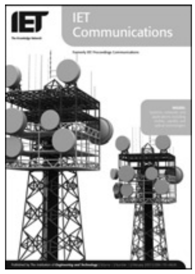
Published in IET Communications
Received on 28th October 2007
Revised on 1st February 2008
doi: 10.1049/iet-com:20070538



ISSN 1751-8628

Distributed splitting-tree-based medium access control protocol for multiaccess wireless networks with multipacket reception

J.-S. Liu[1] *C.-H.R. Lin*[2]

[1]*Department of Computer Science and Information Engineering, Providence University, Taiwan, Republic of China*
[2]*Department of Computer Science and Engineering, National Sun Yat-Sen University, Taiwan, Republic of China*
E-mail: chhliu@pu.edu.tw

Abstract: The conventional medium access control (MAC) protocols assume that only one packet can be received at a given time. However, with the advent of spread spectrum, antenna arrays or sophisticated signal processing techniques, it is possible to achieve multipacket reception (MPR) in wireless communication networks. A distributed splitting-tree-based MAC protocol that can exploit the MPR capability in the networks is proposed. For the MPR MAC protocol, a closed-form expression of the system throughput is derived based on a Markov chain model. The experimental results show that the MPR protocol can considerably increase the spectrum efficiency compared with the splitting-tree algorithm with its conventional collision resolution method.

1 Introduction

In multiaccess wireless networks, the conventional assumptions on the reception capability of the common channel are that (1) a packet is successfully received if and only if there are no concurrent transmissions, and (2) its transmission failure is a result of frame collision and is not caused by medium error. Based on the assumptions, MAC protocols are sought to coordinate the transmissions of all stations for the efficient utilisation of the limited channel reception capability under error-free channels. For example, ALOHA [1], splitting-tree algorithm [2] and first-come first-serve algorithm [3] have been proposed and their performance studied. However, with the development of spread spectrum multiple access, space–time coding and new signal processing techniques, the correct reception of one or more packets in the presence of other simultaneous transmission becomes possible, which breaks the first assumption. To this end, the concept of multipacket reception (MPR) was introduced in [4]; nevertheless, the research on MPR is still limited, and the medium access control (MAC) protocols based on it usually assume the existence of a central coordinator to schedule transmission [5, 6]. To get rid of the centralisation overhead, recently some MPR MAC protocols have been proposed in a distributed manner [7, 8]. However, these distributed protocols still adopt the second assumption to complete themselves based on an ideal MPR model of capability M. That is, in these protocols, packets can be received correctly whenever the number of simultaneous transmissions is no more than M. In contrast, when there are more than M transmissions, no packet can be decoded.

Apparently, the ideal channel model adopted in above may be theoretically tractable but could be unrealistic even under the most recent wireless physical layer (PHY). Thus, in this work, we consider a more general MPR model that can take into account the fact that given M packets transmitted simultaneously, there may be only $k < M$ packets to be correctly received due to unpredictable variations in the wireless channel [4]. With this model, we extend the splitting-tree algorithm in [2] to be a distributed MPR MAC protocol and calculate its system throughput under different offered traffic loads and different MPR capabilities. In addition, we also analyse the throughput performance of the conventional tree algorithm in the MPR model as a special case of the protocol. These analytical results are further verified with that of simulation experiments. Both results show that the MPR protocol can considerably increase the spectrum efficiency compared with the tree algorithm with its collision resolution method, and the fact is confirmed by their consistent values.

The paper is organised as follows. In Section 2, we briefly summarise the MPR model and introduce the proposed MPR MAC protocol. Following that in Section 3, we calculate the system throughput for the protocol and the splitting-tree algorithm. To verify the correctness of our analysis, the theoretical results are examined with the experiments in Section 4. And finally, conclusions are drawn in Section 5.

2 System modelling

In this work, we consider a communication network with N competing stations operated under a common wireless channel with MPR of capability M, in which each station independently generates equal-sized packets with probability q.

2.1 MPR channel

As shown in [4], the wireless channel is characterised by the probability of having k success in a slot when there are n concurrent transmissions. That is

$$C_{n,k} = P[k \text{ packets are correctly received} | n \text{ are transmitted}], \quad 1 \le n \le M, \quad 0 \le k \le n \tag{1}$$

The reception matrix of the channel in a network with MPR of capability M is then defined as

$$\boldsymbol{C} = \begin{pmatrix} C_{1,0} & C_{1,1} & & & \\ C_{2,0} & C_{2,1} & C_{2,2} & & \\ \vdots & \vdots & \vdots & & \\ C_{M,0} & C_{M,1} & C_{M,2} & \cdots & C_{M,M} \end{pmatrix} \tag{2}$$

As shown above, the matrix in fact represents a general form for the reception probabilities and can apply to various MPR-capable systems. Thus, instead of using a certain communication system and its particular error characteristic, in this work, we consider a general wireless network with the packet success probability P_s, in which each element of the reception matrix $\boldsymbol{C}$ is represented by

$$C_{n,k} = B(k, n, P_s) \tag{3}$$

Here $B(k, n, P_s)$ denotes the probability mass at the value k of a Binomial random variable with total n trials and a success probability P_s. That is

$$B(k, n, P_s) = \binom{n}{k} P_s^k \cdot (1 - P_s)^{n-k} \tag{4}$$

Defined as above, the matrix $\boldsymbol{C}$ can also accommodate the ideal MPR model mentioned previously by setting $P_s = 1$, as

$$\boldsymbol{C}_{\text{ideal}} = \begin{pmatrix} 0 & 1 & & & \\ 0 & 0 & 1 & & \\ \vdots & \vdots & \vdots & & \\ 0 & 0 & 0 & \cdots & 1 \end{pmatrix} \tag{5}$$

The special case is also verified in our experiments in Section 4.

2.2 The Splitting-tree-based MPR MAC protocol

The distributed MAC protocol is conducted by the splitting-tree algorithm in [2] with buffer size of 1 and operated under a common MPR wireless channel modelled in above. Thus, as the conventional tree algorithm, this MAC let each station maintain a conceptual stack. At the end of each slot n, it determines its position in the stack, l_n, according to the following procedure.

1. When $l_n = 0$, this packet will be transmitted in slot n; when $l_n > 0$, the packet is not transmitted.

2. When $l_n = 0$ and a collision is reported in slot n, then $l_{n+1} = 1$ with probability 1/2 and $l_{n+1} = 0$ with probability 1/2.

3. When $l_n = 0$ and a failure is reported for the competing station, then $l_{n+1} = 0$.

4. When $l_n = 0$ and a failure is reported not for the competing station, then $l_{n+1} = 0$ and the station becomes inactive.

5. When $l_n = 0$ and a capture is reported, then $l_{n+1} = l_n$ and the active station becomes inactive.

6. When $l_n > 0$ and a capture is reported, then $l_{n+1} = l_n - 1$.

7. When $l_n > 0$ and slot n is reported idle, then $l_{n+1} = l_n - 1$.

8. When $l_n > 0$ and a collision is reported, then $l_{n+1} = l_n + 1$.

9. When completing a collision resolution period (CRP), each station becomes active again.

As shown above, the protocol procedure is quite simple. However, its implications are worth noting. First, the protocol lets each station maintain its own stack to cooperatively resolve the medium access contention. It is the distributed solution that a splitting-tree-based MAC protocol may outperform the centralised ones in [5, 6], for its lower

doi: 10.1049/iet-com:20070538

complexity to implement. Second, as mentioned previously, the MPR MAC protocol considers that $k \le M$ stations may not all succeed in the access contention even though the capability of MPR is M. If so, the protocol will arrange the set of failing stations to transmit again. It is the point that makes the protocol different from the conventional splitting-tree algorithm [2] and other MPR MAC protocols [7, 8]. The difference is so conducted because the assumption made by the splitting-tree algorithm that packet collisions can only be resolved by splitting of stations is not always necessary or even sensible, as indicated in [6]. In fact, the MPR capability opens new options for resolving a collision beyond that of being all successful or otherwise all failed. That is, when two or more packets are simultaneously transmitted and not all successfully received, instead of splitting, it is more sensible to enable the same set of stations again because with a high possibility the failure is caused by instantaneous fading channel conditions rather than packet collisions. This is reflected in (3) and (4) of the procedure. The failure signal lets the set of unsuccessful stations retransmit their packets, and the signal is issued by the receiver when it detects $k \le M$ simultaneous transmissions but fails to correctly decode all of them. Assuming that a receiver can estimate the number of competing transmitters with a method such as that given in [9], it can issue the failure signal based on this estimation, in addition to the collision and capture signals originally given in the splitting-tree algorithm.

To be clear, in Fig. 1, we show an example of three stations, namely, A, B and C, carrying out the MPR MAC protocol for their wireless medium access. At the beginning of CRP, all the three stations have packets to send and their l_1s are all of 0. Suppose the MPR capability is 2 (i.e. $M = 2$), they will collide with each other at the first time slot, SL1. Such a collision causes a tree split as shown by the stack implementation that splits the initial state of 1 into left (L) and right (R). At the end of SL1, the stations randomly decide their l_2s. Unfortunately, all of them choose 0 again, implying another collision at SL2. At SL3, only A and B have l_3s of 0 for transmission. However, even when $M = 2$, there may be $k < 2$ packets to be correctly received due to channel error. In this case, B fails while A does not. With our protocol, instead of splitting, we enable the set of unsuccessful stations, now B only, to transmit again. That is, at SL4, B remains $l_4 = 0$ in active state for further transmission while A becomes inactive to stop its access. As shown by the stack implementation, B is now the only one on the top to have the first transmission priority. After receiving a capture signal, B becomes inactive and C decreases its position value as $l_5 = 0$, saying that C is on the stack top at SL5. However, at the end of SL5, a failure is reported. Thus C continues to transmit at SL6 and then succeeds at that slot. Finally, at SL7, all stations are inactive and remove the last stack element, which indicates the end of CRP.

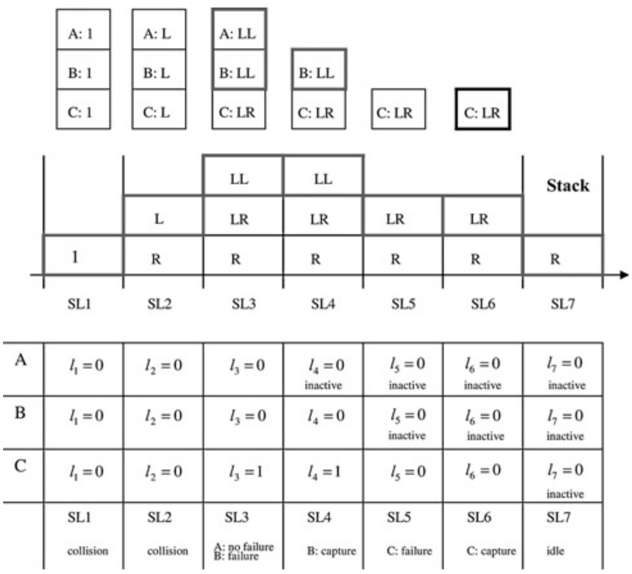


Figure 1 *Example for the MPR MAC protocol*

3 Performance analysis

In this section, we analyse the MPR MAC protocol with a Markov chain modelling scheme. To start with, we divides the time axis into CRPs as exemplified in Fig. 2, where the ith CRP is dedicated to the transmission of the packets generated in the $(i-1)$th CRP. With the structure, the state of Markov chain is characterised by the number of competing stations at the beginning of the nth CRP, and denoted by X_n. Similarly, the time slots required to resolve the CRP, that is, the CRP's length, is denoted by Y_n. Given this model, it is easy to see that X_{n+1} depends only on Y_n and the packet arrival probability. Consequently, X_{n+1} depends only on X_n, with the transition probability represented by

$$
\begin{aligned}
P[X_{n+1} = m_{n+1} | X_n = m_n] &= \sum_{k=1}^{\infty} P[X_{n+1} = m_{n+1} | Y_n = k, X_n = m_n] \\
&\quad \cdot P[Y_n = k | X_n = m_n] \\
&= \sum_{k=1}^{\infty} P[X_{n+1} = m_{n+1} | Y_n = k] \\
&\quad \cdot P[Y_n = k | X_n = m_n]
\end{aligned} \tag{6}
$$

To obtain the probability, the two items in the last equation of (6) should be resolved. The first item, that is, $P[X_{n+1} = m_{n+1} | Y_n = k]$, can be obtained by considering that for each slot, the packet arrival probability of a station has the Bernoulli distribution. Let q be the arrival probability. Then,

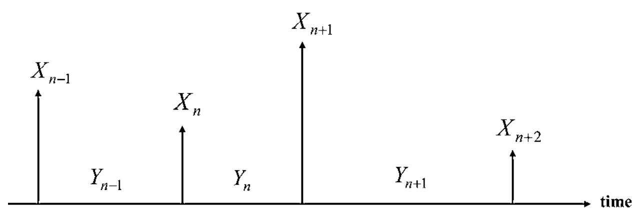


Figure 2 *Transmission structure for the MPR MAC protocol*

IET Commun., 2008, Vol. 2, No. 9, pp. 1170–1176
doi: 10.1049/iet-com:20070538

the probability of no arrival after k slots is $(1-q)^k$ for each station. Thus, we have

$$P[X_{n+1}=m|Y_n=k] = \binom{N}{m}\cdot(1-(1-q)^k)^m\cdot((1-q)^k)^{N-m} \tag{7}$$

Given the above, we now proceed to obtain the second item in (6), that is, $P[Y_n=k|X_n=m_n]$, with the consideration of the following two cases.

1. $X_n=m>M$: In this case, the MPR MAC protocol resolves a collision by splitting the competing stations into two distinct sets, based on the conventional splitting-tree algorithm. As one can expect, recursive formulas are usually possible for solving tree problems. In this case, it also holds. For doing so, we let NL denote the random variable for the number of stations in the left subtree and TL be that for the number of slots expended. Similarly, NR and TR denote the random variables for the right subtree. In addition, we let $\phi(m,k)$ be $P[Y_n=k|X_n=m]$ to simplify the notation. Given that we can derive $\phi(m,k)$ with a recursive form as follows

$$\begin{aligned}
\phi(m,k) &= P[Y_n=k|X_n=m]\\
&= \sum_{nL=0}^{m} P[Y_n=k, NL=nL|X_n=m]\\
&= \sum_{nL=0}^{m} P[NL=nL|X_n=m]\\
&\quad \cdot P[Y_n=k|NL=nL, Xn=m]\\
&= \sum_{nL=0}^{m}\binom{m}{nL}\left(\frac{1}{2}\right)^m P[Y_n=k|NL=nL, X_n=m]\\
&= \sum_{nL=0}^{m}\binom{m}{nL}\left(\frac{1}{2}\right)^m \sum_{tL=1}^{k-2} P[Y_n=k, TL=tL|NL\\
&= nL, X_n=m]\\
&= \sum_{nL=0}^{m}\binom{m}{nL}\left(\frac{1}{2}\right)^m \sum_{tL=1}^{k-2} P[TR=k-tL-1, TL\\
&= tL|NL=nL, NR=m-nL]\\
&= \sum_{nL=0}^{m}\binom{m}{nL}\left(\frac{1}{2}\right)^m \sum_{tL=1}^{k-2} P[TL=tL|NL=nL]\\
&\quad \cdot P[TR=k-tL-1|NR=m-nL]\\
&= \sum_{nL=0}^{m}\binom{m}{nL}\left(\frac{1}{2}\right)^m \sum_{tL=1}^{k-2} \phi(nL,tL)\\
&\quad \cdot \phi(m-nL, k-tL-1)
\end{aligned} \tag{8}$$

2. $X_n=m\le M$: In this case, a station may receive different numbers of transmissions with different probabilities ($C_{m,k}$s). This is represented by the Markov chain model shown in Fig. 3, based on the MPR model given in (2). Alternatively, it can be shown by the following probability transition matrix

$$\boldsymbol{P} = \begin{pmatrix} 1 & & & & \\ C_{1,1} & C_{1,0} & & & \\ C_{2,2} & C_{2,1} & C_{2,0} & & \\ \vdots & \vdots & \vdots & & \\ C_{M,M} & C_{M,M-1} & C_{M,M-2} & \cdots & C_{M,0} \end{pmatrix} \tag{9}$$

Let random variable $T_{i,j}$ denote the first passage time from state i to state j. With the transmission matrix in (9), we can obtain the probability mass function for $T_{i,j}$ by

$$f_{i,j}^{(n)} = P[T_{i,j}] = P[X_n=j, X_{n-1}\ne j, \ldots, X_1\ne j|X_0=i],$$
$$n=1,2,\ldots \tag{10}$$

As shown in [10], the first passage time probabilities can be expressed in a matrix form, $\boldsymbol{F}^{(n)}=\{f_{i,j}^{(n)}\}$, as

$$\boldsymbol{F}^{(1)} = P$$
$$\boldsymbol{F}^{(n)} = P\boldsymbol{F}^{(n-1)} - P\boldsymbol{F}_{\rm d}^{(n-1)} = P\boldsymbol{F}_0^{(n-1)}, \quad n=2,3,\ldots \tag{11}$$

where $\boldsymbol{F}_{\rm d}$ denotes a diagonal matrix formed by the diagonal elements of $\boldsymbol{F}$ and $\boldsymbol{F}_0$ denotes a matrix resulting from setting the diagonal elements of $\boldsymbol{F}$ to zeros. That is, $\boldsymbol{F}=\boldsymbol{F}_{\rm d}+\boldsymbol{F}_0$. Consequently, the desired metric, $\phi(m,k)$, for this case can be obtained by

$$\phi(m,k)(=P[Y_n=k|X_n=m]) = \boldsymbol{F}^{(k)}(m,0) \tag{12}$$

where $\boldsymbol{F}^{(k)}(m,0)$ denotes the $(m,0)$th element of matrix $\boldsymbol{F}^{(k)}$.

3.1 System throughput for the MPR MAC protcol

1. *Steady-state probability*: Let $\pi(m)$ be the steady state probability of $X=m$. That is, $\pi(m)=\lim_{n\to\infty}P[X_n=m]$. For obtaining its value, we first substitute (7), (8) and (12) into (6), which provides each transition probability from

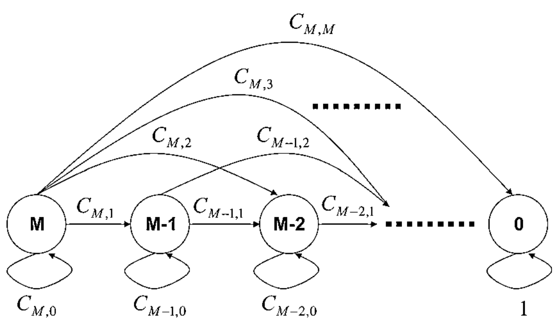


Figure 3 *Markov chain model for the MPR model of capability M*

doi: 10.1049/iet-com:20070538

state m to state n, $0 \leq m, n \leq N$. The probabilities obtained are then collected into a transition matrix, say P_s. Finally, by solving $\pi \cdot P_s = \pi$, the steady-state probability for each state m can be resulted.

2. *Throughput calculation*: In the system, a regenerative cycle is considered as the interval between two consecutive states of $X = 0$. According to the renewal theory, $\pi(m)/\pi(0)$ represents the number of state $X = m$ happened in the cycle. Given that each time the system enters the state of $X = m$, it may expend k time slots to resolve the CRP with the probability of $\phi(m, k)$. Thus, the expectation of CRP for $X = m$ can be obtained by

$$l(m) = \sum_k \phi(m, k) \cdot k \tag{13}$$

Taking the above into account, the system throughput, λ_d, can thus be represented by the average number of packets transmitted in the cycle over the average cycle length. That is

$$\lambda_d = \frac{\sum_{m=0}^{N} \pi(m)/\pi(0) \cdot m}{\sum_{m=0}^{N} \pi(m)/\pi(0) \cdot l(m)} \tag{14}$$

3.2 System throughput for the conventional splitting-tree alogrithm

For comparison, we also calculate the system throughput for the conventional splitting-tree algorithm in [2]. Similarly, instead of using the ideal MPR model with the reasons given previously, we analyse the tree algorithm with the general MPR model as follows

$$\phi(m, k) = \begin{cases} \boldsymbol{F}^{(k)}(m, 0), & m = 1 \\ P(m, 1), & k = 1, 1 < m \leq M \\ \displaystyle\sum_{nL=0}^{m} \binom{m}{nL} \left(\frac{1}{2}\right)^m & \\ \displaystyle\sum_{tL=1}^{k-2} \phi(nL, tL) & \\ \cdot \phi(m - nL, k - tL - 1), & k > 1, m > 1 \end{cases} \tag{15}$$

where nL and tL are the values given previously in (8). As can be seen, the first equation denotes the fact that even given only one packet transmitted in the wireless channel, the receiver may not correctly receive it with a certain probability, and thus, the tree algorithm may require more slots to complete the transmission. The second represents that with the MPR model, there is a probability of $P(m, 1) = C(m, m)$, $m \leq M$, that the bulk of m packets can all be successfully received in a single time slot without retransmission. The third denotes the possibility that a station may fail to decode all the packets received, and the algorithm on it will respond to the situation with the same collision signal broadcasted in the channel, despite whether the number of competing transmissions is under the MPR capability of M or not, and let the corresponding transmitters further split the tree and retransmit their packets latter.

4 Experimental results

In this section, we report on experiments made in order to verify the performance results derived previously. In particular, we compare the throughput performance λ_D of the MPR MAC protocol with that of the splitting-tree algorithm. For doing so, we consider these protocols working under a network of $N = 20$ stations and vary the following three parameters to see their performance differences. The first parameter is total traffic load $\lambda = q \cdot N$, and its value is given by 1, 2, 4 and 8, respectively. The second parameter is packet success probability P_s, and its value ranged from 0.1 to 1 with a step size of 0.1. Among these, the highest value of $P_s = 1$ denotes a perfect channel, in which no packet will be lost whenever the number of simultaneous transmissions is no more than M; that is, it represents the ideal MPR model with C_{ideal} in (5). On the contrary, the lowest value of $P_s = 0.1$ denotes a highly error-prone channel, in which most of the packet transmissions will fail. The third parameter is MPR capability. Now, $M = 5$ and 10 are the values under consideration. With the above, we conduct different experiments, and their results are shown in Fig. 4, in which the four subfigures represent that for $\lambda = 1$, 2, 4 and 8, respectively. Based on these results, we have the following observations.

First, the MPR protocol's λ_D is much higher than that of the conventional splitting-tree algorithm under any combination of M and P_s, except that for $P_s = 1$. This denotes the potential that the protocol can significantly outperform the tree algorithm under usual cases. Second, a higher P_s leads to a higher λ_D in all the experiments. This is expected because the former actually represents a better channel, fewer packet errors and access collisions, and thus better throughputs despite the protocols and Ms involved. Third, the MPR MAC protocol can increases λ_D by increasing M while the splitting-tree algorithm has no such improvements with the same increments of M, except that for high P_ss. For this point, we note that the conventional tree algorithm simply regards transmission failures as the results of collisions. Thus, a low P_s in fact has the same effect as a large number of collisions, which eventually leads to a low λ_D. In the situation, increasing M helps nothing because now channel errors dominate the network. However, as P_s increases, medium access collisions become dominating. In this case, the tree algorithm operates well because it is originally designed for resolving such collisions, and increasing M actually increases its capability of collision resolution. On the contrary, the proposed MAC protocol can distinguish the channel errors from medium access collisions and operate differently on them to improve the system performance for all the P_ss.

Finally, we pay our attention to the results obtained from the ideal MPR model of $P_s = 1$. As can be seen, without

doi: 10.1049/iet-com:20070538

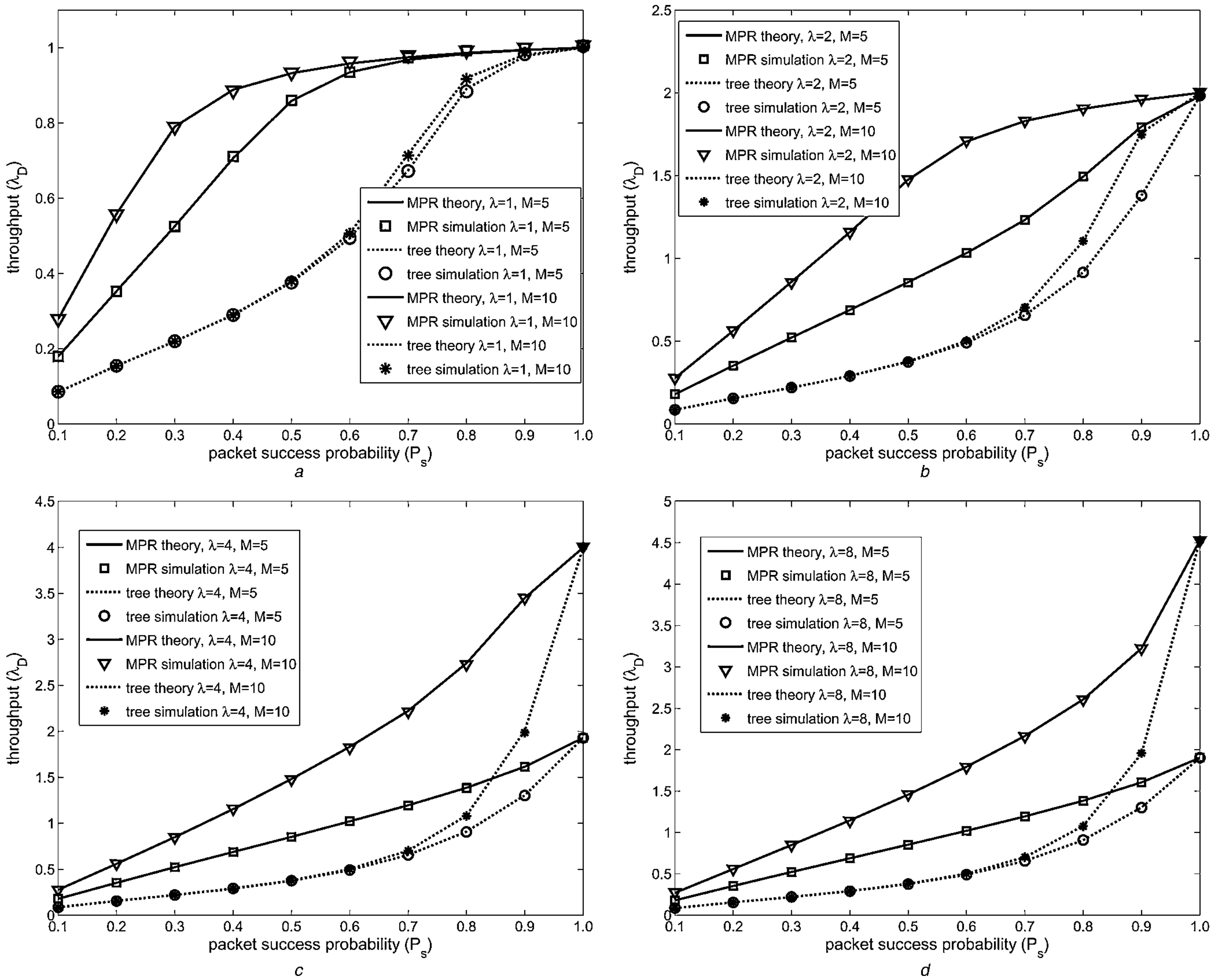


Figure 4 *Throughput results for the MPR MAC protocol and the splitting-tree algorithm under different offered traffic loads and different MPR capabilities*

a $\lambda = 1$
b $\lambda = 2$
c $\lambda = 4$
d $\lambda = 8$

channel errors, the conventional splitting-tree algorithm has the same λ_Ds as the MPR MAC protocol. That is to say, by simply assuming an error-free channel and an ideal PHY of MPR capability M, the tree algorithm can easily enjoy the capacity increment proportional to M without any requirement to modify the algorithm itself. However, the perfect channel assumption is still unrealistic even under the most recent wireless PHY, and the improvements based on it can rapidly degrade when P_s decreases, as shown in this figure.

5 Conclusion

In this paper, we propose an MPR MAC protocol and derive its system throughput with a general MPR model. In addition, we also analyse the throughput performance for the conventional splitting-tree algorithm in the MPR model as a special case of the protocol. Our experiments clearly confirm the correctness of our derived model and show their results to be very consistent with those of the theory. From both the theory and the experiments, we can see that the MPR MAC protocol can significantly outperform the splitting-tree algorithm under different degrees of channel error. The performance benefits come from our special attentions to the splitting-tree algorithm's characteristics and careful modifications to preserve its simplicity and efficiency. In some sense, this work may serve as an example showing how to improve a conventional MAC protocol with a general MPR model. Extending other MAC protocols with the general MPR model may be done in a similar way, which will be our future work.

6 Acknowledgment

This work was supported by the National Science Council, Republic of China, under grant NSC96-2221-E-126-001.

7 References

[1] ABRAMSON N.: 'The Aloha system – another alternative for computer communications'. Proc. Fall Joint Computer Conf. (AFIPS), 1970

[2] CAPETANAKIS J.: 'Tree algorithms for packet broadcast channels', *IEEE Trans. Inf. Theory*, 1979, **25**, (5), pp. 505–515

[3] GALLAGER R.: 'Conflict resolution in random access broadcast networks'. Proc. AFOSR Workshop Communication Theory and Applications, September 1978, pp. 74–76

[4] GHEZ S., VERDU S., SCHWARTZ S.C.: 'Stability properties of slotted aloha with multipacket reception capability', *IEEE Trans. Autom. Control*, 1988, **33**, (7), pp. 640–649

[5] TONG L., ZHAO Q., MERGEN G.: 'Multipacket reception in random access wireless networks: from signal processing to optimal medium access control', *IEEE Commun. Mag.*, 2001, **39**, (11), pp. 108–112

[6] ZHAO Q., TONG L.: 'A dynamic queue protocol for multiaccess wireless networks with multipacket reception', *IEEE Trans. Wirel. Commun.*, 2004, **3**, (6), pp. 2221–2231

[7] ZHENG P.X., ZHANG Y.J., LIEW S.C.: 'Multipacket reception in wireless local area networks'. Proc. IEEE Conf. on Communications (ICC06), June 2006, pp. 3670–3675

[8] ZHENG P.X., ZHANG Y.J., LIEW S.C.: 'Analysis of exponential backoff with multipacket reception in wireless networks'. Proc. IEEE Conf. Local Computer Networks, November 2006, pp. 855–862

[9] BUDIANU C., BEN-DAVID S., TONG L.: 'Estimation of the number of operating sensors in large-scale sensor networks with mobile access', *IEEE Trans. Signal Process.*, 2006, **54**, (5), pp. 1703–1715

[10] KAO E.: 'An introduction to stochastic processes' (Duxbury Press, 1996)

doi: 10.1049/iet-com:20070538