Separating Authentication, Access and Accounting: A Case Study with OpenWiFi

Kok-Kiong Yap, Yiannis Yiakoumis, Masayoshi Kobayashi, Sachin Katti, Guru Parulkar, and Nick McKeown

Stanford University
NEC

OPENFLOW-TR-2011-1

Abstract:

Guest WiFi systems has become an integral part of our lives. Nonetheless, guest WiFi systems continue to be plagued by issues that are largely unresolved. In this technical report, we argue for the separation of authentication, access and accounting. To understand our proposal and understand how it can help to open up wireless access, we prototyped the OpenWiFi system. We describe OpenWiFi, its uses, implementation and the numerous challenges we faced. Finally, we outline trials we have undertaken with OpenWiFi to share our experiences.

Initial Posting : July 2011
Last Updated : September 2011

Separating Authentication, Access and Accounting: A Case Study with OpenWiFi

Kok-Kiong Yap[*] Yiannis Yiakoumis[*] Masayoshi Kobayashi[†]
Sachin Katti Guru Parulkar Nick McKeown
Stanford University NEC[†]
{yapkke,yiannisy,mkobaya1,skatti,parulkar,nickm}@stanford.edu

ABSTRACT

Guest WiFi systems has become an integral part of our lives. Nonetheless, guest WiFi systems continue to be plagued by issues that are largely unresolved. In this technical report, we argue for the separation of authentication, access and accounting. To understand our proposal and understand how it can help to open up wireless access, we prototyped the OpenWiFi system. We describe OpenWiFi, its uses, implementation and the numerous challenges we faced. Finally, we outline trials we have undertaken with OpenWiFi to share our experiences.

1. MOTIVATION

Guest WiFi services are everywhere and an integral part of our lives. Guest WiFi (as opposed to home or workplace WiFi access) are provided as (1) a complimentary service in airports, hotels, cafes, restaurants, libraries, etc.; (2) a subscription-based service, e.g., FON, Boingo, AT&T, etc.; or (3) part of the infrastructure supporting specific events such as workshops, conferences, etc. Such open wireless networks has become an integral part of our lives keeping us more connected than ever—an important aspect that should be preserved as EFF argued [5].

From the users' perspectives, there are three common issues with guest WiFi services—slow discovery, identification of guest WiFi and non-uniform authentication. The first issue is that WiFi has a slow discovery mechanism that sweeps through 11 to 14 frequencies to discover the available networks, incurring significant delay. Therefore, this limits the use of guest WiFi to fairly static applications. The second issue is of identification of guest WiFi. There is no standardized way to indicate that a particular SSID is providing a guest WiFi, forcing users to guess and try the possibilities one at a time. We focus on the last issue which concerns non-uniform authentication because discovery and identification are receiving significant attention at standard bodies like 802.11u

Our focus in this technical report is the issue of non-uniform authentication, or responsibility delegation from the perspective of guest WiFi providers. It is understandable that providers of guest WiFi might be concerned of DMCA takedown notices accusing them of downloading illegal content. To avoid bearing responsibilities for the users, providers have to authenticate the users and monitor their traffic. This creates significant complexity in the system—making such system out of reach of many. Moreover, this burdens the user with many logins, one for each WiFi network.

To provide an efficient system for authentication or responsibility delegation in guest WiFi, we propose decoupling authentication, access and accounting. Most (if not all) of today's implementation of guest WiFi services couples these functionalities. The most common setup is exemplified by ChilliSpot [1] that uses WiFi access points to provide access, a RADIUS server for authentication and custom software for accounting. Such arrangement is hard on everyone—network owners bears substantial cost and/or complexity [6]; while users have to remember many different passwords and account names.

By decoupling authentication, access and accounting, we have separated the task at hand into three distinct notions—authentication to establish identity; access for transferring data; and accounting to delegate responsibility. There are many benefits to this separation:

1. Such a separation allows for an efficient guest WiFi services because access can be independently provided. This means access can be provided by a home owner, a restaurant, public library, enterprises or a paid service like Boingo—enabling the coverage to be expanded without restrictions of physical deployment.

2. The separation allows for accounting and authentication to be out-sourced to third parties, thus opens up the possibility of building a guest WiFi service that rides on the coattails of other deployments. By outsourcing accounting and authentication, network owners can lower the cost and complexity they face to provide guest WiFi. This will allows and incentivize many others (including normal home owners) to provide guest WiFi. One can easily imagine how this allows a guest WiFi deployment to ride on the coattails of deployments in homes, cafes, airports, etc.

3. Since the authentication can be separated from access and accounting, the guest WiFi service can provide a consistent interface to users making it much more

The initial idea of OpenWiFi was described to the authors by Larry Alder from Google, which served as the inspiration for this work. We also thank David Bird (of Coova) for his comments.

[*] The content of this technical paper is contributed by all of the above authors. However, this article has been written by the authors marked, hence they assume sole responsibility for its content and how it is presented.

user friendly. Definitely, there would be many different guest WiFi providers, e.g., FON, AT&T, Boingo, but they can all use similar authentication mechanisms such as OAuth or OpenID. In our prototype, we demonstrate how we have used open authentication services such OpenID and Facebook Connect to authenticate users.

We contend that the separation of authentication, access and accounting would not only present the users with a familiar and convenient authentication mechanism, but also has the potential of providing a simple mechanism with which small network owners—such as home owners, cafe and restaurant owners, libraries, etc.—can provide guest WiF. This in turn facilitates the widespread availability of WiFi access. Motivated by the prospects of decoupling of authentication, access and accounting, we build a prototype and experimented with it. We also share our experience of our prototypic system here.

2. BACKGROUND AND RELATED WORK

Typical guest WiFi systems consists of the following three components:

1. *Access:* To reach the Internet, this guest WiFi services require a connection to the Internet. This connection can be dedicated to the guest WiFi service or shared with other purposes, e.g., a cafe might use the same connection for its point of sales system.

2. *Authentication:* To control access to the network and thus Internet, a guest WiFi need to implement authentication. Authentication pertains to verifying the identity of the users and hence allow responsibility to be traced back to the appropriate party. Such a service is not only needed for network systems but also many online services for the same reason. Unsurprisingly, many online websites have figured out how to collaborate and out-source this task of authentication—a trend we exploit to make guest WiFi easier and better.

3. *Accounting:* Finally once the identity of the user is established, the user can freely used the access, subjected to whatever constraints deemed appropriate by the network owner. Now, the traffic of the user would be monitored and logged for purposes of responsibility delegation and/or billing.

With these components in mind, we understand how the systems today implement these three aspects of a guest WiFi system.

2.1 ChilliSpot/CoovaChilli

ChilliSpot [1] and CoovaChilli [2] are an open-source implementations of guest WiFi based on using a RADIUS server for authentication and the accounting is done using custom software (Fig. 1). These components are typically run by a single party who has to setup the entire configuration.

2.2 Fon.com

Fon.com is a community-based company. A user/member buys and installs a FON hotspot at his home which provides two SSIDs, one private and one for public FON use—where the guest WiFi is on a different local subnet and TCP traffic are throttled down to 3 Mbps. As a member, he can then use all FON hotspots around the world. Others can buy FON credit to use available FON hotspots.

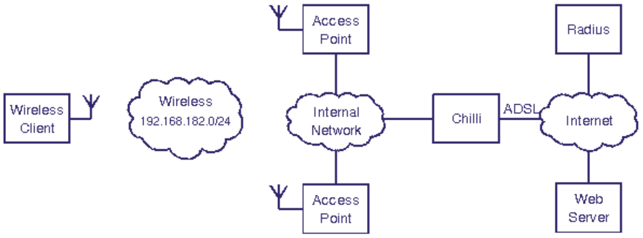


Figure 1: Network Layout for a ChilliSpot WiFi Hot Spot (copied from [1])

2.3 Simple Consumer Guest WiFi

Many manufacturers are also providing WiFi AP that supports dual SSIDs that allows consumers to provide a separate network for visitors at home. An example is Airport Extreme (from Apple) that is said to provide guest WiFi support. Users can configure wireless encryption, but there is no option for throttling, accounting, firewalls, etc. This means the system only have access and authentication without accounting.

3. USE CASES FOR GUEST WIFI

We envision many uses for a guest WiFi service that decouples authentication, access and accounting—which we will refer to as OpenWiFi. In the following, we describe some of these uses of OpenWiFi:

1. OpenWiFi can simply be used by a guest WiFi provider like Boingo. This makes the setup of guest WiFi easier and allows their customers to use existing account information. For companies like Boingo, this means less complexity because they have essentially out-source authentication to third-parties like Google or Facebook.

2. The OpenWiFi system can also be used to host access for events like conferences and workshops. Often these events require support for wireless access, which is often done by distributing some WPA passphrase. Such an arrangement is cumbersome for the users who has to remember the passphrase and also does not protect the event hosts who has to shoulder responsibility for the users. Instead, we can imagine how the venue can provide just the access, with the authentication done by some well known provider, and the accounting handled by a third-party or through an open-source implementation of the accounting service.

3. Often we have visitors in homes, offices or enterprises. Ideally we would want to give them access to network connectivity. However, current arrangement is clumsy with manual entry of their MAC address for MAC filtering or some shared WPA passphrase. By using OpenWiFi, one can innovate in the authentication. For example, access might be granted to anyone who is friends of the home owner on Facebook, which can be verified using the Open Graph protocol. This

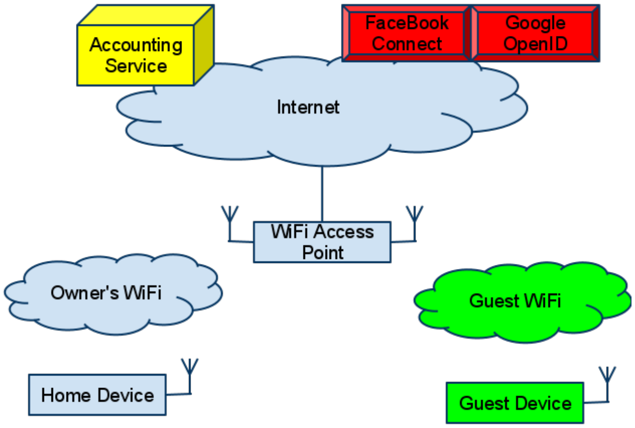


Figure 2: OpenWiFi Setup with an independent SSID for guest WiFi; an accounting service in the cloud and open authentication service provided by reputable service providers.

can of course be done at a larger scale for offices and enterprises.

4. OpenWiFi can also be used to create a community WiFi system, in which a community can pool together resources to host a single accounting service used by all its members. The accounting service can then provided the authentication of choice across all the WiFi access points in the community. A community might even try to augment other guest WiFi systems like Google WiFi in Mountain View.

In the above examples, there is a lot of different possible combination of access, authentication and accounting. Such flexibility is made possible for the decoupling of three orthogonal aspects of guest WiFi.

4. A PROTOTYPE OF OPENWIFI

To demonstrate the feasibility and utility of our proposal, we designed OpenWiFi—a guest WiFi service that embodies our proposal to decouple of authentication, access and accounting.

In OpenWiFi (Fig. 2), the authentication can be provided by any provider. Ideally, this service should be provided by someone where the guest users already have an account with, hence avoiding the creation and maintenance of another set of account information by the users. Fortunately, large providers like Google, Facebook, Wordpress, etc. are providing such authentication services using technologies such as OAuth and OpenID.

The access in OpenWiFi is provided by an existing WiFi access point. This can be a dedicated access point or one that is shared for other purposes (e.g., as in FON). Ideally, we would like the access point to be able to support multiple SSIDs which allows us to present multiple distinct WiFi networks to the users. This feature is common in modern WiFi chipsets. Such a capability in turn allows the guest WiFi provider to present a consistent SSID to the guest users across multiple networks—provided by cafes, homes, libraries, etc.

Finally, the traffic of the guest users have to be accounted for, i.e., tracked and logged, for the purpose of responsibility delegation or billing. This requires support in the WiFi access point for maintaining the appropriate information. Such statistics gathering can be implemented fairly easily and the information can then be aggregated by the guest WiFi provider.

4.1 A Typical User Login in OpenWiFi

To illustrate the functionality of OpenWiFi, let us run through how a guest user login:

1. Like all guest WiFi service, the user starts by associating with the guest WiFi SSID and getting an IP address. Now, the user fire up her browser and go to some site in the Internet.

2. Instead of providing the content of the site, the guest WiFi provider redirects the user to a login page. From the login page, the user is sent to a site (such as Facebook, Google or Wordpress) for authentication.

3. Upon successful authentication, an access token is then returned to the guest WiFi provider, who can now retrieve information of the user.

4. Now, the user can start using the guest WiFi access. Traffic generated by the user can now be mapped to the user identity received in step 3. This can also be used for billing if applicable.

The above procedure closely resemble the guest WiFi services provided today, presenting a well-understood mechanism to the users. However, we are able to present an even easier login mechanism than today's service by making use of well-known providers—such as Google and Facebook—for authentication. Beyond providing a login the user already understand, this avoids having the user register for yet another account.

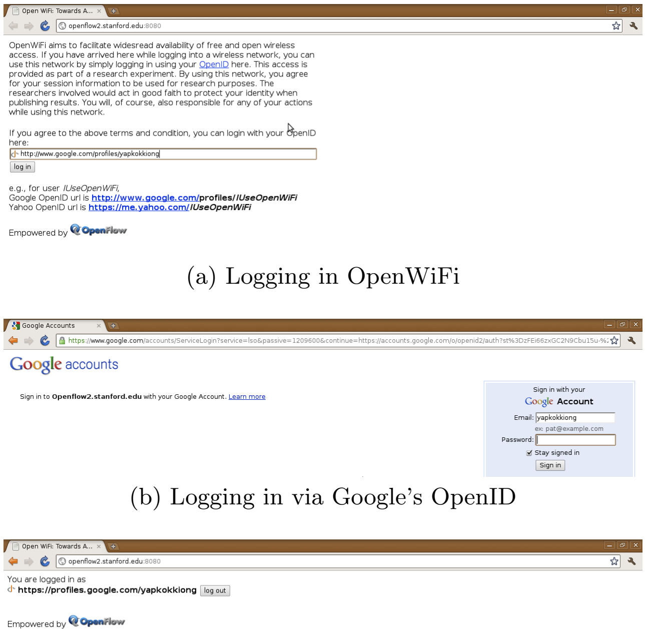
4.2 Implementation of a Prototype

We implemented a prototype of our OpenWiFi design using OpenFlow, Openwrt and web services such as OpenID and Facebook Connect.

Authentication in our system can be provided by Facebook Connect (using OAuth [3]) or OpenID [4] (e.g., with Google OpenID profile). An example of the login process using OpenID with Google is shown in Fig. 3 and Fig. 4. This authentication is provided by a web server with an appropriate backend. We have used webpy and Apache web server for our implementation.

For purpose of access control, redirection and accounting, we used an OpenFlow-enabled WiFi access point for our prototype. Specifically, we make use of the Pantou OpenFlow port [8] with Openwrt backfire distribution to create our access point. Via the control and statistics provided by the OpenFlow software switch, we are able to provide access control, redirection and accounting by defining these features in software within an OpenFlow controller, which also serves as our accounting service. Authentication and flows are also logged in the controller to a SQLite database.

For the hardware, we could have used any OpenFlow-enabled WiFi access point. In experiments and deployments, we have used both Linksys WRT54GL (which can support only a single SSID) and TP-Link TL-WNR1043ND (which



Open WiFi: Towards A...

openflow2.stanford.edu:8080

OpenWiFi aims to facilitate widesread availability of free and open wireless access. If you have arrived here while logging into a wireless network, you can use this network by simply logging in using your OpenID here. This access is provided as part of a research experiment. By using this network, you agree for your session information to be used for research purposes. The researchers involved would act in good faith to protect your identity when publishing results. You will, of course, also responsible for any of your actions while using this network.

If you agree to the above terms and condition, you can login with your OpenID here:

http://www.google.com/profiles/yapkokkiong

log in

e.g., for user *IUseOpenWiFi*,
Google OpenID url is **http://www.google.com/profiles/*IUseOpenWiFi***
Yahoo OpenID url is **https://me.yahoo.com/*IUseOpenWiFi***

Empowered by OpenFlow

(a) Logging in OpenWiFi

Google Accounts

https://www.google.com/accounts/ServiceLogin?service=lso&passive=1209600&continue=https://accounts.google.com/o/openid2/auth?st%3DzFEi66zxGC2N9Cbu15u-%2

Google accounts

Sign in to **Openflow2.stanford.edu** with your Google Account. Learn more

Sign in with your
Google **Account**

Email: yapkokkiong
ex: pat@example.com
Password:
☑ Stay signed in
Sign in

(b) Logging in via Google's OpenID

Open WiFi: Towards A...

openflow2.stanford.edu:8080

You are logged in as
https://profiles.google.com/yapkokkiong log out

Empowered by OpenFlow

(c) Successful login and confirmation

Figure 3: Series of screenshots during the login process in OpenWiFi when logging in using OpenID via Google.

supports multiple SSID with its Atheros chipset). This allows us to create virtual SSID and therefore supports multiple WiFi networks from the same box. Together with slicing [7], we can create an extensive guest WiFi network riding on the coattails of deployments.

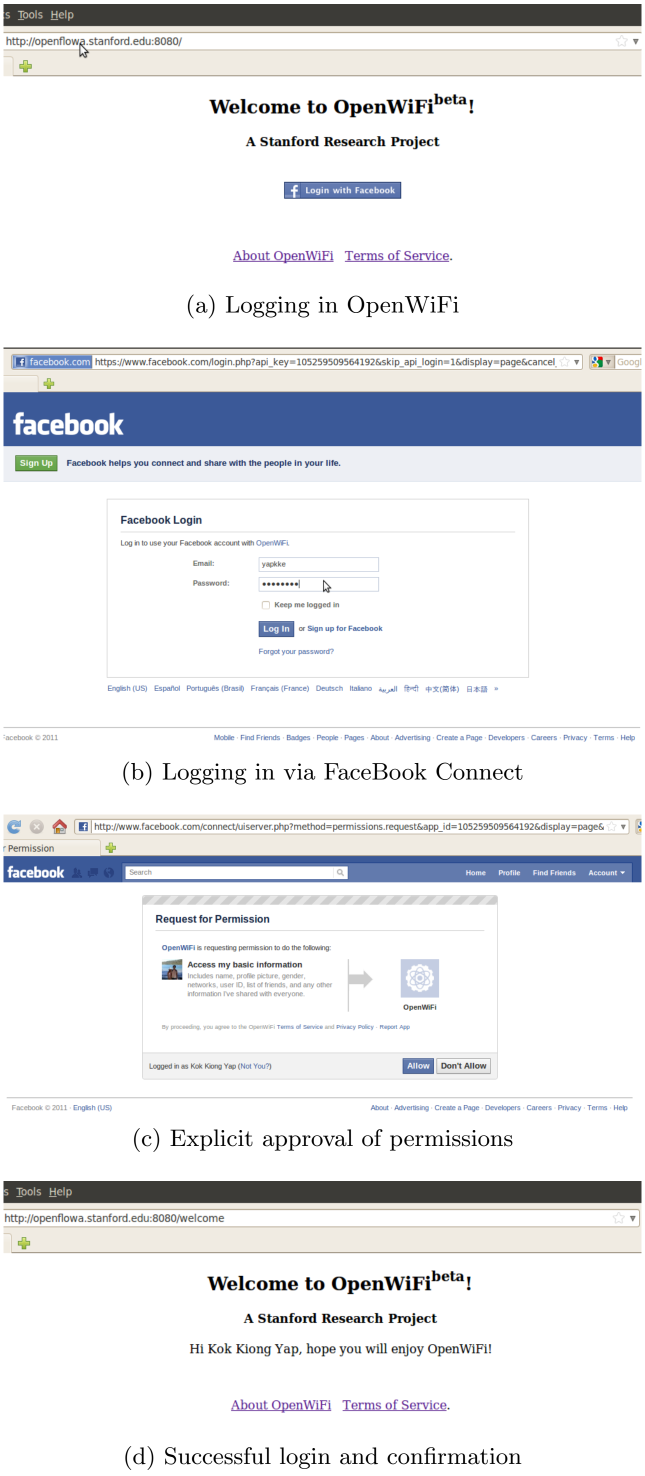
Using the above setup, the following control mechanism is implemented in the OpenFlow controller:

1. When a user first login into OpenWiFi, an IP address is assigned via DHCP. However, the client is marked as unauthenticated. During this time, the ARP, DNS and DHCP traffic of the client is handled and the rest of the traffic is blocked.

2. When the user fire up her browser and goes to some site, the HTTP traffic (i.e., TCP port 80 or 8080) is hijacked and the IP address of the destination is rewritten to that of the accounting server. Pretending to be the site that user wanted to access, the server sends a redirect to the actual login page (which is specifically HTTP redirect 403).

3. Now the user clicks on the appropriate button to go to the authentication site. Here, we can allow the user a choice of which authentication service to use. At this point, the controller also marks the user as pending authentication—forwarding traffic to the authentication service.

4. Now the authentication service (i.e., OpenID or FaceBook Connect) authenticates the user and establish his identity, and ask for the user permission to reveal information to the guest WiFi service. Once authenticated, an access token is returned to the controller and the user is redirected back to a specified site.



s Tools Help

http://openflowa.stanford.edu:8080/

Welcome to OpenWiFi^{beta}!

A Stanford Research Project

Login with Facebook

About OpenWiFi Terms of Service.

(a) Logging in OpenWiFi

facebook.com https://www.facebook.com/login.php?api_key=105259509564192&skip_api_login=1&display=page&cancel_

facebook

Sign Up Facebook helps you connect and share with the people in your life.

Facebook Login

Log in to use your Facebook account with OpenWiFi.

Email: yapkke
Password: ••••••••
□ Keep me logged in
Log In or Sign up for Facebook
Forgot your password?

English (US) Español Português (Brasil) Français (France) Deutsch Italiano العربية हिन्दी 中文(简体) 日本語 »

Facebook © 2011 Mobile · Find Friends · Badges · People · Pages · About · Advertising · Create a Page · Developers · Careers · Privacy · Terms · Help

(b) Logging in via FaceBook Connect

http://www.facebook.com/connect/uiserver.php?method=permissions.request&app_id=105259509564192&display=page&

r Permission

facebook Search Home Profile Find Friends Account

Request for Permission

OpenWiFi is requesting permission to do the following:

Access my basic information
Includes name, profile picture, gender, networks, user ID, list of friends, and any other information I've shared with everyone.

OpenWiFi

By proceeding, you agree to the OpenWiFi Terms of Service and Privacy Policy · Report App

Logged in as Kok Kiong Yap (Not You?) Allow Don't Allow

Facebook © 2011 · English (US) About · Advertising · Create a Page · Developers · Careers · Privacy · Terms · Help

(c) Explicit approval of permissions

s Tools Help

http://openflowa.stanford.edu:8080/welcome

Welcome to OpenWiFi^{beta}!

A Stanford Research Project

Hi Kok Kiong Yap, hope you will enjoy OpenWiFi!

About OpenWiFi Terms of Service.

(d) Successful login and confirmation

Figure 4: Series of screenshots showing how a user can login into a guest WiFi using Facebook Connect

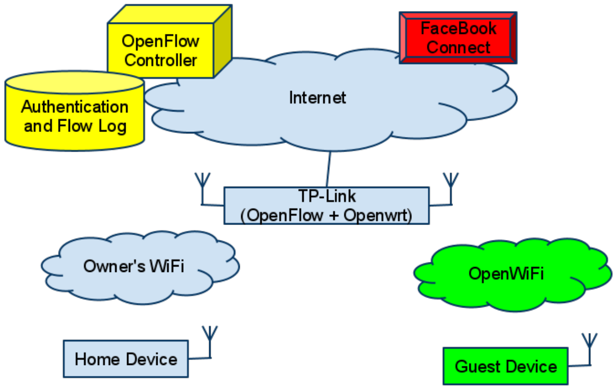


Figure 5: OpenWiFi Implementation with Facebook Connect and OpenFlow

5. Using the access token, the identity of the user is retrieved and the user is presented with a confirmation page. We can also redirect the user to the original site she was trying to access in step 2 here. The user is officially marked as authenticated at this point.

6. Now we log the traffic of the user together with the established identity for accounting. Using OpenFlow, this implies collecting flow statistics while data packets are processed at line rate (without being tunneled back to a central location).

4.3 Challenges

There are several challenges associated with the implementation of OpenWiFi. Here, we share the experience:

1. *Mapping Login and Traffic:* To associate a user identity to her traffic, we need to associate a web session with a particular user in a particular network. Since an accounting service can be serving many different physical networks, we can have many web sessions being mapped to many users in different networks. In general, this is a difficult problem without putting artificial constraints or requirements on the system, such as to authentication no more than one user at any point of time or to implement deep packet inspection in the network to identify cookie for the web session.

2. *Identifying Traffic to Authentication Services:* During a login to OpenWiFi, a client pending authentication will be allowed to access an external site for authentication. However, how do we identify this authentication site?[1] Taking the example of Google, this authentication request can be directed to many possible servers due to load-balancing. Further, content of the authentication page can be served by CDN like Akamai. One might think that we can observe the DNS traffic and determine the identity of the servers but that is fragile for two reasons, namely client can cache DNS entries and thus the avoid sending out any requests and servers might dynamically encode the IP addresses of their servers directly in their webpages.

3. *HTTPS Redirection:* When we redirect a client to the authentication page (in step 2 of the description for our implementation), our server is spoofing as the site the user wants to visit. Shall this site be accessed via HTTPS, our server would not be able provide appropriate credential and thus would not be able to redirect the user to the authentication page. While this is not a critical problem today, it can become an issue if HTTPS pages become more common.

4. *Rate Limiting:* One common feature in guest WiFi systems that co-exists with other networks (such as in FON) is the ability to rate-limit. Interestingly, rate-limiting uplink traffic for guest users is simple but rate limiting the downlink is difficult without any control of the upstream router. For example, if the access point is also the modem for the DSL connection, then a guest user can retrieve a large volume of data from a server which can congest the last-mile connection, unless we can rate-limit at the DSLAM which is usually not the case. A coarse solution is to only allow TCP traffic, which will then be throttled appropriately.

5. *Device/OS/Browser Compatibility:* The cross product of device, OS and browser is a large set which impose an interesting challenge. To ensure compatibility across this large set is an ongoing task for any guest WiFi provider and a tedious task that requires much perseverance or incentive. By separating access, authentication and accounting, we hope to push this task to those who are motivated to do so, e.g., the accounting service.

5. DEPLOYMENTS AND TRIALS

To test and verify our prototype, we deployed it and experimented with the system. As of 8 August 2011, we have served Internet connectivity to over 60 unique individuals.

5.1 Home Trials

As part of an experimental home network deploymen [9], we added the OpenWiFi as a service to the homes to allow visitor access of home networks. In doing so, we demonstrate how we can ride on the coattails of other deployments by combining OpenWiFi with slicing/virtualization technologies. Within the initial month of deployment, we have gathered 30 users.

5.2 Workshop Wireless Access

We also used OpenWiFi to support wireless access in a day-long workshop. Using Facebook Connect, the login procedure was self-explanatory and many were able to navigate it. However, many of the participants were concerned about the privacy implications of using a Facebook login—a concern that might be exacerbated by the fact that such events are expected to have unregulated wireless access. Nevertheless, we have a good number of users (i.e., 12 out of 30) and were able to provide a useful service that better serves the security policies of the network provider. The traffic for the

[1] In the words of a Coova developer that integrated Facebook/Oauth authentication with CoovaChilli: "when using something like Oauth, the access controller has to adjust the walled garden in a per session (per device) basis in order to allow the user to authenticate at their home provider—which isn't known beforehand."

day long workshop were mainly HTTPS, HTTP, SSH and IMAPS, with volume ranging from 30 MB to 190 MB per hour.

6. DISCUSSION AND CONCLUSION

Our experience indicates that the decoupling of access, authentication and accounting is a very powerful concept that enables many interesting use cases for guest WiFi—an idea that we hope would be adopted by future guest WiFi deployment. For example, one can now build a network by riding on the coattails of other deployments like home, cafes, restaurants, etc. This allows for system wide optimization, and can possibly reducing the discovery burden for guest WiFi shall a few dominant accounting providers emerge.

Beyond the use cases described, we also recognize that the system separates authentication from security. The WPA mechanism has always been to provide both authentication and security in WiFi. This can be an issue as described in [5]. By moving authentication to the cloud, we have also reduce the scope of WPA for guest WiFi. This means one use WPA merely for link encryption (by assigning some default passphrase) which in turn implies better security for guest WiFi users.

Overall, we are excited about the potential of OpenWiFi and how the decoupling of access, authentication and accounting broadens the possibility of guest WiFi—fixing practical issues of responsibility delegation while maintaining efficiency of the system.

7. REFERENCES

[1] Chillispot - open source wireless lan access point controller. spice up your hotspot with chilli. http://www.chillispot.info/.

[2] Coovachilli / coova : Open source captive portal access controller and radius software. http://coova.org/CoovaChilli.

[3] Oauth community site. http://oauth.net/.

[4] Openid foundation website. http://openid.net/.

[5] P. Eckersley. Why we need an open wireless movement. https://www.eff.org/deeplinks/2011/04/open-wireless-movement, April 2011.

[6] A. K. Gupta. Wi-fi access for retail: Tips on how to address key challenges. http://www.networkworld.com/news/tech/2011/042511-wifi-retail.html, April 2011.

[7] R. Sherwood, G. Gibb, K.-K. Yap, G. Appenzeller, M. Casado, N. McKeown, and G. Parulkar. Can the production network be the testbed? In *Proceedings of the 9th USENIX conference on Operating systems design and implementation*, OSDI'10, pages 1–6, Berkeley, CA, USA, 2010. USENIX Association.

[8] Y. Yiakoumis. Pantou : Openflow1.0 for openwrt. http://www.openflow.org/wk/index.php/Pantou_:_OpenFlow_1.0_for_OpenWRT.

[9] Y. Yiakoumis, K.-K. Yap, S. Katti, G. Parulkar, and N. McKeown. Slicing home networks. HomeNets 2011, 8 2011. (in conjunction with SIGCOMM 2011).