Wi-Fi Neighborcast: Enabling Communication Among Nearby Clients

Ranveer Chandra
Microsoft Research
Redmond, WA 98052
ranveer@microsoft.com

Jitendra Padhye
Microsoft Research
Redmond, WA 98052
padhye@microsoft.com

Lenin Ravindranath
Microsoft Research
Bangalore, India
leninr@microsoft.com

1. INTRODUCTION

The density of Wi-Fi Access Points (APs) has increased rapidly in urban downtowns [7] and enterprise and campus networks [3]. In most scenarios, a Wi-Fi client has the choice to associate with more than one available AP [3, 12]. Therefore, it is increasingly common to find physically nearby Wi-Fi clients that are associated with different APs.

A number of applications can benefit from the ability to send and receive information from nearby clients. For example, clients can perform better AP selection if they have knowledge about the performance of clients associated to neighboring APs. Applications such as "buddies near me" can discover clients that are nearby but associated to another AP. Similarly, geocasting [11] based applications can reach out to more clients in the same region than being limited to only one wireless network.

Most existing schemes to enable these applications require significant infrastructure changes or extensive manual profiling. For example, 802.11k modifies the APs and client drivers for better AP selection. BeaconStuffing [4, 10] modifies the APs to send information about load on its network. Existing implementations of "buddies near me", such as AOL's buddy list [1], require extensive wardriving.

In this paper, we propose a novel mechanism, called Neighborcast. Using this mechanism nearby clients can communicate with each other even when they are associated to different APs. We leverage multicast to achieve this functionality. Each AP is assigned a globally unique multicast group ID, and a Wi-Fi client with an IP address joins the multicast group corresponding to all APs around it. We implement the multicast functionality using either IP multicast, Application Level Multicast or a web-server based scheme using RSS feeds.

Unlike prior work in this area, neighborcast does not require any modifications to the APs or kernel-level software changes at the clients. Furthermore, by using IP multicast or Application Level Multicast, all neighborcast traffic generated by a Wi-Fi client is local, i.e. propagates only to nearby APs. Hence, neighborcast is more scalable than a completely centralized publish/subscribe based scheme.

2. NEIGHBORCAST

Neighborcast enables Wi-Fi clients to discover and communicate with other nearby Wi-Fi clients, irrespective of the AP they are associated with. Our goal is to achieve the neighborcast functionality without modifying the APs.

For simplicity, we define two Wi-Fi clients to be near if they hear beacons from a common AP. [1] Using this definition, two clients are near even when they are associated to different APs (on different frequency channels). Furthermore, using the above definition, neighborcast achieves more range than Wi-Fi since two clients can now communicate even when they are not in communication range of each other, as long as they hear a common AP.

Neighborcast builds on the concept of multicast groups, and works as follows. It assigns a globally unique group ID to every AP, which is derived from its unique BSSID (a 6 byte identifier). A Wi-Fi client subscribes to (joins) the multicast group of all APs it discovers as part of the scanning process. It publishes updates (presence, performance) on the multicast group of the AP it is associated with. We illustrate the neighborcast mechanism in Figure 1. C1 is associated to AP1, and C2 and C3 are associated to different APs, although they are in the range of AP1. Neighborcast allows C1, C2 and C3 to communicate with each other using a common multicast group (corresponding to AP1).

Using the above mechanism, neighborcast constructs an IP overlay among physically near wireless clients. This overlay spans different Wi-Fi networks and frequency channels. Therefore, an update published by a client on its AP's multicast group is received by all clients in range of the AP even when they are associated to another AP. Furthermore, as we show in Section 3.3, neighborcast can be extended to estimate the relative distances of nearby Wi-Fi clients.

[1]Note that a Wi-Fi client processes beacons from APs it is not associated with as part of the scanning process [6].

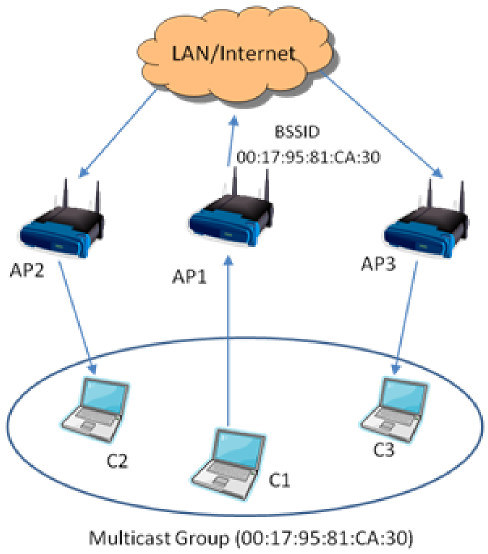

Figure 1: Neighborcast Mechanism: Clients C1, C2 and C3 are all in range of AP1, and hence are part of the same multicast group although they are associated to different APs.

We describe three approaches to implement neighborcast in the rest of this section. Each scheme differs in the overhead of publish/subscribe and the ease with which neighborcast can be deployed.

2.1 Using IP Multicast

In this scheme, the multicast group identifier of an AP corresponds to an IP multicast address (for example, 228.xxx.xxx.xxx), which is derived from the AP's globally unique BSSID. To achieve this mapping from a 6-byte BSSID to 3 bytes in the multicast IP address, we simply reuse the last 3 bytes of the AP's BSSID. For example, a BSSID of 00:17:95:81:CA:30 will correspond to a multicast IP address of 228.129.202.48 (as 0x81 = 129, 0xCA = 202, 0x30 = 48).

We expect most APs in the same subnet to be from the same vendor and hence have the same BSSID prefix (first 3 bytes). Therefore, in most scenarios, our simple algorithm will yield unique multicast IP addresses for each AP. However, we note that the above mapping might sometimes lead to IP address collisions. We are currently exploring stronger hash functions to reduce the probability of these collisions.

Using this approach, a client maps the BSSID of every AP it hears as part of the scanning process to a corresponding multicast IP address. It then subscribes to all the above multicast groups. It publishes information on the IP multicast group of the AP it is associated with.

This approach has several advantages. It is completely distributed, and does not require extra infrastructure or modifications to existing standards and protocols. Furthermore, traffic generated by neighborcast is local, i.e. limited to nearby APs and does not travel far in the network. Consequently, this technique scales well with the number of APs (and hence multicast groups) in the network.

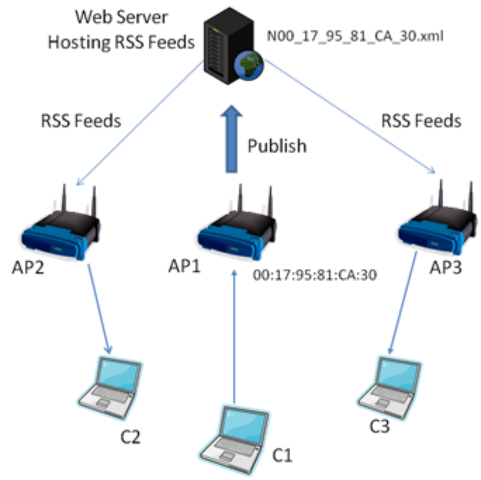

Figure 2: Neighborcast using RSS Feeds

A concern about the usefulness of this approach stems from the limited deployment of IP multicast over the Internet. However, we note that IP multicast has been widely deployed at the edge: in enterprise and campus networks. Therefore, in scenarios where all APs are part of the same subnet, we expect neighborcast to be implemented over IP multicast. For wide area Internet, following two techniques can be used instead.

2.2 Using Application Level Multicast

Given the limited deployment of IP multicast on the Internet, we can implement neighborcast using Application Level Multicast (ALM) [2] in scenarios when not all APs are part of the same subnet, for example in residential and shopping areas. This scheme requires a rendezvous server on the Internet for discovery and management of group membership across multicast groups [2].

We use a simple ALM scheme similar to ALMI [13] to implement multicast. While ALMI builds a spanning tree for routing, we assume connection between every pair of clients in the multicast group. To join a multicast group, the client contacts the rendezvous server (central controller), which replies with a list of IP addresses of clients that are members of the group. It also sends an update to other clients in the group with the new client's IP Address. Clients multicast data by unicasting separately to all clients in the multicast group. Similarly when a client leaves a group, it informs the rendezvous server, which then sends this update to other clients in the group. Clients that are part of a multicast group also send periodic heartbeat messages to the rendezvous server to inform them their existence.

A drawback of the ALM technique is the need of a rendezvous server to handle control messages. However, note that only the control traffic goes through the server, and not the data traffic. We are currently exploring an alternative ALM technique using a separate rendezvous server per AP in order to limit the overhead of neighborcast's control messages.

2.3 Using RSS Feeds

Another mechanism to implement neighborcast's publish subscribe scheme is using RSS feeds. We deploy a web server on the Internet, on which each AP has a corresponding web feed. A web feed is identified by an XML file. The link for the XML file is obtained using the AP's BSSID. For example, the web feed of the AP with BSSID 00:17:95:81:CA:30 is *http://webserver/N00_17_95_81_CA_30.xml* Clients associated to the AP publish items on the web server. Clients in range of an AP use its BSSID (and hence the web feed link) to subscribe to the RSS feeds from the AP. We illustrate this mechanism in Figure 2.

This approach is the easiest to deploy. The communication with the RSS server takes place over HTTP. Therefore, it requires no support from the APs, minimum software modifications at the clients, and does not require multicast support in any form. The drawback of this approach, however, is that all neighborcast traffic goes through the web server, raising scalability concerns.

3. APPLICATIONS

In this section, we discuss a few applications that can benefit from Neighborcast.

3.1 Improving AP selection

As more public WLANs are deployed, on many occasions a client can choose from multiple APs. Similarly, in corporate environment, APs are being deployed ever more densely, and a client can usually choose between multiple APs.

The question of which AP to associate with is a difficult one, and has recently received much attention from the research community [3]. Most existing wireless clients simply rank APs based on signal strength of the beacons from the AP, and generally associates to the AP with the highest signal strength. It is easy to see that in a hot-spot environment, this policy will not work well, as many clients may associate with the same AP. Ideally, clients could take into account load on the AP, interference near AP, and the signal strength of their packets at the AP into account when making the association decision. Other factors such as load on other APs in the system may also be factored into the decision.

Many researchers have proposed various ways of collecting and distributing this information. It includes BeaconStuffing [4], MDG [3] as well as protocols such as 802.11k. The neighborcast approach provides a simple way by which a client may learn about other nearby clients and the performance they are experiencing at their respective APs, before making the association decision. The following is a brief outline of how the association process works with neighborcast.

For the purposes of this discussion, we will assume that the clients use IP multicast for communication. However, the ALM or RSS feed approaches work just as well. We assume that every client in the WLAN is running the Neighborcast Load Balancing application. When a client is associated with an AP, it derives the multicast IP address of the group for that AP and joins the multicast group. It periodically publishes information about channel conditions and load on the multicast group. The

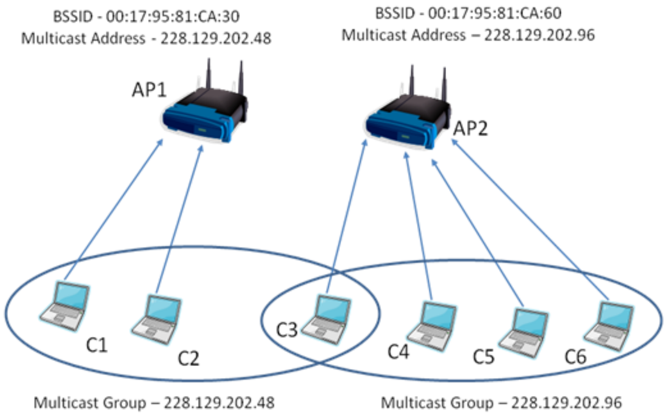

Figure 3: Improving User Association in WLAN using Neighborcast

information includes signal strength of APs beacons as seen by the clients, a summary of traffic generated by the client in past one minute, and an estimate of RTT to the first hop gateway on the subnet. The client also scans for APs around it and subscribes to their multicast groups as well. The client now receives information about (i) load generated by other clients on the AP it is associated with (ii) channel conditions and load information on other nearby APs. Based on this information, the client may wish to switch to another AP. We do not advocate a specific algorithm for selecting the AP - our approach will work with most existing algorithms.

Figure 3 shows an example scenario where C1 and C2 are associated to AP1 and clients from C3 to C6 are associated to AP2. Client C3 is in the range of AP1 and hence subscribes to its multicast group. Using the information published in both the groups, C3 learns of load on both APs and their channel characteristics. Based on this knowledge, C3 will be able to make a more informed decision about whether to associate with AP1 or AP2.

Even if only some of the clients in the system run the neighborcast application, an AP selection algorithm may be able to make use of partial information. More information can be made available if some of the clients are run in promiscuous mode and report on traffic generated by non-neighborcast enabled clients.

The neighborcast approach is completely client driven, requires no infrastructure support and no modification at the APs. It can be easily used in existing WLANs.

3.2 Cooperative Problem Diagnosis

The AP selection application can be generalized to cooperative problem diagnosis. Clients can use neighborcast to cooperate with other clients nearby to diagnose and localize Wi-Fi problems in an automated manner. The algorithms would be similar those used in [5], however, neighborcast can be used to propagate information. For example, when a client is experiencing poor connectivity, it can collaborate with other clients associated to its AP and clients associated to nearby APs to diagnose and localize the problem. The problem can be localized to the gateway or to the AP or to the Wi-Fi card of the client or to the client's physical location.

3.3 Buddies Near Me

Social Networking, i.e. networking people based on shared context is rapidly gaining popularity. Services such as AOL's "Buddies near me" [1] allow people to determine whether any of their friends are in the neighborhood. This information can be leveraged by a variety of applications such as gaming.

Many of these location services rely on large centralized databases, compiled by extensive war driving. However, we note that for these applications, it is not necessary to know the precise location of each client. We only need to determine that a group of clients is close to each other. Neighborcast can simplify this process.

Clients advertise about their existence on multicast groups of all the APs they hear. In these advertisements they report the signal strengths of all the APs they can hear. We make two observations about this information. First, if two clients hear the same AP with high signal strength, we can say with a high confidence that the two clients are close to each other. Second, if two clients hear an AP with low signal strength they may be either far away from each other or close to each other. In other words, we have low confidence in any "nearness" prediction we make with this information.

We are working on a new metric based on these two observations for characterizing the distance between a pair of nodes, and the confidence we have in this prediction. By combining these metrics across a group of clients in a transitive manner, we can determine whether the clients in the group are close to each other. This scheme is similar to the approach used by the NearMe Proximity Server [9].

Note that this method does not provide information about where the clients are: it simply tells us whether clients are close to each other. Also, we do not need to know the locations of the APs themselves, so there is no need for war driving.

4. PRELIMINARY RESULTS

We have implemented neighborcast using IP Multicast and Application Level Multicast as described in Section 2. We tested it in a subnet with 10 APs. We deployed 5 clients that were associated to 3 different APs. All clients could hear one common AP and hence were part of that AP's multicast group.

Our preliminary results show that IP Multicast takes less time to deliver a message to a group compared to ALM. However, IP multicast is unreliable, since APs do not send ACKs for the transmitted packets. On the other hand, we have implemented ALM using TCP connections, so there is no packet loss. We are working on more detailed experiments to identify when one approach performs better than the other.

5. CONCLUSION AND FUTURE WORK

We have proposed a novel framework called neighborcast, using which nearby clients can communicate with each other even when they are associated to different APs. We described three different ways to implement neighborcast and compared the efficiency of each approach. We showed how the neighborcast functionality can benefit three useful applications: AP selection, fault diagnosis and "buddies near me".

In the future, we plan to build the three applications of neighborcast described in Section 3. We are currently working on two key issues that need to be addressed for building these applications: authentication and validation, i.e. only authorized clients should be able to join a multicast group and malicious clients should not be allowed to publish incorrect data. For some applications, such as AOL's Buddies Near Me, we expect the application using neighborcast to perform authentication and validation. We are also exploring the use of reputation-based mechanisms [8] to solve the security concerns for other applications.

6. REFERENCES

[1] AOL Buddies Near Me feature. http://www.msnbc.msn.com/id/17689221/.

[2] S. Banerjee, B. Bhattacharjee, and C. Kommareddy. Scalable application layer multicast. Technical report, UMIACS TR-2002.

[3] I. Broustis, K. Papagiannaki, S. V. Krishnamurthy, M. Faloutsos, and V. Mhatre. MDG: measurement-driven guidelines for 802.11 wlan design. In *Proceedings of the Annual ACM International Conference on Mobile Computing (MobiCom)*, 2007.

[4] R. Chandra, J. Padhye, L. Ravindranath, and A. Wolman. Beacon-Stuffing: Wi-Fi Without Associations. In *Proceedings of the Workshop on Mobile Computing Systems and Applications (HotMobile)*, 2007.

[5] R. Chandra, V. N. Padmanabhan, and M. Zhang. WiFiProfiler: Cooperative diagnosis in wireless LANs. In *Proceedings of the Annual ACM/USENIX International Conference on Mobile Systems, Applications and Services (MobiSys)*, 2006.

[6] IEEE802.11b/D3.0. Wireless LAN Medium Access Control(MAC) and Physical (PHY) Layer Specification: High Speed Physical Layer Extensions in the 2.4 GHz Band, 1999.

[7] G. Judd, A. Akella, S. Seshan, and P. Steenkiste. Self-Management in Chaotic Wireless Deployments. In *Proceedings of the Annual ACM International Conference on Mobile Computing (MobiCom)*, 2005.

[8] P. Kollock. The Production of Trust in Online Markets. In *Advances in Group Processes (Vol. 16)*, 1999.

[9] J. Krumm and K. Hinckley. The NearMe Wireless Proximity Server. In *Proceedings of International Conference on Ubiquitous Computing (UbiComp)*, September 2004.

[10] V. Mhatre and K. Papagiannaki. Using Smart Triggers for Improved User Performance in 802.11 Wireless Networks. In *Proceedings of the Annual ACM/USENIX International Conference on Mobile Systems, Applications and Services (MobiSys)*, 2006.

[11] J. C. Navas and T. Imielinski. Geocast - Geographic Addressing and Routing. In *Proceedings of the Annual ACM International Conference on Mobile Computing*

(MobiCom), 1997.

[12] A. Nicholson, Y. Chawathe, M. Chen, B. Noble, and D. Wetherall. Improved Access Point Selection. In *Proceedings of the Annual ACM/USENIX International Conference on Mobile Systems, Applications and Services (MobiSys)*, 2006.

[13] D. Pendarakis, S. Shi, D. Verma, and M. Waldvogel. ALMI: An Application Level Multicast Infrastructure. In *3rd Usenix Symposium on Internet Technologies and Systems*, 2001.