Enhancement Of Auto Rate Fallback (Earf) For Dynamic Rate Selection

Awni Hammouri

Department of Information Technology
Mutah University
61710 Mutah, Karak, Jordan

Abstract
Bit rate is an important issue since it can increase the throughput and make the network faster. However, in order to increase the rate, we should take the channel condition into account. Our proposed algorithm aims to increase the rate based on the negotiation between the sender and the receiver. Therefore, we make the receiver respond to the sender and based on its response, the sender can decide whether to increase the rate or keep the current one. IEEE 802.11 b provides a bit rate within the rates: 1, 2, 5.5 and 11 mbps, and to increase or decrease the rate it uses Auto Rate Fallback algorithm which depends on number of successive packet to make a decision.
Key words:
Bit rate selection algorithm, Auto Rate Fallback (ARF), Enhanced Auto Fallback (EARF), Bit error rate.

1. Introduction

Since introduced in 1980s, wireless network has been continuously studied and developed by many researchers and companies. Among various kinds of wireless network (which are usually distinguished by coverage and/or speed), wireless LAN is the one most popularly used in these days because of the maturity of its technology and practicality. Wireless LAN currently provides wider coverage than Blue tooth and faster and more stable data transfer than WiBro or WiMAX [7]. Many approaches have been proposed for wireless LAN and one of them, IEEE 802.11, is chosen as a standard, which is also known as Wi-Fi [4][6].

Still there are several protocols in 802.11 itself – a, b, g and n. They have different frequency range and data rate. Among those, the newest one, 802.11n (Released on Oct. 2009) has come into spotlight because it has the fastest data rate and widest indoor/outdoor range and Wi-Fi devices which recently come out to market usually provide 802.11n protocol. Table 1 shows the comparison of those protocols in summary [5][6].

Table 1: Summary of IEEE 802.11 standards

Protocol	Frequency	Data rate
802.11a	5GHz	6 to 54 Mbps
802.11b	2.4GHz	5.5 to 11 Mbps
802.11g	2.4GHz	6 to 54 Mbps
802.11n	2.4GHz/5GHz	7.2 to 150 Mbps

Like other network, to maximize throughput is the most significant issue in wireless LAN technology. While either constant bit rate or variable bit rate is used, variable bit rate is better for wireless LAN when the network connectivity is often unstable. To maximize throughput, one may control bit rate according to status of the network from time to time. If one selects high bit rate, the data can be transferred faster compared to low bit rate but the possibility of data loss also gets higher. On the other hand, when low bit rate is used, the data transfer would be stable but it may cause waste of bandwidth in case that the network connection has redundant capacity besides the transfer [1].

To resolve this issue, some bit rate selection algorithms have been suggested [3][5], and we've found one possibility of how to improve them. Therefore, we propose a modified version of bit rate selection algorithm and show how this algorithm is more effective than the earlier version with a simulation result.

In this paper, the decision of increasing the bit rate depends on the negotiations between the sender and the receiver. If the receiver accepts the higher rate, the sender increases the rate, otherwise, the current rate will be used.

In the rest of paper, we will first introduce related work which is considered as a reference and its limitation, and then we discuss the detail of our proposed algorithm. Next, simulation environment and the experiment result will be discussed and we will wrap up with conclusion.

2. Related Work

Auto Rate Fallback (ARF) [3] is the first bit rate selection algorithm, initially developed for Wave LAN 802.11 cards, which implemented the 802.11 DSSS standard. Wireless

Manuscript received February 5, 2018
Manuscript revised February 20, 2018

network condition may change due to many reasons, such as transient noise or moving receivers. Since the best transmission bit rate, determined by the number of consecutive successful transmissions, depends on the network condition, one might face situations where a higher or lower rate can be the best for the moment.

The ARF algorithm is designed to utilize the best rate and subsequently optimize the application throughput. How the algorithm works is explained, as followed:

- When an ARF enables transmitter starts a session, it will use the highest bit rate and will send a probe message. If the probe message is acknowledged, the transmitter will use the bit rate to send the actual payload, otherwise the bit rate will be decreased to the next lower step and send the probe message again.
- During a transmission the ARF algorithm sets a counter, which counts the number of consecutive successful transmissions. If the counter hits 10, it will send a new probe message on a higher bit rate if it exists. In case of an acknowledgement, the transmitter will alter to the new bit rate and in case of loss, the transmission bit rate will be reset back to the previous bit rate.
- But if the current transmission rate faces 1 or 2 packet failures due to network changes, the transmitter will decrease its bit rate.

The algorithm is simple and easy to implement by keeping small state information. Nevertheless, ARF suffers from two problems [5]:

- If the network condition changes frequently and fast, it cannot adapt effectively. For example, in networks where the channel condition changes occur in a shorter time period than 10 transmissions, ARF will always utilize the lower bit rate.
- If the network reaches a stable state, ARF will still try to alter to the next higher bit rate periodically, despite the fact that all probe messages will not get acknowledged. Consequently, the ARF algorithm wastes possible bandwidth in stable states.

ARF can handle short-term variations in network condition. It is also feasible, if the wireless receiver changes its position. But in practice, a worker with a laptop will not change its position while working. It is unlikely that someone types while moving. Accordingly, there will be a period of time where the network condition will be stable.

In this case, ARF will not perform well since it will still try to increase the bit rate every 10 successful transmissions and waste 1/11 of the available bandwidth, as described previously. So consequently, for a better bandwidth utilization where the network condition is stable, we have to decrease the number of probe messages.

In contrast to ARF, the Adaptive ARF (AARF) can continuously change the threshold, where a probe message should be sent on the next higher bit rate, to better reflect the network condition [5]. The current threshold will be determined according to history data by using Binary Exponential Back off (BEB). Similar to ARF, AARF will fall back to the previous bit rate, but in addition, AARF will increase the time where the next probe will be sent by multiplying the current threshold by two, until a number of 50 successful consecutive transmissions is not exceeded. Setting the required number of consecutive successful transmission on a high value might improve the throughput. But a high value also implies a slower reaction to short-term network condition changes.

A Soft Rate is proposed by [9] in which the log-likelihood ration is used of the correct bit at sender side. At the receiver side, the BER is precisely computed by the hints from the sender. The approach reduces the error in increasing or decreasing the rate due to the high interfering.

An algorithm to determine the optimal data rate of different data sources is proposed by [10]. It aims to minimize the total distortion for a given bit rate constraints.

The authors of [11] develop a quality of experience system for the management of the HTTP adaptive bit rate. To sum up, AARFs behavior will be the same as ARF if the networks condition varies fast. But since AARF reduces the number of probe messages when the network condition is stable, it can offer a better throughput.

The proposed method differs from the existing work in terms of the way of sensing the channel. The negotiations between the sender and the receiver is the base of changing the bit rate. This approach aims to reduce the number of dropped packets that most of the existing approach suffer from. Moreover, the throughput of the network will be increased as well.

3. Proposed Algorithm

3.1 Motivation

Traditional bit rate selection algorithms share some common problems, as followed:

- Since they are probing the network conditions, some bandwidth overhead is always present.
- Using probe messages limits the capability of the algorithm to react to short-term network condition changes without degrading application throughput significantly.

- Also, the fact, whether the new bit rate to be selected is feasible, is not really evaluated.

3.2 Algorithm Requirements

Based on the problems, we define the algorithm requirements, as followed:

- Negotiation between the sender and the receiver should not create additional bandwidth overhead.
- Next higher bit rate should be evaluated.
- Altering to the next higher bit rate only if the application throughput can increase.

In our algorithm (Section 3.3), we need to know how to compute Signal to Noise and interference Ratio (SNR), Bit Error Rate (BER), and Frame Error Rate (FER). Therefore, we explain the definition of these terms in the following paragraphs.

The SNR is calculated by the following formula [8].

$$SNR = 10 \times log \frac{Rx_{Power}}{Noise} \quad (1)$$

where Rx_{Power} is the frame signal at the receiver. It is calculated by propagation model and $Noise$ is calculated from the receiver sensitivity of the data rate used by the frame.

If other frames arrive at the receiver when it is currently receiving one frame, the SNR of this receiving frame is

$$SNR = 10 \times log \frac{Rx_{Power}}{Noise + \sum_{i=1}^{i-1} Rx_{Power_i}} \quad (2)$$

where Rx_{Power_i} is the signal strength of other frames at the receiver.

When SNR and modulation method are known, BER can be derived theoretically. We also can obtain BER value from empirical curves measured for particular product. There is a table which gives the relationship between BER and SNR and Modulation of Intersil HFA3681B chipset under AWGN channel model [8]. The table is used to calculate BER from SNR.

Since Preamble, PLCP header, data of one 802.11b frame may be transmitted with different modulation scheme and different modulation schemes have different receiver noise floor, we should calculate their SNR, BER separately. Next, we can calculate FER of that frame and determine whether the frame is received correctly or not [8].

3.3 Enhanced Auto Rate Fallback

The main idea for Enhanced Auto Rate Fallback (EARF) is to use the packet payload for making negotiations between sender and receiver, and based on these negotiations the algorithm will decide to increase the bit rate or stay on the same bit rate. Accordingly, negotiations between sender and receiver can be done without additional bandwidth overhead, since control information will be exchanged piggyback on the MAC frames.

Unlike the traditional algorithms, the proposed algorithm triggers the bit selection procedure after three successful transmissions. Having a shorter checking time results in a better ability to react to short-term network condition changes. The sender requests the receiver on the fourth message to raise the transmission bit rate to the next level. After receiving the request, the receiver calculates its frame bit error (FER) for the new requested bit rate and accepts the new bit rate if frame bit error rate is low enough by using the ACK frame. With permission in the ACK frame the sender starts to transmit with the new bit rate on the fifth message. In case of dropping network condition, the ARF bit rate decrease scheme will be used.

According to the algorithm requirements, EARF evaluates the next higher bit rate before altering to it, there will be no bandwidth overhead and finally, we can also handle with short-term network condition changes efficiently. Two boxes below explain the algorithms for the sender side and the receiver side.

Algorithm 1: Sender side

```
Set success_counter = 0
While sender_has_frames {
        If MAC_ACK_received then
                success_counter ++
        else
                success_counter = 0

        if successCounter == 3
                new_frame = new_frame || new_rate

        if EARF_ACK == YesAck then
                current_rate = newRate
}
```

Algorithm 2: Receiver side

```
for each received_paket{
   if new_frame == new_frame || net_rate then
      FBE = compute_FER(SNR){
      if FBE <sender_req then
         Response YesAck
```

4. Simulation Environment

For the simulation, we use Exata 2 Simulator to prove the feasibility of the proposed algorithm (EARF).

In our scenario as depicted in Figure 1, we have 8 nodes and constant bit rate (CBR) as an application between Node 2 and Node 6.

EARF needs to know some parameters from both physical and MAC layers in order to detect the FER. Moreover, EARF needs to activate the conventional ARF which already exists on the Exata simulator in order to compare the result of both algorithms.

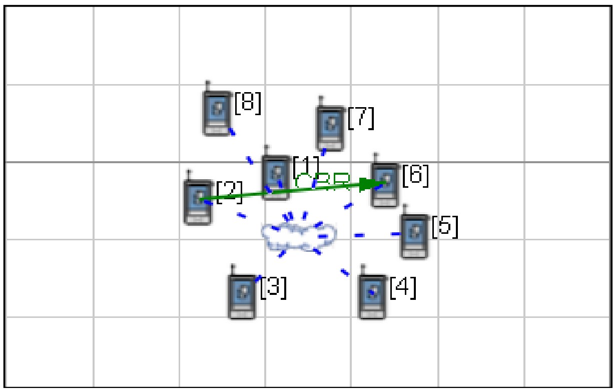

Fig. 1 Simulation Components

Firstly, Figure 2 explains the parameters setting in physical layer. Secondly, Figure 3describes the parameters setting in Mac layer. And finally, Figure 4 shows the general parameters setting in our simulation

[-] Radio Type	802.11b Radio
[-] Enable Auto Rate Fallback	Yes
Specify Broadcast Data Rate	No
Transmission Power at 1 Mbps (dBm)	15.0
Transmission Power at 2 Mbps (dBm)	15.0
Transmission Power at 6 Mbps (dBm)	15.0
Transmission Power at 11 Mbps (dBm)	15.0
Receive Sensitivity at 1 Mbps (dBm)	-94.0
Receive Sensitivity at 2 Mbps (dBm)	-91.0
Receive Sensitivity at 6 Mbps (dBm)	-87.0
Receive Sensitivity at 11 Mbps (dBm)	-83.0
Estimated Directional Antenna Gain (dB)	15.0
Packet Reception Model	PHY802.11b Reception Model

Fig. 1 Parameters used in physical layer

[-] MAC Protocol	802.11
Short Packet Transmit Limit	7
Long Packet Transmit Limit	4
RTS Threshold (bytes)	0
Stop Receiving after Header Mode	No
Station Association Type	None
Enable Directional Antenna Mode	No
Specify Network Security Parameters	No
MAC Propagation Delay	1 micro-seconds
Enable Promiscuous Mode	No
Enable LLC	No

Fig. 3 Parameters used in Mac layer

[-] Number of Channels	1
Channel Frequency [0]	2.4 GHz
Pathloss Model [0]	Two Ray
[-] Shadowing Model [0]	Constant
Shadowing Mean (dB) [0]	4.0
Fading Model [0]	None
Propagation Limit (dBm) [0]	-111.0

Fig. 4 General parameters used in our simulation

5. Simulation Results

We conducted a simulation to see the feasibility of our work. In this section, we will show the results from the simulation shown in the previous section.

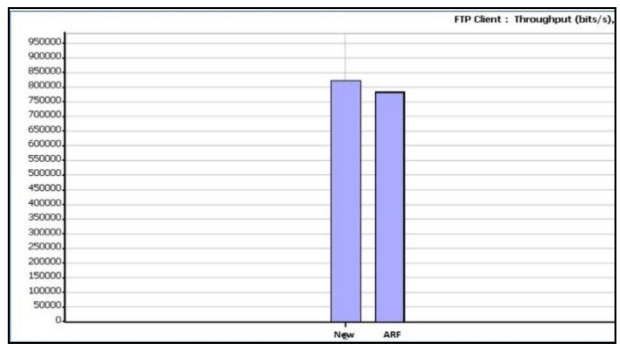

Fig. 5 Throughputproposed method compared to the one from ARF

Throughput

The first result of our work is the throughput. As we can see from Figure 5, the throughput of the new technique shows a small improvement compared to the current ARF. We claim that our technique is helpful in term of increasing the rate which affects the throughput. Our analysis is that we try to increase the rate before ten successive packets received and the other important thing in our approach is we do not use any additional packet for increasing the rate. We use only the packets transmitted from sender to receiver and the ACK control packet sent from receiver to sender.

Packet Retransmission

Another metric we have measured is the number of packets that have been retransmitted due to time out. Figure 6 shows that our work performs well and has fewer numbers of retransmitting packets. This happens because of two reasons. Firstly, we use the packet payloads between sender and receiver to make a decision. Secondly, the sender does not increase the rate until one receives an ACK from the receiver which tells one to increase the rate.

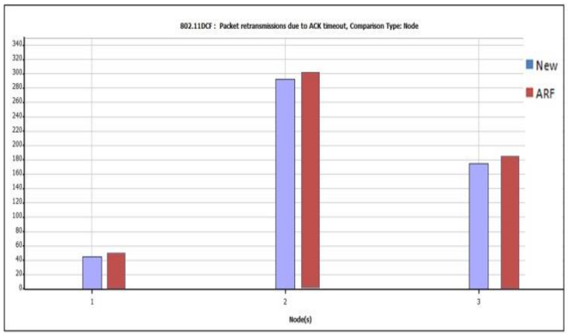

Fig. 6 Number of packet retransmissions due to ACK timeout

6. Conclusion

We propose a modified algorithm to increase the bit rate based on ARF algorithm. Our algorithm depends on the end nodes and the negotiations between them. Also, our algorithm performs better than ARF in terms of packet dropping and having the ability to take a decision whether increase the rate or not. Our algorithm starts to increase the rate after the third successive transmission while the ARF need ten successive transmissions before trying to increase the rate. Therefore, our algorithm makes the receiver participate in deciding whether or not increase the rate.

We made a simulation using Exata 2.1 Simulator to test our new technique. From the simulation, our algorithm shows an improvement from the previous algorithm in terms of throughput and number of packet dropping. We believe our algorithm performs well and can be a good improvement for the ARF algorithm.

References

[1] Bicket, J.C. 2005. Bit-rate selection in wireless networks. Master Thesis in Computer Science and Engineering MIT.
[2] Comer, D.E. 2009. Computer Networks and Internet 5th edition. Pearson-Prentice Hall.
[3] Kamerman, A. and Monteban, L. 1997. WaveLAN-II: A high-performance wireless LAN for the unlicensed band. AT&T Bell Laboratories Technical Journal, pages 118-133.
[4] Kurose, J.F. and Ross, K.W. 2008. Computer Networking: A Top-Down Approach. Pearson International Edition. Addison Wesley.
[5] Lacage, M., Manschaei, M.H., and Turletti, T. 2004. IEEE 802.11 Rate Adaption: A Practical Approach. Modeling, Analysis and Simulation of Wireless and Mobile Systems.
[6] Pediatric, Wi-Fi, http://en.wikipedia.org/wiki/Wi-Fi, Nov. 2011
[7] Wikipedia, Wireless network, http://en.wikipedia.org/wiki/Wireless_network, Nov. 2011
[8] Xiuchao, W. 2009. Simulate 802.11b Channel within NS2. Technical Report. School of Computing, National University of Singapore.
[9] Mythili Vutukuru, Hari Balakrishnan, and Kyle Jamieson. 2009. Cross-layer wireless bit rate adaptation. In Proceedings of the ACM SIGCOMM 2009 conference on Data communication (SIGCOMM '09). ACM, New York, NY, USA, 3-14.
[10] J. F. Wu, S. C. Chan, A. L. Liu and H. C. Wu, "An efficient rate allocation algorithm for transmission and storage of compressed biomedical signals in wireless health monitoring systems," 2016 IEEE Region 10 Conference (TENCON), Singapore, 2016, pp. 2694-2697.
[11] W. Zhang, Y. Wen, Z. Chen and A. Khisti, "QoE-Driven Cache Management for HTTP Adaptive Bit Rate Streaming Over Wireless Networks," in IEEE Transactions on Multimedia, vol. 15, no. 6, pp. 1431-1445, Oct. 2013.

Awni Hammouri received the B.S. degree in Computer Science from Yarmouk University, Jordan in 1985. M.S. and Ph.D. degrees in Computer Science from Illinois Institute of Technology, Chicago, Illinois, U.S.A in 1989 and 1994 respectively. He is now with Mutah University, Jordan.