hBox – Connecting Homes

Matti Siekkinen, Jukka Manner, Sasu Tarkoma, Antti Ylä-Jääski

Helsinki University of Technology
P.O.Box 5400, FI-02015 TKK, Finland
Email: {matti.siekkinen,jukka.manner,sasu.tarkoma,antti.yla-jaaski}@tkk.fi

***Abstract*— Home networks are an increasingly popular platforms for the deployment of ICT solutions for common people. Universal Plug-and-Play has enabled the seamless connection of different home appliances and the services they offer. However, interconnecting home networks, e.g. among friends, in a way that is flexible, easy to deploy and use, and yet well controllable still remains a challenge. This paper tackles this problem and proposes a novel approach for home network interconnection. Our solution consists of a device called *hBox* which is simply plugged to the residential gateway and controlled with the mobile phone of the home network's administrator. The solution requires no manual configuration and the user authorizes new home networks and shares resources to them using his/her mobile phone. We describe the architecture of our solution and present a prototype implementation of the hBox.**

I. Introduction

Homes are an important emerging environment for deploying ICT technologies. Various multimedia devices, e.g., home theaters and media servers, are already common in many homes. The industry has worked to define ways to increase the value of these stand-alone devices through interconnecting them, e.g., a set-top box can discover a media server and enable the user to watch home videos or even Internet content on her familiar TV set. The primary technology to enable this networking of independent devices is Universal Plug-and-Play (UPnP) [1].

UPnP is an XML-based service discovery protocol which also includes mechanisms to actually use the services over SOAP. The primary use case of UPnP is local networks, UPnP runs over multicast and thus only works in closed networks. In order to further progress the interconnection of devices and people, we would like to make these home devices available across the Internet, between people, e.g., families and friends. The standard UPnP is not applicable here since it is designed for small-scale closed deployments.

Our work aims at providing a solution to interconnect devices and services between multiple home networks with each other through the public Internet. When designing solutions for interconnecting homes, we have to make sure that the system is very simple to use, yet secure enough for people to trust it, and that it works with all the legacy devices that are deployed at homes already. Thus, the solution must itself be plug-and-play.

There exists prior work in this area. Authors in [2] propose a solution for sharing individual pieces of content located at home PCs or UPnP AV Media Server to remote clients. However, their solution is currently applicable to only multimedia content and deployment is not very user friendly. The architecture presented in [3] has similar features to the one described in this paper. However, that solution requires modifying the gateway's software and connecting to a new network needs manual configuration of the gateway. Furthermore, key distribution in an issue. In [3], the authors assume pre-shared keys. Intel's Device Relay [4] suffers from the problem that only two networks can be connected, while we want to be able to connect an unrestricted number of home networks to each other. Furthermore, Device Relay does not give users the control to specify which devices of a home network should be shared. Instead, all devices are shared automatically. The solution presented in [5] does not target easy usage and simple deployment. They rather aim for efficient content distribution within a community of home networks.

In this paper we present a novel approach to interconnect multiple homes and their devices and services together. The fundamental goals of our work is to design a system that is plug-and-play taking into account that people want to have control over which devices and content they share with others; in a similar fashion to content distribution in the Internet. In our system a UPnP enabled embedded device (hBox) is plugged into the home network. This device is capable of creating and managing secured tunnels into other home networks which have a similar device plugged enabling UPnP devices and services to be discovered and used between multiple homes. A key concept in our system architecture is the use of a mobile phone to control the services and their use. We use the phone's address book and send authentication messages over SMS. The mobile phone communicates locally with the hBox (e.g. via Bluetooth/WLAN) and remotely through the mobile network with the phones of the users to whom access rights are shared. This makes it possible to share access rights securely and allow standard UPnP devices communicate without any modifications to plug-and-play operations. Users do not need to rely on third party services for authentication and access rights distribution.

The main contributions of our approach are:

- Seamless plug-and-play of hBox with no changes required for legacy UPnP devices and residential gateways which can be of any technology (ADSL, Cable, FTTH, etc.).
- Usage of the mobile phone in authentication and access rights distribution with no need for third party services.

978-1-4244-6273-5/09/$26.00 ©2009 IEEE

The key concept here is the a-priori trust relationship established among friends and family members which is linked to a phone number in our mobile phone's address book.

Although we have built a first proof-of-concept prototype, the focus of this paper is presenting our conceptual solution. Complete prototype implementation and thorough evaluations are part of ongoing work.

II. Architecture

A. Requirements

Our solution is based on implementing necessary functionality in each home network to enable sharing of UPnP services. We identified the following set of requirements for our solution:

Easy to deploy: We do not want users to have to modify in any way the software running at the gateway or the shared devices themselves.

Easy to use: Sharing new devices or allowing a new home network to be connected should not require any manual configuration of the gateway or devices.

Security: The solution should support the interconnection of home networks in a secure fashion. Mechanisms are needed for authentication, authorization, confidentiality, and data integrity.

Do not rely on external services: We do not want the solution to be based on third party services, for example a third party key distribution service.

Control over sharing: Contrary to e.g. Intel's Device Relay, we want to be able to explicitly share and unshare particular UPnP services to specific home networks.

Share with unlimited number of networks: We do not want our solution to be limited to sharing devices only between two home networks.

Flexible architecture that enables new service concepts: The architecture of our solution should not be limited to home-to-home scenario only. For example, it should be possible for a broadcasting company to use our solution to distribute specific content to the set-top boxes located at their clients' homes.

B. Overview

The starting point for our solution is that home networks are connected to the Internet via a residential gateway, such as an ADSL or cable modem/router, and that the devices to be shared support UPnP. In addition, we assume that each administrator of the home networks to be connected has a mobile phone with local networking capability (e.g. WLAN or Bluetooth).

UPnP allows devices to act as control devices or control points. Control points invoke actions at the services offered by control devices. UPnP is targeted for usage in small-scale closed networks. It uses multicasting within a home network for service discovery and, understandably, these multicast messages are not forwarded by residential gateways towards broadband networks. However, that makes it difficult to interconnect UPnP-based home networks in a controllable way.

In order to overcome this issue, some kind of forwarding of UPnP related protocol messages is needed between home networks that we want to interconnect. Such an approach requires to have a forwarder/receiver at each home network and, moreover, that forwarder/receiver needs to be such that its operations can be controlled by users in a convenient manner. Hence, we propose to add to each home network an additional device which we call *hBox*. As we will show in this paper, this box enables a home network interconnection solution that largely fulfills the requirements listed in Section II-A. There are three main steps to share services to remote home networks and to use services offered by a remote home using the hBox:

1) hBoxes residing in home networks are authenticated and interconnected. This step requires also a way to distribute initial keys which are used in the authentication. That can be done in multitude of ways, such as predistributing keys, assuming PKI, or using a kind of out-of-band channel to distribute the keys. In the example scenario that we sketch in Section II-C, we choose the latter choice and, specifically, use a mobile phone to do the key exchange via cellular network.
2) User shares a service from a home to another by authorizing the hBox in the local home to start forwarding/receiving specific UPnP traffic to/from the hBox of the other home network.
3) To use remote services, hBox at local home translates necessary UPnP related protocol messages from local control points targeted to remote control devices and forwards them to the remote hBox. Similarly, the remote hBox forwards back the resulting messages.

C. Example scenario

We describe the architecture in more detail by going through an example scenario. In this example, two homes, those of Alice and Bob, will get connected after which Alice shares to Bob a device which is subsequently accessed by a device in Bob's home network. Figure 1 illustrates this scenario. Note that our solution works with more than two homes as well.

The hBox connects to the gateway, e.g. via Ethernet cable and has WLAN and optionally Bluetooth support. The box acts as a control point that collects information about existing devices in that home network. An administrator of a home network uses his/her mobile phone to control the sharing of devices, i.e. the admin controls the hBox with the phone. The communication between them happens via WLAN or Bluetooth.

First Bob requests for a connection to Alice's Home. This request contains the IP address of Bob's hBox (DEV4) (or alternatively an address of a public rendezvous point which we explain in Section III) that should be allowed to access Alice's home. Alice generates and distributes a master key to be used between Alice's and Bob's homes and at the same time configures its hBox (DEV3) to allow access from Bob's home with the generated key. We propose that this initial phase (steps

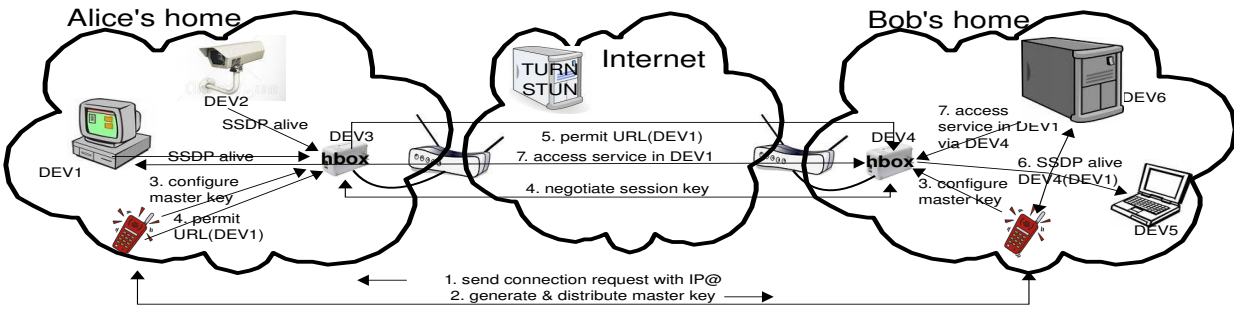

Fig. 1. Architecture of our solution.

1-3) happens directly from phone to phone via the cellular network (e.g. GSM/UMTS). The idea is that trust relationship has been established beforehand: Alice and Bob are friends and, hence, share phone numbers. The communication happens e.g. via SMS.

In the next step, hBoxes negotiate a session key which is used to encrypt traffic between the hBoxes. This session key can be a very short term one or even permanent depending on the user's preferences. Then, Alice decides to share a service offered by device DEV1. A message is sent by Alice's phone to the hBox and further forwarded to Bob's hBox. After this step, the Bob's hBox knows about the service in DEV1 and adds it as a service of an embedded device within its own device (DEV4) description. Then, that hBox multicasts a new device announcement message within Bob's home. Finally, during step 7, DEV6 accesses the shared service of DEV1 which it has learned about earlier by querying the device and service descriptions of DEV4, i.e. Bob's hBox, which that hBox has forwarded to DEV1 through Alice's hBox.

In Figure 1, step 2 describes that Alice's phone generates and distributes a master key, which implies that the master key is a symmetric key. However, it is also possible to use asymmetric key cryptography when authenticating and negotiating a symmetric session key. In that case, Alice's and Bob's phones exchange public keys during steps 1 and 2.

From the user's perspective the scheme is simple. The mobile phone runs a GUI. In the first step, Bob simply selects Alice from the phone's address book to request connection to. The phone is connected to the hBox and will take care of formulating the necessary SMS message to be sent to Alice. Alice's phone will receive the request and prompt her to either accept or reject it. If Alice accepts the request, her phone generates and sends to Bob's phone a reply SMS message that contains the shared key and address to connect. When Alice wants to share a service in DEV1 to Bob, she uses the GUI first to select that service from the list of available devices and their services in her home (automatically discovered through UPnP) and then to select Bob's home from the list of connected networks. Afterwards, accessing the service in DEV1 through the hBox is transparent to DEV6 in that it happens, from DEV6's point of view, in the same way as accessing a local device.

It should be noted that most likely both residential gateways act also as a NAT and firewall. Therefore, as depicted in the figure, it may be necessary to resort to some kind of traversal solutions such as TURN/STUN/ICE. We discuss this issue more in Section III.

III. Design Details

We go through here some of the design details concerning our solution.

Ensuring secure connectivity: The hBox needs to manage master/session keys for each remote home network. It must negotiate new session keys periodically using the master keys. The keys along with the associated IP address provide means for access control. The hBox also encrypts and decrypts traffic to and from hBoxes of connected home networks using session keys in order to preserve confidentiality, and provides data integrity protection if users prefer.

Keeping track of and advertise remote services shared to its local home network: The hBox first decrypts and parses a `permit` message sent by a remote hBox of a connected home network. Then, it queries the newly shared device's description via the remote hBox and includes this description with the list of shared services as an embedded device in its own description. Finally, it advertises this new device by multicasting a new device announcement message where it specifies itself as a proxy device type. Afterwards, other devices request for the description of the hBox and discover this embedded device and list of its services after which they can further request descriptions for the individual services of that device. These requests, as well as any future control messages to those services, are forwarded by the local hBox to the remote one which takes care that the request reaches the corresponding device and response is sent back to the requester via the local hBox. Similarly, when the hBox receives a `deny` message indicating that a specific remote service is no longer shared, it removes it from its description and re-announces itself in the local network.

Keep track of local unshared services and services shared to remote home networks: The hBox maintains a list of services found in the local home network. This list is provided to the mobile phone of the admin where he/she can select services to share. In addition, it maintains a list of local services that are already shared. When a device with shared services leaves

the local network, it sends a `ssdp:bye` message which the hBox forwards to the hBoxes of those remote networks to which this device's services were shared. They then multicast the message locally. Finally, when a user decides to share or unshare a service, the hBox crafts necessary `permit` or `deny` messages and sends them to the correct remote hBoxes.

NAT/firewall traversal: NATs create significant challenges for end-to-end communications because they allow the use of private IP addresses and provide mapping service between the private and public domains. There are several different NAT types and the first step in NAT traversal is to detect the types of NATs on the communication path. This NAT detection can be achieved using STUN and, more recently, using the Interactive Connectivity Establishment (ICE) [6] protocol. In the case of symmetric NATs, the fallback solution is to use a relay (a TURN [7] server) that provides public rendezvous point for devices in private addressing domains. In our mobile-phone-controlled solution, the SMS messages that are used to configure the hBoxes can include both the IP addresses of the domains and the IP address of the public relay (relayed-transport-address).

Mobile phone software: The mobile phone needs to run some software that enables the admin to control connecting to other home networks and select devices to be shared/unshared to/from specific networks. This control can be implemented, for example, as a custom protocol over Bluetooth (as we did, see Section IV) or using a web interface with HTTP protocol over WLAN. When sharing or unsharing a device, the software communicates the corresponding control messages to the hBox which then implements necessary actions as described in previous sections. In addition, when a user wants to request connecting to another network, the software needs to craft the specific SMS message and sends it to the admin of the other network. Similarly, when the phone receives a connect request by SMS, it parses it, prompts the user, sends required control messages to the hBox, and generates and sends the reply SMS message.

IV. Prototype

We have designed and implemented a prototype of the hBox as a proof of concept. Our prototype is an early version and does not implement the complete design as described in previous sections. Specifically, connecting the networks via SMS exchange (steps 1-3 in Figure 1) is not yet implemented. In our prototype, the keys are currently prestored in the hBox.

A. Prototype Components and Interaction

There is a GUI software running in the mobile phone. Figure 2 illustrates how the mobile phone communicates with the hBox. We chose to use Bluetooth in this prototype. The control happens between the mobile phone's GUI software and hBox's UPnP control software. However, in between these two components there is a BT proxy daemon which essentially relays traffic between the two main components. The GUI communicates with the BT proxy daemon via Bluetooth over a RFCOMM connection. The daemon and the UPnP control are connected to each other through a local TCP connection. Message exchange happens using a very simple text-based protocol. The GUI sends control commands to the daemon which forwards the command to the UPnP control. The control component replies whether the command was successfully executed and if so, it also includes in the reply the set of return values if any (e.g. a local UPnP device list). This reply is again forwarded by the daemon to the GUI.

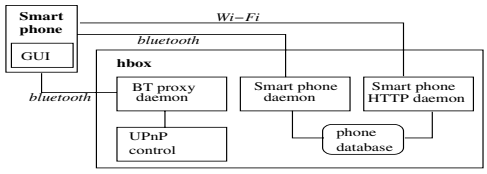

Fig. 2. Interaction between the mobile phone and the hBox.

Figure 3 shows a message sequence diagram between the components in the same scenario illustrated in Figure 1. Note that right after Bob's hBox receives a permit from a remote hBox, it requests for the description of the shared device. Once it gets the description, the hBox adds this description as an embedded device into its own description and re-announces its presence. Afterwards, control points of Bob's home, i.e. devices that wish to use the services of other UPnP devices' services, use the shared device's service as if it was a local one while the hBox takes care of tunneling requests to Alice's home.

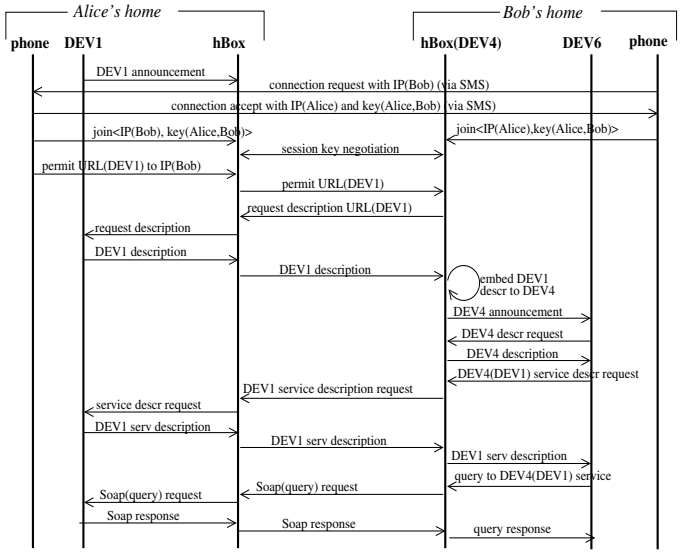

Fig. 3. Sequence of messages when sharing a device to another home network.

B. Deployment

One of our requirements for the interconnection architecture was to make the deployment as easy as possible. Our solution approaches this requirement in two ways. First, the hBox is meant to be ready-to-use component which the user could

purchase off the shelf and simply plug it into the home network gateway via Ethernet. This means that modifications are required neither to the user's residential gateway (e.g. ADSL box) nor to the home network's UPnP devices, their services, and the UPnP protocol itself.

The second way we ease the deployment of the prototype has to do with the way the GUI software is installed into the mobile phone of the home network admin. In Figure 2, there are two additional daemons running in the hBox that we have not mentioned so far: mobile phone and mobile phone HTTP daemons. The mobile phone daemon scans via Bluetooth for any or specific type of mobile phone devices. After finding one, the daemon automatically sends the GUI software to the phone. It also stores the phone description into a local device database so that the software will not be sent again to the same phone. The HTTP daemon allows manually controlling the device database via HTTP, i.e. one can add new or remove existing devices from the database, to enable and turn off the scanning.

C. Implementation

We implemented hBox on Infineon ADM5120 based Edimax BR 6104KP which is high performance and flexible System-on-Chip. The hBox implementation was embedded into this device by plugging in a USB flash disk where we store the hBox components. We selected Squidge as the operating system for the controller since that is able to run from a USB storage device. However, the gateway controller that we used comes by default with a loaded Midge image and, thus, needs to be 'reflashed' with appropriate firmware in order for it to support Squidge. A Bluetooth dongle was inserted into the second USB slot. The end result is shown in Figure 4.

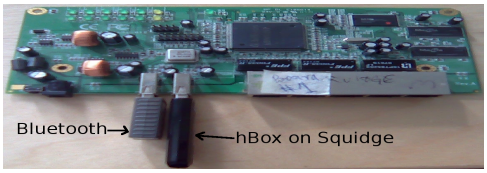

Fig. 4. The hBox prototype.

We set up two prototype hBoxes in a configuration as in Figure 1 but with the difference that the gateways were directly connected to each other. Thus, the difference is that in this experiment, we did not need to rely on NAT traversal techniques such as TURN, which we would need to do if the hBoxes were plugged into e.g. ADSL modem/routers. We formed two private networks (Alice's and Bob's homes) behind the gateways and ran a UPnP clock device in DEV1 in Alice's home. At a certain point of time this clock was successfully shared to Bob and subsequently accessed by DEV6, as in the figure.

We are currently working on a more complete prototype in which we use Host Identity Protocol (HIP) [8] for secure communication between the hBoxes. HIP also enables NAT traversal.

V. Conclusion

In this paper, we presented a novel solution for interconnecting home networks' services. Our solution is easy to deploy since the only thing the user needs to do is to to plug the hBox into his/her gateway. Usage is simple with the mobile phone and at the same time the user has full control over which services are shared and to who. Access rights and authentication relies on pre-established trust relationship and do not require third party services. In some cases, NAT traversal may require access to a public TURN server. Users can share their home services to any number of other homes and the architecture is flexible enough to allow also other than home-to-home scenarios, such as from broadcasting company to home clients.

We are not aware of other competing solutions that can achieve our requirement list presented in the paper. The value of our work lies also in providing a platform that enables development of new types of services for home networks while supporting legacy services by using standard protocols. We envision two commercial deployment scenarios: 1) hBox is manufactured and sold by third party at e.g. local Wallmart, 2) hBox is a value-added service offered by the ISP that connects a specific home networks to the Internet. We consider the latter one more likely in which case the ISP would also take care of after-sales service.

We are planning to extend our solution to cover also the cases where a user visiting another home could utilize that home's hBox to access services of another home to which the visiting user has authorized access. In addition to introducing HIP in the prototype, we are extending it to include also the SMS authorization/authentication phase. Once we have a more complete prototype, we plan to perform thorough performance evaluations in order to quantify the impact of the hBox for different kinds of services, especially such that are more sensitive to delay and jitter like A/V streaming.

References

[1] U. Plug and P. U. Forum, "UPnP device architecture version 1.1," http://www.upnp.org/specs/arch/UPnP-arch-DeviceArchitecture-v1.1.pdf, Oct. 2008.

[2] P. Belimpasakis, S. Moloney, V. Stirbu, and J. Costa-Requena, "Home media atomizer: remote sharing of home content - without semi-trusted proxies," *Consumer Electronics, IEEE Transactions on*, vol. 54, no. 3, pp. 1114–1122, August 2008.

[3] R. Chowdhury, A. Arjona, J. Lindqvist, and A. Ylä-Jääski, "Interconnecting multiple home networks services," *Telecommunications, 2008. ICT 2008. International Conference on*, pp. 1–7, June 2008.

[4] "Intel tools for UPnP technology forum," http://software.intel.com/en-us/forums/intel-tools-for-upnp-technology/.

[5] H. Y. Lee and J. W. Kim, "An approach for content sharing among upnp devices in different home networks," *Consumer Electronics, IEEE Transactions on*, vol. 53, no. 4, pp. 1419–1426, Nov. 2007.

[6] J. Rosenberg, "Interactive connectivity establishment (ICE): A protocol for network address translator (NAT) traversal for offer/answer protocols," (work in progress), Oct. 2007.

[7] J. Rosenberg, R. Mahy, and P. Matthews, "Traversal using relays around NAT (TURN): Relay extensions to session traversal utilities for NAT (STUN)," (work in progress), Feb. 2009.

[8] A. Gurtov, M. Komu, and R. Moskowitz, "Host identity protocol," *The Internet Protocol Journal*, vol. 12, no. 1, 2009.