DELFT UNIVERSITY OF TECHNOLOGY
Faculty of Electrical Engineering
Telecommunication and
Traffic Control Systems Group

Title : Throughput analysis of some mobile packet radio protocols in Rician fading Channels

Author : C.Y. Liu

Type : Graduation thesis
Size : 78
Date : July 9, 1991

Professor : Prof. Dr. J.C. Arnbak

Mentor : Prof. Dr. R. Prasad

Codenumber : A-428

Period : November 1990 - June 1991

The throughput of packet radio channels is investigated theoretically using the interference model in Rician fading environment. Three types of packet protocols are considered, namely:

i) slotted ALOHA,
ii) unslotted nonpersistent ISMA, and
iii) slotted nonpersistent ISMA.

Numerical results are presented, indicating the effect of propagation impairments on channel capacity. The results are of importance for mobile data networks, wireless office communications and other packet systems with contention-limited performance.

Summary

In this report the performance of the following protocols in a Rician fading channel is discussed:

- slotted ALOHA
- unslotted nonpersistent Inhibit Sense Multiple Access (np-ISMA)
- slotted np-ISMA

The throughput of these protocols is investigated in the presence of n-interfering signals whose random amplitudes are considered as either Rician or Rayleigh distributed; the desired signal is always considered as Rician distributed. So, in the report they are referred to as Rice + n Rice situation and Rice + n Rayleigh situation. The Rice + n Rayleigh situation is not a realistic situation, but theoretically this is interesting to investigate. The receiver capture effect is also taken into consideration.

In case of Rice + n Rayleigh situation the performance of slotted ALOHA is found to be better than that of unslotted np-ISMA for low offered traffic and any value of inhibit delay fraction d. However, for high traffic density the unslotted np-ISMA performs superior to slotted ALOHA. Slotted np-ISMA has not been investigated for the Rice + n Rayleigh case, since Rice + n Rice is a more realistic situation. All protocols are investigated for this case.

In case of Rice + n Rice the throughput of slotted ALOHA is better than unslotted np-ISMA with a high d. But slotted np-ISMA is found to perform better than slotted ALOHA. For small d, both unslotted and slotted np-ISMA have better performances than slotted ALOHA.

Symbol list

B	expected duration of busy period
d	inhibit delay fraction
G	mean offered traffic
I	expected duration of idle period
I_o	zero order modified Besselfunction
K	Rice factor
K_d	desired Rice factor
K_u	undesired Rice factor
$\bar{L}_c$	expected length of a cycle
M	product of capture ratio and the power ratio
P	sum of the direct power
P_o	mean power
P_i	average received interfering power
P_s	power of the desired signal
P_n	cumulative interfering power
p_{capt}	probability of being able to capture the receiver
$p_{succes}(z_o)$	probability of succes of the test packet
$Prob\{P_s/P_n<z_o\}$	Probability that the packet is destroyed
$Q(\alpha,\beta)$	Marcum's Q-function
r	signal amplitude
R_n	probability that the test packet is overlapped by n interfering signals
S	channel throughput
S	power of the direct signal
t_w	duration of a time slot
U	time during a cycle that the channel is used without conflicts
W	a random variable
Y	a random variable
z_o	capture ratio

Z_n	signal to interference power ratio for the test packet and n contenders
λ	average number of packets per time slot
σ^2	average power of the scattered signal
τ	duration of packets

Contents

1. Introduction

Within the telecommunication market mobile communication is a rapidly growing sector. This trend is likely to continue because of the growing mobility and of the introduction of the Pan European digital cellular network GSM by 1992. With this network almost anyone in the (West) European country can move from country to countries without facing any communication problems.

In cellular networks there is one base station and a number of mobile terminals per cell. During the access to communicate to the base station, the terminals will compete with each other. This motivates an investigation of which multiple access protocols suit the required situation. In this report, three (random) Multiple Access protocols are investigated. These protocols are slotted ALOHA [1]-[8], unslotted and slotted NonPersistent Inhibit Sense Multiple Access (np-ISMA) [20]-[24].

The investigation includes two extreme channel situations: Rice + n Rayleigh, and Rice + n Rice. The Rice + n Rayleigh situation applies within a large cell where the test terminal is near the base station and the interfering terminals are far away from the base station.
From [9]-[12] we know that for short range transmissions the channels are Rician. So, the desired signal is Rician distributed and the interfering signals are Rayleigh distributed. The Rice + n Rice describes a situation where the test terminal and the interfering terminals are within a small cell. So, the signals coming from both the test terminal and the interfering terminals are Rician distributed. This situation is also valid for indoor environment [13],[14].

The development of packet communications has been based largely on the assumption that when two packets collide, both will be lost. In fact, this is too pessimistic, since many receivers are able to extract the stronger of the overlapping packets. This is called the capture effect. This also has been taken into consideration.

The organisation of this report is as follows. In chapter 2, descriptions of multiple access, capture, and fading are given. The protocols slotted ALOHA, unslotted and slotted nonpersistent ISMA will be discussed without and with capture respectively, in chapters 3 and 4. Chapters 5 and 6 contain, respectively, the throughput analysis and the compared results. Finally, conclusions are given in chapter 7.

2. Multiple Access, Capture and Fading aspects

A general communication network consists of a number of terminals, which are stationary or mobile, that communicate with each other over communication channels. In this report we study those communication networks where multiple terminals share the same channel in their attempt to communicate. This sharing may be necessary because of efficiency considerations, or due to the nature of the application. We call such networks multiple access networks. Due to the sharing, if more than one terminal transmit simultaneously, the reception will be garbled; and if no terminal transmit, the channel will be idle. It is desirable to have multiple access schemes which resolve mutual conflicts among the terminals, such that the channel is used most of the time. Multiple access networks include satellite networks, local area networks, and packet radio networks.

Multiple Access

Existing multiple access protocols may be categorized according to how much coordination is required between the potentially conflicting terminals. Random access is an optimistic protocol in which a terminal with a message will transmit it immediately, hoping that no other terminals will transmit at the same time and thus collide with it. Since there is no coordination among the terminals, message collisions may occur. Algorithms must be developed such that collided messages are retransmitted efficiently.

Slotted ALOHA

The " most random " of all multiple access schemes is that of pure ALOHA [1]-[8], in which a station wishing to send, simply does so. Successfully received messages are acknowledged by the receiving station. But the maximum utilization of this protocol is only 18% because of the increasing numbers of collision by increasing traffic. Pure ALOHA can be improved by taking a slotted scheme in which transmission time is split into time slots and transmission is allowed to begin only at the beginning of a slot. This simple change doubles the maximum utilization of the pure ALOHA method. This method will be discussed in the next chapter.

Nonpersistent ISMA

A more coordinated protocol is the nonpersistent Carrier Sense Multiple Access (CSMA). Each terminal must listen to the channel before transmission. Nonpersistent means that if the channel is already in use, the terminals do not continuously sense, when it will be idle. Instead they will wait a random period of time. But collisions can not be avoided due to propagation delay, or collisions between two terminals which are out-of-range of each other or if they are separated by some physical obstacle opaque to UHF signals. Two such terminals are known as "hidden" from each other [20]-[24].

In this report we discuss nonpersistent ISMA which can solve the "hidden" problem. Here we have an inbound and an outbound channel. In order to prevent collisions among data packets transmitted by terminals to a common base station using the inbound channel, inhibit bits will be put on the outbound channels with the "busy" condition. After the receive operation the inhibit bits will be removed, which means "idle". With this method most of the collisions can be avoided.

Slotted nonpersistent ISMA

In this version the transmission time is split into time slots and transmission is allowed to begin only at the beginning of a slot, as mentioned for slotted ALOHA.

For all the protocols mentioned above the performance can be improved by using the capture-model. This will be discussed next. The mathematical aspects can be found in the next chapters.

Capture

Within a cellular network there is one base station per cell. During the transmission of a mobile user to the base station, there will be disturbances such as interference. The interference is coming from other users within the cell or outside the cell. The development of packet communication has been largely based on the assumption that when two packets interfere, both will be lost. By using the capture-model, this will not allways happen: a radio receiver can capture a desired packet in the presence of n-interfering packets, if the power of the desired packet P_s sufficiently exceeds the joint interfering packet power P_n during a certain time slot of duration t_w ($0 < t_w \leq \tau$), where t_w is the capture "window".
An illustration of the model is given in Fig. 2.1.

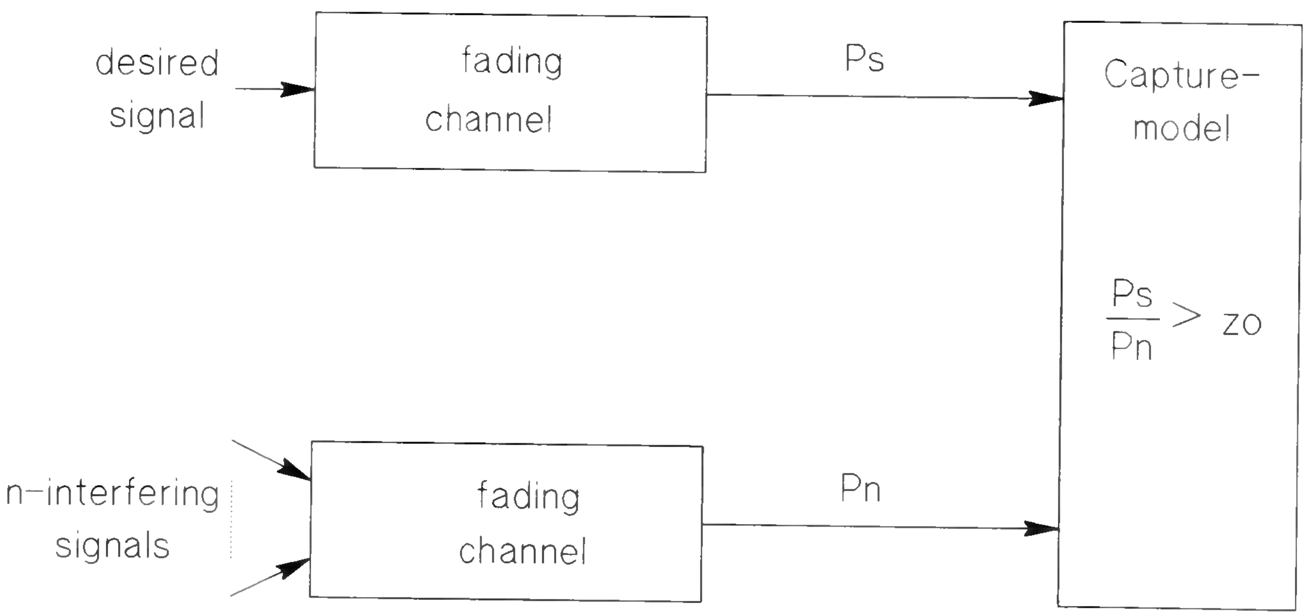


Fig. 2.1. The capture-model

The fading aspects

The signal transmission in a mobile environment is always through the air. So there is no doubt that the signals will suffer disturbances during the transmission, such as interference with other signals and reflections. The variations of the amplitudes can be constructive or destructive. In the former case the signal amplitude will be high and in the latter the amplitude will be low. We call this effect fading. In this section we describe the (fast) multipath fading and (slow) shadowing.

Multipath fading

This kind of fading is caused by reflections of the radio waves due to obstacles, so that radio waves reach the base station or mobile user in more than one way. This fading is fast and the distribution of the amplitude can be modelled by the Rayleigh distribution [3].

The Rayleigh distribution is:

$$f_r(r) = \frac{r}{P_o} \exp\left(-\frac{r^2}{2P_o}\right) \qquad (2.3.1)$$

where r is the signal amplitude
and P_0 is the average power.

In a situation that the distance between the base station and the mobile user is small, and the environment is static there is a fixed spatial pattern of maxima and minima. Mostly there is a line-of-sight path. The fast component can then better be modelled by a Rician distribution[15]:

$$f_r(r) = \frac{r}{\sigma^2} \exp\left[-\frac{r^2+S^2}{2\sigma^2}\right] I_o\left[\frac{Sr}{\sigma^2}\right] \qquad 0 \leq r < \infty \qquad (2.3.2)$$

where r is the signal amplitude

σ^2 is the average fading power

S is the peak value of the directly received signal.

The Rician distribution is characterized by the parameter K, the Rice-factor, which is the ratio of the peak power and the average fading power received over specular paths

$$K = \frac{S^2}{2\sigma^2} \qquad (2.3.3)$$

Shadowing

This kind of fading is caused by buildings, trees, hills, etc. Shadowing of a radio signal leads to a gradual, change of the local mean, as the mobile user moves. The local mean power is lognormally distributed within an area of about 500 m. In this investigation we have not considered it.

3. Multiple access protocols

3.1. Slotted ALOHA

In 1971, the University of Hawai had invented the ALOHA system [1-8].
This random access protocol is very simple, and it consists of the following modes:

1. Transmission mode. Users transmit at any time they desire, encoding their transmissions with an error detection code.

2. Listening mode. After a message transmission, a user listens for an acknowledgment (ACK) from the receiver. Transmissions from different users will sometimes overlap in time, causing reception errors in the data in each of the contending messages. We say that the messages have collided. In such cases, errors are detected, and the users receive a negative acknowledgment (NAK).

3. Retransmission mode. When a NAK is received, the messages are simply retransmitted. Of course, if the colliding users were to retransmit immediately , they would collide again. Therefore, the users retransmit after a random delay.

4. Timeout mode. If after a transmission, the user does not receive either an ACK or NAK within a specified time, the user retransmit the message.

Message arrival statistics

Let the number of packets generated and retransmitted in the network be Poisson distributed, with a mean generation rate of λ packets per second. Every packet will have the same duration τ. Then the mean offered channel traffic G in packets per time slot is

$$G = \tau\lambda \tag{3.2.1}$$

The probability that n packets are generated is given by

$$Pr[n]=\frac{G^n}{n!}\exp(-G) \tag{3.2.2}$$

For zero packet the probability is just *exp(-G)*. In an interval two packets time long, the mean number of packets generated is *2G*. So the probability of no other traffic being initiated during the entire vulnerable period is thus given by $P_0 = exp(-2G)$.
Using $S = GP_0$ we get the throughput

$$S = G\exp(-2G) \tag{3.2.3}$$

With a simple modification of this access algorithm, namely that the messages are required to be sent in the slot time between two synchronisation pulses, and can be started only at the beginning of a time slot, the rate of collisions can be reduced by one half [2]. This version is called the slotted ALOHA. The throughput is now

$$S = G\exp(-G) \tag{3.2.4}$$

In fig. 3.1. a comparison of the two throughputs are shown.

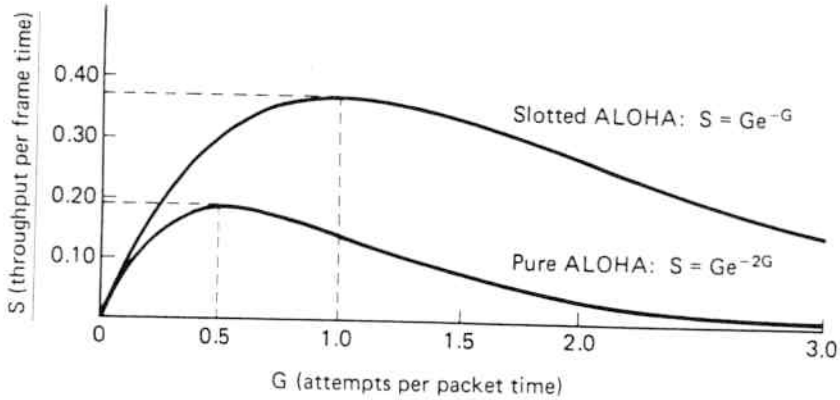
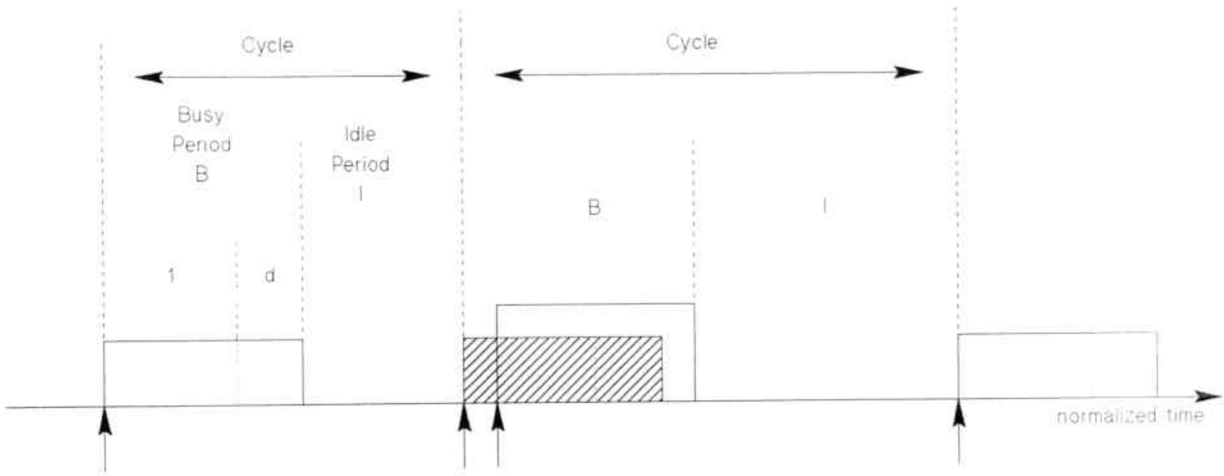


Fig.3.1. Throughput of Pure ALOHA and slotted ALOHA

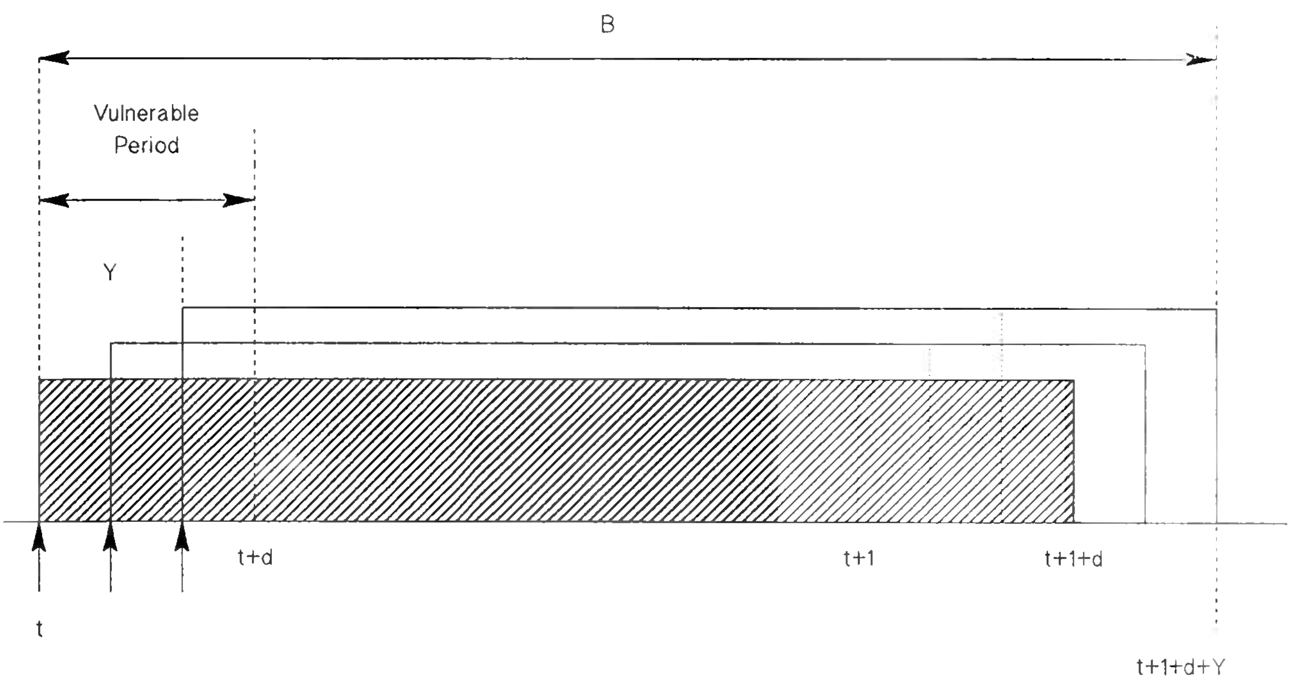
3.2. Unslotted Nonpersistent ISMA

Carrier Sense Multiple Access (CSMA) [18],[19] reduces the probability of collision by allowing terminals to sense the carrier due to other users' transmissions. But CSMA can not avoid collisions if two terminals are out-of-range of each other or if they are separated by some physical obstacle opaque to UHF signals. Two such terminals are known as "hidden" from each other. Therefore we instead consider an other category in this investigation: Inhibit Sense Multiple Access (ISMA) [20]-[24].

ISMA reduces the two problems of collision among data packets and hidden terminals. In order to prevent collisions among data packets transmitted from mobile terminals to a common base station, the inbound multiple access channel may be supplemented by



(a)



(b)

Fig. 3.2.1. Unslotted np-ISMA timing (a)Cycle structure,
(b) Unsuccesful transmission period

a broadcast inhibit-signalling channel (base-to terminals). In the latter, inhibit bits indicate the state of the inbound channel: busy or idle. As soon as the base station receives an inbound packet, the outbound signalling channel broadcasts the "busy" condition to all terminals. A fraction of the constant packet length is necessary to inhibit the mobile packet transmissions, defined as the inhibit delay fraction d, a dimensionless quantity. The time fraction required to reverse this condition is assumed to have the same value too. Thus mobile terminals are inhibited from transmission until the inbound channel is free, thus preventing most packet collisions. The inhibited packets are rescheduled according to a retransmission distribution. An illustration of ISMA timing is given in Fig. 3.2.1.

Similar to ALOHA we assume that the number of packets generated in the network be Poisson distributed, with a mean generation rate λ . The mean offered traffic is G . The packets are of identical length, normalized to have length 1.
Let t be the time of arrival of a packet which senses the channel idle. Because of the inhibit delay fraction d , there should be only one packet arriving between t *and* $t+d$ (vulnerable period). If there are more than one packet, the transmission will be unsuccesful. In the case of a collision the channel will therefore be busy for some (random) duration between $1+d$ and $1+2d$. This period in which a transmission takes place is referred to as the transmission period (TP). Suppose that $t+Y$ is the time of occurence of the last packet arriving between t and $t+d$. The transmission of all packets arriving in (t, t+Y) will be completed at $t+Y+1$. And because of the inhibit delay d, the channel is sensed idle after $t+Y+1+d$.

The expression for throughput is

$$S = \frac{U}{B+I} \tag{3.2.1}$$

where U denotes the time during a cycle that the channel is used without conflicts, B is the expected duration of the busy period and I is the expected duration of idle period.

The probability that the channel is used without conflicts, is the probability that no other terminal senses the channel during the vulnerable period d. Therefore

$$U = \exp(-dG) \tag{3.2.2}$$

The average duration of an idle period is $1/G$. The duration of a busy interval is $1+\bar{Y}+d$, where $\bar{Y}$ is the expected value of Y.
The distribution of Y is

$$\begin{aligned} F_Y(y) &\triangleq \Pr\{ \text{ no arrival occurs in an interval of length } d-y \} \\ &= \exp\{-G(d-y)\} \qquad (y \leq d) \end{aligned} \tag{3.2.3}$$

The average of Y is therefore given by

$$\bar{Y} = E(Y) = d - \frac{1}{G}(1-\exp(-dG)) \tag{3.2.4}$$

So using the above derived expressions, the throughput is

$$S = \frac{\exp(-dG)}{(1+2d) + \frac{1}{G}\exp(-dG)} \tag{3.2.5}$$

where the term in the denominator is called the expected length of a cycle ($\bar{L}_c$).

3.3. Slotted Nonpersistent ISMA

As in every slotted system (e.g. slotted ALOHA) each terminal is restricted to start transmission only at the beginning of a time slot, and the duration of a time slot is assumed to be exactly to the transmission time of a single packet. Thus there is either no collision or complete collision of the packets in which case an unsuccesful packet will be subsequently retransmitted after a random number of slots.

We will take the same environment as described for the nonpersistent ISMA protocol. The difference here is that we have a slotted axis, see Fig. 3.3.1.

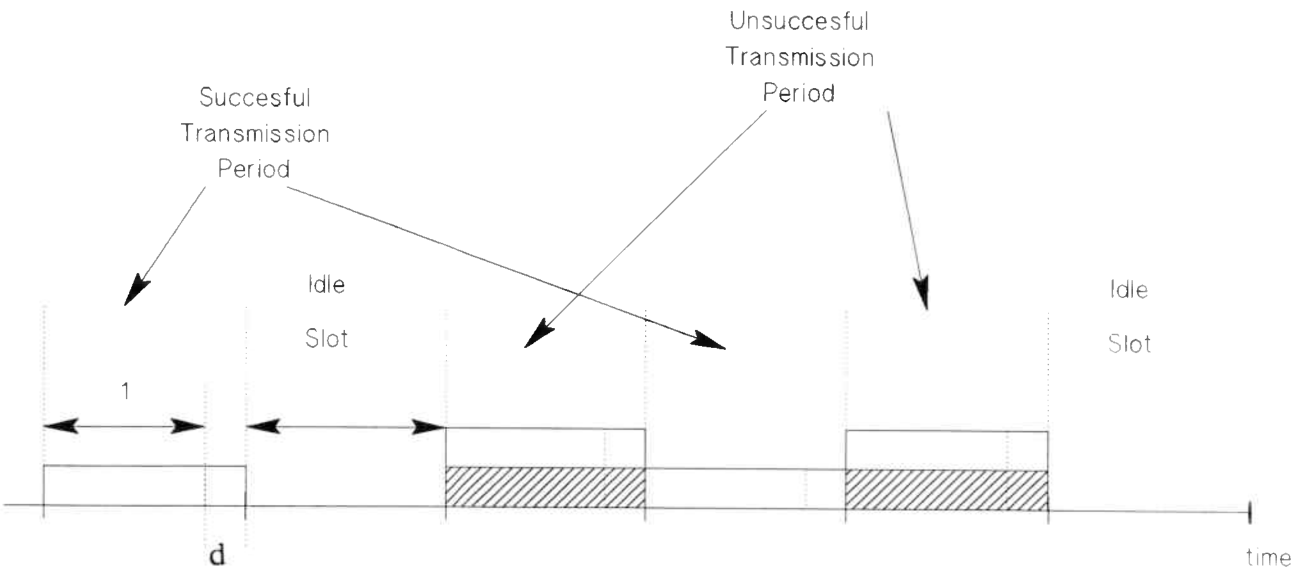


Fig. 3.3.1. Slotted nonpersistent ISMA

The length of an idle period is at least one time slot. For the idle period to be only one slot long means that there is at least one arrival in the first slot of the idle period. For it to be two slots long means that there is no arrival in its first slot and there is at least one arrival in its second slot. Continuing this reasoning and considering the Poisson scheduling process we have [19]:

$$I = \frac{d}{1-\exp(-dG)} \tag{3.3.1}$$

A collision occurs if two or more packets arrive within the same slot and are rescheduled for transmission in the next slot. A busy period will contain k transmission periods if there is at least one arrival in the last slot of each of the first k-1 transmission periods, and no arrival in the last slot of the kth transmission period. The busy period is

$$B = \frac{1+d}{\exp(-dG)} \tag{3.3.2}$$

The expected useful time is found as :

$$U = \frac{B}{1+d} P_{suc} \tag{3.3.3}$$

where P_{suc} is the probability of a succesful transmission period. We have

$$P_{suc} = \frac{prob[single\ arrival\ within\ a\ slot]}{prob[more\ arrivals\ within\ a\ slot]} = \frac{dG\exp(-dG)}{1-\exp(-dG)} \tag{3.3.4}$$

Using (3.3.1), (3.3.2), (3.3.3) and (3.2.1), one gets the throughput

$$S = \frac{dG\exp(-dG)}{1+d-\exp(-dG)} \tag{3.3.5}$$

4. The capture-model

In this chapter the behavior of the capture-model will be given. In section 4.1. a description of slotted ALOHA is presented with capture and in section 4.2. unslotted nonpersistent ISMA with capture. In section 4.3 slotted nonpersistent ISMA with capture is discussed.

4.1. Slotted ALOHA with capture

As we can see in section 3.1., if there is more then one transmission, all packets will be destroyed because of overlapping. Now in this section a more realistic model is given: the capture-model.

When data packets competing for access to a common radio receiver arrive from different distances and with independent fading levels, it is no longer certain that all colliding packets will always be annihilated by each other.
We shall assume that there is a test packet and n interfering packets. If the power of the former (P_s) sufficiently exceeds the joint interfering power (P_n) during a certain section of time slot of duration t_w ($0< t_w \leq \tau$), capture is assured.
So, the test packet is destroyed in the collision if (and only if)

$$P_s/P_n < z_o \quad \text{during } t_w \quad \text{with } n>0 \tag{4.1.1}$$

with z_o is the capture ratio [3].
The probability of being able to capture the receiver in an arbitrary time slot is

$$p_{capt}(z_o)=1-\sum_{n=1}^{\infty} R_n Prob\{P_s\ /P_n<z_o\} \tag{4.1.2}$$

Using the capture probability (4.1.2), the channel throughput can be stated as

$$S = G\left[1 - \sum_{n=1}^{\infty} R_n F_{Z_n}(z_o)\right] \qquad (4.1.3)$$

with $F_{Zn}(z_o) = \text{Prob}\{P_S/P_n < z_o\}$

Defining the signal to interference ratio for the test packet and n contenders

$$Z_n \triangleq \frac{P_s}{P_n} \qquad 0 \leq Z_n < \infty \qquad \textbf{(4.1.4)}$$

and the random variable

$$W \triangleq P_n \qquad 0 \leq W < \infty \qquad \textbf{(4.1.5)}$$

We may write the two-dimensional p.d.f.

$$f_{Zn,W}(z,w) \triangleq f_{P_s,P_n}(p_s,p_n) \left|\frac{\partial(p_s,p_n)}{\partial(z,w)}\right| \qquad (4.1.6)$$

By virtue of the stochastic independent P_s and P_n, this becomes

$$f_{Z_n,W}(z,w) = f_{P_s}(zw) f_{P_n}(w) w \qquad (4.1.7)$$

so the p.d.f. for Z_n

$$f_{Z_n}(z) = \int_0^{\infty} f_{P_s}(zw) f_{P_n}(w)\, w\, dw \qquad (4.1.8)$$

and the corresponding distributed function

$$F_{Z_n}(z_o) = \int_0^{z_o} dz \int_0^{\infty} f_{P_s}(zw) f_{P_n}(w)\, w\, dw \qquad (4.1.9)$$

4.2. Unslotted Nonpersistent ISMA with capture

In section 3.2. the nonpersistent ISMA without capture is discussed. Now we will take the capture effect into consideration, which means that the colliding packets will not always be destroyed, but will have a certain probability of being received successfully by the receiver.

The throughput of nonpersistent ISMA with capture is defined as the ratio of the probability of success of the test packet, and expected length of the cycle assuming that there is an infinite population of users with the total arrival rate of (new plus inhibited, rescheduled) packets, G , is Poisson distributed.
Thus the throughput is:

$$S \triangleq \frac{P_{succes}(z_o)}{\overline{L_c}} \tag{4.2.1}$$

where the expected length of a cycle is defined as the sum of a busy period and the following idle period, as we can see in section 3.2.

$$\overline{L_c} \triangleq 1+2d+\frac{1}{G}\exp(-dG) \tag{4.2.2}$$

The probability of success is

$$P_{succes}(z_o)=\sum_{n=0}^{\infty} R_n(n+1)[1-Prob\{P_s/P_n<z_o\}] \tag{4.2.3}$$

and R_n , the probability of n interfering packets overlapping a test packet, is given by

$$R_n = \frac{(dG)^n}{n!}\exp(-dG) \tag{4.2.4}$$

So finally the throughput is

$$S = \frac{\sum_{n=0}^{\infty} (n+1)R_n[1-Prob\{\frac{P_s}{P_n}<z_o\}]}{1+2d+\frac{1}{G}\exp(-dG)} \tag{4.2.5}$$

4.3. Slotted nonpersistent ISMA with capture

Using the capture effect on the results derived in section 3.3, now not all the packets are annihilated anymore by collision.

Expression (3.3.2) is here, too:

$$U = \frac{B}{1+d} P_{suc} \tag{4.3.1}$$

Because we are now taking the capture effect in consideration, the probability of a succesful transmission period is

$$P_{suc} = \frac{dG}{1-\exp(-dG)} \sum_{n=0}^{\infty} R_n (n+1)[1-F_{Z_n}(z_o)] \tag{4.3.2}$$

Using (3.2.1), (3.3.1), (3.3.2), (4.3.1) and (4.3.2) we finally get the throughput for the slotted nonpersistent ISMA with capture

$$S = \frac{dG}{1+d-\exp(-dG)} \sum_{n=0}^{\infty} R_n (n+1)[1-F_{Z_n}(z_o)] \tag{4.3.3}$$

5. Throughput analysis of Rician channel with n-interferers

From different investigations, [9]-[12], it is known that in the presence of line of sight paths the multipath fading characteristics are Rician distributed. In a small cellular and indoor environment we have this situation [13],[14], whereas signals from far away, they are Rayleigh distributed. In the following sections we will see what kinds of effects the combination of different fading signals will have on the throughput.

5.1. Slotted ALOHA with Rice + n Rayleigh

We can imagine a Rice + n Rayleigh situation as follows. If within a large cell with a radius (r), there is a (mobile) terminal (T) nearby the base station (B) within a radius $0.2 \leq r < 1$ km, the signal coming from T is Rician distributed.
The n-interfering terminals (i) is far away from the base station for a distance to the base station greater than 1Km. Thus, the signals coming from these terminals are Rayleigh distributed. The situation is shown in Fig. 5.1.

The p.d.f. of a Rician signal is :

$$f_{Ri}(r) = \frac{r}{\sigma^2} \exp\left[-\frac{r^2+S^2}{2\sigma^2}\right] I_o\left[\frac{Sr}{\sigma^2}\right] \qquad 0 \leq r < \infty \qquad (5.1.1)$$

with r is the signal amplitude

σ^2 is the average of the fading signal

S is the peak value of the received signal

Because we are interested in the instantaneous power of test packet, the corresponding power p.d.f. becomes:

$$f(P_s) = \frac{1}{\sigma^2}\exp\left[-\frac{2P_s+S^2}{2\sigma^2}\right] I_o\left[\frac{\sqrt{2P_s}\ S}{\sigma^2}\right] \qquad (5.1.2)$$

with $P_s=1/2\ r^2$

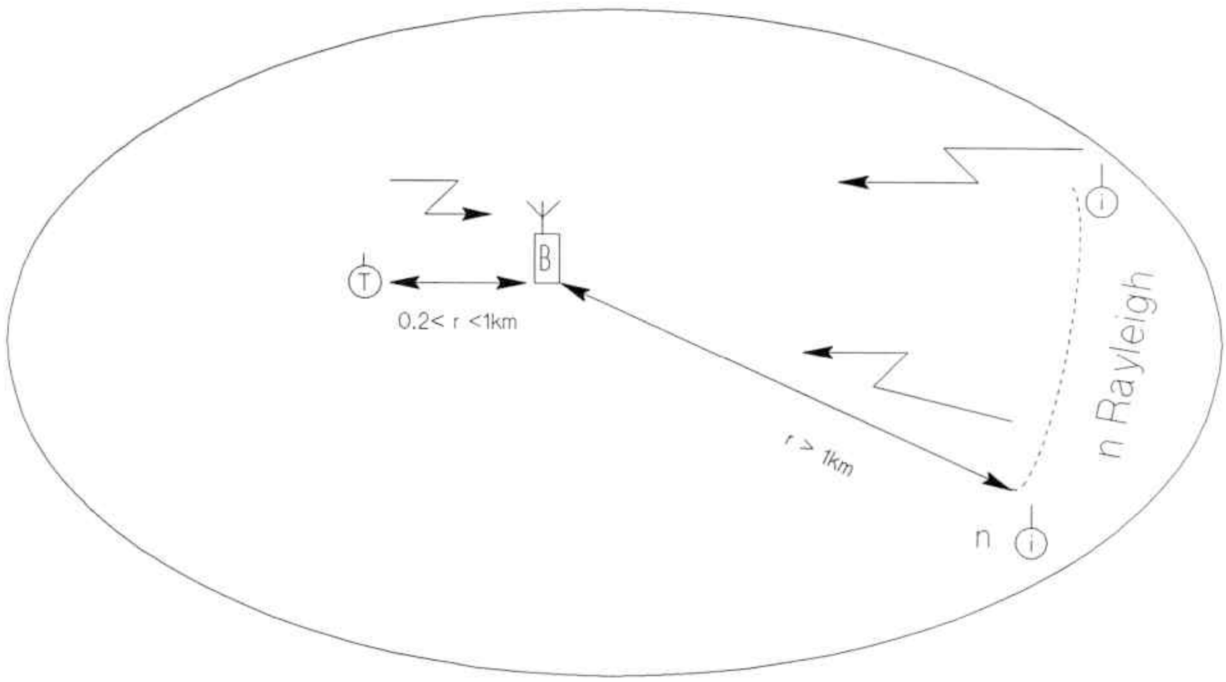


Fig. 5.1. n-Rayleigh interferers; T=desired terminal
i=interfering terminals, B=base station

The interfering signals from far away are Rayleigh distributed with a p.d.f.

$$f(r) = \frac{r}{P_o}\exp(-\frac{r^2}{2P_o}) \qquad (5.1.3)$$

with P_o is the mean power.

The corresponding power p.d.f. of one Rayleigh signal is

$$f_{P_i}(p_i) = \frac{1}{\overline{P}_i}\exp(-\frac{p_i}{\overline{P}_i}) \qquad (5.1.4)$$

By convolving this n-times we get a Gamma-distribution [3]

$$f_{P_n}(p_n) = \frac{1}{P_o}\frac{(p_n/P_o)^{n-1}}{(n-1)!}\exp(-\frac{p_n}{P_o}) \qquad (5.1.5)$$

Now we can use (4.1.9) to calculate the signal-to-interference distibution function (with $P_o=\sigma^2_I$).

$$F_{Z_n}(z_o) = \int_0^{z_o} dz \int_0^{\infty} \frac{1}{\sigma^2} \exp\left[-\frac{2zw+S^2}{2\sigma^2}\right] I_o\left[\frac{\sqrt{2zw}\,S}{\sigma^2}\right] \frac{1}{\sigma_I^2}\frac{(w/\sigma_I^2)^{n-1}}{(n-1)!}\exp(-\frac{w}{\sigma_I^2})\; w\, dw \qquad (5.1.6)$$

Let $t=w/\sigma_I^2$ and , $\epsilon=S/\sigma$ and $y=\surd(2zw)/\sigma$

$$F_{Z_n}(z_o) = \int_0^{\infty} dt \left[\int_0^{\sqrt{2z_o\frac{\sigma_I^2}{\sigma^2}t}} y \exp\left[-\frac{y^2+e^2}{2}\right] I_o[ey]\,dy\right] \frac{t^{n-1}}{(n-1)!}\exp(-t) \qquad (5.1.7)$$

This equation can be simplified, because we know that the integral between brackets is a Marcum's Q-function

$$[\ 1 - Q(\epsilon, \surd(2z_o t\sigma_I^2/\sigma^2))\ ] \qquad (5.1.8)$$

where

$$Q(\alpha,\beta) = \int_{\beta}^{\infty} x \exp[-\frac{1}{2}(x^2+\alpha^2)] I_o(\alpha x) dx \qquad (5.1.9)$$

So finally we get

$$F_{Z_n}(z_o) = \int_0^{\infty} \frac{t^{n-1}}{(n-1)!} \exp(-t) dt - \int_0^{\infty} \frac{t^{n-1}}{(n-1)!} \exp(-t) Q(\epsilon, \sqrt{2z_o \frac{\sigma_I^2}{\sigma^2} t}) dt \qquad (5.1.10)$$

The first integral is equal to 1 , and the second one we can calculate using the Laguerre-integration method [16].

In the numerical results we will use the parameters K and M, where

$$K = \frac{S^2}{2\sigma^2}$$

$$M \triangleq z_o \frac{\sigma_I^2}{\sigma^2} \qquad (5.1.11)$$

We have defined the parameter M, because we do not know what the value of ratio σ_I^2/σ^2 is.

A relation between K and M is found as follows. Writing the signal to interference ratio as :

$$\frac{P_s}{P_n} = \frac{\frac{S^2}{2} + \sigma^2}{\sigma_I^2} = \frac{\sigma^2}{\sigma_I^2}(K+1) \qquad (5.1.12)$$

Because we consider $P_s/P_n < z_o$ we write

$$\frac{\sigma^2}{\sigma_I^2}(K+1) < z_o \tag{5.1.13}$$

$$M > (K+1) \tag{5.1.14}$$

Before looking at the results, there is one remark that have to mention about the accuracy and the calculation time. For the calculation, we have used a 8086-computer with co-processor. The calculating time varies from one half hour to about five hours, with an average of three hours. The duration have to do with some combination of values, that causes a longer calculating time. Because of the long calculation we mostly have taken an accuracy of 1E-3.

For Fig. 5.1.1. we take a fixed K=4dB and σ_I^2/σ^2=2dB, and then increase z_o; we see that the throughput decreases. For the following figures we will only use the parameters K and M. We see from Fig. 5.1.2. that if M is increasing, the throughput is decreasing. If the Rice-factor is increasing, the throughput is increasing (Fig. 5.1.3).

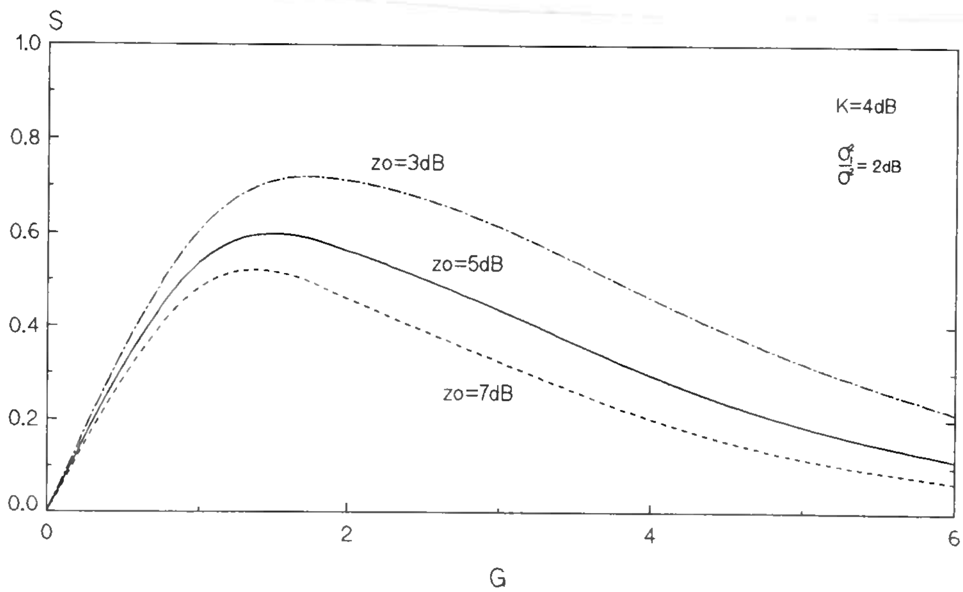


Fig. 5.1.1 The influence of z_o with fixed K and σ_I^2/σ^2

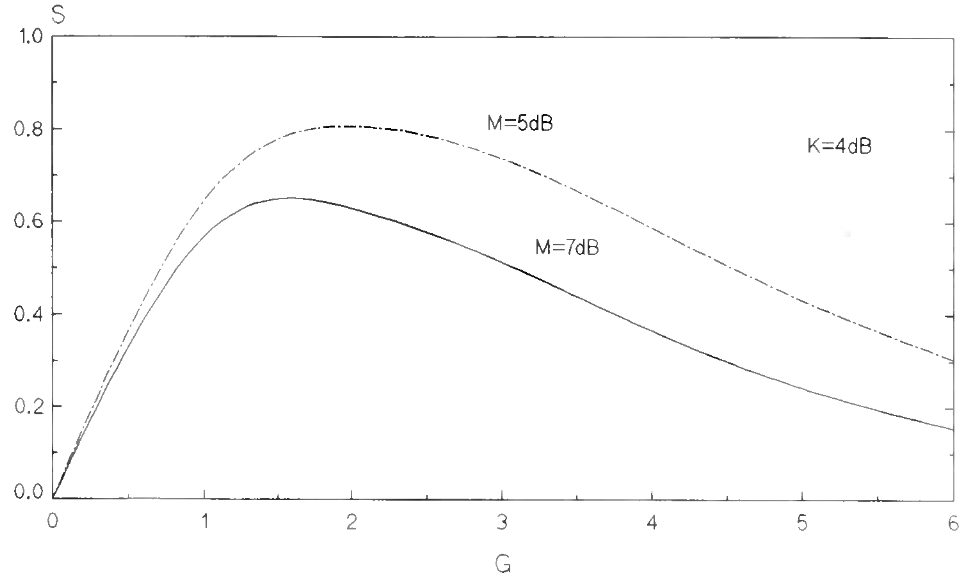


Fig. 5.1.2. The influence of the parameter M

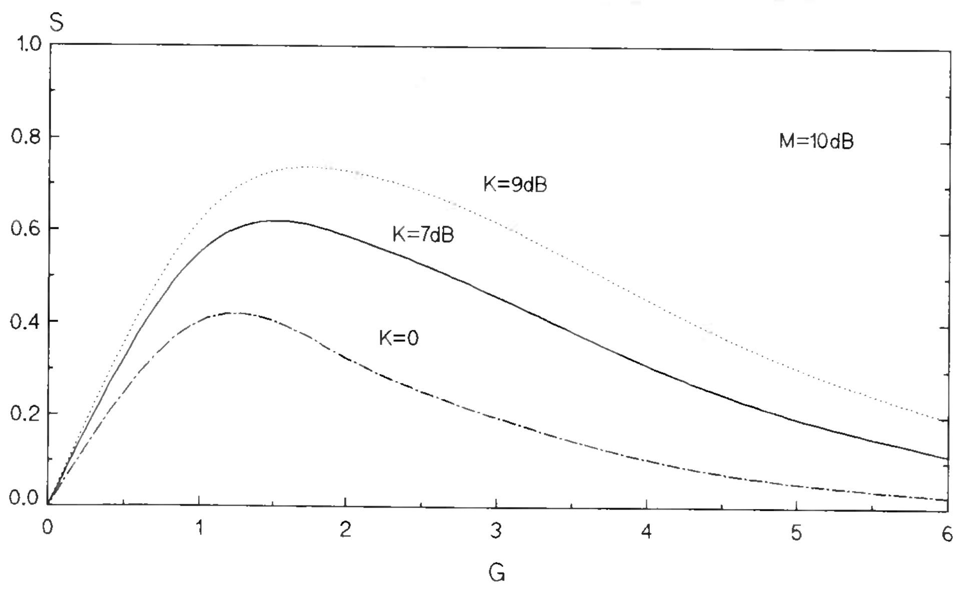


Fig. 5.1.3. Increasing throughput by increasing K

5.2. Slotted ALOHA with Rice + n Rice

Next we will investigate a Rice test signal with n-Rician interfering signals. This situation can be imagined as follows. Within a small cell with a radius $0.2 \leq r < 1$ km there is one base station (B) and a desired terminal (T) and n-interfering terminals (i). All signals are Rician distributed. This situation is shown in Fig. 5.2.

The p.d.f. of the instanteneous power of the test packet is :

$$f(P_s) = \frac{1}{\sigma^2} \exp\left[-\frac{2p_s + s^2}{2\sigma^2}\right] I_o\left[\frac{\sqrt{2P_s}\, S}{\sigma^2}\right] \tag{5.2.1}$$

with $P_s = 1/2\ r^2$.

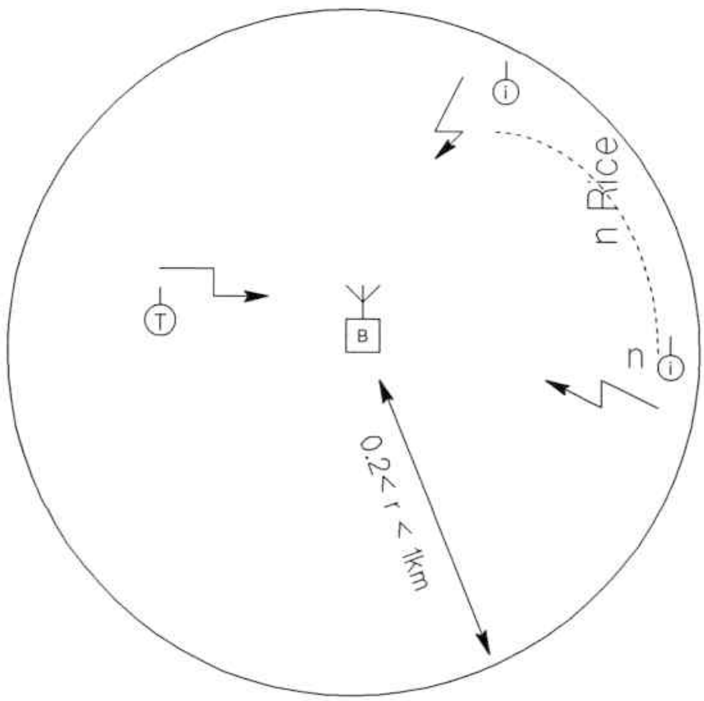


Fig. 5.2.1 n-Rice interferers; T=desired terminal, i=interfering terminals, B=base station

Now we need an expression for the sum of n-Rician signals.

According to van Trees [17, p.411], the sum of n-Rician independent signals is :

$$P_n = \sum_{i=1}^{n} p_i^2 \qquad (5.2.2)$$

The p.d.f. of the sum is

$$f_{P_n}(p_n) = \frac{1}{2\sigma^2}\left(\frac{p_n}{P}\right)^{\frac{n-1}{2}} \exp\left[-\frac{p_n+P}{2\sigma^2}\right] I_{n-1}\left[\frac{\sqrt{Pp_n}}{\sigma^2}\right] \qquad 0 \le P_n < \infty$$

(5.2.3)

$$with \quad P = \sigma^2 \sum_{i=1}^{N} S_i^2$$

But the expression above for P is not correct, if only for dimensional reasons. To find the p.d.f. of the sum of N signals we first have to find the charateristic function of the Rician p.d.f.. By definition the characteristic function of a p.d.f. is the Fourier transform of this p.d.f.. After multiplying the characteristic function N times and then calculating the inverse Fourier transform we found :

$$P = \sum_{i=1}^{N} S_i^2 \qquad (5.2.4)$$

The detail derivation can be found in [25].

Using (4.1.9), (5.2.1), (5.2.3) and (5.2.4) the distribution function becomes

$$F_{Z_n}(z_o) = \int_0^{z_o} dz \int_0^{\infty} \frac{1}{\sigma^2} \exp\left[-\frac{2zw+S^2}{2\sigma^2}\right] I_o\left[\frac{\sqrt{2zw}\,S}{\sigma^2}\right]$$

(5.2.5)

$$\frac{1}{2\sigma^2}\left(\frac{w}{P}\right)^{\frac{n-1}{2}} \exp\left[-\frac{w+P}{2\sigma^2}\right] I_{n-1}\left(\frac{\sqrt{Pw}}{\sigma^2}\right) w\, dw$$

By using the transformation $t=\surd(2zw)/\sigma$ and $\alpha=S/\sigma$
$x=\surd w/\sigma$ and $\epsilon=\surd P/\sigma$

we get

$$F_{Z_n}(z_o) = \int_0^{\sqrt{2z_o}\,x} t \exp\left[-\frac{t^2+\alpha^2}{2}\right] I_o[\alpha t]\,dt \int_0^{\infty} x\left(\frac{x}{\epsilon}\right)^{n-1} \exp\left[-\frac{x^2+\epsilon^2}{2}\right] I_{n-1}[\epsilon x]\,dx \qquad (5.2.6)$$

The first integral is a Marcum's Q-function. So we can write

$$F_{Z_n}(z_o) = [\ 1-Q(\alpha,\sqrt{2z_o}\ x)\ ] \int_0^{\infty} x \left(\frac{x}{\epsilon}\right)^{n-1} \exp[-\frac{x^2+\epsilon^2}{2}]\ I_{n-1}[\epsilon x]\ dx$$

$$= \int_0^{\infty} x \left(\frac{x}{\epsilon}\right)^{n-1} \exp(-\frac{x^2+\epsilon^2}{2})\ I_{n-1}[\epsilon x]\ dx \qquad (5.2.7)$$

$$- \int_0^{\infty} x \left(\frac{x}{\epsilon}\right)^{n-1} \exp(-\frac{x^2+\epsilon}{2})\ I_{n-1}[\epsilon x]\ Q(\alpha,\sqrt{2z_o}\ x)\ dx$$

The first integral is equal to *1* , and the second one we can integrate numerically. Before we do that we write the equation as follows:

$$F_{Z_n}(z_o) = 1-\frac{1}{\epsilon^{n-1}}\exp[-\frac{\epsilon^2}{2}]\int_0^{\infty} x^m \exp[-\frac{x^2}{2}]\ I_{n-1}[\epsilon x]\,Q(\alpha,\sqrt{2z_o}\ x)\ dx \qquad (5.2.8)$$

Using the relation

$$S = G\ [\ 1-\sum_{n=1}^{\infty} R_n\ F_{Z_n}(z_o)\ ] \qquad (5.2.9)$$

we can calculate the throughput.

For the calculation, different values of the Rice-factor and capture ratio are used. It is assumed that all interfering signals have the same average fading power, so $\sigma_i^2 = \sigma^2$. The Rician test signal have also the same σ^2 . Thus we neglect near-far effects. Because all interfering signals are identical, we can write P as follows:

$$P = nS^2 = n2\sigma^2K \quad , \text{and} \quad \epsilon = \surd(P)/\sigma = \surd(2nK) \qquad (5.2.10)$$

From Fig. 5.2.1 we see that if the capture ratio is increasing the throughput is getting lower. Fig. 5.2.2 shows that if the Rice-factor is increasing, the throughput is *decreasing.* In fact it is not so surprising, because we had assumed that the Rice-factor of the desired signal and the undesired signals are the same. So if the Rice-factor is getting higher the sum of interfering signals is having great influence.

We now assume that the Rice factor of the desired signals and the interfering signals are not the same. For the following figures we are using the following parameters:

K_d = desired Rice factor,
K_u = undesired Rice factor, and
z_o = the capture ratio.

In Fig. 5.2.3 we take K_d higher than K_u and increase z_o, which decreases the throughput. If we keep z_o and K_d fixed and increase K_u, Fig. 5.2.4, the throughput decreases. For Fig. 5.2.5. we take the inverse situation with K_u and z_o fixed. With a higher K_d the throughput is higher.

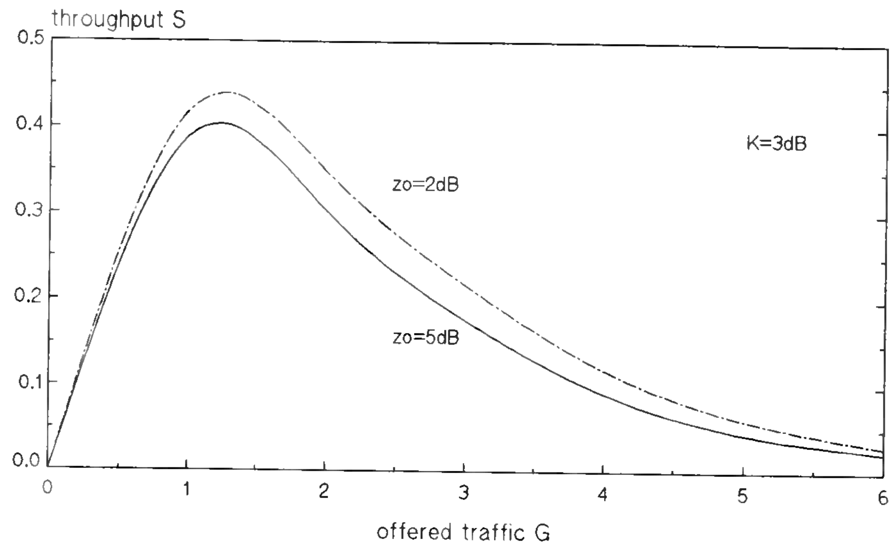


Fig. 5.2.1 Influence of z_o on the throughput with K=3dB

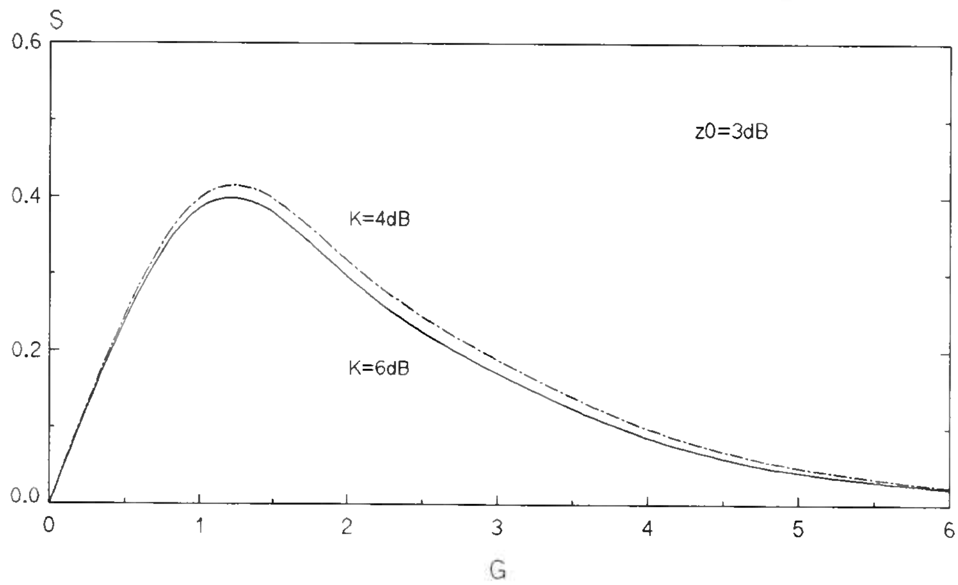


Fig. 5.2.2 Influence of K on the throughput with z_o=3dB

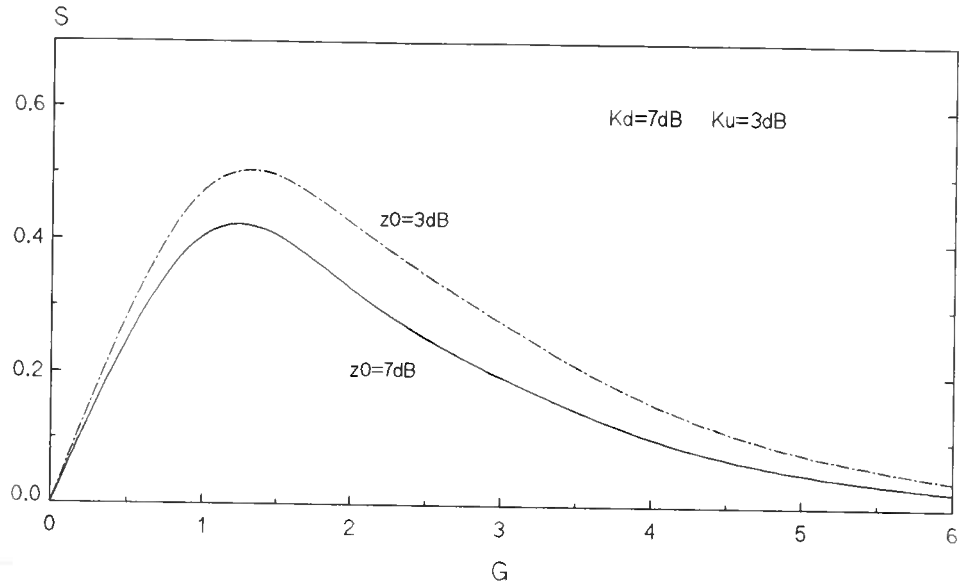


Fig. 5.2.3 Throughput by Kd=7dBand Ku=3dB with
z_o=3dB and z_o=7dB

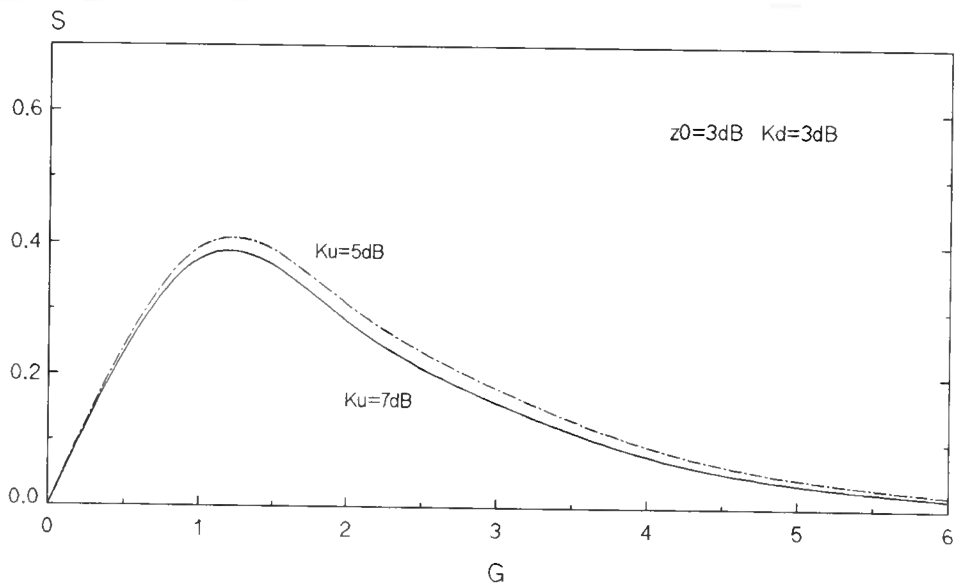


Fig. 5.2.4 Throughput by Kd=3dB and z_o=3dB with
Ku=5dB and Ku=7dB

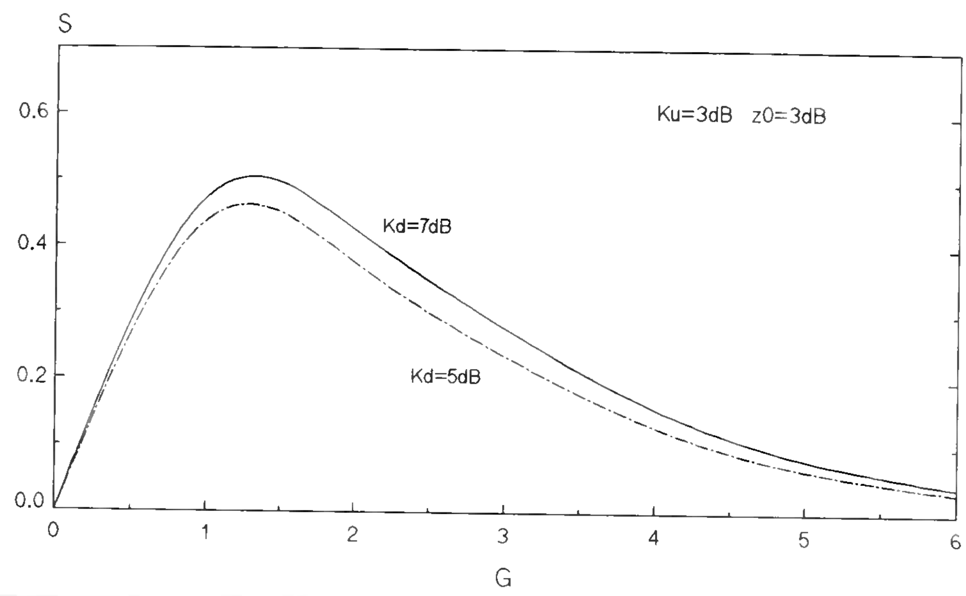


Fig. 5.2.5 Throughput by Ku=3dB and z_0=3dB with Kd=5dB and Kd=7dB

5.3. Unslotted np-ISMA with Rice + n Rayleigh

In section 4.2 an analysis is given for the nonpersistent ISMA protocol with capture. Here we want to investigate the influence of n Rayleigh distributed interfering signals on the nonpersistent ISMA throughput with a Rician distributed test signal. Therefore the results of section 5.1. are also useful.

So, according to (4.2.5) the throughput of nonpersistent ISMA is

$$S = \frac{\sum_{n=0}^{\infty} (n+1) R_n [1 - Prob\{\frac{P_s}{P_n} < z_o\}]}{1 + 2d + \frac{1}{G}\exp(-dG)} \tag{5.3.1}$$

In section 4.1. we had defined $F_{Z_n}(z_o)$ = Prob{ $P_s/P_n < z_o$ }, and the expression for Rice and n Rayleigh is already derived as

$$F_{Z_n}(z_o) = \int_0^{\infty} \frac{t^{n-1}}{(n-1)!}\exp(-t)dt - \int_0^{\infty} \frac{t^{n-1}}{(n-1)!}\exp(-t)\, Q(\epsilon, \sqrt{2z_o \frac{\sigma_I^2}{\sigma^2} t}\,)dt \tag{5.3.2}$$

Using (5.3.1) and (5.3.2) the throughput is given by

$$S = \frac{\sum_{n=0}^{\infty} \frac{(n+1)}{(n-1)!} R_n \int_0^{\infty} t^{n-1}\exp(-t)\, Q(\epsilon, \sqrt{2z_o \frac{\sigma_I^2}{\sigma^2} t}\,)dt}{1 + 2d + \frac{1}{G}\exp(-dG)} \tag{5.3.3}$$

After manupulating with (5.3.3) we get

$$S = \frac{\exp(-dG) + \sum_{n=1}^{\infty} \frac{(n+1)}{(n-1)!} R_n \int_0^{\infty} t^{n-1}\exp(-t)\, Q(\epsilon, \sqrt{2z_o \frac{\sigma_I^2}{\sigma^2} t}\,)dt}{1 + 2d + \frac{1}{G}\exp(-dg)} \tag{5.3.4}$$

The results for different values of the Rice-factor K, M ($= z_0\sigma_1^2/\sigma^2$) and d is given in the following figures. For the calculation we mostly have taken an accuracy of 1E-3, and the average calculating time is about three hours. Fig.5.3.1 shows the influence of the inhibit delay fraction d . The higher d is, the worse the results are. For Fig. 5.3.2. we take M=4dB, d=0.05 fixed and K=0 and 3dB. We see that if K increases, the throughput increases. For Fig. 5.3.3. M=4dB, d=0.5 and K=0 and 3dB. Compare Fig. 5.3.2. and Fig. 5.3.3. we see that the throughput is higher with a smaller d. That holds also for Fig. 5.3.4 and Fig. 5.3.5. If M increases the performance decreases.

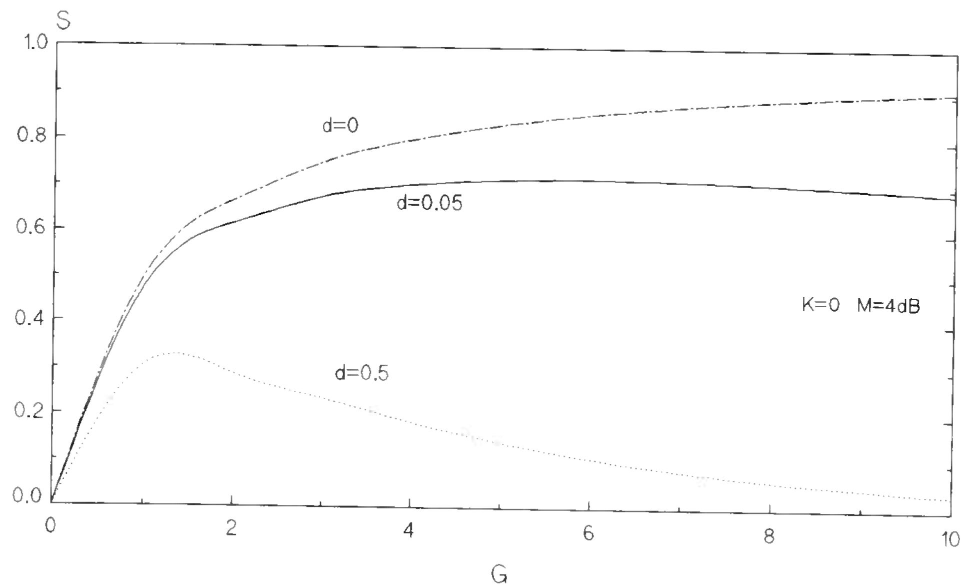


Fig. 5.3.1. The influence of d on the throughput.

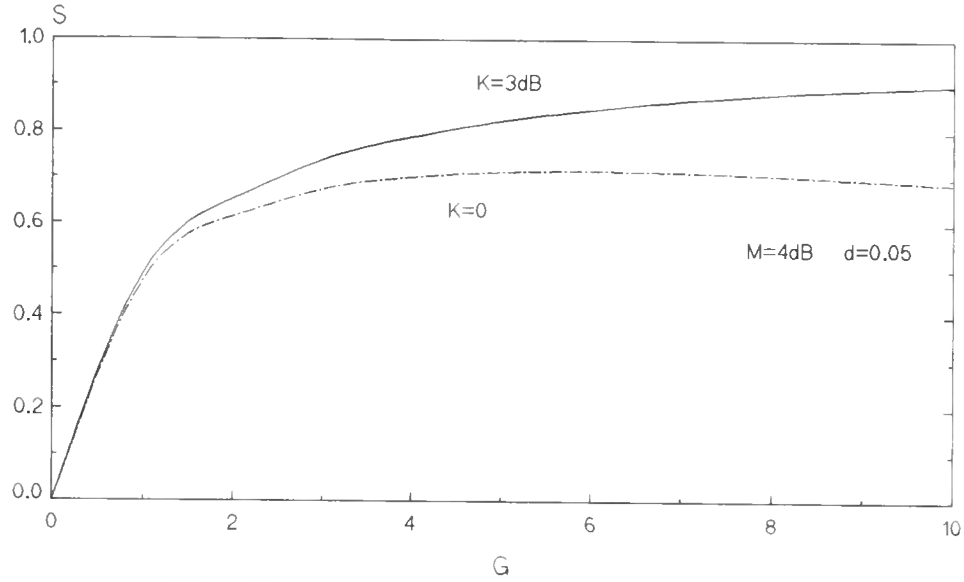


Fig. 5.3.2 The influence of the Rician factor on the throughput with M=4dB and d=0.05

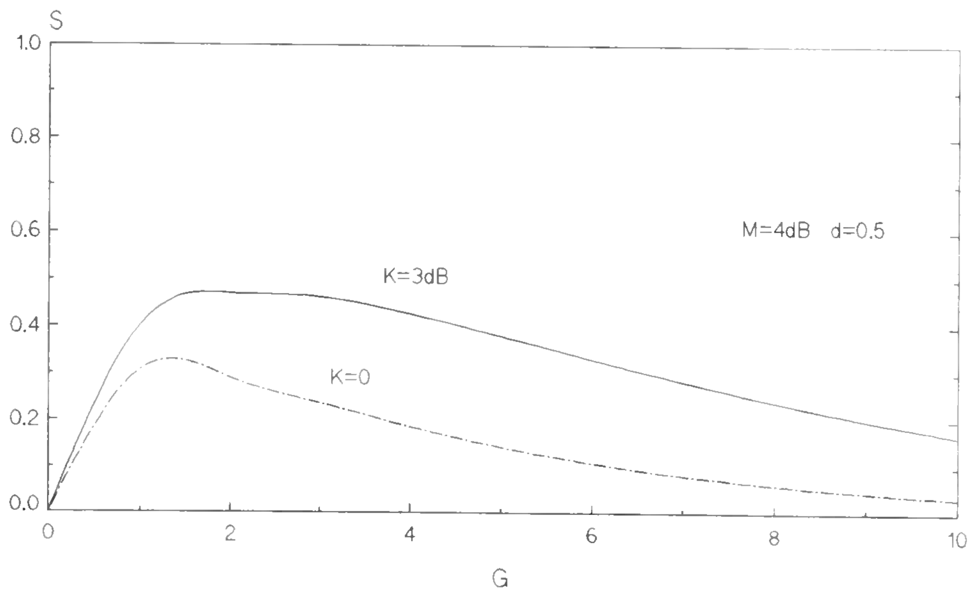


Fig.5.3.3 The influence of the Rice factor on the throughput with M=4dB and d=0.5

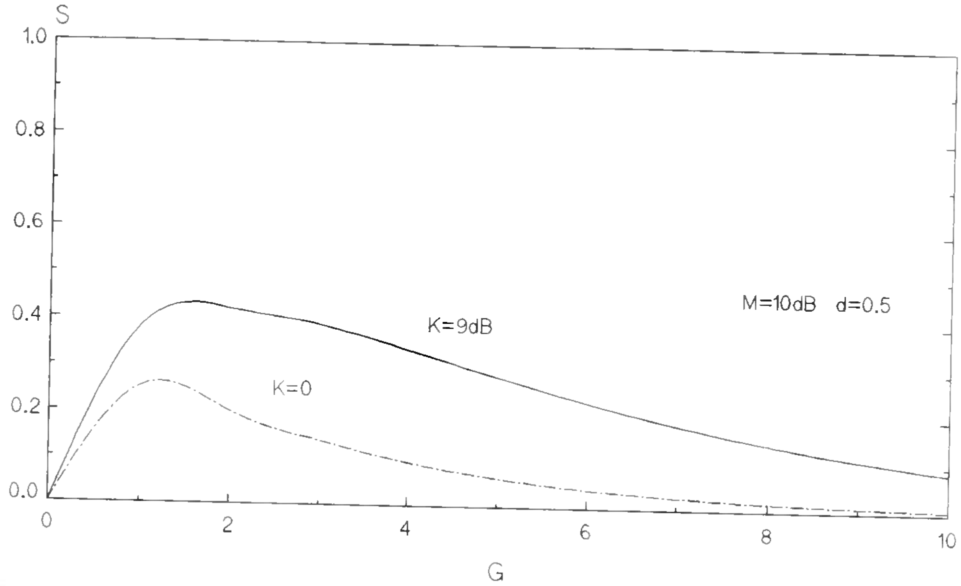


Fig. 5.3.4 The influence of the Rice factor on the throughput with M=10dB and d=0.5

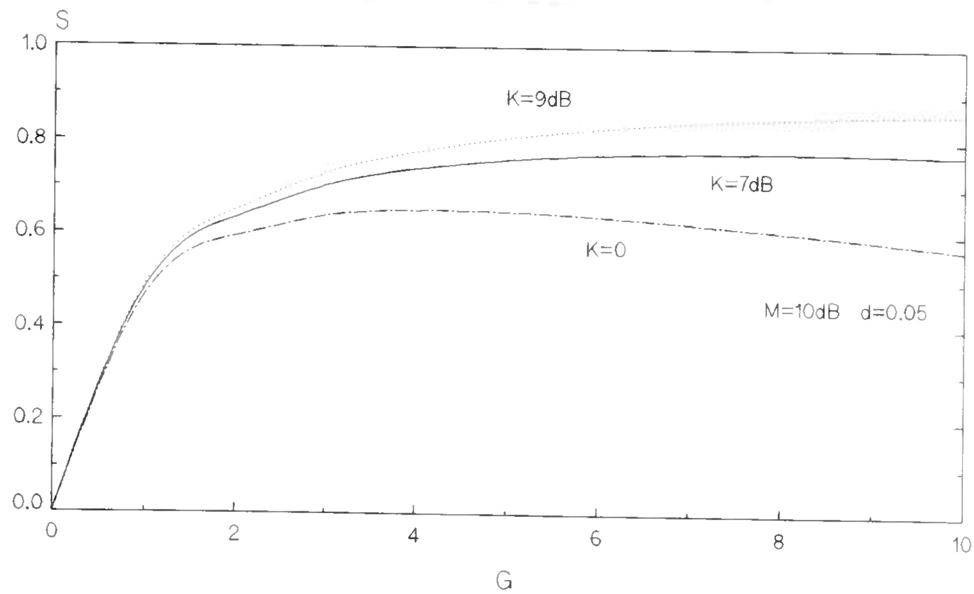


Fig. 5.3.5 The influence of the Rice factor on the throughput with M=10dB and d=0.05

5.4. Unslotted np-ISMA with Rice + n Rice

In the same way as in the previous section, we continue with the investigation of nonpersistent ISMA, only this time we are doing with a Rician distributed test signal and n-Rician distributed interfering signals.

The distribution function $F_{Z_n}(z_o)$ for this kind combination of fading is derived in section 5.2. For clearity it is given here again:

$$F_{Z_n}(z_o) = 1 - \frac{1}{\epsilon^{n-1}} \exp[-\frac{\epsilon^2}{2}] \int_0^\infty x^m \exp[-\frac{x^2}{2}] I_{n-1}[\epsilon x] Q(\alpha, \sqrt{2z_o}\, x) dx \qquad (5.4.1)$$

Using (4.2.5) the throughput is then

$$S = \frac{\sum_{n=0}^{\infty} \frac{(n+1)}{\epsilon^{n-1}} R_n \exp[-\frac{\epsilon^2}{2}] \int_0^\infty x^m \exp[-\frac{x^2}{2}] I_{n-1}[\epsilon x] Q(\alpha, \sqrt{2z_o}\, x) dx}{1 + 2d + \frac{1}{G}\exp(-dG)} \qquad (5.4.2)$$

After manipulating with the summation we obtain a computable expression for the throughput

$$S = \frac{\exp(-dG) + \sum_{n=1}^{\infty} \frac{(n+1)}{\epsilon^{n-1}} R_n \exp[-\frac{\epsilon^2}{2}] \int_0^\infty x^m \exp[-\frac{x^2}{2}] I_{n-1}[\epsilon x]\, Q(\alpha, \sqrt{2z_o}\, x) dx}{1 + 2d + \frac{1}{G}\exp(-dG)} \qquad (5.4.3)$$

Fig. 5.4.1 shows the influence of the capture ratio z_o on the throughput with K_d=7dB, K_u=3dB and d=0.05. If z_o increases, the throughput decreases. Fig. 5.4.2 shows the same situation, but now with d=0.5. For Fig. 5.4.3 and Fig. 5.4.4 we take K_d=K_u=3dB and 7dB with fixed z_o and d; increasing K_d=K_u will decrease the throughput. For Fig. 5.4.5 the fixed values are: z_o=3dB, K_d=3dB, d=0.05; if K_u increases the throughput decreases. An inverse situation is happening in Fig. 5.4.6; keeping z_o, K_u and d fixed with increasing K_d will increase the throughput. And Fig. 5.4.7 shows that if z_o increases, the throughput decreases with fixed K_d=K_u=7dB and d=0.05. We conclude that if d is reduced, the performance improves.

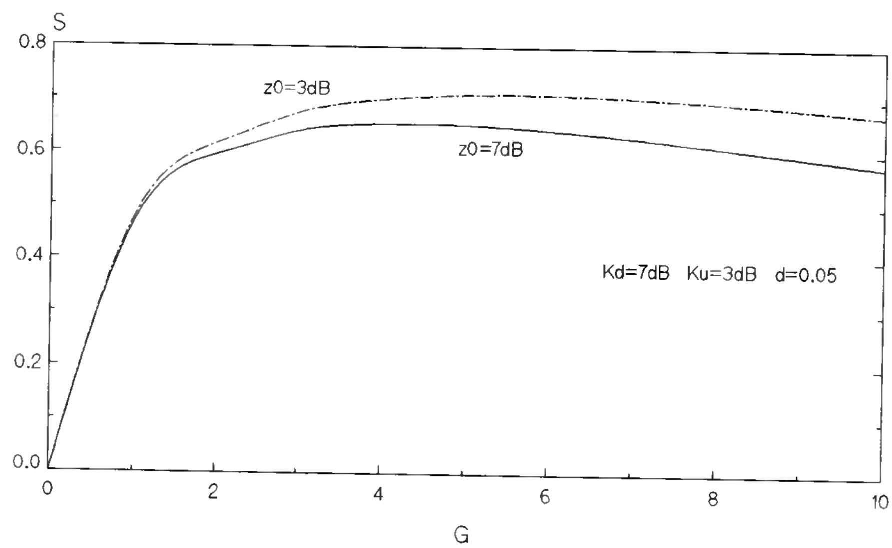


Fig. 5.4.1 The influence of the capture ratio on the throughput with Kd=7dB, Ku=3dB and d=0.05

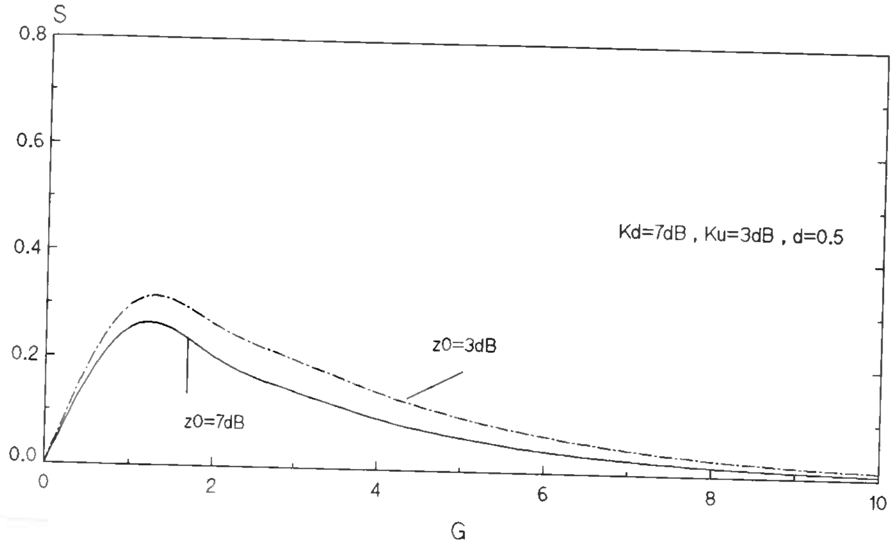


Fig. 5.4.2 The influence of the capture ratio on the throughput with Kd=7dB, Ku=3dB and d=0.5

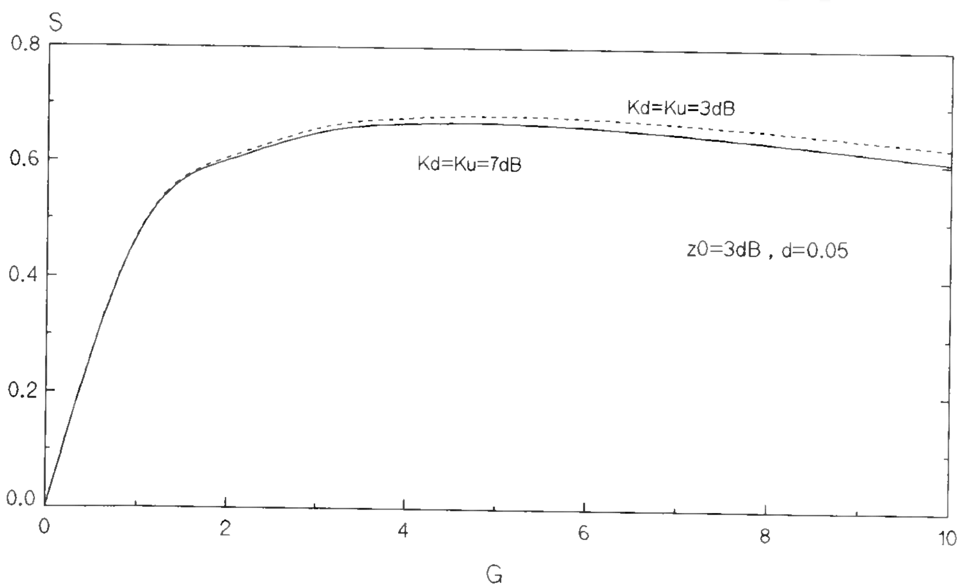


Fig. 5.4.3 The throughput by Kd=Ku with zo=3dB and d=0.05

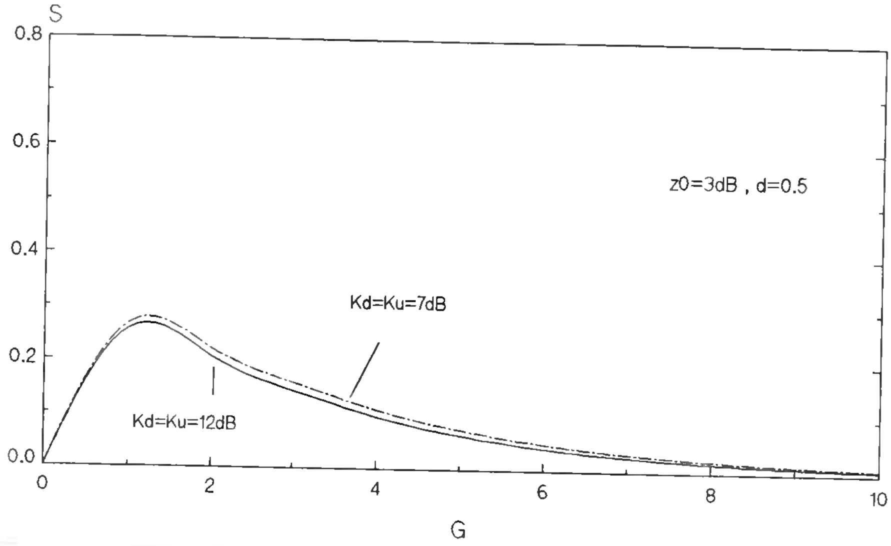


Fig. 5.4.4 The throughput by Kd=Ku with zo=3dB and d=0.5

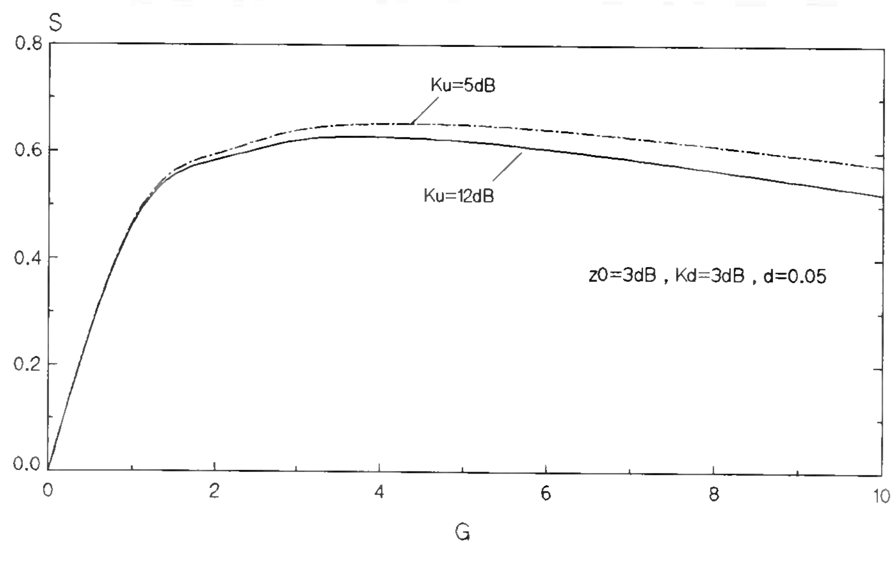


Fig. 5.4.5 The influence of Ku with zo=3dB, Kd=3dB and d=0.05

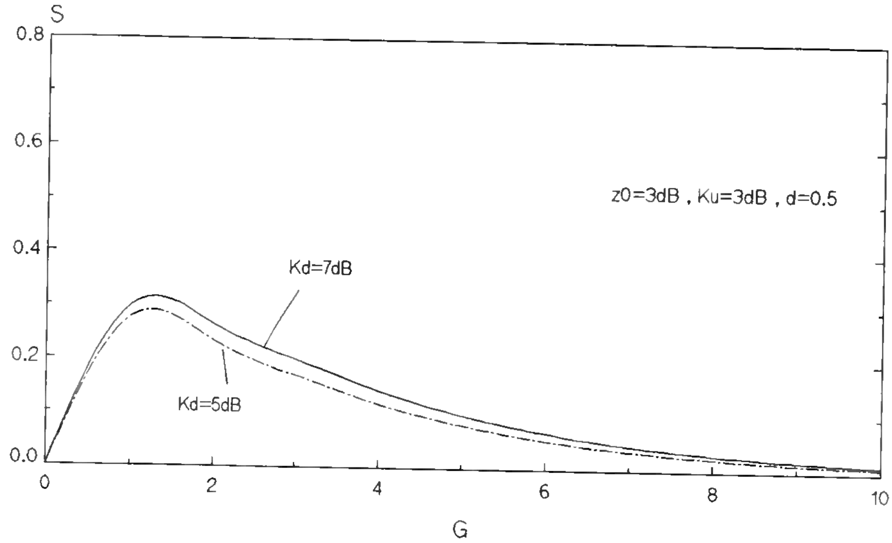


Fig. 5.4.6 The influence of Kd with zo=3dB, Ku=3dB and d=0.5

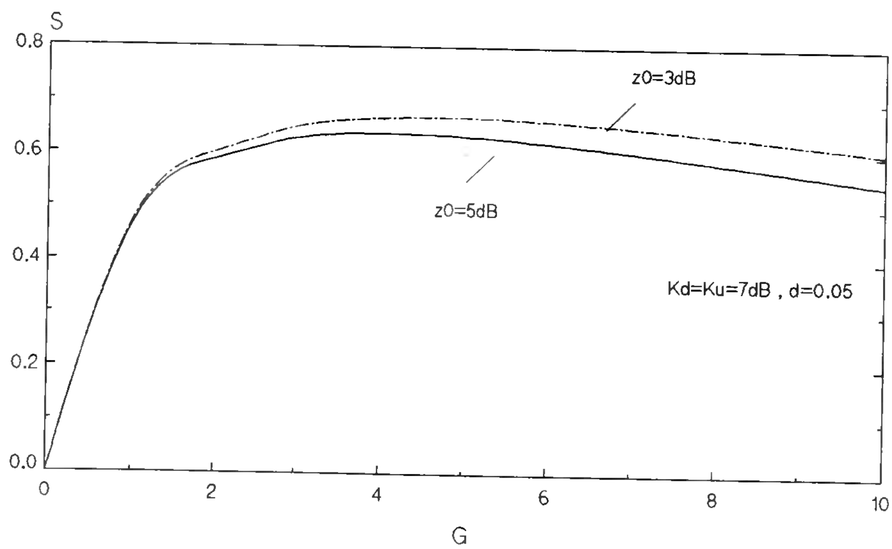


Fig. 5.4.7 The influence of zo with Kd=Ku=7dB and d=0.05

5.5. Slotted np-ISMA with Rice + n Rice

For slotted np-ISMA we will not consider the Rice + n Rayleigh situation but only for the Rice + n Rice situation, since the latter is the realistic situation. In chapter 6 we shall compare all the multiple access sytems with each other.

The expression for the throughput is given here again:

$$S = \frac{dG}{1+d-\exp(-dG)} \sum_{n=0}^{\infty} R_n (n+1)[1-F_{Z_n}(z_o)] \tag{5.5.1}$$

where R_n is given in (4.2.4) and $F_{Zn}(z_o)$ in (5.4.1).
Fig. 5.5.1 shows a decreasing throughput with increasing z_o for fixed K_d, K_u and d. This is also happening in Fig. 5.5.2, but with d=0.5. With z_o=3dB and d=0.05 (Fig. 5.5.3) and increasing K_d=K_u, the throughput is decreasing. Fig. 5.5.4 shows the same behaviour with d=0.5. For Fig. 5.5.5. we fix z_o=3dB, K_d=3dB and d=0.05; if K_u increases, the throughput decreases. An inverse situation can be seen in Fig. 5.5.6; by increasing K_d, the throughput increases. Fig. 5.5.7 shows a decreasing throughput by increasing z_o.

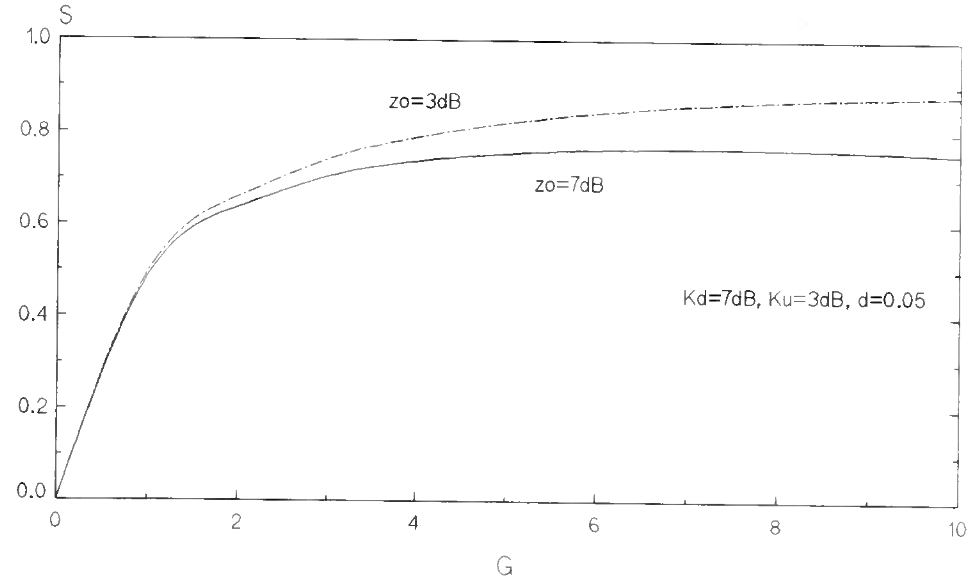


Fig. 5.5.1. Throughput of slotted ISMA with Kd=7dB, Ku=3dB, d=0.05 for zo=3dB and zo=7dB

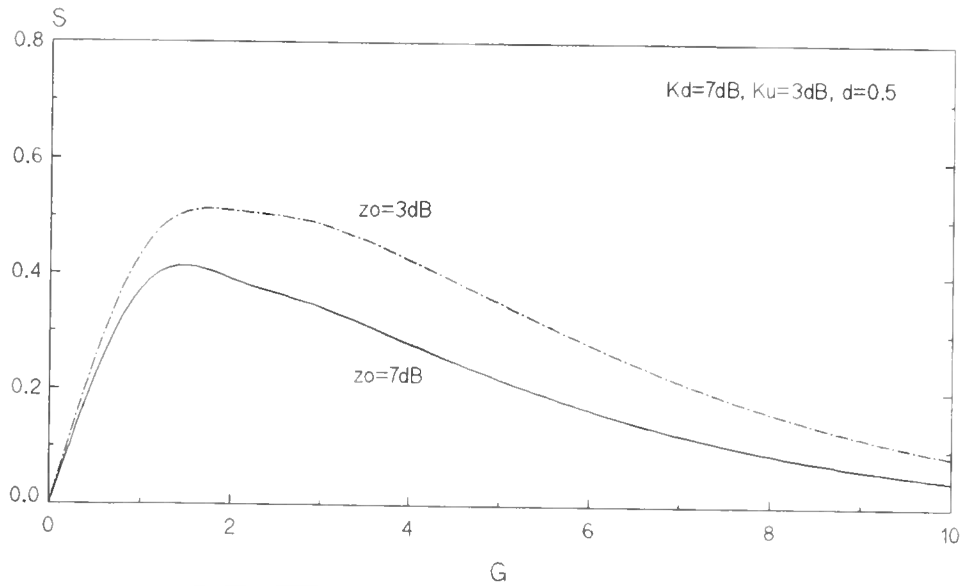


Fig.5.5.2. Throughput of slotted ISMA with Kd=7dB, Ku=3dB, d=0.5 for zo=3dB and zo=7dB

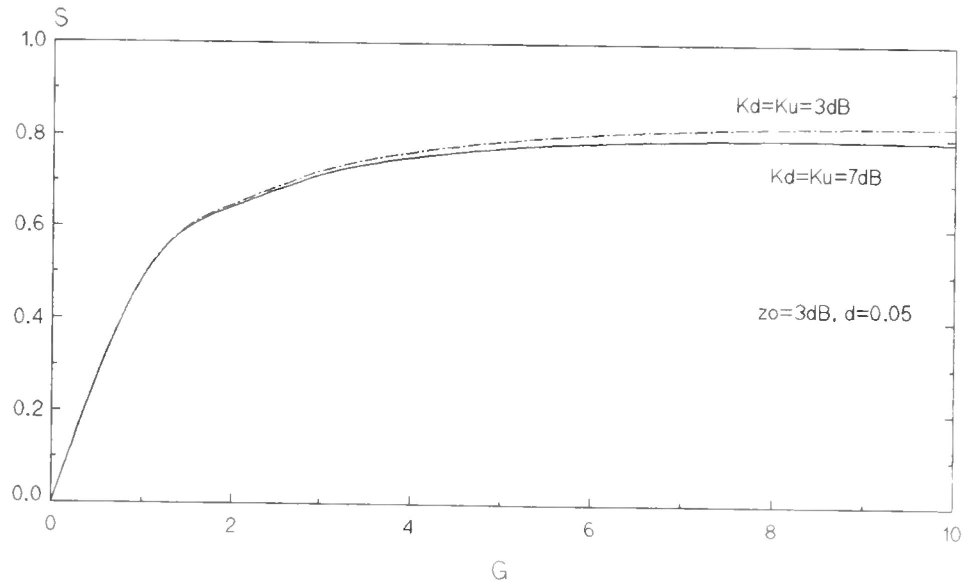


Fig. 5.5.3. Throughput of slotted ISMA with zo=3dB, d=0.05 for Kd=Ku=3dB and 7dB

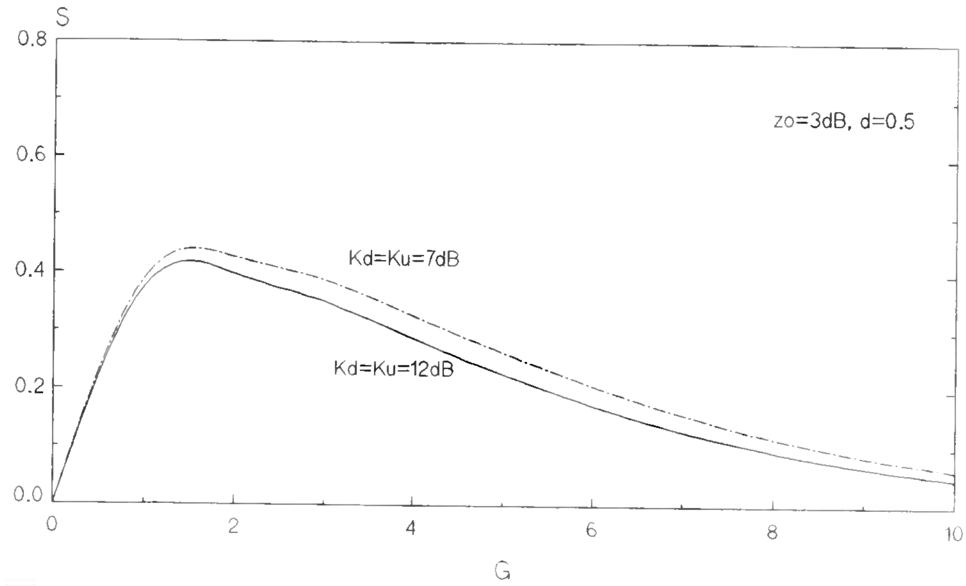


Fig. 5.5.4. Throughput of slotted ISMA with zo=3dB, d=0.5 for Kd=Ku=7dB and 12dB

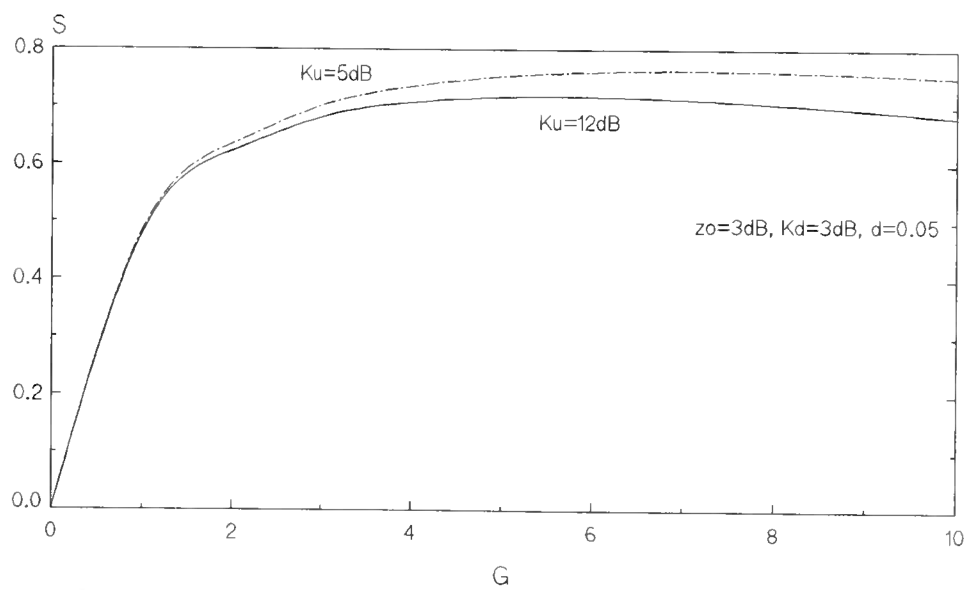


Fig. 5.5.5. Throughput of slotted ISMA with zo=3dB, Kd=3dB, d=0.05 for Ku=5dB and 12dB

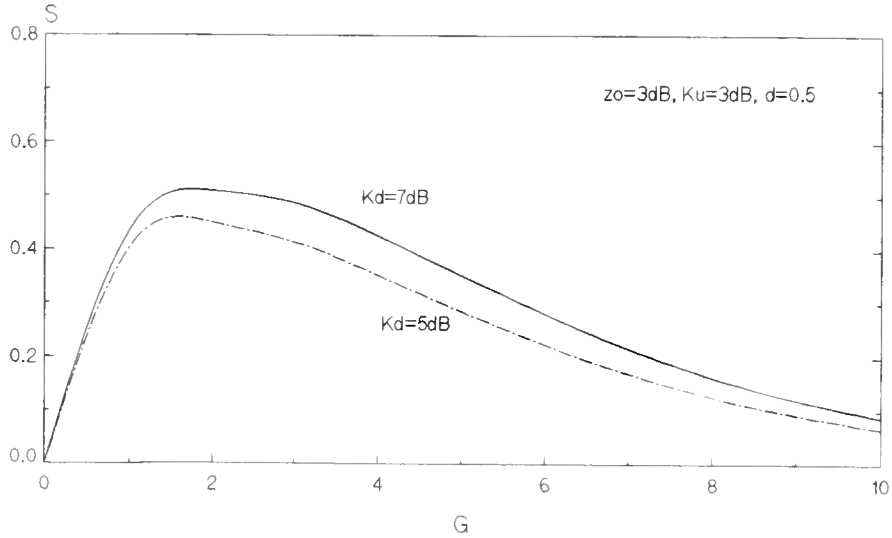


Fig. 5.5.6. Throughput of slotted ISMA with zo=3dB, Ku=3dB, d=0.5 for Kd=5dB and 7dB

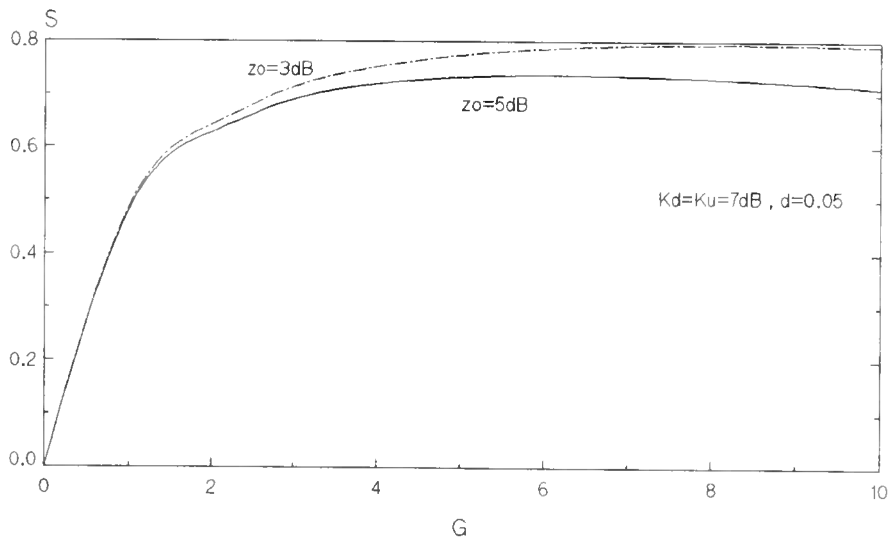


Fig. 5.5.7. Throughput of slotted ISMA with d=0.05, Kd=Ku=7dB for zo=3dB and 5dB

6. Comparison of the multiple access sytems

Given the results for slotted ALOHA, unslotted and slotted np-ISMA in the previous sections, it will be very interesting to see what the diffences are between these multiple access systems. First a comparison for the situation of Rice and n Rayleigh is given and then Rice and n Rice.

6.1. Comparison for Rice + n Rayleigh

The results of section 5.1. for slotted ALOHA obtain in (5.1.9) are compared with the results obtain for unslotted nonpersistent ISMA in (5.3.4) is given in Fig. 6.1.1 and Fig. 6.1.2. First we will take $d=0.05$. In Fig. 6.1.1 it can be seen that for light traffic, slotted ALOHA is better than unslotted nonpersistent ISMA. But the performance of slotted ALOHA is decreasing very rapidly, while unslotted nonpersistent ISMA is still increasing slightly. Besides the high throughput stays nearly constant for about $G \approx 19$ (although not plotted).

Fig. 6.1.2 : if $d=0.5$ the performance of slotted ALOHA at low traffic load is much higher (nearly two times) than unslotted nonpersistent ISMA, whereas by high traffic loads the opposite is happening.

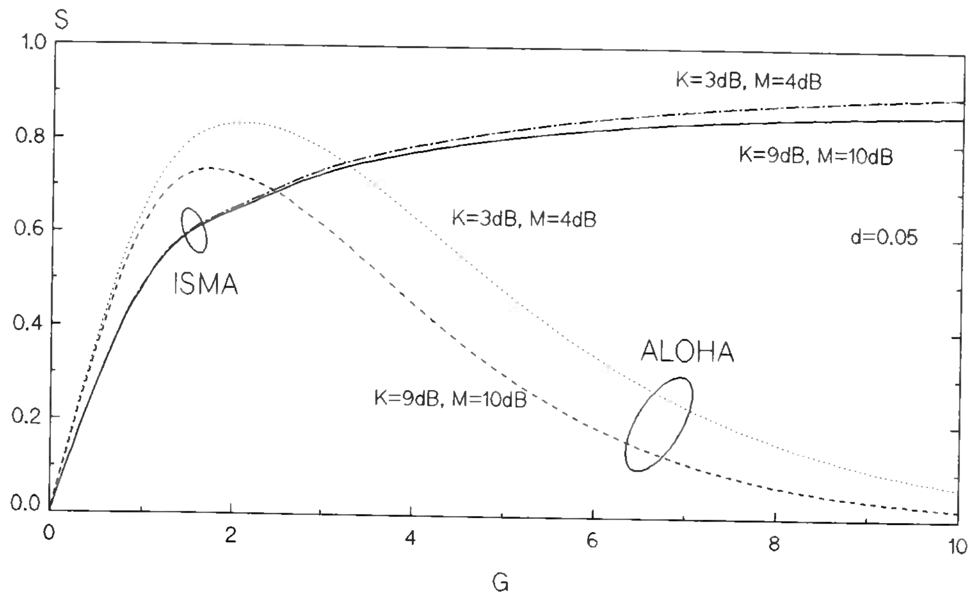


Fig. 6.1.1 Slotted ALOHA versus unslotted np-ISMA with d=0.05

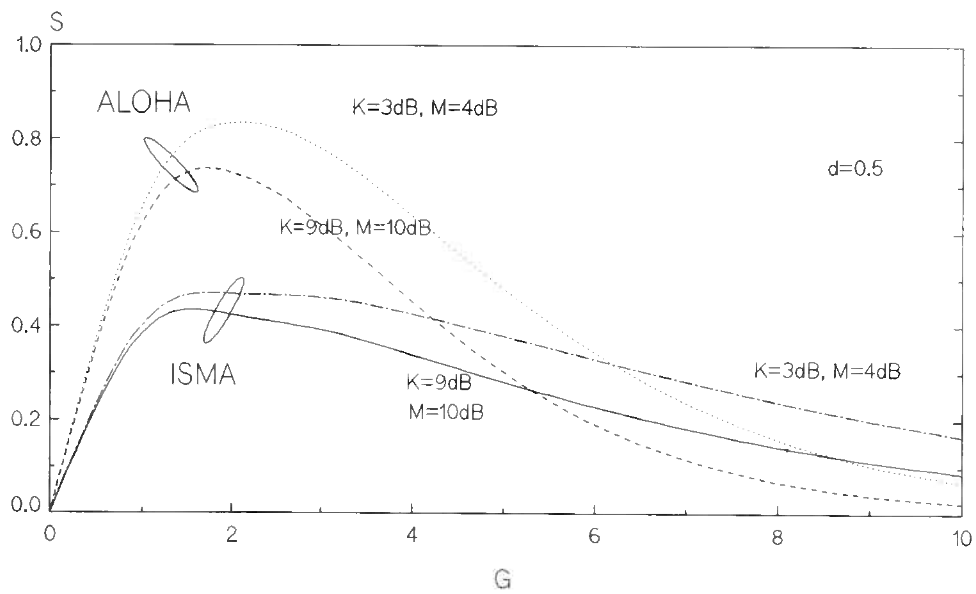


Fig. 6.1.2 Slotted ALOHA versus unslotted np-ISMA with d=0.5

6.2. Comparison for Rice + n Rice

In the situation for Rice + n-Rice the performances between slotted ALOHA and unslotted nonpersistent ISMA are different from the results in the previous section.

First we will again take d=0.05. From Fig. 6.2.1 it can be seen that unslotted nonpersistent ISMA is better than slotted ALOHA; besides, the throughput of unslotted ISMA is just decreasing while that of slotted ALOHA is nearly zero. At high traffic loads it is always better to use unslotted nonpersistent ISMA .

The opposite situation is obtained for d=0.5. From Fig. 6.2.2 we see that by low traffic density slotted ALOHA is nearly two times better than unslotted nonpersistent ISMA, while by high traffic density the performances are even bad. So by a higher inhibit delay fraction *d* , it is better to use slotted ALOHA. But for Rice + n Rice is d=0.5 not realistic.

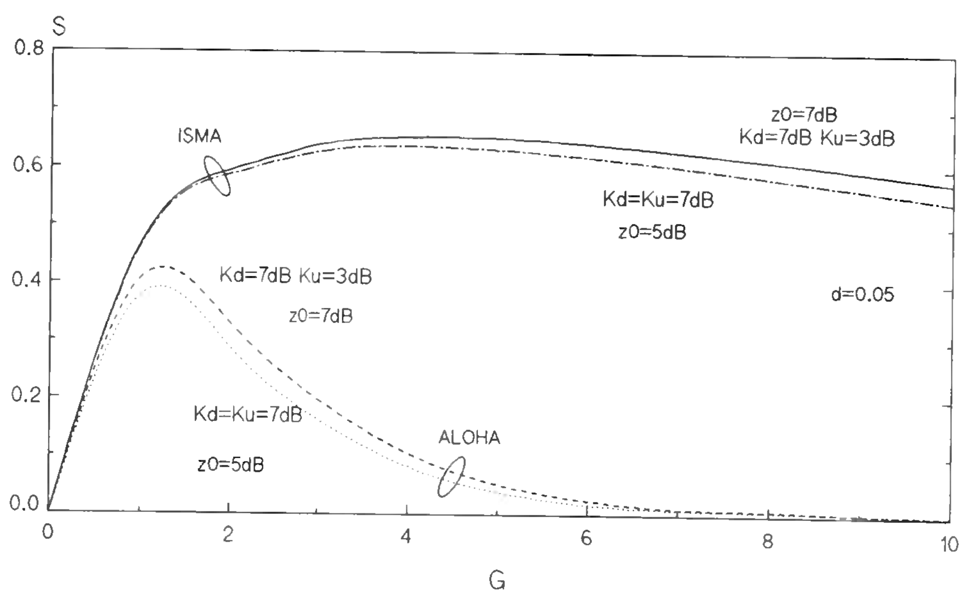


Fig. 6.2.1 slotted ALOHA versus unslotted np-ISMA with d=0.05

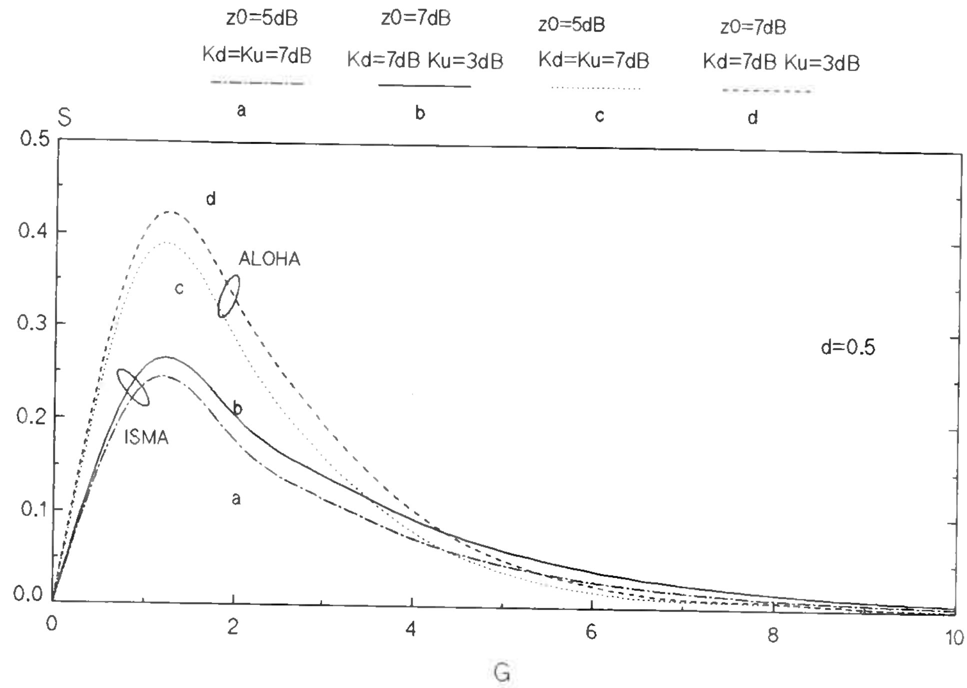


Fig. 6.2.2 slotted ALOHA versus unslotted np-ISMA with d=0.5

6.3. Comparison slotted ALOHA ↔ unslotted np-ISMA ↔ slotted np-ISMA for Rice + n Rice

For slotted np-ISMA we have not investigated the Rice + n Rayleigh situation. We have only done for the realistic situation Rice + n Rice. So, in this section we shall compare the investigated systems with each other for the Rice + n Rice situation.

In the following two figures we can see the three protocols in one figure. The first one is calculated with an inhibit delay d=0.05.

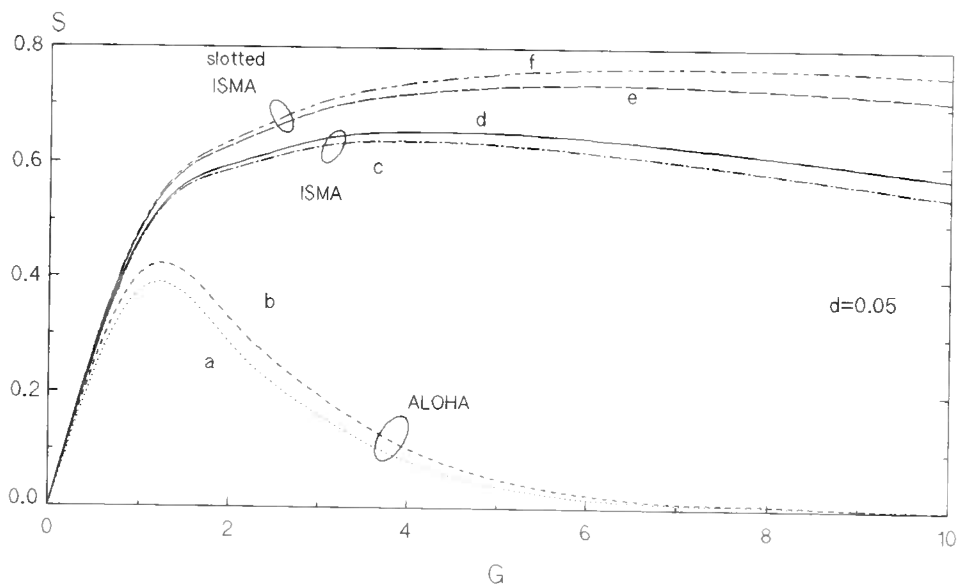


Fig. 6.3.1. Comparison ALOHA-ISMA-slotted ISMA with d=0.05

ALOHA	(a) Kd=Ku=7dB, zo=5dB
	(b) Kd=7dB, Ku=3dB, zo=7dB
ISMA	(c) Kd=Ku=7dB, zo=5dB
	(d) Kd=7dB, Ku=3dB, zo=7dB
slotted ISMA	(e) Kd=Ku=7dB, zo=5dB
	(f) Kd=7dB, Ku=3dB, zo=7dB

From the results it can be stated that both unslotted np-ISMA and slotted np-ISMA the always perform better than slotted ALOHA; slotted np-ISMA is better than unslotted np-ISMA.

The second figure is calculated with d=0.5.

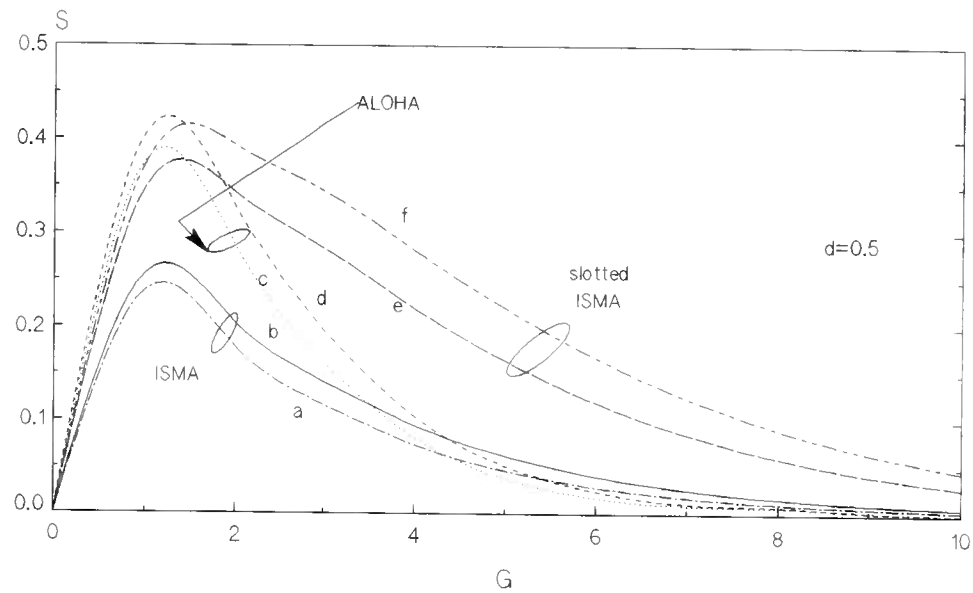


Fig. 6.3.2. Comparison ALOHA-ISMA-slotted ISMA for d=0.5

ISMA	(a) Kd=Ku=7dB, zo=5dB
	(b) Kd=7dB, Ku=3dB, zo=7dB
ALOHA	(c) Kd=Ku=7dB, zo=5dB
	(d) Kd=7dB, Ku=3dB, zo=7dB
slotted ISMA	(e) Kd=Ku=7dB, zo=5dB
	(f) Kd=7dB, Ku=3dB, zo=7dB

At low traffic loads slotted ALOHA is a little better than slotted np-ISMA, but from the results we may conclude that slotted np-ISMA is much better than slotted ALOHA. And ALOHA is in turn better than unslotted np-ISMA for d=0.5.

7. Conclusions and recommendations

From the interesting results obtained in this investigation of slotted ALOHA, unslotted nonpersistent ISMA and slotted nonperdrawnsistent ISMA, the following conclusions are drawn.

i) Slotted ALOHA

- For the situation Rice + n Rayleigh we have found the relation: M > K + 1, with $M=z_0\sigma_I^2/\sigma^2$ and $K=S^2/2\sigma^2$.
- By increasing M , the throughput is getting lower and increasing K the throughput is getting higher.
- For the situation of Rice + n Rice we had first assumed that the desired and undesired Rice-factor, respectively K_d and K_u, are identical ($K_d=K_u=K$). In this special case, for a fixed z_o and increasing of K , the throughput is decreasing. That occurs because the summation of the interferers have more influence.
- If the Rice-factors are equal and z_o is increasing, the results are worse, which is easily understood.
- In case of $K_u > K_d$ and fixed K_d and z_o , the throughput is getting lower by increasing K_u.
- For $K_d > K_u$ with a fixed K_u and z_o and increasing K_d, the performance is better.

ii) Unslotted np-ISMA

- We have stated that for the case Rice + n Rayleigh with the unslotted np-ISMA the results are better for a small inhibit delay fraction and a high K.
- The condition M>K holds here, too.
- For the situation Rice + n Rice the results are again better for a small inhibit delay fraction.
- Further what we have found for slotted ALOHA, they are also valid here.

iii) Slotted np-ISMA

- All results for small d are better than for a high d.
- Further what has stated for slotted ALOHA for Rice + n Rice, is also valid here.

iv) Comparison slotted ALOHA ↔ unslotted np-ISMA ↔ slotted np-ISMA

a) Rice + n Rayleigh case

- By comparing the results of slotted ALOHA and unslotted np-ISMA we conclude that in case of Rice + n Rayleigh, the use of slotted ALOHA is better than unslotted np-ISMA, both for d=0.05 as by d=0.5, if the traffic load is small.
- Conversely the use of unslotted np-ISMA is recommended at high traffic loads.

b) Rice + n Rice case

- In case of Rice + n Rice by a high value of d (e.g. d=0.5) the use of slotted ALOHA is better than unslotted np-ISMA. But slotted np-ISMA is recommended because the performance is better than slotted ALOHA.
- For small d (e.g. d=0.05), the use of unslotted np-ISMA is always much better than slotted ALOHA. And slotted np-ISMA is better than unslotted np-ISMA.

Recommendations

We have seen while comparing between slotted ALOHA and unslotted np-ISMA, that there is a point where they cross with each other and where unslotted ISMA is begin to perform better than slotted ALOHA. So it will be very interesting to find a bound where the one can take over the other.

Further it may also be interesting to include near-far effect and shadowing in the model.

References

[1] **N. Abramson,**
" The Throughput of Packet Broadcasting Channels ",
IEEE Transaction in Communication, vol. COM-25, pp.117-128, January 1977.

[2] **A.S. Tanenbaum,**
Computer Networks, New York: Prentice-Hall, Inc., Englewood Cliffs, 1988.

[3] **J.C. Arnbak and W. van Blitterswijk,**
" Capacity of Slotted ALOHA in Rayleigh-Fading Channels ",
IEEE Journal on Selected Areas in Communications, vol.SAC-5, no.2, pp.261-269, February 1987.

[4] **C. van der Plas and J.P.M.G. Linnartz,**
" Stability of mobile slotted ALOHA network with Rayleigh fading, shadowing and near-far effects ", IEEE Trans. on Vehicular Technology, vol. 39, pp. 359-366, November 1990.

[5] **R. Prasad and J.C. Arnbak,**
" Effects of Rayleigh fading on packet radio channels with shadowing ",
Proceedings IEEE, Tencon 1989, Bombay, India, pp.27.4.1-27.4.3, November 1989.

[6] **R. Prasad and J.C. Arnbak,**
" Enhanced packet throughput in radio channels with fading and shadowing ",
Proc. Canadian Conf. in Electrical and Computer Engineering, Vancouver, Nov. 1988, pp. 78-80.

[7] **R. Prasad and J.C. Arnbak,**
" Enhanced throughput in packet radio channels with shadowing ",
Electronics Letters, Vol. 24, pp. 986-988, August 1988.

[8] **V.O.K. Li,**
" Multiple Access Communications Networks ",
IEEE Communications Magazine, vol. 25, no.6, pp. 41-48, June 1987.

[9] **A.M.D. Turkmani, J.D. Parsons, Feng Ju and D.G. Lewis,**
" Microcellular radio measurements at 900, 1500 and 1800 MHz ",
Fifth International Conference on Mobile Radio and Personal Communications,
Warwich

[10] **R.J.C. Bultitude and G. Keith Bedal,**
" Propagation Characteristic on Microcellular Urban Mobile Radio Channels at 910 MHz ",
IEEE Journal on Selected Areas in Communications, vol.7, no.1, January 1989.

[11] **P. Harley,**
" Short distance attenuation measurements at 900 MHz and 1.8 MHz using low antenna heights for microcells ",
IEEE Journal on Selected Areas in Communications, vol.7, no.1, January 1989.

[12] **E.Green,**
" Path loss and signal variability analysis for microcells ",
Fifth International Conference on Mobile Radio and Personal Communications,
University of Warwich, 11-14 December 1989.

[13] **R. J. C. Bultitude,**
" Measurement, Characterization and Modeling of Indoor 800/900 MHz Radio Channels for Digital Communications ",
IEEE Communications Magazine, vol. 25, no. 6, pp. 5-12, June 1987.

[14] **T. S. Rappaport and C. D. McGillem,**
" UHF Fading in Factories ",
IEEE Journal on Selected Areas in Communications, vol. 7, no. 1, pp. 40-48, January 1989.

[15] **J.G. Proakis,**
Digital Communications (2nd edition), McGraw-Hill Book Company, New York, 1989.

[16] **M. Abramowitz and I.A. Segun (Eds)**
Handbook of mathematical functions, Dover publications, Inc. New York, 1965.

[17] **H.L. van Trees,**
Detection, Estimation, and Modulation Theory PT. 1, Wiley, New York, 1968.

[18] **L. Kleinrock and F. A. Tobagi,**
" Packet switching in radio channels: Part 1- Carrier Sense Multiple Access nodes and their throughput-delay characteristics ",
IEEE Transactions on Communications, Vol. COM 23, pp.1400-1416, Dec. 1975.

[19] **R. Rom, M. Sidi,**
Multiple Access Protocols Performance and Analysis, Springer-Verlag New York Inc., 1990.

[20] **K.J. Zdunek, D.R. Ucci and J.L. Locicero,**
" Throughput of Nonpersistent Inhibit Sense Multiple Access with Capture ",
Electronics Letters, Vol.25, No.1, 5th January 1989.

[21] **R. Prasad,**
" Throughput Analysis of Nonpersistent Inhibit Sense Multiple Access in Multipath Fading and Shadowing Channels ", European Transactions on Telecommunication and Related Technologies, Vol. 2, pp.313-317, May-June 1991.

[22] **R. Prasad,**

" Performance Analysis of Mobile Packet Radio Networks in Real Channels with Inhibit-Sense Multiple Access ", Accepted for publication in IEE Proc. I, 1991.

[23] **J.P.M.G. Linnartz,**

" Near-Far Effects in some Random-Access Radio Systems with Fading ", Proc. of the 1990 Bilkent Int. Conf. on New Trends in Communication, Control and Signal Processing, Bilkent University, Ankara, pp. 504-510, 2-5 July 1990.

[24] **R. Prasad and J.C. Arnbak,**

" Capacity analysis of non-persistent Inhibit Sense Multiple Access in channels with multipath fading abd shadowing ", IEEE Proc. 1989 Workshop on Mobile and Cordless Telephone Communications, London, September 1989, pp.129-134.

[25] **M. Loog,**

Task report in preparation.

Appendix A: Program for slotted ALOHA for nRayleigh

```
Program Ri_Ra;            { werk met repeat }
{$N+}
uses crt;
var  R, zdb, zdec, finalsom, overlap, a, e, throughput,

     voorsom, firstsom, sigs,

     produkt : Real;

     geluid, inf, n_int, g, nh :Integer;

     thro : Text ;             test1,test2,hulp, tussen, Q_uit: double;


function fac(n:integer):double;
var  som : double;
begin
     som :=1
     ;
     while n>1 do
     begin
          som := som * n;
          n := n-1;
     end;
     fac := som;
end;



function power(a:extended;b:real):extended;

{-----------------------------------------------------}
{-              Calculates a^b                      -}
{-----------------------------------------------------}

var
  hlp : extended;

begin
  if a=0 then power:=0 else
  begin
    if a>0 then
    begin
      power:=exp(b*ln(a));
    end
    else
    begin
      hlp:=exp(b*ln(abs(a)));
        if b/2 = trunc(b/2) then power:=hlp else power:=-hlp;
    end;
  end;
end;


PROCEDURE FACULTEIT (GETAL:DOUBLE;VAR UITKOMSTF:extended);
BEGIN {FACULTEIT}
      IF GETAL>1 THEN
        BEGIN
            UITKOMSTF:=GETAL;
            WHILE GETAL <>1 DO
            BEGIN
                GETAL:=GETAL-1;
```

```
            UITKOMSTF:=UITKOMSTF*GETAL;
         END;
      END ELSE
      UITKOMSTF :=1;
END; {FACULTEIT}


PROCEDURE BESSELFUNCTIE (WAARDE :DOUBLE;ORDE:DOUBLE;VAR UITKOMSTB:DOUBLE);
CONST EPS =1E-5;
VAR Kb,Zb :DOUBLE;PO:INTEGER;
  XX,CC,BESSELOUD,BESSELNIEUW,ss :double;
  ond,boven, TERM1,TERM2,TERM3,TERM4:extended;

BEGIN {BESSELFUNCTIE}
    BESSELOUD :=10;
    BESSELNIEUW :=0;
    Kb:=0;po:=0;
    XX:= 0.25*sqr(waarde);
    WHILE ABS(BESSELNIEUW-BESSELOUD) >EPS DO
    BEGIN
        FACULTEIT(Kb,TERM3);
        FACULTEIT (orde+kb,TERM4);
        OND :=TERM3*TERM4;
        Zb:=kb;
        term2 := power(xx,zb);
        BOVEN:=TERM2;
        BESSELOUD :=BESSELNIEUW;
        BESSELNIEUW :=BESSELNIEUW+BOVEN/OND  ;
        Kb:= Kb+1; PO:=PO+1;
    END;
    ss := power(waarde/2,orde);
    UITKOMSTB:=BESSELNIEUW*ss;
END; {BESSELFUNCTIE}


pROCEDURE RIJ (AA,BB:DOUBLE;START :DOUBLE;VAR UITKOMSTR:DOUBLE);
CONST EPS= 1E-5;
VAR
  Nr,X,Y,RIJOUD,RIJNIEUW,UITKOMSTBESSEL,UITKOMSTMACHTA:DOUBLE;

BEGIN {RIJ}
    Y:= AA/BB;
    X:=BB*AA;
    RIJOUD:=10;
    RIJNIEUW :=0;
    Nr:=START;
    WHILE ABS(RIJNIEUW-RIJOUD) > EPS DO
    BEGIN
        uitkomstmachta := power(y,nr);
        BESSELFUNCTIE (x,nr,UITKOMSTBESSEL);
        RIJOUD :=RIJNIEUW;
        RIJNIEUW:= RIJNIEUW+ UITKOMSTMACHTA*UITKOMSTBESSEL;
        Nr:=Nr+1;
    END;
    UITKOMSTR:=RIJNIEUW;  {writeln(uitkomstbessel); }
END;{RIJ}

PROCEDURE MARCUMQ (WAARDEA,WAARDEB:DOUBLE;VAR UITKOMSTQ :DOUBLE);
VAR AANVANG,QFACTOR,UITKOMSTRIJ, mqbes, xm:DOUBLE;

BEGIN {MARCUMQ}
    QFACTOR:=EXP(-0.5*(SQR(WAARDEA)+SQR(WAARDEB)));

    if waardeA = 0 then uitkomstQ := exp(-0.5*sqr(waardeB))
    else
    IF WAARDEA<WAARDEB THEN
    BEGIN
```

```
        AANVANG:=0;
        RIJ(WAARDEA,WAARDEB,AANVANG,UITKOMSTRIJ);
        UITKOMSTQ := QFACTOR*UITKOMSTRIJ;
    END
    ELSE
    if waardeA = waardeB then
    begin
        xm:=sqr(waardeA);
        besselfunctie( xm, 0, mqbes);
        uitkomstQ:=0.5*(1+mqbes*exp(-sqr(waardeA)));

    end
    else

    BEGIN
        AANVANG :=1;
        RIJ (WAARDEB,WAARDEA,AANVANG,UITKOMSTRIJ);
        UITKOMSTQ:=1-QFACTOR*UITKOMSTRIJ;

    END; {write(uitkomstrij,'  '); }
END;{MARCUMQ}


Function Laguer ( tel : Integer) : double;
var   x, w : Array [1..15] of Real;
      i : Integer;
      lag_prod, bet, lag_som : double;

begin
  { x[1] := 0.137793470540;
   x[2] := 0.729454549503;
   x[3] := 1.808342901740;
   x[4] := 3.401433697855;
   x[5] := 5.552496140064;
   x[6] := 8.330152746764;
   x[7] := 11.843785837900;
   x[8] := 16.279257831378;
   x[9] := 21.996585811981;
   x[10]:= 29.920697012274;

   w[1] := 3.08441115765e-1;
   w[2] := 4.01119929155e-1;
   w[3] := 2.18068287612e-1;
   w[4] := 6.20874560987e-2;
   w[5] := 9.50151697518e-3;
   w[6] := 7.53008388588e-4;
   w[7] := 2.82592334960e-5;
   w[8] := 4.24931398496e-7;
   w[9] := 1.83956482398e-9;
   w[10]:= 9.91182721961e-13;
   }
   x[1] := 0.093307812017;
   x[2] := 0.492691740302;
   x[3] := 1.215595412071;
   x[4] := 2.269949526204;
   x[5] := 3.667622721751;
   x[6] := 5.425336627414;
   x[7] := 7.565916226613;
   x[8] := 10.120228568019;
   x[9] := 13.130282482176;
   x[10]:= 16.654407708330;
   x[11]:= 20.776478899449;
   x[12]:= 25.623894226729;
   x[13]:= 31.407519169754;
   x[14]:= 38.530683306486;
   x[15]:= 48.026085572686;
```

```
    w[1] := 2.18234885940e-1;
    w[2] := 3.42210177923e-1;
    w[3] := 2.63027577942e-1;
    w[4] := 1.26425818106e-1;
    w[5] := 4.02068649210e-2;
    w[6] := 8.56387780361e-3;
    w[7] := 1.21243614721e-3;
    w[8] := 1.11674392344e-4;
    w[9] := 6.45992676202e-6;
    w[10]:= 2.22631690710e-7;
    w[11]:= 4.22743038498e-9;
    w[12]:= 3.92189726704e-11;
    w[13]:= 1.45651526407e-13;
    w[14]:= 1.48302705111e-16;
    w[15]:= 1.60059490621e-20;


    lag_som := 0;

    For i := 1 to 15 do
    begin
        bet := sqrt ( 2 * zdec * sigs * x[i] );
        MARCUMQ ( e, bet, Q_uit );
        lag_prod := w[i] * power( x[i], tel - 1 )* Q_uit;
        lag_som := lag_som + lag_prod;
    end;
    Laguer := lag_som;
end;



{----- mean program --------}

BEGIN

    writeln;
    write ('Rice factor = ');
    read ( R );
    writeln ( R);

    {R := exp ( 0.1 * R * ln(10) ); }
    R := power( 10, 0.1*R);


    write ( 'capture ratio(z0) = ' );
    read ( zdb );
    writeln ( zdb );
    zdec := power( 10, 0.1*zdb );
    write ( 'approx. of infinity = ' );
    read ( inf );
    writeln ( inf );
   { write ( 'number of interferers = ' );
    read ( n_int);
    writeln ( n_int );
    }

    writeln ('ratio of average power inter/desired = ');
    read ( sigs );
    writeln ( sigs );

    assign ( thro, 'k9m10.dat' );
    rewrite (thro);
    e := sqrt ( 2*R );

    g := 0;
    writeln(g);
    writeln( thro, g );
```

```
For g := 1 to 6 do
begin

   finalsom := 0;

   nh := 1;
   Repeat

      overlap := power( g,nh) * exp( -g )/fac(nh);
      tussen := 1/fac(nh-1);
      hulp := Laguer( nh);
      produkt := overlap * ( 1 - tussen*hulp );
      finalsom := finalsom + produkt;
      nh := nh + 1;

   Until (produkt<1E-3);


   throughput := g * ( 1 - finalsom );
   writeln ( throughput );
   writeln ( thro, throughput );

end;
Close (thro);
writeln ( 'EINDE' );

For geluid := 1 to 30 do
begin
   sound(2000);
   delay (20);
   nosound;
end;

END.
```

Appendix B: Program for slotted ALOHA with n-RICE

```
{$N+}
{$M 16384,0,655360}

Program Ri_Ri;
uses crt;

var  Rd, Ru, zdb, zdec, a, e, sigm2 : Real;

     inf, n_int, g, nh , geluid :Integer;

     thro : Text ;

     epsi, hh, trap, dist_hulp, hulpsom, dist, overlap,

      finalsom, throughput, Q_alpha: double;


function fac(n:integer):double;
var  som : double;
begin
     som :=1
     ;
     while n>1 do
     begin
         som := som * n;
         n := n-1;
     end;
     fac := som;
end;


function power(a:extended;b:real):extended;

{-----------------------------------------------------}
{-                  Calculates a^b                  -}
{-----------------------------------------------------}

var
  hlp : extended;

begin
  if a=0 then power:=0 else
  begin
    if a>0 then
    begin
      power:=exp(b*ln(a));
    end
    else
    begin
      hlp:=exp(b*ln(abs(a)));
        if b/2 = trunc(b/2) then power:=hlp else power:=-hlp;
    end;
  end;
end;


PROCEDURE FACULTEIT (GETAL:DOUBLE;VAR UITKOMSTF:extended);
BEGIN {FACULTEIT}
      IF GETAL>1 THEN
        BEGIN
            UITKOMSTF:=GETAL;
            WHILE GETAL <>1 DO
            BEGIN
```

```
            GETAL:=GETAL-1;
            UITKOMSTF:=UITKOMSTF*GETAL;
         END;
      END ELSE
      UITKOMSTF :=1;
END; {FACULTEIT}


PROCEDURE BESSELFUNCTIE (WAARDE :DOUBLE;ORDE:DOUBLE;VAR UITKOMSTB:DOUBLE);
CONST EPS =1E-5;
VAR Kb,Zb :DOUBLE;PO:INTEGER;
  XX,CC,BESSELOUD,BESSELNIEUW,ss :double;
  ond,boven, TERM1,TERM2,TERM3,TERM4:extended;

BEGIN {BESSELFUNCTIE}
    BESSELOUD :=10;
    BESSELNIEUW :=0;
    Kb:=0;po:=0;
    XX:= 0.25*sqr(waarde);
    WHILE ABS(BESSELNIEUW-BESSELOUD) >EPS DO
    BEGIN
        FACULTEIT(Kb,TERM3);
        FACULTEIT (orde+kb,TERM4);
        OND :=TERM3*TERM4;
        Zb:=kb;
        term2 := power(xx,zb);
        BOVEN:=TERM2;
        BESSELOUD :=BESSELNIEUW;
        BESSELNIEUW :=BESSELNIEUW+BOVEN/OND  ;
        Kb:= Kb+1; PO:=PO+1;
    END;
    ss := power(waarde/2,orde);
    UITKOMSTB:=BESSELNIEUW*ss;
END; {BESSELFUNCTIE}


pROCEDURE RIJ (AA,BB:DOUBLE;START :DOUBLE;VAR UITKOMSTR:DOUBLE);
CONST EPS= 1E-5;
VAR
  Nr,X,Y,RIJOUD,RIJNIEUW,UITKOMSTBESSEL,UITKOMSTMACHTA:DOUBLE;

BEGIN {RIJ}
    Y:= AA/BB;
    X:=BB*AA;
    RIJOUD:=10;
    RIJNIEUW :=0;
    Nr:=START;
    WHILE ABS(RIJNIEUW-RIJOUD) > EPS DO
    BEGIN
        uitkomstmachta := power(y,nr);
        BESSELFUNCTIE (x,nr,UITKOMSTBESSEL);
        RIJOUD :=RIJNIEUW;
        RIJNIEUW:= RIJNIEUW+ UITKOMSTMACHTA*UITKOMSTBESSEL;
        Nr:=Nr+1;
    END;
    UITKOMSTR:=RIJNIEUW;  {writeln(uitkomstbessel); }
END;{RIJ}

PROCEDURE MARCUMQ (WAARDEA,WAARDEB:DOUBLE;VAR UITKOMSTQ :DOUBLE);
VAR AANVANG,QFACTOR,UITKOMSTRIJ, mqbes, xm:DOUBLE;

BEGIN {MARCUMQ}
    QFACTOR:=EXP(-0.5*(SQR(WAARDEA)+SQR(WAARDEB)));

    if waardeA = 0 then uitkomstQ := exp(-0.5*sqr(waardeB))
    else
    IF WAARDEA<WAARDEB THEN
```

```
    BEGIN
        AANVANG:=0;
        RIJ(WAARDEA,WAARDEB,AANVANG,UITKOMSTRIJ);
        UITKOMSTQ := QFACTOR*UITKOMSTRIJ;
    END
    ELSE
    if waardeA = waardeB then
    begin
        xm:=sqr(waardeA);
        besselfunctie( xm, 0, mqbes);
        uitkomstQ:=0.5*(1+mqbes*exp(-sqr(waardeA)));

    end
    else

    BEGIN
        AANVANG :=1;
        RIJ (WAARDEB,WAARDEA,AANVANG,UITKOMSTRIJ);
        UITKOMSTQ:=1-QFACTOR*UITKOMSTRIJ;

    END; {write(uitkomstrij,'  '); }
END;{MARCUMQ}



Function hulp_Trap ( m:Integer ; val: Real ): double;

var  one, two, three, four, Bes1, Bes2, Bes_hulp, Q_B : double;

Begin

     one := power( val, m );
     two := exp( - sqr(val) /2 );
    { Bes1 := sqrt( 2*m*R );  }
    { Bes2 := sqrt( sigm2 ); }
    { Bes_hulp := Bes1 * Bes2 * val; }
    Bes_hulp := epsi * val;

     Besselfunctie ( Bes_hulp, m-1, three );
     Q_B := sqrt( 2*zdec )*val;
     MARCUMQ ( Q_alpha, Q_B, four );
     hulp_Trap := one * two * three * four ;

end;

Function Rep_Trap ( m, lowlim, uplim, step : Integer ): double;

var Go : Boolean;
    B_int, sum_Trap, val, Integ : double;
    cor, t, test : Integer;    h : Real;


Begin
    Go := TRUE;
    B_int := 0;
    REPEAT

      FOR test := 1 to 2 do
      begin

        cor := 1;
        sum_trap := 0;
        t := 2 ;
        REPEAT

            h := ( uplim - lowlim )/step;
            val := lowlim + ( 2*t - 2 )* h - cor* h;
```

```
          Integ := hulp_Trap ( m, val );

      {    writeln('t=',t:2,' int=',Integ:4,' step=',step:2);  }

          sum_Trap := sum_Trap + Integ*h;
          cor := cor + 1;
          t := t + 1;
      UNTIL ( t=step+1 ) OR ( Integ < 1E-4 );
      val := uplim;
      Integ := hulp_Trap ( m, val );
      sum_Trap := sum_Trap + h*0.5*Integ;
      val := lowlim;
      Integ := hulp_Trap( m, val );
      sum_Trap := sum_Trap + h*0.5*Integ;

      If test = 1 then  B_int := sum_Trap;
      step := step*2 ;

    END;
  UNTIL ( abs(sum_trap - B_int) < 1E-4 );
  Rep_Trap := sum_Trap -(1/3)*( sum_Trap - B_int );

end;



{----- mean program --------}

BEGIN

  writeln;
  write ('desired Rice factor = ');
  read ( Rd );
  writeln ( Rd);

  {R := exp ( 0.1 * R * ln(10) ); }
  Rd := power( 10, 0.1*Rd);

  write ('undesired Rice-factor = ');
  read (Ru);
  writeln(Ru);
  Ru := power( 10, 0.1*Ru );


  write ( 'capture ratio(z0) = ' );
  read ( zdb );
  writeln ( zdb );
  zdec := power( 10, 0.1*zdb );


  assign ( thro, 'RiRi351.dat' );
  rewrite (thro);
 {  e := sqrt ( 2*R );   }

  g := 0;
  writeln(g);
  writeln( thro, g );

  For g := 1 to 6 do
  begin

     finalsom := 0;
     nh := 1;
     Repeat
        WRITE('n=',nh:2);
```

```
            overlap := power( g, nh )* exp (-g) / fac(nh);
            epsi := sqrt( 2*nh*Ru ) ; {*sqrt( sigm2 ); }
            Q_alpha := sqrt( 2*Rd );
            hh := power( epsi,2 );
            dist_hulp := (1/power( epsi, nh-1 )) * exp( -hh/2 );

            WRITE (' dist_hulp=',dist_hulp:4);
            Trap := Rep_Trap ( nh, 0, 12, 4 );
            WRITELN(' trap=',trap:4);

            dist := 1 - dist_hulp * Trap;
            hulpsom :=  overlap * dist;
            finalsom := finalsom + hulpsom;
            nh := nh + 1;

        Until ( hulpsom < 1E-4 );

        throughput := g*( 1 - finalsom );
        writeln ( 'throughput=', throughput:14 );
        writeln ( thro, throughput );

    end;

    Close (thro);
    writeln ( 'EINDE' );

    For geluid :=1 to 150 do
    begin
        sound(4000);
        delay(30);
        nosound;
    end;

END.
```

Appendix C: Unslotted ISMA for n-Rayleigh

```
Program Ri_Ra;              {werk}
{$N+}
uses crt;
var  R, finalsom, overlap, a, e, throughput, mult, term0,

     cycle, dg, d, produkt : Real;

     geluid, g, nh :Integer;

     thro : Text ;              hulp, tussen, Q_uit: double;


function fac(n:integer):double;
var  som : double;
begin
     som :=1
     ;
     while n>1 do
     begin
          som := som * n;
          n := n-1;
     end;
     fac := som;
end;


function power(a:extended;b:real):extended;

{----------------------------------------------------}
{-                  Calculates a^b                  -}
{----------------------------------------------------}

var
  hlp : extended;

begin
  if a=0 then power:=0 else
  begin
    if a>0 then
    begin
      power:=exp(b*ln(a));
    end
    else
    begin
      hlp:=exp(b*ln(abs(a)));
        if b/2 = trunc(b/2) then power:=hlp else power:=-hlp;
    end;
  end;
end;


PROCEDURE FACULTEIT (GETAL:DOUBLE;VAR UITKOMSTF:extended);
BEGIN {FACULTEIT}
      IF GETAL>1 THEN
         BEGIN
              UITKOMSTF:=GETAL;
              WHILE GETAL <>1 DO
              BEGIN
                   GETAL:=GETAL-1;
                   UITKOMSTF:=UITKOMSTF*GETAL;
              END;
```

```
        END ELSE
        UITKOMSTF :=1;
END; {FACULTEIT}


PROCEDURE BESSELFUNCTIE (WAARDE :DOUBLE;ORDE:DOUBLE;VAR UITKOMSTB:DOUBLE);
CONST EPS =1E-5;
VAR Kb,Zb :DOUBLE;PO:INTEGER;
  XX,CC,BESSELOUD,BESSELNIEUW,ss :double;
  ond,boven, TERM1,TERM2,TERM3,TERM4:extended;

BEGIN {BESSELFUNCTIE}
    BESSELOUD :=10;
    BESSELNIEUW :=0;
    Kb:=0;po:=0;
    XX:= 0.25*sqr(waarde);
    WHILE ABS(BESSELNIEUW-BESSELOUD) >EPS DO
    BEGIN
        FACULTEIT(Kb,TERM3);
        FACULTEIT (orde+kb,TERM4);
        OND :=TERM3*TERM4;
        Zb:=kb;
        term2 := power(xx,zb);
        BOVEN:=TERM2;
        BESSELOUD :=BESSELNIEUW;
        BESSELNIEUW :=BESSELNIEUW+BOVEN/OND  ;
        Kb:= Kb+1; PO:=PO+1;
    END;
    ss := power(waarde/2,orde);
    UITKOMSTB:=BESSELNIEUW*ss;
END; {BESSELFUNCTIE}


pROCEDURE RIJ (AA,BB:DOUBLE;START :DOUBLE;VAR UITKOMSTR:DOUBLE);
CONST EPS= 1E-5;
VAR
  Nr,X,Y,RIJOUD,RIJNIEUW,UITKOMSTBESSEL,UITKOMSTMACHTA:DOUBLE;

BEGIN {RIJ}
    Y:= AA/BB;
    X:=BB*AA;
    RIJOUD:=10;
    RIJNIEUW :=0;
    Nr:=START;
    WHILE ABS(RIJNIEUW-RIJOUD) > EPS DO
    BEGIN
        uitkomstmachta := power(y,nr);
        BESSELFUNCTIE (x,nr,UITKOMSTBESSEL);
        RIJOUD :=RIJNIEUW;
        RIJNIEUW:= RIJNIEUW+ UITKOMSTMACHTA*UITKOMSTBESSEL;
        Nr:=Nr+1;
    END;
    UITKOMSTR:=RIJNIEUW;  {writeln(uitkomstbessel); }
END;{RIJ}

PROCEDURE MARCUMQ (WAARDEA,WAARDEB:DOUBLE;VAR UITKOMSTQ :DOUBLE);
VAR AANVANG,QFACTOR,UITKOMSTRIJ, mqbes, xm:DOUBLE;

BEGIN {MARCUMQ}
    QFACTOR:=EXP(-0.5*(SQR(WAARDEA)+SQR(WAARDEB)));

    if waardeA = 0 then uitkomstQ := exp(-0.5*sqr(waardeB))
    else
    IF WAARDEA<WAARDEB THEN
    BEGIN
        AANVANG:=0;
        RIJ(WAARDEA,WAARDEB,AANVANG,UITKOMSTRIJ);
```

```
        UITKOMSTQ := QFACTOR*UITKOMSTRIJ;
    END
    ELSE
    if waardeA = waardeB then
    begin
        xm:=sqr(waardeA);
        besselfunctie( xm, 0, mqbes);
        uitkomstQ:=0.5*(1+mqbes*exp(-sqr(waardeA)));

    end
    else

    BEGIN
        AANVANG :=1;
        RIJ (WAARDEB,WAARDEA,AANVANG,UITKOMSTRIJ);
        UITKOMSTQ:=1-QFACTOR*UITKOMSTRIJ;

    END; {write(uitkomstrij,'    '); }
END;{MARCUMQ}


Function Laguer ( tel : Integer) : double;
var   x, w : Array [1..15] of Real;
      i : Integer;
      lag_prod, bet, lag_som : double;

begin
   { x[1] := 0.137793470540;
    x[2] := 0.729454549503;
    x[3] := 1.808342901740;
    x[4] := 3.401433697855;
    x[5] := 5.552496140064;
    x[6] := 8.330152746764;
    x[7] := 11.843785837900;
    x[8] := 16.279257831378;
    x[9] := 21.996585811981;
    x[10]:= 29.920697012274;

    w[1] := 3.08441115765e-1;
    w[2] := 4.01119929155e-1;
    w[3] := 2.18068287612e-1;
    w[4] := 6.20874560987e-2;
    w[5] := 9.50151697518e-3;
    w[6] := 7.53008388588e-4;
    w[7] := 2.82592334960e-5;
    w[8] := 4.24931398496e-7;
    w[9] := 1.83956482398e-9;
    w[10]:= 9.91182721961e-13;
    }
    x[1] := 0.093307812017;
    x[2] := 0.492691740302;
    x[3] := 1.215595412071;
    x[4] := 2.269949526204;
    x[5] := 3.667622721751;
    x[6] := 5.425336627414;
    x[7] := 7.565916226613;
    x[8] := 10.120228568019;
    x[9] := 13.130282482176;
    x[10]:= 16.654407708330;
    x[11]:= 20.776478899449;
    x[12]:= 25.623894226729;
    x[13]:= 31.407519169754;
    x[14]:= 38.530683306486;
    x[15]:= 48.026085572686;

    w[1] := 2.18234885940e-1;
    w[2] := 3.42210177923e-1;
```

```
    w[3] := 2.63027577942e-1;
    w[4] := 1.26425818106e-1;
    w[5] := 4.02068649210e-2;
    w[6] := 8.56387780361e-3;
    w[7] := 1.21243614721e-3;
    w[8] := 1.11674392344e-4;
    w[9] := 6.45992676202e-6;
    w[10]:= 2.22631690710e-7;
    w[11]:= 4.22743038498e-9;
    w[12]:= 3.92189726704e-11;
    w[13]:= 1.45651526407e-13;
    w[14]:= 1.48302705111e-16;
    w[15]:= 1.60059490621e-20;


    lag_som := 0;

    For i := 1 to 15 do
    begin
       bet := sqrt ( 2 * mult * x[i] );
       MARCUMQ ( e, bet, Q_uit );
       lag_prod := w[i] * power( x[i], tel - 1 )* Q_uit;
       lag_som := lag_som + lag_prod;
    end;
    Laguer := lag_som;
end;



{----- mean program --------}

BEGIN

    writeln;
    write ('Rice factor = ');
    read ( R );
    writeln ( R);

    {R := exp ( 0.1 * R * ln(10) ); }
    R := power( 10, 0.1*R);      R:=0;

    write('ratio z0*sigl/sigd=');
    read(mult);
    writeln(mult);
    mult:=power(10, 0.1*mult);

    write('inhibit delay fraction  d=');
    read(d);
    writeln(d:3);

    assign ( thro, 'is03_05.dat' );
    rewrite (thro);
    e := sqrt ( 2*R );

    g := 0;
    writeln(g);
    writeln( thro, g );

    For g := 1 to 7 do
    begin

       finalsom := 0;
       dg := d*g;
       term0 := exp(-dg);
       nh := 1;
       Repeat
          write(' nh=',nh:2);
```

```
        overlap := power( dg,nh) * exp( -dg )/fac(nh);
        tussen := (nh + 1)/fac(nh-1);
        hulp := Laguer( nh);
        produkt := tussen * overlap * hulp ;
        finalsom := finalsom + produkt;
        nh := nh + 1;
    Until produkt < 1E-4;

    cycle := 1 + 2*d + exp(-dg)/g;

    throughput := ( term0 + finalsom )/cycle ;
    writeln ( ' throughput=',throughput:14 );
    writeln ( thro, throughput );

    For geluid:= 1 to 15 do
    begin
        sound(2000);
        delay(20);
        nosound;
    end;

  end;
  Close (thro);
  writeln ( 'EINDE' );

  For geluid := 1 to 30 do
  begin
      sound(4000);
      delay(30);
      nosound;
  end;

END.
```

Appendix D: Unslotted ISMA for n-Rice

```
Program Ri_Ra;                {}
{$N+}
uses crt;
var  Rd, Ru, zdb, zdec, finalsom, overlap, a, e, throughput,

     epsi, Q_alpha, hh, hulpsom, d, dg, term0, cycle,

     produkt : Real;

     geluid, inf, n_int, g, nh :Integer;

     thro : Text ;

     haak1, haak2, haak3, dist, hulp, tussen, Q_uit: double;


function fac(n:integer):double;
var  som : double;
begin
     som :=1
     ;
     while n>1 do
     begin
          som := som * n;
          n := n-1;
     end;
     fac := som;
end;


function power(a:extended;b:real):extended;

{-----------------------------------------------------}
{-                  Calculates a^b                  -}
{-----------------------------------------------------}

var
  hlp : extended;

begin
  if a=0 then power:=0 else
  begin
    if a>0 then
    begin
      power:=exp(b*ln(a));
    end
    else
    begin
      hlp:=exp(b*ln(abs(a)));
        if b/2 = trunc(b/2) then power:=hlp else power:=-hlp;
    end;
  end;
end;


PROCEDURE FACULTEIT (GETAL:DOUBLE;VAR UITKOMSTF:extended);
BEGIN {FACULTEIT}
      IF GETAL>1 THEN
        BEGIN
            UITKOMSTF:=GETAL;
            WHILE GETAL <>1 DO
            BEGIN
```

```
            GETAL:=GETAL-1;
            UITKOMSTF:=UITKOMSTF*GETAL;
         END;
      END ELSE
      UITKOMSTF :=1;
END; {FACULTEIT}


PROCEDURE BESSELFUNCTIE (WAARDE :DOUBLE;ORDE:DOUBLE;VAR UITKOMSTB:DOUBLE);
CONST EPS =1E-5;
VAR Kb,Zb :DOUBLE;PO:INTEGER;
  XX,CC,BESSELOUD,BESSELNIEUW,ss :double;
  ond,boven, TERM1,TERM2,TERM3,TERM4:extended;

BEGIN {BESSELFUNCTIE}
    BESSELOUD :=10;
    BESSELNIEUW :=0;
    Kb:=0;po:=0;
    XX:= 0.25*sqr(waarde);
    WHILE ABS(BESSELNIEUW-BESSELOUD) >EPS DO
    BEGIN
        FACULTEIT(Kb,TERM3);
        FACULTEIT (orde+kb,TERM4);
        OND :=TERM3*TERM4;
        Zb:=kb;
        term2 := power(xx,zb);
        BOVEN:=TERM2;
        BESSELOUD :=BESSELNIEUW;
        BESSELNIEUW :=BESSELNIEUW+BOVEN/OND  ;
        Kb:= Kb+1; PO:=PO+1;
    END;
    ss := power(waarde/2,orde);
    UITKOMSTB:=BESSELNIEUW*ss;
END; {BESSELFUNCTIE}


pROCEDURE RIJ (AA,BB:DOUBLE;START :DOUBLE;VAR UITKOMSTR:DOUBLE);
CONST EPS= 1E-5;
VAR
  Nr,X,Y,RIJOUD,RIJNIEUW,UITKOMSTBESSEL,UITKOMSTMACHTA:DOUBLE;

BEGIN {RIJ}
    Y:= AA/BB;
    X:=BB*AA;
    RIJOUD:=10;
    RIJNIEUW :=0;
    Nr:=START;
    WHILE ABS(RIJNIEUW-RIJOUD) > EPS DO
    BEGIN
        uitkomstmachta := power(y,nr);
        BESSELFUNCTIE (x,nr,UITKOMSTBESSEL);
        RIJOUD :=RIJNIEUW;
        RIJNIEUW:= RIJNIEUW+ UITKOMSTMACHTA*UITKOMSTBESSEL;
        Nr:=Nr+1;
    END;
    UITKOMSTR:=RIJNIEUW;  {writeln(uitkomstbessel); }
END;{RIJ}

PROCEDURE MARCUMQ (WAARDEA,WAARDEB:DOUBLE;VAR UITKOMSTQ :DOUBLE);
VAR AANVANG,QFACTOR,UITKOMSTRIJ, mqbes, xm:DOUBLE;

BEGIN {MARCUMQ}
    QFACTOR:=EXP(-0.5*(SQR(WAARDEA)+SQR(WAARDEB)));

    if waardeA = 0 then uitkomstQ := exp(-0.5*sqr(waardeB))
    else
    IF WAARDEA<WAARDEB THEN
```

```
    BEGIN
        AANVANG:=0;
        RIJ(WAARDEA,WAARDEB,AANVANG,UITKOMSTRIJ);
        UITKOMSTQ := QFACTOR*UITKOMSTRIJ;
    END
    ELSE
    if waardeA = waardeB then
    begin
        xm:=sqr(waardeA);
        besselfunctie( xm, 0, mqbes);
        uitkomstQ:=0.5*(1+mqbes*exp(-sqr(waardeA)));

    end
    else

    BEGIN
        AANVANG :=1;
        RIJ (WAARDEB,WAARDEA,AANVANG,UITKOMSTRIJ);
        UITKOMSTQ:=1-QFACTOR*UITKOMSTRIJ;

    END; {write(uitkomstrij,'  '); }
END;{MARCUMQ}


Function Laguer ( m : Integer) : double;
var   x, w : Array [1..15] of Real;
      i : Integer;
      een, one, two, three, four, Q_b, lag_prod, bes_hulp, lag_som : double;

begin

   x[1] := 0.093307812017;
   x[2] := 0.492691740302;
   x[3] := 1.215595412071;
   x[4] := 2.269949526204;
   x[5] := 3.667622721751;
   x[6] := 5.425336627414;
   x[7] := 7.565916226613;
   x[8] := 10.120228568019;
   x[9] := 13.130282482176;
   x[10]:= 16.654407708330;
   x[11]:= 20.776478899449;
   x[12]:= 25.623894226729;
   x[13]:= 31.407519169754;
   x[14]:= 38.530683306486;
   x[15]:= 48.026085572686;

   w[1] := 2.18234885940e-1;
   w[2] := 3.42210177923e-1;
   w[3] := 2.63027577942e-1;
   w[4] := 1.26425818106e-1;
   w[5] := 4.02068649210e-2;
   w[6] := 8.56387780361e-3;
   w[7] := 1.21243614721e-3;
   w[8] := 1.11674392344e-4;
   w[9] := 6.45992676202e-6;
   w[10]:= 2.22631690710e-7;
   w[11]:= 4.22743038498e-9;
   w[12]:= 3.92189726704e-11;
   w[13]:= 1.45651526407e-13;
   w[14]:= 1.48302705111e-16;
   w[15]:= 1.60059490621e-20;


   lag_som := 0;

   For i := 1 to 15 do
```

```
    begin
        een := w[i]*exp(x[i]);
        one := power( x[i], m);
        two := exp ( -sqr(x[i]/2 ));
        bes_hulp := epsi * x[i];
        Besselfunctie ( bes_hulp, m-1, three);
        Q_B := sqrt (2*zdec)* x[i];
        MarcumQ ( Q_alpha, Q_b, four );
        lag_prod := een*one*two*three*four;
        lag_som := lag_som + lag_prod;
    end;
    Laguer := lag_som;
end;


{----- mean program --------}

BEGIN

    writeln;
    write ('desired Rice factor = ');
    read ( Rd );
    writeln ( Rd);

    {R := exp ( 0.1 * R * ln(10) ); }
    Rd := power( 10, 0.1*Rd);

    write ('undesired Rice-factor = ');
    read (Ru);
    writeln(Ru);
    Ru := power( 10, 0.1*Ru );

    write ( 'capture ratio(z0) = ' );
    read ( zdb );
    writeln ( zdb );
    zdec := power( 10, 0.1*zdb );
    write('inhibit delay fraction  d=');
    read(d);
    writeln(d:3);

    assign ( thro, 'si773_5.dat' );
    rewrite (thro);
  {  e := sqrt ( 2*R );    }

    g := 0;
    writeln(g);
    writeln( thro, g );

    For g := 1 to 10 do
    begin

        finalsom := 0;
        dg := d*g;
        term0 := exp(-dg);
        nh := 1;
        Repeat
            WRITE('n=',nh:2);

            haak1 := nh + 1;
            overlap := power( dg, nh )* exp (-dg) / fac(nh);
            epsi := sqrt( 2*nh*Ru ) ; {*sqrt( sigm2 ); }
            Q_alpha := sqrt( 2*Rd );
            haak2 := 1/ power( epsi, nh - 1 );
            haak3 := exp ( -sqr(epsi)/2 );
            dist := Laguer (nh);
            hulpsom := haak1*overlap*haak2*haak3*dist;
```

```
            finalsom := finalsom + hulpsom;
            nh := nh + 1;

        Until ( hulpsom < 1E-3 );

        cycle := 1 + 2*d + exp( -dg )/g;
        throughput := ( term0 + finalsom )/cycle;
        writeln ( 'throughput=', throughput:14 );
        writeln ( thro, throughput );

    end;

    Close (thro);
    writeln ( 'EINDE' );

    For geluid :=1 to 30 do
    begin
        sound(4000);
        delay(30);
        nosound;
    end;

END.
```