MultiNet: Reducing Interaction Overhead in Domestic Wireless Networks

Anthony Brown Richard Mortier Tom Rodden

School of Computer Science, University of Nottingham, UK
firstname.lastname@nottingham.ac.uk

ABSTRACT

We present *MultiNet*, a novel method for securely associating devices with a domestic wireless network. We show that MultiNet has usability benefits over currently deployed commercial solutions while being backwards compatible with existing devices. MultiNet reduces the interaction overhead of secure association by focusing on users' interactions rather than the network's requirements. This leads to a novel architectural arrangement of the home network infrastructure: the network is dynamically re-configured to accept each pre-configured device, rather than the current norm where each device is configured to be acceptable to the pre-configured network. Assuming devices are pre-configured for a unique, device-specific network name and passphrase, MultiNet constructs an out-of-band visual channel via an intermediary *network controller* device to convey the device's configuration to the network. This makes the interaction to join a device to the wireless network lightweight and identical across all devices, considerably reducing the interaction overheads for users.

Author Keywords
Usable security; domestic environments; 802.11

ACM Classification Keywords
C.2.1 Network Architecture and Design: Wireless communication

General Terms
Human Factors; Design; Measurement.

INTRODUCTION

Home networks are now commonplace in the developed world. Households typically make a broadband connection accessible throughout the dwelling by providing a home 802.11 wireless (Wi-Fi) network via *access point (AP)* functionality built into the home router. Wireless networks are secured by associating devices[1] with the network using the 802.11i standard. Commonly called WPA2, this authenticates devices and the network using the Extensible Authentication Protocol (EAP) [1], and provides a secure channel between each device and the AP.

These protocols have their roots in corporate/enterprise environments which are quite different from home wireless networks. This is reflected in their design goals (e.g., scalability to many nodes) and assumptions about users (e.g., skilled network and systems administrators configuring edge nodes) that are not directly appropriate for the home network. Home networks are typically small in size, supporting between 5 and 20 devices, with network elements physically accessible. Home network infrastructure is predominantly self-managed by residents who are not typically expert in networking technology and have no motivation to become expert. The heterogeneity of devices connecting to the network is also startling: a single household might have PCs, games consoles, phones, printers, cameras, televisions and media players.

Inherent in traditional Internet Protocols (IP) is an "end-to-end approach" where the network core is simple and stable, and it is presumed that clients will be configured to fit the network. The diversity of network-connected devices in today's homes makes this presumption problematic: these devices vary widely in complexity, capabilities and interaction styles [7]. Each device offers quite idiosyncratic means of configuring network properties to allow them to join the network. Recent trends towards Wi-Fi enabled domestic devices, such as sensors, web cams and bathroom scales which have limited interaction capabilities further exacerbates the problem. Consider, for example the challenge involved adding a devices such as a bathroom scale or printer that simply do not provide either a keyboard or screen to your network. The net result is that many users experience considerable difficulty configuring, managing and expanding their home networks [14, 15, 19]. In the case of wireless networks users often relax the network's security to reduce the management burden [8]. For example, a warbiking exercise by Sophos in September 2012 found 27% of London Wi-Fi networks remained unsecured.[2]

The challenge we address is twofold. First, how to enable the construction of secure wireless networks while reducing the interaction overhead involved in adding devices, secondly how to create a system that enables the simple revocation of access to devices. The key contribution of this paper is the development of a means of association and revocation to/from wireless networks that is driven by users' interactional needs; that is consistent across all devices and independent of the de-

[1] For readability, we refer to *all* things that connect to the network as *devices*, rather than using context-specific station or endsystem.

[2] http://www.sophos.com/en-us/press-office/press-releases/2012/09/sophos-reveals-wifi-security-issues.aspx

CHI 2013, April 27–May 2, 2013, Paris, France.

vices interaction capabilities and does not require any modification to client software. Traditional approaches have the user undertake the burden of configuring devices to fit the network and have no easy means of revocation. In contrast, MultiNet reconfigures the network to meet the needs of devices. We exploit the physical arrangement of domestic natures to allow details about the clients current configuration to be provided to the network via a proximal out-of-band channel (currently implemented using QR codes). This arrangement allows devices to be securely introduced to and removed from network through an interaction that is lightweight, consistent, and device-agnostic.

Before presenting the design and implementation details of MultiNet we briefly reflect on the interactional overheads imposed by currently deployed approaches to wireless association. We then present the motivation, design and implementation details of MultiNet, followed by an assessment of its technical performance. Finally, we present and discuss results of a lab-based user study directly comparing MultiNet to Wi-Fi Protected Setup (WPS). Our results show that MultiNet is technically feasible and offers significant usability benefits, making it considerably easier for users to manage and maintain their home wireless networks by reducing interaction overhead across different devices.

All the code we describe is publicly available under open-source licenses at http://multinet-80211.github.com/.

THE INTERACTIONAL OVERHEAD OF ASSOCIATION

The complexity involved in current approaches for securely associating devices is widely criticised for the burden it places on users. The joining problem presumes two devices: the access point running within the home router, and the device being associated (PC, phone, etc). Typically, neither is well suited to having users perform complex tasks: the router is a black box whose only interface is a set of web pages, often oriented toward a network engineer;[3] and although traditional devices such as PCs and smartphones permit complex user interaction, this is increasingly not the case with devices such as games consoles, TVs, fridges, bathroom scales and so on. A few moments trying to join a printer or set of scales to one's home network is sufficient to understand this!

This increasingly diverse range of wireless devices simply cannot provide consistent, straightforward means to enter the security credentials (SSID, passphrase) typically required of a WPA2 secured wireless network. The widely used "Select Network and Enter Passphrase" interaction commonly used by devices with screens and text input capability suffers from well-known problems with passphrase recall and input error [17]. The industry attempted to address these problems through the introduction of new usability-focused association methods, notably WPS with Push Button Configuration (PBC). However, even though the protocol focuses on usability [25] it still suffers in the diverse device ecosystem of the home: the position and appearance of the push button varies considerably between devices due to aesthetic, design and form-factor constraints. Some devices also opt for a "soft" button, often hidden within the setting and configuration menus, further complicating the situation.

These factors combine to impose significant interaction complexity on the user, with each device requiring new knowledge to be acquired before it can be joined to the network. We reduce that complexity by providing an approach to building a secure network that is both:

Interactionally light, requiring minimal interaction complexity; and
Interactionally consistent, across all devices irrespective of the interaction capabilities of the device being added.

We achieve this by amending the infrastructure so that the user can cause it to configure to fit the device's existing settings, rather than having to alter each device's settings to fit the network. We also allow the network to play a more active role in the interaction involved in device association by providing a dedicated network controller. When the user attempts to associate a device with the network using our network controller, our infrastructure creates an **on-demand virtual access point** matching the existing configuration of that device. This enables us to provide consistent interaction across all devices without loss of functionality. Before we introduce the details of MultiNet, we briefly review current approaches to Device Association and Pairing.

CURRENT APPROACHES TO DEVICE ASSOCIATION

There are several currently deployed methods for securely associating a Wi-Fi device to a Wi-Fi network. The most common is *manual passphrase entry*, supported by almost all devices as a fallback when other methods fail. The user must have a secret that is also known to the AP: the combination of an alphanumeric passphrase (formally, the Pairwise Master Key) and the network name (formally, the Service Set Identifier, SSID) in the case of WPA2. This information permits the user to add the device to the network. The SSID is usually broadcast in the clear enabling the user to select it from a list on most devices. The network's passphrase is usually either pre-configured and printed on the bottom of the AP, or chosen by the user when they initially configure the AP. Issues of memorability, passphrase confusion, incorrect recall and input error mean that users find passphrases difficult [17]. This affects the usability of manual entry methods and is exacerbated by the different input methods afforded by different devices. Indeed, the multiple step, many device, acronym-filled setup experience involved in configuring a secure wireless network has been suggested as the root cause of many inexperienced users leaving their networks partially or completely unsecured [8].

To address these usability problems, the Wi-Fi Alliance created WPS. This offers three methods of configuration: *in-band*, which requires users to initiate the pairing process by manually entering a 4 or 8 digit PIN into the device being joined; *out-of-band* (OOB),[4] which allows the user to pass credentials between the device and the AP using RFID or

[3] Although a new generation of routers are emerging "app enabled", e.g., http://home.cisco.com/, these are technology specific and cannot reduce the burden of needing to configure the joining device.

[4] Not yet widely implemented in consumer hardware.

NFC tokens; and *push-button configuration (PBC)*, which requires the user to condition the device by selecting the desired network from a list and pushing a button on the AP, causing it to listen for a connection attempt for 120 seconds before timing out.

Unfortunately there are considerable problems with WPS. Current implementations of in-band WPS are vulnerable to brute force attacks on the PIN.[5] Out-of-band methods are not widely deployed. Push button methods rely on physical access to both the device and the AP which can pose problems if either the device is not very mobile (e.g., a TV or refrigerator) or, as is increasingly common, the AP is hidden away and not easily accessible. Finally, Kou *et al* [12] suggest the number of available paring methods and the lack of consistency of implementation across devices reduces user experience when using WPS. We next review device pairing mechanisms presented in the academic literature.

DEVICE PAIRING APPROACHES

MultiNet provides a lightweight, consistent interaction to securely associate a device with a wireless network. This problem – securely associating a device with a home network – is a particular case of the general problem of securely pairing two devices, which has been extensively studied by the usable security (HCIsec) community. It was first addressed by the "Resurrecting Duckling" protocol [21] which suggests that devices should be connected by a physical connection such as a cable for the pairing process to occur. "Talking to strangers" [3] takes a similar approach but uses an infrared connection as the OOB channel to transfer credentials between devices. Both need little user involvement but do require that the devices have a compatible hardware interface, often not the case in today's complex device ecosystem.

Numerous other projects have tried to address this problem through the use of OOB channels and user involvement. "Seeing is believing" (SiB) [13] introduces visual OOB channels for mutual authentication of two camera-equipped wireless devices, using 2D barcodes to explore the general problem of bootstrapping encrypted communication between mobile devices and access points. Blinking Lights [18] simplifies this to requiring only a unidirectional visual OOB channel to mutually authenticate devices. A single flashing LED on one device and the camera on the second device enable transfer of the short authentication code.

"Network-in-a-Box: How to Set Up a Secure Wireless Network in Under a Minute" [2] uses location-limited channels to perform the OOB key exchange in an attempt to address the usability issues of the pairing problem. These location-limited channels usually have lower bandwidth and higher latency than normal network communication mediums. Their implementation uses short-range infra-red to provide an intuitive point-to-authenticate gesture for OOB public key exchange, bootstrapping the wireless joining process. Similar approaches using different OOB channels have been suggested, e.g., "Loud and clear" [6] and HAPADEP [20] both use auditory channels for credential exchange. For a detailed review of other methods see [10, 11].

Each of these approaches reduces the burden on the user but places considerable constraints on the hardware and/or software capabilities available on the AP and, worse, on the joining devices. For example, SiB and Blinking Lights require all devices to have a camera to read the network credentials, Network-in-a-Box requires the devices and AP have an IR sender and receiver respectively. Such requirements are often impractical for low-cost single function devices like sensors and media streamers, due to the increased manufacturing costs. They can also rarely be met by the devices already deployed in the home, making backwards compatibility difficult to achieve.

Instead, MultiNet abstracts the hardware requirements of the OOB channel into a third device, the network controller, thus placing no constraints on the hardware of the associating device. Using a dedicated device to control and configure home networks, has previously been shown to be beneficial to users. For example, ICEbox [26] is a network appliance that acts as a centralised point of control for managing the home network providing a interface for users to interact with its configuration. ICEbox supports device association through a number of traditional pairing methods.

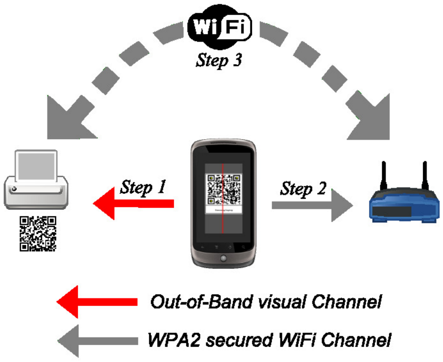
However there is a more fundamental difference between MultiNet and the various methods described above. All these methods *configure the device* by conveying information *from the AP*: via QR Codes (SiB), via a flashing LED (Blinking Lights), via auditory channels (Loud and Clear, HAPADEP), via gestures (Network-in-a-Box). In contrast, MultiNet *configures the AP* by *conveying information from the device*. This provides security and usability benefits which we discuss following exposition of the detailed design and implementation of MultiNet.

MULTINET DESIGN

MultiNet aims to reduce the interactional overhead of associating devices with domestic wireless networks by reconfiguring the network infrastructure to fit the device being associated. We modify the AP software running on the home router to enable it to create on-demand *new* virtual access points that use specified WPA2 credentials. Further minor extensions to AP software running on the home router enables use of a third intermediary device, the *network controller*, to create a OOB communication channel with the AP in the home router. This channel is implemented as a WPA2-secured administrative network between just the network controller and the AP. This arrangement enables a consistent configuration interaction to be used across all devices without imposing new hardware or software constraints, maintaining backwards compatibility with legacy equipment.

The current implementation uses a smartphone as the network controller, with its camera providing the visual OOB channel by reading the required credential specification from a QR Code affixed to the exterior of the associating device. This approach ensures backwards compatibility with existing devices: they require *no* software or hardware modification but

[5] S. Viehböck, US-CERT Vulnerability Note VU#723755 – Wi-Fi Protected Setup (WPS) PIN brute force vulnerability, http://www.kb.cert.org/vuls/id/723755.



Step One	Use the camera on the network controller to read the QR Code on the device to be joined to the network.
Step Two	Request that the device is joined to the network by sending the SSID/passphrase pair for the device's virtual network to the AP.
Step Three	Establish the virtual network allowing the pre-configured device to join the network.

Figure 1: MultiNet interactions to add a device.

must simply be pre-configured with an SSID and passphrase which are affixed on its surface as a QR Code. These credentials are securely conveyed to the AP by the network controller. Upon receipt, the AP creates a virtual access point with the specified credentials pair with which the device automatically associates. The overall protocol flow is depicted in Figure 1. The result is that the AP offers multiple networks, roughly one per device. MultiNet poses a number of deployment questions to which we now discuss each in turn.

Credential Encoding

As previously stated, we encode credentials in QR Codes, making a simple scan of the QR Code MultiNet's basic interaction mechanism. QR Codes were chosen for their ease of machine readability, simple point and capture interaction method, relative familiarity, and their ability to encode sufficient information: we must potentially encode a unique credential (SSID and passphrase) for every device manufactured. The maximum length SSID is 25 bytes, and the maximum length WPA2 passphrase is 75 bytes, allowing ample credentials in combination. Figure 2 shows a MultiNet QR Code affixed to a device.

Device Pre-Configuration

MultiNet assumes devices are pre-configured with their SSID and passphrase; how to achieved this, is an important question. Ideally, all devices would be pre-configured by the manufacturer but this unlikely in the immediate future. However, this is not as serious a problem as might be thought as MultiNet enables this process to be outsourced. Outsourcing, where others (e.g, relatives or professionals) are asked to undertake the more technical aspects of management, is a widely used approach to domestic network management approach [16]. One could envisage outsourcing similarly taking care of device pre-configuration, resulting in a printed stick-on QR Code being affixed to the device. Negating the need for localised configuration and problems inherent in remote support and debugging that arise [7, 23].

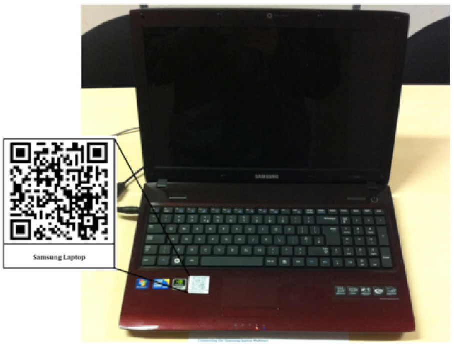


Figure 2: Laptop with MultiNet QR Code encoded credentials.

Bootstrapping the Network Controller

The first step in using MultiNet is to securely associate the network controller with the administrative network, enabling it to establish a trusted signalling channel to the router over which other devices can be introduced. This only needs to be done once per network controller, and we envisage a range of ways to achieve this. For example, the router manufacturer might bundle a dedicated QR Code scanner pre-installed with the MultiNet software and pre-configured to use the administration network. Alternatively, any WPA2 compliant device able to install applications that possesses a camera can become a network controller via installation of appropriate software. As most modern smartphones and tablets fit these criteria they provide a convenient and incrementally deployable solution on which we focus for the rest of this paper.

The process of setting up the network controller is straightforward. The AP has two QR Codes affixed to its surface: one contains the install location of the network controller application; the other contains the credentials for the the secure administrative network. Pointing a standard QR Code reader application running on the putative controller at the first QR Code causes the user to be taken to a website where the controller application can be installed as with any other application. This application is then run and the device's camera pointed at the second (normally hidden) QR Code containing the SSID and passphrase for the pre-configured administrative network on the router. This securely associates the network controller with the AP, providing a trusted signalling channel over which new devices can be subsequently associated.

Adding Devices

Associating a new device to the network is straightforward. In our prototype we manually configure each device with its own individual SSID/passphrase, which are encoded as a QR Code, printed and affixed to the device. As noted above, in a full-scale deployment we envisage that device configuration would occur at time of manufacture, at point-of-sale, or via a third-party service; and the QR Code would be affixed in a device-appropriate manner.

The configuration application is run on the network controller, which is pointed at the QR Code on the device. The network controller then communicates with the router to configure the virtual AP specified by the credentials encoded in the QR Code (Figure 1). This creates the appropriate network with the specified credentials, at which point the network controller displays a success indication to the user, and the device associates with the network as soon as networking is enabled on it.

Compatibility With Existing Devices

Backwards compatibility for legacy devices can be achieved by the home owner using the MultiNet network controller to generate a QR Code with a unique SSID and passphrase. The legacy device can be manually configured to this SSID and passphrase, and the generated QR Code printed out and affixed to it. Once this initial configuration is complete, the legacy device will then behave as any other MultiNet-enabled device. This process creates the possibility of new deployment and service models, e.g., allowing creation of the outsourced configuration models described by Shehan and Edwards [19]. Manufacturers, retail outlets and network professionals could offer network configuration services, generating and printing configurations, and affixing them to the device for later use by the device owner.

Removing Devices

The network controller acts as a user interface to the network, showing a list of connected devices as in Figure 4. To remove a device, a user simply selects the device from the list and chooses remove. The network controller then sends a message to the MultiNet access point over the secure signalling channel, to remove the relevant network. Once removed preconfigured device can no longer access the network using its stored credentials, revoking its access to the network.

MULTINET IMPLEMENTATION

Figure 3 depicts an overview of MultiNet's implementation. The AP uses the *hostapd* user-space daemon to provide wireless access point and wireless authentication functionality conforming to the IEEE 802.11i WPA2 specification; *hostapd* is also used in many commercial AP implementations. We modified *hostapd* to enable dynamic creation and destruction of WPA2 secured networks by allocating memory for relevant state information, e.g., WPA2 configuration and corresponding virtual network interfaces, for up to 50 virtual networks.

After creating a new network, the corresponding virtual network interface is connected to the standard Linux layer 2 bridge (*br0*) so it can communicate with the other devices on the network. This bridge interconnects all the per-device virtual APs, i.e., all the configured wireless networks, so network traffic can flow freely between all devices (other than the network controller when it is running the MultiNet configuration app). Finally we modified the configuration reload code to only de-authenticate stations when the SSID or passphrase of a network is changed or removed from the configuration file. After modification, the configuration file is re-read whenever *hostapd* receives a `SIGHUP`, issued each time a new virtual AP is configured.

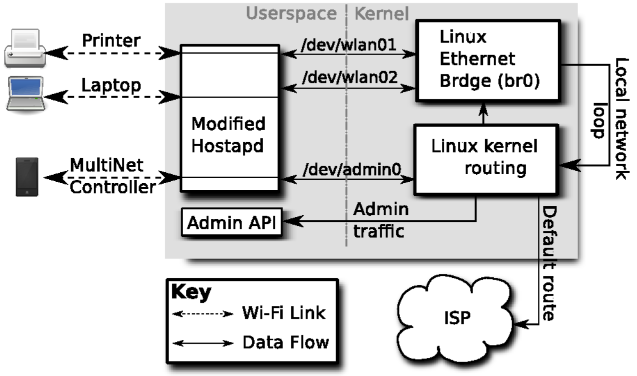
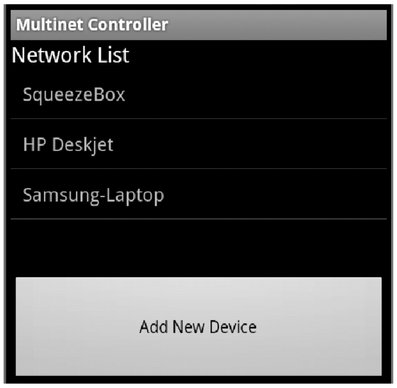


Figure 3: Overview of MultiNet implementation.


Multinet Controller

Network List

SqueezeBox

HP Deskjet

Samsung-Laptop

Figure 4: MultiNet network controller interface.

The MultiNet AP also provides a set of RESTful web services via which the network controller creates and destroys networks. These are only accessible over HTTPS using the trusted WPA2-secured administrative network configured on the network controller. Each network created is presented as a distinct virtual interface joined to the global bridge and routed to the outgoing upstream network interface. The only exception is the network controller network which is neither joined to the bridge nor routed upstream, ensuring its isolation and protecting the administration functions from all non-trusted devices connected to the network. As a result, the network controller only uses this network when running the MultiNet configuration application – the controller's normal network connectivity is via a standard device specific network.

The network controller is implemented as a simple application.[6] On startup, this attempts to join the administrative network. If successful, it displays a list of configured networks and a large button that can invoke the *add new device* action. It also provides feedback to the user when a device is successfully added to the network. The interface (Figure 4) is deliberately kept very simplistic to avoid confusion. We next discuss security related considerations of MultiNet.

[6] Android-only in our prototype.

SECURITY CONSIDERATIONS

Users are known to have limited understanding of the security issues surrounding Wi-Fi networks [9]. MultiNet aims to simplify the security model through three aspects of its design: its use of a proximate OOB channel, its use of QR Codes for credential storage, and its use of per-device credentials (SSID and passphrase). We will deal with each in turn.

MultiNet uses a proximate OOB channel which creates a straightforward threat model for the network controller and devices: *physical access is required*. This maps the wireless network's security onto the physical security of associated devices and has three important features. First, it removes the burden of managing and entering secure passphrases from the user, allowing the automatic generation of credentials. These can be longer and use a wider range of characters than those typically input manually, substantially increasing the search space and attack time for brute force attacks [22]. Secondly, it removes the reliance on Diffie-Hellman key exchange, used to construct a secure signalling channel on an insecure medium, used in methods like WPS which are vulnerable to man-in-the-middle attacks [12]. Finally, at no point is the network left in an open state as is the case with WPS PBC [24].

Using QR Codes for credential storage has advantages and disadvantages. Physical attacks require line of sight and reasonable proximity to capture a decodable image of the credentials. Such attacks are often difficult to carry out covertly, and thus have reasonably high chance of detection. Regrettably, if an attacker successfully reads the QR Code they have all the information required to decrypt the WPA2 secured traffic for that device or to spoof an AP to which that device will automatically connect. However, a number of strategies can be adopted to mitigate this potentially unacceptable risk. For example, QR Codes could be removable for secure storage; privacy covers could be added requiring a flap to be lifted to expose the QR Code; or, on devices with screens, transient codes could be displayed on screen when a physical button is pressed or application is launched.

MultiNet requires each device to have its own unique credentials. This enables temporary or permanent revocation of network access to a specific device, currently not possible with existing WPA2 deployments without reconfiguring all devices. Temporary revocation is useful in the home for visitation and punitive control; permanent revocation is useful when devices are sold or stolen. Re-associating temporarily revoked devices is identical to the normal association procedure. Re-associating permanently revoked devices has more overhead: a new SSID and passphrase pair must be generated, the device reconfigured, and the QR Code reprinted and reattached. If the network controller is lost, stolen or compromised, the AP must also be reconfigured with credentials for a new administrative network which must similarly be printed and affixed to the AP before the new network controller can be configured.

Finally the reduced interactional overhead combined with device specific credentials facilitates the creation of new security policies, impractical with existing mechanisms. For instance, revoking access after a set time period has elapsed, a period of absence, or a set number of connections has been made. We next briefly evaluate MultiNet's technical feasibility in terms of performance impact.

NETWORK PERFORMANCE

The purpose of our performance measurements is to examine the feasibility of MultiNet for deployment in domestic environments given that the high number of virtual access points it creates is outside the expected operating envelope of the protocols and components.

Method

We constructed a test environment to measure throughput, latency and jitter as the number of connected devices (and thus configured virtual access points) increased. These metrics were chosen as they are standard indicators of network performance. We used 27 identical Samsung R5800 laptops with fresh installs of Microsoft Windows 7 as clients, and an eeePC netbook as the AP. The standard networking tools *ping* and *iperf* were used to measure latency, throughput (over TCP) and jitter (over UDP). For each of these three performance indicators we configured the AP to offer the required number of networks and to act as a traffic sink. We connected the first Samsung laptop configured to act as a traffic source. Measurements were taken over a 60 second period after which the AP was re-configured and the experiment repeated, each time with an additional client laptop connected. We then repeated the whole experiment three times to generate the full dataset.

Results

The AP-to-device **throughput** (Figure 5a) displays approximately linear reduction as the number of networks and associated overheads increases, as expected. At 20 networks there is a 13% reduction in maximum throughput, and by 50 networks this figure has risen to 27%.

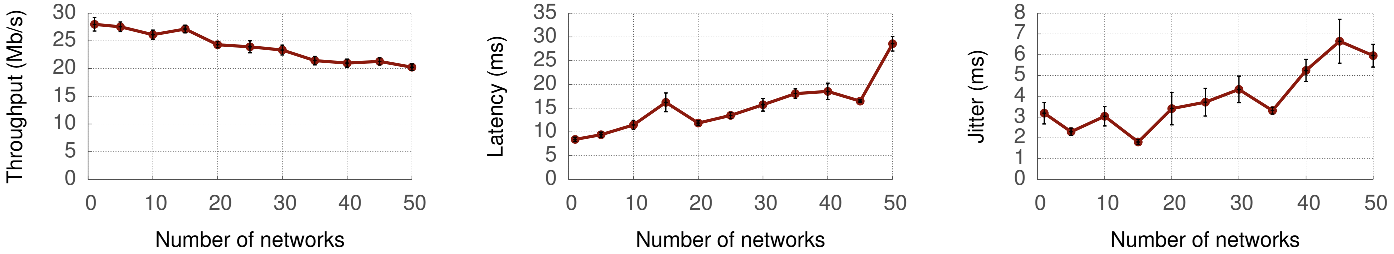
As the number of networks increases the **latency** on the networks also increases from 8 ms to 15 ms for up to 20 networks (Figure 5b). Although there appears to be the expected linear upward trend in per-packet latency as the number of networks and associated overheads increases, the degree of variation observed suggests that there are more significant factors directly affecting per-packet latency.

The **jitter** increases notably for more than 30–35 networks, but seems relatively constant when the number of networks is below that (Figure 5c).

Overall these measurements show that MultiNet has a limited impact on network performance for less then 25 networks, supporting its feasibility for deployment into the home.

DEVICE CONNECTION TIME

Device connection times for MultiNet and WPS were also measured, to check that MultiNet's performance is comparable to existing systems. These measurements were performed on the test set-up described in the user study. This was motivated by the need to check there were no adverse effects on device connection time with different device types.



(a) Throughput

(b) Latency

(c) Jitter

Figure 5: Device to AP network performance as the number of configured networks increases. The mean ± standard error is shown

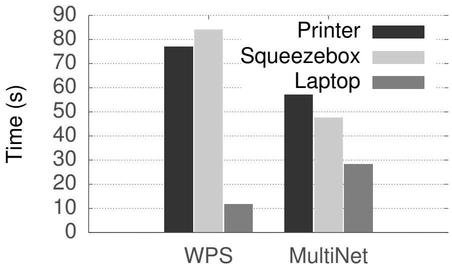


Figure 6: Average device connection time (s).

	WPS	QR Codes
Never	12	6
Occasionally	1	8
Often	3	2

Table 1: Prior exposure to WPS and QR Codes.

Age Group	18–24	25–34	35–44	45–54	>54
Participants	4	9	0	2	1

Table 2: User trial participant age range.

Device connection time has two components: the time required for the user to complete the appropriate configuration steps to initiate device connection; and the time taken for the relevant key exchanges and network configurations to take place to securely associate the device with the network. The first is measured in our user trial, presented below. The second is independent of the user, and is measured from the last step in the configuration process to the point where the device receives an IP address from the DHCP server running on the AP.

This test was performed 16 times per device and averages calculated and shown in Figure 6. In the case of the printer and Squeezebox, the average device connection time is notably lower with MultiNet compared to WPS, but the Windows 7 laptop connects more quickly with WPS than with MultiNet. This is because the laptop connects when Windows 7 detects presence of the pre-configured wireless network as it polls for available wireless networks, in contrast to the user initiated action of connecting via WPS. If the user were to boot the laptop after configuring the network, or were to manually initiate a poll, we expect the connection time for the laptop would come down.

USABILITY ANALYSIS

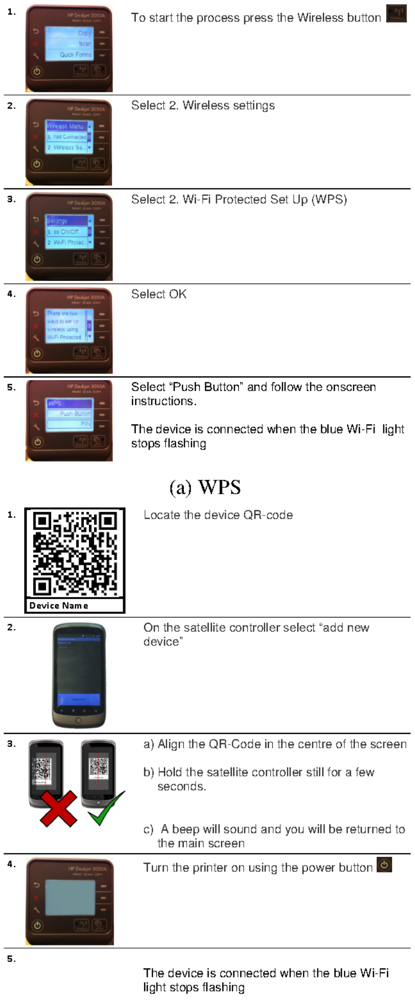
MultiNet's primary aim is to be more usable than current approaches, while being at least as secure and having minimal performance impact. To evaluate this we conducted a user study comparing MultiNet with WPS using PBC. This was chosen as it is a standardised usability-focused joining method with good market share [25]. The study was undertaken in lab conditions with all 16 participants performing the task of building a network using both WPS and MultiNet. All participants undertook each task in the same lab with the same facilitator following a short pre-study survey to assess past exposure to home networking, WPS and QR Codes. Post-study, semi-structured interviews were also used to explore participants' reactions to both systems.

Participants

The sixteen participants, ten male and six female, were recruited from our university campus using posters and mailing lists. No incentive to participate was given but a £10 Amazon voucher was offered as an inconvenience allowance. Ten participants listed themselves as the person who normally configures their own home network. Only three of the participants were very confident they would be able to configure a new wireless-enabled device they had purchased. Twelve of the sixteen participants had never used WPS before and six had never used QR Codes. The participants' home networks varied in size and complexity, with the number of connected devices ranging from 3 to 15 with a mean of 5.6 (SD = 3.8). Two participants were unable to provide detailed information on the makeup of their home networks. Tables 1 and 2 give more details on the composition of the participants.

Method

The user trial consisted of a single task to construct a network consisting of three consumer devices: an HP Deskjet 3050A e-All-in-One Printer; a Squeezebox Radio; and a Samsung laptop running Windows 7. There were two conditions: C1, connecting the three devices using WPS PBC, and C2, connecting the three devices using MultiNet. The order in which the subjects experienced the two conditions was randomised to minimise carryover effect between conditions. In each condition the devices were identical and added to the network in



1. To start the process press the Wireless button

2. Select 2. Wireless settings

3. Select 2. Wi-Fi Protected Set Up (WPS)

4. Select OK

5. Select "Push Button" and follow the onscreen instructions.

The device is connected when the blue Wi-Fi light stops flashing

(a) WPS

1. Locate the device QR-code

2. On the satellite controller select "add new device"

3. a) Align the QR-Code in the centre of the screen

b) Hold the satellite controller still for a few seconds.

c) A beep will sound and you will be returned to the main screen

4. Turn the printer on using the power button

5. The device is connected when the blue Wi-Fi light stops flashing

(b) WPS

Figure 7: Instructions for adding the printer to the network for WPS (top) and MultiNet (bottom).

the same order. We assumed that the MultiNet network controller had been previously bootstrapped to the network as this step would be carried out at network installation time, perhaps with the aid of the network service provider.

Participants received instructions on the use of the required features of the router before the task began in both conditions. Following the manufacturer's instructions for adding the printer, it took an experienced systems' administrator over 10 minutes to achieve a successful connection via WPS. Although more concise, instructions for the other devices were still excessively complex. Thus, to ensure consistency in the level of guidance provided, we rewrote instructions for all devices in both conditions. Great care was taken to keep the level, style and amount of instruction consistent between the two conditions. See Figures 7a and 7b for examples of the instructions provided.

Immediately after experiencing each condition, participants were asked to complete a System Usability Scale (SUS) questionnaire to gather their opinions on the systems usability. The SUS [5] is a "quick'n'dirty" usability evaluation tool comprising 10 short Likert scale questions designed to quickly gather users' opinions on system usability (effectiveness, efficiency and satisfaction). It generates a score from 0 to 100, with higher scores indicating greater usability.

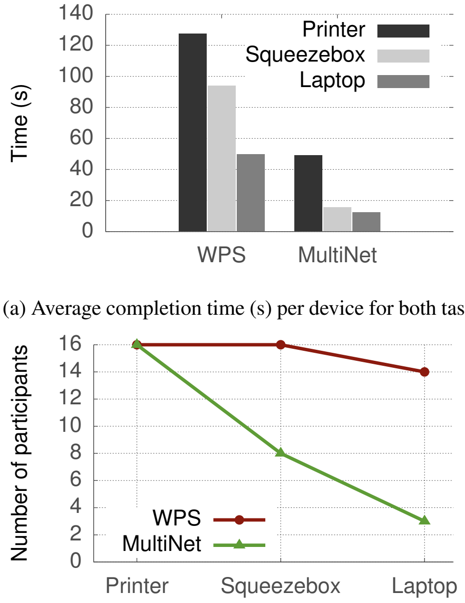
To compare both systems directly, we use *task completion time* as a measure of effectiveness, with lower times indicating greater effectiveness. We measured the time taken to complete each step in configuring the device as participants moved through the task under each condition. Timing started when the participant interacted with either the device or the instructions, and stopped when they had completed the last step in the configuration sequence. In the case of WPS we defined this as when the participant released the WPS button, successfully activating it. For MultiNet it was defined as the point the participant first turned on the device after successfully scanning the appropriate QR Code. These endpoints were chosen so the different connection times for the two methods did not affect the measured task times. (We discuss impact of connection time in the previous section.)

As well as task completion time we recorded whether the participant used the available instructions. Instructions were placed face down in front of each device and participants were asked to refer to them only if they felt they needed to. Instructions were recorded as having been used if a participant turned them over. After the participants had experienced both test conditions, a short semi-structured interview was conducted to understand their broader reactions. Each participant was asked the same questions and the answers were recorded for later analysis.

Results

The mean SUS scores attained in our trial are WPS 81.88 ($SD = 13.24$), and MultiNet 92.50 ($SD = 11.03$). Empirical evaluation of large numbers of SUS scores across a range of products has shown that scores above 73 indicate good usability and score above 85 represent excellent usability [4]. The SUS scores thus indicate that both WPS and MultiNet are usable solutions, with MultiNet rated as more usable by the participants in our trial. A paired t-test on the SUS scores shows this difference in usability to be significant ($t(15) = 5.36, p < 0.001$) with MultiNet having the higher mean score.

The mean task time measurements for connecting the printer using WPS and MultiNet were 127.56 seconds ($SD = 69.23$) and 48.06 seconds ($SD = 46.69$) respectively. A paired t-test shows a significant difference in task time for the printer of ($t(15) = 5.65, p < 0.001$) with MultiNet having the lower times. For the Squeezebox Radio, the mean task time using WPS was 94.37 seconds ($SD = 32.57$) and using MultiNet was 15.55 seconds ($SD = 9.76$). Again, a paired t-test shows a significant difference in task times for the Squeezebox Radio of ($t(15) = 10.29, p < 0.001$) with MultiNet having the lower times. Finally, connecting the Laptop using WPS gave a mean task time of 49.87 seconds ($SD = 33.6$) while using MultiNet the mean task time was 12.26 seconds ($SD = 5.23$). Once more, a paired t-test shows a significant



(a) Average completion time (s) per device for both tasks.

(b) Participants' use of instructions.

Figure 8: Evidence for the learnability of MultiNet.

difference in task time for the Laptop of ($t(14) = 4.61, p < 0.001$)[7] with MultiNet having the lower times. These results are summarised in Figure 8a. Analysing task completion time across all devices show that, using WPS the mean task time was 91.47 seconds ($SD = 57.21$), while using MultiNet the mean task time was 25.92 seconds ($SD = 32.17$). A paired t-test shows a significant difference in task completion time ($t(47) = 10.07, p < 0.001$) with MultiNet achieving the lower mean times.

Observed instruction usage is shown in Figure 8b. Observing the trends shown in Figure 8, task completion times decrease in proportion at least as quickly with MultiNet as with WPS; and there is a marked decrease in instruction use with MultiNet, from 16 to 3, compared with the small decrease from 16 to 14 for WPS. This suggests that users found the consistent point and connect interface of MultiNet easy to remember and learn.

Post Trial Interviews

All participants stated that they preferred MultiNet in post trial interviews. Users commented on the lightweight nature of the interaction:

> "find QR Code, scan QR Code; it's done. It is a lot more straightforward."

Users also commented on the consistency of the interaction across all devices as a factor in their preference for MultiNet:

> "[MultiNet] is the same each time so it's a lot easier to remember what to do. I did not need the instruction after the first time."

Users highlighted the ways in which the system might fit existing network settings in their homes. They highlighted that the router is not always conveniently located in relation to the device to be connected.

> ”When at home the devices will be positioned differently, the router may be in a cupboard or somewhere fiddly to get to, and the devices may be up flights stairs, which would be annoying."

They were also concerned about the need to coordinate interaction across two physical locations that might be some distance apart.

> "WPS would be a problem if the router was a long way away from the device"

> "[With WPS] if say, the router was in another room in the house that would be incredibly annoying"

This suggests that there is an advantage to localising interactions involved in joining a device to the network to the device's location.

DISCUSSION

We have presented and evaluated *MultiNet*, a new approach to joining devices to domestic wireless network that focuses on the process from the perspective of the user interaction. Configuring the network infrastructure to the device and introducing a network controller enables a consistent configuration metaphor without imposing new constraints on devices, whilst maintaining backwards compatibility with existing equipment. It also addresses the high interaction overhead of today's home networks by moving the configuration task from the devices to the network controller, in effect creating a single point of configuration for joining any device to the network. This is achieved by creating and maintaining multiple virtual APs, incurring negligible performance cost.

The initial performance evaluation suggests that MultiNet is technically quite feasible: although there is some impact on throughput, latency and jitter, it is acceptably small for up to 25 devices. While capping the number of devices to 25 is a limitation of MultiNet, we do not believe it will significantly affect today's domestic wireless deployments: participants in our user trials reported a mean of 5.6 devices connected to their home networks, which tallies with the 4.3 devices found in the *2011 Connectivity Report* that surveyed a thousand UK homes.[8] However, this is not a complete technical evaluation of the system. A more detailed study is needed to fully understand how the changes we have made will interact with many devices on a busy network. We did not investigate this performance loss in detail, but we hypothesise that it is related to both 802.11 beaconing interval and re-keying time.

The user trial shows that both systems performed well in a lab environment and all of the participants managed to complete the entire task in both conditions. However, MultiNet produced significantly better SUS scores across the trial, and all participants stated that they preferred MultiNet over WPS. The dramatically reduced instruction usage observed as participants moved through the task with MultiNet is an indication that the consistent interaction approach was readily

[7] One participant experienced a hardware failure while configuring the laptop with WPS so that data point has been removed.

[8] http://thenextweb.com/uk/2011/12/09/uk-households-have-an-average-of-4-6-devices-connected-to-wifi

learned by users. The qualitative comments also suggest a preference for the use of a network controller and that this was more easily applicable to peoples' domestic deployments where access to the AP is often limited. The participants also highlighted a number of real world issues with WPS, commenting on the problems arising from the spatial arrangement of the networking infrastructure in their homes and its unsuitability for distributed devices. One limitation of the user study is that we did not consider the usability of bootstrapping the network controller. We felt that it was not necessary to include as it is a one-off task that uses a roughly similar interaction to normal device association.

One presumptions in the design of MultiNet is the availability of a network controller. The network controller is required to complete the configuration process and reusing an existing device offers a low cost path to adoption. We envisage this actually being an app on a mobile device such as a smartphone, analogous to the approach taken by Cisco in their "Connect Cloud" platform. However, the concept of configuring the network to the device could be implemented using other proximate OOB transfer techniques. For example, one could easily replace the QR Codes with credentials encoded on a USB storage device. In this scenario plugging the USB device into the AP would act as the OOB channel, allowing the correct network to be configured on the AP without need for a network controller.

CONCLUSION

With MultiNet we have shown that it is possible to design a system for building home wireless networks with improved usability without compromising backwards compatibility, functionality or security. Adapting the network infrastructure to the device enables the creation of a trusted signalling channel over which credentials captured using the network controller can be securely conveyed to the router, enabling the granting and revocation of access to pre-configured devices. This eases the burden of configuration placed on the user, reducing the overall interaction overhead of the system.

The use of a network controller as a configuration intermediary in MultiNet enables the provision of a uniform interface across all devices to be associated with the network; our study shows this is beneficial to the usability of the system as a whole. This uniform interaction reduces the interactional overhead of device configuration in today's diverse home device ecosystem. The mobility of the network controller is also a better fit for the home context as it is sensitive to the spatial and aesthetic pressures evident in domestic environments.

ACKNOWLEDGMENTS

This work was supported by the Doctoral Training Centre in Ubiquitous Computing, RCUK grant EP/G037574/1 and by Horizon Digital Economy Research, RCUK grant EP/G065802/1.

REFERENCES

1. Aboba, B., and Simon, D. Extensible Authentication Protocol (EAP) Key Management Framework. RFC 5247, IETF, Aug. 2008.
2. Balfanz, D., Durfee, G., and Grinter, R. E. Network-in-a-box: how to set up a secure wireless network in under a minute. In *Proc. 13th USENIX Security Symposium*, USENIX Association (2004).
3. Balfanz, D., Smetters, D. K., and Stewart, P. Talking to strangers: Authentication in ad-hoc wireless networks. In *Proc. NDSS* (2002).
4. Bangor, A., Kortum, P. T., and Miller, J. T. An Empirical Evaluation of the System Usability Scale. *International Journal of Human-Computer Interaction 24*, 6 (July 2008), 574–594.
5. Brooke, J. SUS— a quick and dirty usability scale. *Usability evaluation in industry 189* (1996), 194.
6. Goodrich, M., Sirivianos, M., Solis, J., Tsudik, G., and Uzun, E. Loud and clear: Human-verifiable authentication based on audio. In *Proc. IEEE 26th ICDCS* (2006), 10.
7. Grinter, R. E., Edwards, W. K., Newman, M. W., and Ducheneaut, N. The work to make a home network work. In *ECSCW 2005*. Springer Netherlands, 2005, 469–488.
8. Ho, J. T., and Dearman, D. Improving users' security choices on home wireless networks. In *Proc. 6th SOUPS*, ACM (2010), 1–12.
9. Klasnja, P., Consolvo, S., Jung, J., Greenstein, B. M., LeGrand, L., and Powledge. "when i am on wi-fi, i am fearless": privacy concerns & practices in everyday wi-fi use. In *Proc. 27th ACM CHI*, ACM (2009).
10. Kobsa, A., Sonawalla, R., Tsudik, G., Uzun, E., and Wang, Y. Serial hook-ups: a comparative usability study of secure device pairing methods. In *Proc. 5th SOUPS*, ACM (2009), 1–12.
11. Kumar, A., Saxena, N., Tsudik, G., and Uzun, E. Caveat emptor: A comparative study of secure device pairing methods. In *Proc. IEEE PerCom* (Mar. 2009), 1–10.
12. Kuo, C., Walker, J., and Perrig, A. Low-cost manufacturing, usability, and security: An analysis of Bluetooth simple pairing and Wi-Fi protected setup. In *Proc. FC'07/USEC'07*, Springer-Verlag (2007).
13. McCune, J., Perrig, A., and Reiter, M. Seeing-is-believing: Using camera phones for human-verifiable authentication. In *Proc. IEEE Symposium on Security and Privacy* (May 2005), 110–124.
14. O'Brien, J., and Rodden, T. Interactive systems in domestic environments. In *Proc. 2nd DIS*, ACM (1997), 247–259.
15. Petersen, M. G. Remarkable computing: The challenge of designing for the home. In *CHI '04*, ACM (2004), 1445–1448.
16. Poole, E. S., Chetty, M., Grinter, R. E., and Edwards, W. K. More than meets the eye: transforming the user experience of home network management. DIS '08, ACM (2008), 455–464.
17. Sasse, M. A., Brostoff, S., and Weirich, D. Transforming the 'weakest link' – a human/computer interaction approach to usable and effective security. *BT Technology Journal 19* (2001), 122–131.
18. Saxena, N., Ekberg, J.-E., and Kostiainen, K. Secure Device Pairing Based on a Visual Channel: Design and Usability Study. *IEEE Trans. Information Forensics and Security 6*, 1 (2011), 28–38.
19. Shehan, E., and Edwards, W. K. Home networking and HCI: What hath God wrought? In *Proc. ACM CHI*, ACM (2007), 547–556.
20. Soriente, C., Tsudik, G., and Uzun, E. Hapadep: Human-assisted pure audio device pairing. In *Information Security*, vol. 5222 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 2008.
21. Stajano, F., and Anderson, R. The resurrecting duckling: Security issues for ad-hoc wireless networks. In *Security Protocols*, vol. 1796 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 2000.
22. Tasoluk, B., and Tanrikulu, Z. A weakest chain approach to assessing the overall effectiveness of the 802.11 wireless network security. *CoRR abs/1103.0464* (2011).
23. Tolmie, P., Crabtree, A., Rodden, T., Greenhalgh, C., and Benford, S. Making the home network at home: Digital housekeeping. In *ECSCW 2007*. Springer London, 2007, 331–350.
24. "Wi-Fi Alliance". Frequently Asked Questions : Wi-Fi Protected Setup. http://www.wi-fi.org/files/WFAWi-FiProtectedSetupFAQ.pdf, 2006.
25. Wi-fi Alliance. Wi-Fi CERTIFIED Wi-Fi Protected Setup: Easing the User Experience for Home and Small Office Wi-Fi Networks. http://www.wi-fi.org/knowledge-center/white-papers/, 2010.
26. Yang, J., and Edwards, W. Icebox: Toward easy-to-use home networking. In *Human-Computer Interaction INTERACT 2007*, vol. 4663 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 2007, 197–210.