Channel Access Throttling for Overlapping BSS Management

Bo Han*, Lusheng Ji†, Seungjoon Lee†, Robert R. Miller†, Bobby Bhattacharjee*
*Department of Computer Science, University of Maryland, College Park, MD 20742, USA
†AT&T Labs – Research, 180 Park Avenue, Florham Park, NJ 07932, USA

***Abstract*—Multiple co-channel WLAN BSSes (i.e., WLAN cells) overlapping in coverage are generally considered undesirable because members of the OBSSes compete for channel access, which typically increases the contention level of wireless medium access and reduces overall system performance. In this paper, we propose to use Channel Access Throttling (CAT) for managing Wireless LAN radio resources for overlapping BSSes (OBSSes). CAT provides an Access Point (AP) of each BSS with a mechanism to control channel access parameters of its member stations on the fly. By coordinating the CAT operations of the OBSS APs, we can enable privileged channel access to an individual BSS at a particular time, for example, by assigning high priority access parameters to member stations associated with the BSS. By controlling how much each BSS may be given the privileged channel access, we can also achieve a proportional partitioning of channel capacity among OBSSes. We present evaluation results obtained from both simulations and experiments using testbed built with Commercial Off-The-Shelf (COTS) WLAN hardware and open-source device driver. Our results show that with CAT, not only can we proportionally partition channel capacity among the OBSSes, but also improve channel utilization efficiency and increase overall capacity.**

***Index Terms*—Wireless LAN; overlapping BSS; IEEE 802.11e; channel access.**

I. INTRODUCTION

IEEE 802.11 based Wireless LAN (WLAN) technology has seen phenomenal growth in popularity over the past decade. A result of this tremendous success is over-crowded deployments of WLAN systems in certain areas, e.g., busy commercial areas such as shopping arcades and compact residential areas such as Multi-Dwelling Unit (MDU) buildings [1]. In these areas, WLAN cells often have to share channels with other WLAN cells in close proximity. When nearby co-channel WLAN cells overlap with each other in coverage, they become what are known as the Overlapping Basic Service Sets (OBSSes). OBSS is generally considered undesirable because members of the OBSSes may interfere with each other and compete for channel access, causing increased channel contention level and decreased performance [2]. Moreover, OBSS also makes determining and planning the capacity of WLANs more difficult, as different BSSes typically have different parameter configuration, hardware characteristics, and distribution of the member stations.

The intuitive solution to the OBSS problem is to assign orthogonal channels to neighboring BSSes [3]. However, because radio spectrum is a scarce resource, when the density of WLAN systems exceeds the number of available channels, OBSS may be inevitable. For example the 2.4 GHz Industrial Scientific and Medical (ISM) band can only accommodate three orthogonal WLAN channels. Although the 5 GHz Unlicensed National Information Infrastructure (U-NII) band has a sufficient number of channels, for several reasons manufacturers of certain types of devices and service providers remain favoring the ISM band. One of such reasons is that radio signal attenuation in 5 GHz is worse than 2.4 GHz over the same distance and condition, which causes shorter signal reach in 5 GHz with the same transmission power. As a result, service providers would have to deploy more APs to cover the same area in 5 GHz than in 2.4 GHz. Moreover, designers for cost and power conscious devices such as battery-powered hand-held devices also often lean towards only supporting 2.4 GHz communication (e.g., Apple iPhone, Nokia N80). In addition, backwards compatibility requirement for legacy WLAN devices which only operate in the ISM band and regulation differences among different countries for license free usage in the 5 GHz band also make the 5 GHz band less appealing. As a result, the 2.4 GHz band remains crowded and the problem of how to manage radio resources both effectively and efficiently among OBSSes is still open.

In this paper, we propose to use a new approach called CAT (Channel Access Throttling) to manage channel capacity for overlapping BSSes. CAT enables an AP to control channel access parameters of its member stations on the fly and dynamically assign different channel access parameters to different member stations. Using this mechanism, an AP can effectively grant prioritized medium access to an arbitrary set of member stations, where the high-priority set can change over a short period of time. By coordinating the CAT operations of the OBSS APs, we can enable privileged channel access to an individual BSS at a particular time. In this paper, we focus on the strong OBSS scenario, where all APs and member stations can hear from each other. In this scenario, APs can overhear other AP's messages (e.g., periodic Beacon messages) and process them for schedule coordination and synchronization. By controlling how much time each OBSS may be given the prioritized channel access, we can also achieve a proportional partitioning of channel capacity among OBSSes. At the same time, because during the allocated duration of each OBSS the holder enjoys privileged access to channel, we can improve channel access efficiency and therefore the overall channel capacity for all OBSSes.

The contributions of this paper are three-fold:

- We present the concept and design of CAT and describe how we can use CAT to help OBSS management.
- We describe our implementation of CAT using commercial off-the-shelf (COTS) hardware and an open-source

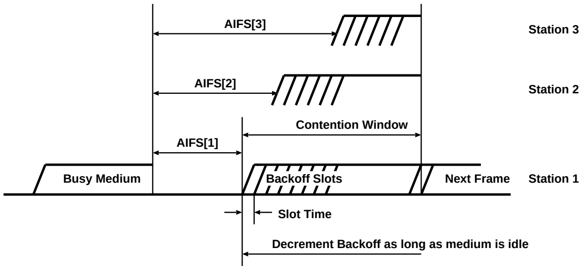

Fig. 1. EDCA Channel Access

device driver.
- We perform both simulation and testbed experiments to demonstrate the advantages of CAT.

The rest of this paper is organized as follows. We first give a brief introduction of the EDCA channel access mechanism in Section II. In Section III, we describe the detailed design of CAT and describe how we use it to manage OBSSes. We describe the CAT implementation and testbed experiment results in Section IV, and simulation evaluation results in Section V. After reviewing related work in Section VI, we conclude in Section VII.

II. BACKGROUND

Virtually all WLAN devices sold today support the channel access mechanism called Enhanced Distributed Channel Access (EDCA). EDCA is defined by the IEEE 802.11e amendment [4]. It supports four different Access Categories (ACs), i.e. AC_VO (voice), AC_VI (video), AC_BE (best effort) and AC_BK (background), and provides Quality of Service (QoS) differentiation between the categories.

An IEEE 802.11e QoS station has a separate output queue and a backoff procedure called EDCA function for each of these ACs. Each AC uses a different set of channel access parameters, which essentially specifies how a station should defer its transmission if encountering a busy channel, and thus get differentiated service. EDCA uses the following channel access parameters:

- *Arbitration InterFrame Space Number (AIFSN)* controls the duration of Arbitration InterFrame Space (AIFS): *AIFS = AIFSN* $\times$ *aSlotTime + SIFS*, where aSlotTime is the duration of a time tick in the WLAN system and the SIFS is the smallest inter-frame gap.
- *CWmin and CWmax* define the region for Contention Window (CW) value. The upper bound for CW is initially set to CWmin, and doubles on every transmission failure, until it reaches CWmax, after which it stays constant at CWmax. The actual CW is chosen randomly between 0 and this upper bound.
- *Transmission Opportunity Limit (TXOPLimit)* is the maximum duration for TXOP (Transmission Opportunity). During a TXOP, the owner member station may transmit multiple frames without observing the usual channel access deferring rules.

For example, the recommended default values of these parameters for IEEE 802.11a/g are summarized in Table I. An AP announces these parameters to its associated stations in its Beacon messages.

	Description	CW_min	CW_max	AIFSN	TXOPLimit
AC_VO	Voice	3	7	2	1.504 ms
AC_VI	Video	7	15	2	3.008 ms
AC_BE	Best Effort	15	1023	3	0 ms
AC_BK	Background	15	1023	7	0 ms

TABLE I
EDCA PARAMETERS FOR IEEE 802.11A/G

Figure 1 illustrates the channel access mechanism for EDCA. Each backoff procedure independently calculates the waiting time before the next transmission. First, the backoff procedure waits for AIFS duration and checks if the channel is idle for the entire AIFS period. If it is, the contention window phase begins with a backoff counter initialized to a random number between 0 and the current CW upper bound. The backoff counter will then be decreased in each following idle time slot. When a backoff counter reaches zero, the station can attempt to transmit a frame in the corresponding AC. If backoff counters of multiple backoff procedures simultaneously become zero in the same QoS station, the highest-priority procedure wins the internal contention, and all other procedures act as if an external collision occurred. Finally, the winner procedure still needs to contend with procedures from other stations for the wireless channel.

III. CAT FOR OBSS MANAGEMENT

A. The Channel Access Throttling Method

The key idea behind CAT is that compared to EDCA, which differentiates channel access priorities among different ACs, CAT differentiates access priorities among member stations. In its simplest form, CAT can employ only two priority groups: a high access priority (or *CAT-high*) group and a low access priority (or *CAT-low*) group. CAT achieves the priority differentiation between the two groups using different EDCA channel access parameters, just as EDCA differentiates the four ACs. Again, in its simplest form, CAT may assign only one member station into the CAT-high group and all the rest to the CAT-low group. In this case, the only CAT-high station gets “exclusive” channel access, because all the other stations have low priority and will not win channel access. CAT may rotate which member station becoming a CAT-high station according to a schedule to partition channel capacity among the member stations. To keep the discussion simple yet illustrative, in the rest of this section, we focus on the simplest CAT configuration with only two priority groups, while we can easily generalize CAT to multiple groups (as discussed later in this section).

We consider two approaches to throttling member station’s channel access: periodic or on-demand. In the *periodic* approach, the AP sets up a schedule relative to a periodic reference time that is available to all member stations. For instance, the beginning of each Beacon interval, also known as the Target Beacon Transmission Time (TBTT), can be used as a reference time in practice. The time between two consecutive reference times is the service cycle period. A CAT schedule configuration contains two sets of EDCA channel access parameters, one for CAT-high and the other for CAT-low. The configuration also lists the starting and ending times for each station to be in CAT-high group and CAT-low group within each service cycle. After receiving the configuration, each member station needs to periodically adjust its EDCA channel access parameters to the specified values at specified times according to its membership in either CAT-high or CAT-low group. CAT can also throttle channel access *on-demand*. In this case, the AP may announce the CAT-low parameters as the default configuration in its Beacon messages just as how a regular EDCA AP announces EDCA parameters to its associated member stations. Then, at specific times, the AP sends CAT-high channel access parameter configuration to a specific member station in a similar fashion to polling messages used in scheduled channel access mechanisms.

Using CAT, we can actually improve the channel access efficiency by setting the CAT-high parameters (e.g., AIFSN, CWmin, and CWmax) to the lowest possible values and minimizing access deferral. Although this emulated channel access by CAT may not be as efficient as true scheduled access mechanisms, CAT does have an advantage over them. For instance the IEEE 802.11e also defines a scheduled channel access mechanism called the HCF (Hybrid Control Function) Controlled Channel Access or HCCA. In HCCA, if a station is polled and given a CAP (Controlled Access Phase), but does not transmit packets, the CAP is wasted unless it is recovered through an explicit rejection by the station or through a timeout by the AP. In CAT, because CAT-low stations are not completely forbidden to access the channel, they still can utilize such a missed opportunity and transmit packets, which can improve overall channel utilization.

B. Using CAT for OBSS Management

We can use CAT to assign different channel access priorities to different OBSSes for different time slices in a controlled and synchronized fashion. For example, during a period of time a single OBSS is given prioritized channel access probability over the other OBSSes, which effectively allows the member stations associated with the OBSS to receive a dedicated allocation of airtime. Because within each OBSS's own time allocation, its member stations do not compete with stations of other OBSSes for channel access, the overall system channel access efficiency can be improved. We can rotate the channel access priorities among the OBSSes, such that we can allow member stations in each BSS to transmit and receive. By controlling how much time each OBSS may remain in CAT-high state, proportional partitioning of channel capacity can also be achieved. Note that within the BSS-level CAT-high period, a BSS can apply a CAT schedule that assigns different channel access priorities to its individual member stations.

To coordinate CAT operattions among multiple OBSSes, we need a mechanism for the APs of the OBSSes to communicate with each other to coordinate their CAT operations. The APs use such a communication mechanism to negotiate, coordinate, and synchronize each OBSSes channel access priority setting. In this paper, we focus on using CAT for the strong overlapping BSSes scenario, and this communication mechanism can simply be the direct communication over the wireless link, i.e., via front end radio interfaces, between the APs. For example, as in our experiments, we can use a Master AP (e.g., with the smallest ID), which sends out a BSS-level CAT schedule, potentially based on the number of BSSes and member stations. In this scenario, all the other APs and member stations can adjust and synchronize their clock based on the periodic messages from the Master AP. An obvious challenge is to determine adequate parameter sets for different BSSes that satisfy potentially diverse QoS requirements for different numbers of member stations. Note that this decision is subject to other limitation related to system parameters (e.g., AIFSN is only 4 bits long, hence the maximum value is 15). We further discuss this aspect later in this section.

Generalizations: The above two-priority-group model can be easily extended to more complex models. For example, we can assign more than one BSSes into the CAT-high group at a time. This can potentially improve the system utilization when some BSSes have light traffic load, and may not always have stations that are ready for frame transmissions. In this scenario, although CAT allows stations in CAT-low BSSes to transmit packets when they sense no channel activity for a sufficiently long time, we can further reduce channel idle time by assigning two or more BSSes to CAT-high group. We can consider further generalizing CAT to have more than two priority groups by assigning different channel access parameters. Furthermore, even different ACs within the same priority group may further be configured with different channel access parameters so they still receive differentiated treatments. Another direction for generalization is to use CAT to manage weakly overlapping BSSes, which is part of our ongoing work.

C. Choosing CAT Parameters

In general, the choice for CAT-high group channel access parameters should mainly depend on how many stations are put into the CAT-high group. Each CAT state transition typically incurs a fixed cost, and frequent changes in CAT-high groups results in lower overall channel efficiency. On the other hand, infrequent group changes typically require a larger number of CAT-high stations, which increase more channel competition among CAT-high stations. In this case, we need to set backoff windows carefully to resolve contention efficiently, depending on the number of OBSSes and the associated member stations.

Ideally, we want to set CAT parameters, such that we have sufficient separation between CAT-high and CAT-low stations. One possible approach to achieving this is employ existing analytical models for EDCA performance. For example, based on Bianchi's seminal work [5] that analyzes IEEE 802.11 DCF performance using a two dimensional Markov chain, Zhu and Chlamtac [6] propose a model that predicts the channel access probability for different EDCA parameter sets. Specifically, when there are two stations STA1 and STA2, and STA1 uses {AIFSN, [CWmin, CWmax]} = {2, [1, 3]}, the model estimates that STA2 using {2, [3, 7]} can split the channel access with STA1 at the ratio of 1:9, while STA2 using {2, [7, 15]} can access the channel less than 2% of time.

While channel access priority separation is certainly the primary concern for choosing parameters for CAT-high and CAT-low group, the CAT-low parameters should also address the concern of overall channel efficiency. This is because if the CAT-high group does not have data to send, the channel is idle and eventually some CAT-low station will complete its channel access deferral and begin to transmit. Since any unnecessary channel idling is a waste of radio resource, the channel access parameters for CAT-low group should be set in such a way to avoid unnecessarily long channel access deferral.

IV. TESTBED EXPERIMENTS

In this section, we present the experiment results of how CAT can effectively partition channel capacity between two OBSSes proportionally. We perform the evaluation using a testbed constructed with COTS WLAN hardware. We first describe the implementation details and then present the results.

A. *CAT Implementation for OBSS Management*

To implement CAT, we modify the open-source MadWifi wireless device driver (v0.9.4) [7]. The Atheros Hardware Abstraction layer (HAL) used by MadWifi driver provides a call interface for configuring EDCA related parameters. If a device receives a Beacon message with EDCA parameters, its MadWifi driver will commit these parameters into Atheros hardware. We modify the driver such that a station changes EDCA parameters when triggered by a timer (periodic) or when receiving a CAT message (on-demand).

The main challenge in the periodic approach is the synchronization of CAT timers among different member stations associated with different APs, as poor synchronization among CAT timers may cause overlapping of CAT-high periods of different stations and increase contention level during these overlapping periods. In our implementation we use the Beacon messages as timer synchronization signals and CAT operation cycle coincides with Beacon interval. Member stations repeat their CAT timer settings, i.e., when to enter and leave CAT-high and CAT-low states, every Beacon interval. Currently, we focus on the strong overlapping OBSS scenario where all the APs and member stations can hear from each other. We assign one AP as the synchronization Master AP (e.g., the one with the smallest MAC address) and others as Slave APs. We modified the driver code to make stations not filter out Beacon messages from the Master AP. Every time a new Beacon message from the Master AP is received by a station, the station re-synchronizes its CAT timer with a Beacon reception.

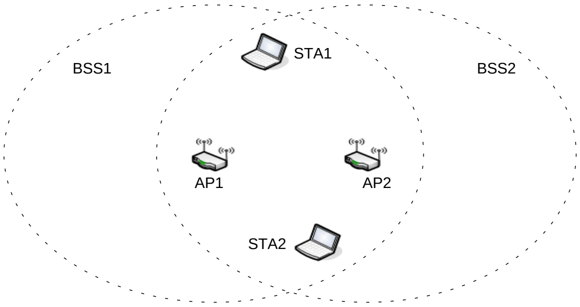

Fig. 2. Experiment Topology

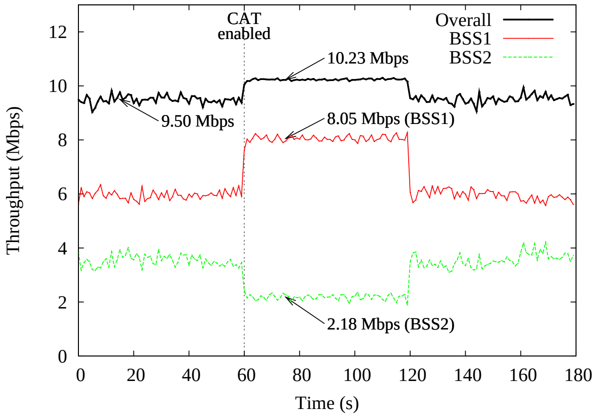

Fig. 3. Partitioning of Channel Bandwidth with CAT

During each CAT-high period, since we have only one CAT-high station in our experiments, we use the smallest possible parameters for efficiency and set {AIFSN, [CWmin, CWmax]} = {2, [0, 0]}. For CAT-low period, we use {15, [3, 7]}. As we mentioned before, by using small CWmin and CWmax values for the CAT-low state, CAT-low stations can access the channel rapidly when the CAT-high station has no packets to send. Also, the current MadWifi driver applies the EDCA parameters to only those packets awaiting in the output queues, and provides no mechanism to reset EDCA parameters for the packets already in transmission states (e.g. during AIFS, contention window counting down, etc). Therefore, any packet already in transmission states effectively blocks the station from entering its new CAT state. For this reason, using small CWmin and CWmax during CAT-low period in our experiments also reduces the impact of such blocking. In the following we show how we can use CAT to partition wireless channel capacity proportionally among member stations belonging to different BSSes.

B. *Experiment Results*

For this group of experiments, we use four wireless nodes—two APs and two member stations (each associated with one AP), placed within the transmission ranges of each other, as shown in Figure 2. We run Iperf [8] on the member stations to generate Constant Bit Rate (CBR) UDP streams at the rate of 12 Mbps. The packets are addressed to the AP. We set the type of service for all data packets to AC_VO category in EDCA because this is the EDCA configuration with the least access deferral and thus the highest efficiency. We set the PHY layer transmission rate for all nodes to be 12 Mbps and the transmission power to 17 dBm. The Beacon interval of the APs is 100ms. In this experiment, BSS1 is in the CAT-high state during the first 80% of each Beacon interval, and BSS2 is in the CAT-high state in the rest of 20%. When not in CAT-high state, these BSSes and their member stations are in CAT-low state.

We show the experiment results in Figure 3. CAT is enabled for both BSS1 and BSS2 at around 60s into the experiment and disabled at around 120s. When CAT is not enabled, the throughput is around 6 Mbps for BSS1 and slightly higher than 3 Mbps for BSS2, due to the difference of channel sensing capabilities of these stations. When CAT is enabled, the two stations achieve throughput close to the ratio of 8:2 (8.05 Mbps vs. 2.18 Mbps). Figure 3 also shows that using CAT can increase the average overall throughput of the system. Without CAT the overall throughput is about 9.50 Mbps. When CAT is enabled the overall throughput increases to 10.23 Mbps.

V. SIMULATION EXPERIMENTS

We have also conducted simulations to evaluate the performance of CAT for OBSS management. In this section, we particularly focus on the VoIP service capacity study which clearly demonstrates the benefits of CAT but is impractical to conduct real testbed experiments because of its scale. We show how many VoIP calls two OBSSes may support using CAT v.s. EDCA.

A. Simulation Setup

For the simulation study, we used an enhanced ns-2 802.11 module called *yans* [9]. Compared to the default 802.11 module in ns-2, the *yans* extension implements a more detailed physical layer model (i.e., Bit Error Rate (BER) based model) and a new MAC layer for 802.11e. This BER based model is closer to real WLAN receiver behavior and more realistic when simulating different transmission rates than the default ns-2 module. We configure 802.11 specific parameters according to IEEE 802.11a. In all simulation runs, we randomly and uniformly deploy member stations in a $n \times n$ square area. The locations of the two APs are ($n/5$, $n/2$) and ($4n/5$, $n/2$). While we use different transmission rates in different scenarios, within any single experiment the rate is fixed.

In our simulations, we focus on the scenario where all traffic is VoIP. We simulate CBR VoIP traffic generated by the G.711 voice codec. We use the same parameters as [10], with sample size being 80 bytes, sample interval at 10 ms, and delay bound of 50 ms. Each VoIP packet payload contains two samples. As a result, the VoIP packet interval is 20 ms and the MAC layer payload length is 200 bytes. The main factor that negatively impacts VoIP QoS is missing packets, which can occur due to packet losses (e.g., transmission errors, queue overflow) or packet expiry (i.e., packet arrival after delay bound). According to the ITU-T's E-model [11], a 5% packet missing rate is enough to reduce user satisfaction level on an otherwise perfect VoIP call to “some users dissatisfied”. Hence, in our experiments we use the 5% packet missing rate as a cut-off threshold to determine how many calls the two OBSSes can support. We increase the total number of stations in the two OBSSes one by one. These stations are randomly distributed between the two OBSSes. Each of the stations generates a full duplex VoIP call with its AP. If a newly added call (station) causes some already admitted calls including itself to have a larger than 5% packet missing rate on either direction for more than 100 seconds, we consider that we have just exceeded the maximum number of calls that the two OBSSes can support.

In the simulation, we use the periodic approach for CAT. At a given time, only one station enters CAT-high state in a round-robin fashion, where the duration of CAT-high period for each member station is equal. The APs also enter CAT-high state because VoIP traffic is bi-directional, and they need to send VoIP packets to their member stations. To meet the 50 ms delay requirement, we have multiple service cycles (i.e., 5) within each 100 ms Beacon interval. For the CAT-high period, we set CWmin = CWmax = 0, and set AIFSN to 1 for APs and to 2 for member stations. All these values are the minimum allowed in the standard, so that we make an efficient use of channel during the CAT-high period. To improve the channel utilization further, we allow the APs to send all downstream VoIP packets in their queues in one transmission opportunity (TXOP). For this, we use the number of active VoIP calls to calculate the appropriate TXOPLimit for the APs to finish sending a packet for each VoIP call. We also set the TXOPLimit for member stations to 0 ms, which means that they can only send out 1 packet for each TXOP. For CAT-low period, we use $\{$AIFSN, [CWmin, CWmax]$\}$ = $\{$15, [511, 1023]$\}$, which sufficiently separates the channel access probability between CAT-high and CAT-low groups. We also simulated the same scenario using EDCA and compared the results.

B. Results

In Figure 4, we show the maximum number of VoIP calls that can be supported by the two OBSSes using either CAT or EDCA, for different PHY layer transmission rates. Each data point represents an averaged result of 9 runs—three different random node placement scenarios and each with three runs with different random seeds. As we can see, the VoIP service capacity of OBSSes with CAT significantly exceeds that with EDCA. For instance, when the physical layer transmission rate is 54 Mbps, two OBSSes implementing CAT can support 85 calls, which is 41.67% higher than 60 calls that they can support with EDCA. This increase is due to several reasons. First, the downlink (AP to stations) is the major bottleneck for EDCA VoIP performance. While all downstream VoIP packets go through the AP, the EDCA configurations do not provide significant channel access priority advantage for the AP. As a result, packets in the AP's queue do not get enough channel access opportunities. In contrast, CAT is able to allocate channel bandwidth in a finer granularity and give the AP sufficient transmission opportunity to send all downstream packets and satisfy QoS constraints. Another reason for CAT's superior performance is that we use shorter contention windows in CAT-high state than in EDCA. The time saving due to shorter channel sensing and backoff allows the two overlapping BSSes to exchange more packets and thus support more calls.

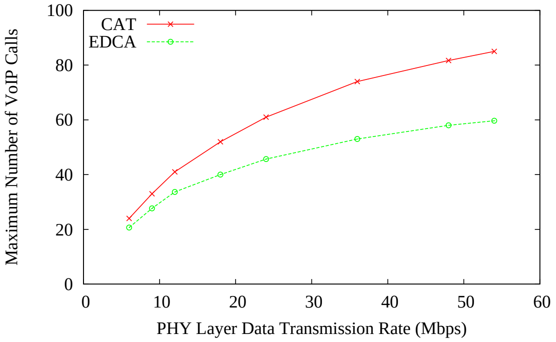

Fig. 4. Maximum number of VoIP calls for the two overlapping BSSes

VI. Related Work

Heck [2] studies the effect of BSS overlapping on the fairness of all the member stations and introduces a prioritization scheme to set different priority levels to stations based on their locations. Velayos et al. [12] address the load-balancing problem for OBSSes by utilizing the Ethernet backbone to broadcast the load level of APs. Stations in overloading BSS are forced to switch to under-loaded BSS. To address the OBSS problem for both contention period and contention free period in the legacy 802.11 WLAN, Fang et al. [13] propose a two-level carrier sensing mechanism. They introduce two new network allocation vectors (NAV), self-BSS NAV and OBSS NAV. Akella et al. [1] study the performance of end-clients in chaotic deployments of 802.11 WLANs. They design automated power control and rate adaptation algorithms to reduce interference among neighboring BSSes. Compared with these works, CAT is built on the EDCA mechanism of 802.11e and manages capacity of OBSSes by dynamically adjusting the channel access probability of each OBSS relative to the other OBSSes.

Another possible solution for OBSS management is to assign orthogonal channels to neighboring BSSes. Raniwala and Chiueh [14] study the channel assignment and routing problems in an 802.11-based multi-channel wireless mesh network. Mishra et al. [3] propose a client-driven approach for channel assignment and load balancing in overlapping 802.11 WLANs. Rozner et al. [15] propose a traffic-aware channel assignment by utilizing observed traffic demands at clients and APs. Kauffmann et al. [16] design a distributed channel assignment algorithm by utilizing Gibbs sampler, which only requires the involved stations to measure interference and transmission delay locally. By contrast, our work proposes a solution for OBSS management in situations where there are simply not enough orthogonal channels available.

VII. Conclusions

In this paper, we propose to use CAT to manage radio resources for OBSSes. CAT provides the AP of each BSS with a mechanism to control channel access parameters of its member stations on the fly. By coordinating the CAT operations of the OBSS APs, we can enable privileged channel access to an individual BSS at a particular time. We can also achieve a proportional partitioning of channel capacity among OBSSes. Our evaluation results, both in simulations and experiments using testbed built with commercial off-the-shelf hardware, show that CAT is able to not only proportionally allocate radio resources to OBSSes, but also improve the overall service capacity of the OBSSes. In the future, we plan to study the weak OBSS management problem and design more efficient scheduler based on the CAT framework.

VIII. Acknowledgement

We thank the anonymous reviewers for their insightful comments and Mathieu Lacage for support on the *yans* simulator. Bobby Bhattacharjee and Bo Han were supported in part by NSF award NeTS-NBD 0626636. Part of this work was done when Bo Han was a summer intern at AT&T Labs – Research.

References

[1] A. Akella, G. Judd, S. Seshan, and P. Steenkiste. Self-Management in Chaotic Wireless Deployments. In *Proceedings of ACM MOBICOM 2005*, pages 185–199, August-September 2005.

[2] K. Heck. Wireless LAN Performance in Overlapping Cells. In *Proceedings of VTC 2003-Fall*, pages 2895–2900, October 2003.

[3] A. Mishra, V. Brik, S. Banerjee, A. Srinivasan, and W. Arbaugh. A Client-Driven Approach for Channel Management in Wireless LANs. In *Proceedings of INFOCOM 2006*, pages 1–12, April 2006.

[4] IEEE Std 802.11e 2005. Local and metropolitan area networks–Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications–Amendment 8: Medium Access Control (MAC) Quality of Service Enhancements, November 2005.

[5] G. Bianchi. Performance Analysis of the IEEE 802.11 Distributed Coordination Function. *IEEE JSAC*, 18(3):535–547, March 2000.

[6] H. Zhu and I. Chlamtac. Performance Analysis for IEEE 802.11e EDCF Service Differentiation. *IEEE Transactions on Wireless Communications*, 4(4):1779–1788, July 2005.

[7] Multiband Atheros Driver for WiFi (MadWifi). Available at http://madwifi.org/.

[8] Iperf. Available at http://dast.nlanr.net/Projects/Iperf/.

[9] M. Lacage and T. R. Henderson. Yet Another Network Simulator. In *Proceedings of the 2006 ACM Workshop on ns-2: the IP Network Simulator*, October 2006.

[10] S. Shin and H. Schulzrinne. Experimental Measurement of the Capacity for VoIP Traffic in IEEE 802.11 WLANs. In *Proceedings of INFOCOM 2007*, pages 2018–2026, May 2007.

[11] ITU-T Recommendation G.107. The E-model, a Computational Model for Use in Transmission Planning, February 2003.

[12] H. Velayos, V. Aleo, and G. Karlsson. Load Balancing in Overlapping Wireless LAN Cells. In *Proceedings of ICC 2004*, pages 3833–3836, June 2004.

[13] Y. Fang, D. Gu, A. B. McDonald, and J. Zhang. A Two-level Carrier Sensing Mechanism for Overlapping BSS Problem in WLAN. In *Proceedings of LANMAN 2005*, pages 1–6, September 2005.

[14] A. Raniwala and T.-C. Chiueh. Architecture and Algorithms for an IEEE 802.11-Based Multi-Channel Wireless Mesh Network. In *Proceedings of INFOCOM 2005*, pages 2223–2234, March 2005.

[15] E. Rozner, Y. Mehta, A. Akel, and L. Qiu. Traffic-Aware Channel Assignment in Enterprise Wireless LANs. In *Proceedings of ICNP 2007*, pages 133–143, October 2007.

[16] B. Kauffmann, F. Baccelli, A. Chaintreau, V. Mhatre, K. Papagiannaki, and C. Diot. Measurement-Based Self Organization of Interfering 802.11 Wireless Access Networks. In *Proceedings of INFOCOM 2007*, pages 1451–1459, May 2007.