國 立 交 通 大 學

電信工程研究所

碩士論文

IEEE 802.11 多重速率無線網路之整合型參數調整策略

ARC: Joint Adaptation of Link Rate and Contention Window for IEEE 802.11 Multi-rate Wireless Networks

研究生：李安智

指導教授：林亭佑 博士

中華民國九十七年八月

IEEE 802.11 多重速率無線網路之整合型參數調整策略
ARC: Joint Adaptation of Link Rate and Contention Window for IEEE 802.11 Multi-rate Wireless Networks

研 究 生：李安智　　Student：An-Chih Li

指導教授：林亭佑　　Advisor：Ting-Yu Lin

國 立 交 通 大 學
電信工程學系
碩 士 論 文

A Thesis
Submitted to Department of Communication Engineering
College of Electrical Engineering and Computer Science
National Chiao Tung University
in partial Fulfillment of the Requirements
for the Degree of
Master
in

Communication Engineering

Auguest 2008

Hsinchu, Taiwan

中華民國九十七年八月

IEEE 802.11 多重速率無線網路之整合型參數調整策略

學生：李安智　　　　　　　　　　　　　　　　　　　　指導教授： 林亭佑 博士

國立交通大學電信工程學系（研究所）碩士班

摘要

IEEE 802.11 在實體層提供多重速率傳輸的能力與機制。不同傳輸速率有各自訊號與雜訊干擾值的需求。若在固定雜訊能量的情況下，節點對於傳輸速率的調整會與傳輸周圍附近的節點干擾有關。而 802.11 DCF亦提供指數退回演算機制 (binary exponential backoff)來降低節點使用通道的機會，解決通道壅塞的問題。傳統的傳輸速率調整機制為，當節點傳輸封包發生失敗時，會同時調降傳輸速率且讓目前的壅塞視窗 (cw_p)呈指數成長，但這樣的調整會讓節點的傳送過於保守，降低通道使用效能；然而，當節點傳送封包成功時，節點會調升傳送速率且重設目前的壅塞視窗，但這樣的調整卻會讓節點的傳送過於積極，造成不必要的碰撞，影響系統的吞吐量。這樣的問題，主要是由於 802.11 沒有將傳輸速率與壅塞視窗，這兩參數的調整一起考慮。

本篇論文由於觀察到上述的現象，所以我們提出了新的速率調整機制名為 ARC (Joint Adaptation of Link Rate and Contention Window)。藉由預測目前通道最佳的壅塞視窗的大小 (optCW)，若$cw_p >$ optCW ($cw_p <$ optCW)，在節點傳輸成功時 (失敗)，ARC會調降 (增大)目前壅塞視窗的值，但仍維持相同的傳送速率 R；否則，才會去調升 (調降)目前的傳送速率R值得大小。ARC的特點就是，它讓節點可顧及傳送速率的穩定，避免不必要的傳輸速率變動的問題。模擬結果顯現，ARC表現比傳統速率調整機制來的出色。最後我們亦提出馬可夫鏈的數學模型來驗證ARC模擬結果。

ARC: Joint Adaptation of Link Rate and Contention Window for IEEE 802.11 Multi-rate Wireless Networks

Student： An-Chih Li　　　　　　　　Advisors：Dr. Ting-Yu Lin

Department（Institute）of Communication Engineer

National Chiao Tung University

Abstract

IEEE 802.11 wireless network supports multiple link rates at the physical layer. Each link rate is associated with a certain required Signal-to-Interference-and-Noise Ratio (SINR) threshold for successfully decoding received packets. Suppose constant noise and no power adjustment exists, apparently SINR is solely affected by the accumulated interference power level I. The method of selecting an appropriate link rate for transmitting/retransmitting packets is generally known as the link adaptation mechanism. Traditional link adaptation approaches try to reduce the transmit rate (hence lower SINR threshold is required) on transmission failures (potentially due to the increased denominator I of SINR), whereas upgrade the transmit rate (hence higher SINR threshold is required) on successful transmissions (potentially due to the decreased denominator I of SINR). The accumulated interference power level I in some sense indicates the medium congestion status. In 802.11, on transmission failures, the DCF performs a binary exponential backoff mechanism to discourage channel access attempts. When traditional link adaptation is applied, both rate reduction and binary backoff represent double penalties for this wireless link, which may cause overly conservative transmission attempts. On the other hand, once transmission succeeds, 802.11 DCF resets the backoff contention window to the minimum value to encourage channel access attempts. At the same time, traditional link adaptation may also decide to increase the data rate, which may lead to overly aggressive transmission attempts. We observe this improper interaction of link rate and backoff mechanism that harms the 802.11 system performance, due to separate consideration of those two parameters.

In this paper, rather than independently dealing with the two parameters, we propose to perform link adaptations by firstly considering if a proper backoff window has been reached. Specifically, if the medium congestion level I can be reduced by imposing a larger backoff window on transmissions, then there may be no need to decrease the link rate, given SINR can be sustained. Conversely, if there is extra interference that may be tolerated in I, a smaller backoff window can be used to encourage more transmission activities while keeping the required SINR. In particular, a joint Adaptation of link Rate and backoff Contention window, abbreviated as ARC, is devised. Our ARC protocol first estimates the optimal contention window (optCW) based on Cali's approximation methods. On transmission successes (failures), the current contention window size cw_p should be compared with optCW. If cw_p > optCW (cw_p < optCW), then cw_p is decreased (increased) to perform more aggressive (conservative) transmission attempts while leaving the link rate R unchanged. Otherwise, R is upgraded (reduced) to the next higher (lower) rate. One nice property of ARC is the ability to intelligently maintain link stability, avoiding unnecessary rate fluctuations. Simulation results show that the proposed ARC protocol outperforms several traditional link adaptation mechanisms. We also propose an analytic Markov chain model on ARC operations for performance validation.

Appreciation

Contents

List of Tables

List of Figures

Chapter 1

Background

IEEE 802.11 plays an important role in wireless communication. Due to the development of various modulation techniques and coding schemes, multiple transmission rates are now supported by 802.11 physical layers. For example, 802.11b provides 4 kinds of data rates (1, 2, 5.5 and 11M bps), while 802.11a/g provides 8 kinds of data rates (6, 9, 12, 18, 24, 36, 48 and 54M bps). Higher transmission rate means higher potential throughput, because it shortens the transmission time in one transmission attempt. However, higher data rate also implies higher packet corruption probability for receiver requires higher Signal-to-Interference-and-Noise Ratio ($SINR$) to successfully decode packets. If the $SINR$ perceived at the receiver is lower than $SINR$ threshold, the signal may not be decoded correctly.

Each data rate is associated with a certain $SINR$ threshold. The method of selecting an appropriate link rate for transmitting/retransmitting packets is generally comprehended as the link (rate) adaptation mechanism. Various rate-adaptive algorithms have been proposed [1, 5, 8–10, 12, 14, 15, 17, 19, 22]. The most commonly used rate adaptation technique is perhaps auto-rate fallback (ARF), which is widely implemented in present wireless devices [10]. We provide a more detailed review on the ARF protocol in Sec 2.2. Based on ARF, in the literature, plenty of rate-adaptive mechanisms have been proposed to improve the ARF performance [1, 5, 9, 12, 14, 17, 22]. Rate adaptation can also be combined with tuning other physical parameters such as power or carrier sense threshold [1, 15].

In general, rate-adaptive schemes can be classified into two categories: open-loop and closed-loop approaches. Open-loop approaches perform rate adaptations based on the information of whether ACK message is successfully returned or not, which we call *implicit feedback*. ARF is such an open-loop strategy. On the other hand, closed-loop approaches require the receiver to gather extra information such as $SINR$ statistic and inform the sender via control messages, called *explicit feedback*. Consequently, closed-loop approaches may result in better rate predictions, at the expense of controlling overhead. Two representative mechanisms in this category are receiver-based auto-rate (RBAR) and opportunistic auto-rate (OAR) protocols [8,19]. Details on RBAR and OAR are provided in Sec 2.3 and Sec 2.4, respectively.

In this paper, we propose an open-loop rate adaptation protocol, entitled ARC, for IEEE 802.11 multi-rate wireless network. A succinct review on the 802.11 MAC operations is presented in Sec 2.1. The proposed ARC protocol performs link adaptations by firstly considering if a proper backoff window has been reached. We estimate the optimal contention window ($optCW$) based on Calí's approximation methods [4]. On transmission successes (failures), the current contention window size cw_p should be compared with $optCW$. If $cw_p > optCW$ ($cw_p < optCW$), then cw_p is decreased (increased) to perform more aggressive (conservative) transmission attempts while leaving the link rate R unchanged. Otherwise, R is upgraded (reduced) to the next higher (lower) rate. One nice property of ARC is the ability to intelligently maintain link stability, avoiding unnecessary rate fluctuations.

The remainder of this paper is organized as follows. In Sec 2, we review the binary exponential backoff (BEB) mechanism in 802.11 standard, and classic rate adaptation works: ARF, RBAR, and OAR. Sec 3 introduces our ARC protocol. Simulation results and comparisons with other major multi-rate algorithms are provided in Sec 4. To validate the simulation results, we mathematically analyze our ARC operations based on the Markov chain model in Sec 5. Finally, in Sec 6, we conclude this paper.

Chapter 2

Preliminaries

2.1 Back-off Mechanism in 802.11 Standard (BEB)

802.11 standard defines two types of media access mechanisms: the Point Coordinate Function (PCF) and the Distributed Coordinate Function (DCF). PCF is a centralized polling-based MAC mechanism, which provides contention-free and time-bounded services. On the other hand, DCF is based on CSMA/CA, mandating stations carrier sense the channel media before transmitting packets. In DCF, every station has a backoff contention window (CW) for collision avoidance. Specifically, at the first transmission attempt, CW is set to the minimum value (cw_{min}). A station generates a backoff timer

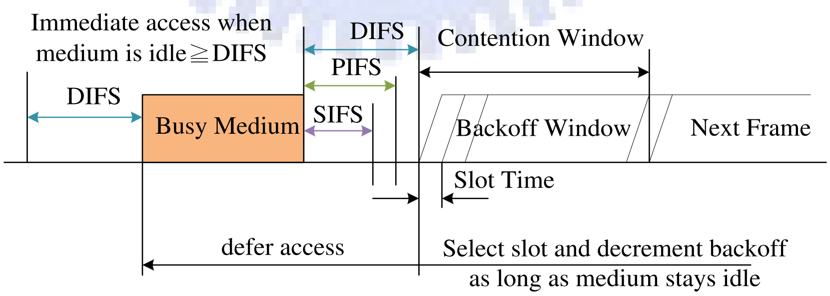


Figure 2.1: 802.11 MAC mechanism.

uniformly from [0,CW-1], and then starts to count down. When the timer counts down to zero, the station gets the privilege to access the channel. On unsuccessful transmission (ACK not returned), a binary exponential backoff (BEB) mechanism is used to relieve the contention level. In particular, the station has to double its CW size until CW reaches the maximum CW (cw_{max}) value. On successful transmission (ACK returned), DCF resets CW back to cw_{min}. The 802.11 MAC operations are illustrated in Fig.2.1.

2.2 Auto-rate Fallback (ARF)

ARF is the most widely implemented rate-adaptive scheme. It was originally used in WaveLAN-II devices, one of the early 802.11 products [10]. The key algorithm of ARF is that sender attempts to upgrade its transmission rate after successfully receiving 10 consecutive ACK frames, whereas the sender switches to a lower rate if it encounters 2 consecutive unsuccessful transmissions (i.e., missing ACK frames or the sender waits longer than timeout). If there is no traffic that has been sent for the present time, then station transmits packet with the highest possible rate. Although ARF is easy to implement, it has one attendant drawback: ARF can not work efficiently under stable or fluctuated channel conditions. That is, either it will constantly try to upgrade the transmission rate (which $SINR$ cannot support), leading to unnecessary packet collisions, or can not react quickly enough to match the fluctuated channel conditions.

2.3 Receiver-based Auto-rate (RBAR)

RBAR is a receiver-based rate-adaptation mechanism [8], which makes the rate adaptation decision based on channel quality estimated at the receiver and informs the sender via RTS/CTS handshaking mechanism. In RBAR, receiver utilizes RTS packet to obtain the RSSI information, and then selects an appropriate data rate provided in CTS to inform the sender. The rate handshaking is confirmed by another Reservation SubHeader (RSH) control message from the sender. Two main drawbacks exist in the RBAR protocol. One is the controlling overhead caused by rate negotiation on a per-packet basis. The other is the fact that RSSI estimation is not precisely supported in most wireless devices, reducing the practical feasibility of RBAR protocol.

2.4 Opportunistic Auto-rate (OAR)

OAR is an opportunistic media access protocol for multi-rate IEEE 802.11 [19]. OAR is extended from RBAR. The key idea of OAR is that the station with good channel conditions is granted to access the channel for a duration that allows multiple packet transmissions, compared with a single packet at the base rate. It also uses RTS/CTS packet exchange to obtain the channel condition. By exploiting the high-quality channel, a station can transmit more data packets when channel condition is good, hence increasing the system throughput. Furthermore, OAR ensures that all stations can access the channel for a equal time-share regardless of their channel condition. OAR has an improved throughput performance than RBAR, at the cost of requiring more communication overhead and extra 802.11 MAC modifications.

Chapter 3

Our ARC Protocol

3.1 Problem Statement

In wireless networks, successful data reception is highly dependent on the Signal-to-Interference-and-Noise Ratio ($SINR$) at the receiver. IEEE 802.11 supports multiple link rates at the physical layer. Each link rate is associated with a certain required $SINR$ threshold for successfully decoding received packets. Suppose constant noise and no power adjustment exists, apparently $SINR$ is solely affected by the accumulated interference power level I. Traditional link rate adaptation approaches try to reduce the transmit rate (hence lower $SINR$ threshold is required) on transmission failures (potentially due to the increased denominator I of $SINR$), whereas upgrade the transmit rate (hence higher $SINR$ threshold is required) on successful transmissions (potentially due to the decreased denominator I of $SINR$). The accumulated interference power level I in some sense indicates the medium congestion status. In 802.11, on transmission failures, the DCF performs a binary exponential backoff mechanism to discourage channel access attempts. When traditional link adaptation is applied, both rate reduction and binary backoff represent double penalties for this wireless link, which may cause overly conservative transmission attempts. On the other hand, once transmission succeeds, 802.11

DCF resets the backoff contention window to the minimum value to encourage channel access attempts. At the same time, traditional link adaptation may also decide to increase the data rate, which leads to overly aggressive transmission attempts. We observe this improper interaction of link rate and backoff mechanism that harms the 802.11 system performance, due to separate consideration of those two parameters.

Motivated by the above observations, rather than independently dealing with the two parameters, we propose to jointly consider the link rate and contention window adaptations in a unified framework. In particular, we propose to perform link adaptations by firstly considering if an optimal backoff contention window has been reached. To obtain the *optimal contention window* (*optCW*), in this paper, we adopt the approximation methodologies introduced by [4].

3.1.1 optCW Estimation

For analytical tractability, the authors in [4] considers a p-persistent version of 802.11 DCF. The results suggest that an optimal transmission attempt probability (p_{opt}) can be obtained by observing number of idle slots and active nodes (M) within the transmission range. Once p_{opt} is available, the value for *optCW* can be approximated. We run several simulation experiments to estimate the *optCW* for various number of active nodes based on this method in ns-2 simulator. Table 3.1 shows some of the results.

Table 3.1: optCW estimation

Active Nodes	p_{opt}	**optCW**
M=5	0.02486	80
M=10	0.01170	171
M=15	0.00777	257
M=20	0.00579	345
M=25	0.00461	433
M=30	0.00384	522

Table 3.2: Parameters used in our ARC protocol

Parameter	**Description**
optCW	optimal contention window size
cw_p	present contention window size
R	current transmission rate
++	increase transmission rate to the next higher one
$--$	decrease transmission rate to the next lower one
C_i	system-tuned incremental constant (default C_i=10)
C_d	system-tuned decremental constant (default C_d=10)
op^+	op^+ can be $'+'$ or $'*'$ (default $op^+ = '+'$)
op^-	op^- can be $'-'$ or $'/'$ (default $op^- = '-'$)

3.2 ARC Algorithm

In this section, we present the operation details of the proposed ARC protocol. As previously stated in Sec 3.1, we observe that the link rate and contention window parameters should be jointly considered in adaptations. The ARC protocol performs link adaptations by firstly checking if the optimal contention window (*optCW*) has been reached (refer to Sec 3.1.1 for mechanism on *optCW* estimation). Specifically, if the medium congestion level I can be reduced by imposing a larger backoff window on transmissions, then there may be no need to decrease the link rate, given $SINR$ can be sustained. Conversely, if there is extra interference that may be tolerated in I, a smaller backoff window can be used to encourage more transmission activities while keeping the required $SINR$. In other words, a joint Adaptation of link Rate and backoff Contention window, abbreviated as ARC, is devised.

Our ARC protocol first estimates the optimal contention window (*optCW*) by exercising Calí's approximation approaches. On transmission successes (failures), the current contention window size cw_p should be compared with *optCW*. If $cw_p > optCW$ ($cw_p < optCW$), then cw_p is decreased (increased) to perform more aggressive (conservative) transmission attempts and leave the link rate R unchanged. Otherwise, R

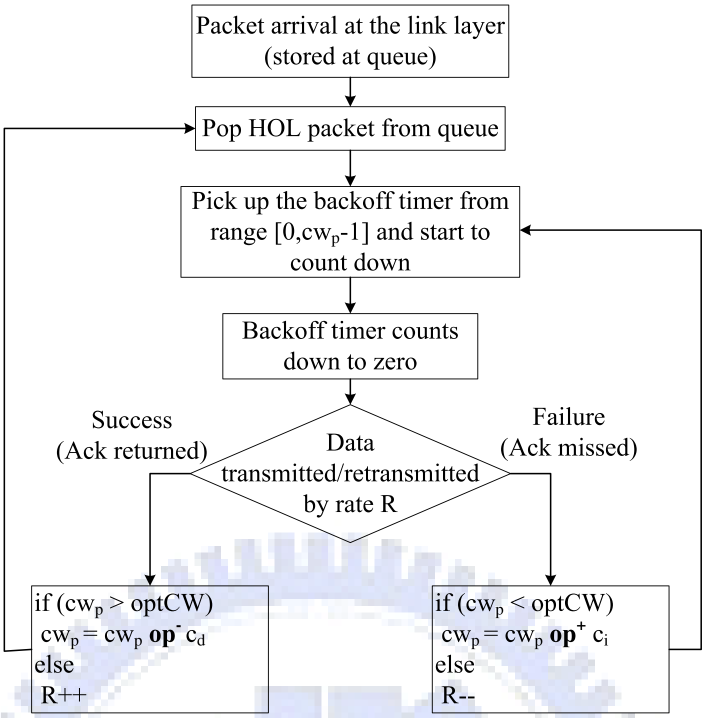


Figure 3.1: Backoff procedure and rate adaptation of the ARC algorithm (Here we omit the illustration of carrier sensing and DIFS for brevity).

is upgraded (reduced) to the next higher (lower) rate. Note that the default binary exponential backoff (BEB) in 802.11 DCF has been proved to be an inefficient mechanism under many communication circumstances. As several previous works have pointed out, the BEB mechanism in 802.11 DCF does not adapt to the wireless environment wisely [13, 21, 23]. Thus for the incremental (decremental) function in our ARC protocol, we propose to use a system-tuned incremental (decremental) constant, denoted as C_i (C_d) for design flexibility in the CW adjustment strategy. Moreover, the CW increment (decrement) operation, denoted as op^+ (op^-), can be an ADDITION (SUBTRACTION) or a MULTIPLICATION (DIVISION) function. Default op^+ (op^-) in ARC is an ADDITION (SUBTRACTION) function with default $C_i = 10$ ($C_d = 10$). Related parameters used in our ARC protocol are summarized in Table 3.2. The detailed operations of ARC are illustrated in Fig.3.1.

3.2.1 Discussion

In the ARC protocol, we do not set the initial CW as $optCW$, because the $optCW$ parameter is estimated at the sender which does not necessarily reflect the contention status at the receiver. Though from simulations results, $optCW$ does give a good indication on setting CW in most cases. Due to the common hidden terminal phenomenon existing in wireless networks, we suggest $optCW$ should only be used as a good reference when tuning CW. Thus in our ARC protocol, the initial CW is set to cw_{min} as the original DCF does. We let real transmission behaviors adjust the CW parameter gradually. Once $optCW$ has been reached, the adaptation on link rate comes into play. In this manner, the ARC algorithm can tolerate the imprecise estimation of $optCW$, and still properly react to the varying channel conditions (without requiring extra controlling overhead). We investigate the hidden terminal problem by running simulations in Sec 4.4 to verify the impact of inaccurate $optCW$ on ARC performance.

Another design feature of ARC is we try to adapt CW before adjusting rate R. According to [7], heterogeneous link rates are not desirable in terms of aggregate throughput. We seek to to maintain rate stability in ARC by avoiding arbitrarily adjusting link rates. In addition, since the $SINR$ value is not practically obtainable by current hardware functionality, an optimal R is not easy to estimate. Thus in the current ARC framework, we propose to tune CW before R.

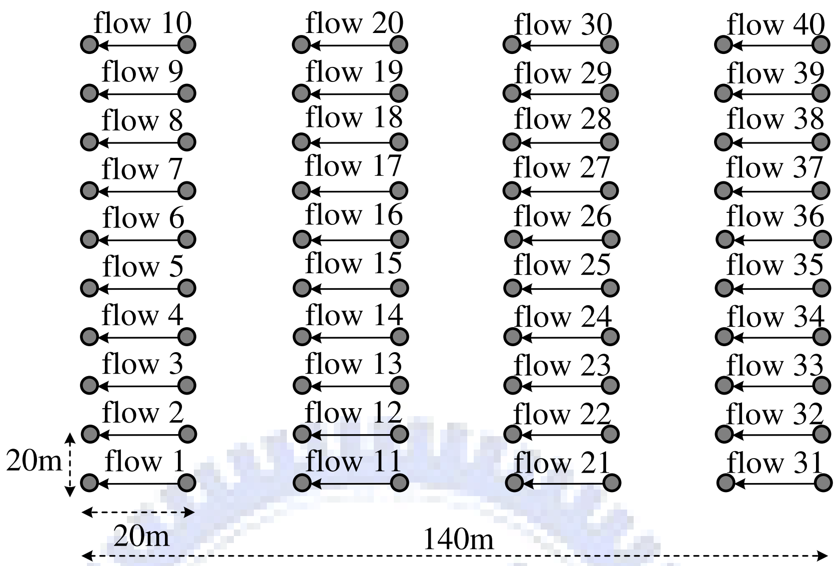
Chapter 4

Simulation Results

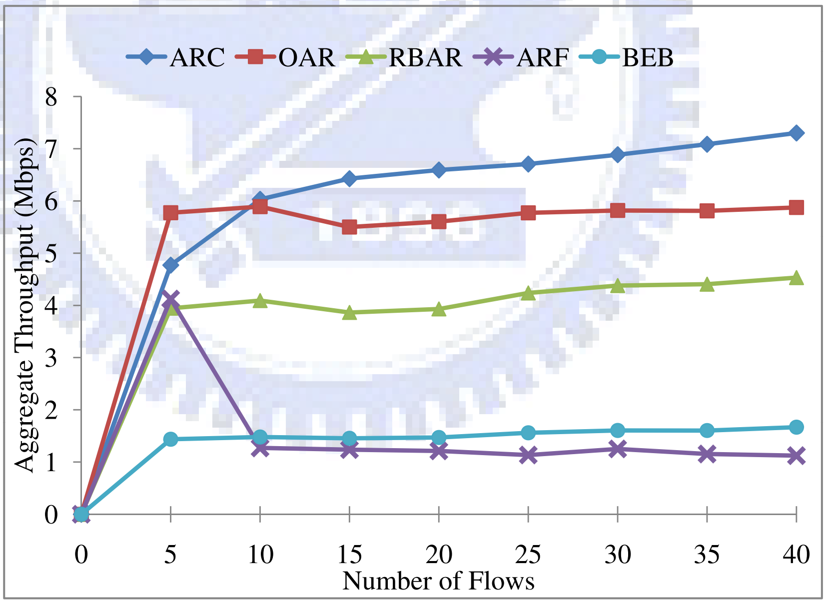
In this section, we run simulations in the ns-2 simulator. We add our ARC module in the dei80211mr library that supports 802.11b multi-rate PHY. Four link rates are available: 1, 2, 5.5, and 11M bps. Two-ray ground radio propagation model is used. CBR traffic (sending rate = 1M bps) is generated with packet size of 1000 bytes. All network nodes are static. We let every node randomly start transmission within the time range from 0 to 2 seconds to reduce initial collisions. MAC parameters $cw_{min} = 32$ and $cw_{max} = 1024$ are used. Total simulation time is 20 seconds. Each statistic is obtained from the average of 20 experiments. For comparison purpose, we also implement BEB (with fixed rate at 2M bps), ARF, RBAR, and OAR mechanisms. For ARF, RBAR, and OAR, the default binary exponential backoff is used as the CW adjustment strategy. Except for BEB, which has link rate fixed at 2M bps, all mechanisms set their starting link rate at 11M bps.

4.1 Grid Network

Fig.4.1(a) illustrates a grid network where nodes are placed uniformly in a rectangular area. A maximum of 40 traffic flows are generated. Fig.4.1(b) shows the system throughput against number of flows for different approaches. As we can see from this figure, our

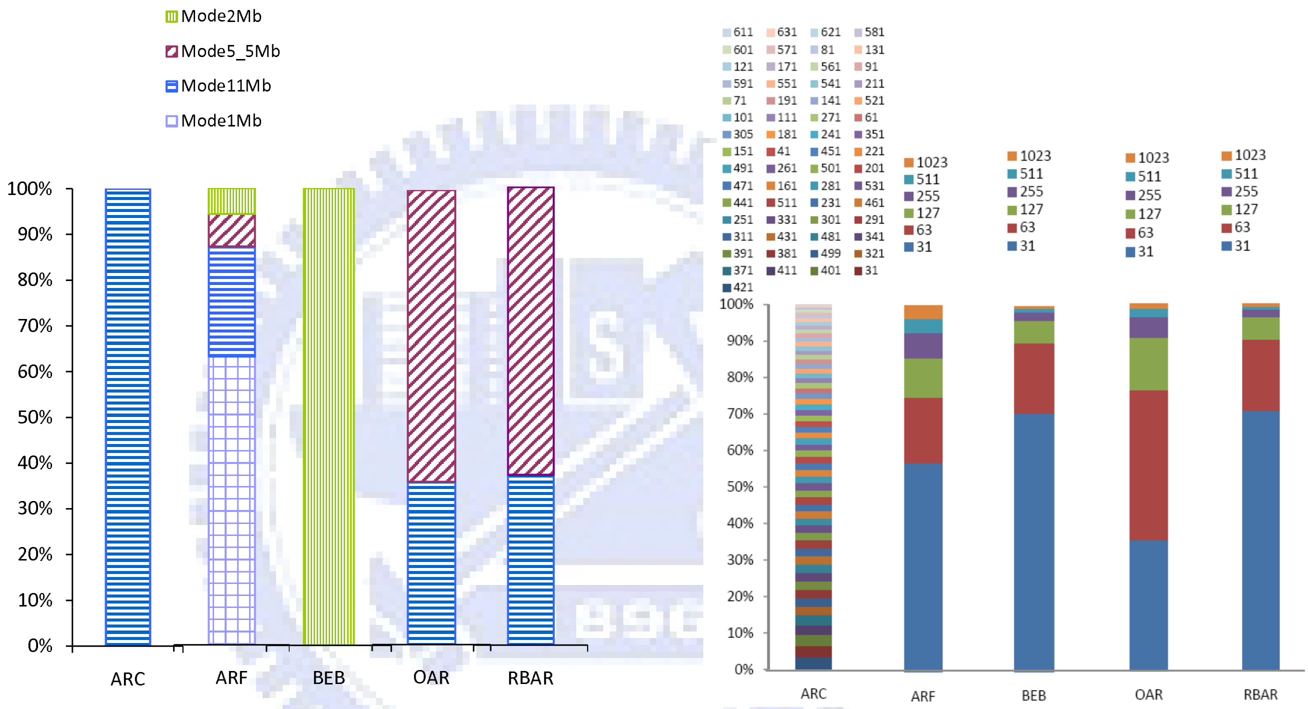


(a) The grid topology



(b) System throughput

Figure 4.1: Performance comparison in grid network.



(a) Link rate utilization (b) CW statistics

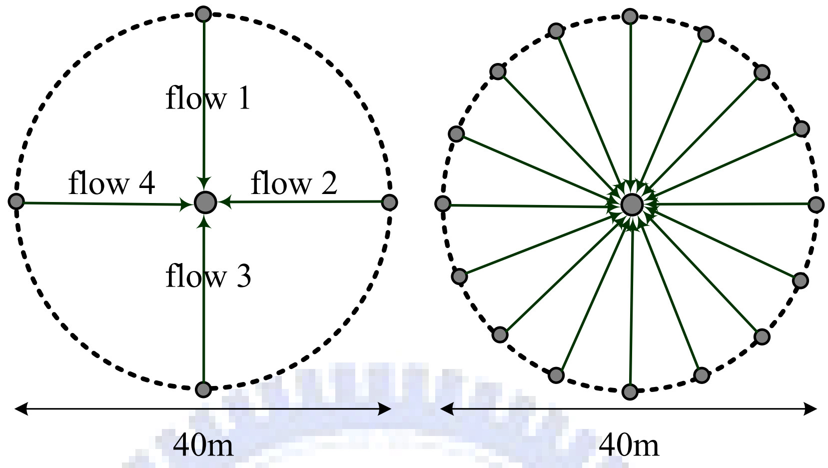
Figure 4.2: Performance comparison in grid network.

ARC protocol outperforms all other strategies when the number of flows grows larger than 10. While throughput of other mechanisms saturates when the number of flows reaches 5, throughput of ARC continues to increase steadily.

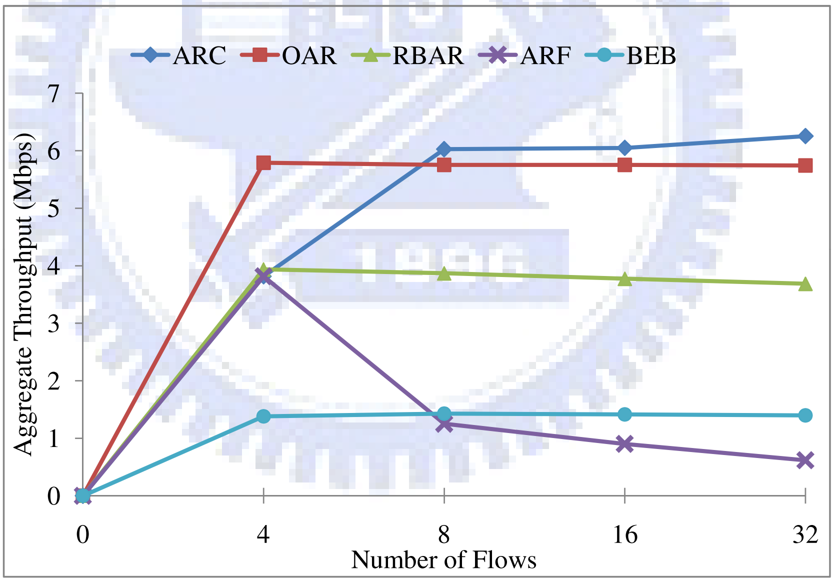
In order to have a better understanding of the detailed link rate and CW adaptation process, we provide the link rate utilization and CW statistics, for the case of 40 flows, in Fig.4.2(a) and Fig.4.2(b) respectively. From those figures, we observe that ARC keeps the link rate steadily at the highest (11M bps), while frequently adjusting the CW values around *optCW* (here $optCW = 698$). Other strategies use binary exponential backoff, thus their CW only takes on a few values. For OAR and RBAR, rates of 11M and 5.5M are used with a larger proportion set at 5.5M bps. For ARF, all four rates are mixed with the major proportion set at the lowest 1M bps. Due to the protocol nature of ARF (presented in Sec 2.2), it is easier to decrease rate (on 2 consecutive failures) than to increase rate (on 10 consecutive successes). Consequently, ARF performs even worse than BEB when the number of flows increases over 10 (shown in Fig.4.1(b)), because it becomes harder for ARF to bounce back to a higher rate in heavily contended environment. With the assistance of judiciously tuning CW, our ARC protocol effectively sustain the high link rate while providing sufficient $SINR$.

4.2 Star Network

In this experiment, we create a scenario to simulate a extremely contended network. Fig.4.3(a) shows a star topology, where the central node is a common receiver. All traffic flows are destined at the central node. In this case, the contention may not be resolved by CW or rate adjustment alone. We test our ARC protocol in this scenario. Fig.4.3(b) plots the system throughput for all strategies. ARC performs the best when number of flows is over 8, though the performance improvement is not as pronounced as that in the grid topology. The throughput of ARC increases steadily despite the extremely contended



(a) The star topology



(b) System throughput

Figure 4.3: Performance comparison in star network.

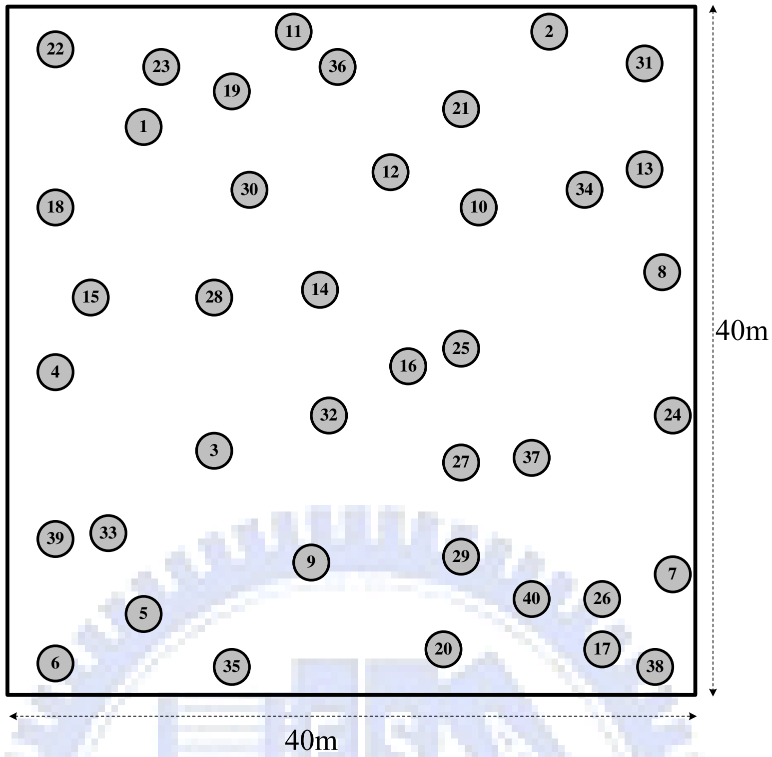
communication behavior.

4.3 Random Network

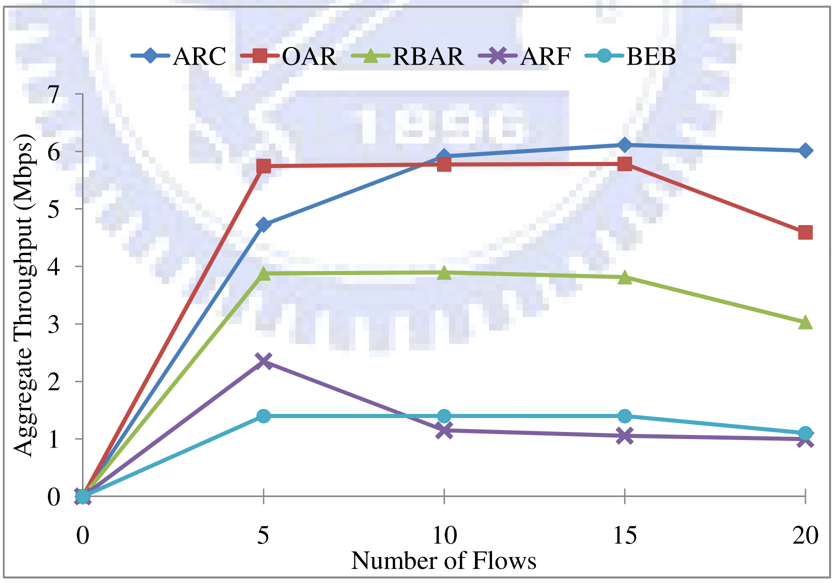
Fig.4.4(a) illustrates a 40-node random network in a 40×40 sq. meters area. We test all strategies by randomly generating a maximum of 20 flows in the network. Fig.4.4(b) shows the system throughput performance. Throughput of both OAR and RBAR decreases noticeably after the number of flows reaches 15, whereas ARC throughput remains high due to the flexibility of jointly tuning the link rate and CW parameters. This experiment again demonstrates the robustness of ARC protocol.

4.4 Hidden Terminal Problem

In this section, we investigate the impact of hidden terminal problem on ARC protocol. Several previous works have analyzed the effect of hidden terminal problem on 802.11 system performance [3, 11]. Specifically, if hidden terminals exist in the networking environment, the observed contention status at the sender is different from that at the receiver. Such inconsistent contention comprehension affects the accuracy of *optCW* estimation in our ARC protocol. We set up a communication scenario in Fig.4.5(a). For communication pair $A \rightarrow B$, we simulate different contention levels at both sides. Define the left circle in Fig.4.5(a) as the *sender zone* and right circle as the *receiver zone*. We denote the flow distribution such as S5R15 to indicate 5 flows and 15 flows are generated at the sender zone and receiver zone respectively. Two *optCW* values are obtained at both the sender and receiver. We experiment both values for *optCW* settings in the ARC protocol. Fig.4.5(b) shows the results. Since whether a transmission succeeds or not is determined by the contention level of receiver, the link throughput of using receiver-estimated *optCW* in ARC is always higher than that of using sender-estimated *optCW*. The result is not

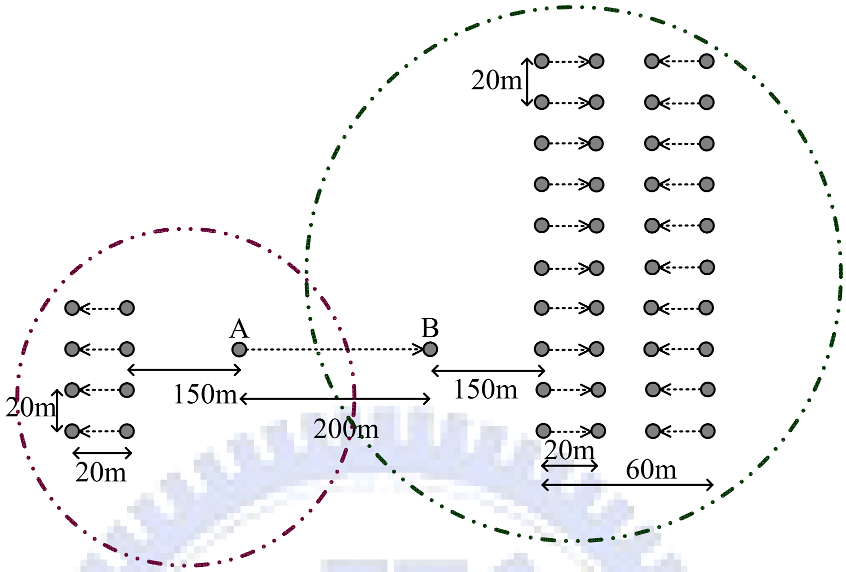


(a) The random topology

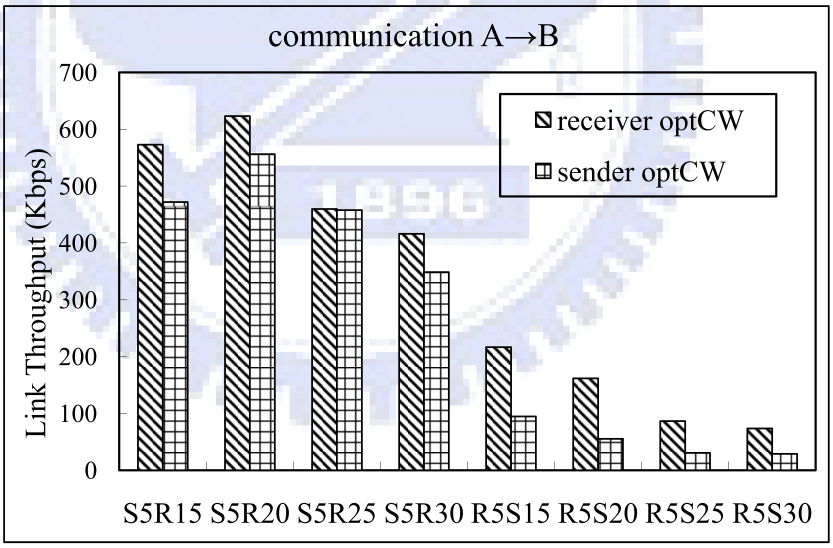


(b) System throughput

Figure 4.4: Performance comparison in random network.



(a) Communication scenario



(b) Link throughput

Figure 4.5: ARC performance influenced by inaccurate *optCW* due to the hidden terminal problem.

surprising. However, in this experiment, we do not count in the controlling overhead for communicating receiver-estimated *optCW* to the sender in real implementations, because we simulate a static network where nodes are stationary. In a mobile network, due to a constantly changing *optCW*, the receiver should inform the sender of this value on a per-packet basis, making the controlling overhead non-negligible. From Fig.4.5(b), we observe that in some cases, using sender *optCW* still can produce comparable throughput as the receiver *optCW* does. Though sender-estimated *optCW* is not as optimal as the receiver-estimated *optCW*, considering the extra communication overhead saved by using sender *optCW*, we observe that ARC actually has the capability of tolerating certain inaccuracy in *optCW* estimation.

4.5 Variants of ARC Protocol

As explained in Sec 3.2, the ARC protocol allows certain design flexibility for tuning CW value. Specifically, op^+, C_i, op^-, C_d are all system-tunable parameters. We denote ARC(+10,-10) as using op^+='+', op^-='-', and $C_i = C_d = 10$, which is actually the default setting in ARC. Based on the same grid topology as in Fig.4.1(a), we run experiments for six sets of ARC parameters. Fig.4.6 shows the system throughput performance. All six sets perform comparatively. Note that ARC(*2,/2) executes CW adjustment similar to BEB, but with CW value bounded around *optCW*. In addition, the rate adaptation comes into play when CW tuning does not work in ARC. As a result, the ARC protocol is robust in various contending environments. Furthermore, from Fig.4.6, we also observe that ARC is self-adaptive, and the performance distinction in different system parameter settings is insignificant.

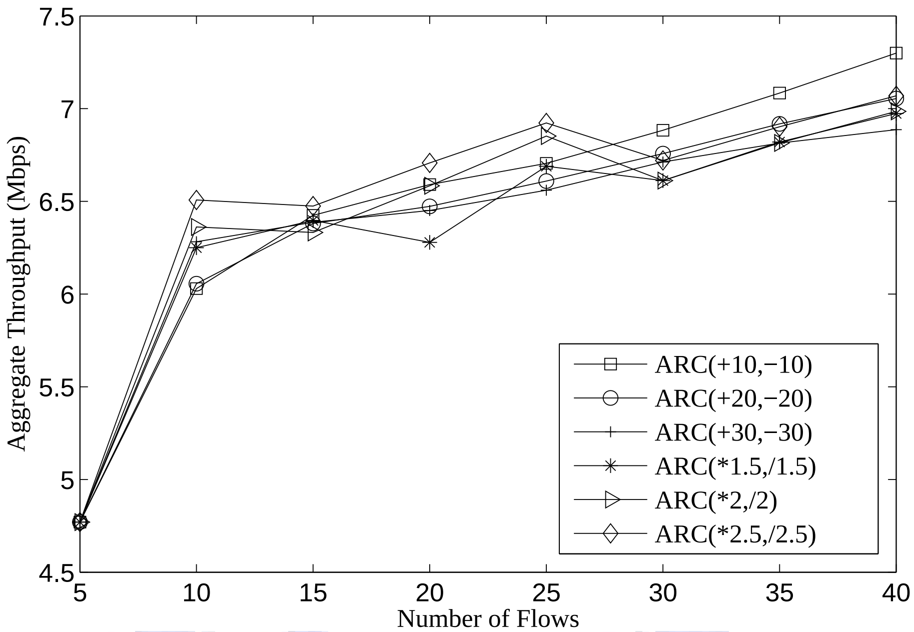


Figure 4.6: Throughput comparison of ARC variants.

Chapter 5

Model Validation

We build a Markov chain model to evaluate the ARC performance. Similar methodology has been used by [2, 6, 16, 18, 20]. However, those works deal with CW and link rate independently. In [2, 18], the authors analyze the fixed rate 802.11 DCF throughput, whereas authors in [6,16,20] analyze the DCF performance under multi-rate environment. Due to the jointly adaptation of link rate and DCF CW size in ARC protocol, we basically extend the Markov chain model from [2] to consider both parameters in transition states. We study an 802.11b network with four rates: 1, 2, 5.5, and 11M bps. Suppose n contending stations exist in the network, and each station always has a packet ready for transmission. Define $R_iB_0^j$ as the state with link rate i (Mbps) in the j^{th} backoff stage when the backoff timer counts down to zero. Fig.5.1 illustrates the simplified Markov chain model (the backoff counting down process is not shown), where p_i^f denotes the failure probability when transmitting with rate i. In 802.11b, each data rate is associated with a certain $SINR$ threshold for some bit error rate (BER), as shown in Table 5.1.

Table 5.1: SINR thresholds in 802.11b

Rate	**1Mb**	**2Mb**	**5.5Mb**	**11Mb**
SINR (dB)	-2.92	1.59	5.98	6.99

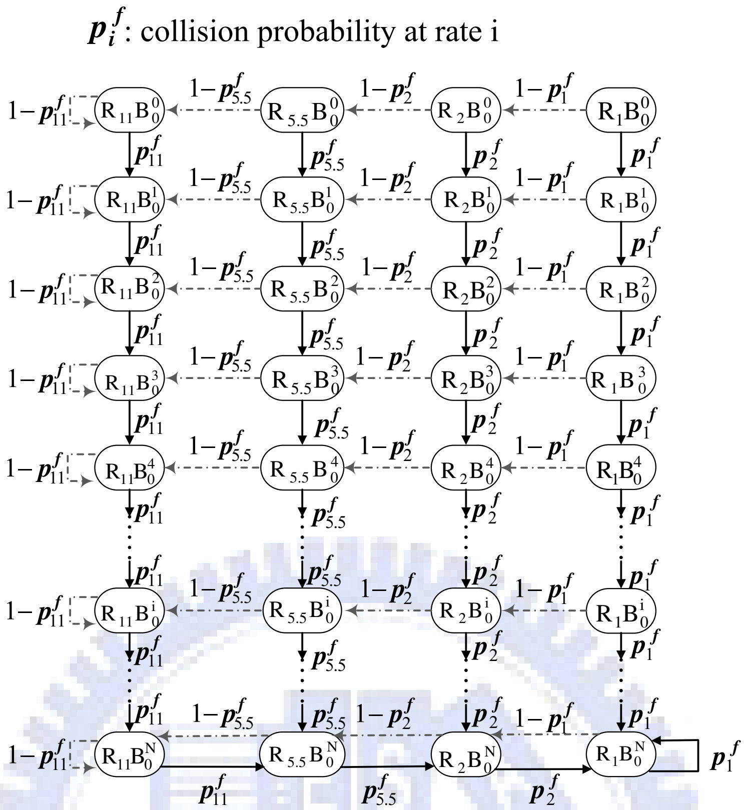


Figure 5.1: Simplified Markov model of ARC operations.

Thus p_i^f can be derived accordingly (here we omit the derivation details due to space limitation).

In our ARC operations, the default incremental function is ADDITION operation with constant increment $C_i = 10$. Define w_j as the CW size in backoff stage j. Then we have

$$w_j = cw_{min} \pm C_i \cdot j, \qquad j \in [0, N] \tag{5.1}$$

where N indicates the backoff stage that CW reaches *optCW*. Based on the Markov chain, we can model the transition probabilities as follows,

$$q_i(j,0) = \begin{cases} p_i^f \cdot q_i(j-1,0), \quad i=1; 0<j<N \\ p_{i+1}^f \cdot q_{i+1}(j,0) + p_i^f \cdot q_i(j-1,0) \\ +p_i^f \cdot q_i(j,0), \quad i=1; j=N \\ (1-p_{i-1}^f) \cdot q_{i-1}(j,0), i=2,5.5; j=0 \\ p_i^f \cdot q_i(j-1,0) + (1-p_{i-1}^f) \cdot q_{i-1}(j,0), \\ \qquad i=2,5.5; 0<j<N \\ p_{i+1}^f \cdot q_{i+1}(j,0) + p_i^f \cdot q_i(j-1,0) \\ +(1-p_{i-1}^f) \cdot q_{i-1}(j,0), \\ \qquad i=2,5.5; j=N \\ (1-p_i^f) \cdot q_i(j,0) + (1-p_{i-1}^f) \cdot q_{i-1}(j,0), \\ \qquad i=11; j=0 \\ (1-p_i^f) \cdot q_i(j-1,0) + p_i^f \cdot q_i(j-1,0) \\ +(1-p_{i-1}^f) \cdot q_{i-1}(j,0), \\ \qquad i=11; 0<j \leqq N \end{cases} \tag{5.2}$$

where $i \pm 1$ indicates changing present link rate to the next higher (or lower) level. Because of chain regularities, we have

$$q_i(j,k) = \frac{w_j - k}{w_j} \cdot q_i(j,0), \quad \forall k \in [0, w_j - 1] \tag{5.3}$$

Following the probability conservation property, we also have

$$1 = \sum_i \sum_{j=0}^{N} \sum_{k=0}^{w_j-1} q_i(j,k) \tag{5.4}$$

From the above derivations, we can now express the initial state $q_1(0,0)$ by p_i^f and w_j, as shown in Eq.5.3. Hence the transmission probability p_{tx} can be derived as

$$p_{tx} = \sum_i \sum_{j=0}^{N} q_i(j, 0), \qquad i \in \{1, 2, 5.5, 11\} \tag{5.5}$$

5.1 Analytic Throughput of ARC

Now we theoretically analyze the system capacity by studying the events that occur in one transmission attempt. Suppose $p_{tx}^{R_i}$ is the transmission probability at rate i. We have

$$p_{tx}^{R_i} = \sum_{j=0}^{N} q_i(j, 0) \times \frac{1}{p_{tx}} \tag{5.7}$$

Let L/L_{ACK} be the length of data/ACK frame size. R_{ACK} is the basic rate used to transmit ACK frame. t_{PLCP}, t_{SIFS} and t_{DIFS} are time periods of physical layer overhead, SIFS, and DIFS, respectively. Then $t_{ACK} = t_{PLCP} + \frac{L_{ACK}}{R_{ACK}}$. Therefore, the successful transmission time for data rate i can be derived as

$$T_s^{R_i} = t_{PLCP} + \frac{L}{R_i} + t_{SIFS} + t_{ACK} + t_{DIFS} \tag{5.8}$$

where successful transmission probability $T_s^{R_i}$ is given under the condition that at least one station is using the channel and only one station transmits with rate i. That is,

$$p_{tr} = 1 - (1 - p_{tx})^n \tag{5.9}$$

$$q_1(0,0) = \frac{2}{\sum_{j=0}^{N}[(p_1^f)^j + (p_2^f)^j(1-p_1^f) + (p_{5.5}^f)^j(1-p_2^f)(1-p_1^f) + (p_{11}^f)^j(1-p_{5.5}^f)(1-p_2^f)(1-p_1^f)](w_j+1)} \tag{5.6}$$

$$P_s^{R_i} = \frac{n(p_{tx}p_{tx}^{R_i})(1-p_{tx})^{n-1}}{p_{tr}}, \tag{5.10}$$

$$\Rightarrow p_{succ} = P_s^{R_1} + P_s^{R_2} + P_s^{R_{5.5}} + P_s^{R_{11}} \tag{5.11}$$

where p_{succ} is the successful transmission probability, equal to summation of all successful probabilities with different data rates. Consequently, the average successful transmission time, T_s, is the summation of transmission time that multiplies successful transmission probability for each data rate. In other words,

$$T_s = \sum_{i}^{11} p_s^{R_i} T_s^{R_i}, \qquad i \in \{1, 2, 5.5, 11\} \tag{5.12}$$

Now, we observe the collision events in the packet transmission. Let $P_e^{R_i}$ be the probability that a collision occurs for data rate i under the condition that more than one station is using the channel. Then,

$$\begin{aligned}
p_e^{R_1} &= [\sum_{i=2}^{n} \begin{pmatrix} n \\ i \end{pmatrix} (p_{tx}p_{tx}^{R_1})^i (1-p_{tx})^{n-i}] \times \frac{1}{p_{tx}} \\
p_e^{R_2} &= (\sum_{i=2}^{n} \begin{pmatrix} n \\ i \end{pmatrix} [\sum_{j=1}^{i} \begin{pmatrix} i \\ j \end{pmatrix} (p_{tx} \cdot p_{tx}^{R_2})^j (p_{tx}p_{tx}^{R_1})^{i-j}] \\
&\cdot (1-p_{tx})^{n-i}) \times \frac{1}{p_{tx}} \\
p_e^{R_{5.5}} &= (\sum_{i=2}^{n} \begin{pmatrix} n \\ i \end{pmatrix} \{\sum_{j=1}^{i} \begin{pmatrix} i \\ j \end{pmatrix} [\sum_{m=1}^{j} \begin{pmatrix} j \\ m \end{pmatrix} (p_{tx}p_{tx}^{5.5})^m \cdot \\
&(p_{tx}p_{tx}^{2})^{j-m}] \cdot (p_{tx}p_{tx}^{R_1})^{i-j}\} (1-p_{tx}^{n-i})) \times \frac{1}{p_{tx}} \\
p_e^{R_{11}} &= 1 - p_{succ} - p_e^{R_1} - p_e^{R_2} - p_e^{R_{5.5}}
\end{aligned} \tag{5.13}$$

Since date rate is inversely proportional to the transmission range, 1Mb has the farthest transmission distance. In the above equation, $p_e^{R_1}$ represents the probability that collision

Table 5.2: IEEE 802.11 DSSS PHY and MAC parameters

Parameter	**Value**
t_{plcp}	192 μs
propagation delay (δ)	1 μs
SIFS	28 μs
DIFS	128 μs
ACK	112 bytes
R_{ACK}	1M bps
slot time (σ)	20 μs
packet length (L)	1000 bytes

occurs when more than 2 stations transmit with 1Mb at the same time. $p_e^{R_2}$ represents the probability that collision happens when there are j stations transmitting with 2Mb and $j-i$ stations transmitting with 1Mb. In the same way, the collision probability of 5.5Mb and 11Mb can be derived accordingly.

On the other hand, we define T_e^i as the time spent in a collided transmission with data rate i. We have

$$T_e^{R_i} = t_{PLCP} + \frac{L}{R_i} + t_{DIFS} + To \tag{5.14}$$

where $To = t_{SIFS} + (t_{PLCP} + \frac{L_{ACK}}{R_{ACK}})$ to indicate the time that a colliding station waits for accessing a channel. Therefore, the average time spent in collided transmission can be expressed as

$$T_e = \sum_{i=1}^{11} p_e^{R_i} T_e^{R_i}, \qquad i \in \{1, 2, 5.5, 11\} \tag{5.15}$$

As a result, the analytic throughput of ARC can be derived as follows,

$$throughput = \frac{p_{tr} \cdot p_{succ} \cdot L}{(1-p_{tr})\sigma + p_{tr}T_s + p_{tr}T_e} \tag{5.16}$$

where σ is the slot time.

Table 5.2 summarizes the 802.11 PHY and MAC parameters used to obtain our an-

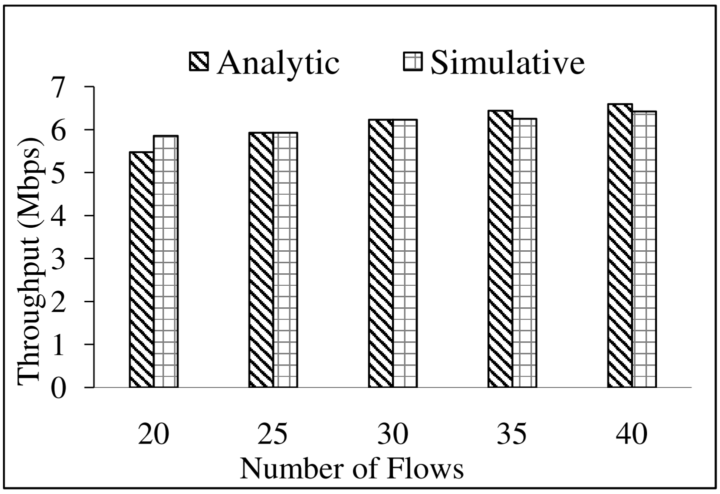


Figure 5.2: Performance validation of ARC.

alytic results. Based on the grid network topology (Fig. 4.1(a)), we run simulations for different number of flows. In this experiment, we set the starting rate to be 1M bps (not 11M bps as in the previous experiments) in order for ARC to experience all possible rates. Fig.5.2 shows the analytic and simulative throughput. The results demonstrate that the simulative data are quite consistent with analytic predictions, hence validating the ARC performance.

Chapter 6

Conclusions

In this paper, we proposed an open-loop link adaptation function that jointly considered the CW parameter for IEEE 802.11 multi-rate communication environments. The proposed ARC protocol does not require extra signalling overhead between the sender and receiver. One nice property of ARC is the ability to judiciously maintain link stability, avoiding unnecessary rate fluctuations. Simulations results showed that our ARC protocol produced more system throughput than other traditional rate adaptation techniques. A Markov chain model on ARC operations was also proposed to validate the performance.

Bibliography

[1] A. Akella, G. Judd, S. Seshan, and P. Steenkiste. Self-management in Chaotic Wireless Deployments. In *Proc. ACM MOBICOM*, Aug. 2005.

[2] G. Bianchi. Performance Analysis of IEEE 802.11 Distributed Coordination Function. *IEEE Journal on Selected Areas in Communications*, 18(3):535–547, Mar. 2000.

[3] M. Borgo, A. Zanella, P. Bisaglia, and S. Merlin. Analysis of the Hidden Terminal Effect in Multi-rate IEEE 802.11b Networks. In *Proc. WPMC*, pages 6–10, 2004.

[4] F. Cali, M. Conti, and E. Gregori. Dynamic Tuning of the IEEE 802.11 Protocol to Achieve a Theoretical Throughput Limit. *IEEE/ACM Transactions on Networking*, 8(6):785–799, Dec. 2000.

[5] C.-C. Chen, H. Luo, E. Seo, N. Vaidya, and X. Wang. Rate-adaptive Framing for Interfered Wireless Networks. In *Proc. IEEE INFOCOM*, pages 1325–1333, May 2007.

[6] W. Chu and Y.-C. Tseng. Performance Analysis of IEEE 802.11 DCF in a Multi-rate WLAN. *IEICE Transactions on Communications*, E90-B(10):2836–2844, Oct. 2007.

[7] M. Heusse, F. Rousseau, G. Berger-Sabbatel, and A. Duda. Performance Anomaly of 802.11b. In *Proc. IEEE INFOCOM*, pages 836–843, Mar. 2003.

[8] G. Holland, N. Vaidya, and P. Bahl. A Rate-adaptive MAC Protocol for Multi-hop Wireless Networks. In *Proc. ACM MobiCom*, pages 236–251, Jul. 2001.

[9] G. Judd and P. Steenkiste. Using Emulation to Understand and Improve Wireless Networks and Applications. In *Proc. USENIX NSDI*, May 2005.

[10] A. Kamerman and L. Monteban. WaveLAN-II: A High-Performance Wireless LAN for the Unlicensed Band. *Bell Labs Technical Journal*, 2(2):118–133, 1997.

[11] S. Khurana, A. Kahol, and A. P. Jayasumana. Effect of Hidden Terminals on the Performance of IEEE 802.11 MAC Protocol. In *Proc. IEEE LCN*, pages 12–20, Oct. 1998.

[12] J. Kim, S. Kim, S. Choi, and D. Qiao. CARA: Collision-aware Rate Adaptation for IEEE 802.11 WLANs. In *Proc. IEEE INFOCOM*, pages 1–11, Apr. 2006.

[13] Y. Kwon, Y. Fang, and H. Latchman. A Novel MAC Protocol with Fast Collision Resolution for Wireless LANs. In *Proc. IEEE INFOCOM*, pages 853–862, Mar. 2003.

[14] M. Lacage, M. H. Manshaei, and T. Turletti. IEEE 802.11 Rate Adaptation: A Practical Approach. In *Proc.ACM MSWiM*, Oct. 2004.

[15] T.-Y. Lin and J. C. Hou. Interplay of Spatial Reuse and SINR-determined Data Rates on CSMA/CA-based, Multi-hop, Multi-rate Wireless Networks. In *Proc. IEEE INFOCOM*, pages 803–811, May 2007.

[16] H. Park and C.-K. Kim. Performance Analysis of Multi-rate IEEE 802.11 WLANs with Channel Error . In *Proc. IEEE ICACT*, pages 1479–1481, Feb. 2007.

[17] D. Qiao and S. Choi. Fast-responsive Link Adaptation for IEEE 802.11 WLANs. In *Proc. IEEE ICC*, pages 3583–3588, May 2005.

[18] D. Qiao, S. Choi, and K. Shin. Goodput Analysis and Link Adaptation for IEEE 802.11a Wireless LANs. *IEEE Transactions on Mobile Computing*, 1(4):278–292, 2002.

[19] B. Sadeghi, V. Kanodia, A. Sabharwal, and E. Knightly. Opportunistic Media Access for Multirate Ad Hoc Networks. In *Proc. ACM MOBICOM*, Sep. 2002.

[20] G. Sharma, A. Ganesh, and P. Key. Performance Analysis of Contention Based Medium Access Control Protocols. In *Proc. IEEE INFOCOM*, pages 1–12, Apr. 2006.

[21] C. Wang, B. Li, and L. Li. A New Collision Resolution Mechanism to Enhance the Performance of IEEE 802.11 DCF. *IEEE Transactions on Vehicular Technology*, 53(4):1235–1246, Jul. 2004.

[22] Y. Xi, B.-S. Kim, J. bo Wei, and Q.-Y. Huang. Adaptive Multirate Auto Rate Fallback Protocol for IEEE 802.11 WLANs. In *Proc. IEEE MILCOM*, pages 1–7, Oct. 2006.

[23] S.-R. Ye and Y.-C. Tseng. A Multi-chain Backoff Mechanism for IEEE 802.11 WLANs . *IEEE Transactions on Vehicular Technology*, 55(5):1613–1620, Sep. 2006.