Association Management for Data Dissemination over Wireless Mesh Networks

Dongwook Lee† Gayathri Chandrasekaran[‡],
Mukundan Sridharan[†], and Prasun Sinha[†]

[†]*Department of Computer Science and Engineering, Ohio State University*
[‡]*Department of Computer Science, Rutgers University*

Abstract

To enable multimedia broadcasting services in mesh networks, it is critical to optimize the broadcast traffic load. Traditionally, users associate with access points (APs) with the strongest signal strength. We explore the concept of dual-association, where the AP for unicast traffic and the AP for broadcast traffic are independently chosen by exploiting overlapping coverages that are typical in mesh networks. The goal of our proposed solution is to optimize the overall network load by exploiting the flexibility provided by independent selection of unicast and broadcast APs. We propose a novel cost metric based on ETT (Expected Transmission Time) and the number of nodes in range of the APs, that are advertised in the beacons from the APs. Users periodically scan and associate with the AP which has the lowest cost metric. The proposed approach reduces the number of APs that handle the broadcast traffic resulting in a heavy reduction in control and data packet overhead. This leads to higher packet delivery rate and enhanced video quality measured in terms of PSNR. Our approach allows the freed up resources at APs to increase the unicast throughput. We compare the performance of our approach with traditional signal strength based association using extensive simulations and real experiments on an indoor testbed of 180 IEEE 802.11b based devices.

Key words: Association Control, Broadcast, Wireless LAN, Dual Association, and, AP switching

---

Email addresses: `leedon@cse.ohio-state.edu` (Dongwook Lee†),
`chandrga@cs.rutgers.edu` (Gayathri Chandrasekaran[‡]),
`mukundan@cse.ohio-state.edu` (Mukundan Sridharan[†]),
`prasun@cse.ohio-state.edu` (Prasun Sinha[†]).

Preprint submitted to Computer Networks *4 September 2007*

1 Introduction

Mesh Networking is emerging as a promising technology that brings Wireless LANs to the masses at a reduced deployment cost. Mesh networks are either flat or hierarchical in terms of the architecture [1,2]. Our study is focused on hierarchical architectures, where the client-client communication is always via the APs (access points) and the APs are connected through a multi-hop wireless backbone. Public deployments of mesh networks are already operational in several cities including Philadelphia, Las Vegas, and Urbana-Champaign. Various types of WLANs, such as city-wide WLANs[1], in-building WLANs, and temporary WLANs, can all benefit from the Mesh-network technology.

While unicast services are essential for providing Internet access to individual users, emerging broadcast services are needed to deliver local news, visitor's information, TV channels, or other multimedia content. In order to support efficient multimedia services with minimal impact to unicast services, it is critical to optimize the multicast load on the network.

In this paper we study the problem of optimizing the broadcast traffic load in the mesh using the concept of dual-association, where users maintain distinct associations for unicast and broadcast traffic. We use the AP with the best signal strength for unicast traffic, but other metrics such as unicast traffic load [3] can also be used to select the unicast AP. For reducing the broadcast traffic load, users select the AP for broadcast services independent of the AP selected for unicast traffic. The selected broadcast APs can be connected to the AP with the backbone access (Main AP or MAP) using any ad-hoc multicast routing protocol. As the multicast structure construction is not the focus of the paper, we choose to connect the selected APs to the MAP using a tree, only for purposes of simplicity. The tree construction and maintenance mechanisms are based on MAODV [4], but any other multicast routing protocol can be used as well. The concept of dual-association was first introduced in our prior work [5] where a simple multicast metric was also proposed. In contrast, this paper proposes a cost metric that captures the global cost on the network (in addition to the cost on the last hop between the user and the AP), and an efficient local synchronization algorithm to make it practicable to implement association based on the cost-metric. Moreover, in this paper we present data from real experiments and extensive simulations.

Our contributions in this paper are as follows. 1) We formalize the problem of efficient association for data dissemination over Mesh-networks. 2)We prove that the problem is NP-hard by showing a reduction from the Steiner tree problem. 3) We propose the dual-association concept and a local synchro-

---

[1] The city of Chaska, Minnesota provides WLAN coverage in a 15 sq miles area since Oct 2004 (www.chaska.net).

nization method for ease of deployment. 4) We introduce a novel metric that optimizes the broadcast traffic load in the mesh and present a heuristic based distributed protocol, called COST, based on our metric. 5) Using simulations in *ns2* we evaluate the performance of our approach and compare it with the traditional approach that uses signal strength based association. The key metrics studied are the size of the tree, and the quality of received MPEG video measured using PSNR (Peak Signal-to-Noise Ratio). 6) We have implemented the distributed approach and compared its performance with the traditional signal strength based approach, on an indoor testbed of 180 nodes with 802.11b radio.

The rest of the paper is organized as follows. Section 2 summarizes related work. Section 3 defines the problem, the notations, and the terminology used in the paper. Dual AP management framework is described in Section 4. The distributed solution is presented in Section 5. Section 6 presents a detailed evaluation of our approach and comparison with signal strength based approach using simulations. The results from the testbed experiments are presented in Section 7. Finally, Section 8 concludes the paper.

2 Related Work

In this section, we outline related work in the areas of mesh networking, controlled association in 802.11 networks, and sub-structure computation in adhoc and mesh networks.

Mesh Networking: Providing connectivity to large communities using wireless back-haul networks, also known as mesh networks, has lately received an increased attention [6,7]. Several companies including Mesh-networks, Firetide, Strix, and BelAir Networks have various commercial products and large-scale public deployments based on the concept of mesh networks.

Controlled Association: The concept of dual-association is first introduced in [5], where a metric based broadcast AP association algorithm is proposed. The algorithm requires global synchronization between APs and users, and the cost metric only concerns with the last hop between the user and the APs. In contrast, this paper proposes a cost metric that captures the global cost in addition to the cost on the last hop between the user and the AP, and an efficient local synchronization algorithm to make it practicable to implement association based on the cost-metric. This paper also experimentally evaluates the performance on a large scale testbed. In [8], the authors present a software based solution called Multinet, that facilitates simultaneous connections to multiple networks by virtualizing a single wireless card. In conjunction with the idea provided in Multinet, our solution consisting of multiple wireless cards can be modified to a solution that operated over a single wireless card. In 802.11 networks user nodes often use the signal strength as the key metric in selecting the AP. Recent work [3] has explored the idea of association control to balance the network load and provide max-min fairness among users. The authors prove that balancing the network load is equivalent to achieving max-min fairness. Although our objective is different from [3], in the presence of unicast flows load-balancing and fairness will make the problem more challenging.

Sub-structure Computation: The idea of constructing backbones or substructures in ad-hoc networks to limit the number of transmissions has been explored by several protocols. The concept of *Minimum Connected Dominating Set (MCDS)* has been used in designing various routing protocols for ad-hoc networks [9,10]. The importance of constructing and maintaining an MCDS in an ad-hoc network has spurred research on finding better approximation algorithms [11,12]. In case of multi-rate channels, the advantage of using a tree like structure for broadcasting multi-media content has been questioned in [13]. Authors show that multiple unicast transmissions, possibly at data rates higher than the base rate, can lead to better latency for broadcasting multimedia content in mesh networks.

3 Terminology and Problem Definition

We represent the connectivity between the users and the access points using a graph $G = (V, E)$, where V (same as $V(G)$) is the set of nodes (users and access points) and E (same as $E(G)$) is the set of edges. E consists of edges connecting users to access points in range, and between access points that are in range of each other. E does not include user-to-user edges as we do not consider ad-hoc communication between the users. V can be partitioned into the set of users, V_u, and the set of access points, V_a. We assume that one of the APs, called the main AP (MAP), has a connection to the backbone Internet and it acts as a gateway to the rest of the APs. An AP in the broadcast tree is referred to as TAP (Tree AP). The APs selected for broadcasting to users are referred to as SAP (Selected AP). We summarize the terminology used in this paper in Table 1.

As mentioned in Section 1, our goal is to optimize the broadcast traffic load in the network. One of the well known metrics for load is defined as the Expected Transmission Time (ETT) [14]. We define the problem of constructing the tree for optimizing the broadcast load, and name it the Mesh Steiner Tree (MESH-ST) problem. The following definition assumes that MAC layer unicast is used for transmissions between the APs, and MAC broadcast is used for the last hop transmission from the AP to the users. Unicast transmissions in

Notation	**Meaning**
AP	Access Point
MAP	Main Access Point
TAP	AP on the broadcast tree
SAP	A TAP selected for broadcasting (i.e., AP with at least one associated broadcast user)
GAP	A Gateway AP (TAP that is not a SAP). So, |GAP| = |TAP| - |SAP|
G	Graph of user nodes and APs
V or $V(G)$	Set of nodes (users and APs)
E or $E(G)$	Set of edges
V_a	Set of APs
V_u	Set of users

Table 1
Relevant Terminology and Keywords

the wireless backbone ensures high reliability and broadcast transmissions on the last hop are used for low overhead for serving multiple users. Other combinations of unicast and broadcast transmissions on the tree links can be used (with appropriate modifications to the definition of MESH-ST), but are not studied in this paper. The definition of the Steiner tree and our definition of the MESH-ST problems are as follows:

- **Steiner Tree (ST):** Given an undirected graph G with nonnegative edge costs and whose vertices are partitioned into two sets, Required nodes and Steiner nodes, ST is a minimum cost tree in G that contains all the required nodes.
- **Mesh Steiner Tree (MESH-ST):** Given a graph G with two vertex partitions V_a and $V_u = V(G) - V_a$, and a node $MAP \in V_a$, MESH-ST is a least cost tree with the MAP as the root and the nodes in V_u as its leaves. The cost $c(v)$ of each node v is the broadcasting cost (backoff time + transmission time) from the node v to all its associated users and the cost $c(e)$ of each internal AP-AP edge e is the ETT for unicast over that edge. The cost of the tree T is defined as follows.

$$Cost_{MESH-ST}(T) = \sum_{e \in \text{AP-AP links}} c(e) + \sum_{v \in \text{broadcasting APs}} c(v) \qquad (1)$$

The first term accounts for the unicast transmissions on the AP-AP links and the second term accounts for the broadcast transmission on the last hop from the AP to the user(s).

There are two main differences between ST and MESH-ST. The Mesh Steiner Tree requires the nodes in V_u to be leaf nodes, and the cost function includes weights of vertices adjacent to the leaf-edges in place of the weights of the leaf-edges. In spite of these differences, in the following theorems we show that the ST problem is reducible to the MESH-ST problem and vice-versa, thus proving that it is also an NP-hard problem. See Appendix for proofs of the theorems.

Theorem 1: [Lower bound] MESH-ST is at least as difficult as ST.

Theorem 2: [Upper bound] MESH-ST is at most as difficult as ST.

Theorem 1 establishes the NP-hardness of the MESH-ST problem. Theorem 2 shows that the problem can be modeled as a ST problem.

The key challenge is in designing a protocol that can efficiently compute the broadcast-AP for each user such that it can lead to a tree with the optimal broadcast load. The selected APs (rather than the end-users) can then be connected using any ad-hoc multicast protocol. Ad-hoc multicast protocols have been well studied in the literature [15–20]. Our focus in this paper is instead on the algorithm for selecting the broadcast-APs for association. Note that we can transform the problem with multiple MAPs to a problem with single MAP by *fusing* the nodes corresponding to the MAPs. Although we study the performance of multiple MAPs in our simulations, the discussion in the rest of the paper assumes a single MAP for simplicity.

4 AP Management Framework

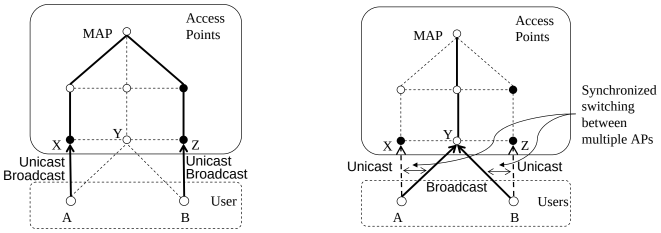
We propose an AP management framework for simultaneous support of higher quality broadcast and unicast services. The framework is characterized by Dual-AP association, Dual-traffic cycles, and Synchronized AP switching.

- **Dual-AP association:** Users requesting broadcast services maintain two independent associations with APs: one for unicast (unicast-AP) and the other for broadcast (broadcast-AP). In this paper the strongest signal strength is used for electing the unicast-AP. Other techniques such as balancing unicast load [3] can also be used. As optimizing unicast load is orthogonal to our work, we do not study it further in this paper. If the unicast-AP is already serving other broadcast users, then the user selects it as a broadcast-AP as well. If not, the user chooses a different AP for broadcast services based on the metric proposed in section 5. Note that this mechanism is not in contrast to the overall goal of optimizing the broadcast traffic load, as selecting the unicast-AP does not lead to increase in the broadcast load.

- **Dual-traffic cycles:** Time is divided into cycles consisting of separate unicast and broadcast periods.
- **Synchronized AP switching:** Users receiving broadcast services from an AP different from its unicast-AP switch between the unicast-AP and the broadcast-AP at the end of the respective periods.

4.1 Dual-AP Association

As discussed in Section 1 highly overlapping coverage areas are common in mesh networks. The separation of the unicast-AP and the broadcast-AP in our framework makes it possible to leverage the wider choice of broadcast-APs to optimize the broadcast traffic load. Consider the example shown in Figure 1(a), where user A selects AP X and user B selects AP Z based on signal-strength. Both unicast and broadcast data are received from the AP with which a user associates. As shown in Figure 1(b), using dual-association, the selection of unicast APs remains unchanged, but the broadcast traffic is now received by both the users through AP Y. This results in a reduction of the number of transmissions from 6 to 4 (highlighted links in Figure 1).



(a) Traditional Association (b) Multi-association

Fig. 1. Unicast and Broadcast Communication. Highlighted links carry broadcast traffic. Arrows on links between users and APs represents associations.

The IEEE 802.11 family of protocols supports multiple transmission rates with various modulation schemes. In accordance with the current IEEE 802.11 standard, we assume that broadcast packets are transmitted at the basic rate. However, the proposed solution can be easily extended to leverage future MAC layer solutions (as discussed in [13]) that can provide the flexibility of transmitting broadcast packets at data rates higher than the basic-rate.

4.2 Dual-traffic Cycles

Users with different unicast-AP and broadcast-AP switch in time between them. Time is divided into cycles where a cycle length is defined as, $T_c = t_b + t_u$, where t_b and t_u are the broadcast and unicast periods respectively. We denote the ratio of t_b to T_c by α. Therefore, $t_b = \alpha T_c$ and $t_u = (1-\alpha)T_c$. T_c and α can be configured by the network provider and can be advertised in the beacons. The effect of these parameters is studied using simulations in Section 6.1.

The broadcast and unicast packets are maintained in separate queues at each AP. At the beginning of the broadcast period, users switch to their respective broadcast APs (possibly requires changing the channel) and the APs start to transmit packets from their broadcast queues. If the broadcast queue in an AP has no packets during a broadcast period, it can send unicast packets to other users (users that have selected this AP for both unicast and broadcast). After the broadcast period, the users switch back to the unicast-AP. In the unicast period, packets are transmitted only from the unicast queue at the APs. In order to prevent under-utilization when APs do not have any downstream data (broadcast or unicast) during the broadcast period, upstream packets from the users are allowed to be transmitted at all times. However, in the broadcast period, downstream broadcast packets from the APs have higher priority. The broadcast packets use a high priority traffic class and the upstream unicast packets use a low priority traffic class. If an AP or a user is still in the process of transmitting a unicast packet at the beginning of the broadcast session, the transmission is completed before switching sessions.

4.3 Distributed Local Session Coordination

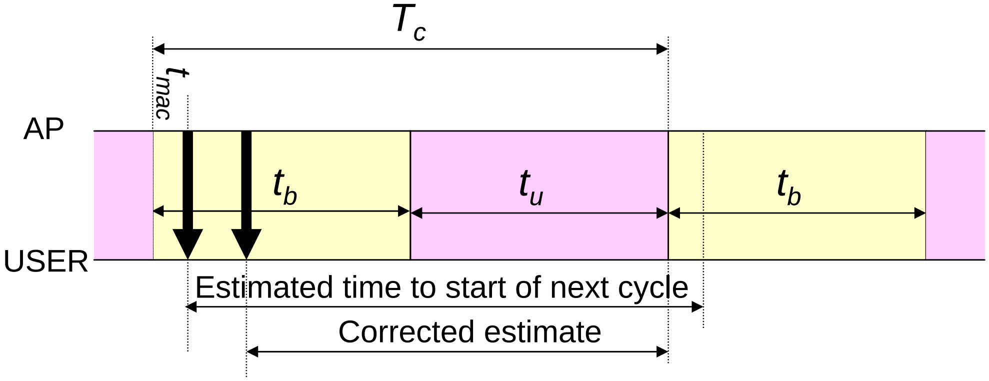


Fig. 2. Session Coordination

Each AP must know the switching schedule of all users associated with it, in order to forward the unicast packets at the appropriate time. One solution to this problem is to use global synchronization, which however requires significant overhead. We propose an alternate localized mechanism where a user follows the schedule dictated by its broadcast-AP and informs it to its unicast-AP. Note that we have assumed that if the unicast-AP is involved in broadcasting then the user selects it for both types of traffic. Thus for switching users, their unicast-APs will not be involved in broadcasting.

A naive approach is to let the broadcast-AP transmit a packet to synchronize all users. The users can then pass the session switching schedule to their unicast-APs. However, the latency at the MAC layer experienced by these packets may lead to inefficient coordination and packet losses at the switching boundaries between unicast and broadcast. To address this problem, we propose to send a pair of packets[2] where the second packet in the pair carries information on the MAC latency experienced by the first packet. This information is used to make corrections on the session schedule. The first packet contains T_c, the length of the cycle, and α, the fractional time for broadcast. The receiving node determines t_p, the processing delay and t_x, the transmission time. So, $T_c - t_p - t_x$ represents the remaining time to the beginning of the next cycle. The second packet contains t_{mac}, the MAC latency experienced by the first packet. So, the remaining time to the next cycle is corrected by a negative offset of $-t_{mac}$, when it is learned from the second packet.

The packet-pair approach can be used for coordination between the broadcast-AP and the user, and also between the user and its unicast-AP. For the latter case, randomized delays are used before triggering the packet-pair mechanism to avoid a burst of packet-pairs from multiple users. If there are multiple users switching between the same set of APs, then overhearing can be leveraged to suppress redundant packet-pairs. The packet-pair mechanism can be used periodically or on demand. A simple way to trigger it on demand is as follows. When a user observes an increased loss of broadcast packets, it sends an offset refresh request packet to the broadcast-AP to re-initiate the packet-pair mechanism. More specifically, a steady increase in the loss rate, along with a high value for the current loss rate is used as a trigger. This requires the user nodes to maintain a short history of loss rates observed in the recent past.

We experimentally show that our local session coordination scheme with an accurate Mac Latency estimation [21] is effective in reducing the loss during session switching (see Figure 3). In the baseline (global synchronization) both unicast and broadcast traffic is received from the same AP. The baseline numbers reflect the ideal performance that can be achieved with perfect global synchronization. For local coordination, one AP is used for downstream unicast and the other for downstream broadcast traffic. For this experiment both broadcast and unicast data are transmitted at 1 Mbps. Each cycle (200 ms) is divided into a broadcast period and a unicast period, where the ratio of the two periods are 1:3. The results in Figure 3 show that the throughput achieved with channel switching is around 90% of the baseline throughput. The loss in throughput is due to low kernel clock frequency (100 Hz), result-

---

[2] Note that unlike the packet-pair approach used in the Internet for estimating capacity, here the pair need not be sent back-to-back.

	Unicast	Broadcast	Total
Global Synchronization	273 kbps	90 kbps	363 kbps
Local Session Coordination	248 kbps	76 kbps	324 kbps

Fig. 3. Throughput with local session coordination compared with global synchronization.

ing in high synchronization granularity of the Stargate devices (see Section 7). In addition, channel switching takes around 15ms due to an inefficient implementation, resulting in reduction in throughput. With lower channel switching delay (can be reduced to $80\mu s$ [22]), and higher clock frequency (such as in a typical laptop), it is possible to further improve the performance.

5 Distributed Association Management

In this section, we present a distributed solution, called COST, for computing the broadcast association to optimize the broadcast load. We propose a metric that is advertised in the beacons from the APs. The AP with the lowest metric is selected as the broadcast-AP, in case the unicast-AP is not involved in broadcasting. If the unicast-AP is also involved in broadcasting, it is selected for both traffics. The following design guidelines form the basis of the solution:

- **Associate with an AP with the smallest ETT to the current tree:** The APs that have low ETT will reduce the load due to transmissions between the APs.
- **Associate with APs supporting more users:** The APs that have a large number of users in range have a higher potential for serving a large number of users. Selecting an AP that has the largest number of users is motivated by the greedy algorithm for minimum set cover algorithm [23]. The greedy algorithm chooses the set that covers the maximum number of leftover elements. So, in our solution APs with more users in range are preferred.
- **Associate to an AP that has a *special user* that is in range of a single AP:** If a user has only a single choice for association, then the corresponding AP is required to be part of the broadcast tree. If other users realize this necessity of the AP, they can be assured that by joining this AP the tree size will not be increased.

5.1 Metric Computation and AP Selection

Each AP needs to compute a metric to advertise. Let $CETT_i$ be the cumulative ETT to the nearest node in the broadcast tree from the AP_i, and N_i be the number of users in range of AP_i. The cost metric C_i of AP_i is defined as follows:

$$C_i = w_i(\beta CETT_i + (1-\beta)\frac{1}{N_i}),\ where \tag{2}$$

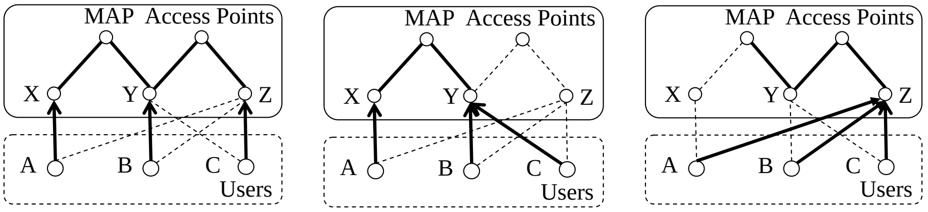
$$w_i = \begin{cases} \epsilon, \text{ AP } i \text{ has one or more special users} \\ 1, \text{ otherwise} \end{cases} \tag{3}$$

where ϵ is a very small number in the range (0,1) but close to 0, and β is a tunable parameter also in the range (0,1). Equation 2 gives a trade-off between the load of AP-AP channel and AP-user channel. These two terms correspond to the two terms of Equation 1. The first term of Equation 2 is the sum of transmission times along the hops to the tree. This reflects the resource consumption in the AP-backbone network. The second term is used to minimize the number of SAPs and thus the resource consumption in the AP-user network. w_i is set to a small value (ϵ) when APs have special users. This reduces the cost of such APs, thus making it highly likely to be selected by other users. The $CETT_i$ is computed using a proactive routing algorithm like DSDV that runs in the AP-backbone. We used the ETT calculation method introduced in [14]. N_i is computed by APs based on periodic scanning messages from users. Note that users periodically scan in all channels to select the best unicast-AP and best broadcast-AP for association. The cost metric and the associated broadcast-AP selection process has three properties: *Tree-preserving*, *Self-reordering* and *Self-convergence*.

- **Tree-preserving property:** The users periodically scan the channel and collect the cost metrics of neighboring APs. Then, they change their broadcast-AP to the minimum cost metric AP. However, once a user selects an AP, the $CETT_i$ of the selected AP becomes zero as the AP joins the broadcast tree. Thus, on the next scan, the user receives smaller cost metric from that AP, which forces the user to stick with the current AP. This tree-preserving property reduces the overhead of tree maintenance.
- **Self-reordering property:** When a user can hear two or more APs that are already part of the tree, their cost metrics are likely to be lower than that of non-tree APs because of zero $CETT_i$. Then, the user (re)selects an AP with the largest number of users among the in-tree APs. This *self-reordering* property attempts to optimize the number of selected APs in the tree, resulting in reduction of the total broadcast load in the network.

- **Self-convergence property:** The users that associate with the same AP in several scan iterations do not change their associations for longer intervals unless the link to the AP breaks. This *self-convergence* property results in stabilization of the tree and optimization of tree maintenance overhead.

Consider the example shown in Figure 4. Assume that the ETT for broadcasting from all APs are the same, and the ETT over each AP-AP link is also the same. When traditional signal strength based association is used, three APs are selected for broadcasting to the users and all five APs are selected in the broadcast tree. Figure 4(b) shows that if $c(e) >> c(v)$, then MESH-ST selects the least number of APs to be in the tree. As a result only three APs are selected in the tree, two of which have associated users. Figure 4(c) shows that if $c(e) << c(v)$, then MESH-ST selects the least number of APs with associated users. So it selects one AP with associated users to minimize the MESH-ST metric (Equation 1) which is joined to the MAP using a tree involving four APs. We observe that based on the computed ETT metrics, the tree may be different from the tree obtained by traditional association.



(a) Signal Strength (b) MESH-ST ($c(e) >> c(v)$) (c) MESH-ST ($c(e) << c(v)$)

Fig. 4. Structure of Broadcast Tree. The arrows represent the associations for broadcast data. The solid lines indicate the links on which data is transmitted.

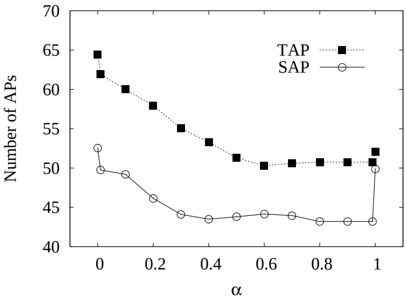


Fig. 5. Broadcast tree size with respect to β. The number of APs is 100 (10x10 grid) and distance of each AP is 90m with 100m radio range. The number of user is 50 with random way-point mobility model and $2m/s$ speed.

Figure 5 shows the average number of TAPs (number of APs in the broadcast tree) and SAPs (selected APs) with respect to β. For $\beta \geq 0.6$ TAPs and SAPs are both at their minimum levels, which is critical to optimize the $Cost_{MESH-ST}$. Moreover, we observe that in this range, the TAPs and

SAPs are insensitive to the exact value of β. Observe that as β approaches 1, the number of SAPs and TAPS remains similar, but for $\beta = 1$, the number of TAPs and especially SAPs is much higher than their lowest values. This shows that both the terms $CETT_i$ and $\frac{1}{N_i}$ are critical to the metric. For a very high value of β such as 0.99, the initial selection of the AP is based on $CETT_i$. After an AP is selected, the $CETT_i$ term becomes zero for that AP for subsequent scans, and the AP selection becomes dependent on the term $\frac{1}{N_i}$.

5.2 Limited Users per AP

Our discussion so far has assumed that an unlimited number of users can associate with an AP. But in reality, the number of users per AP is often bounded. A typical limit is 32 users for most 802.11 based APs. Our protocol can be easily extended to support such limits. A simple extension requires a flag in each beacon message. The flag is set only when the AP is saturated (associated with the maximum number of users allowed). If a user decides to associate with an AP which is already serving its maximum allowable number of users, the AP replies back to the user's association message indicating so, thus forcing the user to associate with another AP. Another way is to advertise progressively higher costs when APs start to get saturated. These approaches have an impact on the number of users that get starved (rejected by all neighboring APs as they are saturated).

5.3 Channel Scan

In this section, we discuss the tradeoffs of using different schemes for scanning channels. Channel scanning is a critical component as frequent scans will lead to better performance of the COST metric, at the cost of higher overhead. The Active Scan technique is widely used with 802.11 which does not require synchronization. Another technique for channel scanning called SyncScan [24] has been proposed to significantly reduce scanning overhead at the cost of global synchronization.

Active Scan: The proposed cost metric based distributed association algorithm needs reliable and up-to-date metric collection from all neighboring APs of a user. To obtain the cost metric, each user performs *active scan*. To refresh the cost of an AP, the AP listens to *probe* message sent by users. The AP sends back a *beacon* message that includes the cost of the AP. However, the overhead of the *active scan* is high, since every user independently searches channels. Moreover, *active scan* probes all channels one by one, which temporarily disconnects an existing session and results in bursty packet losses. The overhead of *active scan* consists of channel switching time (t_s), transmission time of a probe packet (t_x), and beacon waiting time (t_{min} for idle channel and t_{max} for busy channel). Each users performs the *active scan* periodically. Thus, overhead of *active scan* per channel, measured in terms of the fraction of time, L_a, is

$$L_{Active} = \frac{RN}{C}\sum_{c=1}^{C}((1-p_c)t_{min} + p_c t_{max} + t_s + t_x) \tag{4}$$

where p_c is the probability that there is at least one AP in channel c, N is the number of users, C is number of channels, and R is the frequency of scan.

SyncScan [24]: SyncScan [24] reduces the overhead by synchronizing short listening periods at the users with periodic *beacon* transmissions from each AP. Users change their channel during the short listening period to receive a *beacon* and return to the original channel. Users synchronize with APs on receiving the *beacon* message and then determine the next listening channel and time. Users periodically send *association update* messages to their associated AP. The *association update* messages carry the list of scanned APs. The APs send the received AP list to the neighboring APs through the backbone channel to facilitate the other APs to update their costs.

However, the following two issues need to be resolved: beacon collision between APs in the same channel and the need for AP synchronization. To avoid collision among multiple APs on the same channel, beacon generation time is randomly varied over a small window. To synchronize APs, NTP (Network Time Protocol) can be used. When NTP is used over ad-hoc networks, synchronization can be achieved with low mean error of $51\mu s$ in light traffic load and $1.5ms$ in heavy traffic load [25]. The overhead of SyncScan in terms of fraction of time per channel, L_s, consists of channel switching (t_s) and beacon waiting time(t_w):

$$L_{SyncScan} = \frac{R}{C}\sum_{c=1}^{C}(2t_s + t_w) = R(2t_s + t_w) \tag{5}$$

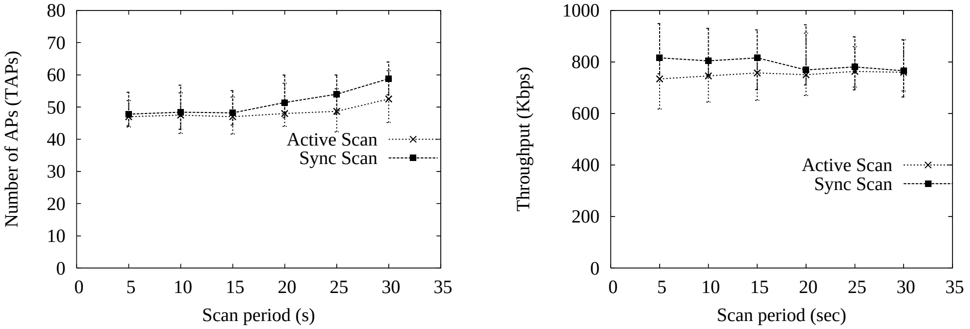
For example, in 802.11a, with 10 users and 13 channels to be scanned once in every 10 seconds, the active scanning overhead is over 13%[3]. However, SyncScan can reduce scanning overhead to 1.5% Figure 6(a) shows that the number of TAPs increases with the increase in the scanning period due to higher inaccuracy in the estimation of the cost. SyncScan uses stale information when

---

[3] According to [24], typically, t_s for Atheros-based NICs is $5ms$, t_{max} is $7ms$, and t_w is $5ms$. We assume that t_x is less than $1ms$ and every channel has at least one AP.

links to APs are broken to make association decisions as the users wait for the synchronized time for scanning channels. This results in larger broadcast tree for SyncScan. However, as the scanning overhead of SyncScan is lower than Active Scan, the throughput is still higher than Active Scan (Figure 6(b)). Note that for SyncScan the global synchronization has been assumed to be perfect without any additional overhead.

If the network requires synchronization for any of its other services, SyncScan is a preferred option. Otherwise, the overhead of the specific global synchronization mechanism, its accuracy and the expected improvement in throughput needs to be considered as well. So, *active scan* is a good choice due to simpler implementation and comparable performance. In our performance evaluation (both simulations and experiments) we have used *active scan*.



(a) TAPs

(b) Throughput

Fig. 6. A Comparison of Global Synchronization based SyncScan and Local Coordination based Active Scan (100 APs and 100 users with 10 m/s maximum speed)

6 Simulations

We present performance evaluation of the proposed cost metric based distributed algorithm (COST) using simulations on Network Simulator ns2 [26]. The default ns-2 channel model has been used in the simulations (two-ray ground propagation model). Packet losses may happen due to collisions and capture effects. We use a grid topology of 10 × 10 APs. The distance between neighboring APs is 200m and radio propagation range of AP is 250m, unless mentioned otherwise. One AP is designated as the main AP (MAP). Users are uniformly distributed in the area and move randomly according to the random way-point model. The maximum speed of a user is 10 m/s with 1 sec pause time. The users associate with APs using active scanning. We use the quarter common intermediate format (QCIF, 176×144 pixels/frame) sequence “Foreman” (first 300 frames from the original 30 fps sequence) encoded at 15 fps.

Simulation Parameters	**Value**
Radio Range	250m
Inter-AP distance on grid	200m
APs in the network	10x10
Video Sequence	“Foreman” sequence, 15 fps, QCIF, 176 × 144 pixels/frame
Video Data Rate	3 “Foreman” streams of 141 kbps = 423 kbps
Number of channels	13
Broadcast Data Rate	6 Mbps
Simulation Time per scenario	50 seconds
Number of Simulation Scenarios	15

Table 2
Default parameters used in the simulations

The encoder generates a stream with a bit rate of 141 kbps. Three such streams are simultaneously broadcast to each user subscribing to the broadcast data. So, video data rate is 141 kbps × 3 = 423 kbps. Each user has a single wireless interface and each AP has two wireless interfaces: backbone interface and local subnet interface. APs communicate with each other through the backbone interface. The backbone interface of all APs share a single channel. APs communicate with users via the local subnet interface. The APs operate in IEEE 802.11a infrastructure mode. The broadcast data rate is 6 Mbps. We assign 12 channels to the local subnet interfaces of APs in a way so as to minimize interference. Priority queuing is being used to give higher priority to the control packets as compared to the data packets. The two types of control packets used in the backbone are the ‘JOIN’ and ‘PRUNE’ packets for management of the broadcast tree. We implement a simple tree management mechanism using these messages. The error bars in the graphs represent the minimum and maximum values among a set of 15 independently and randomly chosen mobility scenarios.

The metrics of evaluation are as follows:

(1) **Queue Length:** Number of unicast and broadcast packets queued at a node with respect to time.
(2) **Throughput:** Broadcast and unicast throughput per user.
(3) **Average Delay:** Average delay experienced by a broadcast packet.
(4) **Average Loss:** Average rate of packet loss for broadcast packets.
(5) **Number of SAPs:** Number of APs with associated users.
(6) **Number of TAPs:** Total number of APs in the tree.
(7) **Number of GAPs:** Total number of Gateway APs. (|GAP|=|TAP|-|SAP|)

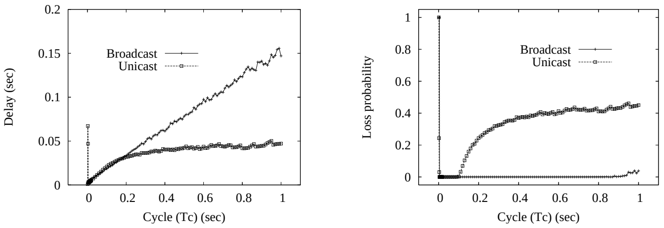
(8) **Fraction of Time per AP-interface for Broadcast Traffic:** Average fraction of time spent on transmission of broadcast packets by each AP-interface.
(9) **Control Packets in Backbone:** Number of broadcast tree management packets (‘JOIN’ and ‘PRUNE’) transmitted.
(10) **Normalized Broadcast Load:** The number of data packets transmitted on the tree normalized over the delivery ratio.
(11) **PSNR (Peak Signal to Noise Ratio):** It is an objective video quality metric computed based on the original video signal and the received video signal. Note that the terms ’signal’ and ’noise’ are not related to the physical channel, but instead the application layer data.

Highlights of our simulations are as follows:

(1) **Impact of cycle length:** Cycle length determines the queuing delay experienced by packets and also the frequency of channel switching. We observe that a long traffic cycle increases transmission delay and packet losses, whereas a short traffic cycle decreases throughput. For the study on impact of cycle length we consider both unicast and broadcast traffic in the network. However, for other studies we only simulate the broadcast traffic.
(2) **Video Quality:** The quality of the video is an indicator of the broadcast traffic received at the user nodes. We observe that COST has lower number of APs selected in the backbone, lower broadcast traffic load on the APs, and lower average delay, resulting in higher average PSNR.
(3) **User Density:** We observe that increase in the user density results in an increase in the number of APs in tree.
(4) **AP Density:** We observe that increase in the AP density results in a decrease in the number of APs in the tree.
(5) **User Speed:** We observe that the speed of users does not significantly affect the number of APs in the tree.
(6) **Optimality:** As the optimal solution is NP-hard, we compare the performance of SS and COST to the optimal solution for small network configurations (owing to the high computation time required for finding optimal solutions).

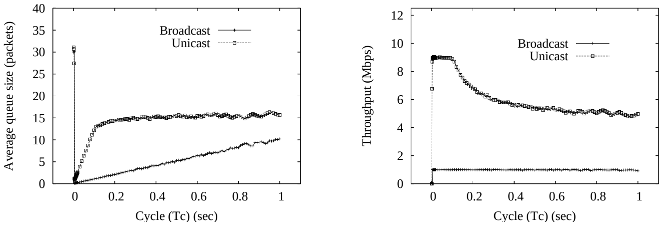
6.1 Impact of the length of cycle on performance

We examine the impact of cycle length, T_c, on the MAC layer performance. In these simulations, $\alpha = 0.5$. The network has one user node switching between a unicast-AP and a broadcast-AP. The user is receiving a downstream broadcast flow of 1 Mbps and unicast flow of 9 Mbps. The raw data rates for broadcast and unicast in the channel are 54 Mbps and 6 Mbps respectively. Each packet



(a) Average delay (b) Loss probability

Fig. 7. Cycle Length (a) Average packet delay and (b) packet loss probability with respect to cycle length, T_c. $\alpha = 0.5$. Both unicast and broadcast queues are 40 packets each.



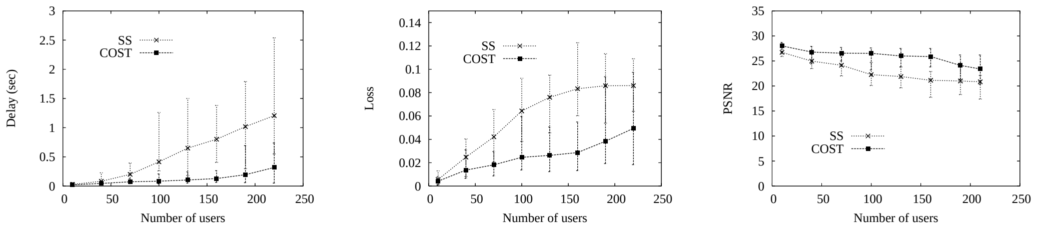
(a)Queue length (b) Throughput

Fig. 8. Cycle Length (a) Queue length and (b) Number of received packets with respect to the cycle length, T_c. $\alpha = 0.5$. Both unicast and broadcast queues are 40 packets each.

has 1460 bytes. The length of the broadcast and unicast interface queues are 40 packets each.

Figure 7 shows the transmission delay and loss probability. Figure 8 shows the queue length and the throughput. We observe that if the cycle length is only a few milli-seconds, the performance can be drastically bad as the length of packet transmission is of the order of the cycle length (broadcast packet duration is 1.94 ms). A low value of T_c leads to increased losses at the switching boundary between unicast and broadcast periods.

Although a high value of T_c reduces the switching overhead, for very high values of T_c, the queues start to overflow. For example, the unicast queue starts to overflow in the broadcast period if the broadcast period is long. As α is fixed in our simulations, increasing T_c increases both the broadcast and the unicast periods leading to increased queue losses. As the unicast flow rate (9 Mbps) is higher than the broadcast flow rate (1 Mbps), the unicast buffer starts to overflow for a lower value of T_c than the broadcast flow. We observe that beyond 100 ms the throughput of unicast flows degrades sharply (Figure 8(b)). The critical length of the broadcast duration when the unicast buffer starts to overflow is determined analytically by the ratio of the total capacity of the queue and the amount of unicast data generated in one broadcast period (i.e., $\frac{1460 \times 8 \times 40}{9000000} = 51.9ms$). Thus ignoring the backoff delay, the critical value of



(a)average delay (b)average loss (c) average PSNR

Fig. 9. Performance of video packets with respect to the number of users.

T_c is twice of $51.9ms$, i.e., $103.8ms$. So according to this calculation T_c should be smaller than $103.8ms$, which matches very closely with the critical points in the four graphs. In general, critical values for both unicast and broadcast queue overflow can be determined to choose the right value for T_c. Note that some applications may not be willing to incur huge latencies even if deep queues allow for a long T_c. The above technique can also be used to adjust the value of α (keeping T_c fixed) to achieve equal queue drop rate for unicast and broadcast flows.

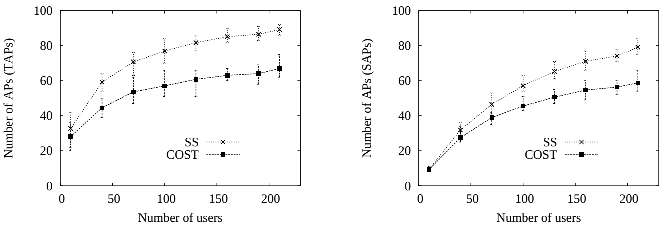
6.2 Video Quality

We have implemented a simple video streaming server and a client. The server sends encoded video with a variable data rate. The client performs 2 seconds pre-buffering to compensate for the bandwidth fluctuations of the wireless channel. Although techniques such as forward error correction, ARQ, and rate control with users feedback can enhance the video quality, these are beyond the scope of the paper.

Figure 9(a) indicates that the average delay increases for SS with increase in the number of users. This is due to the selection of more number of APs in the broadcast tree leading to increased congestion. However, with COST there is only a modest increase in delay with increasing number of users, as the COST metric is designed to minimize the broadcast load. The loss rate for COST is lower than SS as shown in Figure 9(b). This results in higher PSNR for COST (Figure 9(c)).

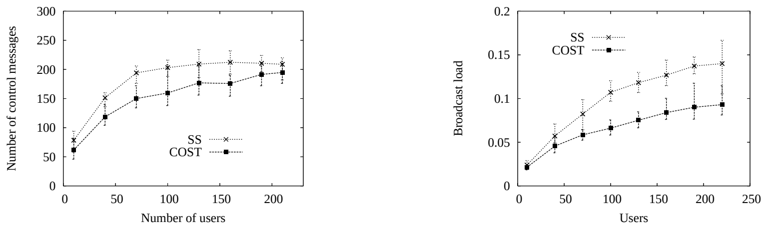
6.3 User density

Figure 10(a) an 10(b) show the number of TAPs and SAPs with respect to the number of users. As the number of users increases, the number of TAPs also increases. We observe that COST has lower number of TAPs than SS. For 160 users, the number of TAPs for COST is 25% lower and SAPs is 23%



(a) TAP (b) SAP

Fig. 10. User Density: Number of selected APs and control messages with respect to the user density.



(a) Control messages (b) Broadcast Load per AP-interface

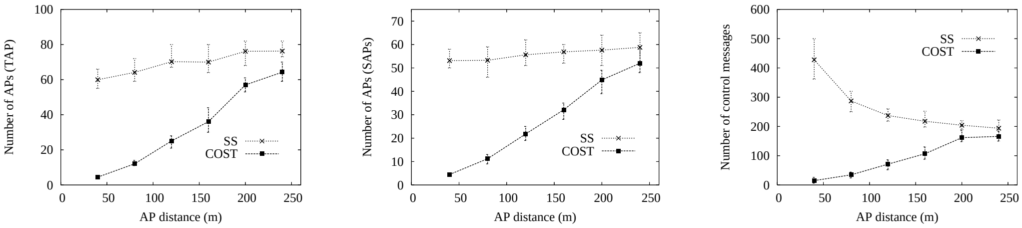
Fig. 11. User Density: Control messages and average fraction of time spent per AP-interface in broadcasting with respect to user density.

lower than SS

Figure 11(a) depicts the number of control packets transmitted to maintain the broadcast tree. SS uses a larger number of control messages. In case of SS, frequent changes of associated AP results in higher control messages. The average number of control messages for COST is 17% lower than SS for 160 users. Figure 11(b) depicts the average fraction of time spent in broadcasting by the two interfaces of the APs. On average, the interfaces of APs in SS consume 0.12 fraction of time to transmit broadcast traffic when the number of users is 160. However, for the same number of users, using the COST metric requires only 0.08 fraction of time per AP interface to deliver the broadcast packets. The freed up resources can be used by unicast users.

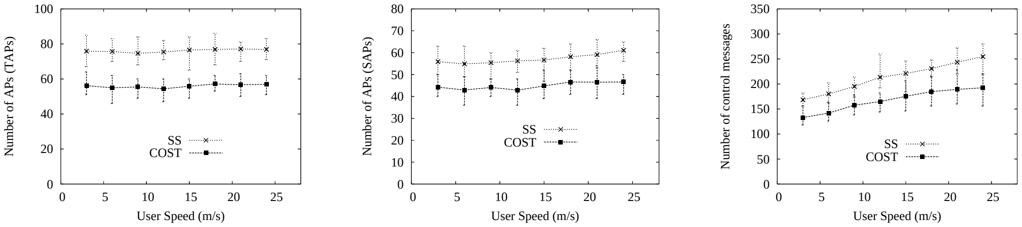
6.4 AP Density

Figure 12(a) and 12(b) show that the number of TAPs and SAPs increase with decrease in AP density (increase in inter-AP separation). In dense scenarios the SAPs and TAPs are much smaller for the COST metric as the COST metric is designed to exploit higher density of APs. As the network gets sparser, the difference between COST and SS is reduced due to fewer instances of node coverage from multiple access-points. Figure 12(c) shows that the number of control messages for tree management in the backbone is small for the COST



(a) TAP (b) SAP (c) Control messages

Fig. 12. AP Density: The number of APs in the tree, and the number of tree maintenance packets with respect to the AP density. Number of users is 100.



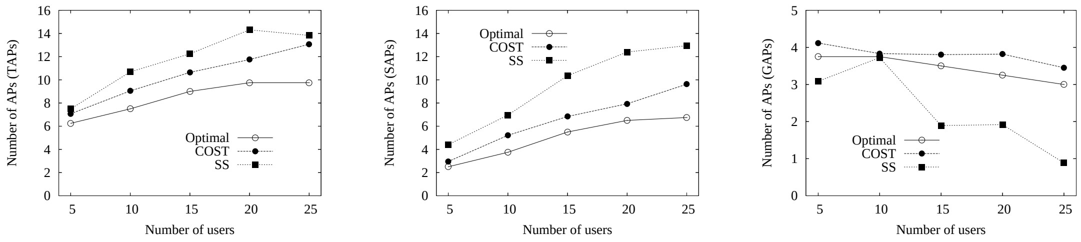
(a) TAP (b) SAP (c) Control messages

Fig. 13. User Speed: The number of APs and control messages in tree with respect to the user speed (m/s). Number of users are 70.

metric. For higher density, there are fewer changes in the tree for the COST metric as nodes often find a tree node in the vicinity. However, for SS, small movement of a node can trigger association with a new AP. With decreasing density, for SS larger movement is needed to trigger a change in association. So, the number of control messages decreases with reduction in AP density. With decreasing density, for the COST metric the advantage of multiple coverage is reduced leading to increased control packets. For 30m separation between adjacent APs in the grid, a reduction of 96% in terms of the number of control messages can be achieved by using the COST metric.

6.5 User speed

Figures 13(a) and 13(b) show the number of TAPs and SAPs versus the maximum speed of a user. We observe that the number of TAPs and SAPs do not vary significantly with the user speed. Figure 13(c) shows that the number of control messages for COST is always lower than SS, as COST leverages higher density of APs. With increasing speed the two protocols require increasing number of control messages due to more frequent changes in associations. The COST metric can reduce the number of control messages by up to 24.4% (for user speed of 24m/s).



(a) TAPs (b) SAPs (c) GAPs

Fig. 14. Optimality (a) the number of TAPs, (b) the number of SAPs, and (c) gateway APs (|GAP|=|TAP|-|SAP|) with respect to the number of users.

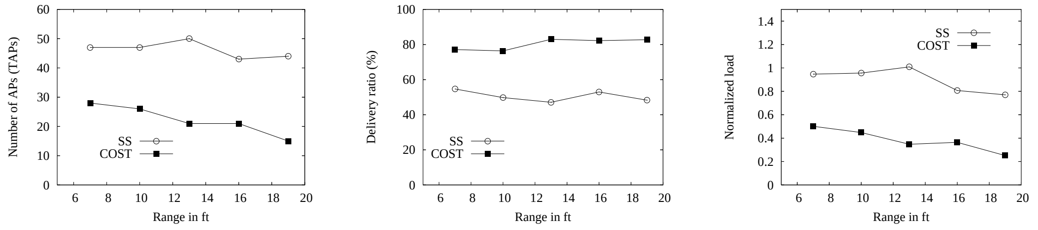
6.6 Optimality

In this section, we evaluate the optimality of COST. To calculate the optimal broadcast tree, we use an ILP (integer linear program) solver. We reduce the MESH-ST problem to the Steiner tree problem as shown in Figure 17 in Appendix A, and compute the Steiner tree by using the ILP solver. Note that the Steiner problem is NP-hard. As the ILP solver takes exponential time to arrive at solutions, we limit our study of optimality to small networks (16 APs and 25 users).

Figure 14 shows the average number of TAPs, SAPs, and GAPs (Gateway APs, i.e., TAPs that are not SAPs) for SS, COST, and the optimal association. As the number of users increases, the number of TAPs and SAPs increases as well. We observe that the number of SAPs for COST is closer to optimal than SS (Figure 14(b)). Compared to the optimal for the case of 20 users, COST and SS have 17% and 31.8% more TAPs, and 18% and 47.5% more SAPs, respectively. From Figure 14(c), we observe that as the number of users increases, the number of GAPs decreases. However, COST has more GAPs than SS. As the GAPs do not forward broadcast packets to their users, their contribution to the total load is lower than that of SAPs. Thus, for the same number of total APs, higher number of GAPs implies a lower broadcast load. Thus, from Figures 14(a) and Figure 14(c), we conclude that COST achieves lower broadcast traffic load than SS although the broadcast tree is larger than the optimal tree.

7 Testbed Evaluation

In this section we present the results obtained by implementing and testing our protocol on Kansei [27,28], which is an indoor testbed of 180 Stargate nodes. The Stargate (sold by Crossbow Inc.) is a 32-bit hardware platform running Linux, which has a PCMCIA wireless interface and an Ethernet interface. The



(a) TAPs (b) Delivery ratio (c) Normalized load

Fig. 15. Experimental results: (a) The number of TAPs with 52 APs and 128 users, (b) Delivery ratio with respect to range, and (c) Normalized Load with respect to range.

nodes are arranged in a 15×12 grid, with an inter-node separation of about 3 feet. An IEEE 802.11b card is used in each node for testing our protocols. The Ethernet interface of all the Stargates are connected together through hubs and switches to a central server, which is used to control the experiments. The Ethernet network is used to load programs, start-stop experiments, and monitor them, while the nodes use their wireless interface to run the protocols.

Alternate nodes in alternate rows are made APs and the rest of the nodes are made users. With this configuration, the network has 52 APs and 128 users. The range of the 802.11 wireless cards can be dynamically configured by changing the power level of the card and also by enforcing a logical topology on the Stargates. For our experiments, we varied the logical range from 7 feet to 19 feet, in steps of 3 feet and studied the total number of APs in the tree. To enforce the logical topology, we programmed the nodes to identify their location and then throw away packets that are from nodes farther than 19 feet. Though the interference patterns observed in this case would be different from an outdoor testbed, this is an innate limitation of any indoor experiment. The protocols were developed and tested using a software system known as Emstar [29]. In our experiments the Active Scan 5.3 technique was used for scanning channels which does not require global synchronization. The experiment uses broadcast traffic from the MAP.

Figure 15(a) compares the size of the tree for COST and SS. We observe that for COST as the range of the nodes increases, the number of nodes in the tree decreases. With increased range, each user has more APs to choose from. The COST metric attempts to minimize the number of selected APs, which is clearly shown in the figure. In SS, increasing the transmission range increases the choices for each user, but the best AP (closest and with highest signal strength) does not change. We observe that our approach can reduce the size of the tree and the mesh network traffic by up to 70% in the tested scenarios.

Figure 15(b) compares the delivery ratio obtained in SS and the cost based approach. We observe that the delivery ratio obtained in the cost based approach is up to 76% higher than that of the signal strength based approach. Given the same interference pattern in the network, the increased delivery ratio of the cost based approach is attributed to reduced collision due to smaller size of the broadcast tree. It is critical to note that on average, SS achieves a delivery ratio of 50% which is too low for the reconstruction of the video file that is being transmitted. In contrast, the cost based approach on an average achieves a delivery ratio of 80%. The decreasing trend in the delivery ratio for SS with increasing range is because of unreliable long links that are formed in the backbone. This effect although observed for the cost based approach is offset by the benefits of the reduced tree size.

Figure 15(c) plots the normalized load for both SS and cost based approaches. We measure load in terms of the total number of packet transmissions. The normalized load is the broadcast load of the network normalized over the delivery ratio. The normalized load for the cost based approach goes down as the range increases. This is because of a reduction in the total number of packet transmissions and increase in number of received packets. In SS, the network load goes up with increase in range due increased collision and reduced delivery ratio.

8 Conclusion

In this paper we studied a novel technique for association that reduces the load for broadcast traffic in mesh-networks. We propose the concept of dual-association, where the AP for unicast traffic and the AP for broadcast traffic are independently chosen by exploiting multiple coverages that are typical in mesh networks. We propose a cost metric based on ETT and the number of nodes in range of the APs, that is advertised in the beacons from the APs. Users periodically scan and associate with the AP with the lowest cost metric. Using extensive simulations and experiments on an indoor testbed of 180 802.11b devices we evaluated the performance of our approach. We observed that the load can be reduced and the performance of both unicast and broadcast data services can be significantly improved using our approach.

9 Acknowledgments

This material is based upon work supported by the National Science Foundation under Grants CNS-0546630 (CAREER Award) and CNS-0403342. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

[1] Wireless Mesh Networks, http://www.ece.gatech.edu/research/labs/bwn/mesh/work.html.

[2] Tradeoffs between BSS and IBSS mode of operations, http://www.bawug.org/howto/pres/20010111/UNIX-AP_files/frame.htm.

[3] Y. Bejerano, S.-J. Han, L. Li, Fairness and Load Balancing in Wireless LANs Using Association Control, in: Mobicom, 2004.

[4] E. Royer, C. Perkins, Multicast ad hoc on- demand distance vector (maodv) routing (2000).
URL `citeseer.ist.psu.edu/royer00multicast.html`

[5] D. Lee, G. Chandrasekaran, P. Sinha, Optimizing broadcast load in Mesh Networks using Dual-Association, in: 1st IEEE Workshop on Wireless Mesh Networks (WiMESH), Santa Clara, CA, USA, 2005.

[6] D. Aguayo, J. Bicket, S. Biswas, G. Judd, R. Morris, Link-level Measurements from an 802.11b Mesh Network, in: Sigcomm, 2004.

[7] R. Draves, J. Padhya, B. Zill, Routing in Multi-radio Multi-hop Wireless Mesh Networks, in: Mobicom, 2004.

[8] R. Chandra, P. Bahl, P. Bahl, MultiNet: Connecting to Multiple IEEE 802.11 Networks Using a Single Wireless Cards , in: IEEE INFOCOM, Hong Kong, 2004.

[9] B. Das, R. Sivakumar, V. Bharghavan, Routing in Ad Hoc Networks Using a Spine, in: Proc. of the 6th International Conference on Computer Communications and Networks , 1997, p. 34.

[10] R. Sivakumar, P. Sinha, V. Bharghavan, CEDAR: a Core-Extraction Distributed Ad-hoc Routing algorithm, Journal on Selected Areas in Communications (Special Issue on Ad-Hoc Networks) 17 (8) (1999) 1454–1465.

[11] S. Guha, S. Khuller, Approximation Algorithms for Connected Dominating Sets, Algorithmica 20 (4) (1998) 374–387.

[12] P. J. Wan, K. M. Alzoubi, O. Frieder, Distributed construction of connected dominating set in wireless ad hoc networks, Mobile Networks and Applications 9 (2004) 141–149.

[13] C. Chou, A. Misra, Low latency Multimedia Broadcast in Multi-rate Wireless Meshes, in: First IEEE Workshop on Wireless Mesh Networks, 2005, pp. 54–63.

[14] R. Draves, J. Padhye, B. Zill, Routing in Multi-Radio Multi-Hop Wireless Mesh Networks, in: ACM Mobicom, 2004.

[15] M. Gerla, C.-C. Chiang, L. Zhang, Tree Multicast Stategies in Mobile Mulithop Wireless Networks, ACM Mobile Networks and Applications 4 (1999) 1193–207.

[16] C. Jaikaeo, C.-C. Shen, Adaptive Backbone-Based Multicast for Ad-hoc Networks , in: Proc. of ICC, New York, NY, 2002.

[17] E. M. Royer, C. E. Perkins, Multicast Operation of the Ad hoc On-Demand Distance Vector Routing Protocol, in: ACM MOBICOM, Seattle, WA, 1999, pp. 207–218.

[18] E. Bommaiah, M. Liu, A. McAuley, R. Talpade, Ad-hoc Multicast Routing protocol utilizing Increasing id-numberS (AMRIS) Functional Specification , Internet-Draft, draft-ietf-manet-amris-spec-00.txt (Nov. 1998).

[19] S.-J. Lee, M. Gerla, C.-C. Chiang, On-Demand Multicast Routing Protocol, in: WCNC, New Orleans, 1999, pp. 1298–1304.

[20] J. J. Garcia-Luna-Aceves, E. Madruga, The Core-Assisted Mesh Protocol, IEEE Journal on Selected Areas in Communications 17 (8) (1999) 1380–1394.

[21] H. Zhang, A. Arora, P. Sinha, Learn on the Fly: Quiescent Routing in Sensor Network Backbones, in: OSU-Technical Report OSU-CISRC-7/05-TR48, Hong Kong, China, 2005.

[22] Maxim 2.4GHz 802.11b Zero-IF Transceivers, http://pdfserv.maxim-ic.com/en/ds/MAX2820-MAX2821.pdf.

[23] P. Slavik, A Tight Analysis of the Greedy Algorithm for Set Cover, in: ACM STOC, Philadelphia, PA, USA, 1996, pp. 435–441.

[24] I. Ramani, S. Savage, SyncScan: Practical fast handoff for 802.11 infrastructure networks, in: Proc. of the IEEE Infocom Conference, Miami, FL, USA, 2005.

[25] J. Elson, L. Girod, D. Estrin, Fine Grained Network Time Synchronization using Reference Broadcasts, in: OSDI, Vol. 36 Winter, 2002, pp. 147–163.

[26] ns2: Network Simulator, http://www.isi.edu/nsnam/ns/.

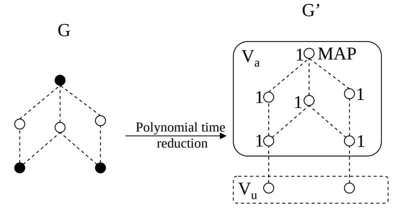
[27] E. E. et. al., Kansei: A Testbed for Sensing at Scale, in: Fourth Symposium on Information Processing in Sensor Networks (IPSN/SPOTS track), 2006.

[28] A. Arora, E. Ertin, R. Ramnath, W. Leal, M. Nesteranko, Kansei: A high-fidelity sensing testbed, IEEE Internet Computing, Special issue on Large-Scale Sensor Networks 10 (2006) 35–47.

[29] EmStar: Software for Wireless Sensor Network, http://cvs.cens.ucla.edu/emstar/.

Appendix. A

Theorem 1: [Lower bound] MESH-ST is at least as difficult as ST.
Proof: We present a reduction from an arbitrary instance of the ST problem to an instance of the MESH-ST problem. Let G be the graph with a set of required nodes R in the ST problem. We construct the new graph G' by modifying G as follows. Let $MAP \in R$ be an arbitrary required node. Create a new node v' for each required node v in R except the MAP. Join v and v' with an edge. All the newly added nodes constitute V_u and the old nodes constitute V_a. The weight of all nodes in V_a is unity. We claim that when the leaf edges are deleted from the solution Z' to MESH-ST(G'), we obtain a solution Z to ST(G).



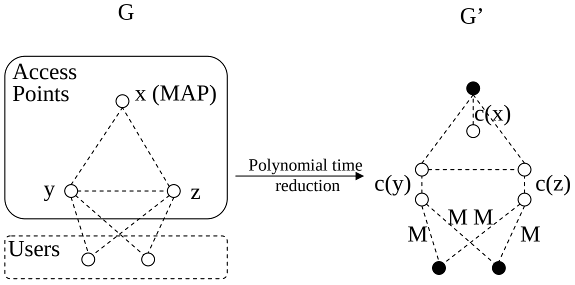
(a) ST(G) (b) MESH-ST(G')

Fig. 16. Reduction from ST to MESH-ST. The highlighted nodes are the required nodes for the Steiner tree problem.

It is easy to see that by removing the leaf edges from a tree, we still obtain a tree. We now prove that if there exists a solution to ST(G) that is smaller than Z, then it leads to a contradiction, thus completing the proof. The cost of a Steiner tree does not include the cost of any nodes. But Z' includes the cost of $|V_u|$ nodes. Therefore, $Cost_{Steiner}(Z) = Cost_{MESH-ST}(Z') - |V_u|$. Let us assume that there exists a solution Y to ST(G), such that $Cost_{Steiner}(Y) < Cost_{Steiner}(Z)$. By augmenting Y with the edges connecting the nodes in V_u, we obtain a tree in G' whose $Cost_{MESH-ST}$ is $Cost_{Steiner}(Y) + |V_u| < Cost_{Steiner}(Z) + |V_u| = Cost_{MESH-ST}(Z')$. As the new tree's cost is lower than $Cost_{MESH-ST}(Z')$, it contradicts the optimality of Z'. **QED.**

Theorem 2: [Upper bound] MESH-ST is at most as difficult as ST.

Proof: We present a reduction from an arbitrary instance of the MESH-ST problem to an instance of the ST problem. Let G be the graph with an instance of the MESH-ST problem. We split each node $v \in V_a$ into v_1 and v_2. All edges incident from V_u on v are now incident on v_2 and the other edges are incident on v_1. A new edge is added between v_1 and v_2 with a cost equal to $c(v)$. The weight of all the edges incident on V_u are M, where M is a very large number (larger than the sum of weights of all AP-AP edges). The MAP and the nodes in V_u are the required nodes in the new graph G'. We claim that the solution Z' to ST(G') represents a solution to MESH-ST(G). The corresponding solution Z for MESH-ST(G) excludes the newly added links.



(a) MESH-ST(G) (b) ST(G')

Fig. 17. Reduction from MESH-ST to ST. The highlighted nodes are the required nodes for the Steiner tree problem.

Note that due to the high cost on the edges to the required nodes in $ST(G')$, only the minimum number of such edges will be selected in Z'. This guarantees that all nodes in V_u will be leaf nodes in Z'. We now prove that if there exists a solution to MESH-ST(G) that is smaller than Z, then it leads to a contradiction, thus completing the proof. As $Cost_{MESH-ST}$ does not include the cost of the leaf edges, $Cost_{MESH-ST}(Z) = Cost_{ST}(Z') - M|V_u|$. Let us assume that there exists a solution Y to MESH-ST(G), such that $Cost_{MESH-ST}(Y) < Cost_{MESH-ST}(Z)$. By adding the corresponding newly added links to Y, we get a tree in G' with a cost of $Cost_{MESH-ST}(Y)+M|V_u| < Cost_{MESH-ST}(Z) + M|V_u| = Cost_{ST}(Z')$. As the new tree's cost is lower than $Cost_{ST}(Z')$, it contradicts the optimality of Z'. **QED.**