Performance Analysis of Multichannel Radio Link Control in MIMO Systems

Jun Li[1], Yifeng Zhou[1], Yuanyuan Liu[2,⋆], and Louise Lamont[1]

[1] Communications Research Centre Canada, Ottawa, ON, Canada K2H 8S2
`{jun.li,yifeng.zhou,louise.lamont}@crc.gc.ca`
[2] School of Mathematics, Central South University, Changsha, China 410075
`liuyy@csu.edu.cn`

Abstract. With rapid advances in wireless communications, multiple-input multiple-output (MIMO) antennas technology has been integrated into next-generation wireless communication standards. In this paper, we introduce a MIMO system model, propose a multichannel radio link control protocol and a dynamic channel scheduling policy. We then conduct a performance study on the multichannel link control protocol with two different scheduling policies (*i.e.*, dynamic and static scheduling) using simulations. Simulation results show that the dynamic scheduling outperforms the static scheduling. It is observed that the average packet delay with the dynamic scheduling increases with the average error rate of parallel channels, but decreases with the variance in the error rates of parallel channels. More interestingly, the number of parallel channels has only an insignificant impact on the average packet delay, when the dynamic scheduling is applied in MIMO systems, from which we confirm that the use of parallel channels is a favorable option for packet data networking in the point of view of the link-layer performance.

Keywords: Mobile communications, MIMO techniques, flow and error control, resource allocation and management, performance modeling and analysis, packet delay.

1 Introduction

With rapid advances in wireless communications, multiple-input multiple-output antennas (MIMO) technology [19] has been adopted for next-generation (*i.e.*, 4G) wireless or mobile communication standards, such as high-speed downlink packet access (HSDPA) [2], IEEE 802.16 (WiMax) [1], and 3GPP Long Term Evolution (LTE) [3], to increase data transmission rate. Since an automatic-repeat-request (ARQ) scheme (*i.e.*, one of the following three classical ARQ schemes: stop-and-wait ARQ (SW-ARQ), go-back-N ARQ (GBN-ARQ), and selective-repeat ARQ (SR-ARQ) achieves reliable transmission of packets over intrinsically unreliable wireless links, ARQ-based radio link control has been

---

[⋆] Corresponding author.

D. Simplot-Ryl et al. (Eds.): ADHOCNETS 2011, LNICST 89, pp. 106–116, 2012.

extensively used in current-generation (*i.e.*, 3G) wireless networks, such as Universal Mobile Telecommunications System (UMTS) [4] and CDMA2000, with the aim at the provisioning of data services. Moreover, it has been reported that these traditional ARQ protocols, which have been developed for single-channel communications, can be generalized to achieve reliable packet transmission over multiple channels [10,13,17,21]. As a result, multichannel ARQ has become an integral part in the radio link control sub-layer of 4G wireless communication standards for high-speed multimedia services [8,11].

Several studies on multichannel ARQ protocols have been reported in the literature. System throughput performance in multichannel ARQ protocols was studied in [7,12,20], which are not directly related to the performance metric studied in this paper. Chang and Yang [5] analyzed the average packet delay for the three classical ARQ protocols over multiple identical channels (*i.e.*, all channels have the same transmission rate and the same error rate). Fujii and Hayashida and Komatu [9] derived the probability distribution function of the packet delay for GBN-ARQ over multiple channels that have the same transmission rate but possibly different error rates. Ding [6] considered ARQ protocols for parallel channels that possibly have both different transmission rates and different error rates, and derived approximate expressions of the mean packet delay for them. Unfortunately, it was reported that these approximation results can substantially deviate from the true values as the error rates become relatively large [6]. The resequencing issue in multichannel ARQ protocols was addressed by Shacham and Chin [18], and recently by Li and Zhao [14], who also studied the packet delay distribution function for SW-ARQ over multiple channels by using an end-to-end analytical approach [15].

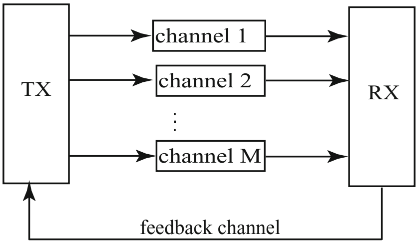
Thanks to studies (*e.g.*, [5,7,20]) on the system throughput, multichannel SR-ARQ has been shown to be the most efficient in terms of the throughput performance among these multichannel ARQ protocols. In comparison, we have a lack of understanding of the packet delay performance of multichannel SR-ARQ. In this paper, we propose a SR-ARQ based link control protocol for MIMO and systematically evaluate the average packet delay performance of the multichannel radio link control protocol with either dynamic or static channel scheduling. We first introduce a MIMO system model, where a transmitter-receiver pair connected by a generic number of forward channels is considered. The multichannel radio link control sub-layer of the MIMO system is composed of two components: the SR-ARQ based protocol for MIMO (SABP) and a packet-to-channel scheduling policy. Under the saturated traffic condition (*i.e.*, packets are always supplied at the transmitter), delay of a packet is measured by the duration between the instant at which the packet is transmitted for the first time and the time it departs from the resequencing queue at the receiver. Using simulations, we investigate the performance of the average packet delay for SABP and the impact of different channel scheduling policies on the average packet delay performance.

The main contributions of this paper are introduction of a MIMO system model and a multichannel radio link control protocol, and the performance evaluation of the multichannel radio link control protocol under static and dynamic channel scheduling policies. Simulation results show that the dynamic scheduling always outperforms the static scheduling. With the dynamic scheduling policy, the average packet delay increases with the average of error rates of the parallel channels, but decreases with the variance in the error rates; with the static scheduling policy, the average packet delay increases with either the average error rate or the variance in the error rates. In addition, if the average error rate among parallel channels remains fixed, the number of parallel channels has an insignificant impact on the average packet delay when the dynamic scheduling policy is applied. However, the average packet delay is severely affected by the number of parallel channels when the static scheduling policy is used.

The rest of this paper is organized as follows. Section 2 describes a MIMO system model. A multichannel radio link control protocol and two channel scheduling policies are introduced in Section 3. Simulation results for the average packet delay are presented and discussed in Section 4, followed by the final section concluding this study.

2 MIMO System Model

In this section, we describe a MIMO system model, where a multichannel radio link control protocol (to be elaborated in Section 3) operates.



TX and RX denote transmitter and receiver, respectively.

Fig. 1. MIMO System Model

A MIMO system consists a transmitter and a receiver. The transmitter-receiver pair communicates data packets for one communication session (*e.g.*, a video file transfer). As illustrated in Fig. 1, the forward link from the transmitter to the receiver consists of M ($M \geq 2$) channels that transmit data packets simultaneously with the multiple antennas equipped in the transmitter and the receiver. Each of the channels is identified with channel i for $i = 1, 2, \cdots, M$, and each channel i is characterized by a data transmission rate and a packet error rate p_i. (The transmission rate of a channel is measured by the maximum number of bytes of data that can be transmitted over that channel during a specified time period; the packet error rate of a channel characterizes the packet loss property of the channel when transmitting packets.) We assume that the packet loss property of a channel is time-invariant, which means that the error rate p_i for channel i is a real number in $(0, 1)$ representing the probability that a packet transmitted over the channel is erroneously received or simply lost. Packet errors that occur in different channels are assumed to be independent. In addition, a high-rate cyclic redundancy check (CRC) error-detection code and a feedback channel are provided in the system. We assume that an erroneous packet can always be detected and that the feedback channel is error-free for transmitting acknowledgement frames.

Each packet to be transmitted is identified by a unique integer number, referred to as the sequence number. We assume that the transmitter has a buffer, referred to as the transmission queue, where there are always packets waiting for transmission. That is, an infinite number of packets are waiting in the transmission queue for first-in-first-out transmission and retransmission with respect to their sequence numbers. Another buffer, referred to as the resequencing queue, is provided at the receiver to temporarily store unqualified packets. An unqualified packet is referred to as a correctly received packet with the property that at least one packet with a smaller sequence number has not been correctly received. All channels have the same transmission rate, and the M channels are time-slotted with one unit (or slot) equal to the transmission time of a packet over a channel. Therefore, the transmission rate of each channel is one packet per slot. All packets, when transmitted from the transmitter to the receiver, have a fixed round trip time (RTT) equal to $(\tau - 1)$ slots, which is assumed to be an even number of slots. A packet experiences the same propagation delay in forward and feedback channels, which is $(\tau - 1)/2$ slots. The transmitter sends multiple packets at a time, one per channel. All channels share the same set of sequence numbers of the packets in packet-to-channel scheduling (to be discussed in Section 3.2). The M channels have possibly different error rates. That is, the packet error rate p_i of channel i, for $i = 1, \cdots, M$, might be different from the packet error rate p_j of channel j when $i \neq j$. By assuming that a perfect channel estimation is accomplished, the transmitter has knowledge about the condition (*e.g.*, the error rate) of each channel, according to which a dynamic scheduling (to be discussed in Section 3.2) can be implemented at the transmitter. A multichannel radio link control protocol (MRLC), which will be detailed in the next section, is used for traffic flow and packet error control.

3 Multichannel Radio Link Control (MRLC)

In this section, we elaborate a SR-ARQ based radio link control protocol and two different channel scheduling policies for traffic flow and error control in the MIMO system described in Section 2.

3.1 SABP: A SR-ARQ Based Link Control Protocol for MIMO

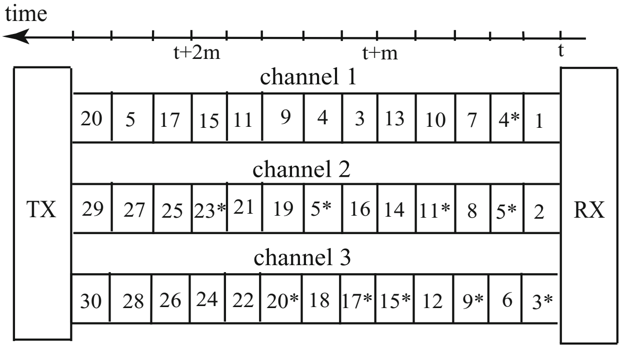
At the beginning of each slot, the transmitter starts transmitting a block of M packets to the receiver and completes transmission at the end of the slot. The receiver receives the block of M packets, which were transmitted in slot t for $t = 0, 1, \cdots$, at the end of slot $t + (\tau - 1)/2$ (see Fig. 2). The packet transmitted over channel i is received erroneously or simply lost with probability p_i. At the receiver, an erroneously received or lost packet corresponds to a negative acknowledgement (NACK), while a correctly received packet corresponds to a positive acknowledgement (ACK). Then the receiver sends an acknowledgement frame containing exactly M acknowledgements (ACKs/NACKs) corresponding to the most recently received block of M packets, to the transmitter. We assume that transmission of the acknowledgement frame takes no time at the receiver and is completed at the end of slot $t + (\tau - 1)/2$.

After sending the acknowledgement frame, the receiver discards erroneously received packets, delivers the qualified packets, and stores the unqualified packets in the resequencing buffer. A qualified packet is a correctly received packet with a sequence number such that all packets with a smaller sequence number have been correctly received. The transmitter receives the acknowledgement frame, which is associated with the block of M packets transmitted at slot t, at the end of slot $t + \tau - 1$. It checks each acknowledgement in the acknowledgement frame, and prepares the next block of M packets to transmit at slot $t + \tau$ according to the following rule: If there is no NACK in the acknowledgement frame, the next block to transmit is composed of M new packets; if the acknowledgement frame contains one or more, for example k, NACKs, the next block of M packets consist of those k old packets, which are negatively acknowledged by the receiver, and $M - k$ new packets (see Fig. 2). Meanwhile, the transmitter removes these positively acknowledged packets from the transmission queue. These selected M packets are to be transmitted in slot $t + \tau$ according to one of the following packet-to-channel scheduling policies.

3.2 Packet-To-Channel Scheduling

To simultaneously transmit a block of M packets over the M channels in a slot, either one of the following two packet-to-channel scheduling policies: dynamic scheduling and static scheduling, can be applied. The dynamic scheduling is illustrated in Fig. 2, where $p_1 \leq p_2 \leq p_3$ is assumed, and works as follows. The best channel (*i.e.*, a channel with the smallest error rate) is assigned to the packet associated with the smallest sequence number in the block; the second best channel is assigned to the packet associated with the second smallest sequence number; and so forth.

The counterpart of the dynamic scheduling is the static scheduling, which is illustrated in Fig. 3 with $p_1 \leq p_2 \leq p_3$. With the static scheduling policy, an old packet (*i.e.*, a packet to be retransmitted) is always assigned to the same channel for retransmission as the originally assigned one, while a new packet (*i.e.*, a packet to be transmitted for the first time) is assigned to a uniformly chosen channel among those available for transmitting new packets. As will be shown from simulation results presented in the next section, the dynamic scheduling achieves a better protocol performance than the static scheduling.



x* denotes transmission error of packet x;
integer numbers in the boxes are sequence numbers of packets.

Fig. 2. Dynamic Channel Scheduling ($M = 3; \tau = 5$)

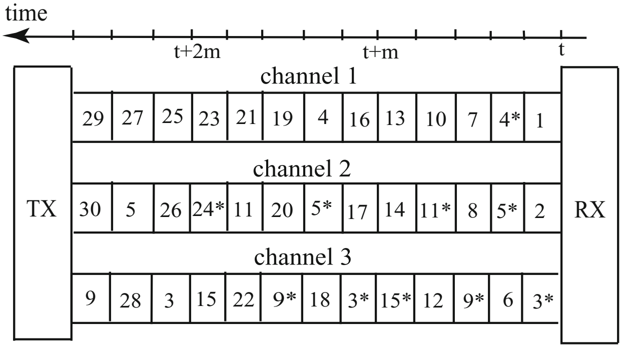


Fig. 3. Static Channel Scheduling ($M = 3; \tau = 5$)

4 Performance Evaluation of MRLC

In this section, we conduct a simulation study to evaluate the performance of the multichannel radio link control protocol with either dynamic or static channel scheduling. The performance metric that we consider is the average packet delay. The delay of a packet is defined as the amount of time (*i.e.*, the number of slots) between the instant at which the packet is transmitted for the first time and the instant at which it departs from the resequencing queue in the receiver. We investigate the impact of the channel scheduling policies and the protocol parameters on the average packet delay performance through simulations.

4.1 Simulation Environment

We use the SimPy simulator [16], which is an object-oriented, process-based discrete-event simulation platform based on the standard programming language

Python. SABP is at first implemented with SimPy. Then two individual processes, one considered as the transmitter and the other as the receiver, form an M-channel MIMO system. Each process independently operates an object of SABP. The transmitter continuously sends data packets and receives acknowledgement frames, and the receiver receives data packets and sends out acknowledgement frames. Data packets are transmitted over M parallel channels, while acknowledgement frames are transmitted via a separate feedback channel with no errors.

In the following simulation analysis, the round trip time of a packet is 4 slots, or $\tau = 5$. We use Δ_i to represent the ratio of p_{i+1} to p_i for $i = 1, \cdots, M-1$, *i.e.*,

$$\Delta_i = \frac{p_{i+1}}{p_i}, \ i = 1, \cdots, M-1. \tag{1}$$

It is clear that, the larger the value of Δ_i, the greater the difference between the error rates of channels i and $i+1$. In addition, we let $\Delta = \Delta_1 = \cdots = \Delta_{M-1}$. Then, the triad (M, Δ, p) will uniquely determine the error rate sequence $(p_1, p_2, \cdots, p_M)$.

4.2 Simulation Results

We plot the simulation results of the average packet delay for SABP with the dynamic and static scheduling in Fig. 4, Fig. 5 and Fig. 6. An important observation in these plots is that, compared with the static scheduling, the dynamic scheduling improves the packet delay performance in the MIMO system. For instance, for $M = 16$, the average packet delay performance can be improved as much as up to 70% when the packet scheduling policy changes from the static scheduling to the dynamic scheduling. When $\Delta = 1.5$, the average packet delay with the dynamic scheduling can be only one third of that with the static scheduling.

The average packet delay is plotted in Fig. 4 for $\Delta = 1.2$, $p = 0.25$, and M varying from 2 to 16. As we expect, the difference of the average delay between the two scheduling policies becomes larger with the increase of M. Meanwhile, as M increases, the average packet delay with the dynamic scheduling slightly increases at first and then slightly decreases. This shows that, under the saturated traffic condition, the overall impact of the number of parallel channels on the packet delay performance is insignificant when the dynamic scheduling is applied. Since the average packet delay approaches a constant limit as the number of channels increases, the use of parallel channels will be a favorable option for high-data-rate MIMO system with SABP for error control. It is noted that, for the multichannel protocol under non-saturated traffic conditions, packet end-to-end delay includes another delay component, the packet waiting time at the transmitter, in addition to the packet delay defined in this study. Under a non-saturated traffic condition, it is clear that the increase of the transmission rate mainly results in the reduction of the packet waiting time at the transmitter, and hence the packet end-to-end delay. So the above observation corroborates the fact that the increase of the number of parallel channels leads to the increase of the transmission rate but the decrease of the overall packet delay for MIMO systems with non-saturated traffic.

In Fig. 5, we plot the average packet delay when $M = 8$, $\varDelta = 1.2$, and p varying from 0.05 to 0.45. The average packet delay increases as p does, while the increasing rate with the dynamic scheduling is smaller than that with the static scheduling. The average packet delay is shown in Fig. 6 when $M = 8$, $p = 0.25$, and $\varDelta$ varying from 1.1 to 1.7. As $\varDelta$ increases, the average packet delay decreases when the dynamic scheduling is applied, but it increases when the static scheduling is used. For example, when $\varDelta$ increases from 1.1 to 1.5, the average packet delay with the dynamic scheduling decreases almost 50%, but the average packet delay with the static scheduling increases 100%. This is because the greater the variance in the error rates, the smaller the error rates of the first few channels. (For instance, in Fig. 6, the error rates of channels 1 to 4 when $\varDelta = 1.2$ are smaller than the corresponding ones when $\varDelta = 1.1$.) Intuitively, the packets transmitted over the first few channels have a larger probability of being correctly received (and delivered to the upper layer). This results in a smaller possibility for the other packets to be queued in the resequencing buffer. Therefore, the average waiting time of a packet queued in the resequencing queue is reduced, and so is the total average packet delay.

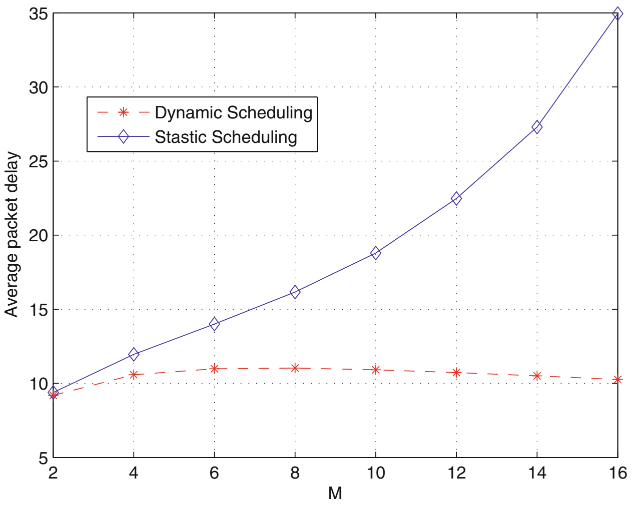


Fig. 4. Average Packet Delay vs. M ($\varDelta = 1.2$, $p = 0.25$)

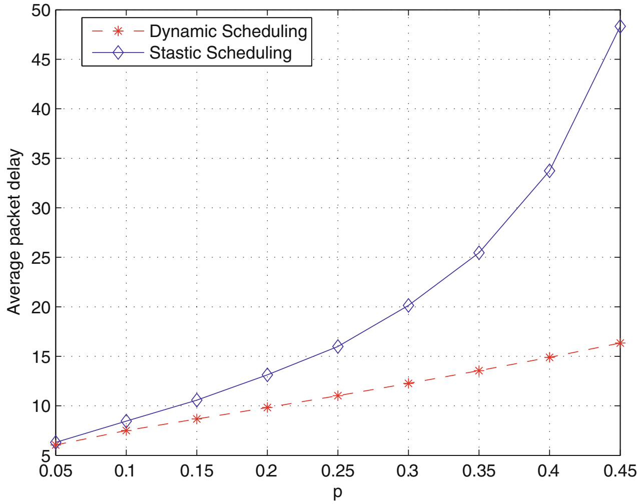


Fig. 5. Average Packet Delay vs. p $(\varDelta = 1.2, M = 8)$

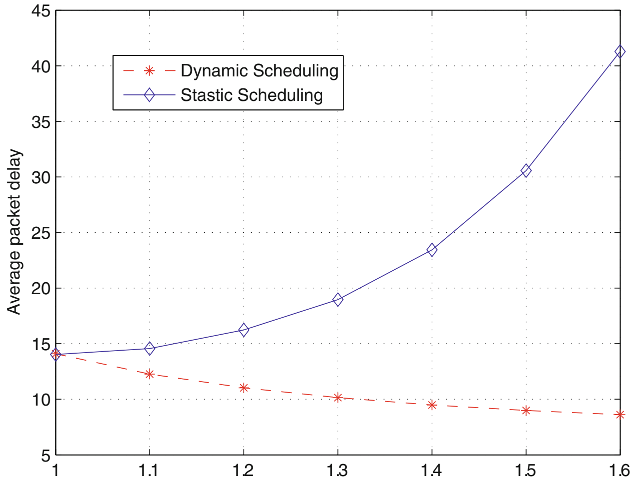


Fig. 6. Average Packet Delay vs. $\varDelta$ $(M = 8,\ p = 0.25)$

5 Conclusion

In this paper, we introduced a MIMO system model, where our proposed multichannel radio link control protocol and a channel scheduling policy operate. We performed a simulation analysis of the average packet delay for the multichannel link control protocol with two different channel scheduling policies: dynamic and static scheduling. From simulation results, we concluded that the dynamic scheduling always achieves a better packet delay performance than the static scheduling. The average packet delay with the dynamic scheduling increases with the average error rate of all channels, but decreases with the variance in the error rates of the parallel channels. More interestingly, we observed that the number of parallel channels has only an insignificant impact on the average packet delay, when the dynamic scheduling is applied in the MIMO system, and hence the use of parallel channels is a favorable option for multichannel packet data networking.

Acknowledgement. This work was supported by Defence Research and Development Canada (DRDC) (under the project entitled "Self-Healing Networked Control Systems For Enhanced Reliability And Safety Of Multivehicle Missions").

References

1. Part 16: Air Interface for Fixed Broadband Wireless Access Systems, IEEE Standard for Local and Metropolitan Area Networks, 802.16 (2004)
2. 3GPP TS 25.321 V7.6.0, Medium Access Control (MAC) Protocol Specification (Release 7), 3GPP Technical Specification Group Radio Access Network (2007)
3. 3GPP TS 36.201, Evolved Universal Terrestrial Radio Access (E-UTRA): Long Term Evolution (LTE) Physical Layer, 3GPP Technical Specification Group Radio Access Network, source (2008), `http://www.3gpp.org/ftp/Specs/html-info/36-series.htm`
4. 3GPP TS 25.322 V5.3.0, Radio link control (RLC) protocol specification (release 5). 3GPP Technical Specification, source (2002), `http://www.3gpp.org`
5. Chang, J.-F., Yang, T.-H.: Multichannel ARQ Protocols. IEEE Transactions on Communications 41(4), 592–598 (1993)
6. Ding, Z.: ARQ Techniques for MIMO Communication Systems. Ph.D. thesis, Department of Electrical and Computer Engineering, Brigham Yong University Provo (2006)
7. Ding, Z., Rice, M.: ARQ Error Control for Parallel Multichannel Communications. IEEE Transactions on Wireless Communications 5(11), 3039–3044 (2006)
8. Forkel, I., Klenner, H., Kemper, A.: High Speed Downlink Packet Access (HSDPA) – Enhanced Data Rates for UMTS Evolution. Computer Networks: The International Journal of Computer and Telecommunications Networking 49(3), 325–340 (2005)
9. Fujii, S., Hayashida, Y., Komatu, M.: Exact Analysis of Delay Performance of Go-Back-N ARQ Scheme over Multiple Parallel Channels. Electronics and Communications in Japan, Part 1 84(9), 27–41 (2001)

10. Ghosh, A., Classon, B., Cudak, M., Jalloul, L.A.: Multi channel stop and wait ARQ communication method and apparatus. United States Patent No. 7,065,068, `http://www.patentstorm.us/patents/7065068.html`
11. Ghosh, A., Wolter, D.R., Andrews, J.G., Chen, R.: Broadband Wireless Access with WiMax/8O2.16: Current Performance Benchmarks and Future Potential. IEEE Communication Magazine 43(2), 129–136 (2005)
12. Hu, T., Afshartous, D., Young, G.: Parallel Stop and Wait ARQ in UMTS - Peformance and Modeling. In: Proceedings of the 2004 World Wireless Congress, San Francisco, CA (2004)
13. Jacobus, J.C.: Multi-channel automatic retransmission query (ARQ) method. United States Patent No. 6,021,124, `http://www.patentstorm.us/patents/6021124.html`
14. Li, J., Zhao, Y.Q.: Resequencing Analysis of Stop-and-Wait ARQ for Parallel Multichannel Communications. IEEE/ACM Transactions on Networking 17(3), 817–830 (2009)
15. Li, J., Zhao, Y.Q.: Packet Delay Analysis for Multichannel Communication Systems with MSW-ARQ. Performance Evaluation 66(7), 380–394 (2009)
16. Matloff, N.: Introduction to Discrete-Event Simulation and the SimPy Language, `http://heather.cs.ucdavis.edu/~matloff/156/PLN/DESimIntro.pdf`
17. Redi, J., Watson, B., Ramanathan, R., Basu, P., Tchakountio, F., Girone, M., Steenstrup, M.: Design and Implementation of a MIMO MAC protocol for ad hoc Networking. In: Proceedings of SPIE 2006, Orlando, FL., vol. 6248, 624802.1–624802.12, Florida, USA (2006)
18. Shacham, N., Shin, B.C.: A Selective-Repeat-ARQ Protocol for Parallel Channels and Its Resequencing Analysis. IEEE Transactions on Communications 40(4), 773–782 (1992)
19. Winters, J.H.: On the Capacity of Radio Communication Systems with Diversity in a Rayleigh Fading Environment. IEEE Journal on Selected Areas in Communications SAC-5(5), 871–878 (1987)
20. Wu, W.-C., Vassiliadis, S., Chung, T.-Y.: Performance Analysis of Multi-Channel ARQ Protocols. In: Proceedings of the 36th Midwest Symposium on Circuits and Systems, vol. 2, pp. 1328–1331 (1993)
21. Zheng, H., Lozano, A., Haleem, M.: Multiple ARQ Processes for MIMO Systems. EURASIP Journal on Applied Signal Processing 5, 772–782 (2004)