Approximation Algorithms for Scheduling Real-Time Multicast Flows in Wireless LANs

Yigal Bejerano[†] Dongwook Lee[‡] Prasun Sinha[⋆] Lisa Zhang[†]
[†]Bell-Labs, Alcatel-Lucent. [‡]Samsung Tech., Seoul, Korea. [⋆]CSE, Ohio-State University.

***Abstract*—*Abstract*—In recent years, numerous large-scale Wireless LANs (WLAN) have been deployed all over the world. However, the shortage of non-interfering channels makes it a challenge for WLANs to efficiently support real-time multicast services. In this paper, we study the problem of efficient scheduling of *real-time multicast flows*. For mitigating interferences, we allow access-points (APs) to transmit simultaneously only if they are mutually non-interfering and our objective is minimizing the fraction of time used by the APs for servicing the multicast flows.**

We introduce two multicast strategies, the *association strategy* for which each user is restricted to receive flows only from its associated AP and the *non-association* strategy for which a user may also decode transmissions from other APs in its vicinity. Under both strategies, the scheduling problem of minimizing the multicast service time is NP-hard and we propose simple approximation algorithms with provable performance bounds. Our simulations clearly demonstrate that the proposed algorithms yield efficient multicast scheduling.

1. Introduction

Large-scale Wireless Local Area Networks (WLANs) are commonly used all over the world to provide Internet connectivity to mobile users. However, they are still lacking efficient mechanisms for delivering real-time data to groups of users, referred to as *multicast services*, which are critical for broadcasting live video and audio streams. A naive implementation of multicast services by leveraging the MAC layer broadcast capability of IEEE 802.11 will lead to heavy collisions due to shortage of non-interfering frequencies. For instance, the dominant 802.11 b/g standards support only three non interfering frequencies. Thus, adjacent APs often use the same frequency and interfere each others transmissions, which significantly degrading the quality of multicast services. One solution to this problem is to use unicast transmissions in place of broadcast, to achieve high reliability. However, such a solution does not exploit the broadcast characteristics of the wireless networks and it generates significant unnecessary overhead. Since the multicast services will be provided in addition to the traditional unicast services, it is critical to reduce the multicast service overhead for increasing the network utilization, while preserving the Quality-of-Service (QoS) to the users. Therefore, *our objective is to design efficient multicast schemes that mitigate interference and reduce the transmission overhead by utilizing efficient packet scheduling.*

Multicast protocols have been well explored for different types of wireless networks. However, scheduling multicast flows in the context of infrastructure mode WLANs has received little prior attention. In [1], [2] the authors propose reliable multicast schemes for WLANs based on positive and negative acknowledgments (ACK and NAK). In [2], the authors propose a probabilistic technique to improve the performance of MAC layer multicast services. In [3] a reliable MAC layer multicast solution is proposed that makes use of overhearing and suppresses unnecessary retransmissions. In [4] the problem of association control for supporting multicast flows while optimizing various network objectives has been studied. These schemes are restricted to communication from a single AP, and they do not perform packet scheduling for eliminating interference across multiple APs, which may lead to low utilization. Unlike these studies, the focus of this paper is on deterministic scheduling algorithms for eliminating interference in WLANs with provable performance bounds. Our scheduling algorithms efficiently utilizes the variable transmission bit-rates to achieve high network utilization, while mitigating interferences between adjacent APs that use the same frequency.

This study addresses the problem of efficient packet scheduling for supporting multicast services in WLANs, under two transmission strategies, *association and non-association strategies*. For both strategies the multicast problem is NP-hard and hard to approximate in general. Therefore, our approximation results focus on common special cases in which WLANs are modeled as Unit Disk Graph (UDG) [11], as explained in Section 4. In such cases, we propose 3 approximation algorithms for the association strategy with approximation ratios of 12, 10 and 5. For the non-association strategy, we propose two approximation algorithms with approximation ratios of $5 \log n$, and $4 \log n$, where n is the number of users. We note that logarithmic approximation is the best possible due to the proved lower bounds. Finally, we evaluate the performance of these solutions using simulations and observe that the non-association algorithms outperform the association-based solutions. Due to space constraints, this *extended abstract* considers WLANs with a single multicast flow. An extension for the cases of multiple flows can be found in [5].

2. Network Model and System Description

A. The Network Model

We consider an IEEE 802.11 WLAN comprises of multiple *access points* (APs) and contains a *network operation center* (NOC) for coordinating the APs' operations. Each AP is associated with a single frequency that is used as shared medium for serving users residing in its transmission range and each user is associated with a single AP for sending and receiving unicast traffic. In this study, we divide the APs according to their associated frequencies and we consider all the *stations*, both APs and users, that share the same frequency as a distinct *single-frequency-WLAN*. We assume that appropriate frequency planning is done separately. We apply our multicast scheme for each single-frequency-WLAN independently. Therefore, in the rest of the paper we focus on a single-frequency-WLAN. We use $\mathcal{A}$ and $\mathcal{U}$ to denote the sets of APs and users of the considered single-frequency-WLAN, respectively. We assume a quasi-static mobility pattern, in

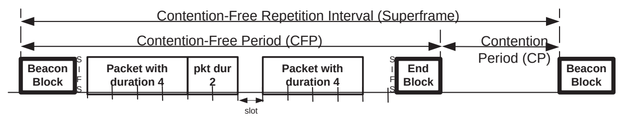

Fig. 1. *A superframe with* 12 *slots that are used for transmitting three packets with different durations.*

which users are free to move from place to place, but they tend to stay in the same physical locations for long time periods, as reported by [6].

In this study we assume only a single multicast flow that generates messages of the same size and at a fixed rate. Recall that for multicast service each message is then transmitted by some of the APs as separate packets. Since in WLANs the APs use various communication bit-rates based on the wireless channel conditions, the time required to transmit a packet may very between different AP-user pairs, depends on the used bit-rate. In our system, time is divided into *time slots* and each packet transmission time requires an integer number of time slots. Accordingly, for each AP $a \in \mathcal{A}$ and each user $u \in \mathcal{U}$ in its transmission range, we define by $d_{a,u}$ the *packet duration*, which is the minimum number of time slots required for transmitting a packet with an appropriate bit rate decodable by the user[1].

Since all the APs use the same frequency their transmissions may interfere with each other. Two APs are *interfering* if a packet sent by one AP may prevent a proper packet decoding in the vicinity of the other AP. These interference relationships are represented by an *interference graph* $G(\mathcal{A}, E)$, where $\mathcal{A}$ is the set of APs and an edge $(a, b) \in E$ represents every pair of interfering APs $a, b \in \mathcal{A}$. In addition, for each AP $a \in \mathcal{A}$ we denote by N_a its set of interfering APs and itself, *i.e.*, $N_a = \{a\} \cup \{a' | (a, a') \in E\}$.

B. An Overview of the Multicast System

Our solutions make use of the architecture discussed below which is based on the Hybrid Coordination Function (HCF) defined by the IEEE 802.11e standard [7]. A similar architecture was also used in [8] to provide efficient unicast services over WLANs. We divide time into repeated periods called *superframes*. Each superframe has a fixed duration and it contains a *Contention Free Period* (CFP) followed by a *Contention Period* (CP). The CFP is used for multicast transmissions without interference, while the CPs support unicast traffic. Each CFP starts with a *beacon block* (BB) in which all the APs transmit beacon messages for initiating a CFP in their vicinities 'almost' at the same time. It ends with an *end block* (EB), in which all the APs send CF-end messages for ending their CFPs. Synchronization of the CFPs eliminates interference that result from simultaneous transmissions of mobile users operating in DCF mode. At each CFP, time is divided into slots. The slots are efficiently allocated to the APs such that interfering-APs are not allowed to transmit simultaneously, which prevents interference from adjacent APs. Thus, in this scheme a single multicast message is usually transmitted several times by different APs and at different time slots. We refer to each such instance as a *packet*. A typical structure of a superframe is illustrated in Figure 1.

[1]Essentially, the channel condition between an AP and a user varies in time, however, we assume that the selected bit-rate enables user to decode the vast majority of the packets sent by its associated AP, given no interference from nearby stations.

3. The Multicast Problem

In this section, we present two possible packet-scheduling strategies for supporting multicast services and compare their performance. The first is a *association-based multicast strategy* that allows an AP a to send a single multicast packet for servicing all its associated users. In this case, we assume minimum packet duration D_a that can be decoded by all the associated users of a. $D_a = max_{u \in \mathcal{U}_a} d_{a,u}$, where $\mathcal{U}_a$ denotes the set of users associated with AP a.

The second is *non-association multicast strategy* that enables users to receive multicast packets from any AP, regardless of their associations. This strategy is based on the broadcast characteristic of the shared wireless medium and each user attempts to decode all the transmitted messages. The packet scheduling needs to ensure that each user is able to decode at least one transmitted packet correctly. In other words, there is at least one AP a that transmit packets with duration of at least $d_{a,u}$. *Recall that the IEEE 802.11 b/g standards support only three non-interfering frequencies, thus, adjacent APs may share the same frequency. The non-association strategy utilizes this shortage for its advantage to improve the system performance, as illustrated in Example 1.*

Example 1: Consider a WLAN with 3 APs denoted by a, b and c that may interfere with each others transmissions. Moreover, assume that the network contains 3 pairs of users such that each pair is associated with a different AP. In each pair, one can decode packets with duration of a single time slot, while the other requires packet duration of at least 2 slots. We also assume that each user can decode a packet sent by any AP with a packet duration of at least x, where x is either 4 or 10 in this example.

Case I - Association-based multicast strategy: Each AP sends a single packet to all its associated users. Thus each AP requires 2 slots and the overall required broadcast time is 6 slots.

Case II - Non-association multicast strategy: This strategy utilizes the user's flexibility to decode packets sent by any AP in its vicinity, rather than decoding messages sent only by its associated AP. Thus, when a single AP send a packet with duration x all the users can decode this packet. When $x = 4$ then this is the most efficient way to broadcast the message and when $x = 10$ the preferred scheduling algorithm is to allow each AP to broadcast the message to its associated users as described in Case I. □

We now provide the formal problem definition for both strategies assuming a single multicast flow with a single message per superframe. Let $\mathcal{A}$, $\mathcal{U}$, $G(A, E)$ denote the APs, users and the interference graph.

Definition 1: *A packet schedule for the* **association multicast strategy** *is termed* **feasible** *if for every AP a and a user u associated with AP a, AP a transmits at each superframe a multicast packet during a period of at least $d_{a,u}$ consecutive slots. Moreover, during the transmission of AP a the neighbors of a are silenced.* □

Definition 2: *A packet schedule for the* **non-association multicast strategy** *is termed* **feasible** *if at each superframe*

1	2	1	2
3	4	3	4
1	2	1	2
3	4	3	4

Fig. 2. *Labels of unit squares. APs from the same square or neighboring squares may interfere with one another. However, APs from different squares* with the same label *cannot interfere.*

and for every user u there is an AP a that transmits a multicast packet with a time period of at least $d_{a,u}$ consecutive slots. Moreover, during the transmission of AP a the neighbors of a are silenced. □

Definition 3: *The* **multicast problem** *seeks for a feasible packet schedule (given by Definitions 1 or 2) that minimizes the required time for servicing the multicast flow.* □

4. Algorithms

Before we present approximation algorithms we first state the hardness of the problem. Under the association strategy, the problem can be viewed as an extension of *graph coloring*; under the nonassociation strategy, it can be viewed as an extension of *set cover*. Both problems are NP-hard [9], and furthermore are hard to approximate [10], [5].

Theorem 1: Both the association and non-association problems are NP-hard, even when every packet duration is one. The association problem cannot be approximated within a factor of $n^{1-\epsilon}$ for any $\epsilon > 0$, unless $Z_{PP} = NP$; the nonassociation problem cannot be approximated within a factor of $O(\ln\ n)$, where n is the number of users.

The above theorem indicates that it is hard to find a provable approximate solution for the problem for general interference graphs. Therefore, we focus our attention to the natural special case of circular region wireless channel, where the induced interference graph is a unit disk graph (UDG) [11]. In such graphs two APs are considered interfering if their distance is bounded by a given value. UDGs graphs have the following properties that make good approximations possible.

Property 1: Consider a UDG $G(V, E)$ and let $G_a(N_a, E_a)$ be the graph induced by the neighbors N_a of any node $a \in V$. Then, the maximum independent set of the graph G_a has at most 5 nodes ([11], [8]).

Property 2: Suppose the APs form a UDG. The region of the APs is partitioned into unit squares and label each square 1, 2, 3 or 4 as shown in Figure 2. The dimension of the squares coincides with the interference range of the unit-disk interference graph. We have (a) At most three APs in each square can be mutually independent; (b) Two APs in two different squares *with the same label* do not interfere.

A. The Association Strategy

For the association strategy, we present three algorithms with approximation ratios 12, 10 and 5 respectively.

Algorithm based on Tiling Squares: We partition the region of the APs into unit squares and label them as described in Property 2. We first consider squares labeled with 1 only. Within each square s of label 1, we assign time slots sequentially to the APs in s. More specifically, we order the APs in s in an arbitrary order, $a_1, a_2, \ldots$, and assign to a_1 time slots

	a	b	c	d
Arbitrary	1 (I)	[6,9] (IV)	5 (III)	[1,4] (II)
LDF	5 (III)	[1,4] (I)	5 (IV)	[1,4] (II)
SCF	1 (I)	[2,5] (III)	6 (IV)	[1,4] (II)

Fig. 3. *Possible coloring under arbitrary node ordering, LDF and SCF. The Roman numerals indicate the order in which the nodes are colored.*

$[1, D_{a_1}]$, assign to a_2 slots $[D_{a_1}+1, D_{a_1}+D_{a_2}]$, etc. Suppose the largest time slot assigned to any square with label 1 is T. We repeat the above process for squares with label 2, starting with time slot $T+1$. We then continue with the squares labeled with 3 and finally those labeled with 4. Due to Property 2,

Theorem 2: The square tiling based algorithm is a 12 approximation for the association strategy.

LongestDurationFirst (LDF): For the association strategy, each user is associated with exactly one AP. For each AP a, recall $D_a = \max_u d_{a,u}$ is the maximum number of time slots required for users associated with a. Since each user can only be served by one AP, an AP a has to be activated for D_a consecutive slots at some point. Therefore, we view the association strategy as the *generalized graph coloring problem*. Given the interference graph, we assign each node a with D_a *consecutive* colors, so that any two neighboring nodes share no common colors. The objective is to minimize the total number of colors needed. The special case of $D_a = 1$ for all a is the classic graph coloring problem.

We present two algorithms, *LongestDurationFirst (LDF)* and *SmallestColorFirst (SCF)*, both of which generalize the well-known result for the classic graph coloring problem. Namely, a graph with maximum node degree Δ can be colored with $\Delta+1$ colors. To accomplish the $\Delta+1$ coloring, we choose an *arbitrary* uncolored node and assign to it the smallest color not yet assigned to any of its neighbors. As we shall see in the following example, the order in which the nodes are colored is key for the generalized chromatic number.

Example 2: Consider the following example that is also illustrated in Figure 3. The interference graph consists of 4 APs, a, b, c and d forming a line. They respectively require $D_a = 1$, $D_b = 4$, $D_c = 1$ and $D_d = 4$. If we color the APs in the order of a, d, c and b, then a is assigned color 1; d is assigned colors 1-4 since a and d are non interfering; and c is assigned color 5 since c and d are interfering. In order for b to get 4 consecutive colors, b would have to be assigned colors 6-9. Therefore, the sequence of colors 2-4 cannot be used by b or any of b's neighbors. Such "wastage" is what we aim to avoid in the two algorithms we propose.

For LDF, we first assume that every D_a is a power of 2. The generalization to all values of D_a is trivial. Let $\mathcal{A}_j$ be the set of APs a such that $D_a = 2^j$, where the max value of j is denoted by J. We start with $\mathcal{A}_J$. For every $a \in \mathcal{A}_J$, we assign to a the smallest 2^J colors that are not already assigned to a's neighbors. We state without proving that these 2^J colors are indeed consecutive. Once every node in $\mathcal{A}_J$ is colored we continue to color the nodes in $\mathcal{A}_{J-1}$, $\mathcal{A}_{J-2}$, etc, in a similar manner. See Figure 4 for the pseudo code. Let us revisit the example in Figure 3. Under LDF, a possible ordering is b, d, a and c. Both b and d are assigned colors 1-4 since they are non interfering; both a and c are assigned color 5.

Let $\Delta_a = \sum_{a' \in N_a} D_{a'}$ be the total number of slots needed

LDF (G)
1 for $j = J, J-1, \dots 0$
2 for $a \in \mathcal{A}_j$
3 assign a smallest 2^j colors not already assigned to a's neighbors

Fig. 4. *LDF for generalized graph coloring on G, assuming every D_a is a power of 2.*

for the neighbors of an AP a. We can view Δ_a as the "generalized" node degree of a and state the following result for the generalized graph coloring.

Theorem 3: If every D_a is a power of 2, then LDF uses at most $\Delta_{a^*} + D_{a^*}$ colors for the generalized graph coloring problem, where $a^* = argmax_a(\Delta_a + D_a)$.

Property 1 implies a lower bound of $\Delta_{a*}/5 + D_{a^*}$. Theorem 3 shows LDF finds a solution that is at most 5 times the lower bound, and therefore implies a 5-approximation if every D_a is a power of 2. For general values of D_a we lose another factor of 2 by rounding D_a up to the next power of 2.

Theorem 4: For a unit-disk interference graph, LDF guarantees a 10 approximation for the association strategy.

SmallestColorFirst (SCF): We now use a more clever ordering to color the nodes in G that would guarantee a 5 approximation without requiring the D_a's to be powers of 2. To decide which node to color, we find for each node a the smallest color not yet assigned to any of a's neighbors. Let this color be h_a. Initially, every node is uncolored and has $h_a = 1$. We then color an uncolored node b that minimizes the value of h_a, *i.e.*, $b = argmin_a h_a$ where the min is taken over all uncolored nodes. We assign to b the smallest D_b colors that are not already assigned to b's neighbors. We state without proving that these D_b colors are indeed consecutive. We now update the value of h_a for all $a \in N_b$. In essence, we color the node that uses the smallest available color. We refer to this algorithm as SmallestColorFirst (SCF). See Figure 5 for the pseudo-code. Let us revisit the example shown in Figure 3. Under SCF, a possible solution first assigns a color 1 and assigns d colors 1-4, both using the smallest color 1. Now the smallest color available to b is 2 and the smallest color for c is 5. Therefore, SCF assigns b colors 2-5 and finally assigns c colors 6.

SCF (G)
1 $h_a = 1$ for all a
2 while there is an uncolored node
3 let $b := argmin_{\text{uncolored } a} h_a$
4 assign b smallest D_b colors not already assigned to b's neighbors
5 update h_a for $a \in N_b$

Fig. 5. *SCF for generalized graph coloring on G, for all values of D_a. (The $argmin$ notation in line 3 indicates $h_b = min_a h_a$.)*

We state the following result, parallel to Theorem 3.

Theorem 5: For any values of D_a, SCF uses at most $\Delta_{a^*} + D_{a^*}$ colors for the generalized graph coloring problem, where $a^* = argmax_a(\Delta_a + D_a)$.

Again, Property 1 and Theorem 5 guarantee that the gap between the upper and lower bounds is at most 5.

Theorem 6: For a unit-disk interference graph, SCF guarantees a 5 approximation for the association strategy.

B. The Non-association Strategy

In addition to the graph coloring aspect, the non-association strategy has the aspect of the set cover problem. Given a collection of sets of elements, the set cover problem aims to find the minimum number of sets whose union contains all the elements. A simple logarithmic approximation of set cover repeatedly chooses the set that covers the most number of uncovered elements. For the non-association strategy we can view each AP as a set of users and we aim to cover all users as quickly as possible. We present two algorithms with logarithmic approximation ratios, both motivated by the greedy approach of set cover. Due to the logarithmic lower bound on approximating set cover presented in Theorem 1 we note that our logarithmic approximations cannot be improved by much.

Two Logarithmic Approximation Algorithms: The two logarithmic approximation algorithms share the following common framework. The algorithms work in rounds, and aim to cover as many uncovered users as possible per time slot during each round. In particular, for round i each algorithm tries all possible duration $d \in \mathcal{D}$ and picks an independent set S of APs for each d. For given d and S the set of users that can be served in this round are those u whose required packet duration $d_{u,a}$ is at most d for some $a \in S$. Each algorithm has a different method of picking an independent set. However, each algorithm always chooses the duration d^i and independent set S^i such that minimizes the ratio of d to the number of users served during round i, which is equivalent to maximizing the number of new users served per time slot.

Round i
1 for $d \in \mathcal{D}$
2 find an independent AP set S
3 let $X_{a,d} := \{\text{unserved } u \mid d_{a,u} \leq d\}$ for $a \in S$
4 let $Y_{S,d} := \cup_{a \in S} X_{a,d}$
5 let $(S^i, d^i) := argmin_{S,d} \frac{d}{|Y_{S,d}|}$
6 serve every user in Y_{S^i,d^i}

Fig. 6. *Round i.*

Before we describe how each algorithm chooses an independent set S given d for each round, we state the following property of the framework. Given d, let $\hat{S}$ be the independent set that maximizes the number of unserved users, *i.e.*, $\hat{S} = argmax_S |Y_{S,d}|$. We say an algorithm is a ρ-approximation for maximum independent set, if it finds an independent set S such that $|Y_{S,d}| \geq |Y_{\hat{S},d}|/\rho$.

Lemma 1: If an algorithm guarantees a ρ approximation for max independent set, then it also guarantees a ρH_n approximation for the non-association strategy, where $H_n = \Theta(\log n)$ is the nth harmonic number.

We now describe two approximation algorithms for finding the max independent in line 2 of Figure 6. We make use of the fact that the interference graph is a unit disk graph. The first algorithm greedily finds an AP independent from those already chosen such that the maximum number of unserved users can be covered. We call this algorithm *GreedyIndependentSet (GreedyIS)*. By Property 1, we have

Lemma 2: For a unit-disk interference graph, GreedyIS has approximation ratio $\rho = 5$ for max independent set.

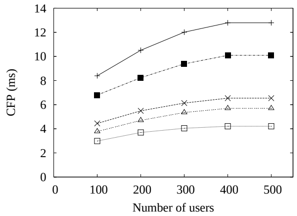

Fig. 7. Multicast AP=50.

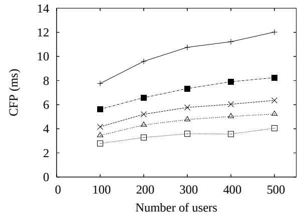

Fig. 8. Multicast AP=75.

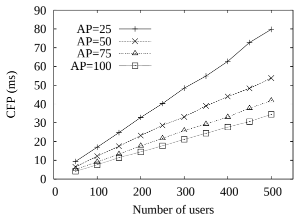

Fig. 9. Unicast.

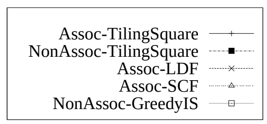

Fig. 10. Legend for Fig 7 and 8.

Combining Lemmas 1 and 2 we have,

Theorem 7: For a unit-disk interference graph, GreedyIS has approximation ratio $5H_n = O(\log n)$ for the non-association strategy.

We refer to the second algorithm as the *TilingSquareIS*. Like for the association strategy, this algorithm partitions the region of the APs into unit squares and labels each square 1, 2, 3 or 4 as shown in Figure 2. Due to Property 2a), the algorithm enumerates all possible subsets of APs of size 1, 2 and 3 within each square and computes the number of users that can be covered by each subset. The algorithm then chooses, for each square, the subset of APs that cover the most number of users. As stated in Property 2b), any two APs in two different squares with the same label do not interference each other. Therefore, the chosen subsets in all squares labeled 1 form an independent set. The same statement holds for labels 2, 3 and 4. The algorithm chooses the best label. Therefore,

Lemma 3: For a unit-disk interference graph, the square tiling based algorithm guarantees an approximation ratio $\rho = 4$ for max independent set.

Combining Lemmas 3 and 2 we have,

Theorem 8: For a unit-disk interference graph, the square tiling based algorithm guarantees an approximation ratio $4H_n = O(\log n)$ for the non-association strategy.

5. Simulations

Simulation settings: We distribute APs and users randomly in a 5.0×5.0 region. Two APs within a distance of unity are assumed to interfere and are not allowed to transmit simultaneously. We assume data rates are supported by 802.11g, and the possible communication bit rate between two stations depends on their distance (based on data in [12]). We assume that the network supports a single multicast flow that generates one message of 1500 bytes per superframe. We evaluate the performance by considering the *total CFP in mill-seconds for sending the multicast flow to all users for various algorithms under different network configurations*. We also compare the performance of our algorithm with unicast strategy in which a dedicated packet is sent to each user. For this strategy, we use a standard graph-coloring method for minimizing CFP duration. Results of our simulations are presented in Figures 7-9, where each point in these charts is an average of 200 runs.

Simulation Results: We observe that both the association and non-association strategies significantly outperform the unicast strategy. Moreover, while the CFP duration of the unicast strategy increases linearly with the number of users, the CFP duration of the proposed strategies expands sub-linearly and converges to a constant CFP duration. This justifies the use of broadcast mechanisms for supporting multicast services, mainly when supporting large number of users. Figures 7-8 show that overall the non-assoc. strategies perform better than the assoc. ones. In particular, the non- association GreedyIS algorithm outperforms all other algorithms, which justifies the usage of the non-association strategy. For the association strategies, we observe that the SCF performs better than LDF. This results from the additional overhead introduced in LDF by rounding required slots up to the next power of 2. Our simulations also suggest that the tiling square algorithms have the worst performance for both strategies. This is not surprising, since they aggressively disallow two APs from neighboring squares to be active simultaneously, even if they are not interfering.

6. Conclusion

This paper proposes association and non-association packet scheduling strategies for supporting real-time multicast services in WLANs. The strategies efficiently utilize the network resources, while mitigating interference across neighboring cells. For these strategies the paper presents several algorithms with provable performance guarantees. Then, through simulations, it demonstrates that the proposed algorithms significantly improve the capabilities of todays WLANs in supporting multicast services.

Acknowledgment

This work has been partially supported by the National Science Foundation through awards CNS-0546630 (CAREER Award) and CNS-0403342.

References

[1] J. Kuri and S. K. Kasera, "Reliable Multicast in Multi-Access Wireless LANs," in *IEEE INFOCOM* 1999.
[2] P. Chaporkar, A. Bhat, and S. Sarkar, "An Adaptive Strategy for Maximizing Throughput in MAC Layer Wireless Multicast," in *MobiHoc'04*.
[3] K. Tang and M. Gerla, "MAC Reliable Broadcast in Ad Hoc Networks," in *IEEE MILCOM* 2001.
[4] A. Chen, D. Lee, and P. Sinha, "Optimizing Multicast Performance in LargeScale WLANs," in *IEEE ICDCS* 2007.
[5] Y. Bejerano, D. Lee, P. Sinha, and L. Zhang, "Efficient Multicast Schemes for Wireless LANs," Bell-Labs, Research Report, Mar. 2006.
[6] D. Kotz and K. Essien, "Analysis of a Campus-wide Wireless Network," in *ACM MobiCom*, 2002.
[7] Bob O'Hara and Al Petrick, *The IEEE 802.11 Handbook: A Designer's Companion, Second Edition*, IEEE press, 2005.
[8] Y. Bejerano and R. S. Bhatia, "MiFi: A Framework for Fairness and QoS Assurance in Current IEEE 802.11 Networks with Multiple Access Points," in *IEEE INFOCOM* 2004.
[9] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W.H. Freeman Publishing Co., 1979.
[10] U. Feige, "A Threshold of $\ln n$ for Approximating Set-cover," in *ACM STOC* 1996.
[11] M. V. Marathe, H. Breu, H. B. Hunt III, S. S. Ravi, and D. J. Rosenkrantz, "Simple Heuristics for Unit Disk Graphs," *Networks*, vol. 25, pp. 59–68, 1995.
[12] M. H. Manshaei and T. Turletti, "Simulation-Based performance analysis of 802.11a Wireless LAN," in *IST* 2003.