WiFiMon:
A Tool for Wi-Fi Performance Monitoring and Verification

Christos Bouras, Computer Engineering and Informatics Department, University of Patras, Patras, Greece

Kurt Baumann, Network SWITCH, Zurich, Switzerland

Vasileios Kokkinos, Computer Engineering and Informatics Department, University of Patras, Patras, Greece

Nikolaos Papachristos, Computer Engineering and Informatics Department, University of Patras, Patras, Greece

Kostas Stamos, Greek Research and Technology Network, Athens, Greece

ABSTRACT

Measuring network quality of a wireless network as experienced by end-users is quite difficult, as there is not a single tool available that can record measurements on all sides of the system. The approach presented in this research work is based on the end-user feedback, giving the opportunity of visualization of network performance in real time. This paper initially presents an overview of the developed tool, called WiFiMon, which has the ability to capture, record measurements and export statistics on the quality of Wi-Fi network as perceived by the end-users. The measurements are initiated by the end-users—without their intervention—after they visit a webpage or use a mobile application. WiFiMon aims to give a clear understanding of the Wi-Fi network conditions by measuring specific parameters of the network, such as download/upload throughput, and correlate these measurements with raw data from various log files to obtain additional information regarding the performance of specific access points. The results reveal the functionality of the proposed tool and its scalability.

KEYWORDS

Crowdsourced Network Performance, Mobile Application, Performance Monitoring, Performance Verification, Wireless Network

INTRODUCTION

In recent years there has been a significant increase in the number of devices that users spend daily working, surfing on the internet, using for social networks or even using to play mobile applications, which normally need internet access. The last technology step of smart devices like smartphones and tablets are becoming an integral part of human life as effective and convenient communication tools.

Therefore, it is imperative to know the requirements of end-users that are connected to a particular Wi-Fi network in order to satisfy their demands for seamless networking. The ability to measure and verify the quality of the wireless network (Wi-Fi) as experienced by end-users is quite difficult, since

DOI: 10.4018/IJWNBT.2019010101

there isn't a single tool that captures and performs network monitoring and verification from the user's perspective (Nielsen, 2015). From a statistical point of view, the large amount of measured data (by end-user activities) allow us to analyze the performance of a wireless network, even in crowded sites such as university campuses or conference venues. In these extreme cases, users usually face a number of issues with the performance of the wireless network. The WiFiMon tool, which is being implemented by GEANT[1], comes as a solution to this problem and tries to fill in the gap of network performance as experienced by end-users (Cooke et al., 2006).

WiFiMon attempts to push the wireless network monitoring and verification cycle at a higher layer. The main focus is placed on improved measurement verification, user friendly GUI development, mobile device app deployment, and WiFiMon as a Service (service automation). It is a tool mainly targeted for wireless network administrators, since the ability to visualize the collected data and monitoring the network status in real time can lead to improved decision making and efficient access points installation (Le et al., 2015).

The process that WiFiMon utilizes can be described as a process of collecting, storing, analyzing, visualizing and consuming data, which aims to correlate the end-users measurements with the information regarding the Access Point (AP) from where the measurement was initiated (Hernandez-Campos & Papadopouli, 2005). This process is not intended to replace the traditional measurements from hardware probes, which are installed at a fixed location. Instead, our intention is to supplement the hardware-probe measurements with "non-invasive" performance measurements captured by the end-users (through WiFiMon) so as to extract a hybrid solution that combines static (hardware-probe based) objective measurements with opportunistic measurements for end-users' devices.

From the practical point of view WiFiMon supports the network administrators to decide if there is a need to re-deploy their WiFi network. This means that WiFiMon Hybrid approach establishes an alert system, which detects performance degradation in real-time and based on the performance benchmark (history). The main benefit of such a system is the performance verification, which in turn results in efficient decision-making. Furthermore, based on the performance history, the system administrators can make short/mid-term prognoses (and network adjustments) of the network performance depending on the end-user's density. Overall, WiFiMon as a service/product, provides a quality statement that allows for a cost-effective network planning (CAPEX/OPEX).

The remainder of the manuscripts has the following structure: The next Section presents the related work in the wireless network performance monitoring. In Section "WiFiMon Architecture", the architectural view of WiFiMon approach is being described by presenting the building blocks of WiFiMon as well as their functionality for both the web based and the mobile-app concepts. Authors also explain the procedure that enables the correlation of the measurements with the information provided in various log files to obtain access point related information, such as Radius logs accounting. Section "Performance and Testing Results" presents the performance and scalability testing results. Finally, the last two Sections include the conclusions and an overview of the planned future steps respectively.

RELATED WORK

In the scope of monitoring there are several projects and initiatives designed to allow the remote monitoring of Wi-Fi APs. Today most of these approaches have the objective to monitor and verify the performance using controllers or to monitor stand-alone base stations using ping or, in advanced cases, base station scripts (Choffnes et al., 2011). In a controller-based Wireless Local Area Network (WLAN), base station status is generally observed by monitoring and processing Simple Network Management Protocol (SNMP) Trap messages. In addition to monitoring, controllers improve network quality through centralized channel partitioning and transmit power control. Traditionally, the service quality experienced by the user or the user's terminal has received little attention, even though air interface scanning has been performed to a certain extent. Cisco Wireless Control System (WCS), was mainly developed for the control and monitoring of Cisco WLAN controllers, but also includes features for collecting and visualizing data from the air interface. On the other hand, Airwave's Wireless Management Suite includes several modules for WLAN monitoring and control. For the monitoring of the air interface AirWave[2] has developed the VisualRF module, which can be utilized to check signal strength and channel partitioning. Both Cisco WCS[3] and AirWave are projects developed to perform network monitoring based on data from base stations and the fixed network without taking into account the end-user perspective. This was our motivation for WiFiMon; and based on the approach that was suggested by Huadong et al. (2014), we use frequently-visited websites to measure and monitor how the users experience the quality of the wireless network, i.e. our approach is based on the end-user perspective.

There are of course some products that focus on the end-user side. Motorola AirDefense Mobile, can be installed on laptops running Windows XP or 2000 with an Atheros based IEEE 802.11 a/b/g network card, such as Netgear (WAG511) or Cisco (CB21AG) allowing monitoring of traffic on the WLAN and focusing heavily on information security. Wireshark[4], is an open source software for Windows and Linux computers for the analysis of network protocols. Wireshark as a service allows the straightforward analysis of traffic on the computer in question but is not so well suited to a comprehensive analysis of WLANs. Analysis of control and beacon frames requires use of the monitoring mode, which is not supported by all operating systems, adapters and drivers and mobile devices. These approaches require an installation from the end-user, while the results of the performance are not communicated to the network administrators.

There are also some online tools/pages, such as Ookla Speedtest[5], where users can manually initiate measurements and get information about the performance of the Wi-Fi network. Performance metrics usually include information about the download and upload throughput and Round-Trip Time (RTT) as mentioned in Saeed et al. (2007) and Sang Young (2007). These online tools however require the users to initiate the measurements, but most importantly they only inform the user that initiated the test for the measurement results and not the network administrator who is responsible for setting up the wireless network and optimizing its performance. In other words, these online tools are efficient for checking the quality of the Wi-Fi network in users' premises; however, for sites with a large number of APs a different, more efficient approach should be followed.

Moreover, some research studies have been made in traffic monitoring in university campus and conference rooms. In Gember et al. (2011), the authors study the differences in the content and flow characteristics of handheld versus non-handheld traffic in university campus. They analyze packet traces from two separate campus wireless networks, with 3 days of traffic. Authors used the above study to inform network management companies and providers for the current status of the network quality in these busy spots (campuses & conference rooms). Having this in mind; WiFiMon gives network administrators a clear overview of the current Wi-Fi topology of the APs, the quality that end-users experience in terms of Wi-Fi as well as take decision on APs alternation in order to improve the performance of Wi-Fi network in these environments.

The main characteristics of WiFiMon that differentiates it from the solutions presented above are: 1. it allows "non-invasive" performance measurements without users' intervention, 2. there is a correlation, triangulation of performance data, AP identifiers (AP-ID), and end-user information (RADIUS Logs), 3. the correlated datasets are posted to the same database so that the network administrator has a view of Wi-Fi-performance (complete history) and the end-user behavior on the campus (heat maps). The large number of correlated datasets allow us performing time series analysis and making predictions to Wi-Fi-capacities, user-frequency (locations). Further, it supports strategic issues in growth wireless connectivity on the campus (university or a conference environment). Therefore, WiFiMon seems to further improve and extend the network monitoring development cycle as well as extend the current state of the art.

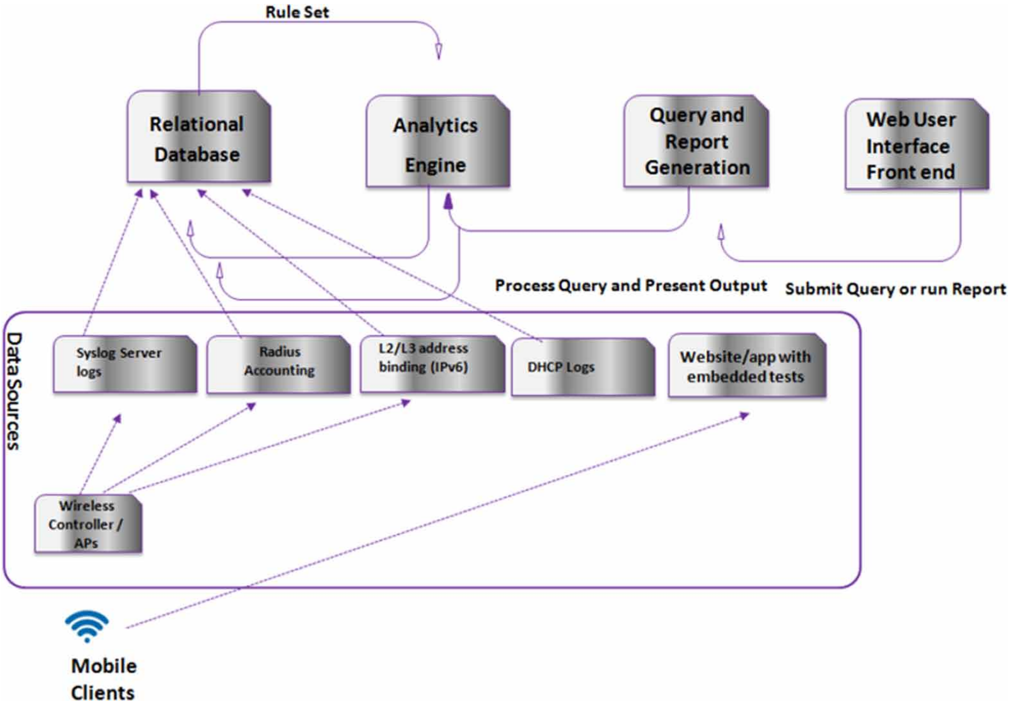
WIFIMON ARCHITECTURE

The main goal of WiFiMon is to provide a wireless performance and verification schema based on end-user behavior on eduroam enabled Wi-Fi campuses (Cao et al., 2010). This indicates that all mobile devices can be considered as raw data suppliers (crowd-sourced approach). The architecture, as depicted in Figure 1, includes four building blocks, containing the data source block, the relational database, the analytics engine, the query and report generator and finally the visualization block. Each block supports the process of collecting, storing raw data, analyzing, consuming and visualizing of reference data respectively.

The overall operation can be described as follows (see Figure 1):

- The data source block includes the performance data generated on websites with embedded JavaScript code, and additional raw data retrieved from log files such as server syslogs, Radius accounting logs and DHCP logs (Biever et al., 2017). These raw data are used for the correlation of the measurements with the AP-identifiers;
- From the data source, raw date will be fed automatically into the Relational Database (RDB), which consists of a PostgreSQL database (PostgreSQL, n.d.) for storing the data, and an InfluxDB database (InfluxDB, n.d.), for time-series support for visualization on Grafana (Grafana, n.d.) (see below);
- The Analytic Engine (AE) sorts the raw data, analyzes it, and provides visualizations using tools that offer the greatest insight on the wireless network performance. Currently, WiFiMon uses Grafana to create the measurement and monitoring dashboards;
- The Query and Report Generation (QaRG) allows a customized search and post them to Web UI;

Figure 1. WiFiMon architecture


Rule Set
Relational Database
Analytics Engine
Query and Report Generation
Web User Interface Front end
Process Query and Present Output
Submit Query or run Report
Data Sources
Syslog Server logs
Radius Accounting
L2/L3 address binding (IPv6)
DHCP Logs
Website/app with embedded tests
Wireless Controller / APs
Mobile Clients

- The Web UI (Java-based) gives real-time options for visualization in form of snapshots (aggregation of correlated data). Collected data for a specific time period, or from a specific AP are some of the options provided through the Web UI.

Measurement Procedure

In this paragraph authors describe the procedure that enables WiFiMon measurements (see Figure 2). Initially, the end-user has to connect to an eduroam enabled network (eduroam, n.d.). Eduroam is the secure, world-wide roaming access service developed for the international research and education community. To start a new eduroam session, the user has to authenticate through IEEE802.1x (EAP/TTLs) with its Identity Provider (IdP) and get authorized to campus local policy from the Service Provider (SP). Then, the DHCP server assigns the user a unique IP address. The authentication process creates new entries to DHCP and Radius logs; while these records are also stored at the RDB (Suntu et al., 2017).

Once connected to the wireless network, the end-user is required to visit a WiFiMon-enabled web page or use WiFiMon-enabled mobile applications for Android or iOS platform, i.e. a web page or a mobile application that has the WiFiMon JavaScript Library embedded (Gregori et al., 2013). The WiFiMon JavaScript Library automatically triggers performance tests, assuming that the user's IP is included in the registered subnets (see paragraph "Measures to Avoid Unnecessary/Repeated Measurements" that follows). In detail, the library uses open source JavaScript-based tools, such as NetTest[6] or Boomerang[7], in order to extract network-performance information. Such information includes download and upload throughput and RTT as experienced by the end-user. Once the tests are completed, the performance results are transferred by JSON post to the RDB, where they are correlated with the DHCP and Radius accounting raw data[8] (see pseudo-code below). The correlated datasets will be also fed into the InfluxDB to allow time series visualizations. Snapshots configured on the Grafana UI enable the visualization of the correlated datasets (Kokkinos et al., 2016)

The algorithm's code of the measurements correlation with Radius is in Table 1.

The measurements from all users are stored centrally in the same RDB. The network administrator can access the WiFiMon Web-UI to visualize the results and get an overview of the wireless network

Figure 2. Architecture flow diagram

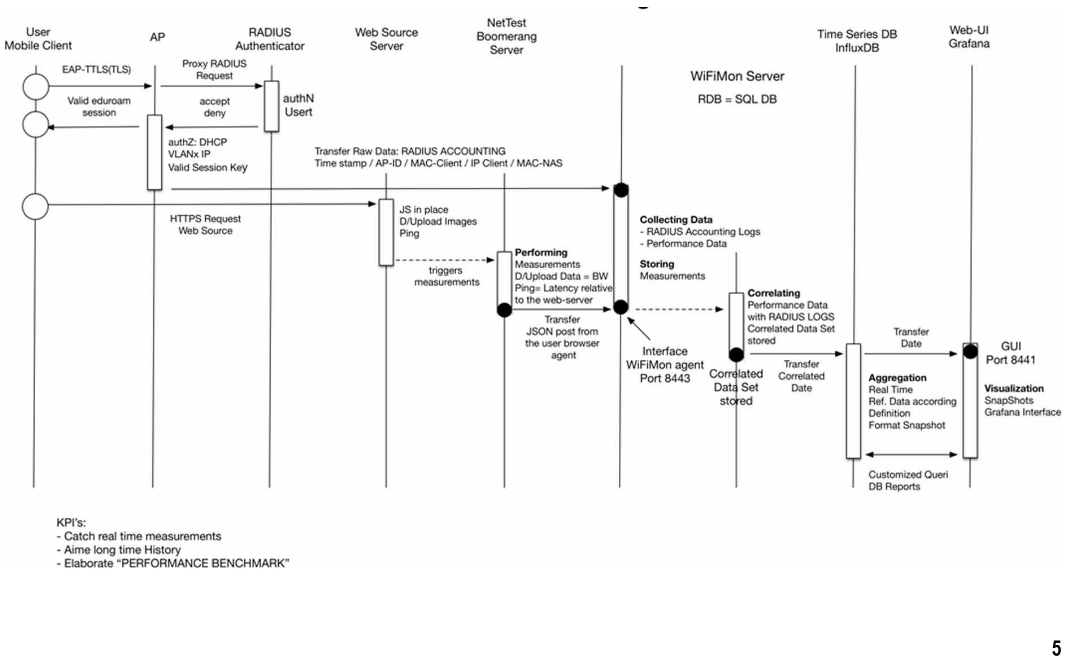


Table 1. Pseudo code for correlating measurements with Radius logs

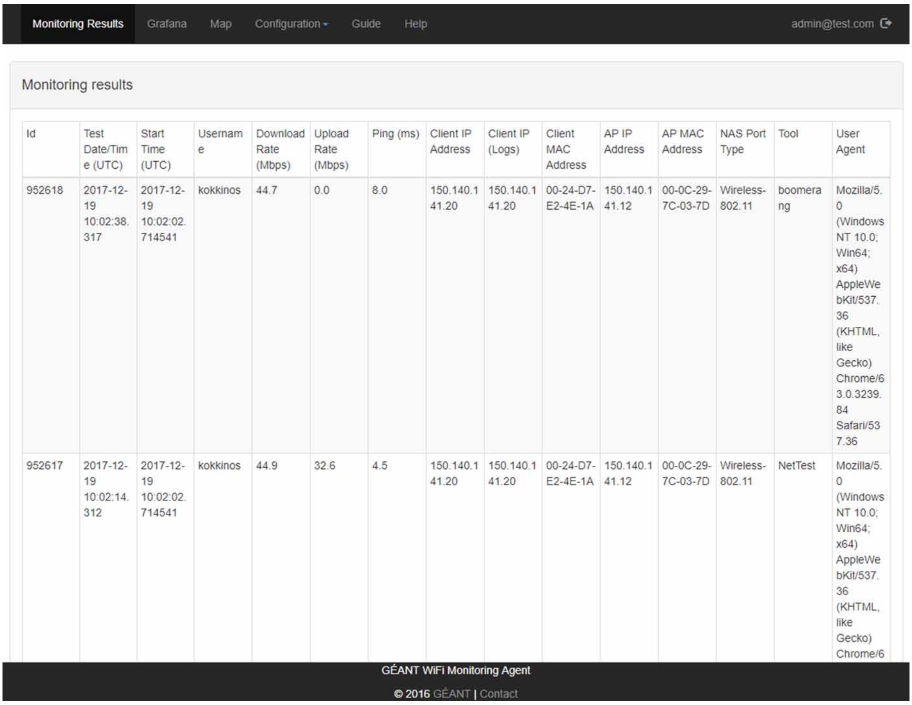
```
CHECK user_IP, timestamp // from measurements
CHECK client_IP, auth_timestamp // from Radius logs
WHILE auth_timestamp < timestamp // in descending order to
    // select the most recent entry
    IF user_IP == client_IP
        INNER JOIN measurement and Radius_entry ON IP
        BREAK
    ENDIF
ENDWHILE
```

overall performance. The Web-UI consists of a Spring Boot (Spring Boot, n.d.) web application that is using Hibernate (Hibernate, n.d.) for data query. It includes the functionalities that are necessary to enable the measurements (i.e. add/remove registered subnets), project the correlated data and allow real-time visualization options, such as measurements for a specific time period, from a specific AP, from specific IPs or users as well as results for selected wireless technologies (e.g. IEEE 802.11b, etc.). The recent measurements are also available through the Web-UI (Figure 3).

Measures to Avoid Unnecessary/Repeated Measurements

Assuming that the JavaScript code is embedded in the official website (or mobile application) of the University that has installed WiFiMon to check the performance of its wireless network, it is

Figure 3. WiFiMon recent measurements



Monitoring results

Id	Test Date/Time (UTC)	Start Time (UTC)	Username	Download Rate (Mbps)	Upload Rate (Mbps)	Ping (ms)	Client IP Address	Client IP (Logs)	Client MAC Address	AP IP Address	AP MAC Address	NAS Port Type	Tool	User Agent
952618	2017-12-19 10:02:38.317	2017-12-19 10:02:02.714541	kokkinos	44.7	0.0	8.0	150.140.141.20	150.140.141.20	00-24-D7-E2-4E-1A	150.140.141.12	00-0C-29-7C-03-7D	Wireless-802.11	boomerang	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/63.0.3239.84 Safari/537.36
952617	2017-12-19 10:02:14.312	2017-12-19 10:02:02.714541	kokkinos	44.9	32.6	4.5	150.140.141.20	150.140.141.20	00-24-D7-E2-4E-1A	150.140.141.12	00-0C-29-7C-03-7D	Wireless-802.11	NetTest	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/6

obvious that the performance tests should not be triggered by users that visit the website but are located outside the University campus. The results of these tests would be relative to their location (and wireless network performance) and therefore not useful for the Wi-Fi network under study. In addition, NetTest and boomerang download fixed size images from a server and measure the time it took to download them in order to calculate the bandwidth and RTT. This means that these tests increase the total size of the hosting websites. Users browsing experience will not change since the scripts that initiate measurements will run only after the page has loaded, but it will have a negative impact especially if a user is uses a metered internet connection. Again, preventing tests from IPs outside the registered subnets is important.

Equally important is to prevent repeated measurements for users inside the registered subnets in order to avoid network overloading. To this direction, once the measurement is initiated, authors set a cookie that prevents re-initiation of the measurements for a specific time period (configurable by the network administrator). Based on the above, the pseudo code to initiate a measurement is presented in Table 2.

Mobile App

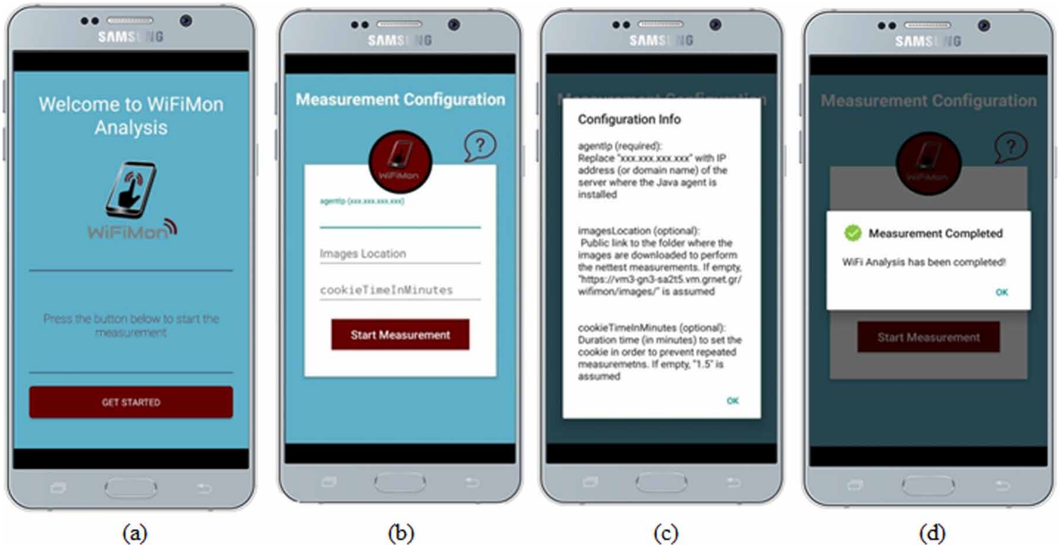
Over the last years, end-users satisfy their communication needs through the use of smart phones. Moreover, most of the communication is not carried out using browser-based functions, but rather using mobile applications. Towards this direction, our goal was the creation of a WiFiMon mobile app that on the one hand would work as a standalone application (where the user could take measurements and see the results) and on the other hand it could be integrated in external mobile applications such as University or conference apps, as cited in Faggiani et al. (2014) and Faggiani et al. (2015). Figure 4 shows an overview of the standalone WiFiMon app working on Android devices available for download[9].

As we can see the WiFiMon disposes a friendly UI ready to be used from any user, giving the opportunity to test his Wi-Fi connection in one click. In the figure above authors present the initial page of the Smartphone application in which the user is being notified about the application and its purpose (Figure 4(a)). In the next step (Figure 4(b)), user can optionally set the parameters of the measurements, such as the WiFiMon server IP address (to post the results to), the location of the images for the performance test, and the cookieTimeInMinutes to prevent repeated measurements. If the user leaves the above fields empty, the mobile application uses the default configurations to reach

Table 2. Pseudo code for performing/storing measurements

```
SET registered subnets// allow measurements only from WiFi subnet
CHECK if cookie is set for the user// avoid repeated measurements and network overloading
IF user_IP inside registered subnets
    IF cookies is not set
        GET timestamp
        CALCULATE download_throughput, upload_throughput, RTT
        GET user_IP, user_agent
        GET user_location // with Google API loader
        POST timestamp, download_throughput, upload_throughput, RTT, uster_IP, user_agent, user_location to Postgres ad InfluxDB databases
        SET cookie
    ENDIF
ENDIF
```

Figure 4. WiFiMon android app UI



Press the button below to start the measurement

Configuration Info

agentIp (required):
Replace "xxx.xxx.xxx.xxx" with IP address (or domain name) of the server where the Java agent is installed

imagesLocation (optional):
Public link to the folder where the images are downloaded to perform the nettest measurements. If empty, "https://vm3-gn3-sa2t5.vm.grnet.gr/wifimon/images/" is assumed

cookieTimeInMinutes (optional):
Duration time (in minutes) to set the cookie in order to prevent repeated measuremetns. If empty, "1.5" is assumed

Measurement Completed

WiFi Analysis has been completed!

the server and generate the measurements (as Figure 4(c) shows). With the default configurations the images are downloaded from our server in Athens and the results are posted in the same server. In addition, the default value for cookieTimeInMinutes is 1.5 minutes. With the above parameters being set up, the user can push the Start button and then the measurement starts sending data to the server. When the measurement has finished, a notification message appears as shown in Figure 4(d) (Baumann et al., 2017).

PERFORMANCE AND TESTING RESULTS

WiFiMon gives Wi-Fi network providers the opportunity to have a clear picture of the network performance behavior as experienced by the end-users. Its functionality and performance were tested in two pilot sites: at the Dublin City University (DCU) and at the TNC2015 conference which took place in Porto, Portugal. The results of the pilot operation at DCU are presented in this section.

Apart from its functionality, WiFiMon was also tested in terms of scalability (when it receives a large amount of measurements) and in terms of Web UI performance as the measurements in the two databases (PostgreSQL and InfluxDB) increases. The second part of this section presents and analyzes the stress testing results.

DCU Test Case

The Wi-Fi infrastructure at DCU consists of a pair of Motorola RFS7000 wireless controllers, with a range of Motorola-dependent APs. DCU uses FreeRADIUS (Nelson & DeKok, 2007) with an ISC DHCP server on Linux. DCU campus includes more than 800 wireless APs. The solution is implemented to use non-tunneled bridge mode, where client devices are placed on local switch VLAN after authentication based on a Radius attribute, which distinguishes categories of users. The authentication to the Wi-Fi network is through the eduroam configured service, for local users and visitors. An AP identifier is included in the Radius logs that enabled the associations of the mobile devices with the APs, allowing location-based monitoring.

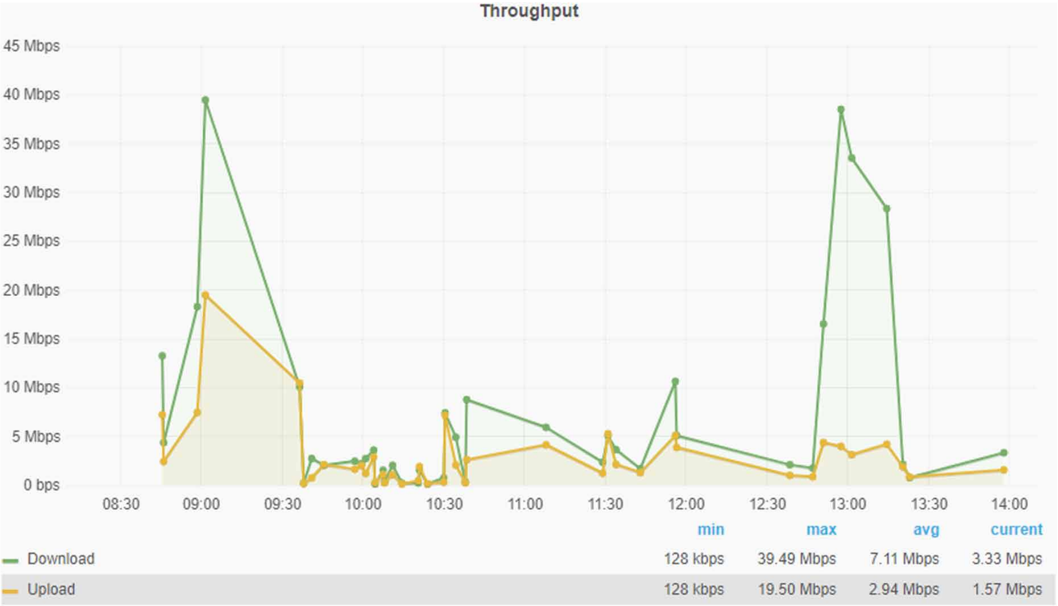
The procedure for the DCU test case was as follows: On May 26, 2017 from 08:30 until 14:00, a number of end-users-initiated performance test from different locations while roaming across the campus. As they were roaming, the end-users made handovers to APs that supported different 802.11 technologies. The performance tests were initiated by visiting a WiFiMon-enabled website, which automatically triggered NetTest measurements, i.e. fixed size images were downloaded from /uploaded to a server in order to calculate the bandwidth and RTT. The server was in Athens, Greece which means that the download/upload rates and RTT were relative to this server's location. Once the measurements were completed, the results were automatically posted to the RDB and correlated with the Radius logs information (see Figure 2), allowing the association of performance results with the individual APs. In the plots/figures below, we can see the test case results and all the dashboards that are available through the WiFiMon UI.

In Figure 5 we can see the first dashboard that presents the timeseries of download and upload throughput, as measured by the end-users. The dashboards provide general information, such as the minimum, maximum, average and current throughputs achieved. In detail, the maximum value of throughput in the case of download is 39.49Mbps, while the maximum upload throughput achieved is 19.50Mbps. The average throughput is 7.11Mbps for the download and 2.94Mbps for the upload; while the minimum values are 128kbps for both cases. From Figure 5 it is clear that the download and upload rates show a big fluctuation range. These variations may have occurred for example due to temporary problems in the network during the measurement process, or because of temporary poor Wi-Fi connectivity of end-users (e.g. if they were located far from the serving AP).

Apart from the download/upload rates, we can see that there are great variations in the RTT timeseries (Figure 6). The minimum RTT achieved is 32ms and the maximum 171ms; however, these values are reasonable since the RTT is measured from Dublin, Ireland to Athens, Greece. In addition, we can see that during the period 09:30 until 10:30 there are many measurements with high RTT values. During this period, the download and upload throughputs achieved were low, indicating poor network performance or temporary network problems, or even connection of end-users to APs that support technologies with lower maximum speeds (e.g. 300Mbps IEEE 802.11an, 130Mbps IEEE 802.11bgn).

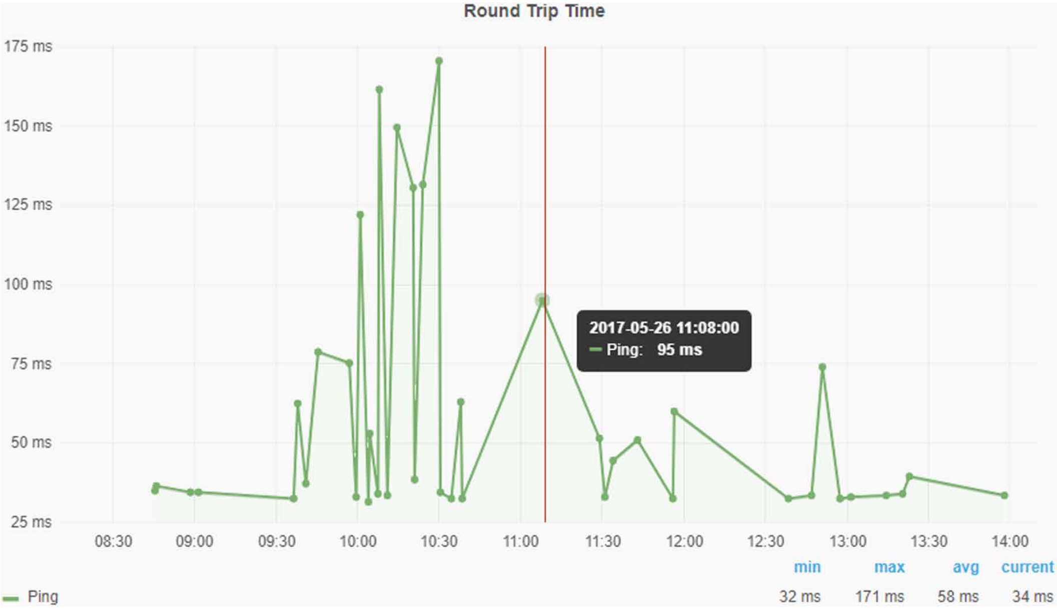
The correlation of the measurement results with the Radius logs offers additional insights. Figure 7 and Figure 8 present the download throughput results per AP MAC address and technology

Figure 5. Download and upload throughput



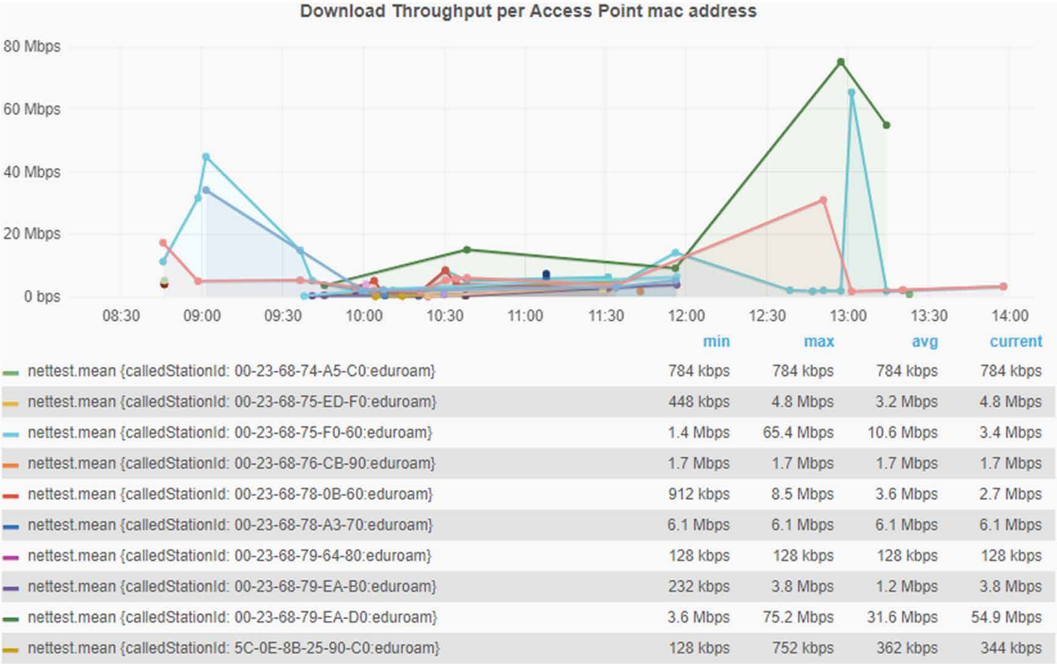
	min	max	avg	current
Download	128 kbps	39.49 Mbps	7.11 Mbps	3.33 Mbps
Upload	128 kbps	19.50 Mbps	2.94 Mbps	1.57 Mbps

Figure 6. Round trip time



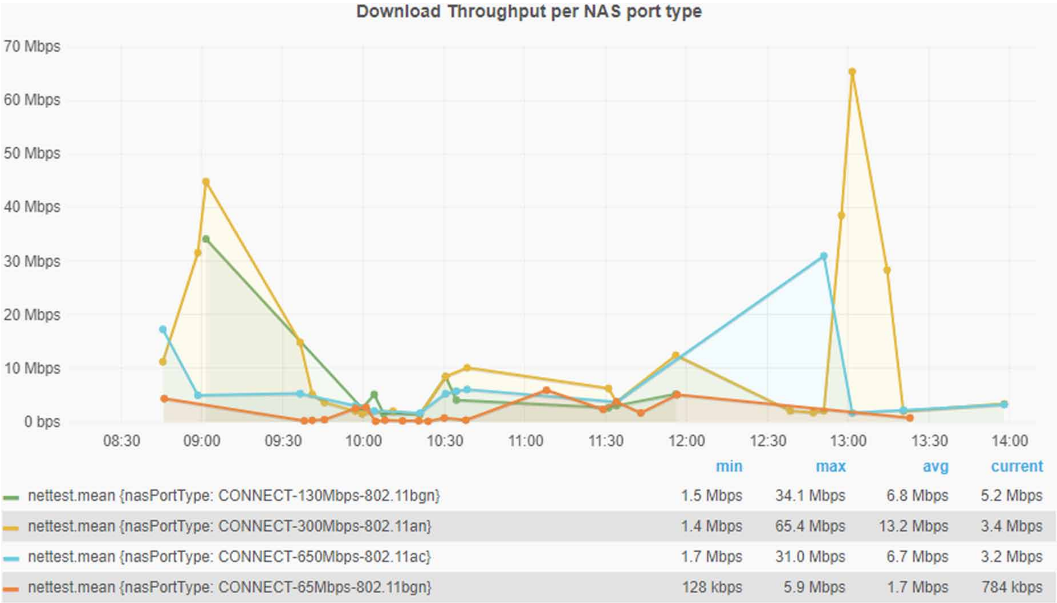
	min	max	avg	current
▬ Ping	32 ms	171 ms	58 ms	34 ms

Figure 7. Download throughput per AP MAC



	min	max	avg	current
▬ nettest.mean {calledStationId: 00-23-68-74-A5-C0:eduroam}	784 kbps	784 kbps	784 kbps	784 kbps
▬ nettest.mean {calledStationId: 00-23-68-75-ED-F0:eduroam}	448 kbps	4.8 Mbps	3.2 Mbps	4.8 Mbps
▬ nettest.mean {calledStationId: 00-23-68-75-F0-60:eduroam}	1.4 Mbps	65.4 Mbps	10.6 Mbps	3.4 Mbps
▬ nettest.mean {calledStationId: 00-23-68-76-CB-90:eduroam}	1.7 Mbps	1.7 Mbps	1.7 Mbps	1.7 Mbps
▬ nettest.mean {calledStationId: 00-23-68-78-0B-60:eduroam}	912 kbps	8.5 Mbps	3.6 Mbps	2.7 Mbps
▬ nettest.mean {calledStationId: 00-23-68-78-A3-70:eduroam}	6.1 Mbps	6.1 Mbps	6.1 Mbps	6.1 Mbps
▬ nettest.mean {calledStationId: 00-23-68-79-64-80:eduroam}	128 kbps	128 kbps	128 kbps	128 kbps
▬ nettest.mean {calledStationId: 00-23-68-79-EA-B0:eduroam}	232 kbps	3.8 Mbps	1.2 Mbps	3.8 Mbps
▬ nettest.mean {calledStationId: 00-23-68-79-EA-D0:eduroam}	3.6 Mbps	75.2 Mbps	31.6 Mbps	54.9 Mbps
▬ nettest.mean {calledStationId: 5C-0E-8B-25-90-C0:eduroam}	128 kbps	752 kbps	362 kbps	344 kbps

Figure 8. Download throughput per NAS port type



	min	max	avg	current
🟩 nettest.mean {nasPortType: CONNECT-130Mbps-802.11bgn}	1.5 Mbps	34.1 Mbps	6.8 Mbps	5.2 Mbps
🟨 nettest.mean {nasPortType: CONNECT-300Mbps-802.11an}	1.4 Mbps	65.4 Mbps	13.2 Mbps	3.4 Mbps
🟦 nettest.mean {nasPortType: CONNECT-650Mbps-802.11ac}	1.7 Mbps	31.0 Mbps	6.7 Mbps	3.2 Mbps
🟧 nettest.mean {nasPortType: CONNECT-65Mbps-802.11bgn}	128 kbps	5.9 Mbps	1.7 Mbps	784 kbps

respectively. From these figures, we can see that WiFiMon provides important information to network administrators regarding the performance of individual APs. Maximum, minimum, average and current speeds for each AP are calculated and together with the information of each AP technology allows the network administrator to evaluate its performance (Lakhina et al., 2004). Specifically, for the DCU case, from Figure 8 we can see that download throughput results from APs that support 802.11an standard are higher compared to the ones initiated from APs that support 802.11bgn standard.

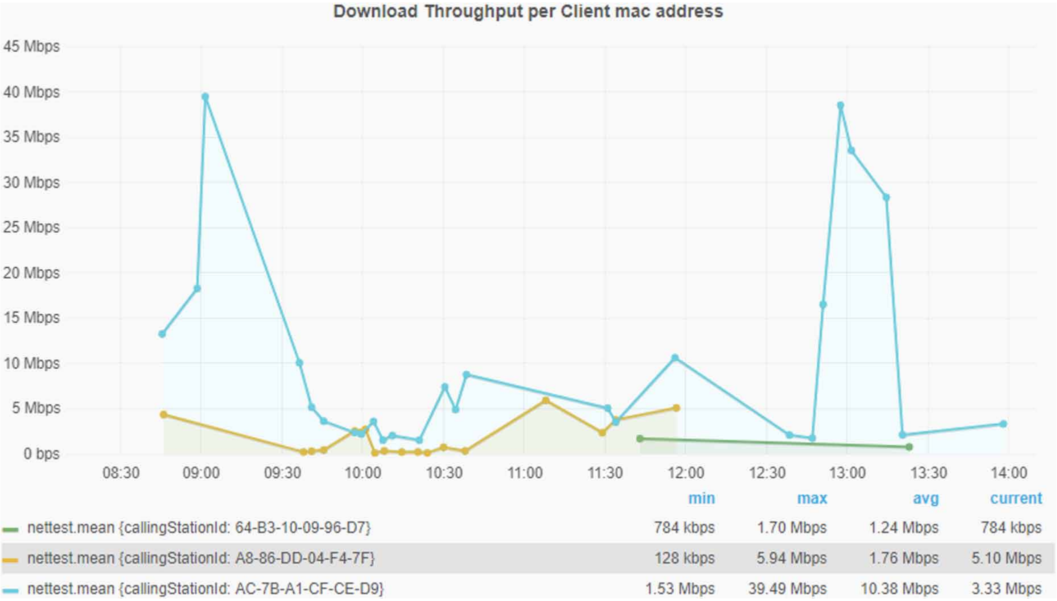
In addition, the Web UI includes dashboards that evaluate the performance of individual users' devices (MAC address), operating system (OS) and browser (Figure 9, Figure 10 and Figure 11 respectively). The information about the client's device MAC address in included in the Radius logs while the OS and browser are extracted from the User-Agent string in the HTTP header. Such dashboards are extremely useful in order to check whether possible poor results are due to AP's or individual devices/OS/browser's poor performance. The former shows that and improved and more efficient APs installation should take place (e.g. update to newest IEEE802.11x standards); while the latter is something out of the administrator's score.

Stress Testing Results

On the 13th and 14th of July 2017, authors initiated a series of tests in order to 1. examine the scalability of the WiFiMon agent when it receives a large amount of measurements, and 2 examine the performance of WiFiMon GUI as the measurements in the two databases used in WiFiMon increases. To this direction, our research team generated a real-time stream of JSON data using the JSON data generator (n.d.). The specifications and the usage of the virtual machines (VMs) that used for the stress testing can be found in Table 3.

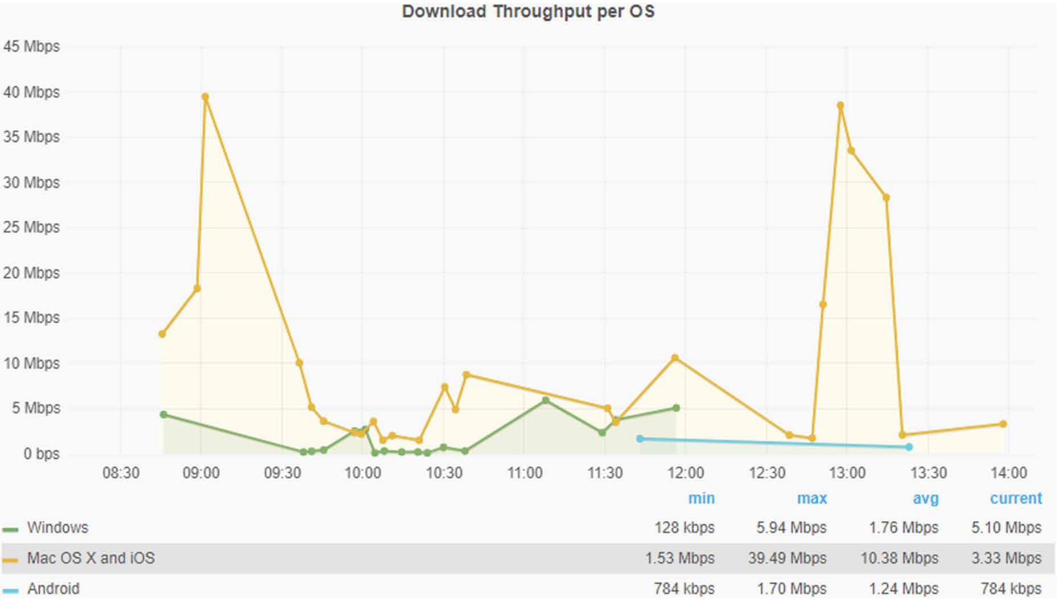
The JSON Data Generator in each VM was configured to perform a sequence of 10 measurements (about 1 per 250 ms), then paused for about 1 second before repeating a new sequence of measurements. These settings generated about 150-200 JSON requests/measurements per minute per VM. The measurements from all VMs were posted in the same WiFiMon instance (vm1), i.e. the specific WiFiMon instance was receiving about 450-600 measurements per minute. The measurements were

Figure 9. Download throughput per Client MAC



	min	max	avg	current
🟩 nettest.mean {callingStationId: 64-B3-10-09-96-D7}	784 kbps	1.70 Mbps	1.24 Mbps	784 kbps
🟨 nettest.mean {callingStationId: A8-86-DD-04-F4-7F}	128 kbps	5.94 Mbps	1.76 Mbps	5.10 Mbps
🟦 nettest.mean {callingStationId: AC-7B-A1-CF-CE-D9}	1.53 Mbps	39.49 Mbps	10.38 Mbps	3.33 Mbps

Figure 10. Download throughput per OS

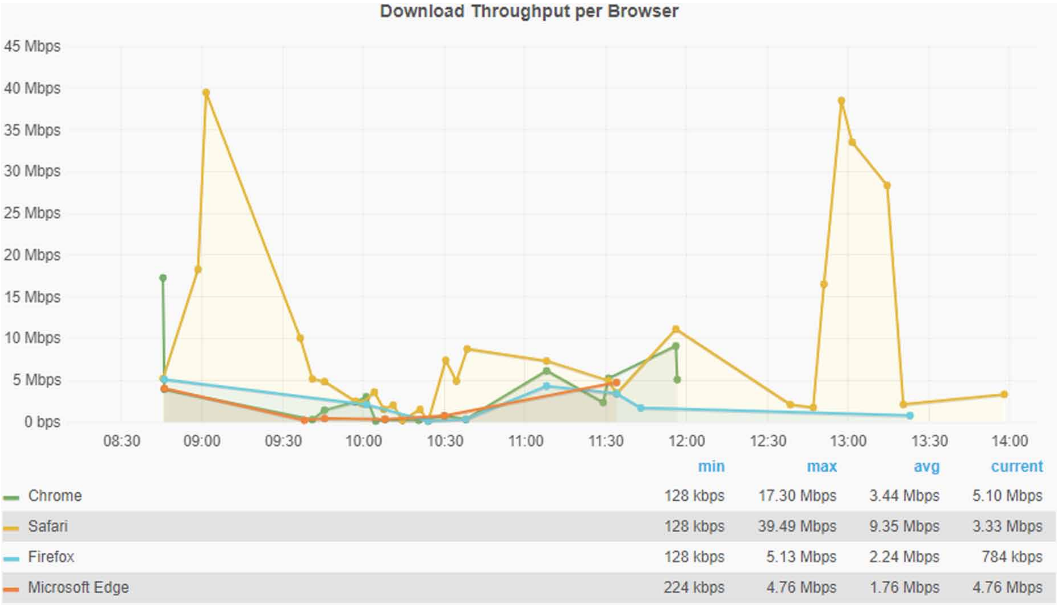


	min	max	avg	current
🟩 Windows	128 kbps	5.94 Mbps	1.76 Mbps	5.10 Mbps
🟨 Mac OS X and iOS	1.53 Mbps	39.49 Mbps	10.38 Mbps	3.33 Mbps
🟦 Android	784 kbps	1.70 Mbps	1.24 Mbps	784 kbps

also correlated with Radius logs to increase the complexity. Samples of Radius logs were manually inserted to the PostgreSQL to this direction.

The simulation lasted for 24 hours. Every 30 minutes, authors captured the count of measurements and the used disk space of vm1 that hosted the WiFiMon instance and databases. Figure 12(a) presents the number of average received measurements per minute and Figure 12(b) the disk space increase in vm1 as recorded from our team every 30 minutes. In addition, authors measured the time required to

Figure 11. Download throughput per browser

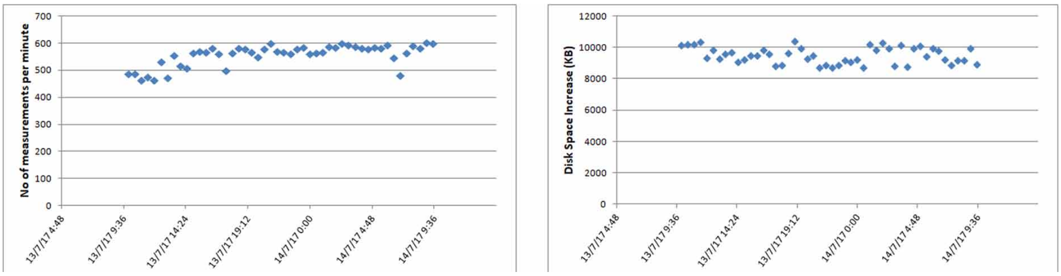


	min	max	avg	current
Chrome	128 kbps	17.30 Mbps	3.44 Mbps	5.10 Mbps
Safari	128 kbps	39.49 Mbps	9.35 Mbps	3.33 Mbps
Firefox	128 kbps	5.13 Mbps	2.24 Mbps	784 kbps
Microsoft Edge	224 kbps	4.76 Mbps	1.76 Mbps	4.76 Mbps

Table 3. Simulation Specification

Virtual Machine	Usage	RAM	CPU	Hard Disk	Operating System
vm1	WiFiMon instance	2GB	1	9.8 GB HD	debian-squeeze
vm2	JSON Data Generator	2GB	1	9.8 GB HD	Ubuntu (12.04)
vm3	JSON Data Generator	2GB	1	9.8 GB HD	Ubuntu-trusty (14.04)
vm4	JSON Data Generator	2GB	1	9.8 GB HD	debian-squeeze

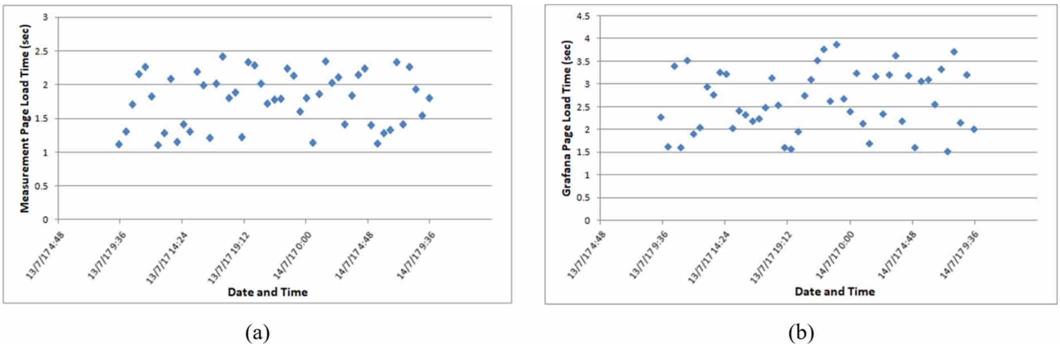
Figure 12. (a) Received measurements per minute; and (b) Disk space increase (recorded every 30 minutes)



load two pages of the WiFiMon Web UI, the measurement page and the Grafana page that includes the time series dashboards (Figure 13(a) and Figure 13(b) respectively).

Overall, our research team simulated more than 800000 correlated measurements during the 24 hours period. Even for this large (and dense) amount of measurements, the smooth and robust performance of WiFiMon was not affected as shown in the two figures above. In normal environments (such as university campuses that include many APs), a realistic amount of measurements would be

Figure 13. WiFiMon Web UI: Load time for: (a) Measurement; and (b) Grafana pages (recorded every 30 minutes)



(a) (b)

around 500 measurements per hour. Therefore, under normal conditions and according to the stress testing results, the WiFiMon tool can provide a robust and efficient solution to measure (and visualize) the quality of a wireless network as experienced by the end-users.

CONCLUSION

In this manuscript authors proposed and presented WiFiMon, a tool that may measure the quality of a wireless network as experienced by end-users. The tool mainly targets at network administrator in order to provide them a means to visualize and evaluate the performance of their wireless network and detect anomalies in real time. The main feature of the above research study is that, in the case of WiFiMon, measurement results, unlike similar tools, are stored in a database controlled by the network administrator, who is the principal responsible for taking steps to improve it. WiFiMon is based on crowd sourcing (Chen et al., 2010), i.e. the end-users' feedback, since the measurements are initiated automatically by the end-users (without their intervention) after they visit a WiFiMon-enabled web page or use a WiFiMon-enabled mobile application. To this direction authors deployed and presented a standalone Android mobile application that enables and stores performance measurements. This mobile app concept may also be adapted as a "generic" library to external mobile applications. The above research team also discussed additional measures that WiFiMon is using to avoid unnecessary network overload, such as limiting the performance measurements to defined IP subnets, and preventing the duplication of measurements with the aid of cookies.

In order ascertain the feasibility of WiFiMon; authors tested its operation at the Dublin City University, Ireland. The expertise gained from this test showed that it is feasible to measure and record performance data on an eduroam enabled campus and to correlate these measurements with log files in order to retrieve information regarding the performance of individual APs, users' devices, browsers and operation systems. Finally, authors conducted stress testing in order to examine how WiFiMon behaves under extreme circumstances. The results of the stress testing revealed a robust and uninterrupted operation even in cases where WiFiMon receives a large and dense amount of measurements.

FUTURE WORK

Having examined the feasibility and performance of WiFiMon, our future work targets are four additional directions: 1. privacy issues, 2. updates to increase usability and visualization options, 3. validation of JavaScript measurements with hardware probes measurements, and 4. prediction of performance degradation.

Privacy Issues

Even though the results from the performance test (i.e. download/upload throughput and ping results) are reasonably considered as non-sensitive information, the fact that the performance test are correlated with log files that may contain sensitive data (such as Radius logs) should be handled with the utmost care. Everything should be in accordance with the campus policies and legal obligations. Our intention is to update WiFiMon to this direction and allow additional options such as showing/hiding or even storing/ignoring user-related data.

Updates

The current version of WiFiMon utilizes influxDB and Grafana as a means to obtain time series for the visualization and monitoring of the wireless network. Our team aims to examine similar tools, such as the Elasticsearch and Kibana stack, that could serve as alternatives to influxDB and Grafana. Our intention is to select the most efficient tool in terms of usability and visualization possibilities that of each tool provides.

Measurements Validation

To calculate download and upload throughput and RTT as experienced by the end-users, WiFiMon uses open source JavaScript-based tools such as NetTest and boomerang. Since the measurements may depend on the users' position and device status, it is necessary to verify whether the measurements obtained through JavaScript are accurate enough. To verify their accuracy, our goal is to set up a hybrid environment, consisting of WiFiMon and hardware probes in different rooms of a university campus or a conference venue (Chen et al., 2012). This way we could compare the static (hardware-probe based) objective measurements with the opportunistic measurements for end-users' devices and examine whether they follow the same trends.

Performance Prediction

Prediction of performance behavior (mainly of performance degradation) will be achieved through recording of long-term histories and making performance benchmarking. Existing work in performance predictions shall be evaluated, and a machine learning approach for mobile-client and APs behavior will be deployed. The visualization of such predictions will further facilitate the decision-making process of the network administrators during network installation.

REFERENCES

Baumann, K., Bouras, C., Kokkinos, V., Papachristos, N., & Stamos, K. (2017). WiFiMon app measuring Wi-Fi performance as experienced by end-users. *Proceedings of the 24th International Conference on Telecommunications (ICT)*, Limassol, Cyprus. Academic Press. doi:10.1109/ICT.2017.7998257

Biever, R., Francis, C., Francis, S., & McCahill, M. (2017). A Cohesive Monitoring Strategy for Campus Wireless Networks. *Presentation at Duke University*. Retrieved from https://meetings.internet2.edu/media/medialibrary/2017/04/21/20170426-biever-cohesive-monitoring-strategy-campus-wireless-networks.pdf

Cao, J., Tu, G., Liang, L., & Jiang, M. (2010). Design of Power Grid Environment Monitoring System Based on WLAN. *Proceedings of the 2010 Second World Congress on Software Engineering*. Wuhan, China (pp. 30-33). Academic Press.

Chen, K., Chang, C., Wu, C., Chang, Y., & Lei, C. (2010). Quadrant of euphoria: A crowdsourcing platform for QoE assessment. *IEEE Network*, *24*(2), 28–35. doi:10.1109/MNET.2010.5430141

Chen, X., Jin, R., Suh, K., Wang, B., & Wei, W. (2012). Network performance of smart mobile handhelds in a university campus WiFi network. *Proceedings of the 2012 Internet Measurement Conference (IMC '12)*. New York, NY (pp. 315-328). Academic Press. doi:10.1145/2398776.2398809

Choffnes, D. R., Bustamante, F. E., & Ge, Z. (2011). Crowd-sourcing Service-Level Network Event Monitoring. *SIG-COMM Comp. Communication Review*, *41*(4), 387–398.

Cooke, E., Mortier, R., Donnelly, A., Barham, P., & Isaacs, R. (2006). Reclaiming network-wide visibility using ubiquitous end system monitors. *Proceedings of the annual conference on USENIX '06 Annual Technical Conference (ATEC '06)*. Berkeley, CA (pp. 32-32).

Faggiani, A., Gregori, E., Lenzini, L., & Luconi, V. (2015). Measuring the Internet Topology with Smartphones. *Proceedings of the 2015 ACM SIGCOMM Workshop on Crowdsourcing and Crowdsharing of Big Data*. London, UK (pp. 45–50). ACM. doi:10.1145/2787394.2787398

Faggiani, A., Gregori, E., Lenzini, L., Luconi, V., & Vecchio, A. (2014). Smartphone-based crowdsourcing for network monitoring: Opportunities, challenges, and a case study. *IEEE Communications Magazine*, *52*(1), 106–113. doi:10.1109/MCOM.2014.6710071

GEANT Wiki. (n.d.). How to deploy eduroam on-site or on campus. Retrieved from https://wiki.geant.org/display/H2eduroam/How+to+deploy+eduroam+on-site+or+on+campus.

Gember, A., Anand, A., & Akella, A. (2011). A comparative study of handheld and non-handheld traffic in campus Wi-Fi networks. *Proceedings of the 12th international conference on Passive and active measurement (PAM'11)*. Berlin, Germany (pp. 173-183). Academic Press. doi:10.1007/978-3-642-19260-9_18

Grafana. (n.d.). *Grafana* [computer software]. Retrieved from at: http://grafana.org/

Gregori, E., Lenzini, L., Luconi, V., & Vecchio, A. (2013). Sensing the Internet through crowdsourcing. *Proceedings of the Second IEEE PerCom Workshop on the Impact of Human Mobility in Pervasive Systems and Applications (PerMoby)*. San Diego, CA (pp. 248-254). IEEE Press.

Hernandez-Campos, F., & Papadopouli, M. (2005). A comparative measurement study the workload of wireless access points in campus networks. *Proceedings of the IEEE 16th International Symposium on Personal, Indoor and Mobile Radio Communications*. Berlin, Germany (pp. 1776-1780). IEEE Press. doi:10.1109/PIMRC.2005.1651747

Hibernate. (n.d.). *Hibernate* [computer software]. Retrieved from: http://hibernate.org/

Huadong, M., Dong, Z., & Peiyan, Y. (2014). Opportunities in Mobile Crowd Sensing. *IEEE ComSoc Magazine*, *52*(8), 29–35. doi:10.1109/MCOM.2014.6871666

InfluxDB. (n.d.). InfluxDB [computer software]. Retrieved from: https://influxdata.com/timeseries-platform/influxdb/

JSON data generation. (n.d.). JSON data generation [computer software]. Retrieved from: https://github.com/acesinc/json-data-generator

Kokkinos, V., Stamos, K., Kanakis, N., Baumann, K., Wilson, A., & Healy, J. (2016). Wireless Crowdsourced Performance Monitoring and Verification - WiFi Performance Measurement Using End-User Mobile Device Feedback. *Proceedings of the 8th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT '16)*, Lisbon, Portugal (pp. 432-437). Academic Press. doi:10.1109/ICUMT.2016.7765398

Lakhina, A., Crovella, M., & Diot, C. (2004). Diagnosing network-wide traffic anomalies. *Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM '04)*. Portland, OR (pp. 219-230). Academic Press.

Le, A., Varmarken, J., Langhoff, S., Shuba, A., Gjoka, M., & Markopoulou, A. (2015). AntMonitor: A System for Monitoring from Mobile Devices. In *Proceedings of the 2015 ACM SIGCOMM Workshop on Crowdsourcing and Crowdsharing of Big (Internet) Data (C2B(1)D '15)*. New York, NY, USA, 15-20. doi:10.1145/2787394.2787396

Nelson, D., & DeKok, A. (2007). Common Remote Authentication Dial In User Service (RADIUS) Implementation Issues and Suggested Fixes. Retrieved from: https://tools.ietf.org/html/rfc5080

Nielsen. (2015). So many apps, so much more time for entertainment. Retrieved from http://www.nielsen.com/us/en/insights/news/2015/so-many-apps-somuch-more-time-for-entertainment.html

Postgre S.Q.L. (n.d.). PostgreSQL [computer software]. Retrieved from https://www.postgresql.org/

Saeed, A., Naseem, S. F., & Zaidi, Z. R. (2007). Mobility estimation for wireless networks using round trip time (RTT). *Proceedings of the 6th International Conference on Information, Communications & Signal Processing*. Singapore, Singapore (pp. 1-5). Academic Press. doi:10.1109/ICICS.2007.4449699

SangYoung., P., Hyo-Sung, A., & Wonpil, Y. (2007). Round-trip time-based wireless positioning without time synchronization. *Proceedings of the International Conference on Control, Automation and Systems*. Seoul, South Korea (pp. 2323-2326). Academic Press.

Spring Boot. (n.d.). Spring Boot [computer software]. Retrieved from https://spring.io/guides/gs/spring-boot/

Suntu, S., Odongo, N., Chege, S., & Bishoge, O. (2017). Robust Secured Roaming in Wireless Local Area Networks. *Journal of Wireless Networks and Broadband Technologies*, *6*(2), 26–42. doi:10.4018/IJWNBT.2017070102

ENDNOTES

[1] The research leading to the results presented in this manuscript has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement No. 731122 (GN4-2)

[2] Available from: http://www.arubanetworks.com/products/networking/management/airwave/

[3] Documentation on: http://www.cisco.com/en/US/prod/collateral/wireless/ps5755/ps6301/ps6305/product_data_sheet0900aecd802570d0.pdf

[4] Available from: http://www.wireshark.org

[5] "Speedtest by Ookla". Available from: http://www.speedtest.net/

[6] "Google code archive". Available from: https://code.google.com/archive/p/nettest/

[7] Available from: https://github.com/SOASTA/boomerang

[8] If Radius accounting logs is not available, WiFiMon can still record the measurements but the correlation with AP information is not visible

[9] Available for download: http://ru6.cti.gr/ru6/system/files/tools_and_mechanisms/WiFiMon_Mobile_App.zip

Christos Bouras is a Professor in the University of Patras, Department of Computer Engineering and Informatics. He is also a scientific advisor of Research Unit 6 in Computer Technology Institute and Press - Diophantus, Patras, Greece. His research interests include Analysis of Performance of Networking and Computer Systems, Computer Networks and Protocols, Mobile and Wireless Communications, Telematics and New Services, QoS and Pricing for Networks and Services, e-learning, Networked Virtual Environments and WWW Issues. He has extended professional experience in Design and Analysis of Networks, Protocols, Telematics and New Services. He has published more than 400 papers in various well-known refereed books, conferences and journals. He is a co-author of 9 books in Greek and editor of 1 in English. He has been member of editorial board for international journals and PC member and referee in various international journals and conferences. He has participated in R&D projects.

Kurt Baumann holds a master's degree in mathematics of the University Zurich (2001). From 2001 to 2005 he worked for IBM as a Security Officer and Customer Engineer. In late 2005 he joined SWITCH in a position of a Project leader for SWITCH connect. Today he is member of the Computing Network Department with focus on Research and Development projects (IoT 5G, SDN) and represents SWITCH's interests in various tasks in GEANT, currently as a task leader GN4-2/SA3T5 "eduPERT" and SC member of SIG-PMV. Further he was a co-founder of the IPv6 Council CH, introduced the SDN Switzerland Meet Up Workshop focusing research, academic ICTs and the Industry.

Vasileios Kokkinos was born in Ioannina, Greece in 1981. He obtained his diploma from the Physics Department of the University of Patras on October 2003. Next, he was accepted in the postgraduate program "Electronics and Information Processing" in the same department and on March 2006 he obtained his Master Degree. In 2010 he received his Ph.D. on Power Control in Mobile Telecommunication Networks from the Computer Engineering and Informatics Department. His research interests include data networks, third and fourth generation mobile telecommunications networks, multicast routing and group management and radio resource management. He has published several research papers in various well-known refereed conferences and articles in scientific journals.

Nikolaos Papachristos was born in 1993 in Patras, Greece. He entered the Department of Information and Communication System Engineering, University of the Aegean in 2011 and obtained his diploma in 2016. After that he obtained his master's degree in "Computer Science and Technology" in Computer Engineering and Informatics Department (University of Patras) in 2018. Currently, he is a PhD student in the same department in the area of IoT using LoraWan & NB-IoT. Nikolas joined RU6 in 2015 and he is also interested in mobile telecommunications networks, mobile software development and D2D communications. He has obtained the Michigan Proficiency in English and the Zertifikat diploma in German. He has participated in many activities at the department of his studies as an IEEE Member, has taken part in IEEE's online contests and has participated in many online courses related to mobile software development as well as telecommunications networks for further education.

Kostas Stamos was born in Patras, Greece in 1978. He has received his Diploma, Master Degree and PhD from the Computer Engineering and Informatics Department at the University of Patras. He has worked for the University of Patras, Research Unit 6 of CTI, for the Technical Educational Institute of Patras and GRNET, the Greek research network. His research interests include network applications, SDN, QoS, and bandwidth on demand.