ICEbox: Bring! Point! Get Configured!

Jeonghwa Yang and W. Keith Edwards
Graphics, Visualization and Usability Center
College of Computing
Georgia Institute of Technology
{jeonghwa, keith}@cc.gatech.edu

ABSTRACT
Complex network configuration for digital devices in the home is hard work for most home users, who are not familiar with network technologies. In this paper, we present our effort through a system called ICEbox to reduce the complexity and increase the usability of configuring such devices. The ICEbox, a special node on the home network, allows users to set up devices only with a simple pointing gesture, without worrying about complicated network configuration issues. All that a user needs to do for network setup is to physically bring the device to the ICEbox and point it toward the ICEbox. The ICEbox then takes care of configuring the device, and can also play a role in ongoing monitoring and reconfiguration of the overall network.

INTRODUCTION
Home broadband adoption is growing and fueling an increase in home networking. However, despite obvious interest on the part of home users, concerns about the complexity of home networking hinder its adoption. Earlier work [2] identified some of the sources of complexity, through ethnographic study of home network early adopters. One of the primary sources of complexity concerns correctly *provisioning* devices for the home network. This task involves configuring or adapting devices to the particular circumstances and context of a specific home network. In addition to initial setup, another particularly troubling aspect of provisioning is that it is inherently fragile - any change to the home network topology, for instance, installation of a second access point or change of Internet Service Provider, has the potential to break the ability of existing devices on the network to communicate with one another, or with the broader Internet.

Currently, most network provisioning is done "by hand," typically by a home user with networking knowledge. This user must understand the topology of the home network, and be able to translate his or her mental knowledge of this topology into settings appropriate for each individual device. Although a number of technologies (DHCP, ZeroConf, and discovery protocols) can help with certain aspects of provisioning, they clearly do not solve the problem.

Fundamentally, the key problem here is that users must be deeply familiar with not only the infrastucture concepts of networking in *general*, but also the configuration details of his or her *specific* home network, in order to effectively manage the network and new client devices. Effective configuration and maintenance of the network and the clients on it requires that users be exposed to the technical concepts and terminology of the underlying network. We believe that solutions to IT@Home must not only reduce this burden significantly, but must do so without extensive changes to the existing (and deeply entrenched) networking protocols common in the home.

Ideally, a better approach to provisioning would strip away as much manual configuration work as possible, leaving only the minimum required actions that must be undertaken by a user. There will always be some work that can only be done by a user because our systems cannot know - without human intervention - what devices should be on any given home network. Our goal is to move toward a model in which *only* this necessary and explicit step of introducing a new device to the network needs to occur, and the technical details of infrastructure configuration are implicitly derived from that.

In this paper, we present our efforts at creating such a system, which relies on a simple gestural interface that hides complex network configuration parameters. Our prototype system, called *ICEbox* (which stands for Installation, Configuration, and Evolution), is a special, privileged node on the home network, tasked with issuing network configurations to new devices and to existing devices when the network topology changes. The provisioning work between the ICEbox and a device is done over a location-limited channel (infrared in our current implementation). The user interface to the ICEbox is quite simple; all that a user needs to do for configuring a device is to bring it to the ICEbox and point it toward the ICEbox. The ICEbox then detects the device and provides it with provisioning information in response to this introduction by the user. This provisioning step can configure not only network parameters suitable for the user's specific home network, but also higher-level application defaults (such as the home's default printer, or file shares that exist in the home). Once this initial configuration information is passed to the client, the client can be installed in the home network

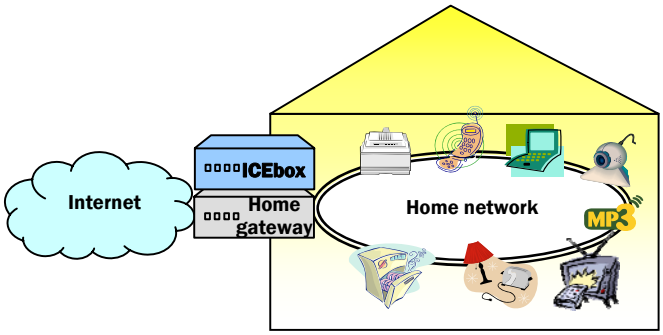


Figure 1. The networked home and ICEbox

and communicate with the ICEbox using standard IP-based protocols. These protocols can allow the ICEbox to monitor home network status, debug problems, and support easier evolution of the underlying network infrastructure.

In the rest of the paper, we describe the details of the ICEbox and our prototype implementation.

ICEbox Vision

The ICEbox is logically located at the boundary of the home network, allowing automatic device configuration for devices entering the home. Figure 1 shows the ICEbox in the networked home.

The configuration process between the ICEbox and a newly joining device is done over infrared. A short-range communication technology such as infrared provides a number of important advantages. First, it allows two devices to communicate in an ad-hoc, secure manner with no pre-configuration required. Thus, it serves as an ideal "bootstrapping" mechanism, since it can operate without explicit human involvement in configuring it. Second, its range of 1 meter provides an implicit gesture recognition boundary, making it amenable to physical interactions (such as pointing), and ensuring that access to the home network is bounded by access to the physical structure of the home itself.

The ICEbox provides a gesture-based interface to a user who wants to attach a new device to the home network. When a user wants to add a new device to the home network, he or she brings the device to the ICEbox and simply points it toward the ICEbox. The device transmits its unique ID, along with device type information, to the ICEbox. The ICEbox then detects the new device and provides it with a set of configurations valid on the home network. The configurations contain an address (e.g. IP address for IP-based network), network layer settings (e.g. network mask, default gateway, and DNS servers for IP-based network), data-link layer settings (e.g. wireless network name (SSID), security mode, channel, and so forth for 802.11), and service-level settings (e.g. default printer and firewall port configurations). The device then configures itself based on the received configurations.

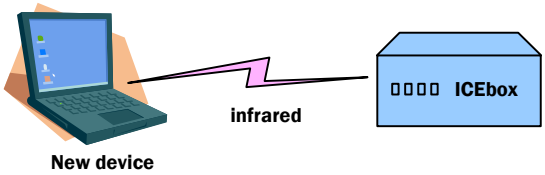


Figure 2. The ICEbox sends configurations to a new device over infrared.

Once a client is deployed in the home, an IP-based protocol is used for communication between it and the ICEbox. A client, when it boots, selects a configuration from the set provided to it during the bootstrap phase, and then contacts the ICEbox to tell it where it has found itself topologically in the home network. After this point, a simple protocol passes heartbeat information between the ICEbox and the client, as well as results of local discovery. The ICEbox aggregates this information to determine the layout of the home network as well as detecting any devices that have disappeared from the network.

The ICEbox notifies other devices of the topology changes to the home network or propagates configuration changes onto the home network to existing devices if necessary. For example, introducing a new printer onto the network might cause the ICEbox to tell existing nodes to change their default printer. More fundamental network changes such as changes of network keys and gateway IP addresses can be propagated down to network nodes as alternate configurations before the changes take effect, allowing seamless continuity of connectivity after the change is made.

In addition to supporting initial configuration and evolution, having such a privileged node on the network, with knowledge of all introduced clients, allows the ICEbox to support a number of unique features. For example, the ICEbox can provide optional authentication, to provide a greater degree of security to prevent unauthorized device or users from joining the home network. Authentication involves an extra step in the initial introduction phase, during which the client that is physically pointed at the network must provide some demonstration of trust (typically in the form of a password for the home network) before it is allowed to join the network. This mechanism can provide greater security than merely restricting physical access to the ICEbox.

Second, the ICEbox can provide a mechanism for device and service discovery. The ICEbox, since it maintains a list of all devices that currently exist on the home network, can serve as a central repository of those devices, allowing applications to find out about the presence and capabilities of them. Essentially, it can provide a self-populating directory service, mitigating many of the problems seen with multicast-based discovery protocols, which are limited to working only on the local link.

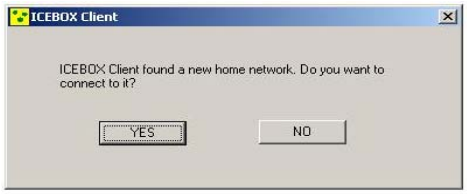
Third, the ICEbox can be used as a point of maintenance for the home network. The heartbeat monitoring protocol used between clients and the ICEbox allows it to detect failures of devices, and then use its aggregate knowledge of the home network to suggest troubleshooting strategies.

Finally, the ICEbox can be used as a proxy lying between the home network and the Internet or between non-IP devices and IP devices.

PROTOTYPE IMPLEMENTATION

We use a stand-alone desktop PC as our prototype ICEbox platform, although we envision that the ICEbox functionality will ultimately be incorporated into a home gateway router. Currently, we have implemented the short-range communication mechanism used for introduction, optional authentication, and the mechanisms for initial configuration of network parameters. We are currently in the process of implementing the post-deployment protocol and the provision of application-layer parameters, as well as the directory service.

The ICEbox runs a copy of the ICEbox server software. When the ICEbox is plugged in, it initializes itself, synchronizing network settings including IP network and wireless network settings such as IP address ranges for devices, subnet mask, default gateways, DNS servers, SSID, and channel with the home gateway router. Then, it listens for device attachment requests over infrared.



Step 1: click **YES**.

Step 2: Enter the password of the home network and click **OK**.

Step 3: Configuration is complete. Click **OK**.

Figure 3. The network configuration process with the ICEbox

Assume a user wants to add a laptop to the home network. The laptop has built-in infrared and an IEEE802.11 card inside. The laptop also runs an ICEbox client application that will communicate with the ICEbox server.

When the user points the laptop to the ICEbox in line-of-sight of infrared, the ICEbox client software on the laptop detects the ICEbox and asks the user if he or she wants to connect to the home network. If the user clicks "YES," the client sends an attachment request to the ICEbox over infrared with the device's MAC address. The ICEbox then provides the client with network and wireless settings. Then, the ICEbox client application sets up the laptop with the received settings and sends the information about the device and services that the laptop is associated with to the ICEbox server which in turn saves these to its central device repository. Finally after all these processes have finished, the ICEbox client informs the user that the laptop has been successfully configured for the home network.

Our current implementation also allows the home network to be optionally "locked" by a home owner. If authentication is enabled, the user has to pass an authentication step before being attached. Step 2 in Figure 3 shows our current authentication interface, which uses a password scheme. As shown in Figure 3, all that the user needs to do for his laptop configuration is two or (in case of authentication check) three clicks. This mechanism allows home users to optionally provide an additional layer of security on top of that provided by restricting physical access to the ICEbox.

RELATED WORK

Zeroconf [5] is an IETF specification to enable devices on the IP network to automatically configure themselves and be discovered without manual intervention. The self-assigned link-local addressing scheme of Zeroconf provides dependable IP networking in the absence of configuration information and infrastructure. Zeroconf, however, only concerns the IP network layer, which only addresses part of the problem: Zeroconf does not support lower-layer configuration details or higher-layer application defaults. It also does not provide the management and update functionality of the ICEbox..

PARC's Network-in-a-box (NiaB) system [2] is the most closely related to ours. NiaB allows users to add laptops to a secure wireless network by walking up to an access point and physically pointing a laptop at the access point. The functionality of NiaB, however, is restricted only to secure wireless configuration. The goal of the ICEbox is to deal with other aspects of network configuration, as well as higher-level service and application configuration, and ongoing evolution of the network.

While NiaB uses a short-range communication mechanism similar to ours, the use of a gesture-based interface to device identification was introduced in the gesturePen [1]. This allows users to select devices through a pointing

gesture using custom tags and a custom stylus called the gesturePen, instead of navigating through traditional user interface widgets such as lists.

A commercial technology that is related to ours is Windows Connect Now (WCN) [4], which provides alternative mechanisms for automatic wireless configuration for home wireless network users. In WCN, a user transports wireless network settings from a PC to to a networked device in the home using a USB flash drive. Then, the transported settings will be automatically configured in the networked device by the WCN service. This system is more manual, and less dynamic, than the one provided by ICEbox, but does not require the presence of hardware for two-way communication.

STATUS AND FUTURE DIRECTIONS

We have completed our initial implementation of the ICEbox introduction protocol, using infrared as our short-range communication mechanism. This implementation provides the initial configuration step for network- and link-layer parameters. As noted earlier, this prototype implementation exists on a desktop PC, for ease of development purposes. The ICEbox client software has been implemented for Windows XP. Both the clients and server have been implemented in C++. Although we used a laptop running Windows XP as a client device in our current prototype, we intend to support various types of devices, platforms, and operating systems in the future, which will be one of implementation challenges to us.

Our next phase of implementation is to complete our IP-based monitoring and update protocol, which should allow better management of the home network, as well as evolution when new infrastructure devices are added.

We plan to begin user evaluations of our technology in the near future. Additionally, one goal of ours is to see how technologies such as ICEbox can better support the highly dynamic home networks we envision for the future. For example, the current ICEbox model is based on introduction of devices that will be long-term residents on the home network. We wish to explore models of how to better support devices that enter the network transiently (for example, allowing a visitor's laptop to access the network for a short period). Such new functionality must be based on study of home networking practices in the real world.

CONCLUSION

Our earlier empirical work has demonstrated how configuration and evolution of the home network are inherently difficult for end-users to accomplish. Further, these tasks are inherently fragile, as seemingly small changes to the network topology can cause users massive frustrations, often requiring that they undertake a reconfiguration of all client devices after such a change.

We believe that the key to simplifying this process is to make the setting of technical details *implicit* in the actions of users that cannot be inferred automatically by the computer. At a minimum, the indication that a particular device should in fact be on the users' home network is one such action.

The ICEbox project is an exploration of how far this strategy can be taken. Our initial implementation allows home users to set up devices in the home with a simple pointing gesture, without worrying about complicated network configuration issues. We aim to continue oure explorations, by adding additional functionality for network monitoring, troubleshooting, and application-level configuration. We believe that approaches such as this can improve the user experience in the home network by facilitating a significantly easier and more natural way of network configuration, administration, and repair.

REFERENCES

1. Balfanz, D., Durfee, G., Grinter, R.E., Smetters, D.K. and P. Stewart, Network-in-a-Box: How to Set Up a Secure Wireless Network in Under a Minute, Proceedings of the USENIX Security Symposium, 2004
2. Gringer, R.E. and Edwards, W.K. The Work to Make a Home Network Work, Proceedings of the Ninth European Conference on Computer-Supported Cooperative Work (ECSCW'05), September, 2005.
3. Swindells, C., Inkpen, K.M., Dill, J.C., and Tory, M. That one there! Pointing to establish device identity, Proceedings of the ACM Symposium on User Interface Software and Technology (UIST 2002), pp. 151 – 160, 2002.
4. http://www.microsoft.com/windowsxp/using/networking/getstarted/windowsconnectnow.mspx
5. http://www.zeroconf.org/