This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the WCNC 2010 proceedings.

Virtual Access Points for Transparent Mobility in Wireless LANs

Yan Grunenberger
Grenoble, France
yan@grunenberger.net

Franck Rousseau
Grenoble Informatics Laboratory
University of Grenoble, France
Franck.Rousseau@imag.fr

***Abstract*—Mobility management in WiFi networks is still an open issue today: there is no standard method defined, and client station mobility is handled either via proprietary protocols, or simply by re-association. However, managing mobility in an infrastructure network is utterly important for several reasons: controlling delay and jitter in communications, managing clients from the network, optimizing performance. We propose the concept of *virtual access points* to manage mobile station in infrastructure networks. In this scheme, stations are not aware that they move, and all the complexity is pushed back inside the network. It is then possible to control mobility from a global point of vue, to optimize network resources for mobile stations, hence providing a better quality of service. Finally, this scheme is compatible with existing clients without any hardware nor software modification.**

***Index Terms*—WLAN, mobility, access point, virtualization, infrastructure.**

I. Introduction

Terminal mobility has been largely studied in the literature. With the widespread of wireless connectivity, users got rid of wires to turn their computing devices to mobile terminals. Just like with a mobile phone, they want to stay connected while moving around. In the case of WiFi networks and in IEEE 802.11 terminology, this means keeping the wireless interface of the mobile station *associated* with an access point, this mode of operation being called infrastructure mode.

In telco networks, the association process is taking place in the network itself; in WiFi networks, mobility decisions are left to the mobile station. Hence, a station has to scan potential channels to discover new access points and request association. Moreover, the interconnection network must react to these successive associations, by keeping bridging tables up to date, or by allocating a new address for example. These procedures cause variability in the delays during mobility, degrading significantly the quality of service for constrained traffic, like voice over IP. Furthermore, since the mobility decisions are taken in the station, it is impossible to provide efficient ressource management in the network of access points, for load balancing or interference mitigation typically.

In this context, we propose to get rid of mobility management in mobile stations and put it entirely inside the network of interconnected access points. To do so we simply change our point of view: we propose to consider stations as being fixed; conversely, the access point to which a station is connected is now mobile. Of course, physically this is not the case, and we introduce the concept of *virtual access point*, which is a mobile entity within the infrastructure network. Every mobile station is therefore associated with its own virtual access point when it connects to the network, the latter moving along with its client.

In this way we totally get rid of the problems introduced earlier, while being fully compatible with legacy clients, without any hardware nor software modification. We will show how we could easily implement this concept using a packet manipulation framework called *PacMap*. Using this lightweight framework for rapid prototyping, we have developed a first implementation in Python to test the concept on a simple scenario, and then natively for improved performance. A first round of experiments with these prototypes gave some insight on performances and limitations of the proposed solution. Finally, taking a look at related work on mobility in WiFi networks, we will conclude and present future work.

II. Mobility management in Wireless LANs

Traditionally, mobility management is divided in two categories: macro-mobility deals with large scale movements, where the mobile node changes of IP network; micro-mobility deals with local mobility inside a single IP subnetwork, comprised of several contiguous attachment points. In the first case, the mobility process is taking place at the data link and network layers, in particular requiring the allocation of a new address. This situation has been widely studied in the literature, in the scope of Mobile IP and its derivatives mainly.

This work clearly targets the second case, micro-mobility, more specifically for IEEE 802.11, which Montavon *et al.* [1] call “L2 handover”. The transfer between two access points in the same 802.11 network has an impact on upper layers, enforcing timing constraints on ongoing connections, typically IP, TCP, RTP. Buffering techniques are commonplace to prevent data losses, but interactive traffic like voice over IP cannot tolerate too large and variable delays during mobility, otherwise degrading severely the quality of service.

In infrastructure mode, every 802.11 access point or AP defines a basic service set or BSS. These BSS are interconnected via a distribution system or DS, and form an extended service set, or ESS, which provides extended wireless coverage. The IEEE 802.11 standard does not provide any specific definition

978-1-4244-6398-5/10/$26.00 ©2010 IEEE

of a distribution system, and the handoff procedure is left entirely at the mobile station. In current implementations, the handoff operations are initiated by the mobile terminal as described in [2]:

- Signal monitoring is performed on the radio link, by listening to beacons and active traffic, and notifications are generated on certain thresholds.
- Scanning is performed across the available channels, the mobile station passively listening, or actively requesting, beacons from nearby access points. Active scanning might cause packet losses, since channel switching is not instantaneous [3]. Some wireless adapters implement a passive mode, the scan taking place during inactivity periods, for example while they are blocked by the network allocation vector, NAV.
- Reauthentication to a new AP. Attempts are made according to a priority list, and the process can be optimized through credential transfer between infrastructure parties to reduce the overhead caused by security related exchanges as well as cryptographic functions.
- Reassociation with the new access point.

This procedure is a source of many important and variable delays while users move in a WiFi network. Mishra *et al.* have made an empirical study of the IEEE 802.11 MAC layer handoff process [2] and measured the durations of these operations. The authors have noticed a great variance in transition delays — from 200 to 1 000 ms, given that a maximum end to end delay of 150 ms is recommended for interactive voice communications.

The main sources of delay identified by Mishra *et al.* are the probe delay (3 to 7 frames), the authentication delay (3 to 4 frames), and the reassociation delay (3 to 4 frames). This work also mentions the bridging delay, caused by the necessary updates of the bridging tables inside the distribution system. According to the authors, the probe delay accounts for 90% of the time spent in the handoff procedure. Other experiments [9] confirm these results: scanning for APs takes too much time.

Placing the handoff decision in the mobile station might seem a good solution in the first place, since it is in the best position to identify nearby access points. However, the main task of a mobile network is to guarantee that the traffic will be delivered to and received from mobile terminals with a consistent level of quality, and the global view needed to achieve this is only available from inside the network.

III. Virtual access points

Since mobile stations are the cause of several problems as stated earlier, we propose that they do not move anymore! Of course, in the physical world, users and their terminals will not remain static, but with a simple change of point of view the wireless stations may appear as motionless for the network: all we need is move the network together with the stations. However, every mobile terminal has its specific behavior depending on its owner, and the solution requires a new level of abstraction to provide every terminal with its own view of the network.

In the case of 802.11 wireless LANs, running in infrastructure mode, it is fairly easy to set up such a configuration. Mobile stations get their view of the network only from the AP they are associated to, and the beacons they get from neighboring APs when they are in scan mode. Providing a custom view to each mobile is then straightforward: (i) create a custom virtual AP for the mobile station, (ii) prevent the mobile station to enter scan mode by moving its virtual AP so that it gets the illusion of being almost static relatively to it, (iii) if the mobile enters scan mode, reply with the custom beacon previously associated to it.

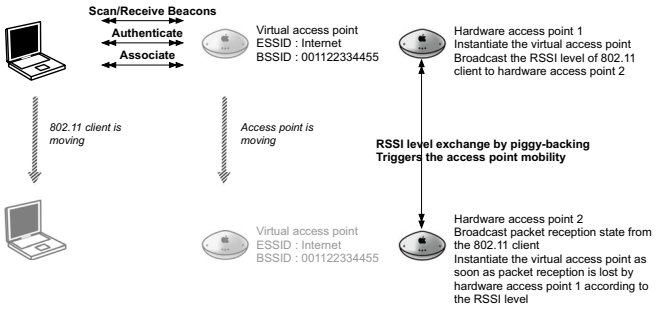

Fig. 1. Mobility management with virtual access points.

By changing of point of view, we turned terminal mobility management into virtual AP mobility management inside an infrastructure network of static hardware APs. This brings two problems:

- detect terminal mobility from the network to ensure that its virtual AP is moving along in the infrastructure network; several solutions are available: signal monitoring, geo-positioning, etc.
- provide connectivity to the terminal wherever it is in the network.

The first point is related to the movements of the terminal among a set of access points, and the key point is to maintain a good signal level to guarantee reliable packet delivery. APs monitor the signal of mobile stations, triggering events on given thresholds.

The second problem is a bit more complex. 802.11 stations rely on two key elements to evaluate their connectivity: on the one hand, acknowledgements received at the MAC level indicate good connectivity; on the other hand, when there is no ongoing traffic, stations rely on the periodic beacons broadcasted by the AP they are associated to.

IV. Implementing Virtual AP

PacMap [4] is a lightweight framework that we have developed to help experimenting with cross-layer architectures. Its main feature is to offer a very simple way to handle raw packets coming from and going to the network in a single place in user-space, requiring no kernel hacking. It is very easy to develop any kind of protocol handler when used jointly with *Scapy*[1] for example, a tool developed in the wireless security

[1] http://www.secdev.org/projects/scapy/

community to ease packet manipulation — parsing, forging, etc. With the 802.11 support offered in these tools, we can emulate the presence of an access point as soon a station is in the radio coverage. From the station point of view, there is no more mobility, its virtual access point is constantly available.

This kinds of approach is already implemented in recent hardware access points. Several networks are advertised using different beacons on a single channel, for example a private encrypted network and an open public network. However, a limitation comes from the fact that these APs running on different hardware are independent entities. Using PACMAP, we can easily transfer all the parameters from one hardware access point to the next, hence moving a virtual AP from the first to recreate it on the second. Virtual APs are now mobile transparently.

Using virtual APs, mobile stations keep a permanent connection to the network, smoothing support for constrained application like voice over IP. Moreover, connectivity control is handled exclusively inside the network, easing the management of wireless infrastructure networks. Finally, virtual APs contain and carry around all the necessary information for their associated mobile station: routing information, caches, available modulations, etc. We can easily customize every connection based on cross-layer information.

We did a first implementation based on simplifying assumptions: dense AP topology, operated on a single channel. Although this set up is not scalable and will cause a large number of collisions when many clients are active simultaneously, it provides a clean and simple experimentation environment: mobility detection and inter-AP signalisation is made easy, as long as neighboring APs are in range. These two aspects are not the main subject of our study, hence it is not a problem to rely on very simple implementations.

This first proof of concept prototype of virtual APs was developed with PACMAP in Python. According to the procedure defined earlier, a client scanning for a network should be proposed a dedicated access point. On the first step, we filter 802.11 probe requests coming from the stations and reply with a custom management frame. We give below a code excerpt to show how simple it is to implement such a handler with PACMAP.

```
1  def proto80211_probereq(packet, length):
2      # AP probe request reception
3      dot11_frame = Packet(packet)
4      dot11_frame.decode_payload_as(Dot11)
5      # Get the client address
6      client = dot11_frame.getlayer(Dot11).addr2
7      # Generate a network ID of type "Client-XX
           :XX:XX:XX:XX:XX" where XX:XX:XX:XX:XX:
           XX is the station's unique hardware
           address
8      ssid = "Client-%s" % client
9      current_timestamp = time.mktime(datetime.
           datetime.now().timetuple())*1e6+
           datetime.datetime.now().microsecond
10     # Build a response packet
11     dot11_answer = Dot11(
12     type = "Management",
13     addr1 = dot11_frame.getlayer(Dot11).addr2,
14     addr2 = bssid,
15     addr3 = bssid)/Dot11ProbeResp(timestamp=
           current_timestamp,cap = 0x0104)/
           Dot11Elt(ID=0,info=ssid)/Dot11Elt(ID=1,
           info="\x82")/Dot11Elt(ID=3,info="\x06")
16     # Send packet
17     pacmap.sendpacket(str(dot11_answer),len(
           str(dot11_answer)),1)
18     return 1
```

The station with a MAC address `00:11:22:33:44:55` will receive a reply from an access point with an identifier `Client-00:11:22:33:44:55`. It will then associate according to the standard procedure.

As soon as the station is associated, it is added in the list of managed stations. From now on, a periodic process is sending a beacon at regular intervals, keeping the station aware that the access point is still available in its radio range, hence still associated to its network. The signal level of the last packet received from the client is piggy-backed in the beacon. Using this *beacon stuffing* [5] technique, the access point keeps its neighbors informed on the signal quality with the station.

Finally, we have to take care of secondary access points operating on the same channel: these APs are over-hearing beacons and take decisions accordingly. We maintain two lists in every AP: a list of managed stations and a list of monitored stations. In this simple scenario we compare the signal level of the packets received from a monitored station with the signal level advertised by its current access point. We do not need any other signaling to migrate the association from one AP to the other: when a secondary AP gets a better signal from the station it registers this station as a new client and starts sending beacons. The former AP over-hears these beacons, learning that the station is now associated to a new AP, deletes its association and stops sending beacons.

To this core, we add a DHCP service, and some glue code to handle ARP traffic: requests are filtered and a reply is send according to the situation of the station in the network. We have implemented a fully functional mobility infrastructure for our simple scenario with only a few lines of Python.

V. MOBILITY EVALUATION

During the evaluation of our mobility solution, we have identified several parameters used in its management. These parameters are mainly responsible for the reactivity of our solution, which has been evaluated in lab environment with two access points running PACMAP with the virtual AP script, according to the setup presented in Fig. 1.

The first parameter is the threshold used to decide when the network traffic should be managed by the new access point rather than the current one. In our approach, the threshold is evaluated by experimentation, using RSSI levels retrieved from the relative position of the client compared to the access point. But this threshold can be used together with other parameters: the load of the access point or the intensity of the radio traffic in the neighborhood. Using a good estimator, we can reach a good load balancing between access points.

In our simple case, we have determined that a margin of 15 dB in signal level received by one access point over another

access point is necessary to achieve a good radio coverage. When the signal level received at one access point is exceeding the level of another access point, we register the station on the new access point and unregister on the old one.

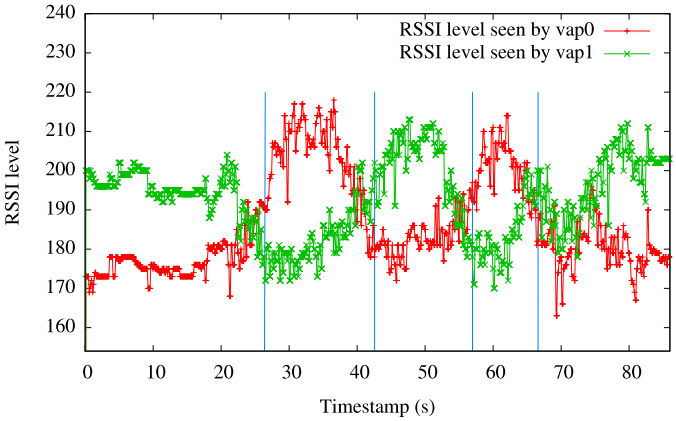

Fig. 2. Signal level evolution as seen by the access points. Mobility decisions on vertical lines.

An example of the evolution of the signal level received by the two access point is presented in Fig. 2. During a first step, the mobile station is static, and its network traffic is managed by the access point vap1. Then, the station is moving and the received signal level is varying on both access points. When the threshold is reached, the network traffic of the client is managed by vap0. As the client is moving between the two access point, three mobility decisions are taken, so the station is always associated to the AP offering the best connectivity according to our straightforward metric. The correlation between signal evaluations on the current access point and the remote access point prevents fluctuations and offers a good compromise between the stability of the decision and connectivity support.

An access point will unregister its client when it receives a beacon sent by the new access point taking the responsibility of the client.

In our case, the beacons are sent every 100 ms, which corresponds to the reactivity time to assess the signal level. But on a more practical point of view, it is possible that a beacon can be sent by the previous access point when the migration should have occurred. The problem is related to the response time of the beacon management by PACMAP. In order to insure the good behavior of the migration mechanism, we introduce a Δ parameter that correspond to the number of beacons sent by the new access point. Using experimental evaluation, two consecutive beacons are enough to avoid any unwanted unregistration. The problem and its solution are presented on Fig. 3: beacons broadcast is stopped according to the effective unregistering of the client using Δ interval.

We tested our solution with bidirectional network traffic using short packets with regular frequency, in order to mimic VoIP traffic. The platform is made of two computers with a wireless card and running PACMAP with the Python script.

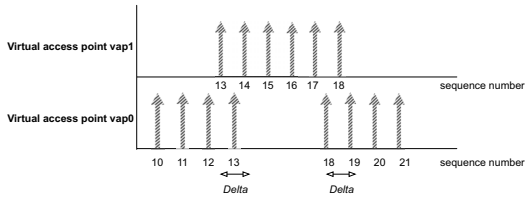

Fig. 3. Beacons broadcast during mobility detection.

The Ethernet interfaces created by PACMAP are connected using an Ethernet bridge. This way, we reproduce a standard infrastructure network.

We used a *ping* of 64 bytes every 200 ms on a client moving between the two access point hosting the virtual access points and we monitor the evolution of the latency according to the movement of the terminal. The *ping* is running from the initial access point.

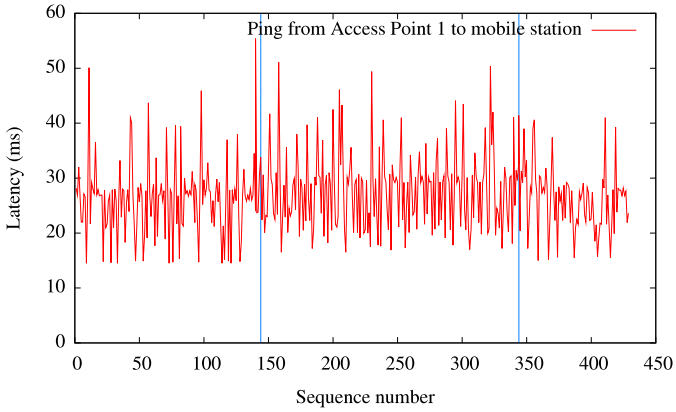

Fig. 4. Evolution of the latency according to the mobility of the terminal, Python code.

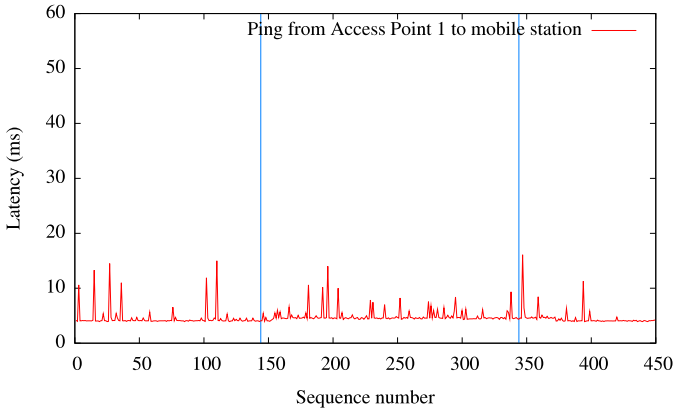

Fig. 5. Evolution of the latency according to the mobility of the terminal, C code.

Fig. 4 presents the evolution of the latency. The moments where the terminal is migrating from one access point to another are shown using the vertical lines, between sequence

number $seq = 144$ and $seq = 344$ where the mobile is connected to the access point AP 2.

Despite the presence of the Ethernet bridge, we do not notice any increase of the latency during the transition phase: we achieve our initial objective, a transparent mobility for the mobile station. The high latency and its large variance during, even when the station is in a static situation, are related to the use of *Scapy* code in the Python script.

Previous results obtained with PACMAP when comparing a Python implementation with a native implementation of the 802.11 state machine indicated that the latter gives much better performances. A native implementation of the virtual access point concept, although much more complex, would be more efficient. We have implemented a native version of this prototype in C code and conducted the same experiments, results are shown on Fig. 5.

We observe that the latency has dramatically decreased using native code. The fluctuations observed are only due to the medium access, and we observe the same property of permanent connectivity with no increase of the latency caused by the transparent handoff procedure. The cost of the network bridge is clearly visible with an increase of 0,4 ms over the mean latency of 4 ms.

VI. ANALYSIS

As stated earlier, the solution that we have implemented to test the virtual AP concept is a simplified scenario: contiguous access points operating on a single channel. Although, it is possible to extend this concept to more complex and generic network setups.

Standard WiFi infrastructure networks only have a few access points for all the stations. With the concept of virtual APs, we need one AP per client, hence increasing the number of beacons to transmit. It is possible to mitigate this problem by reducing the frequency at which beacons are sent according to the number of clients on the same channel. Although, in our scenario this will reduce the reactivity of the handoff procedure.

This drawback is dependent on the type of control channel we use for the signaling between access points. In our prototype we chose the simplest solution, the radio channel. However, using the distribution system instead of the access network to carry the signaling can greatly improve the reactivity of the solution. Specialized interoperability protocols like 802.11r or LWAPP can be used for this purpose. With such protocols, APs can periodically exchange information with each other on the stations they are managing and monitoring, and take timely handoff decisions.

Using a single radio channel in the infrastructure network was a nice approach to simplify our prototype. In a more realistic scenario, separate channels must be used to reduce interference between adjacent BSS, decrease collision rate, and guarantee better connectivity. A promising solution to switch stations from one channel to another is based only on beacons: we have experienced that standard clients like iPhones and iPods would switch their channel when they receive a beacon for the network they are currently associated to with a different channel specified in it. With an adequate off-band signaling protocol, a new AP could request channel switching of a station to its current AP, then triggering a handoff with channel switching.

VII. RELATED WORK

IAPP [6] was a first attempt to propose a protocol suitable for mobility management in 802.11 networks relying on inter-AP communication. This standard was withdrawn and solutions left to the device makers. There is no generic mobility support available in access points, only proprietary solutions when available. Mobility is mostly handled in the client stations.

Mishra *et al.* [2] have developed heuristics to minimize the number of active scanning phases needed and improve the response time. The authors then introduced the use of neighbor graphs [7] and proposed a modification of IAPP. This optimization relies on a proactive caching system in which neighborhood relations are used to send context information to appropriate stations.

Shin *et al.* have also proposed a cache mechanism in [8]: it relies on the information gathered from previous active scans. The author introduce selective scanning too, based on a channel mask and used to optimize the scanning cycle for future handoffs. These propositions are efficient only in dense and stable networks. Moreover, the reactivity of the clients is worse when new access points appear in the network.

Velayos *et al.* [9] proposed a different approach: they modify the behavior in 802.11 wireless cards, triggering a scanning cycle as soon as a packet loss which is not due to a collision is identified. The authors demonstrate that the scan can be initiated as soon as three consecutive packets are lost in absence of collision, reducing the handoff delay from 900 ms to 3 ms when no authentication is used and the station is actively transmitting traffic.

Mhatre *et al.* [10] have explored the triggers that can be used to identify the need to initiate a handoff. The active scan using probing should be conducted only when no information has been obtained using passive ways. The authors propose and evaluate a series of triggers on the RSSI signal to trigger an access point change. They evaluate an algorithm using beacon detection, a threshold algorithm, then a hysteresis-based algorithm as well as a trend analysis algorithm. During evaluation, the threshold algorithm and the beacons detection algorithm are exposing larger delays (between 530 and 860 ms). For the other algorithms evaluated, the delays are much lower (140 to 450 ms), as they are proactive algorithm, triggering a handoff when the connection is still present.

Even if mobility models are proposed in order to anticipate the movement of the user, to our knowledge there is no proposal of 802.11 handoff enhancement taking into account a mobility decision coming from the network itself, like in cellular networks for example. 802.11 networks represent a good opportunity to study the micro-mobility mechanisms: indeed, taking into account the physical parameters, like the

RSSI level, or the presence of beacons, can bring great enhancements to micro-mobility management.

VIII. Conclusion

Mobility management is a huge problem in the 802.11 wireless networks. The standard does not include any generic mechanism, resulting in a complete void for an obvious functionality in the 802.11 typical use case. The current proposed mechanisms are only relying on the quality of the client implementation, which is responsible for its own mobility management. During mobility, the 802.11 protocol is inducing delays that are not compatible with sensitive applications such as VoIP, despite being a good application candidate for theses networks. By placing the management of mobility at the core of the access point infrastructure network, we try to propose an elegant solution providing efficient mobility support with existing 802.11 clients, as well as new possibilities of management from the infrastructure network itself. Experimenting this approach has required the development of a virtual access point solution, which shows promising performances for delays in a mobility situation.

But, more interestingly, the virtual access point approach can be extended to solve other common problems in wireless networks: fast reauthentication, per-device parameters management (bursting mode, security parameters), and mitigate performance anomaly — as beacons are aimed at specific devices, it is possible to selectively force certain parameters like the data rates allowed.

Acknowledgments

This work was partially supported by the French National Research Agency (ANR) project AIRNET under contract ANR-05-RNRT-012-01, project ARESA under contract ANR-05-RNRT-01703, and project ELAN under contract ANR-08-VERS-008.

References

[1] N. Montavont and T. Noël, "Analysis and evaluation of mobile IPv6 handovers over wireless LAN," *Mobile Networks and Applications*, vol. 8, no. 6, pp. 643–653, Dec. 2003.

[2] A. Mishra, M. Shin, and W. Arbaugh, "An empirical analysis of the IEEE 802.11 MAC layer handoff process," *ACM SIGCOMM Computer Communication Review*, vol. 33, no. 2, pp. 93–102, Jan. 2003.

[3] P. Bahl, R. Chandra, and J. Dunagan, "SSCH: slotted seeded channel hopping for capacity improvement in IEEE 802.11 ad-hoc wireless networks," in *Proceedings of MobiCom'04*, Sep. 2004, pp. 216–230.

[4] Y. Grunenberger and F. Rousseau, "A lightweight packet manipulation framework for cross-layer experimentation," Under submission, 2009.

[5] R. Chandra, J. Padhye, L. Ravindranath, and A. Wolman, "Beacon-Stuffing: Wi-Fi Without Associations," in *Proceedings of the 8th IEEE Workshop on Mobile Computing Systems and Applications (HotMobile 2007)*, Feb. 2007.

[6] IEEE Std 802.11f™–2003, "IEEE Trial-Use Recommended Practice for Multi-Vendor Access Point Interoperability via an Inter-Access Point Protocol Across Distribution Systems Supporting IEEE 802.11™ Operation," IEEE Standard, 2003.

[7] A. Mishra, M. Shin, and W. Arbaugh, "Context caching using neighbor graphs for fast handoffs in a wireless network," in *Proceedings of INFOCOM 2004*, Mar. 7–11 2004.

[8] S. Shin, G. Forte, A. Rawat, and H. Schulzrinne, "Reducing MAC layer handoff latency in IEEE 802.11 wireless LANs," in *Proceedings of the Second International Workshop on Mobility Management & Wireless Access Protocols*, Jan. 2004, pp. 19–26.

[9] H. Velayos and G. Karlsson, "Techniques to Reduce IEEE 802.11 b MAC Layer Handover Time," in *Proceedings of IEEE International Conference on Communications*, 2004.

[10] V. Mhatre and K. Papagiannaki, "Using smart triggers for improved user performance in 802.11 wireless networks," in *Proceedings of the 4th International Conference on Mobile Systems, Applications and Services*, Jan. 2006, pp. 246–259.