ET (Smart) Phone Home!

Leandro Collares
University of Victoria, BC, Canada
leco@cs.uvic.ca

Chris Matthews
University of Victoria, BC, Canada
cmatthew@cs.uvic.ca

Justin Cappos
NYU Poly, Brooklyn, NY
jcappos@poly.edu

Yvonne Coady
University of Victoria, BC, Canada
ycoady@cs.uvic.ca

Rick McGeer
HP Labs, Palo Alto, CA
rick.mcgeer@hp.com

Abstract

Most home users are not able to troubleshoot advanced network issues themselves. Hours on the phone with an ISP's customer representative is a common way to solve this problem. With the advent of mobile devices with both Wi-Fi and cellular radios, troubleshooters at the ISP have a new backdoor into a malfunctioning residential network. However, placing full trust in an ISP is a poor choice for a home user. In this paper we present *Extra Technician* (ET), a system designed to provide ISPs and others with an environment to troubleshoot home networking in a remote, safe and flexible manner.

Categories and Subject Descriptors C.2.3-4 [*Computer-Communication Networks*]: Network Operations, Distributed Systems

General Terms Experimentation, Management

Keywords smartphones, distributed systems, sandbox environments, network troubleshooting, human-computer interaction

1. Introduction

Smartphones with different processing capabilities and operating systems that access resources via the Internet have recently become ubiquitous. Applications that run on those devices are now countless. In this work, we propose to use smartphones to troubleshoot network connectivity issues.

Network management, on the other hand, is not a novel topic and has been performed since primordial stages of computer networks. Depending on the type of network, several strategies are employed to diagnose problems. As for home network management, considerable amount of research has been conducted in the fields of Computer-Human Interaction (CHI) and Computer Supported Collaborative Work (CSCW). According to studies, like those of Edwards [13] and Calvert [8], home users find it extremely complex to set up, troubleshoot and secure home networks, and often have to rely on other individuals to have these tasks accomplished.

A failing Internet connection in a household can be a topic of much anguish to a home user. The user has to spend their own time talking to technical support representatives, or worse (for the user) pay someone else to come to their house to help troubleshoot their problems with their ISP. On the other hand, ISPs have to maintain call centers with network troubleshooting staff, which ultimately makes the Internet service more expensive. The ISP-customer relationship can be tarnished as the user often blames the service provider for the outage, even if it is within their own network. Outages also promote a general pessimism about computers among novice users.

The administration of home networks entails an intricate labour division among family members according to their knowledge of networking concepts. This fact is aggravated by the relationships with companies householders have to maintain to keep home networks operational. Studies [16, 20] reveal that some families rely on 3-7 external entities for infrastructure support. Interactions with multiple stakeholders can be frustrating as householders are often directed to a different company to resolve their problems. Besides, home users are wary about conceding access to their personal devices to external support professionals.

Since the number of smartphones has been increasing and the possibility of having at least one of them in a household is high, the use of those devices as back door into home networks could provide a new avenue for troubleshooters to gather much more detailed information before making a

SPLASH'11 Workshops, October 23–24, 2011, Portland, Oregon, USA.
Copyright © 2011 ACM 978-1-4503-1183-0/11/10. . . $10.00

prognosis. The main novelty of this approach is the use of a fully operational network (3G) to debug the malfunctioning home network, which will be accessed via Wi-Fi. However, this scenario raises security and privacy concerns that have to be addressed. Section 2 presents details of the outlined scenario and a solution to minimize the risks introduced.

2. Technical Approach

As mentioned in section 1, the Wi-Fi radio of a smartphone is a gateway to home networks and the 3G network a powerful tool to troubleshoot home network connectivity issues.

Home users can have test suites sent to their smartphones upon request or have them installed on the devices after hiring the ISP, or even an independent troubleshooter. This approach provides administrators with information on the network from end devices' perspective (a perspective which is new to network debugging), which could anticipate the adoption of measures to solve problems and, therefore, improve network availability levels. On the other hand, home users would have a valuable tool to rule out problems in their home networks before contacting ISPs. Home users could alternatively ask reliable people from their social networks to perform home network troubleshooting with the test suites.

Though smartphone-based troubleshooting of network issues sounds promising, there are some practical issues that must be overcome, most paramount, user security and privacy. Opening a back door to a user's network is obviously a bad idea from a security perspective. Although in the intended use case it provides useful information, if a malicious user (either from within the ISP, or a hacker) were able to control the network debugging program, they would have full access to the user's home network. This could lead to snooping of the user's private data, or the introduction of malware into the home network. We propose the use of sandboxing techniques to mitigate these risks, to both eliminate the possibility of the network debugger being hijacked, and to control what access the debugger has to the user's network and smartphone.

A home network debugging suite based on a sandbox environment for smartphones provides users with valuable, safely-obtained information about their network. Sandboxes limit the resources and access of applications and, therefore, prevent security breaches and minimize the impact on local machines.

The applications are confined to sandboxes created in both the home computer and the smartphone, which means personal information from customers will remain private and safe. Moreover, the impact on devices' performance is minimal since the use of CPU, memory, network bandwidth and storage space is limited by environmental variables. It is also worth emphasizing that it is possible to control the network aspects that are exposed to householders, which caters for both beginners and advanced customers.

We focus on the development of tools for troubleshooting home network issues using the *Seattle* [4, 9] platform and its sandbox environment *RePy* installed on smartphones. This approach to network troubleshooting enables users to solve connectivity issues and might also reduce the interactions between users and technical support personnel or at the very least make those occasions more agreeable.

We also recommend extensions to *Seattle* that can enhance the troubleshooting ability of home users and network administrators while retaining an acceptable privacy / security trade-off. Concepts can be applied to other network management scenarios as well.

In the remainder of the paper we first discuss related work on home network troubleshooting and present information on *Seattle* and *RePy*. We then describe tools that were developed with *Seattle* and the general challenges to implement home network troubleshooting using that platform on mobile devices. Finally, we assess the benefits and trade-offs of using *Seattle* in the aforementioned scenario and present concluding remarks.

3. Related Work

As mentioned in Section 1, CHI and CSCW studies reveal that home network management can be daunting even to qualified individuals [15].

Home networks inherited the complexity of the TCP/IP stack, designed to support ARPANET, the precursor to the Internet, which was a military-class, highly resilient network of networks. Ease of use at host computers, which would correspond to PCs in a home network, was not a design goal of ARPANET, as this network was conceived to be used by experts [21].

Home networks are also highly dynamic and heterogeneous. Growth is not usually planned and new devices are added to the network on a whim. Furthermore, the physical location of content shared among users hinges on domestic routines, home layouts and even aesthetic principles [13, 22], which leads to countless variations of home network configurations.

Privacy expectations are extremely high in home networks as users often store deeply personal information in their computers and are unwilling to grant ISPs or external troubleshooters access to those devices [13, 21].

All the aforementioned factors have a considerable bearing on home network troubleshooting. Due to network complexity, several non-expert users rely on resetting, unplugging and replugging actions for troubleshooting [21, 23]. Other users turn to parents or friends who are more familiarized with network concepts for informal technical support [19]. Analysis of interactions between home users and technical support professionals over the phone suggests that the latter are usually unaware of the particularities of home networks [20]. On the other hand, householders have great difficult providing accurate information on their networks.

Their descriptions tend to include domestic aspects (e.g. devices' physical positions and "owners") but ignore information about network topology [13, 18]. One key challenge in network management tool design is to decide which network aspects should be exposed to householders so that they can build conceptual models that predict how the network works [13].

As for existing network management / troubleshooting tools, home users usually consider tools built into operating systems and routers cryptic. Hidden command line tools e.g. *ping* and *traceroute* do not provide root causes of problems [18, 23]. Mac OS *Network Utility* [1] streamlines common troubleshooting tools (*netstat*, *ping*, *lookup*, *traceroute*, *whois*, *finger* and *portscan*) by means of a graphical user interface (GUI). Furthermore, *Network Utility* presents information on IP addresses, MAC addresses, link speed and status, and transfer statistics.

Other tools only deal with specific home network issues. Systems like *Network-in-a-box* [7] and ICE*box* [25], for example, focus on wireless network configuration. Management suites such as IBM *Netview*, HP *OpenView* and *PacketTrap* were designed for enterprise-wide networks and, therefore, fail to address typical home networking tasks. Moreover, those applications require professional network managers to be effectively used [23].

Network Magic [3], *Eden* [24] and MAGNETO [14] exemplify comprehensive applications for home network management. Pure Networks's *Network Magic* is a Windows-based tool that provides users with a basic visual map of the home network and allows troubleshooting by means of a "wizard-like" interface. *Eden* is an interactive visual tool that attempts to address issues as network conceptual models and heterogeneous home network environments. *Eden* not only lends itself to network management but also works as a Linux router that can be run on PCs or replace the software of Broadcom-based consumer routers. MAGNETO is a distributed management architecture based on agents deployed to both ISP and home network environments. Fault diagnosis is accomplished via Bayesian inference. A more holistic approach was taken by Dixon *et al.* [12] who propose *Home*OS, an operating system that addresses heterogeneity across homes.

A wide range of applications use smartphones due to their relatively powerful processors, storage capabilities and built-in sensors. Smartphones have been employed in WAN monitoring and visualization as well [17]. However, to the best of our knowledge, there are no applications that use a sandbox environment and smartphones to troubleshoot home network connectivity issues.

4. Seattle and Repy

Smartphone capabilities create enticing possibilities for home network troubleshooting. However, they also introduce security vulnerabilities that could compromise users' data, if exploited by ill-intentioned individuals. Consequently, having a sandbox environment associated with a smartphone-based home network troubleshooting tool is of utmost importance.

Seattle [4, 9] is a free, community-driven platform which runs on multiple operating systems (Linux, Mac OS, Windows) and currently offers limited support for Android and iOS.

Seattle can be employed for research on a myriad of topics including cloud and ubiquitous / mobile computing, distributed systems and peer-to-peer networking and its users can utilize resources from other end-users systems, provided they donate resources from their own machines.

Seattle hinges on *RePy* (or Restricted Python) [10], which is a sandbox environment. *RePy* assures that programs will run only inside sandboxes and, consequently, pose no threat to machines whose resources have been shared.

As mentioned before, most users are neither able to perform connectivity issues on their home networks nor interact successfully with support staff over the phone. Furthermore, users are not comfortable with the idea of granting access to storage devices to professionals during troubleshooting processes due to privacy concerns. *Seattle*-based troubleshooting tools address all these issues and, consequently, can change the way home network management is carried out.

5. Home Network Troubleshooting

Firstly we consider a simple scenario: a user, who has access to the Internet by means of a wired LAN, cannot load a page on a browser. The list of potential problems could include:

- disconnected cable
- Ethernet interface down
- invalid IP address provided by DHCP
- site temporarily unavailable
- DNS-related problems

Even though OS-provided tools are deemed complex by average users [18, 23], the following steps could be used to illustrate measures taken to pinpoint the cause of the problem or at least narrow it down:

- check computer's IP and gateway to rule out DHCP-related problems,
- ping the local interface to check if the network adapter is working accordingly,
- ping the default gateway to check it is reachable, or other network hops close within the network,
- try loading a different page as the intended one might be temporarily down and
- ping the intended website using its hostname instead of its IP address to check if DNS is working properly

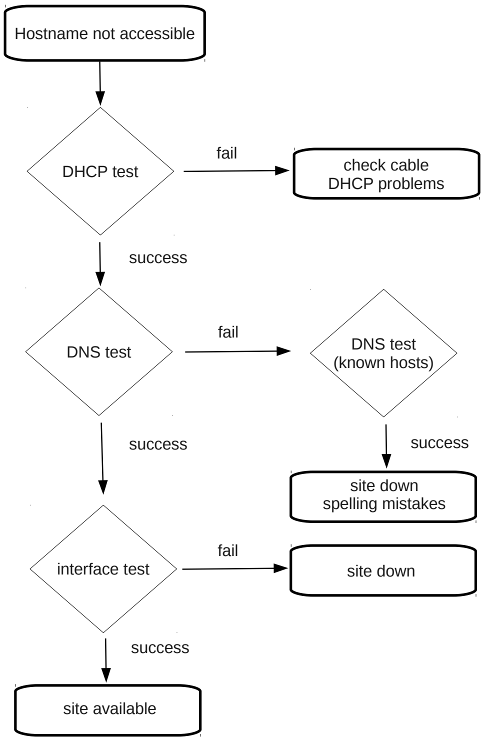

Figure 1. Developed suite flowchart

In the case of a home network, this could be taking place in the form of a support person explaining the steps over the phone, then the user reading aloud the results from several commands.

Despite some constraints, a similar troubleshooting process can be implemented with *Seattle* and *RePy*. Paradoxically, the properties which make *Seattle* such a good environment for distributed systems implementation limit its use as a platform for network debugging. As the *Seattle* platform strives to be safe, there is a myriad of restrictions on the type of interactions *RePy* programs can make with the system. The *RePy* API limits the user to simple file operations (rate limited and on files in the local directory only), basic rate limited TCP and UDP connections, network operations like DNS lookups and getting the local IP addresses, and a few system operations like threading, time, and random numbers. For safety reasons (and a lack of previous need) ICMP *ping* has not been implemented in *Seattle*. Though UDP *ping* is possible, it requires the existence of UDP *ping* servers, which are rare on the Internet. Furthermore, it is not possible to pinpoint the default gateway's IP address via *Seattle*. These particularities do not prevent us from developing several tools for troubleshooting connectivity issues as shown in the following section.

6. Implementation

The *RePy* primitives provide the basis for a troubleshooter to write a script. That script can run tests and report back the results. By means of a *command-line* interface, users running the test suite receive messages regarding possible problems that prevent the page from being loaded: network cable, IP address assignment, name resolution, spelling mistakes, site temporarily down. Figure 1 depicts a basic flowchart of the developed test suite.

The test suite was successfully run on remote computers within the *Seattle* platform and locally on Linux and Windows machines, and on Maemo-based Nokia N810 tablets.

6.1 Ping Testing

Ping is not available in *Seattle* and the platform does not offer support for ICMP packets. Therefore we wrote code that would run on server and client to implement UDP *ping*.

This tool can be employed to check if there is disruption in communications, provided UDP servers are available on ISPs' networks.

6.2 DNS Testing

Several tests can be run to ascertain DNS is working properly. We can keep a small list of hostnames whose IP addresses are unlikely to change and the corresponding IPs. Using *gethostbyname_ex()* function with one of those hostnames as argument and comparing the result with the IP address stored, we can determine if name resolution works.

That approach, however, might result in false alarms since the IP address returned might depend on the country the user is. To avoid that we created a pair hostname / IP address that will not change under any circumstances.

6.3 HTTP Connections

After confirming name resolution is working accordingly and considering ICMP *ping* is not possible within the *Seattle* realm, we verify if a hostname is available by trying to establish TCP connections on port 80 to all IP addresses returned by *gethostbyname_ex()* function.

This test can be easily implemented thanks to networking functions provided by *RePy* API.

6.4 Opening Internal Web-pages

One useful debugging tool which falls out of HTTP connection is the ability to query the web interfaces of functioning hardware. This could be used to open and relay a home router's status page, or log page.

6.5 Timed Operations

RePy provides timers, so timing network operations is easy. Very slow connections often can look like they are just not working, so it is vital to be able to time network operations to confirm they are within expected norms.

7. Evaluation and Implementation Constraints

Although *RePy* API comprises a small set of functions, it provides enough capabilities to develop a powerful and safe test suite for home network troubleshooting. *RePy* inherits simplicity and elegance from *Python* and can be quickly mastered by individuals with some knowledge of other programming languages.

Limitations imposed by the platform (see section 5) can be overcome to a certain extent. Considering that the network base address is the first IP address in a given subnet and the broadcast address is the last one, it is feasible to deduce the most probable default gateway's IP addresses.

The communication between the two radio systems (Wi-Fi and 3G) is probably the main challenge in implementing troubleshooting applications in *Seattle* as the platform support for multi-homed devices is still embryonic. ICMP ping would be also a welcoming addition to *RePy*'s network API.

ET can incorporate new functionalities in the future as it is based on *RePy*, which is an ever-evolving environment.

8. Conclusions and Future Design

Home networking technologies are still intimidating to most users, who have to rely on friends, relatives and phones calls to ISP customer representatives to diagnose network problems. Furthermore, home users are reluctant to allow troubleshooting tests to be executed on their devices owing to privacy concerns.

Extra Technician, ET, developed with *RePy* to be run on smartphones, constitutes a new approach to home network troubleshooting that successfully addresses the aforementioned issues. Based on the implementation of the test suite for home network troubleshooting previously described, we strongly believe it is feasible to deploy *RePy*-based ET(*Extra Technician*) on smartphones.

ET can be sent to smartphones in reply to text messages sent by customers or after the ISP is hired. Alternatively, users can download the script using the smartphone connection to the Internet.

Technically inclined individuals that belong to users' social networks (e.g. relatives, friends) can install ET on their smartphones to perform home network troubleshooting. The information presented on the screen of the phone will help the user pinpoint the root cause of the problem and solve it. ET can also run in background for continuous monitoring of the home network conditions and the user will then receive an email or text message on the phone when the network is down. GPS information provided by the smartphone can be employed to generate geographically-aware test suites. It is worth emphasizing that information regarding the location of the device will not leave the phone under any circumstances.

ISPs can develop several versions of ET tailored according to customers' knowledge of networks. NOC (Network Operations Center) personnel will also have the possibility of writing ETs to deal with specific situations. To detect problems on the ISP infrastructure (e.g. optical and radio links) with accuracy ETs will be sent to multiple clients of a certain region. Results will be gathered at the NOC and provide professionals with "snapshots" of the network from different perspectives. Technical visits to remote areas will be avoided since ETs are quickly sent to users by means of cellular networks.

Sandboxes rules will keep users' files private and safe. The impact of running ETs on smartphones and PCs will be minimal as a result of tightly-controlled environmental variables.

A graphical user interface (GUI) for ET could also be envisioned. However, some key issues have to be addressed during the design process. Any alternative to *Seattle* would have to offer at the very least the same level of security of *RePy*. Furthermore, the migration from a command-line to a graphical interface would probably reduce the scope of smartphones that are able to run ET and increase energy consumption, especially with regard to the continuous monitoring mode. Users could also interact with ET by means of voice commands in a way similar to conversations between iPhone users and *Siri* [5].

Though ET is a rudimentary proof-of-concept tool for network troubleshooting, it can certainly be tailored to fulfil the requirements of systems in which smooth network operation is of paramount importance, e.g. online entertainment platforms such as *iTunes* [2] and *Steam* [6] and clients based on the *BitTorrent* protocol [11].

References

[1] Apple Network Utility. `http://www.apple.com/macosx/apps/all.html#network`. Accessed on October, 9 2011.

[2] iTunes. `http://www.apple.com/itunes/`. Accessed on October, 9 2011.

[3] Network Magic. `http://www.purenetworks.com/`. Accessed on August, 1 2011.

[4] Seattle - Open peer-to-peer computing. `https://seattle.cs.washington.edu/wiki`. Accessed on August, 1 2011.

[5] Siri. Your wish is its command. `http://www.apple.com/iphone/features/siri.html`. Accessed on October, 9 2011.

[6] Steam - Online Game Platform. `http://steampowered.com/`. Accessed on October, 9 2011.

[7] D. Balfanz, G. Durfee, R. E. Grinter, D. K. Smetters, and P. Stewart. Network-in-a-box: How to set up a secure wireless network in under a minute. In *Proceedings of the USENIX Security Symposium*, pages 207–222. USENIX, 2004.

[8] K. L. Calvert, W. K. Edwards, and R. E. Grinter. Moving toward the middle: The case against the end-to-end argument in home networking. In *Proceedings of the Sixth ACM Conference on Hot Topics in Networks*. ACM, 2007.

[9] J. Cappos, I. Beschastnikh, A. Krishnamurthy, and T. Anderson. Seattle: a platform for educational cloud computing. *SIGCSE Bull.*, 41(1):111–115, 2009.

[10] J. Cappos, A. Dadgar, J. Rasley, J. Samuel, I. Beschastnikh, C. Barsan, A. Krishnamurthy, and T. Anderson. Retaining Sandbox Containment Despite Bugs in Privileged Memory-Safe Code. In *The 17th ACM Conference on Computer and Communications Security (CCS '10)*. ACM, 2010.

[11] B. Cohen. The BitTorrent Protocol Specification. `http://www.bittorrent.org/beps/bep_0003.html`. Accessed on October, 9 2011.

[12] C. Dixon, R. Mahajan, S. Agarwal, A. J. Brush, B. Lee, S. Saroiu, and V. Bahl. The home needs an operating system (and an app store). In *Proceedings of the Ninth ACM SIGCOMM Workshop on Hot Topics in Networks*, pages 1–6. ACM, 2010.

[13] W. K. Edwards, R. E. Grinter, R. Mahajan, and D. Wetherall. Advancing the state of home networking. *ACM Communications Magazine*, 54(6):62–71, 2011.

[14] F. Garcia-Algarra, P. Arozarena-Llopis, S. Garcia-Gomez, and A. Carrera-Barroso. A lightweight approach to distributed network diagnosis under uncertainty. In *Proceedings of the International Conference on Intelligent Networking and Collaborative Systems*, pages 93–98. IEEE, 2009.

[15] R. E. Grinter, W. K. Edwards, M. Chetty, E. S. Poole, J.-Y. Sung, J. Yang, A. Crabtree, P. Tolmie, T. Rodden, C. Greenhalgh, and S. Benford. The ins and outs of home networking: The case for useful and usable domestic networking. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 16(2):1–28, 2009.

[16] R. E. Grinter, W. K. Edwards, M. W. Newman, and N. Ducheneaut. The work to make a home network work. In *Proceedings of the European Conference on Computer-Supported Cooperative Work*, pages 469–488. Springer Netherlands, 2005.

[17] K. Mitchell, N. J. P. Race, and M. Clarke. CANVIS: context-aware network visualization using smartphones. In *Proceedings of the 7th International Conference on Human Computer Interaction with Mobile Devices & Services*, pages 175–182. ACM, 2005.

[18] E. S. Poole, M. Chetty, R. E. Grinter, and W. K. Edwards. More than meets the eye: transforming the user experience of home network management. In *Proceedings of the 7th ACM Conference on designing interactive systems*, pages 455–464. ACM, 2008.

[19] E. S. Poole, M. Chetty, T. Morgan, R. E. Grinter, and W. K. Edwards. Computer help at home: methods and motivations for informal technical support. In *Proceedings of the 27th International Conference on Human Factors in Computing Systems*, pages 739–748. ACM, 2009.

[20] E. S. Poole, W. K. Edwards, and L. Jarvis. The home network as a socio-technical system: Understanding the challenges of remote home network problem diagnosis. *Computer Supported Cooperative Work*, 18(2-3):277–299, 2009.

[21] E. Shehan and W. K. Edwards. Home networking and hci: what hath god wrought? In *Proceedings of the SIGCHI Conference on human factors in computing systems*, pages 547–556. ACM, 2007.

[22] P. Tolmie, A. Crabtree, T. Rodden, C. Greenhalph, and S. Benford. Making the home network at home: Digital housekeeping. In *Proceedings of the European Conference on Computer-Supported Cooperative Work*, pages 331–350. Springer/Kluwer, 2007.

[23] J. Yang and W. K. Edwards. A study on network management tools of householders. In *Proceedings of the 2010 ACM SIGCOMM Workshop on home networks*, pages 1–6. ACM, 2010.

[24] J. Yang, W. K. Edwards, and D. Haslem. Eden: supporting home network management through interactive visual tools. In *Proceedings of the 23rd annual ACM Symposium on user interface software and technology*, pages 109–118. ACM, 2010.

[25] J. Yang and E. W. K. Icebox: Toward easy-to-use home networking. In *Proceedings of the IFIP Conference on Human Computer Interaction*, pages 197–210. Springer, 2007.