A Real-time Network at Home

Ferdy Hanssen, Pieter Hartel, Pierre Jansen, Hans Scholten, Wiek Vervoort
Distributed and Embedded Systems group
Faculty of Computer Science — University of Twente
PO-Box 217, 7500 AE, Enschede, the Netherlands
Phone: +31 53 489 3744, Fax: +31 53 489 4590
E-mail: hanssen@cs.utwente.nl

Abstract— **This paper proposes a home network which integrates both real-time and non-real-time capabilities for one coherent, distributed architecture. Such a network is not yet available.**

Our network will support inexpensive, small appliances as well as more expensive, large appliances.

The network is based on a new type of real-time token protocol that uses scheduling to achieve optimal token-routing through the network. Depending on the scheduling algorithm, bandwidth utilisations of 100 percent are possible. Token management, to prevent token-loss or multiple tokens, is essential to support a dynamic, plug-and-play configuration.

Small appliances, like sensors, would contain low-cost, embedded processors with limited computing power, which can handle lightweight network protocols. All other operations can be delegated to other appliances that have sufficient resources. This provides a basis for transparency, as it separates controlling and controlled object. Our network will support this.

We will show the proposed architecture of such a network and present experiences with and preliminary research of our design.

Keywords— **real-time, network, domotics, embedded systems**

I. Introduction

In the beginning there were only a few main-frames shared by many people. This is called the first wave of computing. Then came the second wave, or the era of personal computing. People do not have to share a computer with other people. This made life much easier, but the relationship between human and computer is sometimes an uneasy one. One distinguishing characteristic is that we are very much aware that the computer is there and that we use it.

Slowly other types of computers entered our homes and replaced their electronic or mechanical equivalents: embedded systems in TV-sets, set-top boxes, hi-fi equipment, microwaves and washing machines. This led to the idea to connect these appliances to make 'something new'. This is called the third wave of computing, or ubiquitous computing. Its highest ideal is to make computers "so embedded, so fitting, so natural, that we use them without even thinking about it"[11]. So, in theory we should be able to control the house, or get services from it, from any device at home. In addition we could have location independant remote access via Internet or mobile phones. But how to do this?

In this context the following problems/challenges exist (no solutions are given yet here):

1. **Compatibility and interoperability**: connecting appliances does not necessarily mean that they can work together. Appliances use different data types, physical networks, network protocols etc.
2. **Network incompatibility**: even if appliances could use the same network (see previous point) they would interfere with each other's proper operation. For instance, multimedia streams can monopolise the network because of their large volume and prohibit data transmission from crucial appliances, like a fire alarm. Or bursty network traffic, as when using the Internet, could cause hiccups in multimedia streams.
3. **Data storage redundancy and data incompatibility**: certain appliances have overlapping or redundant storage, with different formats, e.g. minidisc recorder and video recorder. As a consequence, music from a minidisc is only available through a minidisc player, and the same applies for video. It is difficult to distribute audio and video if there is not one data format that is understood by all appliances.
4. **Resources**: when even the smallest of appliances need to be connected, costs for the embedded computer system become increasingly important. To reduce costs, these systems must be lean and be as small as possible.
5. **Ad hoc configuration**: the configuration of the system must be dynamic. That is, a user of the system should be able to add new appliances to the network without resetting the rest of the system. Setup and configuration must be done automatically. The reverse —removing appliances without system

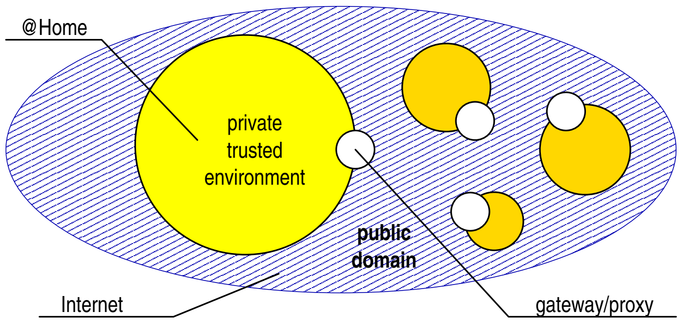

Fig. 1. @HA architecture

interruptions— should also be true. To accept such interruptions, the system must be robust and fault-tolerant.

6. **World wide interconnection**: at some point the house will be connected to the outside world, where different networks and protocols are used, and different levels of security are needed.

II. At Home Anywhere

We will resolve, or give directions on how to resolve these problems/challenges within the *At Home Anywhere* (@HA) research project. In the following sections the objectives will be made more explicit, following the points above.

The @HA architecture constitutes two types of rings (figure 1). The inner ring is the system at home (@Home). The second, outer ring is the Internet. As there are many houses, the Internet contains many @Homes, but from one house's viewpoint there is one @Home and all the rest belongs to the Internet. A gateway, or proxy, connects both worlds.

The first four sections refer to @Home, the fifth one to the connection of @Home with the Internet.

A. *Compatibility and interoperability* and *network incompatibility*

Presently, a house has several separated distribution and communication infrastructures: telephone, cable TV and radio, satellite TV, PC network, connection between thermostat and central heating boiler, etc. In most cases these infrastructures are isolated islands that interconnect only on rare occasions. In general, these infrastructures can be divided in three classes:

- *entertainment*: audio, video, games, etc. This class requires high bandwidth and real-time responses. Characteristic for this class is isochronous, streaming data.
- *control*: sensors and actuators, e.g. central heating control, fire detection, burglar alarm, etc. Control uses low bandwidth, but requires a high degree of dependability. Some devices may need real-time services.
- *information*: PC applications, World Wide Web browsing, etc. This class uses bandwidth in bursts of data, and only needs best-effort responses.

The first step to connecting appliances is one common, inexpensive infrastructure that supports entertainment, control and information. This infrastructure may incorporate different types of networks, including wireless networks. Most efforts till now concentrate on only one class of appliances, mostly entertainment.

The first objective of the @HA network is to be a network for entertainment, control and information that supports both real-time and non-real-time data, as well as streaming media. This network will be based on a new variety of a rotating token protocol, giving bandwidth to the appliance that has the token. In existing timed-token networks, every node in the network is visited once during one rotation of the token. In the worst case, timed-token networks have a utilisation that is quite low, around 33 percent. Main properties are described in [8] and [10]. Examples are the IEEE 802.4 token bus, the IEEE 802.5 token ring and FDDI.

We propose a new type of real-time token protocol, that can be used on top of low-cost network hardware, e.g. IEEE 1394, Bluetooth or ethernet. In contrast to other protocols, the token does not follow a fixed rotation path, visiting every node during each rotation. Instead, the token is scheduled and follows a dynamic route, visiting only those nodes that need attention. The scheduler is decentralised and resides in every node. When necessary, a node may reschedule the token and give it a new route through the network. This happens when new connections are added, or old connections are changed in the network (e.g. for changing the quality of service (QoS) for certain streams). Before applying the new schedule, the node will check if this schedule is feasible. Token management needs special attention to make this type of network robust. It has to be investigated what happens if the token is lost (e.g. the device that holds the token is removed), or if multiple tokens exist.

B. *Data storage redundancy and data incompatibility*

A key issue for integration is a mixed-media storage server, that will provide storage for all appliances

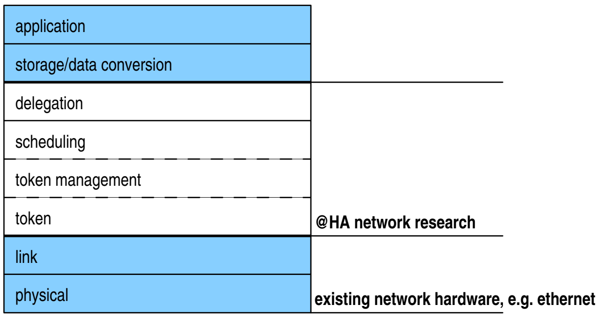

Fig. 2. @HA network

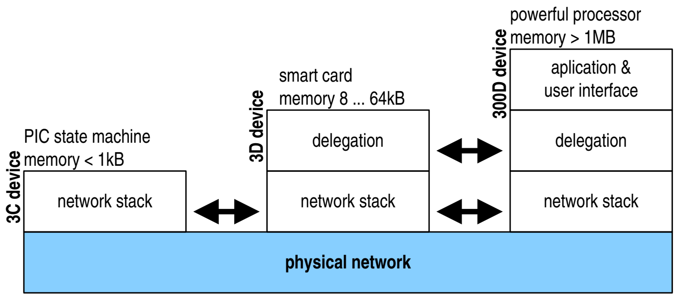

Fig. 3. @HA classification of devices

and devices at home. The storage server must support streaming media, as this is one of the important data types in the system. One of the functions the storage server must offer is simultaneous real-time recording and playback of multiple video and audio streams on harddisk and optical disc (DVD). Time shifting (the ability to pause live broadcasts) is one of the applications of such a storage server. Information is stored once and will be accessible by all. Because not all appliances use the same data format, real-time data conversion is needed between storage and appliance. Efficient storage and retrieval of data is essential, as the amount of data in the file server is huge. E.g. one hour video in DVD quality measures several gigabytes of data. A side effect of this all is that peripherals are shared by all. A good starting point for this research is the work done in our group by Bosch [1].

We will assume that such a storage server is present in the system. Because real-time data conversion is a high level service in the system, our assumption does not interfere with other objectives (see figure 2).

C. Resources

Even small appliances (resource-lean), like a temperature or light sensor, should be connected to the network, but their size and price preclude 'heavy' processors to accomplish that. We will investigate the concept of delegation, or controlled invocation: small systems use their limited processing power to implement at least a network stack to connect to the network and a small real-time operating system kernel to handle the lightweight protocols. But even such an implementation with a small footprint operating system kernel may be too resource-rich. In some cases a simple state machine that handles the network protocols is already enough to do the job.

Depending on the type of appliance, a balance must be found between hardware and software. Hardware can range from a simple Peripheral Interface Controller (PIC) to a full-fledged processor system, while software can range from a simple run-time system executing a state machine to an operating system executing a complex programme. In our case we distinguish three classes of appliances:

- *3C (3+ cent) appliance*: simple devices that implement only a network stack to connect to the system. There is not even enough processing power to implement the delegation protocol. Network handling is done by a PIC executing a state machine. Memory requirements for this class of devices are low (< 1kB). Other devices can scan these devices to get their data. A temperature sensor is an example of such a device.
- *3D (3+ dollar) appliance*: medium complex devices that implement network stack and delegation protocols on a small, smart card-like embedded processor. Memory requirements are medium, between 8 and 64kB. Examples are PC peripherals, like printers and scanners.
- *300D (300+ dollar) appliance*: powerful devices, controlled by a complex embedded computer. These devices can scan 3C devices and are used for delegation by 3D devices. The embedded computer can be used for other functions as well, e.g. it can get an electronic programme guide (EPG) from the satellite video stream, interpret the EPG and switch on and off the digital video recorder according to the preferences of members of the household. 300D devices have a fully-fledged operating system (e.g. Linux) and contain peripheral devices, like disks. Their memory requirements are high (> 1MB). Examples are TV-sets and PDAs.

An overview of this classification is depicted in figure 3.

Delegation is the basis for the first type of location transparency (home location transparency): the user interface and underlying application of any appliance are available anywhere in the house. For instance, the settings of the thermostat of the central heating may be inspected and changed on any available display, be it TV-set, PDA or PC.

The concept of delegation is not new. The X Window System separates application and user interface, so each can execute on a different network node. But this system has a lot of information interchange between client and server and is not suitable.

HAVi [5] does something similar. It distinguishes between controlling and controlled device. Applications and user interfaces (called havlets) are written in Java and allow for flexible and powerful extensions and modifications. But some devices we have in mind might be too small, even for Java virtual machines. HAVi is heavily based on the IEEE 1394 network standard [6] and uses its underlying protocols. Additionally, HAVi is only meant for audio and video devices. Even so, HAVi is very promising and could be a good starting point for our own research.

Our network at home constitutes a closed environment. This has two consequences. Firstly, delegators can trust delegatees and vice versa. Secondly, devices understand a common base vocabulary. It should then be possible to base a delegation protocol on the exchange of keyword-value pairs, with typing constraints added to ensure consistency and robustness. This is essentially the same approach as taken in the CC/PP proposals [2][9]. See also section II-E for location transparency of the second kind (world wide location transparency).

We will develop and implement lightweight protocols suitable for 3C, 3D and 300D type embedded systems. This includes modelling and simulating these protocols. The protocols must support delegation. A model for delegation as well as location transparency will be researched and demonstrated in a prototype. The prototype will include 3C, 3D and 300D appliances.

D. *Ad hoc configuration*

Ad hoc configuration is generally known as plug-and-play. Both network scheduling (section II-A) and lightweight protocols (section II-C) must support a robust and fault-tolerant dynamic (ad hoc) configuration.

E. *World wide interconnection*

The Internet uses its own protocols (telnet, FTP, HTTP etc.) and is highly unregulated. This poses an inherent threat to the integrity of the system at home. However, the Internet also brings opportunities for the system at home. E.g. systems support and software updates could be performed by the manufacturer of an appliance without user interference. Or, the other way around, a user could inspect and control the state of the system at home from anywhere at the world as if he were at home.

This is the second type of location transparency (world wide location transparency). In this case delegation must be secure, and security policies and enforcement become key issues [3]. We will explore world wide location transparency, and investigate if home location transparency basics still apply and can be used as a basis.

III. Simulation and prototype

Although no final choices have been made yet, for demonstration of the prototype the two following areas of application are chosen:

integrated home system
- central storage server;
- sensors controlled from any suitable device at home;
- audio and video anywhere at home (record, playback, time shifting);
- PC applications.

health care ubiquitous computing
- real-time wireless monitoring and alarms;
- real-time remote assessment in case of emergencies;
- secure access to files and information from any (mobile) device;
- consultation by video and audio.

We have created a simulation of our real-time network token protocol on top of ethernet [4]. This simulation currently supports real-time and non-real-time data traffic, where the real-time data traffic is scheduled using pre-emptive Earliest Deadline First (EDF) [7]. Various recovery and network management techniques are implemented to handle network or node failures. Also ad hoc configuration of newly plugged-in network devices is supported.

In figure 4 we show one of the results we obtained by simulation [4]. The network simulates three real-time streams, with periods of 0.01, 0.02, 0.05, 0.1, and 1 seconds. The offered stream loads which are used, are 0.1, 0.2, 0.5, 0.7, and 0.8. Each simulation ran for five minutes, where the actual measurements were taken between the 10^{th} and 290^{th} second, and then averaged. The maximum real-time stream utilisation was limited to 80 percent of the total network bandwidth, so that 20 percent of the network bandwidth stays available for non-real-time traffic. The graph

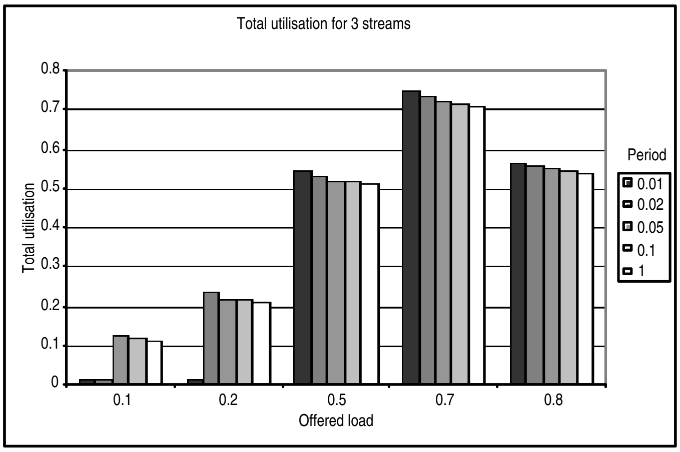

Fig. 4. One of the simulation results

shows the actual utilisation measured, including protocol overhead.

Some utilisations are zero. The combination of period and offered load meant that all streams were rejected, because the computation time was below the minimal token holding time. This limit prevents the token holding time from becoming too small to be of use for the transmitting node. Also, for an offered load of 0.8, not all streams were accepted because the utilisation was not allowed to go above 0.8.

The graph shows that the protocol overhead is higher with smaller periods, but not very high.

IV. Related research

In this section we will present some other projects related to home networking. The following projects all have home networking as subject, but most are restricted in one way or another. Some are for audio and video systems only, others lack real-time behaviour or assume unlimited bandwidth, yet others lack integration of the home network with the Internet or take it for granted.

A. HAVi

Philips, in collaboration with Grundig, Hitachi, Panasonic, Sharp, Sony, Thomson, and Toshiba, announced a specification for interoperability for audio and video equipment called HAVi. HAVi is meant to give a standard for the transfer of digital audio and video information between HAVi devices, as well as processing this information (render, record and play). HAVi devices are intelligent and capable of controlling other HAVi devices over the network. Devices may be connected and disconnected to and from the network without affecting other devices. After connection a device searches for a free ID and automatically configures itself. This feature is called 'hot plug-and-play'. Although HAVi does not address other devices than audio and video, its protocols might be a good starting point for our network, as is its ability to export user interfaces by means of Java code (havlets).

B. Jini

SUN's Jini is a connection technology that enables devices to connect to a network so they form an impromptu community. Like HAVi, Jini detects and configures devices in the network dynamically, though it uses a different mechanism. Each device may use services provided by other devices in the network. Because Jini is based on Java, it is portable to a wide range of devices. Jini defines connecting and configuring devices, but there are no specifications for the exchange of information between devices. Neither does it give real-time requirements, which makes it unsuitable for certain classes of applications. Platform independency and the use of Java are useful ideas for our network.

C. Java Card

With Java Card, smart cards can be programmed in a subset of Java enjoying the benefits of object orientation. Both the card's operating system and the more application specific programming can be done in Java. The second major advantage of Java Card is the support for card applets. These applets take care of all application specific processing in a structured, efficient and secure manner. Card applets are downloadable and provide the opportunity to dynamically manage the services provided by a card. This facility may form the basis for the implementation of a delegation service. Smart cards communicate to the outside world over a serial line interface. This could be used as the basis for the implementation of a network stack. Java based smart cards can be bought for as little as 3 dollars, which makes them suitable for use in inexpensive home appliances.

D. Universal Plug and Play

Universal Plug and Play (UPnP) is an extension to plug-and-play and transparently connects appliances, PCs and services. It is entirely based on Microsoft operating systems. The transport layer may be anything from ethernet to Firewire and USB systems. Again, no real-time requirements are specified, and its dependency on Microsoft operating systems limits its use to certain platforms.

E. HomeRF

The HomeRF Working Group (HRFWG) provides the foundations for consumer devices that use wireless digital communication for connections, specifically mobile voice and data. HRFWG has developed a specification called the Shared Wireless Access Protocol, or SWAP. It addresses connectivity between the Internet and devices at home, but operates with limited bandwidths. The protocol may be useful for mobile devices in our environment.

F. DVB Multimedia Home Platform

The Multimedia Home Platform (MHP) provides the interconnection of multimedia equipment via the in-home digital network. MHP plans to cover the whole set of protocols, APIs, interfaces, and recommendations to implement digital interactive multimedia at home, and is working on the first version of the specification. We keep contacts with MHP researchers in the industry.

G. Phenom

The Phenom project is a joint project between the Eindhoven Embedded Systems Institute (EESI) of the Eindhoven Technical University and Philips Research Laboratories. "The aims of the Perceptive Home Environment (Phenom) project are to investigate techniques and requirements for creating ambient intelligent systems in the home environment, and to explore the possibilities for new application areas made possible through the presence of an ambient intelligent environment." We do not aim at the creation of 'ambient intelligent systems', but rather at the integration of home networking and the Internet and its enabling technologies. Our network's topology is dynamic, while Phenom envisions a more static one. We will closely follow the progress of this research.

V. Concluding remarks

This paper describes a proposal for a real-time network at home, and the project has started by building a prototype of this network. Already we have successfully built a simulation of our networking ideas. This simulation can achieve a network utilisation of 100 percent, has a straightforward feasibility analysis for the real-time streams, and can quickly recover from network errors while doing a good job of maintaining soft real-timeness of the real-time streams.

We are currently investigating IEEE 1394 as a base network protocol for our network, and we are building a prototype of our network on ethernet for a real-time variant of Linux.

References

[1] H.G.P. Bosch, Mixed-media file systems, Ph.D. thesis, University of Twente, ISDN 90-365-1277-8, 1999

[2] CC/PP Working Group web site, http://www.w3.org/Mobile/CCPP/

[3] Dieter Gollmann, *Computer security*, chapter 10, John Wiley & Sons, ISBN 0-471-97844-2, 1999

[4] Tjalling Hattink, HOTnet, a real-time network protocol, Master's thesis, University of Twente, 2001

[5] Home Audio and Video Interoperability web site, http://www.havi.org/

[6] IEEE 1394 web site, http://www.1394ta.org/, complete specifications at http://www.ieee.org/

[7] C.L. Liu and J.W. Layland, Scheduling Algorithms for Multiprogramming in a Hard Real-Time Environment, *Journal of the ACM*, **20(1)**:40–61, 1973

[8] N. Malcom and W. Zhao, The Timed Token Protocol for Real-Time Communications, *IEEE Computers*, **10(1)**:35–41, January 1994

[9] Resource Description Framework web site, http://www.w3.org/RDF/

[10] K.C. Sevcik and M.J. Johnson, Cycle Time Properties of the FDDI Token Ring Protocol, *IEEE Transactions on Software Engineering*, **SE-13(3)**:376–385, March 1987

[11] Mark Weiser, Hot Topics: Ubiquitous Computing, *IEEE Computer*, **26(10)**:71–72, October 1993