

CS CAPSTONE TECHNOLOGY REVIEW

NOVEMBER 21, 2017

SMART HOME INTERCOM SYSTEM

PREPARED FOR

OREGON STATE EECS

D. KEVIN MCGRATH

PREPARED BY

JORDAN DAVIS

Abstract

The Smart Home Intercom System has many dependencies. Three of these dependencies are user interface packages, network protocols, and operating systems. This document describes three possible approaches to each and chooses one approach to use per dependency.

CONTENTS

1	Network Protocol	2
1.1	Overview	2
1.2	Criteria	2
1.3	Potential Choices	2
1.3.1	B.A.T.M.A.N. Advanced	2
1.3.2	Netsukuku	2
1.3.3	Qt Network Programming	2
1.4	Discussion	3
1.5	Conclusion	3
2	User Interface	3
2.1	Overview	3
2.2	Criteria	3
2.3	Potential Choices	3
2.3.1	Qt 4.8	3
2.3.2	Kivy	3
2.3.3	GTK+	4
2.4	Discussion	4
2.5	Conclusion	4
3	Operating Systems	4
3.1	Overview	4
3.2	Criteria	4
3.3	Potential Choice	4
3.3.1	Raspbian	4
3.3.2	Windows IoT	4
3.3.3	Snappy Ubuntu	5
3.4	Discussion	5
3.5	Conclusion	5
	References	6

1 NETWORK PROTOCOL

1.1 Overview

Smart Home Intercom System units need to be able to securely connect to each other without requiring the backbone of external networking devices. The units should be scalable as more get added on or taken off without interrupting the other devices.

1.2 Criteria

The selection criteria for network protocols is ultimately the amount of potential support or write-ups a protocol has online. The other factors that were considered are OSI layer operations, ease of adding or removing nodes and ease of use.

1.3 Potential Choices

1.3.1 *B.A.T.M.A.N. Advanced*

B.A.T.M.A.N. Advanced, or batman-adv, is a networking protocol that can be used to create mesh networks. It is added on the second layer of the OSI model, meaning that all traffic it sends out uses raw data [1]. It can be mixed with non-B.A.T.M.A.N. systems as well, which is potentially beneficial for some stretch goals [2]. There are also optimized guides available online for Linux systems [3]. One of the primary benefits of batman-adv is that it can support many subnets but it does require each device have its own IP address previously assigned [4].

1.3.2 *Netsukuku*

Netsukuku is another potential mesh network protocol that operates on level three of the OSI model [5]. Netsukuku is primarily for larger peer-to-peer networks, wherein IP addressing and routes are calculated on the fly. Netsukuku faces an interesting challenge within itself because it cannot dynamically interchange roles on the protocol. It assigns the roles of "Static Node" and "Guest Node" that would make it potentially hard to work with on a system that could change easily [6]. Perhaps the most challenging thing about Netsukuku is that it is implemented in vala and python. Vala is a programming language that focuses on allowing programmers the ability to write in high-level languages but don't want the overhead that they require [7]. While the language itself is not too much of an issue, the potential lack of support given vala is a more obscure language and the age of Netsukuku make it a non-optimal solution.

1.3.3 *Qt Network Programming*

Qt is predominantly a user interface technology, but it has modules for network programming. "The QtNetwork module offers classes that allow you to write TCP/IP clients and servers" [8]. It offers a full listing of classes available for its module and examples of its usage [9]. These examples, particularly multicast receiver and multicast sender would help the project move forward faster. There are potential issues with Qt as it lacks much documentation on interconnecting with only its own nodes, as a mesh network would require. Despite this, it is an open source software and could therefore be modified to fit our needs [10].

1.4 Discussion

Batman-adv offers the ability to operate on layer 2 while Netsukuku forces operations on layer 3. The ability of batman-adv to send raw data among its nodes instead of touching the data, like Netsukuku does, makes it stand out. No protocol or package makes adding and removing nodes particularly easy, with batman-adv being the easiest as a node can be scripted to connect itself. Qt is an incredibly fitting package with a module that is supportive of what the project needs. However, it falls under an unknown category as what is being asked of it might simply not be possible.

1.5 Conclusion

The optimal solution is to split the difference. Qt has a network protocol that operates via the same operations as a normal device would while batman-adv operates on layer 2, allowing it to encapsulate and send out everything it is told to. Because of this Qt is the first implementation to test and it can be run alongside batman-adv if Qt doesn't have the security or ability to be molded as it needs to be.

2 USER INTERFACE

2.1 Overview

The first thing anyone will see when they look at a unit will be its user interface. It will need to be responsive and customizable for a users own individual preferences while making sure it doesn't fall short on options available to the user.

2.2 Criteria

The criteria for user interface (UI) is proven ease of use for the user. User interfaces should optimize the user experience, while still offering optimal performance. The technology should offer extensive examples of it being used for purposes similar to those required. API references are a necessity to streamline the making of the interface.

2.3 Potential Choices

2.3.1 Qt 4.8

Qt 4.8 is an open source application and UI framework that supports many platforms [10] [11]. Qt even offers a guide for starting with the Raspberry Pi [12]. It offers support for both Linux/X11 and Windows systems [13] [14]. Along with supporting various operating systems, it contains class libraries and integrated development tools to help support its own environment [13] [14]. Qt is primarily cross-platform which means that it will be easier to develop with [11]. Beyond that, Qt contains examples and guides for its systems [15].

2.3.2 Kivy

Kivy is an "open source Python library for rapid development of applications that make use of innovative user interface" [16]. It can run on a Raspberry Pi and is partially written in "C using Cython" [16]. Kivy offers an extensive and detailed API [17]. Beyond its extensive API are examples and a detailed getting started guide [18]. Because it is written using Python, there is potential for it to easily run on other devices, should more testing be required.

2.3.3 GTK+

GTK+ can be used in a multitude of ways [19]. It offers a large number of interface options along with offering an all inclusive toolkit [20]. GTK+ is supported by a majority of operating systems and has packages to allow for operations in many programming languages [20].

2.4 Discussion

All three choices offer extensive API catalogues and can be supported on a single board computers. Examples are offered by each package that show it doing what will be required of it. Even cross platform usage is possible with all options. The primary difference is that Qt 4.8 uses C++, Kivy uses Python and GTK+ can use either Python or C++. Perhaps the most interesting point here is that, despite all being user interface packages, only Qt 4.8 offers an IDE and user interface to go along with itself for testing.

2.5 Conclusion

The ideal package is Qt 4.8 primarily because it contains an easily usable and optimized development environment.

3 OPERATING SYSTEMS

3.1 Overview

Operating systems will either increase or reduce overhead and allow for ability of potential hardware and software addons.

3.2 Criteria

For the Smart Home Intercom System project, the operating system should be concealed and locked away from the end user. The operating system should be able to handle hardware that is attached to it as well as support running Qt and potentially the B.A.T.M.A.N. protocol.

3.3 Potential Choice

3.3.1 Raspbian

Raspbian is a Debian distribution optimized for the Raspberry Pi [21]. Because it is based off of Debian, there is extensive support for Raspbian [22]. Additionally, Raspbian is fairly well supported by the Raspberry Pi community [23]. Due to Raspbian being a variety of Debian, it is possible to modify installation packages to make boot time into the smart home intercom system software minimal.

3.3.2 Windows IoT

Windows IoT is a variety of Windows 10 made for the Raspberry Pi [24]. It is optimized to be used with Visual Studios and used to create automation projects [25]. With an ISO image of only 651.8 megabytes [24], Windows IoT is incredibly lightweight. The major downfall of Windows IoT is requiring non-onboard hardware. For instance, the Raspberry Pi camera module is not supported via Windows IoT and requires an external webcam.

3.3.3 *Snappy Ubuntu*

Snappy Ubuntu is relatively lightweight and offers an optimized approach to software packages [26]. It forces programs to not interrupt the system when running and only passes through what it needs. A snap contains metadata that will allow apps to be updated as they are run or called. This allows the newest changes to be pushed onto the device [27].

3.4 Discussion

While Snappy Ubuntu offers the ability to add or remove packages on the fly, the ideal system would be unable to connect to the internet which would force the primary utilities of snappy to falter. Windows IoT requires an external webcam due to being unable to use certain parts of the Raspberry Pi hardware. Raspbian is not lightweight, as it is the basic operating system for a Raspberry Pi, and would require extensive cleaning up to make it lightweight and supportive of the end goal.

3.5 Conclusion

Ultimately, Raspbian is the most optimized solution for our problem given current available technologies. It forces certain issues to be addressed, such as removing packages that were previously installed but will be the most supportive of Qt and it has the ability to use batman-adv.

REFERENCES

- [1] B.a.t.m.a.n. advanced. [Online]. Available: <https://www.open-mesh.org/projects/batman-adv/wiki/Wiki>
- [2] B.a.t.m.a.n. advanced quick start guide. [Online]. Available: <https://www.open-mesh.org/projects/batman-adv/wiki/Quick-start-guide>
- [3] Debian batman-adv autostartup. [Online]. Available: https://www.open-mesh.org/projects/batman-adv/wiki/Debian_batman-adv_AutoStartup
- [4] B.a.t.m.a.n. general questions. [Online]. Available: <https://www.open-mesh.org/projects/batman-adv/wiki/Faq#BATMAN-General-questions>
- [5] Netsukuku - p2p mesh routing. [Online]. Available: <http://netsukuku.freaknet.org/>
- [6] Netsukuku topology. [Online]. Available: http://netsukuku.freaknet.org/main_doc/topology.pdf
- [7] Vala - compiler for the gobject type system. [Online]. Available: <https://wiki.gnome.org/Projects/Vala>
- [8] Qt 4.8 network programming. [Online]. Available: <http://doc.qt.io/qt-4.8/network-programming.html>
- [9] Qt 4.8 network examples. [Online]. Available: <http://doc.qt.io/qt-4.8/examples-network.html>
- [10] Open source versions of qt. [Online]. Available: <http://doc.qt.io/qt-4.8/opensourceedition.html>
- [11] Supported platforms. [Online]. Available: <http://doc.qt.io/qt-4.8/supported-platforms.html>
- [12] Raspberry pi beginners guide. [Online]. Available: http://wiki.qt.io/Raspberry_Pi_Beginners_Guide
- [13] Support for linux/x11. [Online]. Available: <http://doc.qt.io/qt-4.8/linuxx11-support.html>
- [14] Support for windows. [Online]. Available: <http://doc.qt.io/qt-4.8/windows-support.html>
- [15] How to learn qt. [Online]. Available: <http://doc.qt.io/qt-4.8/how-to-learn-qt.html>
- [16] Kivy: Cross-platform python framework for nui development. [Online]. Available: <https://kivy.org/#home>
- [17] Kivy framework. [Online]. Available: <https://kivy.org/docs/api-kivy.html>
- [18] Getting started >introduction. [Online]. Available: <https://kivy.org/docs/gettingstarted/intro.html>
- [19] The gtk+ project. [Online]. Available: <https://www.gtk.org/>
- [20] Gtk+ features. [Online]. Available: <https://www.gtk.org/features.php>
- [21] Frontpage - raspbian. [Online]. Available: <http://www.raspbian.org/>
- [22] Raspbian faq. [Online]. Available: <http://www.raspbian.org/RaspbianFAQ>
- [23] Best operating systems for raspberry pi 2 & 3. [Online]. Available: <https://www.raspberrypistarterkits.com/products/operating-systems-raspberry-pi/>
- [24] Windows 10 iot core for raspberry pi 2 / 3. [Online]. Available: <https://www.microsoft.com/en-us/download/details.aspx?id=53360>
- [25] Windows 10 iot core. [Online]. Available: <https://developer.microsoft.com/en-us/windows/iot>
- [26] What is a snap? [Online]. Available: <https://docs.snapcraft.io/snaps/intro>
- [27] Snaps on ubuntu. [Online]. Available: <https://www.ubuntu.com/desktop/snappy>