WiFi Overcast: Enabling True Mobility for Realtime Applications in the Enterprise

Nabeel Ahmed, Usman Ismail
`{n3ahmed, uismail}@cs.uwaterloo.ca`

David R. Cheriton School of Computer Science
University of Waterloo

Technical Report CS-2009-01

January 27th, 2009

WiFi Overcast: Enabling True Mobility for Realtime Applications in the Enterprise

Nabeel Ahmed, Usman Ismail
{n3ahmed, uismail}@cs.uwaterloo.ca

David R. Cheriton School of Computer Science
University of Waterloo

Abstract. Enterprises are increasingly deploying Wireless LANs to provide mobile access to users in corporate offices. However, existing enterprise WLANs are far from being truly mobile. In particular, they do not adequately support *continuous mobility*, where users access the network on-the-go. Furthermore, WLANs that do provide continuous mobility support require client modifications, making them hard to deploy in practice [20]. In addition, with the growing interest in realtime applications such as voice and video, users are increasingly placing additional (QoS) demands on the network, which for inadequately designed WLANs, does not scale to large numbers of users [10]. In this paper, we propose Overcast, a novel WLAN architecture that targets scenarios demanding continuous mobility and real-time support for 802.11 clients. Overcast does not require client modifications and supports all 802.11 standards. Though Overcast borrows some features from prior WLAN designs, it improves on them by incorporating a novel RF mapping framework (proposed in [3]) for accurate online detection of RF interference. We describe the architecture of Overcast in detail and discuss our current efforts in realizing such a system on off-the-shelf commodity hardware. We also describe an example application of Overcast to highlight it's usefulness in supporting realtime applications in continuously mobile user environments.

1 Introduction

Falling prices and demand for a mobile workforce have caused a proliferation of wireless LANs in modern enterprises. In recent years, the emergence of new usage paradigms such as *continuous mobility* and the growing interest in applications such as voice and video have further driven widespread adoption [17]. Continuous mobility is defined as an active use of a wireless network while the client is moving within its area of coverage. This is in contrast to nomadic use, where mobility is constrained to a small region or cell. Nomadic use is the dominant usage pattern in WLANs today but this is expected to change as support for continuous mobility improves.

Recent growth in the use of smart-phones with large screens and greater processing power has also spurred demand for real-time application support in continuously mobile scenarios [17]. This has created a myriad of challenges for network designers because the popular 802.11 standard has limited support for continuous mobility. Continuous mobility significantly increases variability in link characteristics, making it hard to guarantee any Quality of Service (QoS). A wireless signal can drop to a deep null with just a $\lambda/4$ movement in receiver position (3.7 cm at 2 GHz) [18].

Researchers have attempted to address the challenges of continuous mobility at all layers of the network stack, from the application layer [16], to the physical layer [17]. In this paper, we present a more comprehensive view of continuous mobility: the architectural requirements for supporting continuous mobility and how they can be realized in today's enterprise 802.11 deployments.

In traditional WiFi deployments, associating and managing the connection to an AP has been solely the responsibility of the client. This includes, among other tasks, selecting a suitable AP and coordinating hand-offs during periods of mobility. However, with the advent of centralized architectures [12] [26], network management tasks have moved progressively up the hierarchy, into the controller. Despite this, client association techniques still pre-dominantly rely on the client to find a suitable AP with which to connect to the network. This is detrimental to continuous mobility because the clients don't have global knowledge of the network state. In our proposed architecture, we relegate the client's role to only associating with a single (virtual) access point. Thereafter, the network is responsible for ensuring that clients are able to meet their Quality-of-Service (QoS) requirements. This is accomplished by managing transmissions from one or more neighbouring APs. In Section 2, we present a key set of requirements necessary to support such a WLAN system. We discuss the Overcast architecture and it's details in Sections 3 and 4 respectively. In Section 4.3, we discuss our current efforts towards realizing Overcast on commodity hardware. In Section 5, we briefly present an example application of Overcast to support in-house VoIP-calling for the enterprise. Related work is covered in Section 6 and we conclude in Section 7.

2 Requirements

We list the set of requirements that are necessary for enterprise WLANs to support continuous mobility for real-time applications.

(1) Guaranteed Coverage: The WLAN should guarantee network coverage throughout the enterprise. Otherwise, network disconnections during mobility could occur, making it difficult to support real-time applications.

(2) Adequate Capacity Provisioning: Prior studies have shown that WLANs today perform poorly in congested scenarios where many clients congregate at one location [15]. This is primarily because the network is not provisioned to simultaneously support many clients. Given the nature of realtime applications that generate continuous streams and require minimal delay jitter on the link, such congestion effects could be catastrophic [13]. Therefore, the network must be provisioned with enough capacity to prevent such congestion collapse from occurring.

(3) Precise Knowledge of RF Environment: In order to support continuous mobility, we should be able to rapidly re-configure the network in response to user movements that change the surrounding RF environment. Handling such changes requires closely tracking clients, detecting performance degradations, and reacting to them. This requires precise knowledge of the RF environment, and in particular, information about interference that dramatically affects client performance.

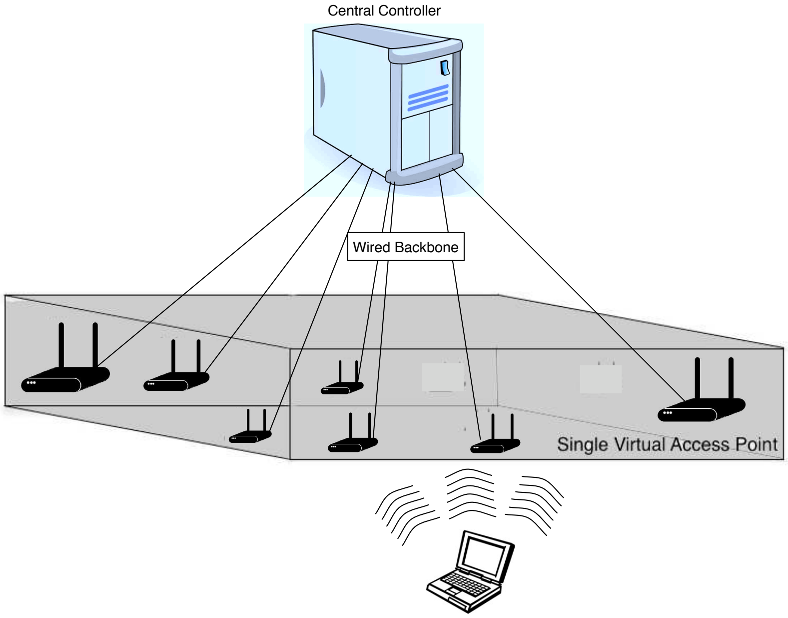

Fig. 1. All APs emulate a single virtual AP that the client associates to only once for the lifetime of the connection

(4) Support for Legacy Clients: Deploying new hardware and upgrading NIC software on the clients is expensive and impractical [19]. We therefore require our design to support continuous mobility while at the same time being backwards compatible with existing IEEE 802.11 standards.

(5) Realizable on Commodity Hardware: In order to make the proposed system practical, we require the fewest possible changes to the existing WLAN infrastructure. Clearly, specialized hardware and software defined radio (SDR) are ill-suited for this purpose. We therefore require that the proposed architecture be realizable on commodity hardware.

3 Architecture

We now describe the architecture of Overcast. Our discussion begins with a broad overview of the system and we then cover specific details. Overcast is a centralized WLAN architecture. In such a design, APs are connected to a Central Controller over a wired Ethernet backplane. The controller is comprised of two parts, one that manages the APs configurations (*control plane*) and the other that performs traffic management/shaping (*data plane*) in the network. Note that, as indicated in [4], an edge router strategically placed between the wired and wireless networks with precise knowledge of a significant fraction of traffic makes the controller's second function possible.

Overcast is a *single channel* architecture [25] where orthogonal channels are used to add capacity instead of preventing inter-cell interference. Each AP can concurrently serve clients on all orthogonal channels by using a separate radio for each. A key advantage of this design is the ability to support *zero-latency* hand-offs. We setup all AP's on one channel to have the same MAC address and ESSID. Hence from the client's perspective, the entire network is a single virtual AP (as shown in Figure 1). Session state (for each client) is maintained at the controller. Therefore, any mobility within the enterprise does not require re-association. Hand-offs are handled entirely by the infrastructure as described in Section 4.2. We note that some commercial vendors such as Meru [26] also employ a similar approach. However, Overcast leverages an online interference mapping engine as part of its design (Section 4.1). This allows it to effectively manage interference in congested scenarios and also rapidly detect interference for clients during periods of mobility.

Clients begin communicating with the infrastructure by first requesting association to the network (as shown in Figure 2). The association request is received by one or more APs in the vicinity of the client which in turn is forwarded to the controller. The controller selects an AP (based on some AP selection strategy) and has that AP respond to the client with an association response. AP selection can be based on a number of criteria as discussed in Section 4.2. Note that although one AP is elected to communicate with the client, the session terminates at the controller, not the AP. This implies for example that the WEP session keys are exchanged between the client and controller as opposed to the AP. Once associated, the client is free to move throughout the entire network. If the client begins to exit the service area of one AP, the controller detects this event (through measurements) and moves the client to a neighbouring, more suitable AP. The client is unaware of this change and assumes it is still receiving packets from the same AP. There are a number of steps to ensure that a client can be migrated to a neighbouring AP that doesn't degrade it's performance. These steps are covered in greater detail in Section 4.2.

Once the client is associated to the network, a minimum QoS requirement (Section 4.1 describes how we obtain these requirements for the client) needs to be maintained at all times as the client moves about within the network. Furthermore, there must always be at least one AP available to service the client at all locations in the building. This is guaranteed through a dense (and uniform) deployment of APs in the enterprise, similar to DenseAP [19]. Prior single channel designs try to avoid dense deployments so as to reduce cross cell interference [26]. We handle such interference through online interference detection and rapid network re-configuration.

In a single channel architecture, co-located clients are expected to use the same channel to communicate. In such scenarios, link interference is inevitable, especially for streaming/realtime traffic [10]. There are a number of ways to address this problem. We can separate conflicting traffic using TDMA-based packet/flow scheduling. We can also eliminate interference by adapting the modulation rate (and hence the amount of forward error correction (FEC)) that is applied to packets. Finally, we can also use adaptive power control to minimize conflicts in the network[1]. We cover the pros and cons of each optimization in more detail in Section 4.2.

---

[1] Power control must be carefully performed, since poorly designed power control strategies could lead to dead-spots in the enterprise

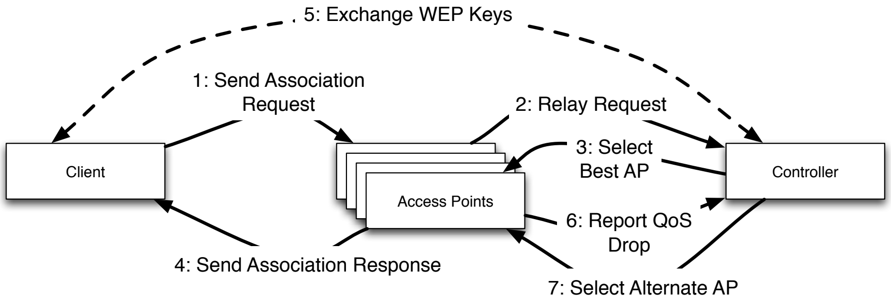

Fig. 2. The sequence of events that occur when a client first associates to the network

4 Infrastructure Details

We now describe some specific aspects of the Overcast architecture in greater detail. Our discussion is divided into measurement collection and network optimization (or control operations).

4.1 Measurement Collection

Link Quality Measurements Link quality measurements are essential to ensure that chosen links provide adequate QoS. Because we need to support legacy clients, we cannot get RSSI, Packet Error Rate (PER) or other such measurements from the client. Prior approaches [5] have addressed this issue by using AP side measurements and assuming channel symmetry. This assumption does not hold in real world deployments and is especially problematic for realtime traffic that is bi-directional and has hard QoS requirements.

In Overcast, we determine downlink link quality by maintaining a running average of PER for each active AP-Client link. We note that real time traffic can tolerate a small amount of packet loss without degrading in quality (e.g. 2% loss rate is generally accepted to be imperceptible in audio streams). Using this fact, we send a small percentage of packets from neighbouring APs not instructed to serve the client (but within it's range). Using these packet transmissions, we can compute the downlink PER rate for such neighbouring APs.

To measure uplink PER, we note that a realtime traffic source will generate a single packet every ‘t’ seconds[2], where t is the packetization interval of the encoding scheme. Using this fact, we can count the number of missed packets and consequently the uplink PER. In fact, since all access points listen to all traffic, we can passively build uplink PER information for all APs in the vicinity of a client. In this way, we obtain information about both the uplink and downlink quality to the selected AP as well as neighbouring alternate APs. This allows us to judge if a migration is needed and which alternate AP(s) can meet the client's QoS requirements.

---

[2] We assume there is no silence suppression or packet aggregation

Interference Mapping - Micro-Probing At the heart of the Overcast architecture is an interference mapping (IM) engine. The IM engine is responsible for discovering interference (or conflict) between AP-AP and AP-client links in the network. There are numerous approaches proposed in the literature to measure interference between links in a network. We choose micro-probing due to its ability to map interference in an online network without client feedback. Micro-probing operates by initiating a series of "micro-experiments" between pairs of links to determine the nature of interference between them. This approach is able to generate an interference map of a moderately sized network (of approximately 20 nodes) on the order of seconds. We elide further details of micro-probing due to space limitations and refer the reader to [3] for additional details.

Opportunistic Experimentation: Prior work has shown that interference measurements are effected by the channel on which they are conducted [21]. Hence we must perform interference measurements on the same channel during the normal operation of the network. To allow for this, we propose *opportunistic* measurements during idle periods. Under bursty traffic, predicting idle time on a channel is almost impossible. However for periodic realtime traffic, APs can learn traffic patterns and inter-leave interference experiments with data traffic.

What-If Analysis for Conflicts: Overcast uses conflict information to migrate clients from one AP to another. Hence, we need to identify not only conflicts in the current (AP-Client) configuration but also those that would arise after a migration. We term such conflicts *What If conflicts*. We leverage the association free nature of our network to conduct experiments for What-if conflicts between the client and the neighbouring APs. Having this information prior to migration allows us to ensure that a migration will not violate QoS requirements.

Learning (QoS) Requirements To ensure that client QoS requirements are met by Overcast, we must elicit them from the clients. However, legacy clients do not report such information. By identifying the encoding scheme by means of packet inspection, we can ascertain the QoS requirements of an application running over the network. However, we do assume that the client is generating only real time traffic and using a well known encoding scheme.

Implicit in the discussion above is the assumption that our measurements are not affected by bursty traffic. To ensure this, Overcast separates channels into realtime and bursty channels. Clients are allocated a channel based on their traffic characteristics. Initially we assume that a client is generating realtime traffic but if we observe variable bitrate traffic, we move the client over to a bursty traffic channel (using methods described in [19]). Channel switching has some associated delay but since a bursty client is generating delay insensitive traffic anyway, QoS guarantees are not required. The separation of traffic onto different channels may seem odd. However, prior work shows that bursty and realtime traffic do not co-exist well. For example, Garg et al[13] show that a $74Kbps$ realtime stream reduces the throughput of a parallel bursty client by as much as 900 Kbps.

4.2 Control Operations

Seamless Handoffs Hand-offs in 802.11 networks encompass three operations: 1) Selecting a new AP, 2) Transferring context and 3) Disassociating from the old AP and re-associating with the new AP. The single channel design precludes the need for operations 2 and 3. All contextual information (including encryption keys for WEP) are stored at the controller and the client is agnostic to hand-offs. The only handoff operation that is necessary is AP selection, which we briefly discuss next.

AP Selection: The AP selection algorithm is crucial in Overcast as it has a significant impact on the QoS that a client eventually receives. Prior proposals use many diverse metrics for AP selection [24]. We note that the metrics of relevance for real time traffic are loss rate, delay Jitter and end-to-end delay. Loss rate is a function of link quality and interference. Delay jitter is caused due to variability in channel access delay caused by contention and the number of clients served by each access point. End-to-end delay can be ignored because the delay budget of realtime applications is large enough (up to $200ms$) to make the end-to-end delay essentially negligible. Another metric of interest is the number of users on a particular channel. This is important because we would like to keep a client on the same channel throughout it's connection session. Therefore, load balancing across channels during AP selection is crucial. We combine these metrics in a utility function shown in Equation 1, where 'Q' is the channel *quality* in isolation, 'I' is the number of interfering links, 'L' is the access point load and 'C' is the channel load.

$$U = F(Q) + F(I) + F(L) + F(C) \tag{1}$$

Each of the functions F(Q), F(I), F(L) and F(C) quantify the effect of the corresponding metric on utility and normalize the value for calculation. The details of the functions are not discussed here but we note that the function can be made as complex or as simple as desired. For example, prior work [13] has noted that a single access point using 11Mbps can support up to 5 VoIP clients with little or not drop in voice quality. However adding a sixth client causes a sudden fall in quality. Hence we can define the value of F(L) to be 1 for up to 5 clients and as a rapidly decaying function as the number of clients increases beyond 5. This is formalized in Equation 2 where α is a normalizing constant.

$$F(L) = \begin{cases} \alpha(1) & \text{if } L \leq 5, \\ \alpha(1/L) & \text{if } L > 5, \end{cases} \tag{2}$$

Note that when a client first joins the network, we have no information about the AP to Client link on which to base AP selection. We therefore use RSSI measurements at the AP and assume channel symmetry. The initial selection can be refined as soon as additional information is available (as discussed in Section 4.1). Also, if the network has no available capacity on one channel within a client's neighbourhood (if for instance, the client moves), it may be forced to migrate the client to an alternate channel. To support this functionality, Overcast implements cell breathing [6] to force clients to begin scanning on other channels. Once a client begins scanning, we use hidden SSIDs, as proposed in [19], to force the client to switch channels.

Network Optimizations We now briefly discuss three network optimization schemes that can be implemented on Overcast to enhance it's performance.

Packet/Flow Scheduling: Realtime traffic is periodic and therefore predictable, a fact we can leverage to appropriately schedule traffic and avoid interference. For example, commonly used voice encoders generate a single, small packet (up to 100 bytes) every 10 to 30 ms. Even at low bitrates, a packet this size is transmitted in less then a millisecond. As stated earlier, the current state-of-the art supports only up to 5 clients per AP. Therefore, there is significant excess airtime to separate conflicting traffic. We can coordinate AP transmissions to mitigate the effects of interference on the downlink. For uplink traffic, we can use CTS-to-self to "clear-the-air" for a particular client.

Modulation Rate Adaptation: Rate adaptation algorithms typically try to use the highest sustainable bitrate. However, this comes at the cost of a potentially higher packet loss rate and susceptibility to interference. Realtime applications are sensitive to packet loss but do not require too much bandwidth. In such scenarios, we prefer to use a lower more robust rate even if a higher rate can be sustained. For uplink traffic, we can not control the rate explicitly but can use some indirect techniques, as discussed in Section 7.

Transmit Power Control (TPC): Many proposed enterprise WLANs support transmit power control for AP load balancing and interference mitigation. Overcast performs load balancing by managing AP-client associations dynamically from the controller. Therefore, TPC helps only to mitigate interference between neighbouring links in conflict. A number of strategies can be used to implement TPC, ranging from per-cell to per-client TPC. However, caution should be exercised when using TPC since poorly designed algorithms can lead to link starvation, increased contention and in the worst case, dead-spots or regions of no coverage.

4.3 Implementation Status

Over the past few months, we have been setting up the infrastructure and protocols for Overcast. We have deployed a 40 node testbed [2] on the 2nd and 3rd floors of the Computer Science building at the University of Waterloo. We have implemented MAC Cloning within the firmware of the Intel 2915ABG radio. This allows us to dynamically assign an arbitrary MAC address to our APs. We have also implemented the micro-probing interference mapping engine for our testbed [3]. Preliminary results confirm that we can quickly and accurately measure interference between links in the network. Furthermore, we have also used MAC cloning to implement *What-if conflict* detection, as discussed in Section 4.1. Currently, we are exploring some of the optimization approaches discussed in Section 4.2.

5 Example Application

We now describe an application that can benefit from the continuous mobility support provided by Overcast. This is a prototypical real-time streaming application and serves to highlight the challenges faced in supporting such applications.

Almost all businesses today deploy an enterprise-wide telephone system so that employees can communicate with one another. It is interesting to note that such systems are common even with the mass adoption of cellular phones. This is because cellular technology is not suitable for enterprise telephony today. Cellular technology is not cost-effective (for the enterprise), beyond administrative control, and also has coverage problems inside large office buildings [11].

The wired (PBX-based) telephone system on the other hand prevents user mobility which is desired by today's mobile workforce. This has created an interest for alternative wireless telephony systems [9]. Coincidently, many businesses already deploy enterprise-wide, high-speed WLAN's. This has prompted research into supporting voice over WLANs (VoWLANs [9]). However, the VoWLANs of today typically only support nomadic clients because cell hand-offs are problematic. Furthermore, the call capacity in these WLANs is typically limited to about 5 clients per AP because they do not handle interference and contention well [10]. Some research has proposed solutions to these problems but they require client modifications and hence are hard to deploy. Using Overcast, a user can associate her WiFi compatible smartphone to the network and run a VoIP service of her choice. After the initial association, the user is free to move anywhere within the coverage area of the network during an ongoing call[3]. By managing interference and providing multiple orthogonal channels at each access point, Overcast ensures that a much larger number of clients can be simultaneously supported, compared to existing VoWLANs.

6 Related Work

There is a large body of literature that proposes optimization schemes for enterprise WLANs. However, we limit our discussion to work on optimizing continuous mobility and providing realtime support on WLANs.

Continuous mobility has been studied in the context of minimizing hand-off latencies in wireless networks. Some prominent work, including [23] proposes neighbour graphs to reduce client scanning time. However, these approaches require offline graph computation, which is cumbersome and prone to inaccuracy. Ramani et al. [22] propose to synchronize beacon transmissions of neighbouring APs to reduce overall scanning time. However this approach requires client modifications. In summary, prior work attempts to minimize re-association overhead by reducing scanning duration. In contrast, Overcast uses MAC address cloning to emulate a single virtual AP that eliminates these delays altogether.

We now briefly cover realtime traffic support over 802.11 networks in prior work. We focus on VoIP as its characteristics have been explored extensively over the last decade [13] [7] [14]. 802.11 networks are notorious for poorly supporting simultaneous VoIP connections [13]. Many proposals have been put forth to improve the dismal call capacity of WLANs [7] [14]. However, these require modifications to the client's MAC layer. Furthermore, most current approaches are only designed for the single cell

---

[3] We note that maintaining an always-on connection to the network can drain the battery of a handheld device [1]. This problem is common to any handheld device that uses the WiFi interface and is not exclusive to Overcast.

scenario. Recently, it was shown that multi-cell deployments support only 2 active sessions per AP on average [10]. This is a three times reduction compared to the single cell case, illustrating the poor support that existing multi-AP WLANs provide for realtime applications.

We now move to works that propose architectures to support continuous mobility and VoIP. SMesh [5] proposes a system for fast, seamless handoffs in wireless mesh networks (WMNs). Each AP advertises a common gateway IP address and BSSID, avoiding DHCP overheads during handoff. However, SMesh requires clients to operate in ad hoc mode, which is not the default 802.11 client behavior. Another architecture, DenseAP [19], uses dense AP deployments to improve performance in enterprise WLANs. Though interesting in principle, DenseAP still incurs an average delay of about 1.5 seconds during client handoffs, which is unacceptable for realtime applications such as VoIP that have a delay budget of up to $200ms$. MDG [8] explores techniques such as channel assignment, power control and client association to improve enterprise network performance. However, MDG requires client modifications which makes it hard to deploy in practice. Trantor [20] was recently proposed as a clean-slate design to enterprise WLANs that also supports realtime applications. However, like other architectures, the benefits from Trantor are only realized with client modifications. Ahmed et al [4] propose centralized scheduling to mitigate the effect of interference in enterprise WLANs. However, their approach does not support continuous mobility and therefore is not applicable for the target scenarios presented in this paper.

Finally, some commercial vendors (e.g. Meru [26], Extricom [12]) also claim to support realtime traffic. However, little is known about their solutions and there is no independent verification of their claims. Furthermore, there is also speculation on whether some of these vendors are even 802.11 standards compliant. In addition, our private discussions with one such vendor revealed that interference mitigation is still a challenge for applications such as VoIP and video.

7 Conclusions and Future Work

In this paper, we present Overcast, an architecture supporting continuous mobility for realtime applications, with support for legacy clients. The design espouses a single channel architecture and includes novel techniques for interference management. We present the architectural details of Overcast, such as mechanisms for accurate measurement of link quality, single virtual AP emulation and application QoS determination. Using a 40 node wireless testbed at the University of Waterloo, we have implemented and successfully tested several features of Overcast and are well on our way to building a comprehensive WLAN system. However, there are still many interesting challenges and avenues for future work:

Security: Overcast has a set of unique security challenges which need to be analyzed and addressed. For example the use of MAC address cloning makes it harder to identify and isolate rogue access points.

Client Control: We would like to centralize as much network management as possible. However, the infrastructure has no control over client-side parameters such as data rate and transmit power. However, there are ways to influence the decisions of clients by strategically modifying the APs' configuration and intelligently scheduling traffic. We are currently exploring several such techniques.

Traffic Separation: Due to the lack of a comprehensive scheme for co-locating bursty and realtime traffic, we are forced to sequester realtime traffic to its own channel(s). However, we are exploring methods to allow realtime traffic to co-exist with bursty traffic while ensuring that per-cell throughput is not significantly affected.

References

1. Y. Agarwal, R. Chandra, A. Wolman, P. Bahl, K. Chin, and R. Gupta. Wireless wakeups revisited: energy management for voip over wi-fi smartphones. In *MobiSys*, 2007.
2. N. Ahmed and U. Ismail. Designing a high performance testbed for centralized control.
3. N. Ahmed, U. Ismail, S. Keshav, and K. Papagiannaki. Online estimation of rf interference. In *ACM CoNext*, 2008.
4. N. Ahmed, V. Shrivastava, A. Mishra, S. Banerjee, S. Keshav, and K. Papagiannaki. Interference mitigation in enterprise wlans through speculative scheduling. In *ACM MobiCom*, 2007.
5. Y. Amir, C. Danilov, M. Hilsdale, R. Musăloiu-Elefteri, and N. Rivera. Fast handoff for seamless wireless mesh networks. In *MobiSys*, 2006.
6. P. V. Bahl, M. T. Hajiaghayi, K. Jain, S. V. Mirrokni, L. Qiu, and A. Saberi. Cell breathing in wireless lans: Algorithms and evaluation. *IEEE Transactions on Mobile Computing*, 6(2):164–178, 2007.
7. R. O. Baldwin, I. Nathaniel J. Davis, S. F. Midkiff, and R. A. Raines. Packetized voice transmission using rt-mac, a wireless real-time medium access control protocol. *SIGMOBILE Mob. Comput. Commun. Rev.*, 5(3):11–25, 2001.
8. I. Broustis, K. Papagiannaki, S. V. Krishnamurthy, M. Faloutsos, and V. Mhatre. Mdg: measurement-driven guidelines for 802.11 wlan design. In *MobiCom*, 2007.
9. L. Cai, Y. Xiao, S. Shen, L. Cai, and J. Mark. Voip over wlan: Voice capacity, admission control, qos, and mac. 2006.
10. A. Chan and S. Liew. Voip capacity over multiple ieee 802.11 wlans. In *IEEE ICC*, 2007.
11. A. de Toledo and A. Turkmani. Propagation into and within buildings at 900, 1800 and 2300 mhz. In *Vehicular Technology Conference*. IEEE, 1992.
12. Exricom Inc. Wireless lan switch datasheet. URL: http://tinyurl.com/4dmrns.
13. S. Garg and M. Kappes. An experimental study of throughput for udp and voip traffic in ieee 802.11b networks. In *IEEE WCNC*, 2003.
14. T. Hiraguri, T. Ichikawa, M. Iizuka, and M. Morikura. Novel multiple access protocol for voice over ip in wireless lan. In *IEEE Int. Symp. Computers and Communications*, 2002.
15. A. P. Jardosh, K. Mittal, K. N. Ramachandran, E. M. Belding, and K. C. Almeroth. Iqu: practical queue-based user association management for wlans. In *MobiCom*, 2006.
16. J. Kristiansson and P. Parnes. Application-layer mobility support for streaming real-time media. *WCNC. 2004 IEEE*, 1:268–273 Vol.1, March 2004.
17. A. Mishra, S. Rayanchu, D. Agrawal, and S. Banerjee. Supporting continuous mobility through multi-rate wireless packetization. In *HotMobile*, pages 33–37, New York, NY, USA, 2008. ACM.
18. S. Mishra, A. Sahai, and R. Brodersen. Cooperative sensing among cognitive radios. In *ICC 2006*, 2006.
19. R. Murty, J. Padhye, R. Chandra, A. Wolman, and B. Zill. Designing high performance enterprise wi-fi networks. In *NSDI*, 2008.

20. R. Murty, A. Wolman, J. Padhye, and M. Welsh. An architecture for extensible wireless lans. In *ACM HotNets-VII*, 2008.
21. D. Niculescu. Interference map for 802.11 networks. In *IMC*, 2007.
22. I. Ramani and S. Savage. Syncscan:practical fast handoff for 802.11 infrastructure networks. In *IEEE Infocom*, 2005.
23. M. Shin, A. Mishra, and W. A. Arbaugh. Improving the latency of 802.11 hand-offs using neighbor graphs. In *MobiSys*, 2004.
24. K. Sundaresan and K. Papagiannaki. The need for cross-layer information in access point selection algorithms. In *IMC*, 2006.
25. White-paper from Aruba Networks. Wlan rf architecture primer: Single-channel and adaptive multi-channel models. URL: http://tinyurl.com/4k7offf.
26. White-paper from Meru Networks. Virtual cells: The only scalable multichannel deployment. URL: http://tinyurl.com/523wet.