead imposed by a networking protocol on the networks in the home? We are interested both in bandwidth overhead imposed on the networks and the processing overhead imposed on devices in the home.

For both efficiency and environmental considerations, the bandwidth overhead of bus topologies, particularly for high-speed traffic, is excessive. Traffic should only travel on wires necessary between source and destination, and, for example, a communication between two adjacent nodes should not cause all the other wires to emit Radio Frequency Interference pollution, nor should it cause all the devices driving those wires to dissipate power

Restricting traffic to the wires needed for the communication frees up the other wires to carry other traffic concurrently, so the network traffic scales as the network is enlarged, greatly improving bandwidth efficiency and latency

The bandwidth overhead of the proposed routing protocol is a few bytes per packet, usually no more than about 10% except for particularly small packets where latency is likely to be more important than bandwidth.

The processing overhead of the protocol is a simple cost/performance trade-off. Implementations exist of the proposed protocol which impose exceptionally low processing overhead, while still providing high bandwidth and low latency within reasonable cost. Cost can be further reduced for other implementations by using processing overhead. In particular, a cheap 8-bit microcontroller, plus about 2000 gates and two 16-byte FIFOs --- probably less than a UART --- can handle the protocol entirely in software. Roughly doubling the logic, and perhaps making the FIFOs larger, would allow close to full bandwidth utilization for one or two channels. Increasing the functionality or bus width further increase either cost or processing overhead, but the simple protocols mean the cost is always likely to be lower than many alternative technologies.



2. *Must network protocol implementations scale based on the needs of individual devices?*

Network implementations must scale. Logical bus and ring topologies, where the available system bandwidth reduces as nodes are added, clearly do not scale, and so it will always be expected that they will be looking for higher speed (and cost) versions of their technology. This is solving the wrong problem: the problem that needs to be solved is the logical bus/ring topology so that the actual physical topology is much more efficiently exploited, and so that the total network bandwidth is many times the bandwidth of an individual link.

Individual devices must be capable of driving the network at the designated speed of the network, and they must be able to store and transmit headers of a reasonable length, perhaps up to eight bytes, for the channels on which they need to communicate. They must not, however, be required to be compatible with future, as yet unknown, upgrades to the network.

3.1.3 Configurability

1. *How much user configuration is required for correct operation of the network protocol?*

Tools exist for mapping the network and for allocating routing headers, although further work will be required of these for the full protocol described including multicast/isochronous traffic.

2. *Is this level acceptable for the level of expertise of users in the home?*

Not yet. What the users must be able to do is to name nodes in a way that they can understand, perhaps identifying a particular device by switching it off then on again, and answering a set of menu questions about where in which room the device is in and what they want to call the device.

3. *Can the network protocol be made more auto-configurable in the future?*

Yes

4. *What are the configurable parameters of the network protocol stack?*

Node names, paths, channels, max. packet length (for a particular device, possibly for the network as a whole), passwords. Possibly network link speed.

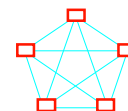
3.1.4 Network address characteristics for candidate network protocols

1. *What device-level addressability does the network protocol provide? Is this sufficient for the expected number of connected, digital devices in the home?*

The proposed routing protocol based on email-like addresses and using individual bytes to select the output port from each routing switch hub. A limit of eight bytes for the header gives $16 \cdot 10^{18}$ possible paths from each node, but the space is likely to be very sparsely populated. If the routing switch hubs each have four ports, it allows 64 000 possible paths, which should be adequate for the home.

2. *Does the addressing structure support efficient routing protocols?*

In spite of the sparseness of the addresses actually used, the routing protocol is exceptionally efficient in terms of both bandwidth (because there can be many communications occurring concurrently) and latency (because there is no contention for a single shared medium) and routing switch processing (because the actual routing is physical and can be done by simple hardware).



3. How are device addresses allocated and assigned? Is the address allocation centralized or distributed?

Ideally, all devices should have unique identifiers, but these are not strictly necessary for a network topology which uses point-to-point links and physical addressing of each link. The device address is therefore its unique position in the network.

There may be multiple possible paths between two devices, some of which may not be usable because of potential loops, and so the configuration/management program must allocate the usable paths and the channels which are to be used across those paths.

The configuration/management program is a JavaPP program which may be centralised or distributed as necessary.

3.1.5 Network protocol routing in the home

An important characteristic of a network protocol is the availability of suitable routing protocols.

1. Does the candidate network protocol have a well-defined, efficient routing protocol?

Yes, as described in the overview and in 4.1.4.

2. How does the routing protocol interact with access networks of importance in the home?

The simple encapsulation of packets makes for exceptionally straightforward mapping of access network protocols. Some examples are shown in Figure 1.

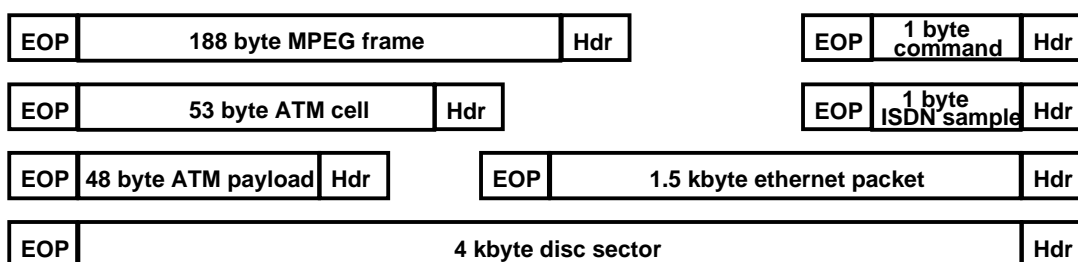


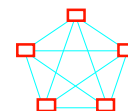
Figure 5: Some examples of possible packets carried on an IEEE 1355 network

3. Are the routing protocols robust? Self-healing? Can they handle loops in network topology?

They are robust. It is possible, even in the home environment, to construct networks with no single point of failure which will affect any other node or link than the one that has failed.. They do not discard packets and so have much less need than other networks for retry and self healing. They are self healing when a link is removed and subsequently replaced. In some circumstances, where more than one link connects between two routing switch hubs, the traffic will automatically switch to an alternative link when one of the group of links is removed. The path allocation tool in the network management and configuration program ensures that no paths are allocated which would generate loops. Note that this still permits traffic along all the links present in the network, but some traffic may be restricted from some links if such use would have caused a loop.

4. Are they efficient enough to support routing of isochronous audio and video data over the home network?

The freedom from contention of the network compared with a shared-medium bus gives an exceptional quality of service for a given link speed. In many cases this will be adequate without any further protocol to provide isochronous services. Without inhibiting other traffic on the network, time markers can be added to packets to indicate the time at which they were transmitted, to even out the flow. In an extreme case, asynchronous traffic can be inhibited every 125



microseconds for the isochronous traffic to have free access to the network. In most circumstances, however, this is not necessary.

5. *Can cost-effective routers for the home be built based on these routing protocols?*

Yes.

The SGS-THOMSON C104 is a single chip 32-port routing switch using a far more elaborate routing protocol than is proposed, and it has routing tables for each port. Estimates a routing switch chip to these protocols with three external ports and one internal port suggest that it should be fewer pins and smaller silicon area than TI's three-port repeater for 1394.

3.1.6 Support for Multiple data link protocols

A network protocol needs to operate over a number of data link protocols in the home. Therefore its relationship to these data link protocols may be important.

1. *Should a candidate network protocol be independent of the underlying data-link protocols?*

Yes, but the complexity of some of the underlying protocols can make this difficult for some of the protocols.

2. *Can it be supported over low-bandwidth networks such as powerline CEBus?*

Yes. Rate adaption is needed, but this just needs buffering to the extent of one or two packets.

3. *If yes, how? If not, then what mechanisms are available to implement interoperability between such low-bandwidth data link protocols and the rest of the home network?*

It would almost certainly be easier to produce a transparent connection between two CEBus networks than to convert packets from the basic network protocol to the CEBus protocol. The conversion could, however, be handled as a JavaPP aPplet, and even performed by a "server" running the aPplet.

3.1.7 Support for Isochronous Data Streams

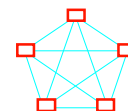
Isochronous data streams are important on the home network for entertainment services such as audio and video distribution. (Not all home networks are required to support isochronous data streams. The network protocol must support this service for those networks that do support isochronous service).

1. *Does the candidate network protocol support isochronous data streams? If so, how?*

Yes, by providing a network with much less contention than a shared-medium bus.

2. *Are there any inherent limitations on the number of isochronous streams that can be supported? What are these limitations?*

No, but there is a limit to the number of multicast channels, and there is a close relationship between multicast channels and isochronous channels. Apart from a few channels such as for configuration, the 256 values of the header byte need to address the output ports of the routing switch hub, and the remainder are available for multicast channels, and so a 16-port hub could support about 230 multicast channels. The cost of these channels is not large, but, to minimise cost, it may be worth reducing the available number to somewhere between 16 and 128.



3.1.8 Extensibility

1. *Is the network protocol extensible to handle new classes of devices, new applications, and new communication media?*

Yes, yes, yes. And it scales.

3.2 Network Management

Network management performs several important functions for the network, such as network configuration, performance monitoring, fault detection, fault isolation, and resource management. Network resources may include connections, bandwidth, buffers, interfaces, entire devices, or a host of other things. In order to perform effective network management, a concise taxonomy of network resources is needed, as well as the mechanisms to define resource objects and instances of those objects. Network resources may be classified by protocol layer, by device, by interface, or by many other means. The taxonomy and mechanism are used to perform several basic network management functions such as configuration, performance, and fault detection. This section begins by requesting information regarding resource taxonomy, resource modelling mechanisms, and the network management functions for candidate network management systems. The final sections request information regarding security and the user interface for the network management service.

Answers as full as possible will be attempted for the following questions. However the general answer is that everything in the network is seen by the configuration and management system as a JavaPP object.

The use of the secure language, Java, for configuration and management is an example of the importance that needs to be given to security in the network. Our homes are our private places and that privacy must be preserved. And if the network is used for intruder detection, as it ought to be, it must be proof against attack from a potential intruder. Statements such as have been seen for other technologies concerning rules for snooping and spoofing are tantamount to a notice on the front door saying "Thieves are requested not to enter these premises" and are intolerable in the context of a home network.

3.2.1 Network Resources Taxonomy

Currently, VHN believes that a taxonomy of network resources should be based on a physical device or box. VHN believes such a taxonomy enables a simple, concrete realization of the home network for the average home user. For further simplification of use, there is perceived value in associating devices with a directory service.

1. *What is the taxonomy of a candidate network management system?*

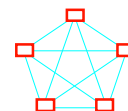
The use of JavaPP for network configuration and management means that all devices and resources on the network are JavaPP objects.

2. *How are resources partitioned in the network?*

Channels are established between those processes in devices which need to communicate, and paths set up along which the channel communication can occur.

3. *How are shared resources partitioned?*

Each device or process within a device which needs access to a shared resource is provided with a request/grant channel to the shared resource. The actual channel used for the access is only then set up when the request is granted to a particular device or process.



4. *How is a device represented in an existing device directory?*

Initially as a unique node on the network, by the devices to which it connects.

5. *Can device identification be easily mapped to new directory services?*

Yes, via the JavaPP configuration program and aPPLets.

3.2.2 Network Resource Modelling

The resource modelling mechanism should be able to define any network resource as well as instantiate all instances of a defined resource.

1. *What is the resource modelling mechanism?*

JavaPP

2. *How are resources defined?*

Protocols are defined by JavaPP aPPLets.

3. *How are resources instantiated?*

By communication between the aPPLets.

4. *How does the instantiation of a resource relate to the taxonomy of the network?*

(not sure what the question is asking)

3.2.3 Configuration

A major function of a network management system is configuring network resources. Network configuration includes setting addresses, names, buffer sizes, filters, etc.

1. *How are default values of configurable resources determined?*

By JavaPP aPPLets, running either on the end-node which is being accessed or on the accessing node.

2. *How are values of configurable resources set?*

By JavaPP aPPLets, running either on the end-node which is being accessed or on the accessing node.

3. *How are values of configurable resources retrieved?*

By JavaPP aPPLets, running either on the end-node which is being accessed or on the accessing node.

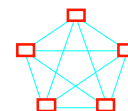
3.2.4 Network performance monitoring

A candidate network management system should be able to easily track performance of a home network. Network performance parameters include percentage of bandwidth in use, number of packets transmitted, number of buffers available, etc. How important is network performance monitoring for a home network?

It is important in that the user is inevitably unqualified to tune the network to improve performance if the performance is unsatisfactory. It is not, however, as important as it is to a public network, and care must be taken that excessive cost is not built into unnecessary performance monitoring equipment and software.

1. *How are default values of performance resources determined?*

The use of Kanban style feedback-based flow control mechanisms ensure that performance is to a large extent self balancing. The routing protocol outlined in the overview includes a priority



mechanism, so that high-priority traffic can at least to some extent take precedence over lower priority traffic, and if the connectivity of the network allows it may even be possible to allocate special paths to the high-priority traffic.

Tools such as Opnet are available for performance estimation for the proposed networks, but it is not envisaged that these will be suitable for the layman user.

There is a trade-off in the network depending on the length of packets and the amount of buffering in each port of a routing switch hub. For an individual communication, longer packets may improve performance if the performance is limited by the end nodes. On the other hand, long packets tend to block other traffic, increasing latency and reducing overall network bandwidth. A shortage of buffering in the routing switch chips does not cause data loss, but it does cause the blocking to propagate back to the source of the communication. Where a path is unique to a particular channel, the packet length can be set as long as required. Where the path, or part of the path, is shared with other traffic, packet length should be set to a maximum which does not cause excessive blocking. Routing switch chips with minimal buffering should give good performance with packets up to 48 or 64 bytes; with more buffering, packet lengths up to 200bytes, suitable for MPEG frames, should be totally satisfactory, and these packet lengths should not cause excessive degradation in networks with switches with minimal buffering.

2. *How are values of performance resources set?*

By setting maximum packet lengths, and by setting paths both for minimum distance and minimum blocking.

3. *How are values of performance resources retrieved?*

By accessing the configuration JavaPP aPplet in each node.

3.2.5 Fault detection and isolation

Fault detection can have a wide variety in causes, classifications, and actions. VHN currently believes all fault indications should include a description, time/date of the fault, and the network resources associated with a fault.

1. *What are the fault resource types? 2. What are the fault resource values?*

Parity error

Disconnection Error

Illegal character

Hardware time-out on link (may be same as disconnection)

Extraordinary End of Packet terminator, to indicate that the preceding packet is not to be trusted

Software time-out on transmission completion

Packet wrong length

Packet for non-existent channel

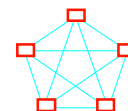
Too many/too few packets

3. *What mechanisms are used to detect faults?*

Parity, time-outs, and an absence of memory addresses.

CRC is provided where necessary for those channels that require it.

The efficiency of the network, in only sending traffic to the end node(s) which actually need the communication, is a major part of fault detection. Channels are set up between processes in two



nodes, with a protocol which both those nodes understand. If a packet comes from the wrong place, it will be acknowledged to the expected source rather than the actual source, and this protocol violation will be detected. If the packet does not arrive for any reason, the sender will time-out waiting for the acknowledgement. If it goes to the right end node but the wrong channel, again there will be a protocol violation when the wrong sending channel gets an unsolicited acknowledgement and the actual sending channel gets no acknowledgement.

Memory addresses are handled entirely locally, so there is no possibility that a bit error in a memory address transmitted over the network can corrupt any part of the node's memory --- such corruption can only occur as a result of a fault within the end-node itself.

4. How are faults associated to a particular network resource(s)?

The path is known for each channel, so any error can be associated with a particular subset of the devices in the network. In most circumstances, the devices can be interrogated to find whether they have detected a fault, and the fault can therefore be pinpointed automatically.

5. How are actions associated with a fault?

If the fault is transient, it should simply be logged and made available for the user if the user is interested. Permanent faults should be notified to the user.

6. How are faults reported to the home user?

Ideally in something along the lines of the form "Kitchen hub, port 14, (with picture of which is port 14) has permanently disconnected from closet hub port 6 (again with picture of where that port is); The network has been reconfigured not to use this link, but you may find it slow for some traffic. Please check the connections or tell me to email a fault report to your service engineer."

7. How are the offending devices causing a fault pin-pointed?

As above, the path is known for each channel, so any error can be associated with a particular subset of the devices in the network. In most circumstances, the devices can be interrogated to find whether they have detected a fault, and the fault can therefore be pinpointed automatically.

3.2.6 Security

Security in a network management context is concerned only with the security of parameters associated with network resources.

No. Security is a fundamental requirement of the home network. The network itself, and the management of it, must ensure that security.

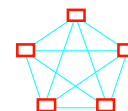
1. How are values of resources protected from being set or retrieved from unauthorized sources?

They can only be set or retrieved by the configuration JavaPP aPplet in each node, and communication between the configuration/management program and these aPplets is secure.

3.2.7 User Interface

An intuitive, easy-to-use user interface is essential for network management services. Such a user interface will allow even a novice home owner to easily identify devices and discover problems.

Agreed, but designing such an interface may not be easy. The presence of this information suggested by the questions in this section underline the need for security --- would you give a crook a complete set of floor-plans and photos of your home?



1. *Can or should the network management system user interface be based on the home's floor plan?*

Could be, but floor plan is two-dimensional whereas the rooms are three-dimensional. Ideal would be a simple (if there is one) VR model of the home, allowing the user to zoom in to place or locate particular end-nodes. And this information can only be used if the network is secure.

2. *Can or should the network management system user interface be based on photos of the inside of the home?*

The screen resolution of photos is unlikely to be adequate for the level of detail that is probably needed. And this information can only be used if the network is secure.

3. *How does the user interface identify a device, an interface, an individual resource, an occurrence of a fault, or an action taken when faults occur?*

In the absence of pictorial information about the location of nodes, the users must identify the nodes with names, preferably that they have given the nodes (but always acknowledging that they will forget the names they have given). Picture plus names is almost certainly best, but again the pictures and even the names are conditional on the security of the network.

3.3 Device and Application Directory

Devices connected to the home network and the applications operating over the home network will often need to locate other devices and applications over the home network. Currently, application-specific or network specific mechanisms exist to allow devices and applications to locate each other. For example, CEBus CAL protocol provides a mechanism to locate devices with specific characteristics on CEBus. Similarly, devices connected to 1394-1995 network can use 1394-specific mechanisms to search for other devices hosting specific functionality by searching the configuration space of all devices on the 1394 network. As is evident, each of these mechanisms is useful only for a portion of the home network and applications are forced to use different mechanisms to locate resources on different parts of the network. A common home-wide device directory alleviates this problem and allows applications to locate all resources anywhere on the network in a uniform, consistent manner. The questions below refer to such a directory service.

As explained in the introductory overview, the mechanisms used conventionally are neither heterogeneous nor secure. Heterogeneity is provided by Sun's FORTH based OpenBoot, but that is still not secure. A configuration mechanism like OpenBoot, but based on Java instead of FORTH, has the required qualities. To make it easier (and more secure and reliable) to program the necessary concurrency and indeterminacy involved, a special class library is added to the language providing CSP-compliant high-level primitives for thread communication and synchronization, and the combination of Java plus the special class library is referred to here as JavaPP.

3.3.1 Directory structure

1. *Is the structure of the directory hierarchical or flat?*

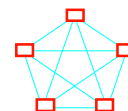
No. It maps the actual topology of the network, providing a set of world-wide-web like links for each of the channels set up between processes in the nodes of the network.

2. *Is the directory centralized or distributed?*

Yes. The JavaPP program can be as centralized or as distributed as is required.

3. *Are the names of objects in the directory variable- or fixed-length?*

The names of objects are the JavaPP representations of those objects, and may be variable or fixed length according to programmers' and users' wishes.



4. *How is information about legacy devices represented in the directory?*

By a JavaPP aPplet modelling the legacy devices and protocols.

5. *What devices are named:*

Nodes, ports, links, processes, channels, protocols --- all JavaPP objects.

6. *All devices on the home network including dumb devices (e.g., light switches)?*

They must be, otherwise how can the user refer to them?

7. *Do devices that just use the backbone for tunnelling become part of the device directory service?*

If they are not used by the network, they do not exist. If they are used by the network in any way at all, they must be named, otherwise how can they be used?

3.3.2 Directory access mechanisms

1. *What is the cost of accessing the directory? Does it support lightweight query and update mechanisms?*

The access mechanism is sufficiently similar to WWW access, and the use of Java is supportive of WWW access, so that any WWW browser should be adequate

2. *What query mechanisms are available for efficiently searching the directory for relevant information?*

A locally hosted local web search engine is required. It probably has some similarity to the file manager/explorer in Windows, but an aPplet designed for the purpose would probably be more appropriate.

3. *Is the directory access mechanism independent of how and where the directory is stored?*

Yes.

3.3.3 Directory schema

In addition to the mechanism used to access and maintain the directory, the schema of the directory entries must be defined.

1. *What objects are represented in the directory?*

Nodes, ports, links, processes, channels, protocols --- all JavaPP objects.

2. *What information about these objects is stored in the directory?*

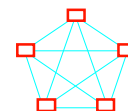
Bearing in mind that the directory can be distributed as necessary amongst all the objects, it holds all the information for the objects to be used by the network.

3. *Is the directory schema independent of directory access mechanisms used?*

Yes.

4. *How do home devices and application directories export information to access networks or incorporate information available in external directories?*

With JavaPP aPplets.



4. Availability

The routing protocol described is compatible with, but is not limited to, IEEE 1355. The 32-port routing switch for 1355 from SGS-THOMSON can be programmed to implement the asynchronous unicast portions of the proposed protocol, and the absence of contention makes it highly suitable also for isochronous traffic.

The proposed simple network routing protocol, for IEEE 1355 and including multicast and priority, has been generically simulated in VHDL by researchers at Nottingham Trent University (UK).

Gbit speeds are not expected to be required in the home, particularly in a switched network where the total network bandwidth is many times the bandwidth of an individual link. But should Gbit speeds be required, an 8-port routing switch chip, with multicast and for the Gbit versions of IEEE 1355, has been developed and proven by Université de Pierre et Marie Curie (UPMC) (France).

Chips for end nodes conforming to IEEE 1355 are available from SGS-THOMSON, Bull, Matra-MHS, and 4Links. Soft macrocells of 1355 circuits are available under licence from SGS-THOMSON, Bull and 4Links.

Network mapping and configuration software for IEEE 1355 has been produced by SGS-THOMSON and PACT, who have also adapted the OpNet tools to 1355 networks. PACT are also just about to publish a cookbook of techniques for building and modelling networks of 1355 compliant components. Other tools for network configuration and loop/deadlock elimination have been produced by the universities of Kent and Oxford.

The special class library for JavaPP greatly eases the programming of thread communication and indeterminacy in Java, and by doing so removes the performance loss from the busy-loops most likely to be used in conventional Java programs. The performance of the primitives is still, however, one or two orders of magnitude lower than if the primitives were built into the language. A possible way to improve their performance further might be to use the CSP-based communication primitives provided in the Virtuoso (Virtual Single Processor) operating system for embedded and DSP systems, from Eonic systems in Belgium.

The special CSP class library for JavaPP has been developed by researchers at the universities of Oxford (UK), Kent (UK), and Twente (Netherlands). Details will be presented at the forthcoming WoTUG conference at Twente from April 13-16.

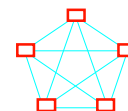
5. IPR statement

The network routing protocol described, and the means of combining a high-speed network with a low-speed network such as CEBus described in the appendix, are the subjects of patent applications from 4Links and Keele University. Licences will be made available to applicants under reasonable terms and conditions that are demonstrably free from any unfair discrimination.

The use of Java or JavaPP for configuration and management has not been patented but was published by the author to two email reflectors on March 14 1997.

6. Acknowledgements

Grateful acknowledgements are given to Peter Thompson, of PACT, chairman of the 1355 Association, for alerting me to the RFI. Also to Peter Thompson and his colleagues Andy Jones and Neil Davies for their encouragement on applying 1355 for the home network. Similarly to Barry



Cook of Keele University, without whom the work on which this response is based would never have been done.

The author has had no part in the work on JavaPP, and is particularly grateful to Peter Welch of University of Kent and Gerald Hilderlink of University of Twente for their work towards JavaPP, and to Peter Welch for comments on an early draft and for text included in the overview section. Also to Alan Chalmers and David May of Bristol University, who organized the meeting at which the ideas on using JavaPP for configuration and management crystallized, and who encouraged this response.

Errors and omissions are the sole responsibility of the author.

Appendix: Physical integration of high-speed network with CEBus

Implementation of phantom circuits

Figure A1 shows transformer connections for phantom circuits with centre-tapped secondary windings. Current flowing out of or into the centre tap flows equally down the two windings and so produces no magnetic flux in the transformer, and so sees no inductance or impedance other than the DC resistance (and leakages) of the windings. As can be seen from Figure A1, the phantom circuit is common mode on each twisted pair but is differential mode between the two twisted pairs.

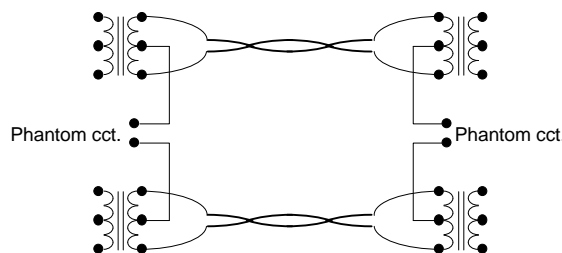


Figure A1: Hybrid transformers with centre tapped secondaries for phantom circuits

Phantom circuit for delivering power

Figure A2 shows possible use of the phantom circuit to deliver power. The use of both wires of each twisted pair to conduct the power supply current reduces the cable resistance and resulting power loss compared with a single dedicated twisted pair for supplying power. The suggested choice of 24 volts is easily supplied by lead-acid batteries, is safe, and can supply 10 to 30 Watts or more per pair of pairs in the cable. The suggested use of positive 0V means the DC-DC converter has to be inverting, and does not have a failure mode where the full 24V can be supplied to the 5V or 3.3V load. The combination of battery charger, power supply and batteries is equivalent to a normal Uninterruptible Power Supply (UPS) except that the UPS does not need the inverter back to mains voltage, so is described here as a "DC-UPS". Use of such a circuit in domestic security alarms is common, but this more general use is perhaps less common.

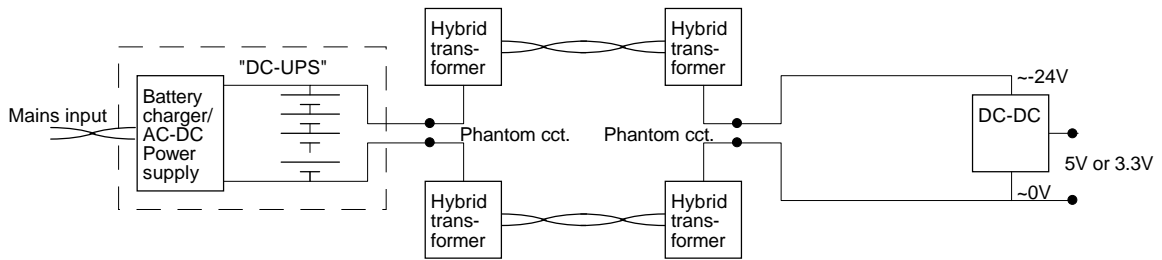
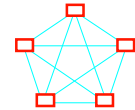


Figure A2: Phantom circuit used for power distribution

Signalling on phantom circuit

Figure A3 shows the phantom circuit used for one of the "Power Line" signalling schemes such as CEBus or LON. These normally connect to the 120V or 240V AC mains, but would connect much more simply to the DC power supply as does the WattCAN version of the CAN standard, or as do the POTS and ISDN phone lines.

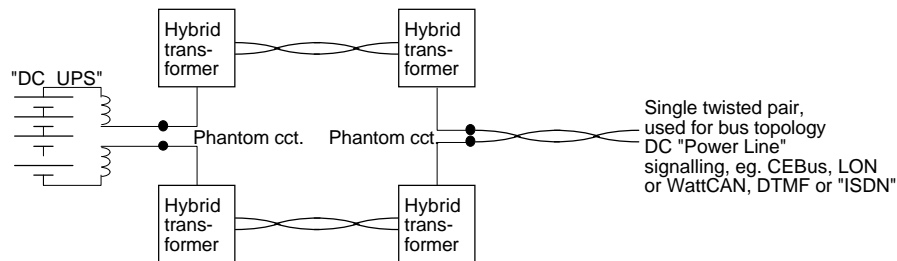


Figure A3: Phantom circuit supplying power and used for bus signalling