

Dynamic Link Adaptation in IEEE 802.11ac: A Distributed Learning Based Approach

Raja Karmakar

Department of IT

TICT Kolkata,

INDIA 700156

rkarmakar.tict@gmail.com

Samiran Chattopadhyay

Department of IT

Jadavpur University,

INDIA 700098

samiranc@it.jusl.ac.in

Sandip Chakraborty

Department of CSE

IIT Kharagpur,

INDIA 721302

sandipc@cse.iitkgp.ernet.in

Abstract—High throughput wireless access networks based on IEEE 802.11ac show a significant challenge in dynamically selecting the link configuration parameters based on channel conditions due to large pool of design set; like number of spatial streams, channel bonding, guard intervals, frame aggregation and different modulation and coding schemes. In this paper, we develop a learning based approach for link adaptation motivated by the multi-armed bandit based distributed learning algorithm. The proposed link adaptation algorithm, *BanditLink*, explores different possible configuration options based on observing their impact over the network performance at various channel conditions. We analyze the performance of *BanditLink* from simulation results, and observe that it performs significantly better compared to other competing mechanisms proposed in the literature.

Index Terms—IEEE 802.11ac; link adaptation; Multi-Armed Bandit

I. INTRODUCTION

With the inception of smart mobile and wireless devices, the demand for wireless capacity has been increasing in rapid pace. The growing developments of smart cities across the world have popularized the need for high speed wireless access networks or wireless local area networks (WLANs), popularly known as wireless hot-spots. The most commercially and commodity favored wireless devices are built over IEEE 802.11 technologies, and the growing demand for capacity and throughput has dragged the technologies towards its high throughput extensions – IEEE 802.11 such as IEEE 802.11n [2] and IEEE 802.11ac [3], [13], commonly known as *high throughput WLANs* (HT-WLANs). To meet this goal, a number of enhancements are introduced in physical (PHY) layer and Medium Access Control (MAC) sublayer. *Channel bonding*, *short guard interval (SGI)*, *multiple input multiple output (MIMO) antenna technologies* and *advanced modulation and coding schemes (MCS)* are used in the PHY layer, whereas *frame aggregation* and *block acknowledgement* technologies are introduced at the MAC layer for having higher data rates.

The most accepted physical layer technology introduced in HT-WLAN is the multi-antenna high data rate communication through MIMO technologies that can significantly boost up the throughput under interference, signal fading and multi-path long distance communications. Another interesting physical

amendment is the channel bonding, where multiple 20 MHz channels are combined together (40/80/160 MHz for IEEE 802.11ac) to increase the channel bandwidth that results significant increase of physical data rate. The SGI (400 ns instead of standard 800 ns) reduces the overhead of additional idle time between symbols when less number of wireless stations are contending for the channels. The modulation of signal along with coding rate is regulated by different levels of advanced MCS techniques supporting up to 6.77 Gbps in IEEE 802.11ac with eight-antenna access points, four 2-antenna stations, 160 MHz (MU-MIMO). In HT-WLANs, MAC sublayer is enhanced by *frame aggregation* and *block acknowledgement* (BA) with the purpose of reducing MAC overhead. Frame aggregation is classified into two categories: *Aggregated MAC Service Data Unit (A-MSDU)* and *Aggregated MAC Protocol Data Unit (A-MPDU)*.

In spite of these enhancements that boost up the physical data rate, all these features have their internal trade-offs which affect the overall network performance, particularly at the end-to-end or application level. The transmission range is reduced for higher channel width values when signal strength is weak. In [5], [6], it is shown that wider channels may increase packet loss and external interference due to inappropriate combination of channels to form wider channel. MIMO has the challenge of transmitter design and placing of multiple antennas which must solve multi-channel synchronization problem [16]. When traffic load and congestion increase in a network, the SGI decreases system throughput since it fails to eliminate the inter-symbol interference. Further, several factors like channel interference, signal fading and signal attenuation cause significant fluctuations in *signal to noise ratio* (SNR). A stable high SNR is required for the sustaining of higher order MCS values [14]. Similarly, the MAC enhancements also have their trade-offs. In a single transmission, frame aggregation sends multiple frames. Hence it may further increase packet loss when the signal level is low and channel error is high [9]. In BACK, multiple acknowledgements are carried in a single frame of acknowledgement. Thus the effectiveness of this mechanism is reduced when rate of packet loss is high in a network [11].

In this paper, we address the problem of *dynamic link adaptation*, where a wireless node automatically selects the list

of link configuration parameters, like number of MIMO spatial streams, channel bonding, SGI, MCS level, frame aggregation level etc., based on the channel condition and channel error probability. Although there exists such link adaptation algorithms for legacy IEEE 802.11 devices, popularly known as adaptive modulation and coding schemes or rate adaptation [17], link adaptation for HT-WLANs is severely difficult – due to the presence of a large number of feature set and their inter-dependencies. For example with 32 different MCS levels, 2 channel bonds (20 and 40 MHz), 2 guard intervals (400 ns and 800 ns), 2 frame aggregation levels (ON/OFF – the minimal configuration), IEEE 802.11n has 512 different configuration possibilities; and IEEE 802.11ac has even far more. Therefore it is extremely difficult to dynamically find out the link parameters based on the channel condition, that can yield a good performance at the application layer services.

In response to such unprecedented challenge, we explore the distributed automotive learning by the wireless devices from the past information such that they can adaptively select the best link parameters when a change in the channel condition is observed. At any time, the system will look at the past information to select a parameter set as well as explore the new set of possible parameters that have not been yet selected for data transmission. Thus two functions of *automotive learning* are followed in this approach: (a) *exploration*, that selects a parameter set that is not yet examined, and (b) *exploitation*, that chooses the best parameter set from the past experience (already explored parametric sets). This *exploration-exploitation* approach is in general very effective in an unpredictable environment where a system always tries to find out the optimal parameter set by acquiring knowledge about the environment. In this paper, we apply the concept of *learning approach* namely *multi-armed bandit* to design an efficient closed-loop link adaptation algorithm for HT-WLANs called *BanditLink*. We consider different link configuration parameters such as channel bonding, SGI, MCS and frame aggregation, to create the set of link parameters with SNR as the measurement of channel condition. We employ the success rate of frame transmission to measure the performance of the complete network. The performance of BanditLink is analyzed through simulation results, and it shows that this mechanism significantly improves the network performance compared to other related methodologies explored in the literature.

II. RELATED WORKS

There are very few works which have considered HT-WLANs to design link adaptation mechanism. MiRA [12] is such a mechanism that considers MIMO, but it performs excessive rate selection in poor channel condition. In Linux system, the default rate adaptation algorithm is Minstrel [1]. It is based on statistical information acquired through channel overhearing, but it is primarily applicable in legacy wireless devices and can not produce high throughput for HT-WLANs since several PHY/MAC enhanced features are not considered. Different MCS values and MIMO system are used in [18] to select appropriate data rate, but the other link parameters are

not considered. Frame aggregation is reported in designing some link adaptation such as [7]. All these mechanisms do not consider all the enhanced features of HT-WLANs and fail to produce high throughput. To solve this problem, Minstrel HT [8] perceives the maximum link parameters of IEEE 802.11n. At the same time, it suffers from random with exhaustive sampling. Several features of IEEE 802.11n have been considered for link adaptation in SampleLite [10]. It is *received signal strength indicator* (RSSI) threshold-based and thus may fail to cope up with all network scenarios.

III. BANDITLINK: SYSTEM MODEL AND DESIGN DETAILS

In this section, we'll describe briefly the standard multi-armed bandit mechanism followed by the details of how we exploit multi-armed bandit to design our system.

A. Multi-Armed Bandit

The multi-armed bandit (MAB) model designs an agent which always attempts to balance the exploration-exploitation trade-off in order to optimize the total value gained over the time period considered. In MAB problem, the agent repeatedly chooses a single option (called an *arm*) from a set of options whose characteristics are initially unknown, and receives a certain reward based on every choice. The agent wishes to maximize the acquired reward.

The K -armed bandit problem is defined by random variables $X_{i,n}$ for $1 \leq i \leq K$ and $n \geq 1$, where each i is the index of an arm of the MAB framework and n is the number of plays. Successive plays of arm i yield rewards $X_{i,1}, X_{i,2}, \dots$ which are independent and identically distributed according to an unknown distribution with unknown expectation μ_i . We denote the expected values of the reward distributions by $\mu = \{\mu_1, \mu_2, \mu_3, \dots, \mu_K\}$. Let $T_i(n)$ be the number of times the arm i has been played by the algorithm during the first n plays. For arm i , the regret is defined by $(\mu^* - \mu_i)$ where $\mu^* = \max_{1 \leq i \leq K} \mu_i$. After selecting an arm, let $r(t)$ be the reward obtained at time t . Therefore, after n plays, the expected value of regret is defined by $E[R(n)] = \mu^*n - \sum_{t=1}^n E[r(t)]$. Now, $\sum_{t=1}^n E[r(t)] = \sum_{i=1}^K \mu_i E[T_i(n)]$. Therefore,

$$E[R(n)] = \mu^*n - \sum_{i=1}^K \mu_i E[T_i(n)] \quad (1)$$

B. ϵ -greedy Algorithm

It is a simple and well-known policy for multi-armed bandit problem [15]. This policy introduces a parameter ϵ called exploration probability. At time t , ϵ is determined by the following equation:

$$\epsilon_t = \min(1, cK/d^2t) \quad (2)$$

K is the number of arms in bandit problem and c is a parameter and $c > 0$. The parameter d is defined as,

$$0 < d \leq \min(\mu^* - \mu_i) [\forall i : \mu_i < \mu^*] \quad (3)$$

This means d is the difference between the expected rewards of the best arm and the second best arm. This policy describes to

play the arm with the largest average reward with probability $(1 - \epsilon)$ and play a randomly chosen arm with probability ϵ . Therefore, $\text{Strategy} = \epsilon \times \text{Explore} + (1 - \epsilon) \times \text{Exploit}$. In [4], it is shown that after any number $n \geq cK/d^2$ of plays, the probability of choosing a suboptimal arm is bounded by $O(c/d^2n)$.

C. Exploiting MAB in Dynamic Link Adaptation of HT-WLANs: BanditLink

BanditLink is based on stochastic multi-armed bandit problem using ϵ -greedy approach. We consider channel bandwidth (cb), guard interval (g), level of frame aggregation (f) and modulation and coding scheme (m) to adjust the data rate of a wireless station. It can be described by a tuple $T < cb, g, f, m >$. cb takes the channel bonding levels, ranging from the minimum bonding bandwidth (minBandwidth) to maximum bonding bandwidth (maxBandwidth). g is either 400ns or 800ns and we consider n_f values of f in this model. In $n_f = 1$, we consider the maximum number of aggregated frames denoted by maxMpdu which is decremented by the amount of dcr in successive higher values of n_f . Thus $\text{minMpdu} = \text{maxMpdu} - (n_f - 1) \times dcr$. m varies from minimum MCS (minMcs) to maximum MCS (maxMcs).

The set of all possible values of T forms a configuration set $\mathcal{C}$. Each value of $\mathcal{C}$ is denoted by k , and thus k is represented by the tuple T . k is considered as an arm of a stochastic MAB problem. Let there are total K number of configurations (arms) i.e. $|\mathcal{C}| = K$. We develop an agent ($\mathcal{A}$) for the transmitter wireless station that follows BanditLink to select the best possible configuration at any time t and calculates the *reward* as well as the *regret* applying the configurations:

- **Reward:** The proportion of successfully transmitted frames is considered as reward and we call it by *frame success ratio* (FSR). Therefore;

$$\text{FSR} = \frac{\text{Number of frames successfully received}}{\text{Number of frames transmitted}}$$

Here the transmitter station assumes that the number of frames acknowledged is the number of frames successfully received by the receiver.

- **Regret:** The configuration producing the maximum FSR is known as optimal configuration. The regret is the difference between the FSRs of the optimal configuration and the selected configuration at any time slot.

D. Metric Selection

At a certain time t , the agent $\mathcal{A}$ selects a configuration $k \in \mathcal{C}$, and calculates FSR (r). Here the MAC layer parameter (FSR) is used to measure the system performance for a given link parameter set. It also calculates the channel's SNR (snr) which can be a good measurement of signal quality. Hence we design a set $R = \{\text{snr}, k, r\}$ called *reward matrix* which creates the metric set for the link adaptation.

E. System Architecture

The system architecture of BanditLink is shown in a block diagram in Fig. 1. The matrix R , set $\mathcal{C}$ and *Epsilon Calculator* create the backbone of the mechanism. At each time of link parameter selection, first the channel's SNR (snr) is computed by the interface hardware abstraction layer. In the *Initial Phase*, some predefined configurations are used from $\mathcal{C}$ to transmit data. After a given period of time, *Reward* (r) is calculated from the receiver's acknowledgement (*ACK*) and R is updated by the selected configuration, snr and r .

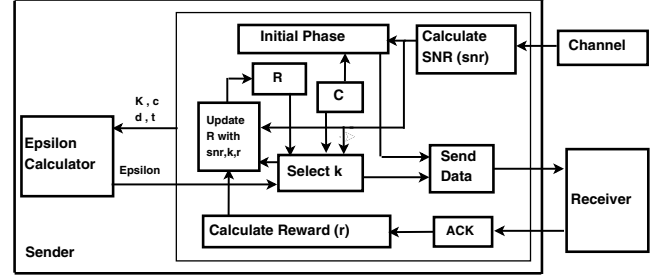


Fig. 1. Block diagram of the proposed mechanism

On the completion of the *Initial Phase*, *Epsilon Calculator* is used for data transmission. Before transmission begins, ϵ is calculated using *Epsilon Calculator* from K, c, d and t following equation 2. Then a configuration (k) is selected from R or $\mathcal{C}$ by applying ϵ -greedy approach and data is transmitted using the link configuration k . Periodically, r is calculated from *ACK* and R is updated accordingly. In this way, consideration of receiver's *ACK* makes BanditLink as a closed-loop mechanism.

F. Framework

The three major modules of MAB framework can be related to BanditLink framework by the following way:

- 1) **Set of arms:** It is the set of configurations ($\mathcal{C}$) – each of them is represented by the tuple $T < cb, g, f, m >$.
- 2) **Action to select an arm:** It is the selection of a particular configuration $k \in \mathcal{C}$ and thus selection of a data rate.
- 3) **Reward after playing the arm:** It is the FSR that is calculated after transmission of data for a given period of time.

The agent $\mathcal{A}$ chooses the next configuration to be used for data transmission based on the sequence of selected configurations in the past along with the obtained FSRs, and this selection is represented by the random variable $X_{i,n}$ for $1 \leq i \leq K$ and $n \geq 1$, where each i is the index of the selected configuration and n is the number of time slots. Successive applications of configuration i yield FSRs $X_{i,1}, X_{i,2}, \dots$ which are independent and identically distributed according to an unknown distribution. For configuration i , we denote the expected value (mean) of this distribution of FSRs by μ_i . Hence the expected values of the FSR distributions are

defined by $\mu = \{\mu_1, \mu_2, \mu_3, \dots, \mu_K\}$. The parameters used by this framework are discussed below:

- 1) $T_i(n)$ is used to represent the number of times the configuration i has been selected during the first n time slots.
- 2) The maximum expected FSR is denoted by μ^* . Thus the expected regret with respect to FSR for the configuration i is defined by $(\mu^* - \mu_i)$.
- 3) $R(n)$ is the value of regret after n time slots.
- 4) At any time slot, the agent $\mathcal{A}$ applies ϵ -greedy policy to select the data rate. The exploration probability is defined by ϵ according to equation (2).
- 5) The parameter d is used to measure the difference between the maximum expected FSR and the second maximum expected FSR at any time slot following equation (3).
- 6) A positive constant c is engaged to calculate the value of ϵ . A small positive constant δ is used to define a range of SNRs in R during exploitation.

G. BanditLink: Description of Algorithm

BanditLink is described by algorithm 1 which is based on the model discussed in the Section III. The algorithm employs the agent $\mathcal{A}$ to select the data rate. There are two phases in this algorithm:

Initial (Cold-start) phase: This is the beginning stage where the algorithm explores some of the configurations from $\mathcal{C}$. The agent $\mathcal{A}$ measures SNR of the channel and chooses the configurations corresponding to the minimum and maximum MCS of every set $\langle cb, g_{min}, f_{max}, m \rangle$ from $\mathcal{C}$. It starts data transmission using those configurations. After a given period of time (t_{dur}), it observes FSR as well as updates R . Hence it selects the minimum and maximum data rates considering all the available channel bandwidths with short guard interval (g_{min}) and the highest used frame aggregation level (f_{max}). In this way it can be able to gather information about the wireless environment for the worst and the best data rates. It can be noted that t_{dur} should be selected by the network administrator such that the channel condition (namely, the SNR) remains almost constant within that duration. Therefore, for a static environment, t_{dur} can be large enough of a single IEEE 802.11 beacon interval (100 ms). Further, let t_{init} be the initial number of rounds for populating R . Then considering 4 different channel bonding levels, t_{init} can be set to 16 (4 bonding levels $\times$ 2 guard intervals $\times$ 2 data rates).

Experience phase: At this phase, the agent $\mathcal{A}$ applies exploration and exploitation schemes based on the value of ϵ . At time t , ϵ_t is calculated from equation (2). Then exploration and exploitation are performed with probability ϵ_t and $(1 - \epsilon_t)$ respectively. At the beginning of a time slot t , the ϵ_t is calculated and SNR of channel is measured. Let this SNR is denoted by x . To exploit the experience, the agent searches for a SNR within the range $(x - \delta)$ and $(x + \delta)$ in R for $\delta > 0$. If this search is successful, then the configuration which has produced the highest FSR for this range is selected with probability $(1 - \epsilon_t)$. Otherwise the configuration providing

Algorithm 1 BanditLink – Algorithmic Description

- 1: **Input:** Parameters c , d and δ , configuration set $\mathcal{C}$, reward matrix R .
 - 2: **Initialization:** Let t_{init} be the number of rounds required for building up initial reward matrix (cold-start). For $t = 1, 2, 3, \dots, t_{init}$, calculate SNR of channel and choose the configurations corresponding to the minimum and maximum MCS of every $\langle cb, g_{min}, f_{max}, m \rangle$ from $\mathcal{C}$ and perform data transmission for a given period of time (t_{dur}). Observe rewards and update R .
 - 3: **while** $t > t_{init}$ **do**
 - 4: At time slot t , calculate ϵ_t by $\epsilon_t = \min(1, cK/d^2t)$.
 - 5: Observe SNR of the channel. Let $x \leftarrow$ Observed SNR.
 - 6: Let $\nu \leftarrow \text{Random}(0,1)$.
 - 7: **if** $\nu \leq \epsilon_t$ **then**
 - 8: **if** $\text{SNR} \in [(x - \delta), (x + \delta)]$ **then**
 - 9: Select $R_S \subset R$, such that R_S contains all the entries from R where the $\text{SNR} \in [(x - \delta), (x + \delta)]$.
 - 10: Choose configuration $k \in \mathcal{C}$ from R_S , that produces the highest reward in R_S .
 - 11: **else**
 - 12: Choose configuration $k \in \mathcal{C}$ from R that produces the highest reward in R .
 - 13: **end if**
 - 14: **else**
 - 15: Select a configuration k uniformly at random in $\mathcal{C}$.
 - 16: **end if**
 - 17: Initiate data transmission with the selected configuration k . After t_{dur} , calculate reward r .
 - 18: Update R with x , k and r .
 - 19: **end while**
-

the highest FSR in R is chosen with probability $(1 - \epsilon_t)$. This phenomenon of exploitation first tries to apply the best configuration whose SNR is close to the current measured SNR such that the selected configuration can still manage to provide the best FSR and thus the best performance. To explore the environment, the agent takes a configuration uniformly selected at random in $\mathcal{C}$ with probability ϵ_t . After selecting a configuration and thus a data rate, data transmission is initiated and R is updated after t_{dur} .

From equation (2) and the values of probabilities of exploration and exploitation, we can observe that the rate of selecting the unexplored configurations is high at the initial stage of *experience phase*, but as time progresses the rate of exploitation i.e., choosing the best configuration increases.

H. Bound Analysis

We analyze the bound over the regret value of BanditLink in terms of time. If $T_i(n)$ be the number of times the configuration i has been selected by the algorithm during the first n slots and the exploration probability is ϵ_t at time t , then for $K > 0$,

$$E[T_i(n)] = \frac{1}{K} \sum_{t=1}^n \epsilon_t \quad (4)$$

Let $p = cK/d^2$. For $t \leq p$, let $E[T_i(n)]$ is upper bounded by p . Therefore, from equation (4), we get,

$$E[T_i(n)] \leq p + \frac{1}{K} \sum_{t=p+1}^n \epsilon_t \leq p + \frac{1}{K} \int_{p+1}^n \epsilon_t dt \quad (5)$$

After time n , exploration is upper bounded by $c/d^2 n$ as shown in [4]. Hence equation (5) can be written as,

$$E[T_i(n)] \leq p + \frac{1}{K} \int_{p+1}^n \frac{c}{d^2 t} dt \leq p + \frac{c}{K d^2} \log\left(\frac{n-1}{p}\right) \quad (6)$$

From equation (1), the expected regret after time n is given by,

$$E[R(n)] = \mu^* n - \sum_{i=1}^K \mu_i E[T_i(n)]$$

Now from equation (1) and equation (6), the expected regret is,

$$E[R(n)] = \mu^* n - \sum_{i=1}^K \mu_i \left[p + \frac{c}{K d^2} \log\left(\frac{n-1}{p}\right) \right] \quad (7)$$

From equation (7), it can be noted that after time n , the expected regret is bounded by

$$\mu^* n - \sum_{i=1}^K \mu_i \left[p + \frac{c}{K d^2} \log\left(\frac{n-1}{p}\right) \right]$$

Hence this regret is bounded by $O(\log n)$ on time n .

IV. PERFORMANCE ANALYSIS VIA SIMULATION

BanditLink has been implemented in network simulator (NS) version NS-3.24.1 and the performance is analyzed through infrastructure IEEE 802.11ac WLAN with multiple wireless STAs contending for channel access and transmitting data among themselves. The A-MPDU type of frame aggregation has been used. The STAs use different bonding levels (20 MHz, 40 MHz, 80 MHz and 160 MHz), 400 and 800 ns guard intervals and 8 different A-MPDU levels where length of each MPDU is 2 KB. The STAs generate data traffic using TCP. The STAs use 5 GHz frequency band for communication with log-normal path loss model (path loss exponent = 0.3) and constant speed propagation delay model. The value of ϵ is calculated dynamically by the equation (2). At any instant of time, d is also determined dynamically by the difference between the expected rewards of the best configuration and the second best configuration. The values of c and δ are set to 1 and 5 respectively.

We evaluate the system performance in terms of MAC throughput, packet loss ratio (PLR) and packet delay under two cases - (i) for different SNR values of channel with one STA and (ii) for different number of STAs with good signal strength ($SNR = 45$ dB). The performance is also examined under dynamic channel conditions: signal strength from good to bad and from bad to good transitions. We evaluate the efficiency of BanditLink with respect to SampleLite [10], Minstrel HT [8] and Minstrel [1].

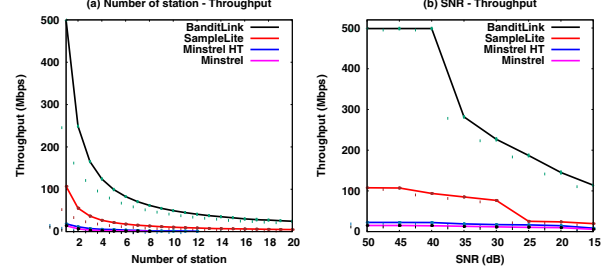


Fig. 2. Throughput comparison

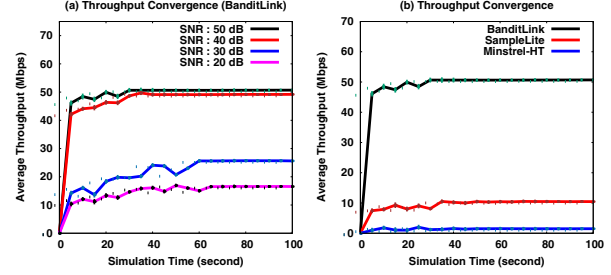


Fig. 3. Throughput Convergence

A. Analysis of Throughput

Fig. 2 compares the channel access throughput among the competing link adaptation mechanisms. With the help of exploration approach, BanditLink always tries to gather information about the wireless environment. At the same time, it tries to apply the best link configuration set for a given channel condition using exploitation. These two approaches make the algorithm highly dynamic in nature by which the system is able to select the best possible data rate after gathering some information about the environment regarding the link parameter set. Application of ϵ -greedy method helps BanditLink to increase the rate of exploitation as time goes by and thus the selection of the best possible configuration increases. Hence in any condition (decrease of SNR value or increase of STA), the system is able to cope up with the situation with the help of experience gathered and it performs better than the other three algorithms in terms of throughput as shown in figure 2.

Protocol convergence: To analyze the protocol convergence behavior, we have observed the average throughput with respect to the simulation time, as shown in Fig. 3. This test is carried out to find the convergence region of the achieved average throughput of BanditLink. Here the number of STA is set to 10. Fig. 3(a) shows the convergence of BanditLink for 4 different values of SNRs. From this graph, it can be observed that initially BanditLink is showing a fluctuating nature in throughput because of higher rate of exploration in early phases of this algorithm. As simulation time increases, rate of exploitation also increases, which always applies the best configuration and thus it is able to produce a stable high average throughput after some time. With the lower values of

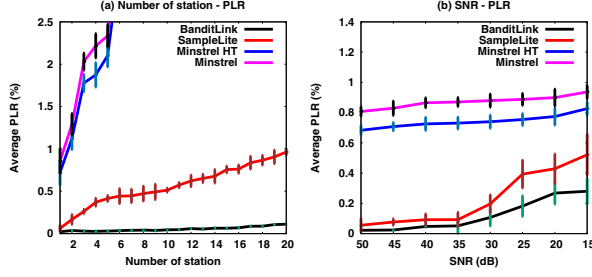


Fig. 4. Performance in terms of packet loss ratio

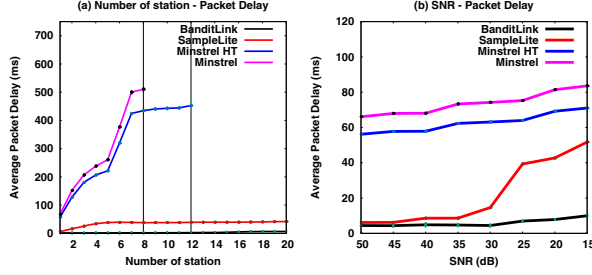


Fig. 5. Performance in terms of packet delay

SNRs, the convergence time of BanditLink increases as it tries to adjust with low signal quality. In Fig. 3(a), the stable state is reached after 70 seconds considering all 4 SNRs.

In Fig. 3(b), the throughput convergence of BanditLink is compared with the other three mechanisms with 10 number of STAs. Here we can observe that the exploitation mechanism helps to reach the steady state in BanditLink. In this graph, SampleLite reaches the steady state approximately at same time as BanditLink, but its throughput is always very less than BanditLink. Minstrel and Minstrel HT are producing very lower throughput than our algorithm in all the cases.

B. Analysis of PLR and Packet Delay

Fig. 4 shows the performance evaluation in terms of PLR. Enhancement of throughput of BanditLink reduces number of packet losses and thus reduces PLR. Packet loss increases due to low signal energy (low snr) or packet collisions (with the increase of wireless stations). In both the cases, the system can still adjust with the network condition by applying the past experience to produce lower PLR than SampleLite, Minstrel HT and Minstrel.

Channel condition deteriorates when signal strength decreases or traffic load increases (higher number of STAs) and thus leads to increase packet transmission time. BanditLink has better adjustment with this situation with the help of past experience (configurations explored in bad network conditions act as references for the same kind of situations in future) which enables the system to produce the lowest packet delay as much as possible.

Fig. 5 and 6(a) depict that BanditLink also provides better performance than SampleLite, Minstrel HT and Minstrel in terms of packet delay.

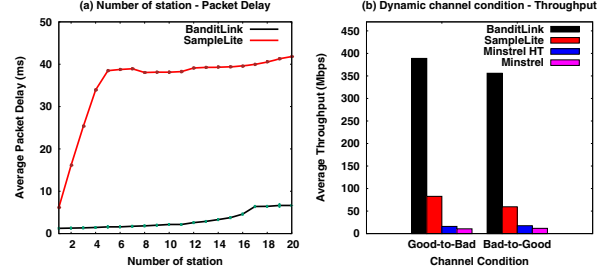


Fig. 6. (a) Performance in terms of packet delay with respect to number of stations excluding Minstrel and Minstrel HT; (b) Performance in dynamic channel conditions where signal strength varies from good to bad and bad to good

C. Analysis of Performance in Dynamic Channel Condition

Fig. 6(b) shows the result where we dynamically change the signal strength of channel from good to bad and vice versa ($15 \text{ dB} \leq \text{SNR} \leq 50 \text{ dB}$). It can be observed that BanditLink is better to adjust with the dynamic channel condition and results in producing better average throughput than SampleLite, Minstrel HT and Minstrel because it not only applies the best possible configuration yet experienced so far, but it also uses some unexplored configurations to gain information about the link parameter set for the present channel condition. This phenomenon makes BanditLink highly dynamic to select the best possible data rate.

D. Analysis of Congestion Window, Slow Start Threshold and Retransmission Timeout

In this section, we analyze different TCP parameters to check how TCP throughput stabilizes in a higher value for BanditLink. The TCP congestion window controls sender side flow of data transmission that is based on network capacity and conditions. The size of congestion window is controlled with the association of *slow start threshold (SST)* and *retransmission timeout (RTO)*. In the following analysis, when number of STA is not mentioned and time is taken in X-axis, the number of STA is 5.

Congestion window: The change of congestion window with respect to time is shown in Fig. 7(a) (Minstrel is not considered in this figure since it has the lowest throughput and highest PLR and delay than the other mechanisms). Applying exploration and exploitation at each instant of data rate selection, BanditLink can be able to set link configuration parameters with their optimal values in maximum cases. It helps to reduce the number of packet loss, which in turn maintains a steady development of congestion window. Rate of packet loss of Minstrel-HT is higher than SampleLite and BanditLink, and thus Minstrel-HT is providing fluctuating and the lowest value of congestion window than that of other two. The maximum value of the congestion window of BanditLink is higher since it produces lower PLR value.

Fig. 7(b) demonstrates the effect of number of wireless stations over congestion window by setting high value of SNR ($\text{SNR} = 45 \text{ dB}$). As the number of STA increases, the number

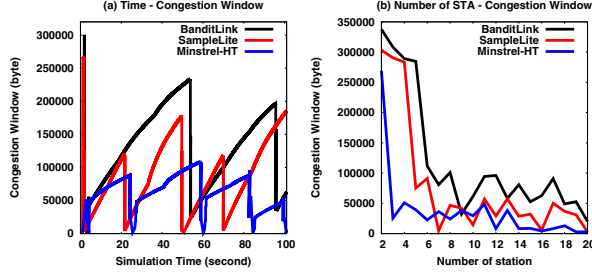


Fig. 7. Performance in terms of congestion window

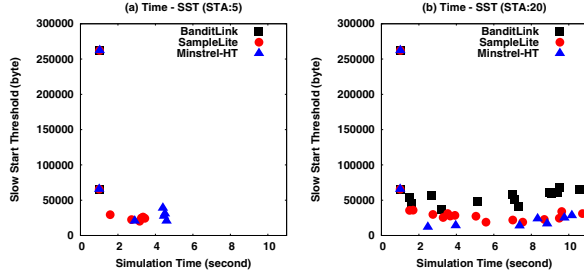


Fig. 8. Performance in terms of slow start threshold (SST)

of wireless link also increases. For example, one AP and n number of STAs will create n links where AP is common. Thus the average size of congestion window decreases as n is increased (we consider AP as sender and all STAs are receivers). In BanditLink, sender always tries to transmit data in the highest possible data rate as much as possible considering channel's signal strength. An appropriate combinations of cb , g , f and m lead to produce the optimal data rate considering the network condition. For a given SNR, Searching of such combinations is the key part of BanditLink. This type of adaptive learning makes it different than SampleLite, Minstrel-HT and Minstrel. Transmission with higher data rate as well as lower PLR, helps to maintain a higher value of congestion window in BanditLink than the other mechanisms (here we have not considered Minstrel since its average throughput is the lowest than the others).

Slow start threshold: The congestion avoidance phase runs until the acknowledgements arrive before their timeouts. Slow start threshold provides a *threshold* value of congestion window up to which the congestion window grows exponentially. After reaching the threshold, if still there is no packet loss in the network, the congestion window grows linearly, otherwise the threshold is set to one half of the present congestion window and the window size is set to its initial starting value. Thus if packet loss and delay increase then the rate of change of slow start threshold also increases. In BanditLink, PLR and packet delay are very low compared to the others as illustrated in Fig. 4 and Fig. 5. This phenomenon helps to reduce the rate of change of slow start threshold in BanditLink. Hence from Fig. 8(a) and Fig. 8(b), it can be observed that the number of SST is less in BanditLink than that of the others as well

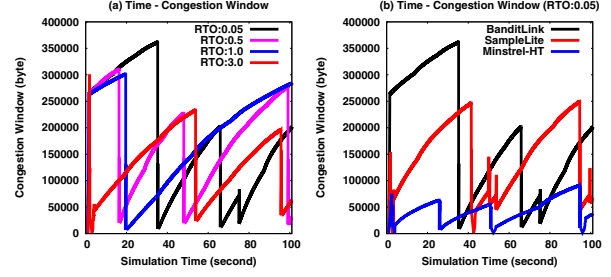


Fig. 9. Performance in terms of (a) different RTOs and (b) RTO:0.05

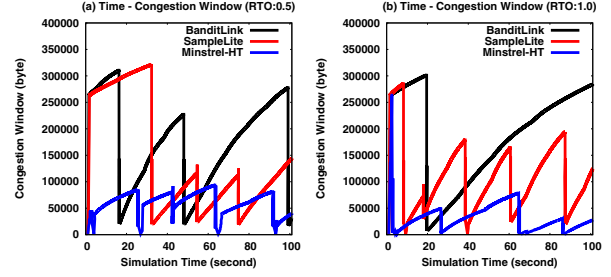


Fig. 10. Performance in terms of (a) RTO:0.5 and (b) RTO:1.0

as the average value of SST is higher in BanditLink. As the number of STA increases, PLR and delay also get high. In this scenario, the PLR and delay of BanditLink are still very low compared to the other mechanisms and thus its rate of change of SST is also lower as shown in Fig. 8(b).

Retransmission timeout: Retransmission timeout can help to identify packet loss in network and locate congestion links. When a packet is sent, the RTO is set for this packet. If sender does not receive acknowledgement of this packet within its RTO then the packet is retransmitted. We have carried out some simulations where we have considered different RTOs (we express RTO in second) to find the impact of it over congestion window with respect to simulation time (Fig. 9 and Fig. 10). When packet delay increases, the number of retransmission increases due to unarrival of packets within their corresponding RTOs. Packet loss also leads to unsuccessful transmission and results in unavailability of acknowledgements with their RTOs. Now it is already explained through Fig. 4 and Fig.5 that BanditLink provides very low PLR and packet delay compared to the other mechanisms. It is achieved through the tendency of setting the optimal value of tuple T for given SNR and it is gained by BanditLink by its automotive learning about the wireless environment.

Fig. 9(a) shows that initially the reaching of the maximum value of congestion window is delayed with the lower values of RTOs. Higher RTO value helps to transmit more data before the timeout. With the help of this concept, the system reaches the maximum possible value of its congestion window earlier when the packet loss is very low as in BanditLink. In Fig. 9(b) and Fig. 10, we can observe that as RTO increases, BanditLink produces much better congestion window size since the rate

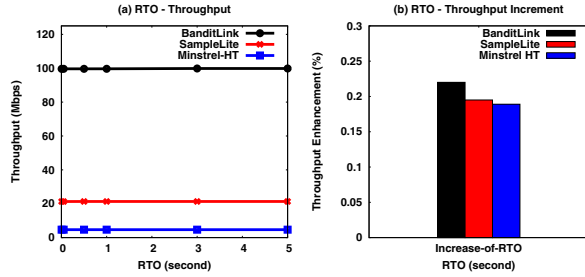


Fig. 11. Performance in terms of (a) Throughput and (b) Throughput enhancement with increase of RTO

of exploitation increases as time progresses.

Retransmission timeout and average throughput: We also evaluate the effect of RTO over achieved average throughput. As RTO increases, the chances of reception of acknowledgements also increase for the delivered packets and thus result in improving the average throughput of the network. This improvement of throughput is better in BanditLink due to the rate of exploitation increases. It always tries to tune its parameter set (T) into some optimal value depending on the current network condition. The tuning is improved (probability of setting the optimal value of T) as time progresses i.e. rate of exploitation increases, which in turn enhances the throughput. Although the average throughput is enhanced as RTO increases, the rate of change of throughput is very small in this regard as shown in Fig. 11(a). From Fig. 11(b), the rates of increases of average throughputs of BanditLink, SampleLite and Minstrel-HT are 0.22%, 0.195% and 0.189% respectively with the increase of RTO. Hence the rate of increase of average throughput is higher in BanditLink since it employs automotive learning methodology.

V. CONCLUSION AND FUTURE DIRECTION

This paper proposes BanditLink, a dynamic link adaptation framework for high throughput wireless access networks, from the inspiration of distributed learning based approaches to adaptively learn and select the best configurations depending on the environmental condition. BanditLink uses the exploration and exploitation features of multi-armed bandit algorithm to learn the effect of link configuration parameters over network performance, and based on this, it selects and applies the link parameters when a data transfer is initiated. The simulation analysis show that BanditLink performs quite better than SampleLite – a recently published work on adaptive link configuration; whereas significantly boosts up the performance compared to Minstrel and Minstrel HT, the default link adaptation mechanism for Linux system. To the best of our knowledge, this is the first paper in this direction where learning based approach is used to adaptively configure link parameters for high throughput wireless access networks. However, the impact of different control parameters for the learning needs to be explored, which we keep as the future direction of this work.

VI. ACKNOWLEDGEMENT

This work is supported by *Innovative Research and Development Program (ISIRD)*, funded by Sponsored Research and Industrial Consultancy, IIT Kharagpur.

REFERENCES

- [1] "Madwifi: Multiband Atheros Driver for WiFi," <http://sourceforge.net/projects/madwifi/>.
- [2] "IEEE standard for information technology– local and metropolitan area networks– specific requirements– part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications amendment 5: Enhancements for higher throughput," *IEEE Std 802.11n-2009 (Amendment to IEEE Std 802.11-2007 as amended by IEEE Std 802.11k-2008, IEEE Std 802.11r-2008, IEEE Std 802.11y-2008, and IEEE Std 802.11w-2009)*, pp. 1–565, 2009.
- [3] "IEEE standard for information technology– local and metropolitan area networks– specific requirements– part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications amendment 4: Enhancements for very high throughput for operation in bands below 6 ghz," *802.11ac-2013 - IEEE Standard for Information technology– Telecommunications and information exchange*, pp. 1–425, 2013.
- [4] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time Analysis of the Multiarmed Bandit Problem," *Machine Learning*, vol. 47, no. 2, pp. 235–256, May 2002.
- [5] L. Deek, E. Garcia-Villegas, E. Belding, S.-J. Lee, and K. Almeroth, "Joint Rate and Channel Width Adaptation for 802.11 MIMO Wireless Networks," in *Proceedings of the 10th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, June 2013, pp. 167 – 175.
- [6] —, "Intelligent Channel Bonding in 802.11n WLANs," *IEEE TRANSACTIONS ON MOBILE COMPUTING*, vol. 13, no. 6, pp. 1242 – 1255, June 2014.
- [7] K.-T. Feng, P.-T. Lin, and W.-J. Liu, "Frame-aggregated link adaptation protocol for next generation wireless local area networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2010, no. 10, April 2010.
- [8] F. Fietkau, "Minstrel HT: New Rate Control Module for 802.11n," <http://lwn.net/Articles/376765/>, March 2010.
- [9] K. Hassine and M. Frikha, "MAC aggregation in 802.11n: Concepts and impact on wireless networks performance," in *Proceedings of the 2014 International Symposium on Networks, Computers and Communications*, 2014, pp. 1–6.
- [10] L. Kriara and M. K. Marina, "Samplelite: A Hybrid Approach to 802.11n Link Adaptation," *ACM SIGCOMM Computer Communication Review*, vol. 45, no. 2, pp. 4–13, April 2015.
- [11] W.-J. Liu, C.-H. Huang, K.-T. Feng, and P.-H. Tseng, "Performance analysis of greedy fast-shift block acknowledgement for high-throughput w lans," *Wireless Networks*, vol. 20, no. 8, pp. 2503–2519, 2014.
- [12] I. Pefkianakis, Y. Hu, S. H. Wong, H. Yang, and S. Lu, "MIMO rate adaptation in 802.11n wireless networks," in *Proceedings of the sixteenth annual international conference on Mobile computing and networking*, September 2010, pp. 257–268.
- [13] E. Perahia and M. X. Gong, "Gigabit wireless LANs: an overview of IEEE 802.11ac and 802.11ad," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 15, no. 3, pp. 23–33, 2011.
- [14] M. Taki, M. Rezaee, and M. Guillaud, "Adaptive modulation and coding for interference alignment with imperfect CSIT," *IEEE Transactions on Wireless Communications*, vol. 13, no. 9, pp. 5264–5273, 2014.
- [15] C. Watkins, "Learning from Delayed Rewards. PhD thesis, University of Cambridge, Cambridge, England," May 1989.
- [16] S. Wu, W. Mao, and X. Wang, "Performance study on a CSMA/CA-based MAC protocol for multi-user MIMO wireless LANs," *IEEE Transactions on Wireless Communications*, vol. 13, no. 6, pp. 3153–3166, 2014.
- [17] D. Xia, J. Hart, and Q. Fu, "Evaluation of the minstrel rate adaptation algorithm in IEEE 802.11g WLANs," in *Proceedings of the IEEE International Conference on Communications*, 2013, pp. 2223–2228.
- [18] Q. Xia, M. Hamdi, and K. Ben Letaief, "Open-Loop Link Adaptation for Next-Generation IEEE 802.11n Wireless Networks," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 7, pp. 3713 – 3725, January 2009.