Security and Efficiency of an Open Wireless Campus Network

Christian D. Tuen
UCSB Graduate Student
christiandt@umail.ucsb.edu

Kine Johnsrud
UCSB Graduate Student
kinej@umail.ucsb.edu

Andreas Løvland
UCSB Graduate Student
asl@umail.ucsb.edu

Abstract

Having a WLAN (Wireless Local Area Network) is now so commonplace in academic campuses, that it is more or less taken for granted by its users. Hence, it is increasingly important to understand usage and usage-changes in a technological world that is rapidly changing.

In this paper, we want to take a look at the general usage, efficiency and security of a campus network as a whole, by using passive network monitoring software. Over a span of 5 weekdays we detected 956 unique devices at the UCSB Davidson Library, by monitoring for two hours daily.

We found that there is a large increase in use of mobile OSes since earlier studies [5], and that OSX has superseded Windows. We observed that some of the most used applications are media-heavy, and that the campus network might not handle the pressure. We also observed that half the traffic being transferred over the unsecured network is secured by means of SSL. We discovered trends in what websites were visited the most, and what application are most widely used. Lastly, we found that there was a substantial amount of control packages, in addition to retransmission-packages and faulty packages, making for a relative low general goodput.

1 Introduction

In this day and age WiFi connectivity is taken for granted, and the number of mobile devices per person has rapidly increased in the later years. Public companies and institutions are not always the first to adapt to rapidly increasing usage of such wireless networks. In this paper we conduct a study of todays usage of a campus network, and talk about how well the network performs. University of California Santa Barbara (UCSB) has recently launched its secure wireless network, but many students are still using the unsecured one.

Firstly, we show what operating systems are used now, and how many of them are on the emerging mobile platform. This is set in comparison to a paper analyzing a campus-network in 2004 [5]. We also analyze whether there are any obvious usage patterns, i.e. if we can get a strong top 10 websites visited and applications used. Since this is an unsecured network, it is also interesting to see whether the connections are generally secured, and whether they are secured well.

This is a network that has a large user base, and the library is often densely packed with people and devices. This led us to believe that this network might be congested, especially at given access points. We show how much of the channel is used to send control packets, as well as analyzing whether packets sent can be considered as goodput (not being retransmissions or faulty packets). Our findings show that the network might be a bit congested, and assume that this is due to a large user base, and a lot of obstacles and movement.

We next describe the environment of our study, the UCSB Davidson Library, and how we conducted our data collection period. In section 3 we present our results of the operating systems comparison, then and now. In section 4 we talk about the security aspect of an open wireless network, and show some performance metrics comparing the secured and unsecured connections. Section 5 shows the usage patterns of the students using this network, by means of websites requested and applications used. In section 6 we discuss our findings regarding the amount of control packages sent in the network, as well as how much of the data sent is actual goodput. The paper gets wrapped up with a conclusion in section 7.

2 Test environment and trace collection

The UCSB Davidson Library is of medium size, and is generally full of people. Observation shows that it is reasonable to assume that all students own a laptop, and many students own one or more other mobile devices in addition (smartphone, tablet, etc.).

The passive monitoring were performed using personal computers, sitting in the same spot in the library each day of the experiment. It was done Monday through Friday in week 43, between 1pm and 3pm. The data was not stripped down to only containing headers, and the complete raw dataset ended up being 26 GB. We did the capturing with two different devices (A Mac and a Linux machine). In hindsight we would have done differently because we have had some compatibility issues, but in general we still think our results are correct and presentable.

We expected to see a wide variety of devices connected, ranging from Mac and Windows PCs, to a wide variety of smartphones and tablets.

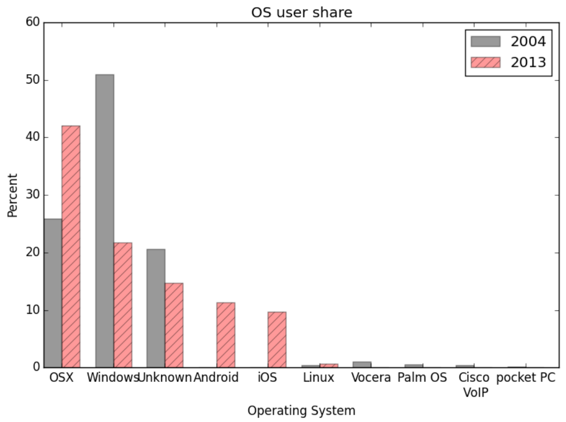

Figure 1: Distribution of Operating Systems

3 Operating Systems

An inspiration for writing this paper is a research paper from 2004 doing an extensive analysis of a campus-wide network [5]. They have made a table showing all the devices(operating systems) seen on the wireless network. They used a passive OS fingerprinting tool named p0f [7], and we started of with that as well. However, the tool is a bit outdated, and we chose to find the operating systems ourselves by looking at the User Agent fields in the request-packets.

This was before the smartphone-era, and in figure 1 we can see the changes in operating systems and devices between now and 2004. In addition to Apples OSX superseding Microsofts Windows, we now have mobile platforms (android and iOS) taking up 20 % of the total. Even though the emerging mobile market is a well known fact by now, it is highly interesting to see its effect on wireless networks and its performance. We will explore this further in subsection 6.2.

Encryption Type	Percent
TLS_RSA_WITH_AES_128_CBC_SHA	29.97%
TLS_ECDHE_RSA_WITH_RC4_128_SHA	17.63%
TLS_RSA_WITH_AES_256_CBC_SHA	13.83%
TLS_RSA_WITH_RC4_128_SHA	11.79%
TLS_RSA_WITH_RC4_128_MD5	9.40%
TLS_ECDHE_ECDSA_WITH_RC4_128_SHA	6.32%
TLS_DHE_RSA_WITH_AES_256_CBC_SHA	3.80%
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	3.45%
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	0.92%
TLS_DHE_RSA_WITH_CAMELLIA_256_CBC_SHA	0.76%

Table 1: List of cryptographic suites and their usage

4 Security and performance

In this section we present the security aspect of an open wireless network, and take a look at the performance differences between secured and unsecured connections.

4.1 Security

From our dataset, we found that 62 % of the connections were unsecured HTTP-sessions, and the other 38 % connections were secured by SSL/HTTPS. The increasing use of HTTPS-connections can be assumed to be an effect of popular social websites like Facebook and Twitter securing their sites by default.

A secured connection that is secured badly is often worse than an unsecured connection, since sensitive information often is the reason for securing the connection in the first place. We therefore found it reasonable to investigate whether the cryptographic suites used were properly secure. The cryptographic suites used is shown in table 1, and they are all considered secure enough by the Standards Organizations [1]. This is natural considering the new enthusiasm for privacy and security, and the fact that the biggest and most used websites needs to be properly secured to keep their reputation.

4.2 Performance

When using SSL, there is reason to believe that there will be a difference in performance. Other studies [6] show that there is a performance-loss when using SSL, because time and processing power is used to encrypt and decrypt. In figure 2, we see that average Round-Trip-Time is in fact larger for HTTP than HTTPS. This can be a cause of various reasons, but shows that there is not a noticeable difference between the two. Average session length is more than three times longer with HTTPS. This is natural, as the servers would like to keep the encrypted connection for a longer period of time, and not having to renegotiate cryptographic suites.

From the two graphs on the right in figure 2, we can see that HTTP has more than double the average amount

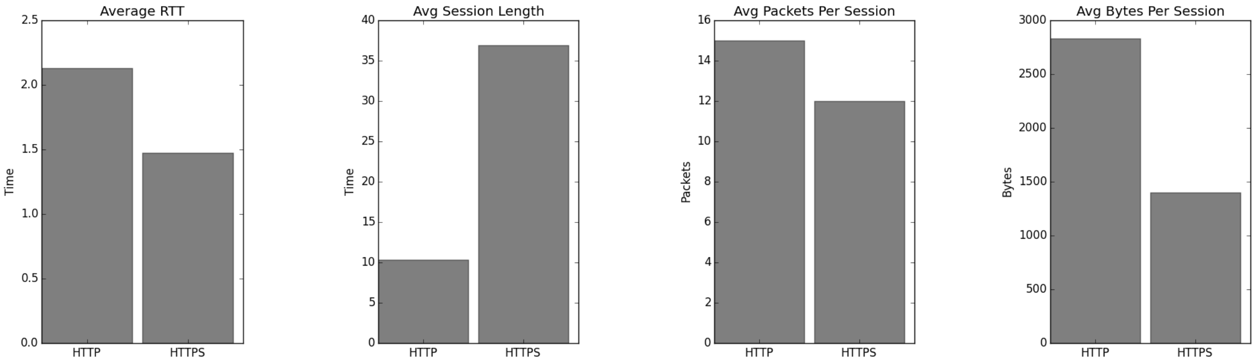

Figure 2: HTTP and HTTPS differences compared to RTT, average session length, average number of packets per session and average number of bytes per session

of bytes per session than HTTPS, even though the average number of packets per session is close to equal between the two. This might be explained by the fact that a lot of video streams (youtube, gauchocast) are sent unsecured, and a lot of secured links are simple websites like Facebook and Twitter, as well as mail services.

5 Usage: Web sites and application

To better understand the usage of a wireless network, it is essential to take a look at what the users are actually doing on the wifi, as well as what the devices are doing without a user noticing it. In this section we will take a look at popular websites, and what applications are using the network. By applications, we count desktop-applications as well as mobile.

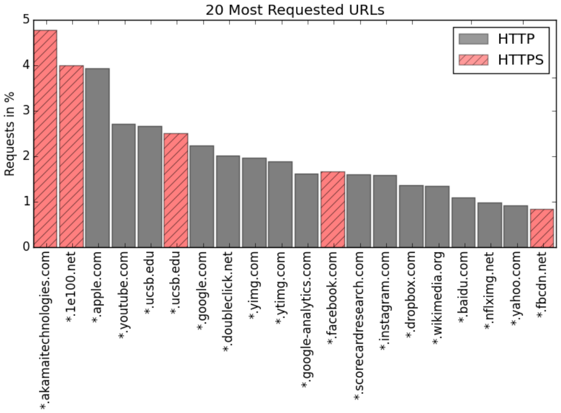

Figure 3: The 20 most requested URLs, color coded by their use of HTTPS or HTTP

5.1 Web sites

In relation to usage-patterns, we initially hoped to get a clear top 10 list of the most popular websites out there in number of requests. We count both the number of HTTP requests, and HTTPS requests by doing a reverse DNS lookup on the ”server hello”-source address. As the results have a long tail and a slow drop off, we included the top 20 URLs in our analyzation shown in figure 3. These 20 URLs account for 41.69 % of all HTTP/HTTPS requests, and should provide a good picture of usage. Most of the URLs are CDNs, which correlates with how CDNs are used in general. For instance, in a sample test, ytimg.com was requested more than 20 times when loading the YouTube frontpage.

If we look at the URLs in terms of web pages, we see that most have two or more occurrences in the list. The reason for this is that bigger companies, handling many requests per second, are only encrypting data that needs to be encrypted while sending data that do not need to be encrypted on unencrypted connections as encryption adds processing load on the servers. All of the sites that are not hidden behind ”Akamai” are either social websites, school related, advertisements or file/video/image related.

5.2 Applications

To retrieve information about applications, we used the HTTP-requests User Agent field. Even though a lot of applications use an encrypted connection to the backend API, most applications also connect to third party unencrypted analytics services. This way, we will still see an HTTP request from these applications, and be able to gather User Agent data from the devices.

In figure 4, we show the number of unique

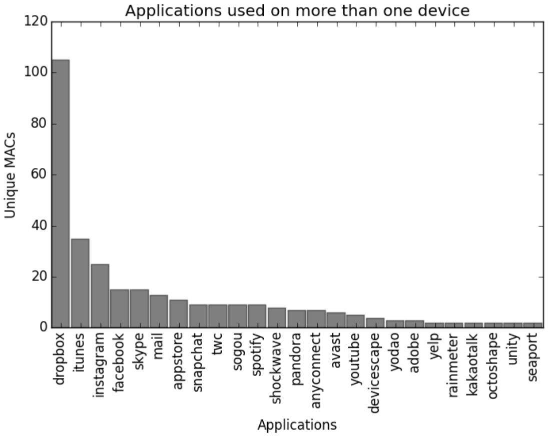

Figure 4: Distribution of unique MAC addresses per application

MAC-addresses sending HTTP-request with application-information. For readability, we choose to show the applications used by more than one single device. This graph shows that the most used applications are all media-heavy applications potentially consuming a lot of the bandwidth. The top application is Dropbox, that automatically downloads new and modified files in the background. Daemon processes running in the background which the user is potentially unaware of, consumes a lot of the bandwidth that could have been used more effectively for other purposes.

6 Control Packets

In trying to understand whether this network is congested or not, it is natural to find out about the ratio between control packets and data packets in general, as well as goodput versus throughput. Throughput is all data sent, including retransmissions and overhead, while goodput is actual useful data received at application level. In this section we will first look at goodput vs. throughput, followed by focusing on sending of RTS and CTS packets.

6.1 Goodput vs. Throughput

When looking into what of the data is considered goodput, we also have to take packet loss and retransmissions into account, as well as flawed packets. This amounts to 68 % of the packets not being goodput, and one third of all the bytes sent (figure 6). In other words, only two thirds of all data sent are useful information. This can both indicate a congested network, as well as a network with a lot of conflicting noise. With many people gathered in a small space trying to make many requests at the same time, this is a reasonable assumption to make. The library is highly used by the universitys students, and due to construction work the library has less space available than usual.

To get a perspective, we found that around 56 % of the data sent was control packets, but due to the relative small size of control packets, it only added up to about 10 % of the total data traffic (figure 5). This indicates that a large amount of the flawed data sent on this network is retransmitted packages, and flawed packages (with checksum error).

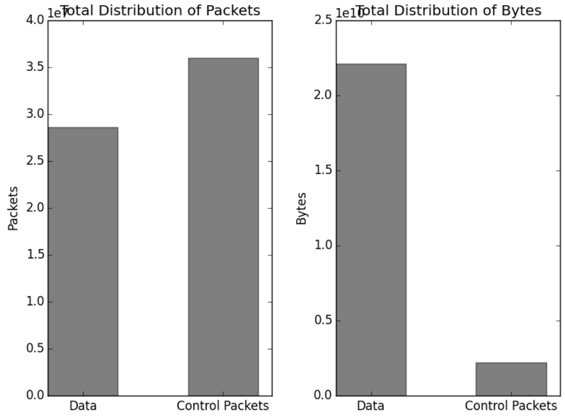

Figure 5: Left: The number of control packets compared to data packets. Right: The number of bytes generated by data and control packets

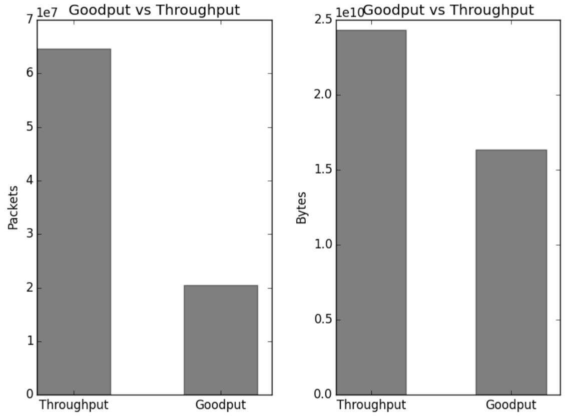

Figure 6: Left: Goodput versus Throughput in term of packets. Right: Goodput versus Throughput in term of bytes

6.2 RTS & CTS

Request-to-Send and Clear-to-Send is an 802.11 mechanism used to reduce frame collisions due to hidden terminal problem. The use of this mechanism is a good fairness-idea, but is often disabled by default in most wireless cards. However, we found that the average amount of RTS and CTS packets sent, were as high as 5 % of the total number of packets. We decided to investigate this further.

We did a lookup of the MAC addresses sending the RTS and CTS packets, and based on vendor (figure 7), it was obvious that this was mainly the Cisco Access Points using this function, but also mobile devices. After Cisco, the top three MAC adresses were from HonHai, LiteOn and Apple. These are all companies making mobile devices (Apple), or making chips used in a variety of smartphones (HonHai, LiteOn). A little research showed that RTS and CTS or CTS-to-Self is something that many smartphones use when WiFi is enabled. Other studies [2] has used this fact to track wifi-enabled smartphones.

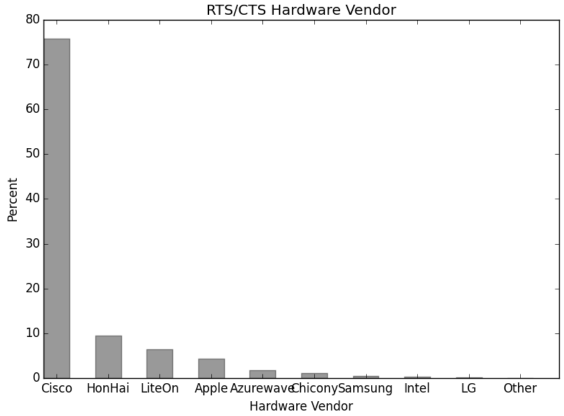

Figure 7: Percentage of unique MAC addresses that has sent RTS or CTS, grouped by vendor

7 Conclusion

We conducted a study of a campus wireless network, in an effort to understand mechanisms and patterns of activity in the network. It is important to remember the context - our results stem from a population of university students, and may not be relatable to corporate firms or other public space venues.

During the hours of capturing, the traffic was constantly high and a large set of clients were connected at all times during these two-hour periods.

We were able to subtract information about what operating systems people used, and then also finding out how many of the connected devices were smartphones and tablets. This was put into perspective and compared to a study from 2004.

We were able to figure out how much data is sent over SSL, and what cryptographic schemes they used. We also concluded that they were all considered safe by the Standards Organizations. To see usage-patterns, we made a list of the top 20 most visited URLs on the Internet, and found out that URL-requests are usually for social websites, university-related, advertisement or video-streaming. When it comes to application-usage, we found that most of the applications with highest user base, are also applications with media-content and generally large files being synced or streamed over the network.

Lastly, we did a study of control packets, to better understand whether the network was performing well or not. 56 % of all packets sent were control packages, but due to their small size it was only about 9 % of the total number of bytes sent. More alarmingly, we found that only two thirds of the data sent is what we consider goodput. With so many users trying to make requests at the same time at such a small place, this is to be expected. The library is heavily used by the universitys students, and due to construction work the library has less space available than usual.

8 Related Work

Our study is a small scale and passive measurement of a single point in a campus network. There are many similar and larger scale studies looking at these kind of characteristics. Henderson et al [5] did an extensive network trace including more than 550 access points and 7000 users over seventeen weeks. This kind of study would be impossible for us to make, but we were still able to compare some of our data. Similar to that campus-study, is a study by Kotz et al [3] using a campus-wide network of 476 access points spread over 161 buildings.

In studying whether the network performs different whether SSL is used or not, Vicenç et al [6] has done an extensive study over the computational and resource demand of encryption and decryption of data. Senad et al [4] has done an extensive vulnerability study of the security architecture of a campus network. Even though we have only been able to study the security between end-user and access point, the security aspect of a university network is highly important and interesting.

References

[1] National institute of standards and technology, nist.gov.

[2] A. B. M. Musa and Jakob Eriksson. Tracking Unmodified Smartphones Using Wi-Fi Monitors. In *Proceedings of SenSys12* (Toronto, ON, Canada, Nov. 2012).

[3] David Kotz and Kobby Essien. Analysis of a Campus-wide Wireless Network. In *Proceedings of MobiCom02* (Atlanta, Georgia, USA, Sept. 2002).

[4] Sanadi Al Maskari, Dinesh K. Saini, Swati Y Raut and Lingraj A Hadimani. Security and Vulnerability Issues in University Networks. In *Proceedings of the World Congress on Engineering* (London, U.K., July 2011), vol. 1.

[5] Tristan Henderson, David Kotz, Ilya Abyzov. The Changing Usage of a Mature Campus-wide Wireless Network. In *Proceedings of MobiCom04* (Philadelphia, PA, USA, Oct. 2004).

[6] V. Beltran, J. Guitart, D. Carrera, J. Torres, E. Ayguad and J. Labarta. Performance Impact of Using SSL on Dynamic Web Applications. In *9th International Parallel and Distributed Symposium (IPDPS05)* (Denver, CO, USA, April 2005).

[7] Zalewski, M. Passive os fingerprinting tool, 2003.