

# On Connecting Smart Appliances to the Internet

Ibrahim Kamel  
Collage of Information Systems  
Zayed University  
Ibrahim.kamel@zu.ac.ae

## Abstract

*This paper presents the state of the art in the Internet appliance technology. It presents the main technologies that are required for enabling networked appliance applications and the different standardization forums that are concerned with these technologies. We outline some of open research problems and propose directions for solving these problems. The papers presents in more details problem definition and solution in the following three areas:*

- *Quality of services in home networking*
- *Memory management in home gateways*
- *Security issues in Internet appliances*

*We present some application scenarios that can benefit from Network appliance applications. Finally, we discuss what is needed to make network appliance a reality.*

## 1 Introduction

With the widespread of the broadband to homes, Internet is becoming more and more popular inside the homes. Internet is used not only for connecting computing devices such as PCs and PDAs but also for home appliances like TV, DVD, washers etc. Merloni Elettrodomestici, an Italy based company announced their Internet washer Margherita2000 [7] that can be connected to the Internet through a modem. The manufacturer or the owner can program the Internet washer remotely. Matsushita Electric showed its Panasonic Internet enabled microwave during the 2001 Consumer Electronic show. Cooking recipes and heating instructions can be downloaded to the microwave.

It is expected in the near future that all the appliances in the house will be connected to the Internet. Remote diagnosis and configuration of the appliances is one of the many advantages that the consumer can gain from Internet appliances. In the entertainment domain there are several interesting applications. Users can download movies on demand and an Electronic Programming Guide (EPG) that contains the TV schedule. Power companies

are keeping an eye on home networking because it will allow them to provide value-added services such as energy management, remote measurement, and better power balance that reduces the likelihood of blackout.

There are several initiatives to define the specification for API and the protocol for networks inside the home, like HAVi [2], UPnP [5,6], Jini [4], HomeRF [3] to name a few. It is expected that multiple home networking will coexist in the home and inter-operate through the home gateway. The home gateway acts as a single point of connection between the Internet and the home. OSGi [9] (Open service Gateway initiative) is a consortium of companies committed to define common specifications for the home gateway. According to OSGi model, the gateway can host services to control and operate home appliances.

There is no consensus among appliance manufacturers on whether the gateway will be a separate box or it will be integrated in home appliances like DTV or STB (Set-Top-Box), or whether the gateway functionality will be centralized in one device or distributed among several appliances. However, the gateway will generally have limited resources, especially main memory. Main memory is required to run system administration, services and applications for home devices. This problem will be more severe if the gateway functionality is implemented in home devices like STB or TV.

The next section briefly describes home networking technologies and home gateway. In Section 3 we discuss various research problem related to connecting smart appliances to Internet. We then choose one of the problems, memory management in home gateway and discuss it in more details and present possible solutions. A general discussion about smart appliance market and research is presented in Section 5.

## 2 Home Networking

There are several initiatives for developing networking protocols that can be used inside the home. Home environment is drastically different than computer lab

environment. After all, the home user might be computer illiterate. Network administration should be intuitive. Home appliances are not as powerful or intelligent as computer equipments. Most of legacy devices are dumb. There might be a small processor and memory in high-end home appliances. From the economic point of view, the profit margin in home appliances is slim and adding intelligence to the devices will increase its prices.

Following the above argument, one can see that using the technologies that are developed for desk computer environment might not be suitable for smart appliance in home environment. Thus, industry and academia started to develop new technologies that are suitable for the home environment. The following are the main requirements for such technology:

**Light Implementation:** Home devices are, in general, inexpensive and cannot afford installing powerful CPU, large memory and disk. Yet there is a need for middleware and protocol stacks to handle the communication among and between the smart appliances and the Internet. The proposed protocol stack should be light. Ideally, the proposed protocol should be an operating-system-neutral middleware.

**Easy-to-Use (Plug and Play):** Home devices are expected to automatically announce their presence and capabilities to every other device on the home network, greatly simplifying installation and setup. No complicated and configuration of network addresses or device drivers.

**Developing Standard:** It is required to develop protocols stack and API, e.g., communication and discovery that are transparent of the device manufacturer. It should allow products from different vendors to comply with one another based on defined connection and communication protocols and APIs.

**Backward compatibility and Upgrade-ability:** The new smart home paradigm should allow the possibility of protocol upgrade and new features to be deployed. Updating functionality can be done by downloading (uploading) new capabilities via the Internet.

**Legacy devices:** It is not expected that home user will replace all their home appliances with smart devices at once. One should expect a number of legacy devices that are connected to the network. Also home devices vary in their capabilities and power. Thus, devices that cannot host full implementation of the protocol stack should be proxy-ed by other devices.

The path from the home appliance to the service providers can be divided into three sections as shown in Figure 1.

The first one is the network inside the home. The second one is the home gateway and the third one is the connection between the home gateway and the Internet (ISP). This connection is using IP protocols over various physical layers, e.g., Cable, ADSL, POTs, and Satellites. There is only one initiative for developing gateway middleware and specifications, OSGi, which is explained below. One of the main issues in connecting home devices to the Internet is the lack of agreed-upon physical network and protocol inside the home. Currently, there are many standard initiatives developing home networking protocols and API assuming different network wiring, e.g., Ethernet, IEEE1394, phone-line, power-line, wireless, etc. Many people will be reluctant to deploy new wires in their homes, especially, old ones. Some proposals are competing with each other like, HAVi, Jini, UpnP. The computer industry, e.g., Microsoft, Intel, are pushing to develop a computer centric home networking environment, while, consumer electronic manufacturers, e.g., Sony, Panasonic, prefer to develop home networking protocols that are independent from the computer. This section briefly mentions some of these initiatives.

## 2.1 HAVi (Home Audio Video Interoperability)

HAVi is a digital AV networking initiative that focuses mainly on audio and video communication [2]. It is built on top of the IEEE 1394 physical layer. It provides a set of services that facilitate interoperability and the development of distributed applications on home networks. When a new device is installed, the system will configure itself to accommodate it. Other services provided by the distributed application system include: addressing scheme and message transfer, lookup for discovering resources, posting and receiving local or remote events, streaming and controlling isochronous data streams.

Each appliance added to the network automatically installs its own application and interface software. HAVi has standardized the Application Programming Interfaces of the most common AV functions. This means that a VCR can search for an appliance that offers a clock with the time-of-day and automatically set its own timers.

## 2.2 HomeRF

HomeRF developed a specification for wireless communications in the home called the Shared Wireless Access Protocol (SWAP) [3]. The SWAP specification defines a new common interface that supports wireless voice and data networking in the home. Set up a wireless home network to share voice and data between PC's,

peripherals, PC-enhanced cordless phones, and new devices such as portable, remote display pads. Access the Internet from anywhere in and around the home from portable display devices.

### 2.3 HomePNA

Home PNA stands for Home Phoneline Networking Alliance. This technology allows all appliances in home to interact over the home's existing telephone wiring without disturbing the existing voice or fax services. With their latest specification, HomePNA 3.0, can reach a data transfer rate of 128 Mbps with optional extensions reaching up to 240 Mbps.

### 2.4 HomePlug

HomePlug's vision is to deliver Internet and multimedia from every home power outlet and to enable the home through worldwide home power-line networking standards [8]. With multiple outlets in almost every room, residential power-lines are already the most pervasive network in the world.

### 2.5 UpnP

Universal Plug and Play (UPnP) is architecture for pervasive peer-to-peer network connectivity of PCs of all form factors, intelligent appliances, and wireless devices [5]. It uses Ethernet in the physical layer. UPnP is independent of any particular operating system, programming language, or physical medium. UPnP learns from the Internet's success and heavily leverages its components, including IP, TCP, UDP, HTTP, and XML. UPnP effort is headed by Microsoft.

### 2.6 OSGi Platform

The OSGi Alliance was initiated in March 1999 by 15 companies in response to technology and market movements related to broadband networking and remote connectivity [9]. The Open Services Gateway specification is a Java based application layer framework that gives service providers, network operators, device makers, and appliance manufacturers vendor neutral application and device layer APIs and functions.

OSGi is concerned with the development and proliferation of specifications to enable delivery of managed broadband services to networks in home, vehicle, mobile, and other environments. This Specification defines an open, common architecture to be implemented in a services gateway such as a set top box,

cable modem, DSL modem, PC or dedicated residential gateway, Figure 1. The Specification consists of Java APIs and a service framework, which provides an environment for electronically downloadable services called bundles. This environment includes a Java runtime and adds life cycle management, persistent data storage, version management and a service registry.

## 3 Research Issues in Internet Appliances

Connecting home appliances to Internet poses real challenges. There are not many research initiatives in this area yet. In addition of being a fairly new area, the market is not there yet. Although industry realized the market potential for this area, but still it lack the killer applications that make the consumer willing to pay more money for the smart devices and infrastructure. Most of the consumer electronic companies now have active research groups who focus on smart appliances. However, they are busy trying to win standardization battles. The following are some areas where research effort are needed:

- ❑ **Quality of service guarantee:** many of the smart appliance applications at home use real-time data objects, like, audio and video. Given that fact that there is limited bandwidth to/from the home, we need to optimize and manage the available bandwidth carefully [11]. At the same time, we need to take the importance (priority), and application-dependant factors, like, whether the application can work with fewer resources or not; whether the application can be paused or not.
- ❑ **Home gateway resource management:** the gateway is a single point of entry sharing resources inside home gateways such as CPU, memory, and disks. These applications compete with each other. At the same time some of the applications depend on each other and some of them are more important than others [1]. Note that traditional memory management for computers are not suitable because of the dependency between tasks and priorities.
- ❑ **Privacy and Security:** Currently Internet reaches to you desktop computer but once it reaches inside the kitchen, bedroom, surveillance system, there might be serious security concern. From the security point of view, these devices might be subject to attacks if they are not well protected [12]. One can think of viruses that hit the refrigerator, VCR or the TV system. Moreover, There will be variety of private data that travel through the Internet. For example, the

front door is not locked, images from inside the home, electricity and water usage. These data

will be vulnerable to both passive and active attacks.

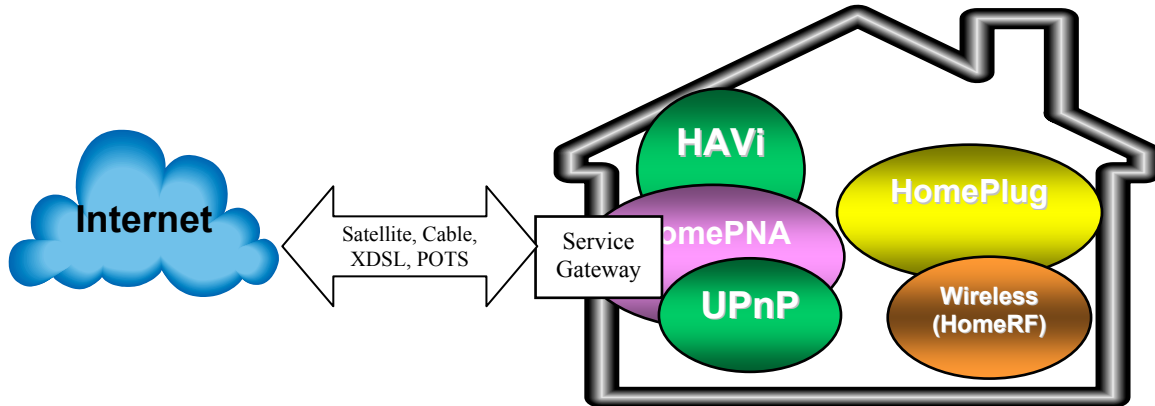


Figure 1: Various networking protocols inside and outside the home

In the following sections we motivate for some real research problems in connecting smart appliances to Internet and provide solutions to such problems. Because of the space limitation we will pick one of these area to be discussed in more details. In the next Section we are discussing memory management in home gateway and present some solutions to the problem.

#### 4 Memory Management in Home Gateways

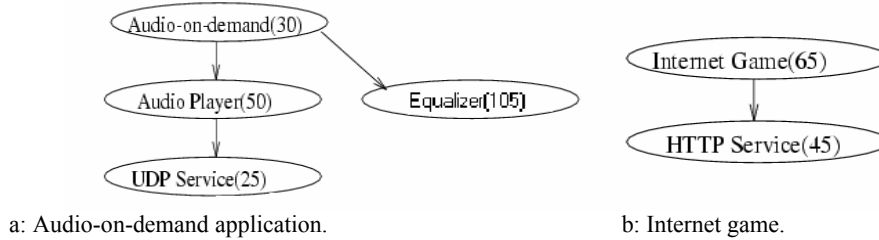
In this section we describe briefly some of the memory management in gateways and proposes new algorithms for managing services and bundles in a limited memory environment. Home gateway is the connection point between the network inside the home and the external world (Internet) as shown in Figure 1. We address this problem in the context of the software architecture proposed by OSGi [9]. Memory management in computers has been an active area of research for several decades [10]. This research spans multiple disciplines such as operating systems, database management to name a few. In the OSGi model, services are implemented in software bundles that can be downloaded separately from the Internet and executed in the gateway [1]. For example, HTTP service is implemented in a bundle. Other applications like home security, energy management, and Internet game each can be implemented in a separate bundle. Bundles can depend on each other; for example, a home security bundle uses an HTTP service to provide external connectivity. Bundle and service management in gateways differs from traditional memory management in the following aspects:

- 1) The memory unit in traditional memory management techniques is the disk page, whereas the memory unit in a gateway application is the memory required to execute the bundle.
- 2) Traditional memory management techniques assume that disk pages are independent (although, depending on the application, disk pages might be dependent on each other, for examples, two pages belong to one loop) while bundles may depend on each other as explained above.
- 3) Bundles might have priority levels associated with them that are determined by the importance of the corresponding applications.
- 4) Kicking a bundle from the memory results in aborting the service, while kicking a page from the memory costs one I/O.

The gateway might need to free memory space to accommodate a new service that is triggered by connecting a new device to the network or upon explicit local or remote request. Although the amount of memory required to execute a service is changing with time, the application service provider (or the author who provides the bundle) can give approximate estimates such as average and maximum amount of memory required to execute the services in this bundle. Memory space might be requested by a running service instance that requires more memory to complete its service. At this time the gateway framework has to pick a victim service instance or instances to terminate to fulfill the request. Note that the victim service instance might be the requesting service itself. The proposed algorithm [6] has to fulfill the following criteria:

1) The number of victim service instances should be minimized.

2) The victim service instance is the least important among the running ones.



**Figure 2: Dependence graph for two currently running applications (numbers indicate memory requirements).**

- 3) The algorithm should be fast enough and does not require much memory.
- 4) The total amount of reclaimed memory is enough to fulfill the requested memory.

This can be best described by the following simple example. Suppose that there are two applications that are running on the gateway. The first application uses audio-on-demand service that depends on the audio player service that, in turn, depends on the UDP service. The service dependency graph for the audio-on-demand is shown in Figure 2:a. The second application (Figure 2:b) is the Internet game, which has a game service that depends on the HTTP service. Now we would like to start the home security service, which requires 100 memory units. Apparently, home security service has a higher priority than both Audio-on-demand and Internet-game. Thus it is reasonable to kick at least one of the services to start the home security service. Assuming that both Audio-on-demand and Internet-game have the same importance level, it makes more sense to kick the equalizer of the audio-on-demand because it results in killing less number of service instances and fulfill the memory demand. In many applications it is possible to kick one service in the application and keep the application running.

#### 4.1 Formal Definition of the Problem

The problem can be described as follows. Let  $S = \{s_1, \dots, s_j\}$  be the set of service instances currently resident in gateway memory. Assume that instance  $s_i$  occupies  $M(s_i)$  storage. Let  $G(S, E)$  be the forest of trees describing the dependency among the set of instances  $S$ . For a vertex  $v$  in  $G$ , let us denote by  $v^*$  the set of vertices of the subtree of

$G$  rooted at  $v$  (including  $v$  itself), and for a subset of vertices  $V \subseteq S$ , let  $V^* := \cup_{v \in V} (v^*)$ .

Given that a new service instance  $s$ , with memory requirement  $M(s)$  has to be created, it might be required to eject some currently existing instances in order to make room for the new instance. Assume that the extra required memory for this operation is  $M_t$  units, that is  $M_t = M(s) - M_f$ , where  $M_f$  is the current amount of available memory. Clearly, when a service instance is ejected, all instances depending on it will be ejected as well. Our goal is to reduce the number of removed instances. More precisely, it is desired to find a subset  $V \subseteq S$  of minimal number of dependents, whose ejection, together with all its dependents, will make available a total memory of at least  $M_t$  units. Letting  $M(S') := \sum_{s \in S'} M(s)$  for any  $S' \subseteq S$ , our problem can be formulated as of finding

$$\min\{|V^*| : V \subseteq S, M(V^*) \geq M_t\}. \quad (1)$$

There is a natural generalization of problem (1), which can be stated as follows. Suppose that different service instances differ in importance and, therefore, are assigned different priorities. In such a case, it is reasonable to assign a weight  $W(s)$  to each instance. Instances with large weights are considered more important. When it is necessary to eject some instances from memory, it is desirable to reduce the number of deleted high priority instances. Thus the problem now becomes of finding  $\min\{W(V^*) \mid V \subseteq S, M(V^*) \geq M_t\}$ . This last problem, in the absence of dependencies, is closely related to the well-known *Knapsack problem*, which is NP-hard in general [6].

## 4.2 Service Replacement Algorithms

In this section we present several algorithms for solving the gateway service replacement problem. The first algorithm selects the service(s) to be replaced based on the amount of memory that might become available. The other two algorithms take into account not only the memory size but also the service dependencies and service priorities. This method is referred to as SDP (Size-Dependency-Priority). The second algorithm is a simple heuristic that runs in  $O(nh)$  time and requires linear space, where  $h$  is the height of the forest and  $n$  is the number of service instances. The third algorithm computes an optimal solution in  $O(n^2)$  time and  $O(nh)$  space.

## 4.3 Simple Algorithms

When a new service arrives or an old service requires more memory and there is not enough memory available, one or more services is simply stopped or replaced to free memory space. This algorithm is similar to the traditional algorithms used in operating system literature for memory management in computers. This simple method makes selection based on the amount of memory and it ignores the service (bundle) dependency and priority. There are several variants of this algorithm depending on which bundle(s) to choose:

**First Fit:** choose the first set of services (it can be one service) such that total memory occupied by them is enough for the request amount. Note that this algorithm does not necessarily scan the whole service list.

**Best Fit:** choose the set of services whose aggregate memory is the minimum value that is bigger or equal to the requested size. We must search the entire service list unless the list is sorted according to the size of memory occupied.

**Worst Fit:** choose the service that occupies the largest memory. In this algorithm, we must search the entire service list if it is not sorted.

## 4.4 SDP Heuristic

Our heuristic greedily tries to pick, as a victim for deletion, the service instance with the smallest number of dependents, whose ejection will free the largest amount of memory. Towards this end, the heuristic will pick for deletion the service instance  $s$  which maximizes the ratio  $M(s^*) / |s^*|$  of the total memory to the number of

dependents. This selection tends to decrease the number of deleted instances. Looking back at the example in Figure 2, we can see that the ratios  $M(s^*) / |s^*|$  for the different service instances are as follows: Audio-on-demand ( $52.5 = [30+50+25+105]/4$ ), Audio-player (37.5), UDP Service (25), Equalizer (105), Internet game (50), and HTTP Service (45). Thus the service instance with maximum ratio is the Equalizer whose removal will give enough memory to start the new service (requiring 100 memory units). Should we have used the First Fit strategy, on the other hand, we might have selected to remove the Audio-on-demand instance, which results in removing 4 instances instead of only 1. Note also that, for this particular example, the Best Fit algorithm would also remove the same instance (the Equalizer) selected by the SDP Heuristic.

Specifically, given the current set of service instances  $S$ , the dependence forest  $G$ , and the memory requirement  $M_t$ , we proceed as follows:

**Step 1.** For each node  $s \in S$ , compute the accumulative size and memory  $|s^*|$  and  $M(s^*)$  as described by the dependence forest  $G$ .

**Step 2.** If  $M_t \leq 0$  then break.

**Step 3.** Pick the node  $s$  with the maximum value of the ratio  $M(s^*) / |s^*|$ . In case of ties, pick the node with the smallest number of dependents  $|s^*|$ . Delete  $s$  and all its dependents.

**Step 4.** For every node  $u$  on the path from  $s$  to the root of the tree containing  $s$ , set  $|u^*| \leftarrow |u^*| - |s^*|$  and  $M(s^*) \leftarrow M(u^*) - M(s^*)$ .

**Step 5.** Set  $M_t \leftarrow M_t - M(s^*)$  and go to Step 2.

It is easy to see that the above heuristic can be implemented in  $O(nh)$  time.

### SDP Optimal

It is well known that the Knapsack problem admits a pseudo polynomial algorithm. In this section, we extend this solution to problem (1). The technique used is dynamic programming. Specifically, let  $S = \{s_1, \dots, s_n\}$  be the current set of service instances listed in pre-order traversal (that is, we recursively traverse the children from left to right then we traverse the root). We shall consider incrementally the sets  $\{s_1\}$ ,  $\{s_1, s_2\}$ ,  $\{s_1, s_2, s_3\}$ , ..., computing for each set the maximum amount of memory that can be achieved by deleting a subset of it of a given number of dependents. Note that  $n$  is a trivial upper bound on the total number of instances (or weight) that can be achieved by any solution. For each  $i$

$\in \{1, \dots, n\}$  and each  $w \in \{1, \dots, n\}$ , let  $S_{i,w}$  denote a subset of  $\{s_1, \dots, s_i\}$  whose total weight is exactly  $w$ , and whose total memory is maximized. Let  $A(i,w) = M(S_{i,w}^*)$  if the set  $S_{i,w}$  exists, and  $A(i,w) = -\infty$  otherwise. We initialize  $A(i,0) = 0$  for all  $i = 1, \dots, n$ , and  $A(0,w) = 0$  for all  $w = 1, \dots, n$ . Let  $L(s_{i+1})$  be the largest  $k \in \{1, \dots, i\}$  such that  $s_k$  is not a child of  $s_{i+1}$ . Clearly  $A(i,w)$  is known for every  $w \in \{1, \dots, n\}$ . The other values of  $A(i,w)$  can be computed incrementally using the following recurrence:

$$A(i+1,w) = \max \{A(i,w), M(S_{i+1}^*) + A(L(s_{i+1}), w - |S_{i+1}^*|)\}$$

if  $|S_{i+1}^*| < w$  and  $A(i+1,w) = A(i,w)$  otherwise. Clearly the sets  $S_{i,w}$  can be easily obtained by a similar recurrence.

The minimum weight of a subset solving our problem is then  $\min\{w \mid A(n,w) \geq M_i\}$ . Thus we get an  $O(n^2)$  algorithm for solving problem (1). The space used is  $O(nh)$  where  $h$  is the height of the forest.

## 5 Discussion

The dream of connecting home appliances to the Internet is becoming feasible because of the recent Internet penetration, and especially broadband, to the homes. Although the broadband penetration in homes is increasing, yet the networked appliances adoption does not go with the same rate. This is due several reasons.

- Lack of One Standard: Several industry-led initiatives for developing standard protocol and API have started. Some of them are addressing the same problem and competing. Unfortunately there is no consensus on which protocol or API to use inside the home. These differences are, partially, due to two reasons. First technical reasons, like, whether to use telephone lines, power lines, Ethernet or wireless as communication media inside the home. Users and homeowners are reluctant to deploy new wiring in their homes especially in old homes. This is why wireless, existing phone-lines and power-lines are interesting options. However each solution has its own advantages and disadvantages. The second reason is competition and market domination. For example, choosing the right platform for middleware implementation is an issue. Whether to use Java platform or native platform that is dependent on specific

operating system. Decision like these will give market advantage for some companies over others. A more fierce battle is about who will be controlling the new smart home. The computer industry wants to push the technology and standard to produce a computer-centric home network. In such environment computer will be the focus of the network. This will give computer manufacturer and software houses the upper hand in the new market. The consumer electronic industry considers home appliances to be their market and they want to protect it, thus, they push for a distributed ad-hoc network environment inside home.

- Although the computer hardware prices are going down, but adding intelligence to home appliances will not come for free. Legacy home appliances like refrigerators or coffee maker are dumb devices in most of the cases. To be connected to the Internet, these devices need CPU and memory, which will increase the cost of these devices. Consumer electronic companies complain that the profit margin of home appliances is very slim so the user has to pay for these additions.
- Killer Application: although there are many good applications that might benefit home users. These applications are fun to have, however, the user is not willing to pay the extra cost and service fee for these application. It is yet to find the killer applications that and provide value to the user life and equal to the service cost. There are several cost items to enable smart home, such as, home network infrastructure, Internet connection, intelligence hardware added to appliances, home gateway to name a few.
- Another important reason that affected home networking field is the lack of a clear business model or food chain among different contributors. Providing valuable services to home users requires the collaboration and coordination between several industries such as, consumer electronics manufacturer, service providers, application providers,

computer software and hardware. It is not yet clear who will charge the user for the services and how these charges will be divided among the different providers.

A combination of the above-mentioned reasons slowed down the emergence of real smart appliance market. There are good interest among researchers to study problems and challenges in home networking and connecting smart appliances to Internet. However, if the smart appliance market do not pick up in year or two, this will definitely have negative effect on the research funds in this area.

## References

- [1] B. Chen, K. Elbassioni, and Ibrahim Kamel, *Bundles Replacement in Gateways*, IEEE International Conference on Wireless LANs and Home Networks, Singapore, Dec 2001
- [2] HAVi Consortium, "HAVi Specification V1.0", 2000, [www.havi.org](http://www.havi.org)
- [3] HomeRF, Technical information, <http://www.homerf.org/techinfo/whitepapers.html>

- [4] Sun Microsystems Inc., Jini Architectural Overview, Sun White Paper, January 1999
- [5] Microsoft Corporation, Universal Plug and Play Device Architecture Reference Specification
- [6] B. Chen, K. Elbassioni, and Ibrahim Kamel, Efficient Service Management in Home Gateway IEEE 4<sup>th</sup> International workshop on Networked Appliances (IWNA4), Gaithersburg, Maryland, USA, January 2002
- [7] Margherita2000, first Internet washing machine, <http://www.margherita2000.com/sito-uk/it/home.htm>
- [8] HomePlug Powerline Alliance [www.homeplug.com](http://www.homeplug.com)
- [9] Open Service Gateway Initiative, OSGi Specification Overview, [www.osgi.org/](http://www.osgi.org/)
- [10] A. Silberschatz and J. L. Peterson, Operating System Concepts, Addison Wesley, 1989
- [11] C. Akinlar, K. Elbassioni and Ibrahim Kamel, *QoS Management in Residential Gateways*, the IEEE International Conference on Wireless LANs and Home Networking (ICWLHN 2002)}, Atlanta, Georgia, August 2002
- [12] C. Akinlar, M. Rahman, Ibrahim Kamel, *On Secured End to End Appliance Control Using SIP*, 5<sup>th</sup> International Workshop on Networked Appliances (IWNA5), Liverpool, October 2002