

On the Stability of the Contention Mechanism of PRMA++

B. Gloss, R. Sigle, U. Weiss

University of Stuttgart, Institute of Communication Networks and Computer Engineering (IND)
Prof. Dr.-Ing. Dr. h. c. mult. P. J. Kühn, Pfaffenwaldring 47, D-70569 Stuttgart, Germany
Tel. +49/711/685-7992, Fax +49/711/685-7983, {gloss,sigle,weiss}@ind.uni-stuttgart.de

Recently the Packet Reservation Multiple Access protocol PRMA++ was proposed as a MAC protocol for future mobile communication systems. As the reservation mechanism of this protocol is based on slotted ALOHA, instabilities in the random access phase can induce it to work inefficiently. This paper studies the contention mechanism of PRMA++. By analysing its dynamic behaviour we identify efficient and inefficient operation points of PRMA++ and point out a numerical method to adjust the retry probability of collided request packets so that PRMA++ operates efficiently. The proposed analysis method drastically reduces computation complexity and hence enables the investigation of systems with a high number of active mobile stations.

1. Introduction

In addition to telephony, data services will play a major roll in future mobile communication systems. In order to provide cost efficient services most data applications (file transfer, web browsing, multimedia etc.) demand a packet based transmission protocol allowing statistical multiplexing.

In 1989 Goodman et al. [4] proposed the PRMA (Packet Reservation Multiple Access) protocol, which offers the possibility to use the same wireless channel efficiently for voice and data communications. To support a variety of different services the European RACE ATDMA (Advanced TDMA) project group proposed an enhanced version of PRMA, called PRMA++ [5].

PRMA++ is a combination of TDMA (Time Division Multiple Access) and slotted ALOHA. The protocol separates different users and services by different time slots. A mobile station that wants to transmit data over the wireless channel must send a bandwidth request first. For that purpose it sends a request packet in special reservation slots (R-slots) using random access. If more than one mobile station send their requests in the same time slot, a collision occurs. The contention mechanism used to resolve the collisions allows the mobile stations to retransmit their requests in the next reservation slot with a fixed retry probability q only. As well known from the slotted ALOHA protocol, an inappropriate q results in an instability of the contention mechanism. In that case, the PRMA++ protocol becomes inefficient. It is occupied resolving collisions while valuable information bandwidth is left unused. To prevent the protocol from operating inefficiently, q must be well adapted to the number of mobile stations and the number of reservation slots.

This paper analyses the contention mechanism of PRMA++ with the help of a two dimensional discrete-state, discrete-time Markov model. It points out a way to determine the best retry probability q for a fixed number of mobile stations so that PRMA++ operates efficiently. Furthermore, the “interim state”- method is proposed to calculate the state transition probabilities. This method has the advantage that the numerical evaluations can be done in short computation time even for a high number of mobile stations. In an example scenario, we use this method to find the best retry probability q for a system with 140 mobile stations in a cell.

The remainder of this paper is organized as follows. Section 2 gives a short introduction to PRMA++ and outlines our modelling assumptions. Moreover, the state diagram of the MAC unit is presented and our method to calculate the transition probabilities is introduced. By analysing the drift of the state variables, efficient and inefficient operation points of PRMA++ are identified. Section 3 explains how to determine the best retry probability q .

2. System Model

2.1. The PRMA++ Protocol

This section gives a short introduction to PRMA++ as it is used in our analysis. For an detailed description of PRMA++ the reader is referred to [3,5]. Fig. 1 depicts the parts of the PRMA++ frame structure which are important for the analysis of the contention mechanism. Three different slot types can be distinguished: Reservation slots (R-slot), Information slots (I-slots) and Acknowledgement slots (A-slots). The R-slots in the uplink are used for bandwidth requests only. Voice and data traffic is transmitted within the I-slots. Each R-slot has got an associated A-slot in the downlink frame for acknowledgments. We suppose the R-slots be inserted in the frame in regular intervals so that a frame can be divided into subframes of equal length.

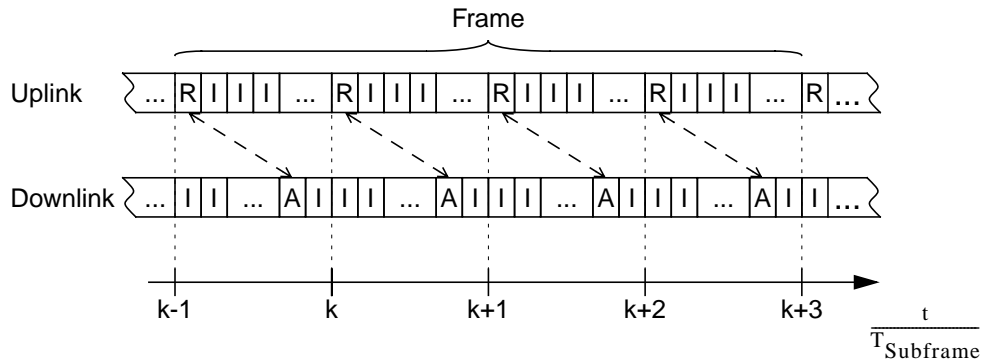


Figure 1. Frame structure of PRMA++ and numeration of the subframes

Each uplink subframe consists of one R-slot and several I-slots. We assume that the R-slot is the first slot in the subframe. Each downlink subframe consists of several I-slots and one A-slot, in which the acknowledgements to the reservation requests are transmitted. The exact position of the A-slot in the downlink is not significant to our analysis, as long as it is disposed behind the corresponding R-slot.

2.2. The Mobile Stations

A mobile station consists of a voice source and a Medium Access Control (MAC) unit. Although our analysis method is principally applicable to voice and data traffic, in this paper we focus on voice communication only. As described by Brady [2], a voice source with silence suppression can be modelled as a talkspurt/silence source. One I-slot per frame is necessary to transmit the voice packets of the talkspurt. During a silence period no transmission takes place. Hence, an eventually allocated I-slot is released and no bandwidth is occupied by the mobile station in this period. We assume the duration of the talkspurts and silence periods to be negative exponentially distributed with mean values $\overline{T}_{Talk} = 1.4$ sec and $\overline{T}_{Silence} = 1.7$ sec.

We model the MAC unit of a mobile station by three discrete states (see fig. 2), namely *Idle*, *Request*, and *Success*.

During the silence period of the voice source the MAC unit of the mobile station remains in the *Idle* state. At the beginning of a talkspurt the MAC unit must demand uplink bandwidth to transmit voice packets. This is done by sending a request packet to the resource manager in the base station in the next R-Slot. Depending on the behaviour of the other mobile stations the request packet of the observed mobile station may collide or may be successfully transmitted.

If no collision occurs, the MAC unit changes to the *Success* state and receives either an “I-slot allocation” or a “request queued” signal in the next A-slot. Which one of these two signals will be received only depends on the availability of I-Slots. Since this does not affect the performance of the contention mechanism, it need not to be considered in this analysis. In case of a collision the MAC unit changes to the *Request* state. In this case, the request is called to be backlogged. The MAC unit tries to retransmit its request in the next R-slot with retry probability q .

If the MAC unit successfully retransmits its request packet, it changes from the *Request* state to the *Success* state. If the request packet collides again or if the MAC unit refrains from sending it, the MAC unit remains in the *Request* state.

At the beginning of the subframe all mobile stations in the system synchronously decide whether they send a request packet or not. Thus, the result of the contention process (collision or successful transmission) can be determined at the beginning of the R-Slot, which also is the beginning of the subframe.

At the end of a talkspurt, the MAC unit immediately changes back to the *Idle* state, independent of the previous state (*Request* or *Success*).

In the following we refer to the *state of the MAC unit of a mobile station* by the shorter expression “*state of a mobile station*”.

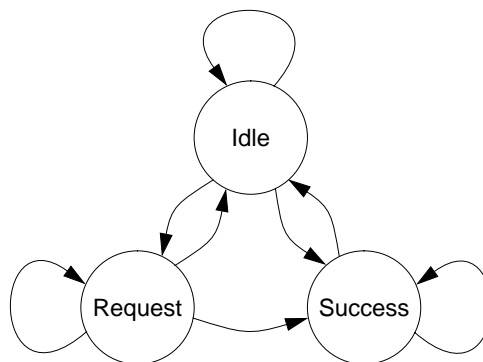


Figure 2: *State diagram of a mobile station's MAC unit*

2.3. The System States of the Contention Process

We consider one carrier of one base station and a fixed number N of established connections. The contention process is modelled by a finite discrete-time, discrete-state Markov model. The system state is described by the two dimensional state vector $\vec{s}(t) = (n(t), m(t))$, where $n(t)$ is the number of mobile stations in the *Request* state (i.e. the number of backlogged requests) and $m(t)$ the number of mobile stations in the *Success* state. Since the number of mobile stations in the system is fixed to N , the equation $n(t) + m(t) \leq N$ must be fulfilled. Thus

$$\mathcal{S} = \{(n, m) | n, m \in [0, N], n + m \leq N\} \quad (1)$$

is the state space of this system. There are $\frac{(N+1)(N+2)}{2}$ different states.

We consider the system at discrete time points $t_k = k \cdot T_{Subframe}$ at the beginning of each subframe. The corresponding system state is denoted $\vec{s}(t_k) = \vec{s}_k$ and the state at the beginning of the next subframe $\vec{s}(t_{k+1}) = \vec{s}_{k+1}$. The state transition probabilities do not depend on the considered subframe, i.e. they are independent of k .

The “Interim State” method

To determine the state transition probabilities from state \vec{s}_k to state \vec{s}_{k+1} we insert an interim state \vec{s}_{k+} . With it the transition from \vec{s}_k to \vec{s}_{k+1} is separated into two steps: the first step from state \vec{s}_k to \vec{s}_{k+} , the second step from state \vec{s}_{k+} to \vec{s}_{k+1} . This approach simplifies the determination of the transition probabilities.

In the first step, we determine the result of the reservation process: we calculate the probabilities of a collision or a successful request transmission. These probabilities depend on the system state \vec{s}_k , on new arrivals in the subframe $k - 1$, and on the retry probability q . During the first step no time elapses!

In the second step, we consider the state changes caused by the end of talkspurts in subframe k . The time which elapses during this transition is the duration of one subframe. By finally concatenating the first and second step we can determine the transition probability from state \vec{s}_k to \vec{s}_{k+1} .

The first step

The possible state transitions of a mobile station’s MAC unit are presented in fig. 3 (left). The right picture of fig. 3 depicts the state space for $N = 10$ mobile stations and the possible transitions for four sample points. The four sample points correspond to four different cases which are distinguished in the calculation of the transition probabilities. The crossed states in fig. 3 do not belong to the state space.

The probability p_T that a single MAC unit leaves the *Idle* state is equivalent to the probability that the silence period ends, i.e. that the next talkspurt begins, during the subframe:

$$p_T = 1 - e^{-\frac{T_{Subframe}}{T_{Silence}}} \quad (2)$$

Given that n_k mobile stations are in the *Request* state and m_k mobile stations are in the *Success* state, the distribution of the number of mobile stations in the *Idle* state which send a first reservation is given by equation 3:

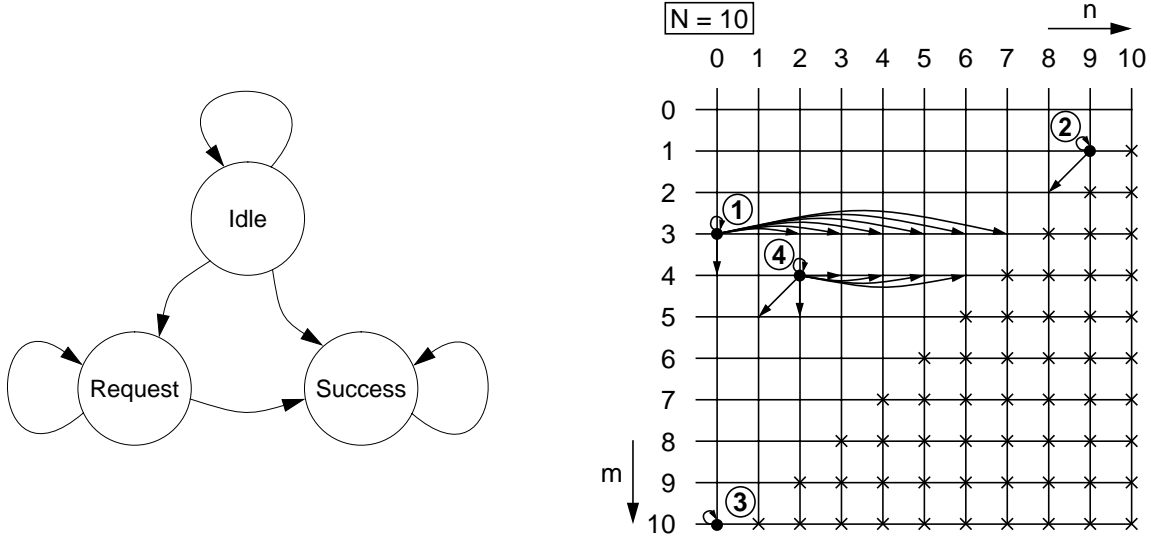


Figure 3. *First step: possible state transitions of a MAC unit (left) and examples for possible system state transitions for $N = 10$ (right)*

$$\begin{aligned}
 P_A(x, n_k, m_k) &= P\{X_A = x \mid n = n_k, m = m_k\} \\
 &= \binom{N - n_k - m_k}{x} \cdot p_T^x \cdot (1 - p_T)^{N - n_k - m_k - x}, \quad x \in [0, N - n_k - m_k] \quad (3)
 \end{aligned}$$

The distribution of the number of mobile stations in the *Request* state which retransmit a request is

$$\begin{aligned}
 P_R(x, n_k) &= P\{X_R = x \mid n = n_k\} \\
 &= \binom{n_k}{x} \cdot q^x \cdot (1 - q)^{n_k - x}, \quad x \in [0, n_k], \forall m_k \quad (4)
 \end{aligned}$$

where q is the retry probability.

For the calculation of the transition probabilities $a_{\vec{s}_k, \vec{s}_{k+}} = a_{(n_k, m_k), (n_{k+}, m_{k+})}$ of the first step we distinguish four cases.¹ One example for each case is presented in fig. 3 (right). All other transition probabilities not included in case 1-4 describe impossible transitions and hence are equal to zero.

Case 1: $n_k = 0, m_k < N$

$$\begin{aligned}
 a_{(0, m_k), (0, m_k)} &= P_A(0, 0, m_k) \\
 a_{(0, m_k), (0, m_k + 1)} &= P_A(1, 0, m_k) \\
 a_{(0, m_k), (x, m_k)} &= P_A(x, 0, m_k), \quad x \in [2, N - m_k]
 \end{aligned}$$

Case 2: $n_k + m_k = N, n_k > 0$

$$\begin{aligned}
 a_{(n_k, m_k), (n_k, m_k)} &= 1 - P_R(1, n_k) \\
 a_{(n_k, m_k), (n_k - 1, m_k + 1)} &= P_R(1, n_k)
 \end{aligned}$$

¹Note that we write all vectors as lying vectors; the components are separated by a comma: $\vec{s}_k = (n_k, m_k)$.

Case 3: $n_k = 0, m_k = N$

$$a_{(0,N),(0,N)} = 1$$

Case 4: others

$$a_{(n_k,m_k),(n_k,m_k)} = P_A(0, n_k, m_k) \cdot (1 - P_R(1, n_k))$$

$$a_{(n_k,m_k),(n_k-1,m_k+1)} = P_A(0, n_k, m_k) \cdot P_R(1, n_k)$$

$$a_{(n_k,m_k),(n_k,m_k+1)} = P_A(1, n_k, m_k) \cdot P_R(0, n_k)$$

$$a_{(n_k,m_k),(n_k+1,m_k)} = P_A(1, n_k, m_k) \cdot (1 - P_R(0, n_k))$$

$$a_{(n_k,m_k),(n_k+x,m_k)} = P_A(x, n_k, m_k), \quad x \in [2, N - n_k - m_k]$$

In order to write the transition probabilities into one matrix A the mapping of the two-dimensional system states to the rows and columns of the matrix must be unique and equal for rows and columns. For each row of the matrix, the starting state (vector \vec{s}_k) is fixed to one state. The destination state (vector \vec{s}_{k+}) runs from the upper left corner of the system state space (see figure 3) to the lower left corner along the m-axis, then in the next column again from top to bottom, and so forth. Thus the $\frac{(N+1)(N+2)}{2} \times \frac{(N+1)(N+2)}{2}$ matrix A writes as follows:

$$A = [a_{\vec{s}_k, \vec{s}_{k+}} \quad \forall \vec{s}_k, \vec{s}_{k+} \in \mathcal{S}] = \begin{bmatrix} a_{(0,0),(0,0)} & a_{(0,0),(0,1)} & \cdots & a_{(0,0),(0,N)} & a_{(0,0),(1,0)} & \cdots & a_{(0,0),(1,N-1)} & a_{(0,0),(2,0)} & \cdots & a_{(0,0),(N,0)} \\ a_{(0,1),(0,0)} & a_{(0,1),(0,1)} & \cdots & a_{(0,1),(0,N)} & a_{(0,1),(1,0)} & \cdots & a_{(0,1),(1,N-1)} & a_{(0,1),(2,0)} & \cdots & a_{(0,1),(N,0)} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ a_{(0,N),(0,0)} & a_{(0,N),(0,1)} & \cdots & a_{(0,N),(0,N)} & a_{(0,N),(1,0)} & \cdots & a_{(0,N),(1,N-1)} & a_{(0,N),(2,0)} & \cdots & a_{(0,N),(N,0)} \\ a_{(1,0),(0,0)} & a_{(1,0),(0,1)} & \cdots & a_{(1,0),(0,N)} & a_{(1,0),(1,0)} & \cdots & a_{(1,0),(1,N-1)} & a_{(1,0),(2,0)} & \cdots & a_{(1,0),(N,0)} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ a_{(N,0),(0,0)} & a_{(N,0),(0,1)} & \cdots & a_{(N,0),(0,N)} & a_{(N,0),(1,0)} & \cdots & a_{(N,0),(1,N-1)} & a_{(N,0),(2,0)} & \cdots & a_{(N,0),(N,0)} \end{bmatrix} \quad (5)$$

The second step

In this step, only the end of talkspurts change the state of the MAC unit of a mobile station. The possible transitions of the MAC unit of a mobile station are depicted in fig. 4 (left). The right picture again represents the system state space for $N = 10$ and an example of the possible transitions for one system state. The probability that a talkspurt ends during a subframe, is

$$p_S = 1 - e^{-\frac{T_{Subframe}}{T_{Talk}}} \quad (6)$$

The distribution of the number of mobile stations which are in the *Request* state and which lose the necessity to make a reservation due to the end of the talkspurt is:

$$\begin{aligned} P_{ER}(x, n_{k+}) &= P\{X_{ER} = x \mid n = n_{k+}\} \\ &= \binom{n_{k+}}{x} \cdot p_S^x \cdot (1 - p_S)^{n_{k+} - x}, \quad x \in [0, n_{k+}] \end{aligned} \quad (7)$$

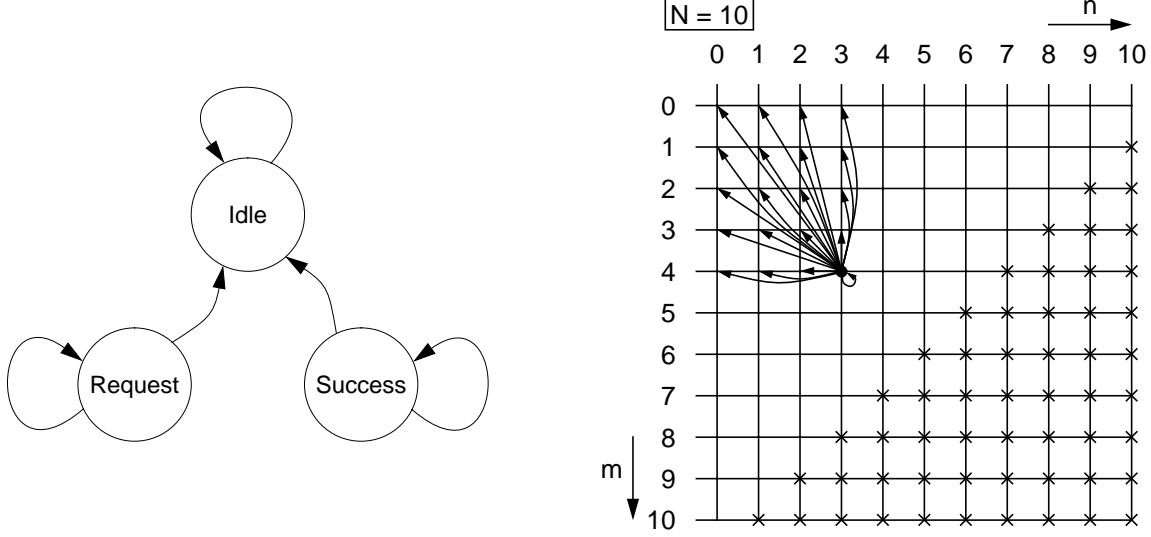


Figure 4. *Second step: possible state transitions of a MAC unit (left) and examples for possible system state transitions for $N = 10$ (right)*

The distribution of the number of mobile stations which are in the *Success* state and which terminate their actual transmission is

$$\begin{aligned}
 P_{ES}(x, m_{k+}) &= P\{X_{ES} = x \mid m = m_{k+}\} \\
 &= \binom{m_{k+}}{x} \cdot p_S^x \cdot (1 - p_S)^{m_{k+} - x}, \quad x \in [0, m_{k+}]
 \end{aligned} \tag{8}$$

Therefore the transition probabilities $b_{\vec{s}_{k+}, \vec{s}_{k+1}}$ for the second step are given by

$$b_{\vec{s}_{k+}, \vec{s}_{k+1}} = \begin{cases} P\{X_{ER} = n_{k+} - n_{k+1}\} \cdot P\{X_{ES} = m_{k+} - m_{k+1}\}, & n_{k+} \geq n_{k+1}, m_{k+} \geq m_{k+1} \\ 0, & \text{otherwise} \end{cases} \tag{9}$$

In the same manner as above, we form the $\frac{(N+1)(N+2)}{2} \times \frac{(N+1)(N+2)}{2}$ transition probability matrix B for the second step:

$$B = [b_{\vec{s}_{k+}, \vec{s}_{k+1}} \mid \forall \vec{s}_{k+}, \vec{s}_{k+1} \in \mathcal{S}] = \begin{bmatrix} b_{(0,0),(0,0)} & b_{(0,0),(0,1)} & \cdots & b_{(0,0),(N,0)} \\ b_{(0,1),(0,0)} & b_{(0,1),(0,1)} & \cdots & b_{(0,1),(N,0)} \\ \vdots & \vdots & \ddots & \vdots \\ b_{(N,0),(0,0)} & b_{(N,0),(0,1)} & \cdots & b_{(N,0),(N,0)} \end{bmatrix} \tag{10}$$

Combining step one and two

In order to transit from state \vec{s}_k to state \vec{s}_{k+1} , the system may pass by any “interim” state \vec{s}_{k+} . Therefore the transition probabilities can be calculated as

$$p_{\vec{s}_k, \vec{s}_{k+1}} = \sum_{\forall \vec{s}_{k+} \in \mathcal{S}} a_{\vec{s}_k, \vec{s}_{k+}} \cdot b_{\vec{s}_{k+}, \vec{s}_{k+1}} \tag{11}$$

This operation is identical to the multiplication of row (n_k, m_k) of matrix A and column (n_{k+1}, m_{k+1}) of matrix B . The matrix P for the transition from state \vec{s}_k to \vec{s}_{k+1} can be

obtained by performing the operation

$$\begin{aligned}
P &= A \cdot B \\
&= \left[p_{\vec{s}_k, \vec{s}_{k+1}} \forall \vec{s}_k, \vec{s}_{k+1} \in \mathcal{S} \right] = \begin{bmatrix} P_{(0,0),(0,0)} & P_{(0,0),(0,1)} & \cdots & P_{(0,0),(N,0)} \\ P_{(0,1),(0,0)} & P_{(0,1),(0,1)} & \cdots & P_{(0,1),(N,0)} \\ \vdots & \vdots & \ddots & \vdots \\ P_{(N,0),(0,0)} & P_{(N,0),(0,1)} & \cdots & P_{(N,0),(N,0)} \end{bmatrix} \quad (12)
\end{aligned}$$

2.4. The Drift

For the stability analysis of the protocol we need some information about the dynamic behaviour of the system. This information can be gained by analysing the *drift* [1]. If the system is in state \vec{s}_k , the drift is the expected excursion from this state during the time of the next subframe. In our two-dimensional state space the drift itself is a vector with two components

$$\vec{D}(n, m) = \begin{pmatrix} D_n(n, m) \\ D_m(n, m) \end{pmatrix} \quad (13)$$

The first component $D_n(n, m)$ is the expected change of the number n of backlogged requests (number of mobile stations in the *Request* state) during the next subframe for the system being in state $\vec{s}_k = (n, m)$. We call this component the “ n -drift”. With the transition probabilities $p_{\vec{s}_k, \vec{s}_{k+1}}$ of equation (12) it is defined as

$$D_n(n, m) = \sum_{\forall (i,j) \in \mathcal{S}} (i - n) \cdot p_{(n,m),(i,j)} \quad (14)$$

The second component, the “ m -drift”, is the expected change of the number of mobile stations in the *success* state. Thus

$$D_m(n, m) = \sum_{\forall (i,j) \in \mathcal{S}} (j - m) \cdot p_{(n,m),(i,j)} \quad (15)$$

For the numerical evaluation of the drift formulas we can exploit that most elements of the matrices A and B are zero. This is due to the fact that starting from one system state only few system states are attainable. By using this knowledge appropriately we save a lot of computational power for the necessary calculations.

The numerical drift analysis in the next section helps to understand the behaviour of the contention mechanism and is further useful to find a suitable retry probability q for a collided request.

3. Numerical Results

For the numerical evaluation of the derived formulas we choose the PRMA++ frame structure for the microcellular environment as proposed by the European RACE ATDMA project [5]. In the microcellular environment the frame consists of 72 slots. Four slots are reserved for the contention mechanism (“reservation slots”), leaving 68 for data transmission (“information slots”). The frame length is 5 ms, hence $T_{Subframe} = \frac{5ms}{4} = 1.25ms$.

Our computer simulations of the PRMA++ protocol show that such a system can bear about 140 voice conversation at the capacity limit of 1% packet dropping. Similar results can be found in [3]. All following results will therefore be presented for a system with a fixed number $N = 140$ of active mobile stations.

Fig. 5 depicts the two components of the drift using equations (14, 15), the n -drift (left figure) and the m -drift (right figure). In this example the retry probability q is chosen to 0.15. n and m run from 0 to N , while $n + m \leq N$ must be satisfied (see section 2.3).

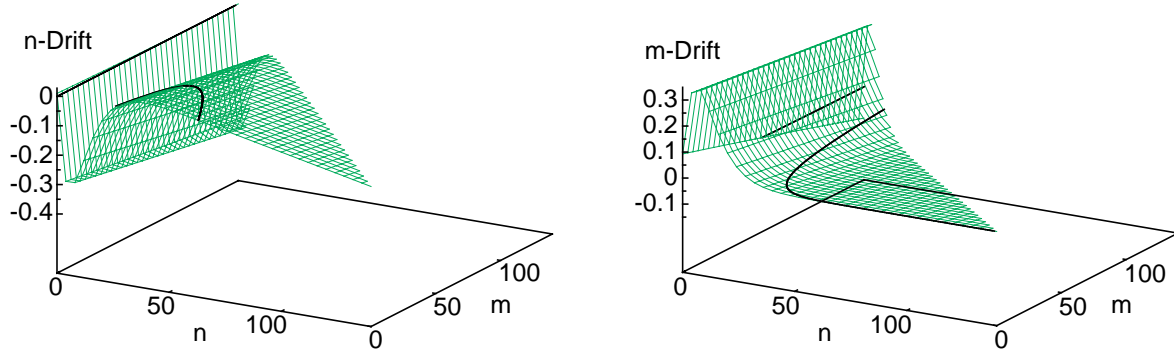


Figure 5. n -drift (left) and m -drift (right) for $N=140$, 4 R-slots, $q=0.15$

To better understand the form of the three-dimensional graphs, fig. 6 displays the n -drift for $m = 0$. This is the front of the left graph in fig. 5.

For a negative n -drift the number of backlogged requests is expected to decrease, while for a positive n -drift it is expected to increase. It is obvious that the n -drift is positive for $n = 0$ and negative for $n = N$.

In fig. 6 we can identify three zero crossing points $P1'$, $P2'$, and $P3'$. For these points the n -drift is zero, i.e. the number n of backlogged requests is expected to be constant. $P1'$ and $P3'$ are stable points of the n -drift, i.e. small variations of n will force the system back into these points. $P2'$ is unstable, as a small variation of n will drive the system in the direction of either $P1'$ or $P3'$ (see the horizontal arrows in fig. 6).

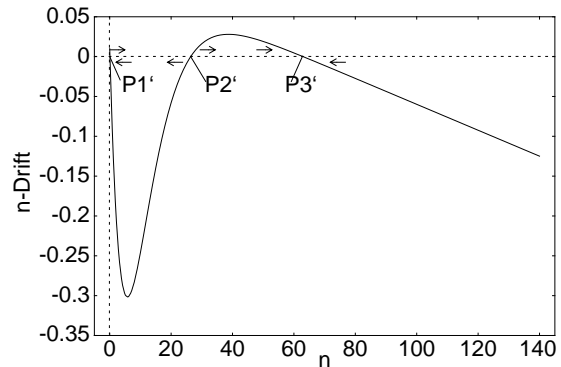


Figure 6: n -drift for $m=0$ ($q=0.15$)

Going back to the three-dimensional graphs of fig. 5 the same observation can be made. In each graph the black curves represent the contour lines for which the corresponding drift is zero. Fig. 7 depicts these contour lines in one single graph.

The solid line in fig. 7 depicts the contour line for n -drift = 0, the dashed line is the contour for m -drift = 0. The dotted diagonal line displays the limit of the region $n+m \leq N$. The right picture additionally shows the direction of the drift vector $\vec{D}(n, m)$.

There are three points for which the n - and m -drift are equal to zero (the crossing points of the contour lines): $P1(n_1, m_1)$, $P2(n_2, m_2)$, and $P3(n_3, m_3)$. As indicated by the vector field of fig. 7 (right), the system is expected to tend towards P1 if it is in the left

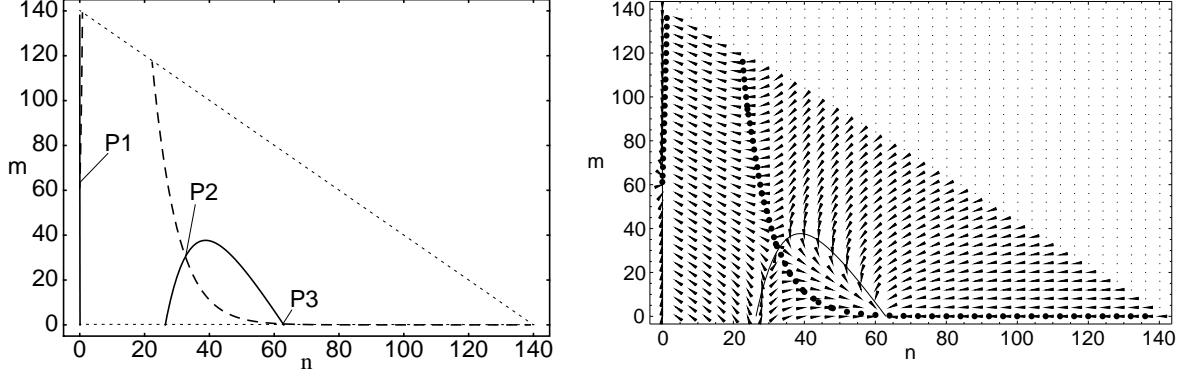


Figure 7. Contour lines for $m\text{-drift}=0$ and $n\text{-drift}=0$ (left) and overlaid vector field (right), $q=0.15$

part of the graph. In the right part of the graph the system tends towards point P3. Point P2 is not a stable point for the drift, so that the system is not expected to remain in this point.

In general, the contention mechanism of PRMA++ is said to be unstable if the number of colliding requests packets is high. Then, voice packets must be dropped before transmission due to the lack of resource reservation. While the protocol is occupied resolving collisions of request packets, the information throughput tends towards zero. To avoid confusions of the term “stability” we prefer to say that PRMA++ then operates *inefficiently*. Conversely, if the number of successful request packets is high, PRMA++ operates *efficiently*.

In fig. 7 P1 is the efficient operation point (the number n_1 of backlogged packets is close to 0) for our system, P3 is the inefficient operation point (the number n_3 of backlogged packets is about 63). In order to assure that PRMA++ operates efficiently, the point P3 must be eliminated. This can be done by adjusting the retry probability q .

Fig. 8 depicts the same contour lines for $q = 0.111$ and 0.117 . We see that for $q = 0.111$ the contour lines are apart from each other, for $q = 0.117$ they touch each other. In comparison to fig. 7 ($q = 0.150$) the points P2 and P3 are eliminated for $q < 0.117$. Only one stable point of operation exists, namely point P1. The system is expected to return always to point P1 in which the number of backlogged requests is close to zero. The contention protocol is stable, PRMA++ operates efficiently.

The above presented result is valid for a frame structure containing four R-slots and 68 I-slots. If the number r of R-slots is reduced to 3, 2, or 1 (and the number i of I-slots is augmented so that $r + i = 72$), the retry probability q must be adapted. The following table presents the values q_x for which the contour lines just touch each other.

Number of R-slots	q_x
4	0.117
3	0.112
2	0.102
1	0.084

Finally we conclude that the PRMA++ protocol always works efficiently for $q < q_x$. Even if the system once gets into a state with many backlogged requests, it is expected

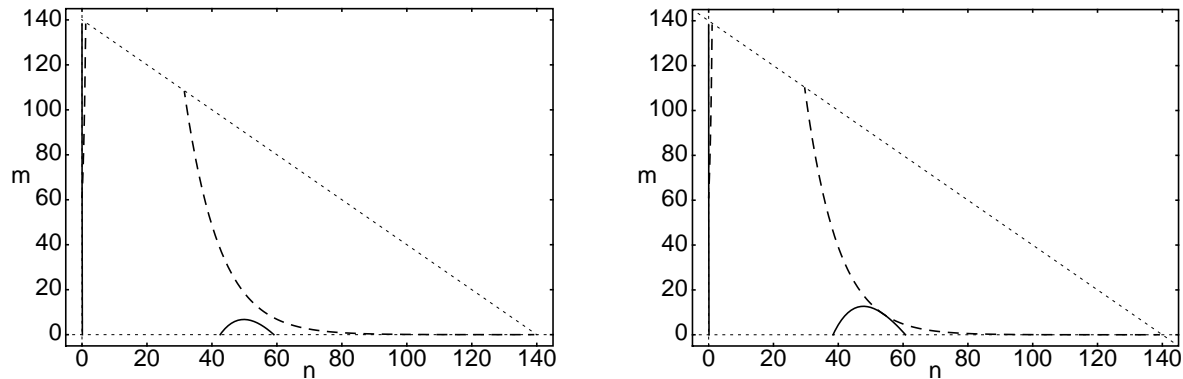


Figure 8. Contour lines of n -drift=0, m -drift=0 for $q=0.111$ (left) and $q=0.117$ (right)

to return to the stable point of operation P1. For $q \geq q_x$ this is not the case. The system may remain in the inefficient operating point P3 for quite a long time.

4. Conclusions

In this paper we presented a detailed analysis of the contention mechanism of PRMA++. With a two-dimensional drift analysis we described the dynamic behaviour of the contention protocol. We determined two stable operation points for a fixed load situation. In one of these points PRMA++ works efficiently, in the other one inefficiently. By adjusting the retry probability q properly the inefficient operation point can be eliminated. Hence, PRMA++ is expected to work always efficiently. This results in a low packet dropping probability and a high data throughput.

Our method of calculating the transition probabilities by introducing an “interim state” reduces the complexity of the numerical evaluations for a high number of mobile stations. Even for 140 mobile stations we could calculate the drift curves on an ordinary workstation within twenty minutes.

5. Acknowledgements

The authors would like to thank Stefan Bodamer for his useful suggestions.

REFERENCES

1. Bertsekas, D.; Gallager, R.; “Data Networks”, Prentice-Hall, 1987
2. Brady, P. T.; “A Model for On-Off Speech Patterns in Two-Way Conversation”, Bell System Technical Journal, Sept. 1969, vol. 48, no. 7, pp 2445-2472
3. Dunlop, J.; Robertson, D.; Cosimini, P., De Vile, J.; “Development and Optimisation of a Statistical Multiplexing Mechanism for ATDMA”, Proc. of VTC 94 - 44th IEEE Vehicular Technology Conference, Stockholm, Sweden, June 1994, pp. 1040-1044
4. Goodman, D. J.; Valenzuela, R. A.; Gayliard, K. T.; Ramamurthi, B.; “Packet Reservation Multiple Access for Local Wireless Communications”, IEEE Transactions on Communications, vol. 37, no.8, August 1989, pp. 885-890
5. Urie, A.; Streeton, M.; Mourot, C.; “An Advanced TDMA Mobile Access System for UMTS”, IEEE Personal Communications, February 1995, pp. 38-47