A NEW METHODOLOGY FOR THE DESIGN OF HIGH SPEED WIRELESS COMMUNICATION SYSTEMS BASED ON EXPERIMENTAL RESULTS

by

Timothy Gallagher

Submitted to the Department Of Electrical Engineering and Computer Science and the Faculty of the Graduate School of the University of Kansas in partial fulfillment of the requirements for the degree of Master of Science

Approved by ______________________________

Chair of Supervisory Committee

____

____

Date thesis accepted ______________________________

The University of Kansas

Abstract

A NEW METHODOLOGY FOR THE
DESIGN OF HIGH SPEED WIRELESS
COMMUNICATION SYSTEMS BASED ON
EXPERIMENTAL RESULTS

by Timothy Gallagher

Chair of the Supervisory Committee: Professor James Roberts
Department of Electrical Engineering and Computer Science

Radio channel characterization is an important step in designing a wireless communication system. By knowing the nature of the channel, it is possible to intelligently design a system that meets a set of required specifications. A measurement system was designed and utilized to obtain experimental data concerning the characteristics of the wireless channel around the University of Kansas, Lawrence campus at 5.8 GHz. In this thesis, a process is developed to characterize any wireless channel at a specified frequency. The results of this investigation were obtained using the 5.8 GHz Unlicensed National Information Infrastructure (U-NII) band. Not only was the receive power found to be log-normal about a distance dependent mean, but the distribution of the Rician specular-to-random ratio, K, was also found to be log-normal about a distance dependent mean. The average value of K was found to decrease as the transmitter-receiver separation increased. A positive correlation was found between the amount of shadowing and K. A new approach to determining percent coverage area was developed that uses this information to obtain a

more accurate result. Normally, Rayleigh fading is assumed over the entire area in a cell. By using the experimental results and the process developed here, it is possible to save as much as 5 dB in ERP when a 90 percent coverage area is specified.

TABLE OF CONTENTS

LIST OF FIGURES AND TABLES

ACKNOWLEDGMENTS

I would like to thank the many people whose advice, hard work and patience have helped me to complete this investigation. My advisor, Dr. James Roberts, spent countless hours directing and encouraging me and for that I am deeply indebted. I would like to thank the other members of my committee, Dr. Joseph Evans and Dr. Glenn Prescott, for their help and advice in all things academic and otherwise. Dan Depardo built the transmitter and offered invaluable practical advice. Without Artur Leung, who spent too many hours on the roof of Nichols Hall, I would have had to be in two places at the same time. The many students involved in the University of Kansas Rapidly Deployable Network (RDRN) have offered their support and asked the right questions when I needed them. The encouragement and support I received from both my family and Donna's family will never be forgotten. Finally, I would like to thank my wife Donna and beautiful new baby girl Morganne. With them, all things are possible.

1 Introduction

1.1 Motivation

The investigation conducted in this thesis is part of a Defense Advanced Research Projects Agency (DARPA) sponsored project at the University of Kansas called the Rapidly Deployable Radio Network (RDRN). RDRN is expected to be the backbone of a broadband wireless communication system initially intended for military use, but with applications in the private sector such as use in disaster situations when the traditional tethered communication system has become inoperable.

In order to determine the expected performance of the system, it is prudent to characterize the radio channel at the frequency to be used for the carrier. One of the carrier frequencies for this project is 5.8 GHz. This frequency is in a newly created band and little experimental work has been done to determine the characteristics of the radio channel at this frequency. Once the characteristics of the radio channel have been determined, it is possible to compare the actual system performance with the predicted and make intelligent choices about the whole system design.

1.2 Broadband Wireless

Wireless communication is hot. The best indicator of this is the millions of dollars companies are spending on FCC licenses and research and development to provide customers with better, more sophisticated products and services. One of the biggest technological challenges in getting consumers to give up their wire-based systems and go wireless is a question of quality. In order for wireless services to compete with their traditional counterparts, it is essential for these services to be at the very least of comparable quality.

The trend in recent years is the consumer's demand for more and faster access to information. This means the service provides need to be able to provide more and more bandwidth. If wireless systems can offer comparable amount of bandwidth, they will be able to compete successfully with wire-based systems. A number of wireless technologies are emerging to satisfy the consumer's quest for bandwidth, such as wireless local loops (WLL) and wireless local area networks (WLAN).

WLLs are being developed as an alternative to traditional wired access to homes. Since the Federal Communications Commission (FCC) has begun allowing long distance carriers the right to offer access directly to the home, the WLL has gained in popularity as a viable alternative to the expense of running wire to a person's home. As anyone who has ever waited patiently for an Internet web page to load can attest, however, faster is better. The bandwidth over a wired line can be small anyway, but if the WLL cannot provide at least that much bandwidth, it will not be able to compete successfully with traditional wire-based systems.

WLANs are also being developed. Ethernet speeds of up to 10 Mbps need to be matched by WLANs if they are going to be successful. The applications for these WLANs are for areas that do not have an existing reasonable infrastructure. Schools and hospitals in rural areas that lack the network infrastructure or consist of buildings that do not allow for easy installation of the wires necessary to connect a local area network (LAN) can save by installing a WLAN.

1.3 Choosing the Carrier Frequency

One of the first questions to be asked during any wireless communication system design process is what will the carrier frequency be. The Federal Communications Commission (FCC) is the government agency in the United States that sets the rules for all uses of the radio spectrum. The FCC has broken the spectrum up into bands that are either licensed or unlicensed. The licensed bands require an explicit license from the FCC before the band can be used. These licenses can take years to acquire and often are very expensive. This may be fine for commercial applications where the company buying the rights to that band will be using the band for many years in the hope that at some point a profit can be realized. This, however, is not suitable for research applications where the focus may be on rapidly developing a prototype. By using the unlicensed bands, development can start more quickly and be less expensive since the FCC does not need to issue a license. As long as the rules are followed for the particular band of interest, the FCC does not need to intervene.

The amateur bands are ideal for research, since there are no output power restrictions. There are several problems, however, with using these bands. When using these bands, the transmitter must send an unencoded identification along with the data. This allows others using the band to know what information is for them and what is not for them. Although the user of one of these bands does not need to explicitly ask the FCC for permission to use the band as with the licensed bands, the user must pass a test to show familiarity with the rules governing that band. In addition, products are not as readily available at these frequencies as they are at some of the other unlicensed bands. This can make it difficult to deliver prototypes.

Another set of unlicensed bands are the Industrial, Scientific, and Medical (ISM) bands. These three bands are at 915 MHz, 2.4 GHz, and 5.8 GHz. These are popular frequencies and therefore, there are cheaper, more easily obtainable commercial products and circuits available for research and development. These bands would be fine except for the FCC rule that states that users of these bands must employ spread spectrum. For high data rate applications, such as video, the chip rate necessary to provide a reasonable processing gain can be very large. This high chip rate can also be a problem for applications that require a great deal of digital signal processing (DSP), such as beam forming. DSP processors, at this writing, are not fast enough to accomplish their tasks at the high chips rates necessary for wideband transmissions. This makes the use of these bands unsuitable for the development of wideband applications.

The FCC on January 9, 1997 created the Unlicensed National Information Infrastructure (U-NII) band by expanding and dividing the 5.8 GHz ISM band into three 100 MHz bands [32]. The frequencies ranges are 5.15-5.25 GHz, 5.25-5.35 GHz, and 5.725-5.825 GHz. These bands were created to foster the development of a variety of short range, high speed devices and digital products for use in the U.S. and in overseas markets.

In order to facilitate the growth of new industries, the FCC decided to keep the restrictions on this band to a minimum. The only major technical restriction is the effective radiated power (ERP) limitation for each of the blocks of frequencies. At 5.15 GHz the maximum ERP is 200 mW; at 5.25 GHz the maximum ERP is 1 W; at 5.725 GHz the maximum ERP is 4 W. The FCC no longer requires the use of spread spectrum at these frequencies, so it is easier to develop wideband applications that use DSP. The high frequency also implies that smaller antennas are required for operation. This can be useful for systems that require antenna arrays.

The major industry the FCC hopes to foster with the creation of this new band is the wireless local area network (LAN) industry. The desire is to develop low cost wireless LANs for education and health care in rural areas where access to the national information infrastructure is limited. These frequencies are also comparable to those used in Europe for their High Performance Local Area Networks (HIPERLAN), so the incentive is there to develop products for possible export as well.

1.4 Characterizing the Channel

In any communication system, the goal is to provide the user with a certain level of quality of service (QoS). One measure of QoS in digital communication systems is the bit error rate (BER). To this end, it is important to know what the transmission medium "looks" like in order to satisfy the requirement.

In wired systems, the wires that are used are characterized using different methods. One method is to specify the amount of signal power loss over a certain length of wire.

This allows the system designer to predict the received power given the transmit power and length of the wire. Then, by taking into account other parameters of the system, such as the modulation technique employed, it is possible to predict what the BER will be. This BER can then be used as the QoS statistic. What is nice about wired systems is that the characteristics of the wire change little over time. This implies that the QoS will remain relatively constant both over the short-term and over the long-term. This, however, is not the case in wireless systems.

Wireless systems suffer from a number of propagation anomalies, such as multipath dispersion, shadowing, diffraction and absorption. The receiver in a wireless system tends to see multiple versions of the transmitted signal due to multipath of the transmitted signal. These multipath waves reach the receiver with different time delays and amplitudes. The received signal is then the vector sum of the multipath waves. If the transmitter, receiver or anything in between moves, the phase relationship between the multipath waves change forcing the resultant signal amplitude to change. The rapid fluctuation in received signal power over short periods of time is called fading [1], [9]. Because the signal varies over time, the QoS also varies over time. One design approach in a wireless channel is then to provide the user with an average QoS. It is therefore important to understand the nature of the fading so that a system that satisfies the requirements can be designed.

Another problem that wireless systems suffer from is shadowing [1], [9], [11]. In wired systems, the signal loss over a specific distance is easily determined simply by sending a signal with a known amplitude in one end and measuring the signal strength at the other end. The path loss is the difference between the input power and the output power. As long as the wire is well insulated, the placement of the wire should have no bearing on the performance. The path loss in a wireless channel, however, is greatly affected by the environment. The long-term average power at a specific distance can vary greatly because the wireless channel is not necessarily uniform in all directions. There may be buildings off in one direction, trees in another and an open field in a third. This variation in average received signal strength at different locations equidistant from the transmitter is called shadowing. It is because the average signal levels can be very different that it is also important to understand the nature of these long-term variations as well as the short-term variations caused by fading.

1.5 BER vs PER

Wireless asynchronous transfer mode (WATM) is an emerging wireless technology that is expected to provide end-to-end ATM connectivity and quality of service (QoS) over a wireless link [34]. In ATM, the basic unit of data transmission is a packet of 424 bits. The QoS statistics are then based on these packets rather than on the individual bits. For these systems and others that are based on packets of data, a more useful statistic is the packet error rate (PER).

1.6 Line-of-Sight, Obstructed and All Locations Cases

The analysis done for this investigation was duplicated for three sets of locations, line-of-sight (LOS), obstructed (OBS) and all locations taken together (ALL). The LOS locations were those places where there was a visual direct path between the transmitter and the receiver. The rest were considered OBS. The reason the locations were divided into these two groups was that it was observed that both the average path loss and fading were vastly different for the two cases. The ALL case was then analyzed for completeness.

1.7 Organization of the Thesis

Chapter 2 presents the theoretical basis for the models used to describe the various parameters associated with a wireless channel. Chapter 3 is a description of the measurement system and experimental procedure used in the investigation. The results are then presented in Chapter 5 and the conclusions are given in Chapter 6.

2 Background

As mentioned previously, BER and PER are popular QoS statistics. These quantities can be readily equated to voice and picture quality in digital audio or video transmissions. If the wireless communication link were for a system where the transmitter and receiver locations are fixed, the system designer could build the system to meet the specified average BER or PER. If, however, the transmission is broadcast or the location of the receiver is unknown, another QoS statistic can be used, the percent coverage area. This is a measure of how much of a given area at any one time meets the BER or PER requirement.

Normally the percent coverage area is calculated by assuming the entire cell suffers from the worst type of fading, Rayleigh fading [19], [35], [36]. The goal of this thesis is to use information obtained about the channel to devise a more accurate process for determining the percent coverage area. In order to do this, certain parameters of the wireless channel must be modeled. The Rician probability density function (pdf) will be shown an excellent model for the fading at a particular location. The Rician pdf is completely described by the Rician parameter, K, which will differ from location to location within an area. The distribution of the K values over an area will be shown Gaussian if K is represented in dB. Other investigations have been conducted [11] that show that K may be log-normal, but no theoretical reason has been postulated and no use has been found for the model. This investigation will explore both problems. The received power has been shown time and again to be Gaussian if the power is represented in dB. By using this information about the channel, a more accurate process for determining the percent coverage area is obtained.

2.1 Packet Radio

Communication systems that packetize data operate slightly differently than systems that send a continuous stream of bits. In packet radio, a packet of N bits is considered as a single entity. If an error occurs in any one bit in a packet, the entire packet is considered to be in error. It will be shown that in a slow fading channel, the performance of packet radio is better than a system that sends a continuous stream of bits.

In discussing QoS for a wireless communication system, the bit error rate (BER) is a commonly used statistic. It is also possible to develop a similar statistic based on the packet error rate (PER). The assumptions are that the fading is slow compared to the packet length and therefore slow compared to the bit period and the decision made on each individual bit is independent of any other decision on any other bit. Since a packet is considered to be in error if at least one bit in the packet is in error, we want to find the average rate at which packets contain at least one bit error. To begin it is necessary to find the probability that at least one bit error occurs in a packet.

$$\text{Prob(at least one bit error)} = 1 - \text{Prob(all N bits are correct)} \tag{1}$$

Since the assumption was made that the bit decisions are independent, (1) becomes

$$\text{Prob(at least one bit error)} = 1 - \text{Prob(bit 1 is correct)} \times \text{Prob(bit 2 is correct)} \times \ldots \text{Prob(bit N is correct)} = 1 - \text{Prob(correct bit)}^{N} \tag{2}$$

The probability that a correct bit decision is made depends on the modulation technique used and the E_b/N_o. The BER is then represented as $E(\gamma)$, where E is the function that defines the BER for a particular modulation technique and γ is the E_b/N_o. The probability that a correct bit decision is made can then be written as

$$\text{Prob(correct bit)} = 1 - E(\gamma) \tag{3}$$

Using equation (3), equation (2) can then be written as

$$\text{Prob}(\text{at least one bit error}) = E_p(N) = 1 - [1 - E(\gamma)]^N \tag{4}$$

The assumption of slow fading then allows the formulation of an expression for the average packet error rate, $\overline{E}_p(N)$.

$$\overline{E}_p(N) = \int_0^\infty \left\{1 - [1 - E(\gamma)]^N\right\} p(\gamma) d\gamma \tag{5}$$

where, $p(\gamma)$ is the pdf of the fading [33].

Since the BER or PER is usually specified as a QoS requirement of the system, the quantity to be solved for is the required average E_b/N_o, $\bar{\gamma}$, necessary to maintain the specified level of QoS. To compare the BER and PER approaches, use equation (4) to find the equivalent PER, $E_p(N)$, for a specified BER, $E(\gamma)$. For example, if the packet size is assumed to be 424 bits (one ATM cell), and the required BER is specified as 10^{-5}, then the equivalent PER will be 0.00423 as shown in equation (6).

$$E_p(N = 424) = 1 - \left[1 - 10^{-5}\right]^{424} = 0.00423 \qquad (6)$$

This result becomes the left-hand side of equation (5). The final step is to use equation (5) to determine the required $\bar{\gamma}$ that produces the equivalent $\overline{E}_p$ for the specified BER.

Figure 1 is an example of the advantage that can be gained by packetizing data in a fading channel. In this example, the fading is assumed to be slow, flat Rayleigh fading. It can be seen that the required Eb/No to sustain an average BER is 14 dB less if the packet size is 424 bits (ATM cell).

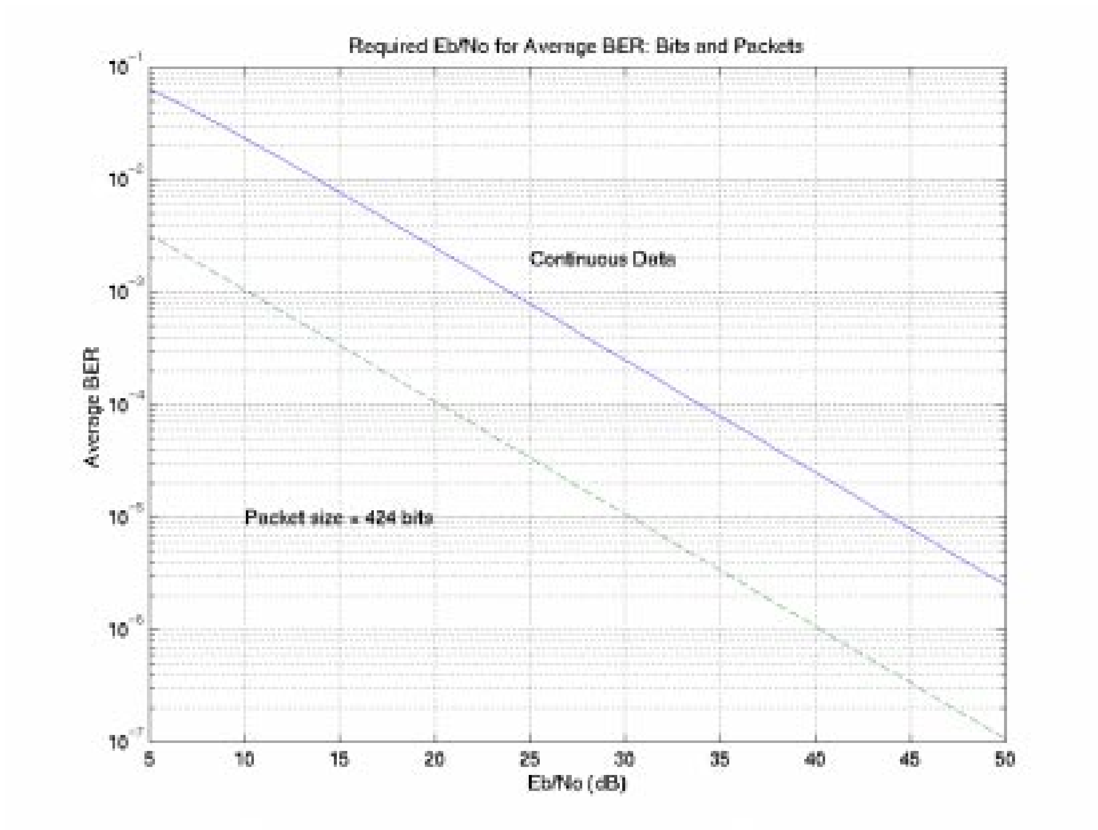

Figure 1: Average BER and non-fading average PER equivalent as a function of Eb/No for continuous data and packets of 424 bits in slow, flat Rayleigh fading.

This can be explained by understanding the nature of the errors that occur in a fading channel. The errors in a fading channel tend to occur during deep fades in signal strength whereas errors in a non-fading channel tend to occur at random intervals. In a non-fading channel, the equivalent PER for a specified BER can be found using equation (4). To maintain this same PER in a fading channel, the actual number of bit errors can increase without decreasing the PER because the bit errors occur in bursts during the deep fades. Since more bit errors can occur without a decrease in performance, less E_b/N_o is required to maintain the link with the desired fidelity.

2.2 Determination of Fading

Fading is a phenomenon caused by short-term changes in the channel that affect the received signal at a particular location. If the instantaneous signal amplitudes can be recorded, the severity of the fading can be determined and other statistics such as BER and packet error rates (PER) can be calculated for that location.

To determine the type of fading at a specific location, instantaneous signal amplitudes must be recorded and a cumulative distribution function (CDF) constructed from that data. This CDF is then compared to a theoretical CDF. The theoretical CDF used in this investigation is the Rician fading distribution for reasons that will be explained shortly. The Kolmogorov-Smirnov goodness-of-fit test is then used to determine how well the experimental data matched the theoretical Rician model. The Kolmogorov-Smirnov goodness-of-fit test is a statistical hypothesis testing technique that compares the CDF of experimental data to a theoretical CDF. It is based on the maximum separation of the two CDFs and the square root of the number of data points. It is relatively simple to implement and offers comparable reliability to other test statistics when the theoretical CDF is completely specified. For more information on the use of this statistic see [2], [5].

2.2.1 Rician Probability Density Function

It has been shown in many cases [26], [29], [30], that the local received signal level resembles a Rayleigh distribution. The Rayleigh distribution is written as,

$$f(r) = \frac{r}{\sigma^2}\exp\left(-\frac{r^2}{2\sigma^2}\right) \quad \text{for } 0 \le r \le \infty$$
$$0 \qquad \text{for} \quad r < 0$$

where,

r = received signal amplitude

σ^2 = variance of the received signal amplitude (7)

Clarke [17] has shown the reason for the Rayleigh distribution by considering the electric and magnetic field components of the signal. These components can be represented by,

$$E_z = E_o \sum_{n=1}^{N} \exp(j\phi_n)$$
$$H_x = -\frac{E_o}{\eta}\sum_{n=1}^{N} \sin(\alpha_n)\cdot \exp(j\phi_n)$$
$$H_y = \frac{E_o}{\eta}\sum_{n=1}^{N} \cos(\alpha_n)\cdot \exp(j\phi_n)$$

where,

E_o = real amplitude of the N waves

η = intrinsic impedance of free space (8)

time variation is of the form $\exp(j\omega t)$

α_n = angle of arrival

ϕ_n = phase of arrival

α_n and ϕ_n are independent

ϕ_n is uniformly distributed between 0 and 2π

As long as N is sufficiently large, the real and imaginary components of each of the three fields are approximately Gaussian with zero mean and equal variance. This is a consequence of the Central Limit Theorem. This implies that the envelope of the three components and therefore the envelope of the signal at the receiver will be Rayleigh distributed. Gans [23] used power spectral densities to reach the same conclusion.

Rice [12] showed that if a dominant signal is present, the distribution becomes the one that bears his name, the Rician distribution.

$$f(r) = \begin{cases} \dfrac{r}{\sigma^2}\exp\left(-\dfrac{\left(r^2 - A^2\right)}{2\sigma^2}\right) I_0\left(\dfrac{Ar}{\sigma^2}\right) & \text{for } A \ge 0,\ r \ge 0 \\ 0 & \text{for } r < 0 \end{cases} \tag{9}$$

where,

r = received signal amplitude

A = peak amplitude of the dominant signal

σ^2 = variance of the received signal amplitude

The Rician distribution can also be expressed in terms of the Rician parameter, K, which is defined as the ratio of the power in the specular component of the signal to the random component, $A^2/2\sigma^2$, and the average E_b/N_o which is written as $\bar{\gamma}$.

$$f(\gamma, K) = \frac{1+K}{\bar{\gamma}}\exp\left(-\frac{\gamma(1+K)+K\bar{\gamma}}{\bar{\gamma}}\right) I_0\left(\sqrt{\frac{4(1+K)K\gamma}{\bar{\gamma}}}\right) \tag{10}$$

The equation in (10) can then be normalized as in (11), where $\Lambda = \gamma/\bar{\gamma}$.

$$f(\Lambda, K) = \frac{1+K}{\bar{\gamma}} \exp\{-[\Lambda(1+K)+K]\} I_0\left(\sqrt{4(1+K)K\Lambda}\right) \tag{11}$$

For K equal to zero, the distribution reduces to the Rayleigh distribution. The Rician distribution has been shown an accurate description of the fading in many cases [24], [27], [28]. The results of this investigation will also show that the Rician distribution is an excellent theoretical model for the measured channel fading in this set of experiments.

2.3 Distribution of the Specular-to-Random Ratio, K

For a specific transmitter-receiver separation, it has been well documented that the long-term average receive powers are distributed log-normally [6], [21], [22], [25], [31]. By examining two extreme cases and making reasonable assumptions about the specular and random components of the received signal, it is possible to postulate a similar distribution for the specular-to-random ratio, K. The Rician probability density function (pdf) is completely defined by K, which is the ratio of the specular signal power to the power in the random component, $A^2/2\sigma^2$. Finding the distributions of the specular signal power and the power in the random component should then lead to a distribution for K.

For the first case, imagine a specific transmitter-receiver separation such that the perimeter of the circle traced out by this radius is such that the surrounding environment is relatively free of clutter, such as an open field. This means that K >> 1 for every point around the circle. For large K, the average receive power, P_{avg}, at a particular location is approximately A^2, where A is the peak amplitude of the dominant signal. Since P_{avg} is known to be log-normal, then A^2 must also be log-normal.

The second case is the opposite scenario. Imagine a transmitter-receiver separation such that the environment surrounding the perimeter of that circle is changing rapidly, such as in an urban area. This means that K << 1 for every point around the circle, so the signal level distribution at each point is approximately Rayleigh. P_{avg} for a Rayleigh distributed signal is $\frac{\sigma^2 \cdot \pi}{2}$. Again, it is known that P_{avg} is log-normal, so in this case, σ^2 must be log-normal.

The two cases demonstrate that A^2 and σ^2 should both be log-normal, but no assumption has been made about the independence of these two quantities. A^2 is the deterministic signal power and depends on the dominant path from the transmitter to the receiver. σ^2, however, depends on the variability of the channel. Since these two quantities depend on different processes, it can be assumed that the two are independent.

K is defined as $A^2/2\sigma^2$. We have previously postulated that A^2 and σ^2 should be independent and log-normal. This means that K(dB) is the difference of two normal random variables. The difference of two normal random variables is itself normally distributed, so K must be log-normal. For a proof that the distribution of the difference of two independent Gaussian random variables is itself Gaussian, see Appendix L.

To determine if the distribution of K is log-normal, the Cramer-von Mises goodness of fit test will be used. This technique is a more powerful technique than the Kolmogorov-Smirnov technique to use when the test is for normality and the average value and standard deviation must be estimated from the data [5].

This log-normal K result is shown true experimentally in the results section. Others have conducted experimental tests that have shown the variation in K over an area is "near-normal"[11], but the results were disregarded because the variation was deemed too large and therefore considered of little practical use. It will be shown in this thesis that the information concerning the log-normal nature of K can be used with benefit when determining the percent coverage area.

2.4 Determination of Receive Power Distribution

In determining a good theoretical model for the distribution of the average receive power, the Central Limit Theorem plays an important role. In general, the theorem states that the sum of independent random variables has a distribution that is approximately Gaussian and converges to a Gaussian distribution as the number of independent random variables approaches infinity [16]. In a wireless channel, the received signal is the result of the transmitted signal passing through, reflecting off and scattering from objects that lie between the transmitter and the receiver. Each of these processes attenuates the signal somewhat, so the final received amplitude is the product of many transmission factors. Since the received signal is a product of these factors, taking the logarithm of the received signal implies that it is the sum of the logarithm of many terms. The Central Limit Theorem then states that if these terms are independent with finite variances, the received signal in dB should have a distribution that is Gaussian. This has been shown in many cases [6], [20], [21], [22], [25], [31]. The Cramer-von Mises goodness of fit test will again be used as the test for normality. The results of this investigation will also show the received power to be Gaussian when it is represented in dB.

2.4.1 Power Law and Determination of the Path Loss Exponent

To account for the distance dependence of the received signal power, two models will be discussed, the power law model and the two-ray model. The power law model defines the average receive power as a linear function of the logarithm of the distance. The two-ray model assumes that the signal that reaches the receiver is a combination of two versions of the transmitted signal arriving at the receiver over different paths.

The amount of decrease in signal strength over distance is called the path loss, l_s. In general, the path loss can be calculated as,

$$l_s = \left(\frac{4\pi rf}{c}\right)^n \tag{12}$$

where,

r = Transmitter-receiver separation in meters
f = Carrier frequency in Hz
c = Speed of light in meters/second
n = Path loss exponent

Normally, the path loss is expressed in dB as,

$$L_s = n \cdot 10 \cdot \log_{10}\left(\frac{4\pi rf}{c}\right) \tag{13}$$

It can be seen that the path loss in dB is a linear function of the path loss exponent. In free space, the path loss exponent is 2. In some cases, however, the actual path loss exponent can be less and sometimes as high as 4. In this investigation, the measured value of the path loss exponent for the LOS locations was 2.232, and was 3.8101 for the OBS locations. As an example, for a transmitter-receiver separation of 1 km and a frequency of 5.8 GHz, the path loss for the LOS locations is 120.2 dB whereas the path loss for the OBS locations is 204.7 dB. This example shows that an accurate determination of the path loss exponent is essential in determining the amount of power necessary to close the link at a certain distance.

The path loss exponent can be determined for an area by using the data collected for determining fading. The mean signal amplitude is found for each location by simply averaging the data collected for fading. These amplitudes can then be plotted as a function of the logarithm of the distance from the transmitter. Since the assumption can be made that the shadowing has a normal distribution in dB (log-normal), linear regression can be used to determine the best line, in a minimum mean square error sense, through the data. The slope of that line is the path loss exponent.

2.4.2 Two-Ray Model

The two-ray model is another model used to describe path loss. The basic geometry is shown in Figure 2 where the received signal is a combination of a direct line-of-sight (LOS) component and a component reflected off the ground.

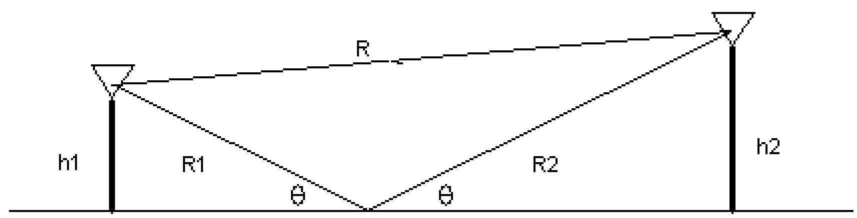

Figure 2: Two-Ray Model Geometry.

The received signal is the sum of the signal that travels along R plus the signal that travels along R1 through R2. The difference in the lengths each signal has to travel causes a phase difference between the two. These phases can either add constructively or destructively thus increasing or decreasing the received signal strength. This model is normally used as an introduction to multipath since it is consists of two paths and is a tractable problem. The results of this investigation will show, however, that this model is a better model for LOS paths than using the standard power law model where the average receive power falls off only as a function of distance.

2.5 Determination of Percent Coverage Area

Before beginning the development of the percent coverage area formula, two quantities must first be determined. It is necessary to determine the received signal power required to allow the transmission to be received with the desired fidelity. This is called closing the link. The required received signal power will be determined by using an equation that balances the carrier-to-noise ratio that is available with the carrier-to-noise ratio that is required. The second quantity to be determined is the correlation coefficient between the Rician parameter, K, and the received signal power. It will be seen that the required received signal power is a function of K and that K and the actual received signal power are correlated.

2.5.1 Link Equation

The basic link equation used to determine a wireless communication system's ability to deliver a certain level of quality of service is [33]

$$\frac{E_b}{N_o} + R_d + M = ERP - L_s - k + G_r - T_s \tag{14}$$

where,

$\frac{E_b}{N_o}$ = Required bit energy to noise ratio in dB to maintain a specified BER (PER)
R_d = Data rate in dB
M = Link margin in dB
ERP = Effective radiated power (transmit power in dB + antenna gain in dB)
L_s = Path loss in dB
k = Boltzman's constant in dB
G_r = Receive antenna gain in dB
T_s = Receive system temperature in dB.

The left-hand side of the equation is the carrier to noise ratio required and the right-hand side is the carrier to noise ratio that is available.

The Rician parameter, K, sets the $\frac{E_b}{N_o}$ requirement. The link equation can then be manipulated to determine any of the components given any of the others. The ERP required to close the link for average path loss and K values over distance can be determined or given all of the system parameters, the link margin can be determined. In this investigation, (14) will be used to determine the required receive power for the percent coverage area calculations.

2.5.2 Correlation between K and the Receive Power

The next section discusses the development of a percent coverage area equation. The first step in the development of the equation is the formation of the joint pdf of K and the receive power. We have previously determined that both K and the receive power are log-normally distributed. If these two random variables are independent, their joint pdf is simply the product of each of their marginal pdfs. If these two random variables are correlated, their joint pdf is a bivariate Gaussian with a finite correlation coefficient, ρ.

By comparing the definitions of K and the receive power, it becomes clear that a correlation between these two random variables should exist. K is defined as the ratio of the specular component of the power to the random component. The received power is the sum of these two components.

$$K = \frac{P_{spec}}{P_{rand}} \quad ; \quad P_{receive} = P_{spec} + P_{rand} \tag{15}$$

$$P_{receive} = P_{rand}(1+K) = P_{spec}\left(1+\frac{1}{K}\right) \tag{16}$$

This relationship in (15) shows that there should be a correlation between K and the receive power since they are functions of the same two quantities. This will be shown true experimentally in the results section.

2.6 Development of Percent Coverage Area Formula

For cellular systems, the percent coverage area is normally determined assuming Rayleigh fading over the entire cell [19], [35], [36]. During this investigation, knowledge will be gained concerning the fading characteristics of the region. The idea is to then use this knowledge to gain a more accurate account of the percentage of the region for which the link closes with the desired fidelity. As will be seen in the results, a significant difference exists between assuming Rayleigh fading as opposed to using the experimental results.

The first step in developing the equation for percent coverage area is determining the joint probability density function between the Rician parameter, K, and the receive power, Pr. Since both K and Pr are log-normal and correlated, the joint pdf is a bivariate Gaussian of the form,

$$f_{\mathrm{Pr},K}(\mathrm{Pr},K) = \frac{1}{2\pi\sigma_{\mathrm{Pr}}\sigma_K\sqrt{1-\rho^2}} \cdot \exp\left\{\frac{-1}{2(1-\rho^2)} \cdot \left[\left(\frac{\mathrm{Pr}-\mu_{\mathrm{Pr}}}{\sigma_{\mathrm{Pr}}}\right)^2 + \left(\frac{K-\mu_K}{\sigma_K}\right)^2 - \left(\frac{2\rho(\mathrm{Pr}-\mu_{\mathrm{Pr}})(K-\mu_K)}{\sigma_{\mathrm{Pr}}\sigma_K}\right)\right]\right\} \quad (17)$$

where,

$\rho =$ Correlation coefficient between K and Pr

$\sigma_{\mathrm{Pr}} =$ Standard deviation of the receive power

$\sigma_K =$ Standard deviation of the Rician parameter

$\mu_{\Pr}$ = Mean receive power at a particular distance
μ_K = Mean Rician parameter at a particular distance.

The probability that the link closes at a specific distance, r_j, can then be found by integrating the joint pdf over K and the receive power, Pr . The integration should be done over all values of K and from $\Pr_{req}$ to infinity where $\Pr_{req}$ is a function of K. $\Pr_{req}$ is found by manipulating the link equation stated in (14) to the form,

$$ERP - L_s = \frac{E_b}{N_o} + R_d + M + k - G_r + T_s \tag{18}$$

The left-hand side of equation (18) is now the required receive power, $\Pr_{req}$. $\frac{E_b}{N_o}$ is determined by K and the required BER, so $\Pr_{req}$ is a function of K. The probability that the link will close at a certain distance is then given as,

$$P[close\ link\,]_{r_j} = \int_{-\infty}^{\infty} \int_{\Pr_{req}(K)}^{\infty} f_{\Pr,K}(\Pr, K)\, d\Pr\, dK \tag{19}$$

Examples of the joint pdfs are shown in Figures 3 and 4 for the measured data with the determined positive correlation and Figures 5 and 6 for no correlation (independent Gaussian random variables). The first of each pair is the joint pdf at a distance of 100m and the second is at a distance of 400m. It can be seen that as the distance increases, less of the volume under the pdf is above the required received signal threshold. This should be expected since the required received signal level is fixed and the average received power decreases with distance. It will also be seen that the average K value will decrease with distance.

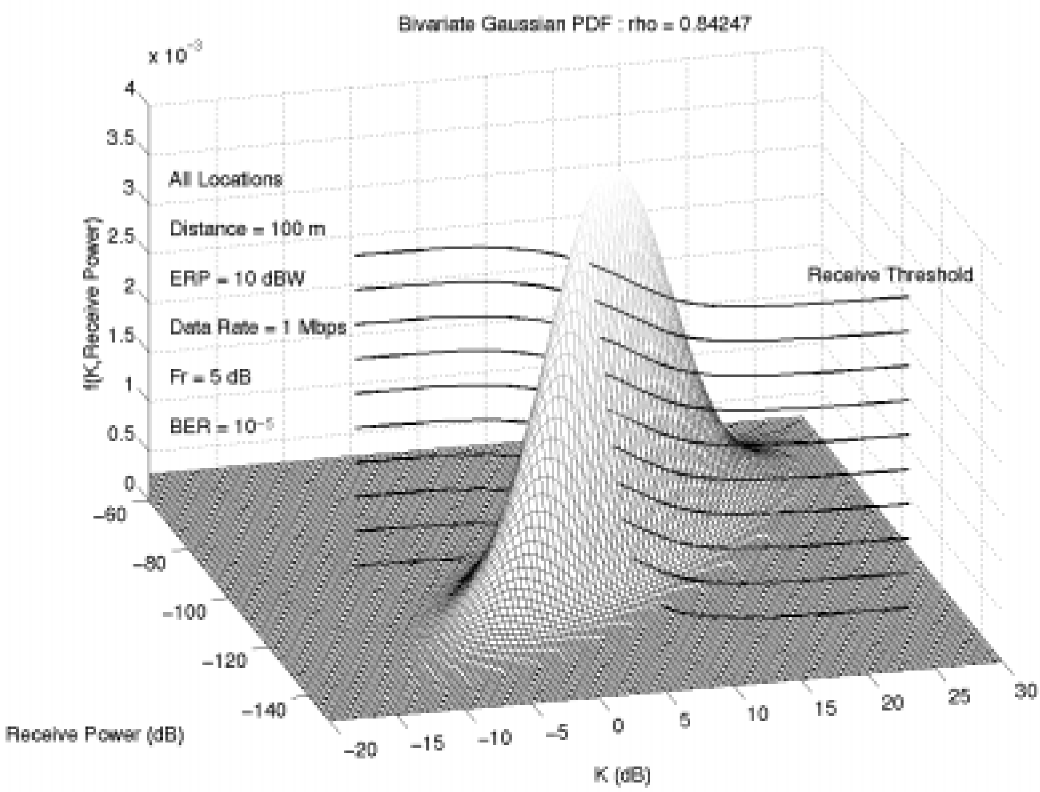

Figure 3: Bivariate Gaussian probability density function using the average K and average received powers determined experimentally for a transmitter-receiver separation of 100m.

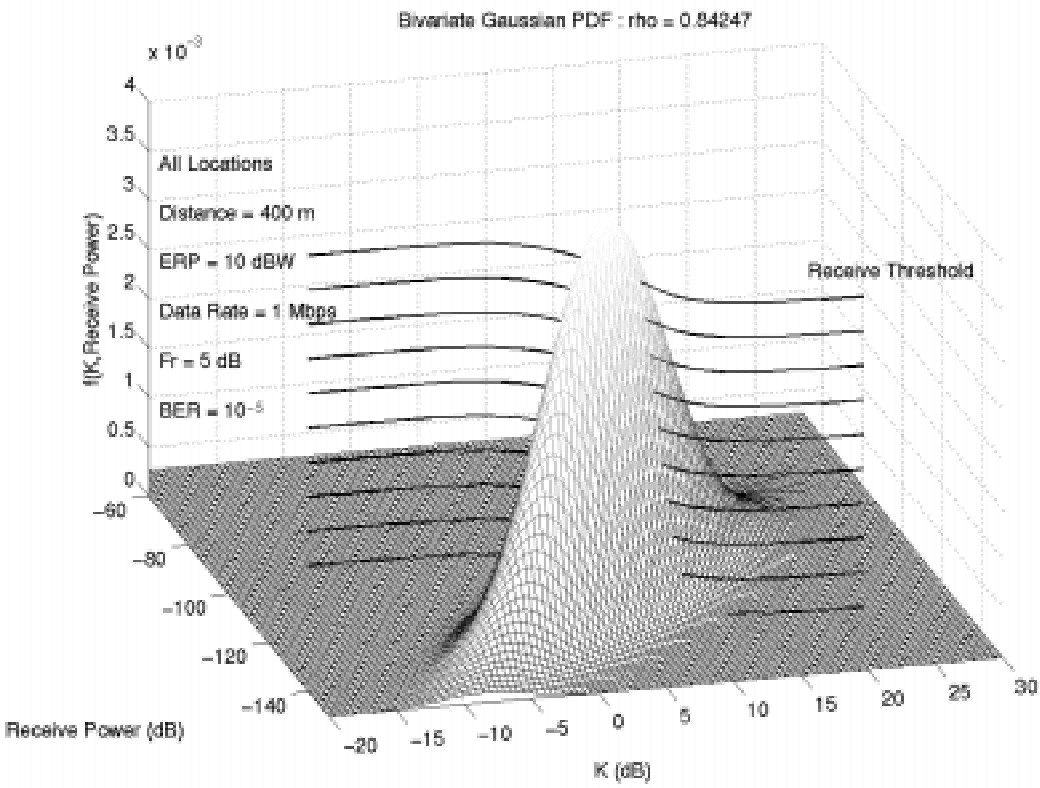

Figure 4: Bivariate Gaussian probability density function using the average K and average received powers determined experimentally for a transmitter-receiver separation of 400m.

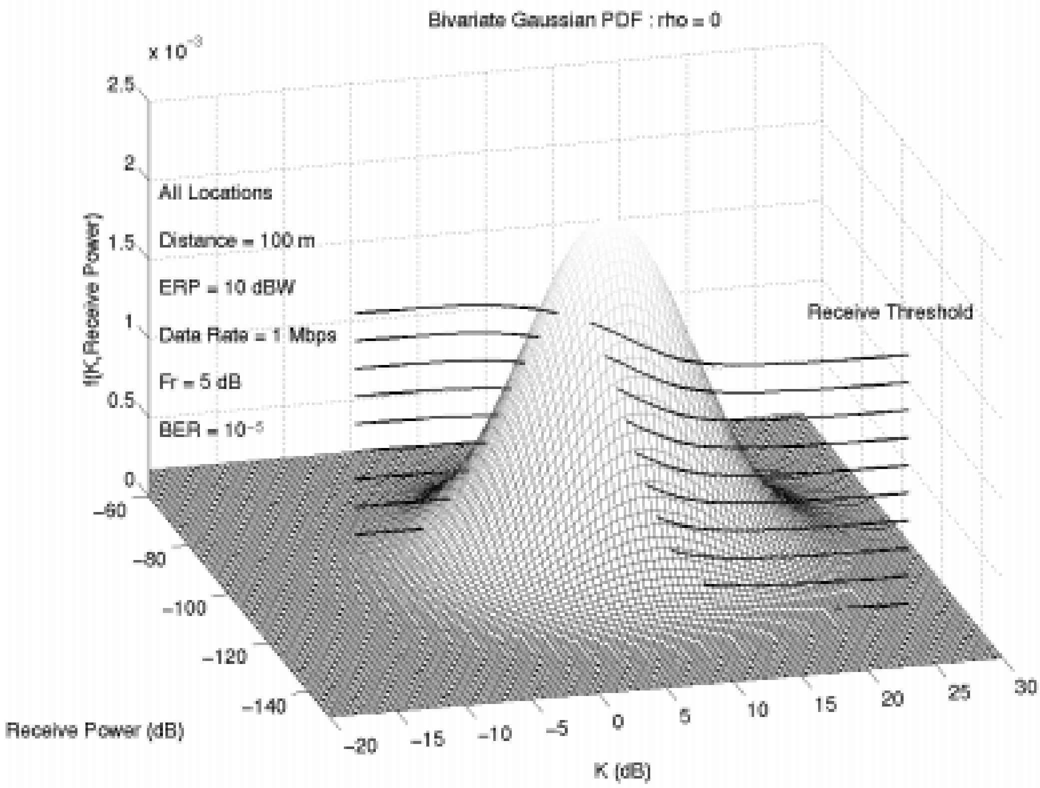

Figure 5: Bivariate Gaussian probability density function for independent K and receive power at a transmitter-receiver separation of 100m.

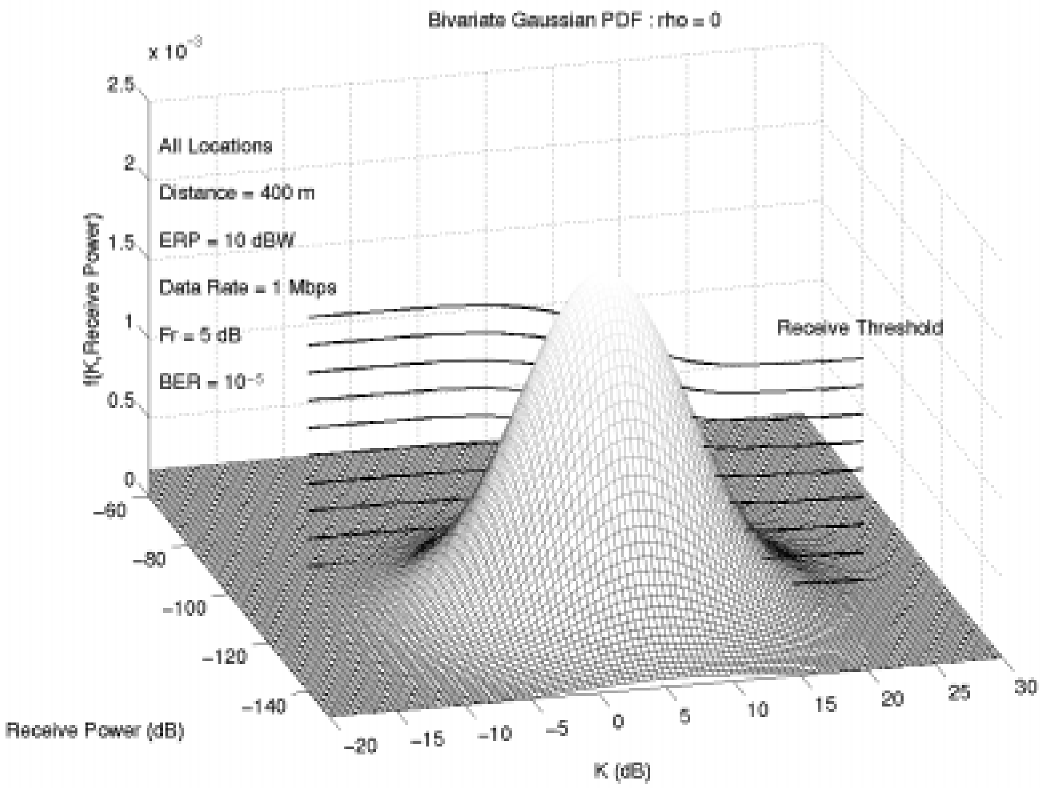

Figure 6: Bivariate Gaussian probability density function for independent K and receive power at a transmitter-receiver separation of 400m.

A closed form solution of the double integral in (19) was not found, so the answer was found numerically. To do this, the infinite limits must be approximated. It was decided to evaluate the integrals within 3σ of the means of both K and Pr. Since both random variables are log-normal, the new limits will cover approximately 99.7% of the area. The new integral then looks like,

$$P\left[close\ link\ \right]_{r_j} \cong \int_{\mu_K - 3\sigma_K}^{\mu_K + 3\sigma_K} \int_{\Pr_{req}(K)}^{\mu_{\Pr} + 3\sigma_{\Pr}} f_{\Pr,K}(\Pr, K)\, d\Pr\, dK \tag{20}$$

The definition of the definite integral is [13],

$$\int_a^b f(x)\,dx = \lim_{\Delta x \to 0} \sum_{i=1}^{n} f(c_i)\,\Delta x \tag{21}$$

Using the definition in (21),

$$P\left[close\ link\ \right]_{r_j} \cong \lim_{\Delta K \to 0} \sum_{i=1}^{n} \int_{\Pr_{req}(k_i)}^{\mu_{\Pr} + 3\sigma_{\Pr}} f(\Pr, K = k_i)\, d\Pr\, \Delta K \tag{22}$$

where,

$$k_1 = \mu_K - 3\sigma_K$$
$$k_n = \mu_K + 3\sigma_K$$
$$\Delta K = k_i - k_{i-1}$$

Then for a reasonably small ΔK, determined by experimentation,

$$P\left[close\ link\ \right]_{r_j} \cong \Delta K \sum_{i=1}^{n} \int_{\Pr_{req}(K_i)}^{\mu_{\Pr} + 3\sigma_{\Pr}} f(\Pr, K = k_i)\, d\Pr \tag{23}$$

The probability that the link closes for a specific distance is approximated by (23), so the next step is to determine for what fraction of a circular area with radius R will the link close. This is done by integrating equation (23) over the entire area.

$$Fractional\ Coverage = \frac{1}{\pi R^2} \int_0^{2\pi} \int_0^{R} P[close\, link]_r \ r\, dr \ \ d\theta \tag{24}$$

The probability that the link closes does not depend on θ, so that can be integrated out.

$$Fractional\ Coverage = \frac{2\pi}{\pi R^2} \int_0^{R} P[close\ link]_r \ r\ dr \tag{25}$$

Now use the definition for a definite integral (21),

$$Fractional\ Coverage = \frac{2}{R^2} \lim_{\Delta r \to 0} \sum_{j=1}^{n} P[close\ link]_{r_j} \left[\frac{1}{2} \left(r_j + r_{j-1} \right) \right] \Delta r \tag{26}$$

where,

$r_0 = 0$
$\Delta r = r_j - r_{j-1}$

Approximate the integral,

$$Fractional\ Coverage \cong \frac{1}{R^2} \sum_{j=1}^{n} P[close\ link]_{r_j} \left(r_j + r_{j-1} \right) \left(r_j - r_{j-1} \right) \tag{27}$$

Then,

$$Percent\ Coverage \cong 100 \cdot \sum_{j=1}^{n} P[close\ link]_{r_j} \frac{\left(r_j^2 - r_{j-1}^2 \right)}{R^2} \tag{28}$$

3 Description of Measurement System and Experimental Procedure

3.1 Equipment

The measurement system consists of a Toshiba Tecra 700 laptop computer running Windows 95, a Hewlett-Packard HP8593E spectrum analyzer and a Noteworthy NWGPS01 Type II PC card Global Positioning Satellite (GPS) receiver. The laptop is used to control both the spectrum analyzer and GPS receiver. Received signal amplitudes are recorded on the laptop and each data point is time and location stamped using information provided by the GPS receiver.

3.2 Operation

The spectrum analyzer is connected to the laptop through a Type II PC card GP-IB interface. The GPS receiver is also connected to the laptop through a Type II PC card. For a full description of the operation of the measurement system used in this investigation, see Appendix N.

3.3 System Verification

The measurement system was verified using a channel simulator developed by the University of Kansas and a handheld Global Positioning Satellite (GPS) receiver. The channel simulator takes as its input fading statistics, so that when a signal is input, the output is an appropriately faded version of the input. The GPS portion of the system was verified by comparing the average location obtained using the GPS receiver with the average location obtained using the measurement system.

The signal power recording portion of the system was verified by using a 450 MHz tone as the input signal to the channel simulator with Rician K values of 0, 3, and 6 dB. The output of the channel simulator was connected to a spectrum analyzer and the signal powers were recorded using the measurement system. The results for each case were K = 0 dB, 2.55 dB and 6 dB, respectively.

3.4 Determination of Locations

It is important in any type of sampling study to obtain a random population. The same is true here. If the locations are more line-of-sight (LOS) than obstructed (OBS), it is possible that the path loss exponent obtained will be too optimistic. On the other hand, if the opposite is true, the path loss results could be such that the amount of power required is in excess of what is really needed. Given this, the first step is to determine which locations will be used in the experiment. Figure 7 is a map of the University of Kansas West Campus. The transmitter was located on the roof of Nichols Hall and the circled areas were the locations that the receiver was placed and data collected.

The decision was made to do the analysis in the following sections for line-of-sight (LOS) locations only, obstructed (OBS) path locations only and all (ALL) of the locations taken together. It will be seen that the receive power and K distributions are significantly different for the LOS and OBS locations. The definition used for the LOS locations was that a direct path could be seen between the transmitter and receiver. All of the other locations were considered OBS. The case where all locations are taken together has been included for completeness. Once the data for the particular locations have been collected, it is possible to determine the fading for those locations, the path loss exponent for the area and the percent coverage area using the Matlab™ programs found in the appendix.

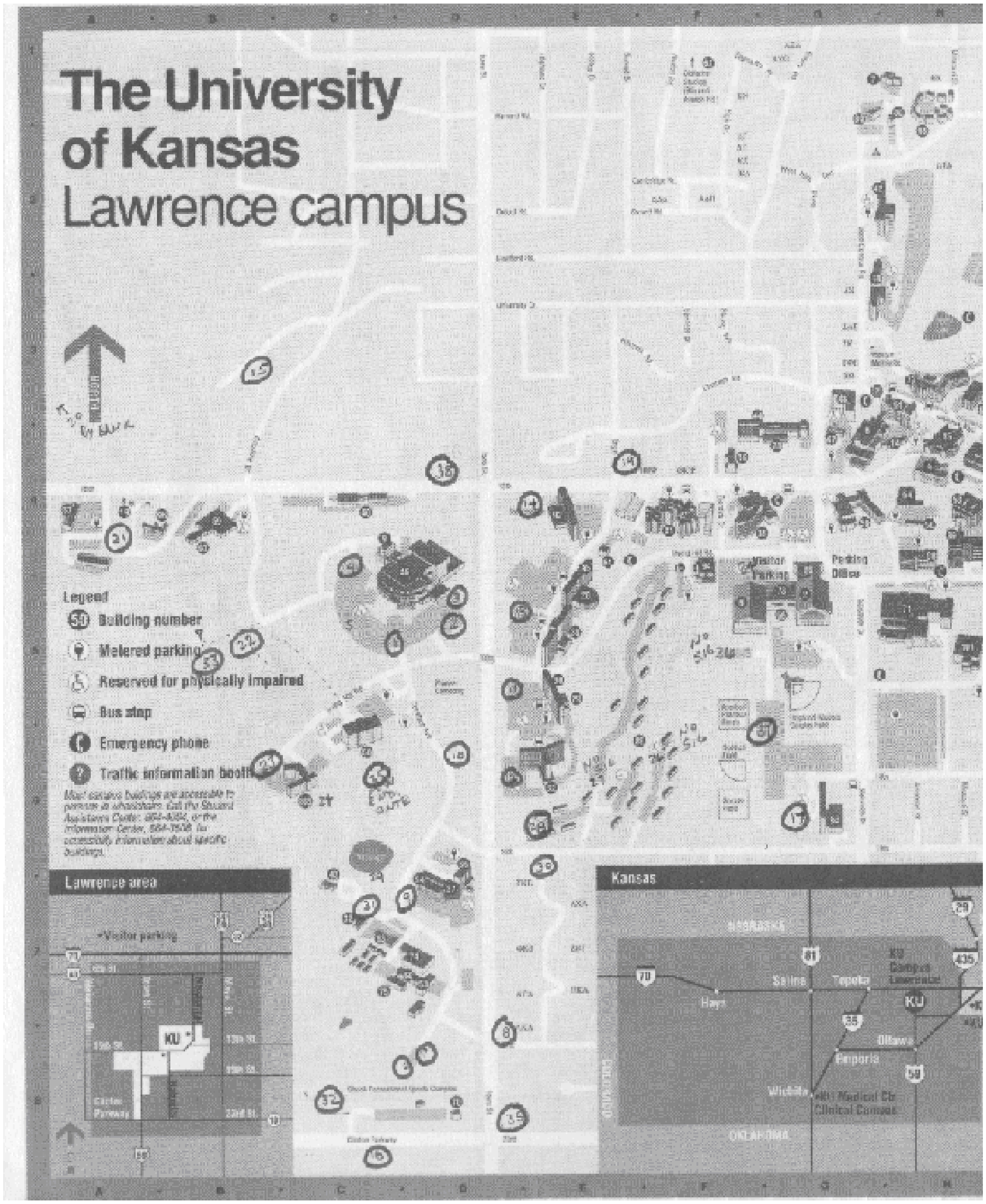

Figure 7: Locations used in the investigation.

3.5 Determination of Fading

The severity of the fading of the received signal at a particular location was determined using the program *sainfo20.m* found in Appendix C. This program constructs a CDF of the collected data and compares it to a theoretical Rician CDF. The input file name was the raw data and the value of K was estimated. The output is a graph of the experimental CDF and the theoretical Rician CDF with the estimated K value. The K-S statistic that appears is a numerical representation of how well the estimated CDF matches the experimental CDF. A K-S statistic of less than 1.518 represents a significance level less than 0.01 [5]. The mean signal level is also calculated and displayed. This value was recorded and used in the calculation of the path loss exponent.

3.6 Determination of Path Loss Exponent

Once the fading of all of the locations has been determined, the path loss exponent for the area can be calculated. In the Matlab™ file *linreg.m,* found in Appendix D, the distances and the mean signal levels for each location were placed in the appropriate vectors. The program calculates the best line (in a mean square error sense) through the data points. The output of the program is the slope of the line (path loss exponent), standard deviation and confidence interval given a particular confidence level. The path loss exponent and standard deviation can then be used to find a percent coverage area given a minimum required signal level.

3.7 Determination of Average K vs. Distance

Since the pdf of K is postulated to be log-normal, a similar analysis can be done for the average K over distance. The program *linreg.m* in Appendix D was used again, only this time the collected K values in dB instead of the mean signal levels were used. The output is the slope of the best line through the data and the standard deviation from that line. This information was used in the determination of the percent coverage area.

3.8 Test for Normality

The use of linear regression for determining the path loss and average K over distance is only valid if the distribution at every distance is normal. To test for normality, the program *cvmdiff.m,* found in Appendix J, was used. The output of the program is a graph of the CDF for the shadowing or K along with the mean and standard deviation. The Cramer-von Mises test was used and if the test statistic is below 0.178, the distribution is normal with 0.01 significance.

3.9 Closing the Link for Average K and Average Path Loss

To determine the ERP needed to close the link with a specified fidelity, the average K and received power were determined experimentally over distance and the system parameters were set for the RDRN specifications. The program to determine the effective radiated power (ERP) is *closelink.m* and can be found in Appendix G. It is necessary to set the equations for both the average K and average path loss. This was done by using the information gained from running *linreg.m* for K and the path loss. The rest of the variables in the program are the system parameters and needed to be set according to the specific system requirements.

The output of the program is a graph that shows the transmit power necessary to close the link as a function of distance for the given system parameters and requirements. It is important to note that the results of this program will merely give an average required transmit power. Due to the log-normal shadowing and K distributions, to close the link at a specific location, the actual values of K and path loss must be used to determine the necessary transmit power.

3.10 Determination of Correlation Coefficient, ρ

It was noted in the background section that there is expected to be a correlation between the received power and K. To determine the strength and direction of this correlation the program *allcor.m* found in Appendix H was used. The elements of the first column of the matrix are the K values in dB for each location. The elements of the second column are the corresponding deviations from the mean path loss in dB determined using *linreg.m*. In this investigation, shadowing is defined as the difference between the actual path loss recorded at a distance and the average path loss expected at that distance. The output of the program is a scatter diagram of K vs. shadowing with the correlation coefficient, ρ. The results of this program were then used to determine the percent coverage area.

3.11 Determination of Percent Coverage Area

The percent coverage area is found by determining the average received power and average K over distance. The probability that the link closes for each distance is then determined by using numerical integration over the specified limits. Each probability is then weighted according to the fraction of the entire area that the annulus defined by the current distance and the previous distance covers to determine the percent coverage area.

The program used to calculate the percent coverage area is called *cvrg_corr1A.m* and can be found in Appendix I. The variables that need to be set are commented in the code. The distances used for the variable r were such that the difference between two successive values is no more than 10 meters. It was found that for differences less than 10 meters, the resulting increase in accuracy was negligible. The appropriate values for the K and path loss vectors were obtained from the results of the linear regression analysis done on the two quantities. The required $\frac{E_b}{N_o}$ values were determined for values of K and specified BER and PER. Tables of these values were obtained by loading either *ebno_BER.mat* for bit error rate or *ebno_PER.mat* for packet error rate. These are tables of $\frac{E_b}{N_o}$ requirements generated by first running the programs in Appendix E and Appendix F.

4 Results

4.1 Determination of Fading and Average Receive Power

The received power data for each location was analyzed to determine the Rician parameter, K, and the average receive power. The CDFs used to determine these quantities can be found in Appendix A. Table 1 is a listing of the results found for each location. A K-S statistic of less than 1.518 represents a significance level less than 0.01 [5]. All of the locations investigated yielded a K-S statistic much less than 1.518. This shows that the Rician distribution is an excellent theoretical model for fading in this set of experiments.

Location	Mean Signal Level (dB)	K (dB)	K-S Statistic
a1	-94.51	3.5	0.5656
a2	-96.99	4.4	0.3943
a3a	-77.15	18.7	0.9442
a4	-99.5	3.7	0.5919
a6	-98.51	10	0.622
a7	-103.2	5	0.4159
a8	-110.9	-2	0.7583
a9	-88.11	6.9	0.4844
a10	-70.53	13.6	0.5338
a11	-98.09	6.5	0.3823
a12	-103.4	-1	0.5011
a13	-93.61	11.8	0.5215
a14	-99.63	6	0.3109
a15	-109.2	-1	0.8597
a16a	-83.35	13.5	0.7303
a17	-122.3	1	0.6884
a19	-117.6	-0.6164	0.6092
a21	-125.6	-2	0.7174
a22	-100.1	5	0.723
a23	-74.28	10.94	0.4209
a25	-68.19	12.91	0.5594
a28	-112.3	-1.7	0.8081
a30	-121.1	-9	0.6998
a31	-100.9	-0.7	0.7712
a32	-90.85	13	0.5068
a33	-95.72	4	0.6219
a35	-93.26	6.808	0.3612
a37	-125.7	3.77	0.3582
a38	-109.2	-0.6435	0.568

Table 1: Mean signal level, K and Kolmogorov-Smirnoff statistic for each location.

4.2 Distribution of K

Figures 8-10 show the average value of K over distance for line-of-sight (LOS), obstructed (OBS) and all locations taken together (ALL). Linear regression was used to determine the best fit line for each case. These figures show a major difference between the LOS and OBS cases. While K is relatively constant (approximately 10.5 dB) over the distance investigated for the LOS case, the K for the OBS case quickly deteriorates towards the Rayleigh value. Figure 11 is the result of the Cramer-von Mises test to determine if the deviation of K from a distance dependent mean is log-normal for the case of all locations. A C-vM statistic of less than 0.178 implies the distribution is normal with 0.01 significance [5]. Figure 11 shows the C-vM statistic to be 0.0477, which implies that a log-normal distribution is an excellent theoretical model for the distribution of K.

The log-normal distribution of K values is a significant discovery. Since the distribution of K can be modeled as a log-normal random variable, the effects of K on the performance of a wireless communication system can be handled mathematically. This more accurate model of the channel can be used to make intelligent decisions about the system design and more accurately determine the expected system performance.

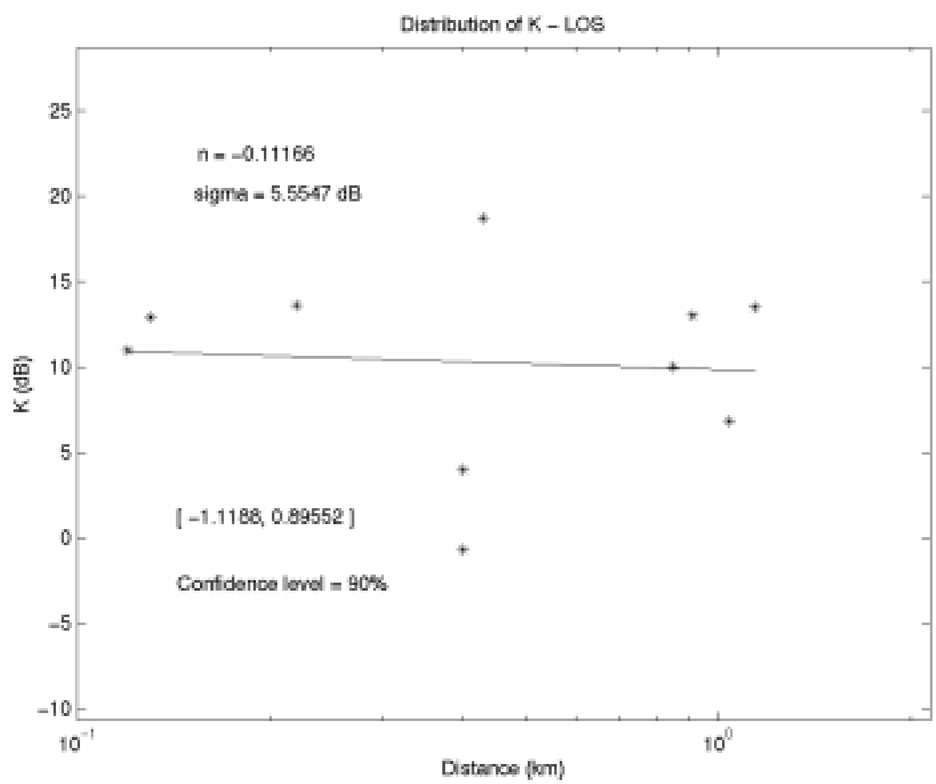

Figure 8 : Distribution of K over distance for LOS locations.

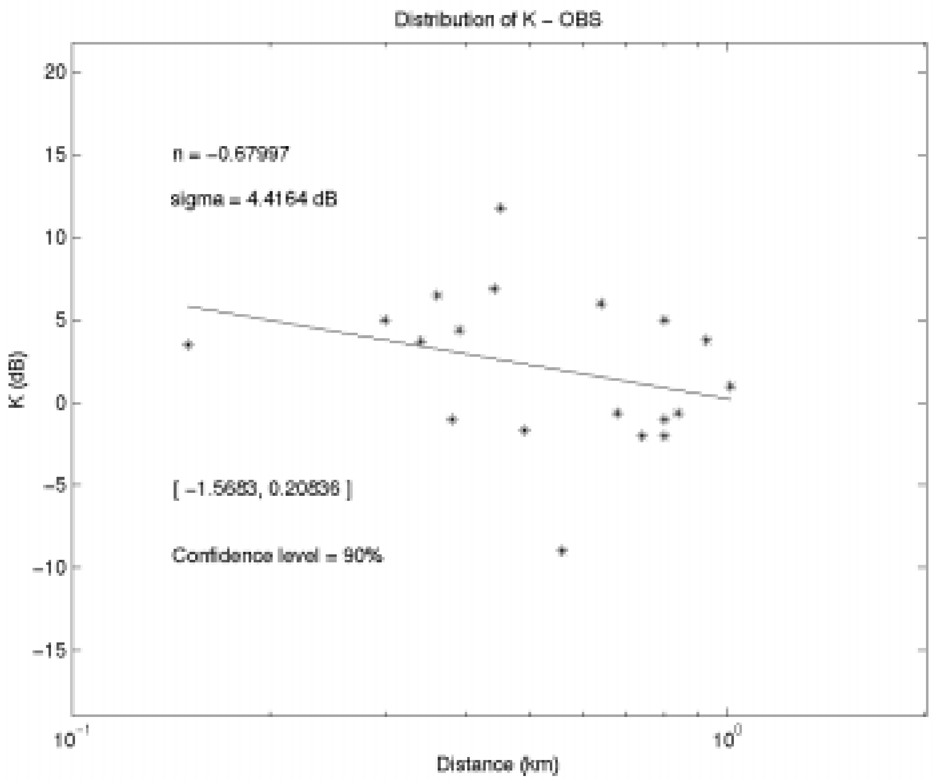

Figure 9: Distribution of K over distance for OBS locations.

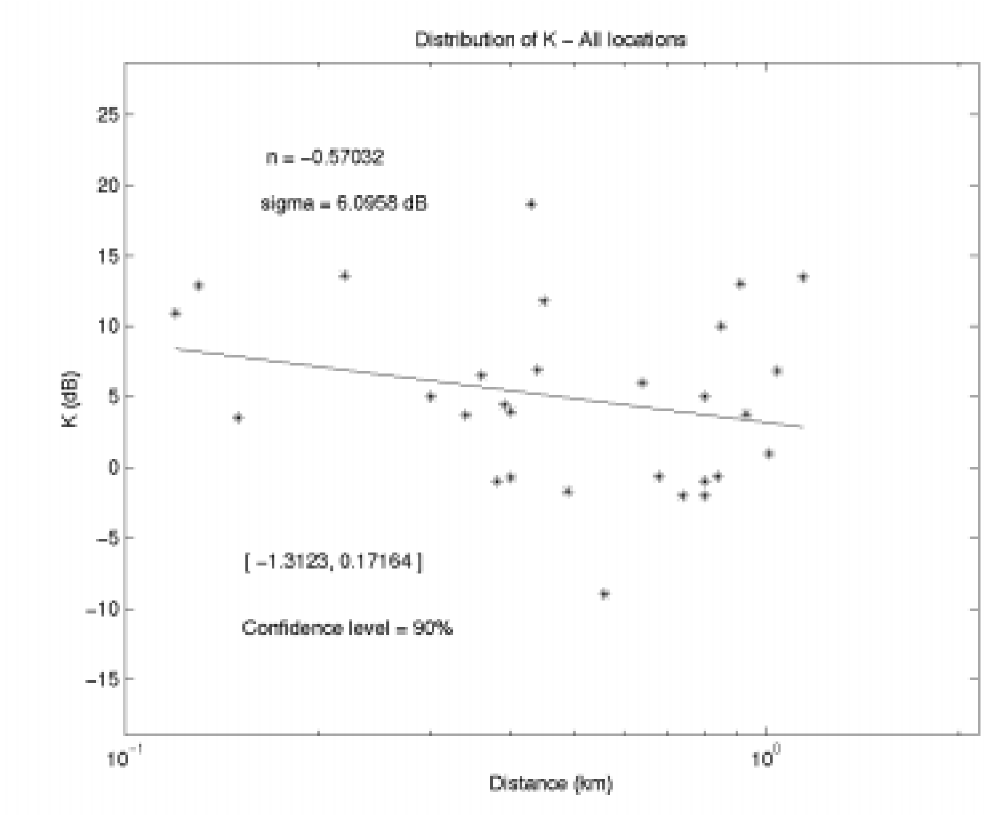

Figure 10: Distribution of K over distance for ALL locations.

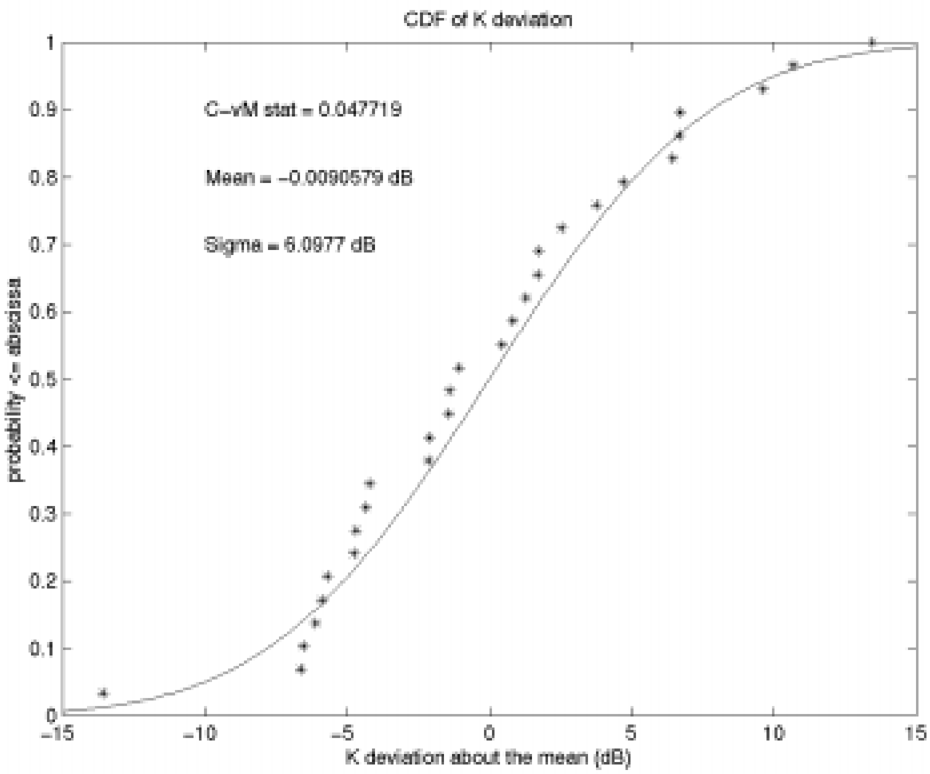

Figure 11: Cramer-von Mises goodness of fit test result for log-normal K deviation distribution.

4.3 Distribution of Path Loss

Figures 12-14 show the average path loss over distance for line-of-sight (LOS), obstructed (OBS) and all locations taken together (ALL). Linear regression was used to determine the best fit line for each case. Again, it is clear that the separation of the locations into LOS and OBS cases shows a marked difference in the path loss exponent between the two. The path loss exponent found for the LOS case was 2.232 and for the OBS case the path loss exponent was found to be 3.8108. Figure 15 is the result of the Cramer-von Mises test to determine if the deviation (shadowing) from a distance dependent mean is log-normal. The C-vM statistic was found to be 0.0493, which is significantly less than the 0.178 value used for 0.01 significance [5]. This confirms that the log-normal distribution is an excellent model for the shadowing distribution in this set of experiments.

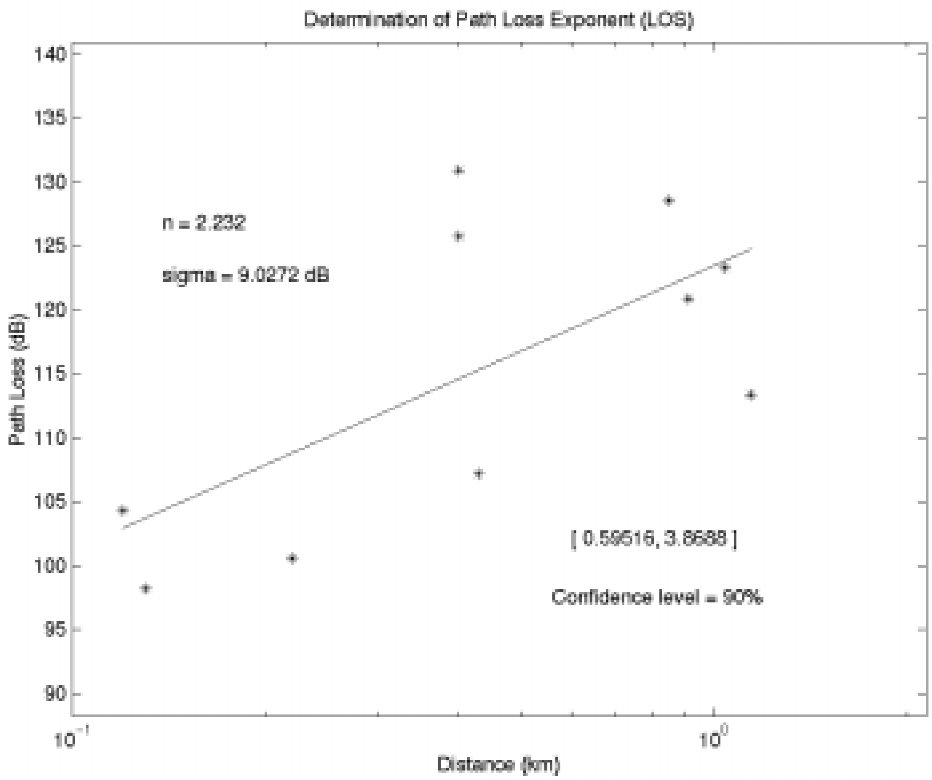

Figure 12: Path loss exponent determination for LOS locations.

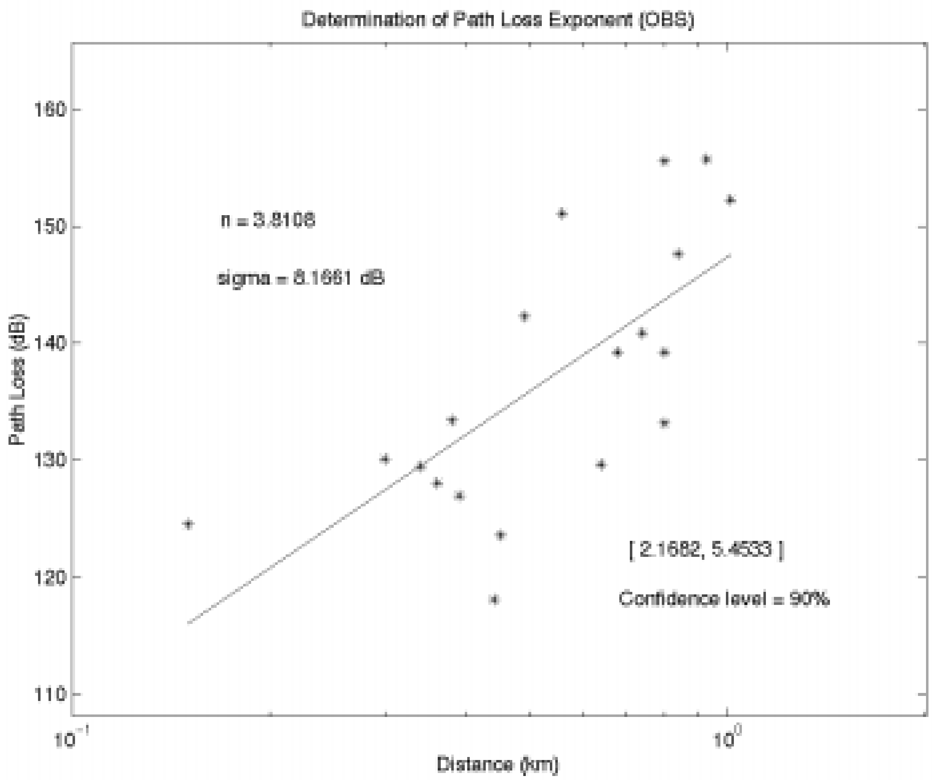

Figure 13: Path loss exponent determination for OBS locations.

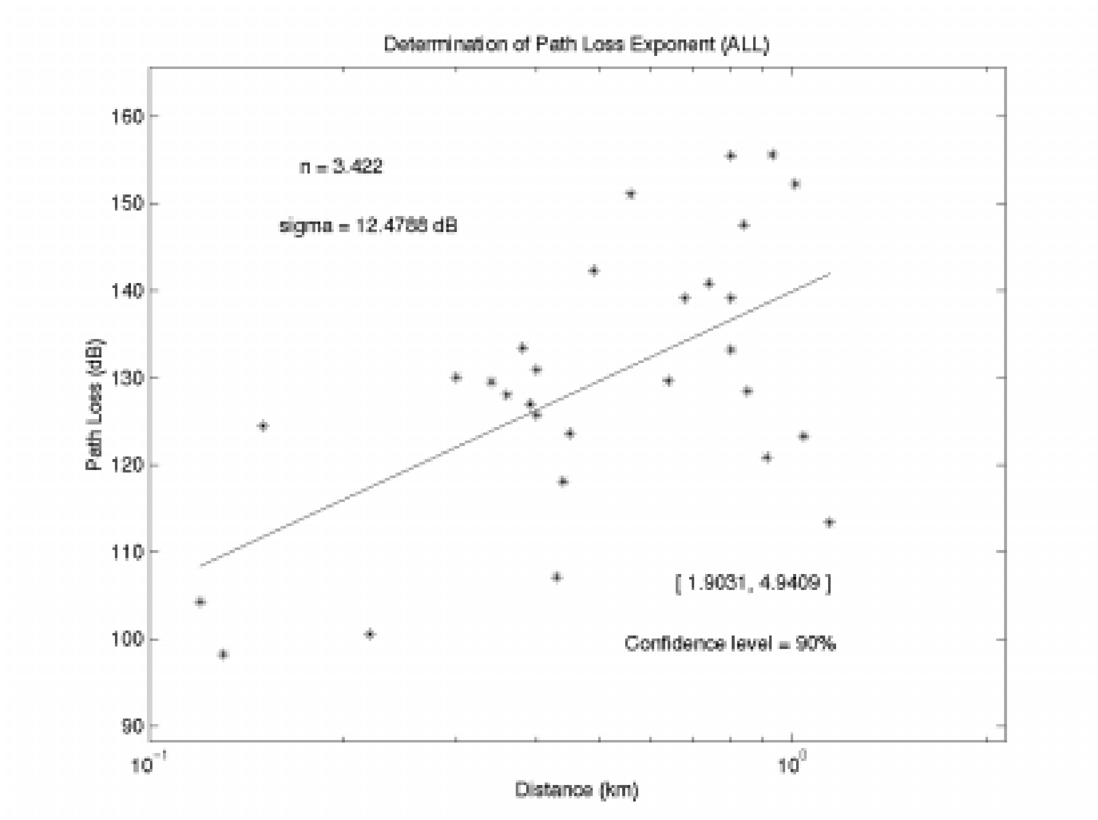

Figure 14: Path loss exponent determination for ALL locations.

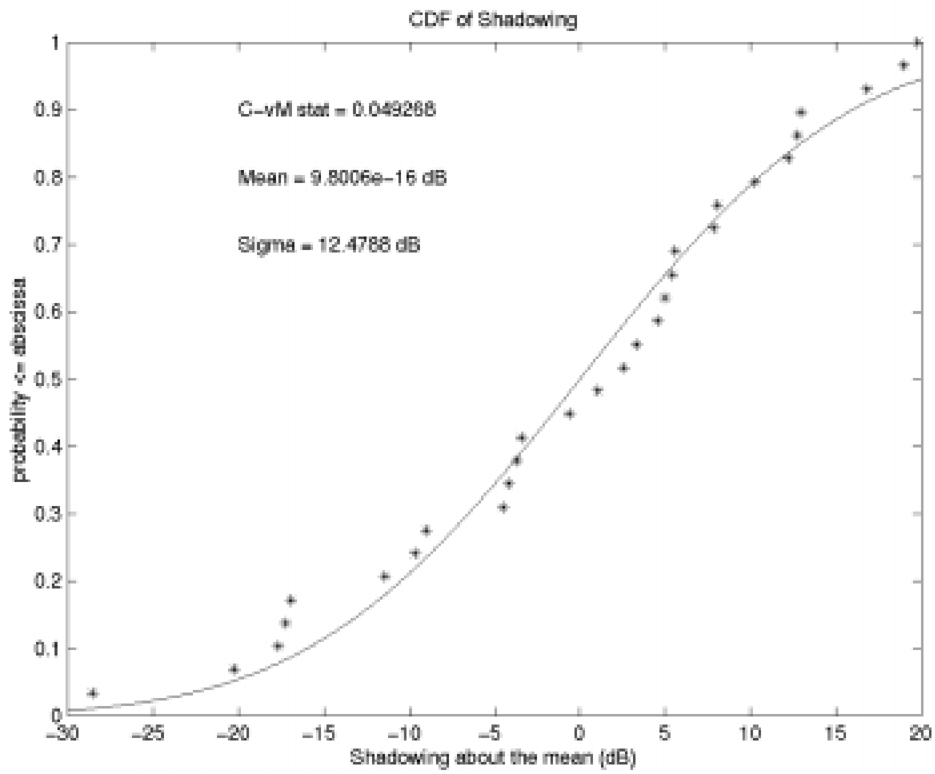

Figure 15: Cramer-von Mises goodness of fit test result for log-normal shadowing.

4.4 Two-ray model

Figures 16-17 show both the power law and two-ray models with the collected data for the LOS case. Since the measurements were taken in a suburban environment, reasonable values for the characteristics of the ground were used. The relative permitivity of the ground was assumed to be 10 and the conductance of the ground was assumed to be 0.005 S/m. What is important to note from these results is the two-ray model fits the collected data for the LOS locations better than the standard power law model for an average receive antenna height of 2 meters as seen in Figure 16. While interesting, the problem is that the results for the two-ray model are sensitive to the antenna heights. Figure 17 is the same comparison with a receive antenna height of 1 meter. It can be seen that the two-ray model does not fit the data as well as the power law model in this case. Since the two-ray model result is so sensitive to antenna heights, it is recommended that the power law model be used when an area is being characterized. If the heights of the antennas are known precisely and the link is point-to-point, the two-ray model appears to be a good model for LOS locations. Figures 18 and 19 show the two-ray model and power law comparison for the OBS and ALL cases. The results for these two cases suggest that the power law model is the better model for the path loss.

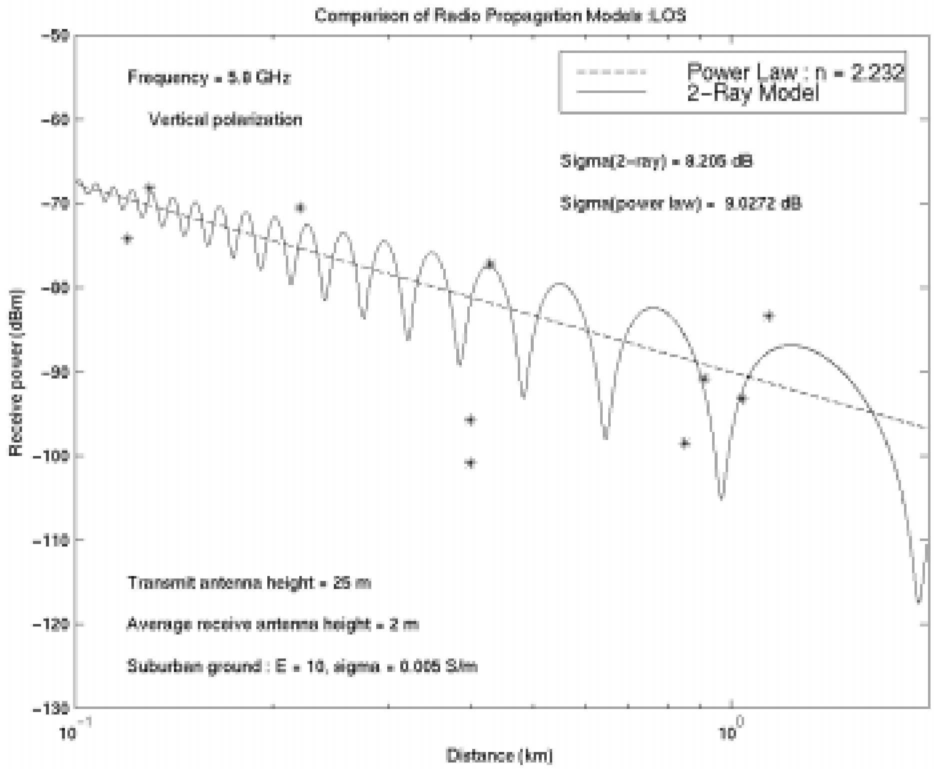

Figure 16: Two-ray and power law radio propagation model comparison for LOS locations for 2m receive antenna height.

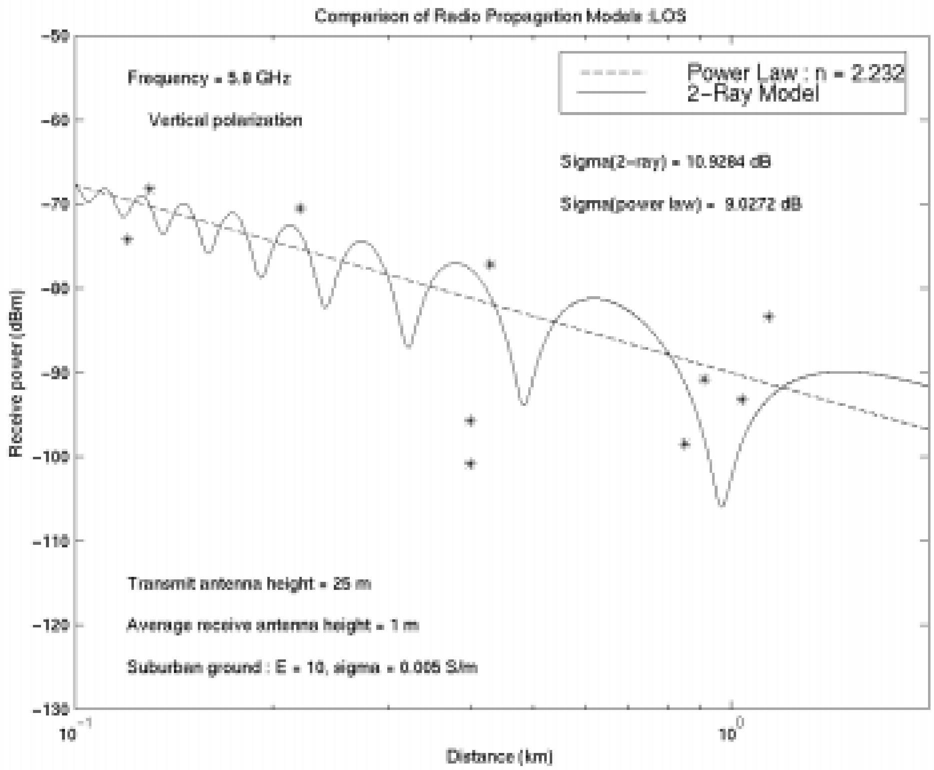

Figure 17: Two-ray and power law radio propagation model comparison for LOS locations for 1m receive antenna height.

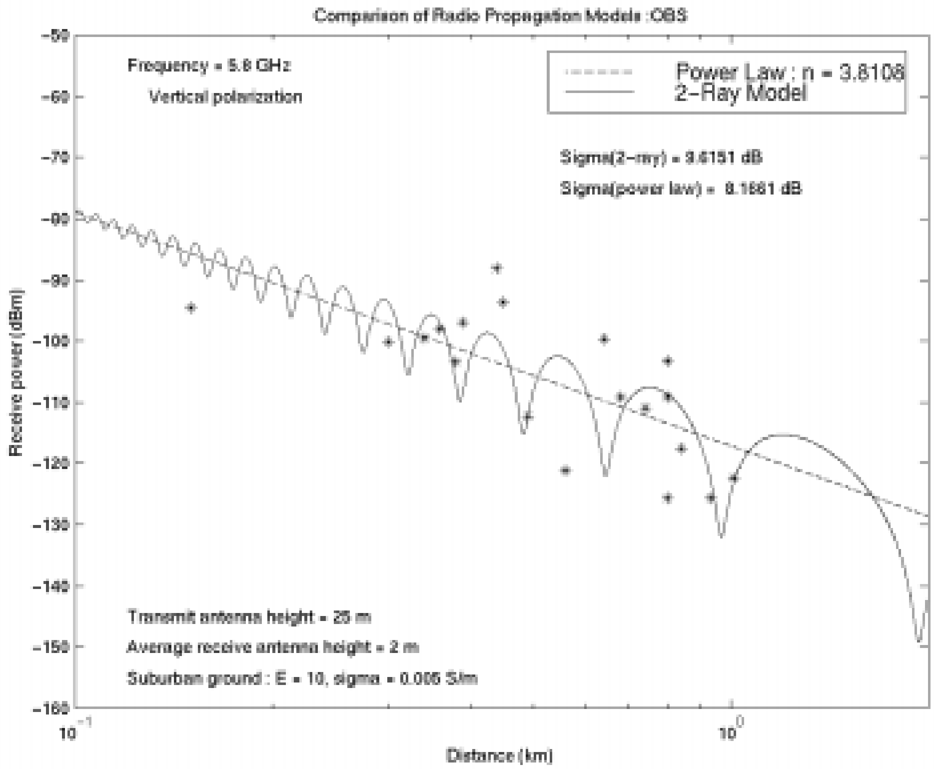

Figure 18: Two-ray and power law radio propagation model comparison for OBS locations.

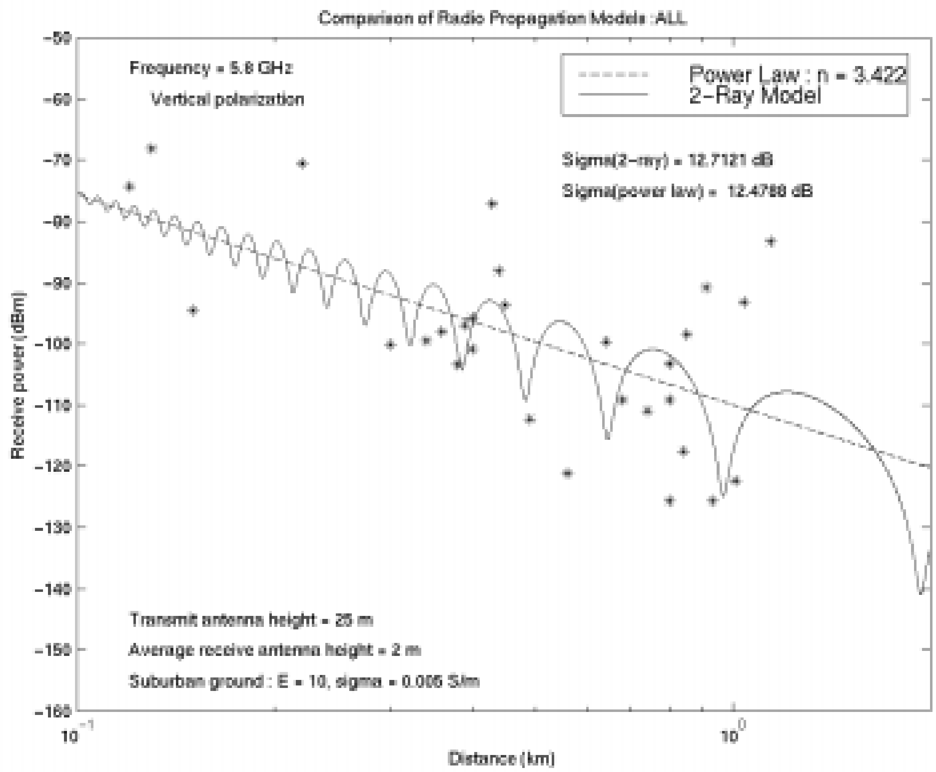

Figure 19: Two-ray and power law radio propagation model comparison for ALL locations.

4.5 Correlation between K and Shadowing

Figures 20-22 are the scatter diagrams for K and shadowing for each case. These also show the correlation coefficient. The correlation coefficient found for the LOS case, -0.88327 implies that there is a strong correlation between the two random variables. The correlation coefficient for the OBS case, -0.6622, is less than that for the LOS case, but it is still a moderate correlation. These results will be used during the determination of the percent coverage area. These results are expected due to the reliance of both K and the received power's dependence on the same two quantities. The received power is the sum of the power in the specular and random component while K is the ratio of the power in the specular component to the power in the random component.

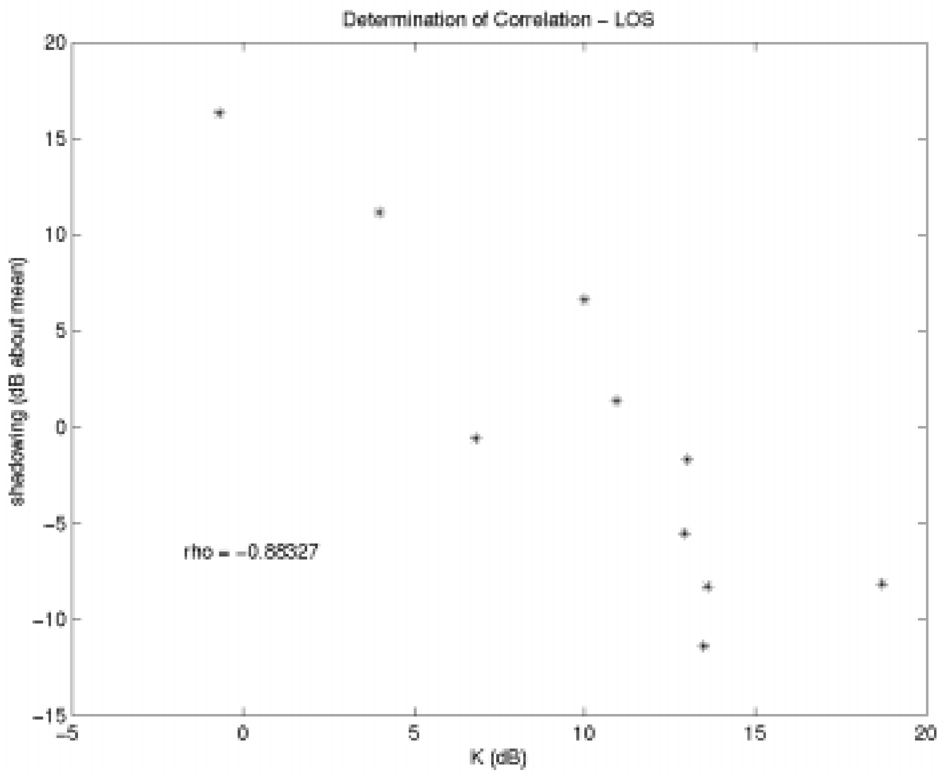

Figure 20: Determination of the correlation coefficient between K and shadowing for LOS locations.

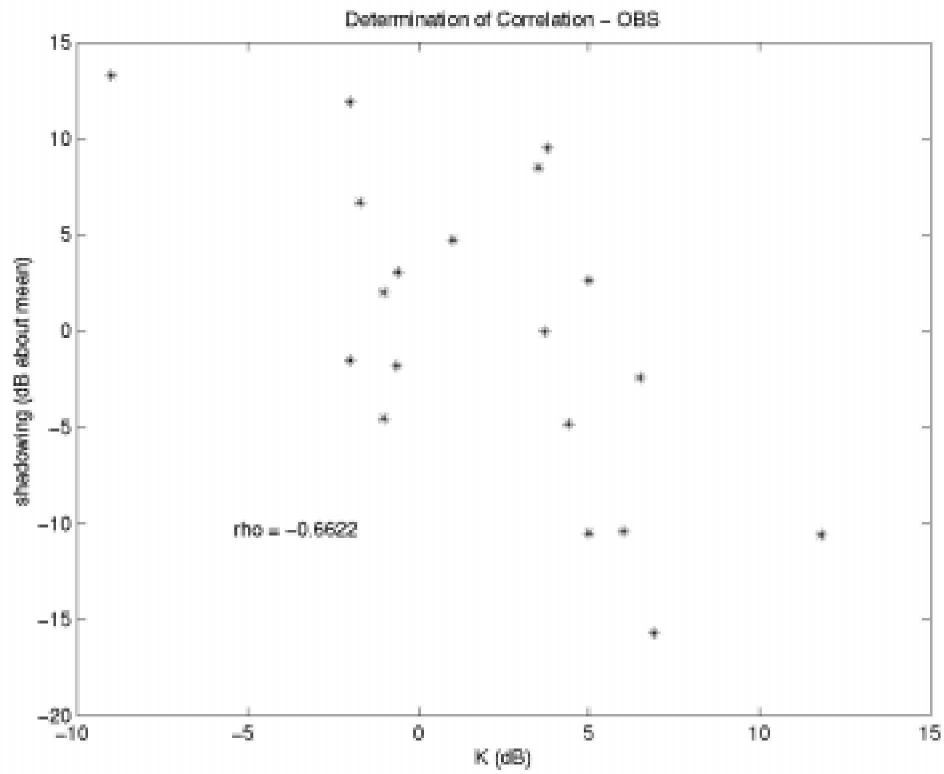

Figure 21: Determination of the correlation coefficient between K and shadowing for OBS locations.

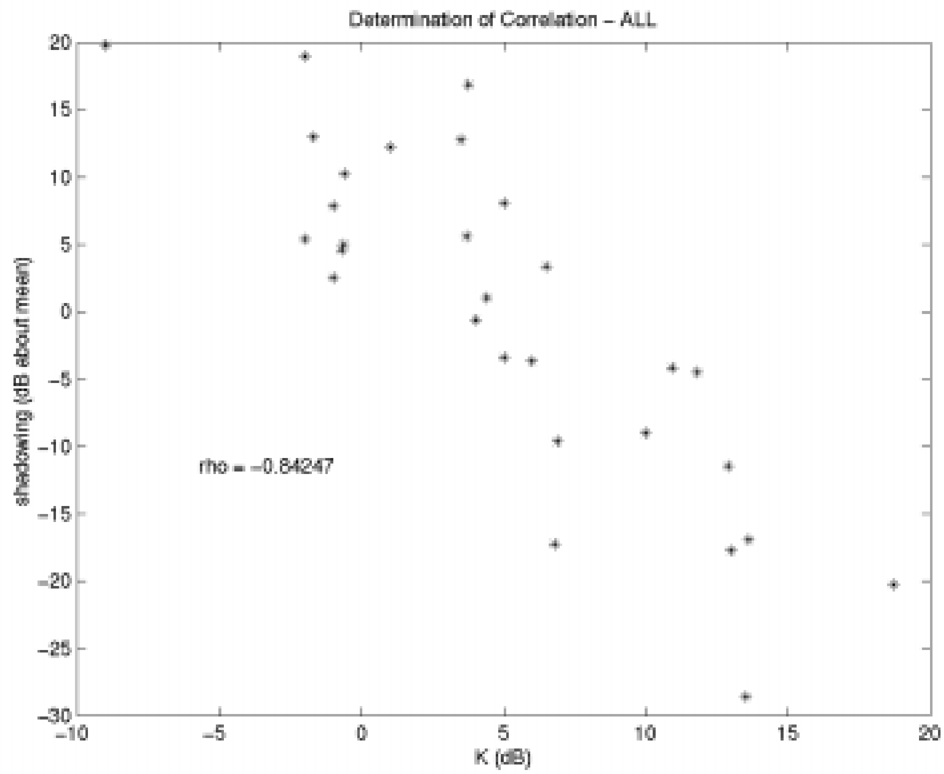

Figure 22: Determination of the correlation coefficient between K and shadowing for ALL locations.

4.6 Closing the Link Example

Figure 23 is the result obtained using the system parameters stated in the figure and determining the required ERP using both the power law model and the two-ray model. The link margin is obtained by subtracting the required ERP from the available ERP. For example, using the power law model, the link margin at 970 m is 10 dB, but using the two-ray model, the link margin is –50 dB because of the large null that occurs at this distance. Since the two-ray model was shown a good theoretical model for LOS locations, the transmitter-receiver separation must be taken into account when determining the amount of ERP necessary to close the link for a specific location.

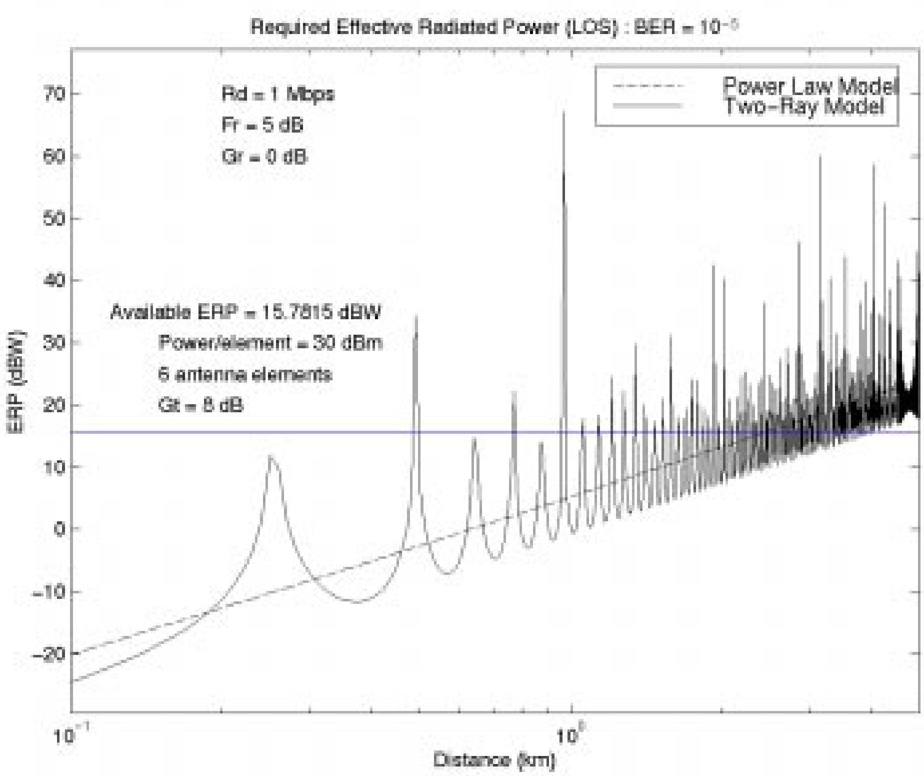

Figure 23: Required effective radiated power for LOS locations using the two-ray model and the power law model.

This page left blank intentionally.

4.7 Percent Coverage Area: K and Receive Power Joint Distribution Comparison

4.7.1 Distance Trade-Off

Figures 24-38 compare the percent coverage area between different assumptions about the relationship between K and the receive power (correlated, independent, Rayleigh channel) for different values of ERP (6, 10, 20, 30, 40 dBW) and different collections of experimental data (LOS, OBS and ALL) for a BER = 10^{-5}. Figure 39-53 are for an equivalent PER = 0.00424.

It is important to note that the correlated curves are the only ones generated using the experimental data that was collected. The independent curves assume the correlation between K and the received power does not exist and the Rayleigh curves assume that the channel between the transmitter and the receiver for every location over the entire area is Rayleigh. The results obtained show that the difference between using the correlated and independent distributions is at most around 5 percent. It could be argued then that since these two cases produce similar results, the independent case could be used as an approximation for the actual distribution if the independent distribution reduces the complexity of the problem. The comparison between the correlated and Rayleigh cases, however, shows a marked difference between the results especially when the ERP is small. Rayleigh channels are assumed because it is a worst case design approach. Assuming the entire area to be Rayleigh, though, could underestimate the percent coverage by as much as 60 percent. This is true whether the locations are LOS, OBS, ALL and whether the QoS statistic is BER or PER.

These graphs can also be read as a distance difference if, as is usually the case, a percent coverage area is specified. For example, if the requirement for percent coverage area is 50 percent and the ERP is 6 dBW, Figure 24 shows that the maximum cell boundary for the Rayleigh assumption is only 110 meters. However, if the actual data is used, the cell boundary is 1.7 km.

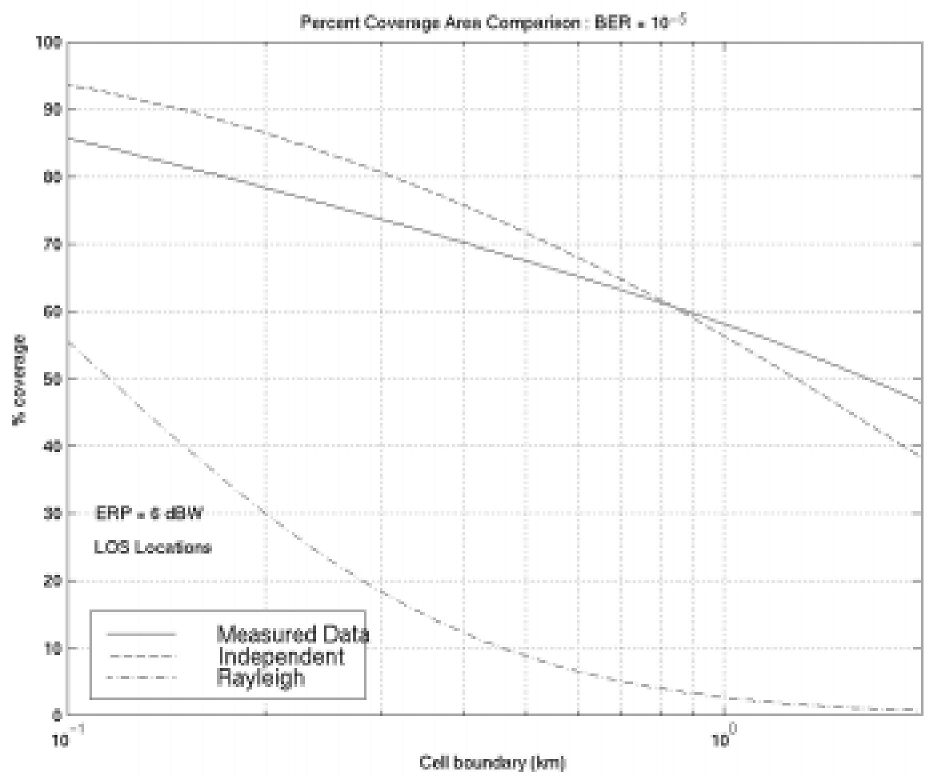

Figure 23: Percent coverage area comparison among the actual correlated K and receive power joint distribution, K and receive power independence and the Rayleigh channel assumption for LOS locations with BER = 10^{-5} and ERP = 6 dBW.

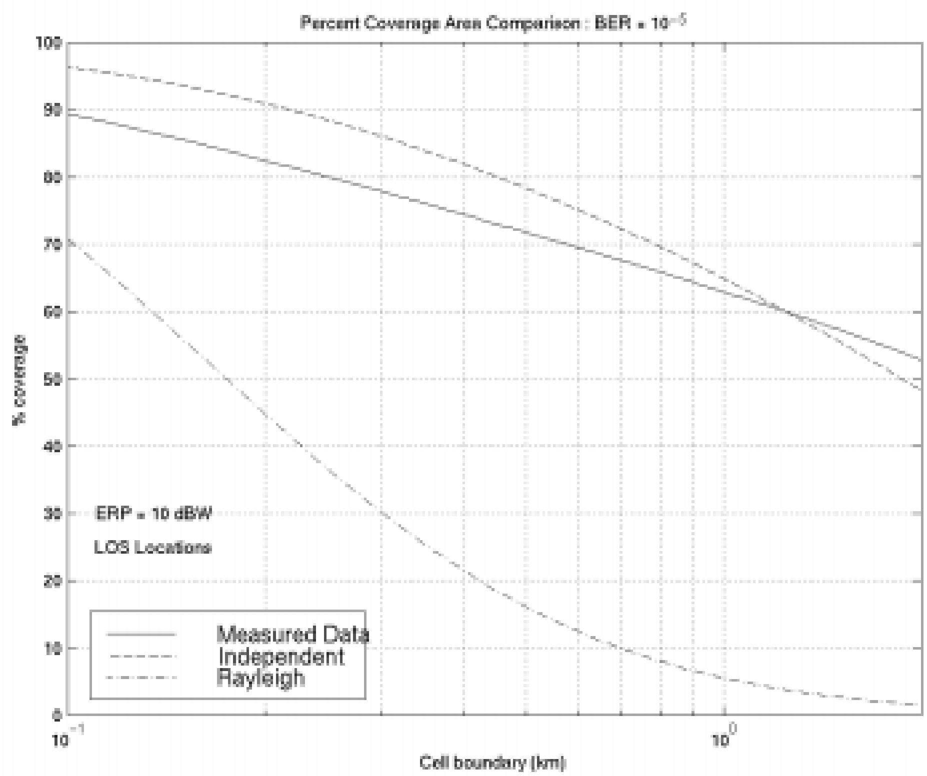

Figure 24: Percent coverage area comparison among the actual correlated K and receive power joint distribution, K and receive power independence and the Rayleigh channel assumption for LOS locations with BER = 10^{-5} and ERP = 10 dBW.

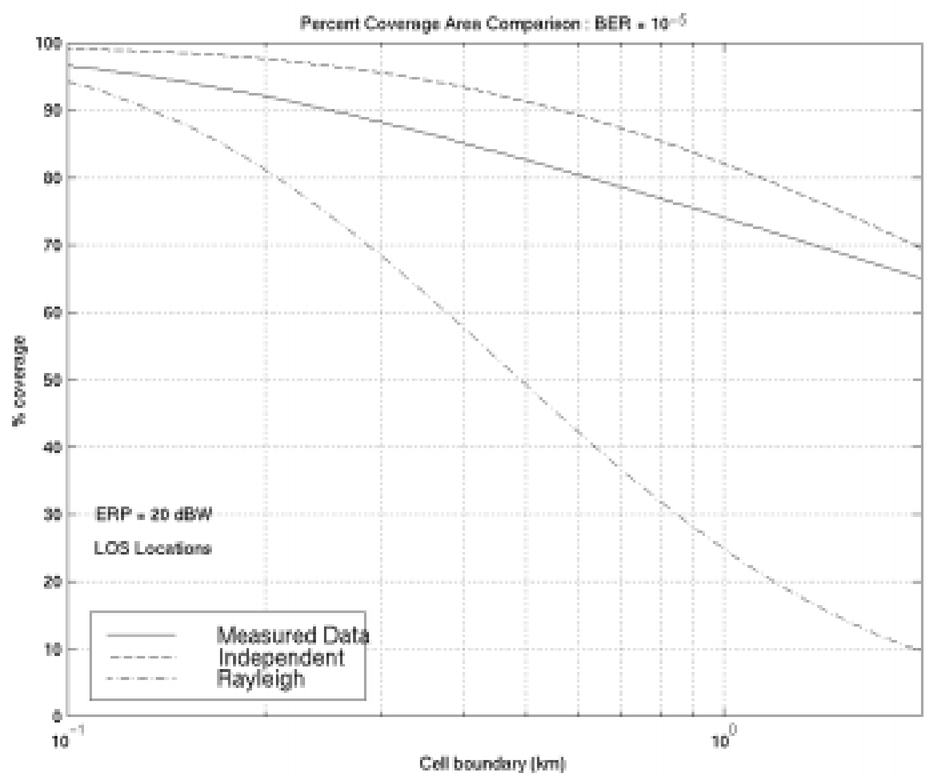

Figure 25: Percent coverage area comparison among the actual correlated K and receive power joint distribution, K and receive power independence and the Rayleigh channel assumption for LOS locations with BER = 10^{-5} and ERP = 20 dBW.

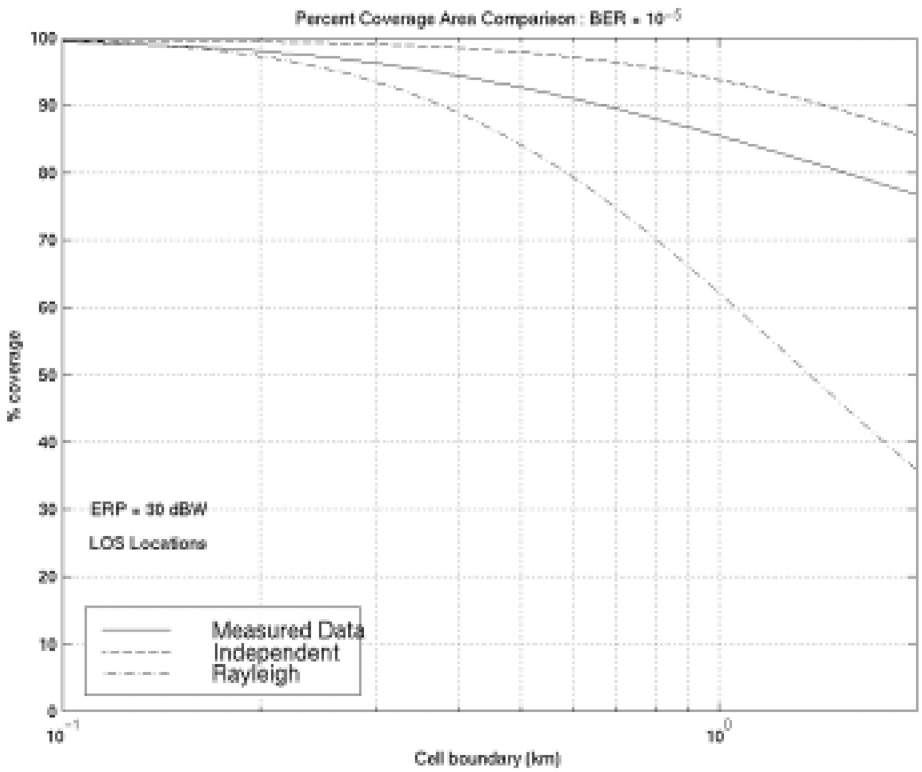

Figure 26: Percent coverage area comparison among the actual correlated K and receive power joint distribution, K and receive power independence and the Rayleigh channel assumption for LOS locations with BER = 10^{-5} and ERP = 30 dBW.

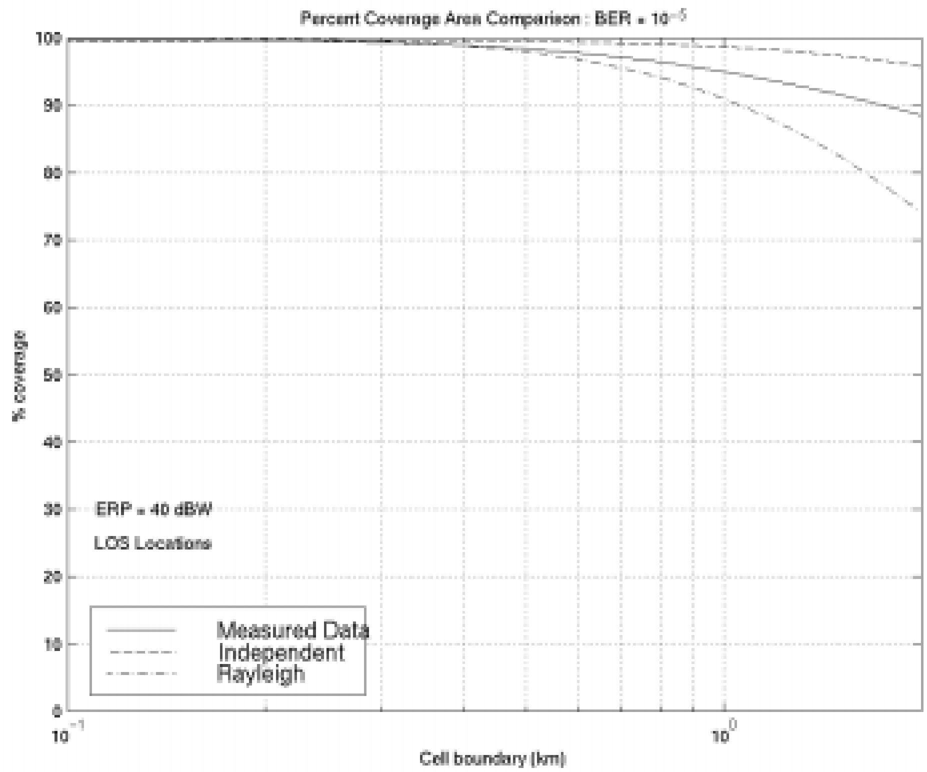

Figure 27: Percent coverage area comparison among the actual correlated K and receive power joint distribution, K and receive power independence and the Rayleigh channel assumption for LOS locations with BER = 10^{-5} and ERP = 40 dBW.

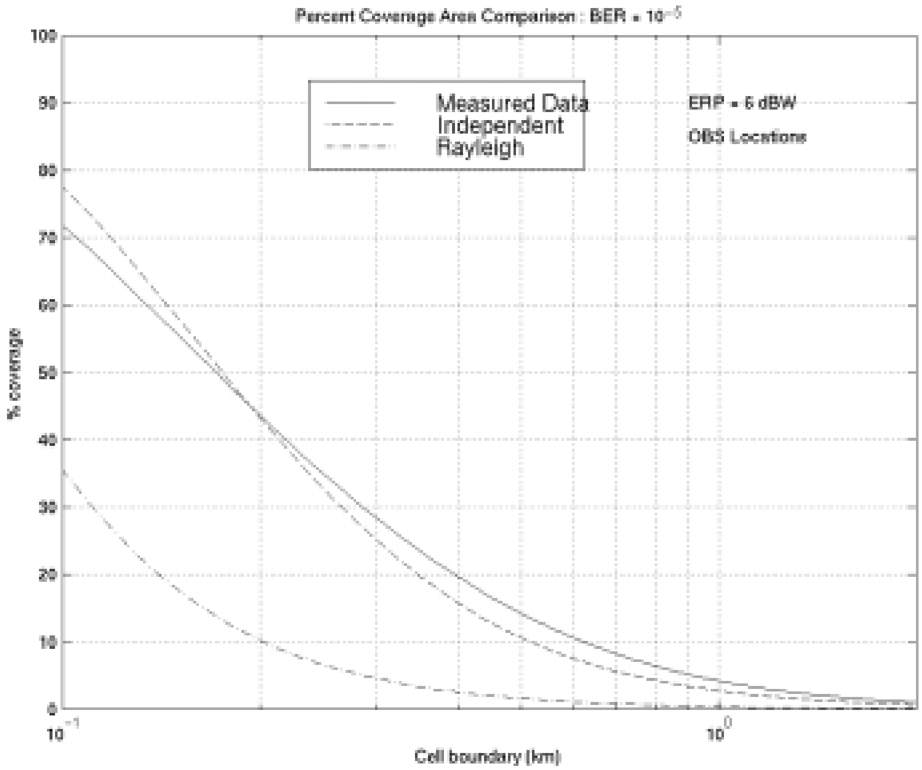

Figure 28: Percent coverage area comparison among the actual correlated K and receive power joint distribution, K and receive power independence and the Rayleigh channel assumption for OBS locations with BER = 10^{-5} and ERP = 6 dBW.

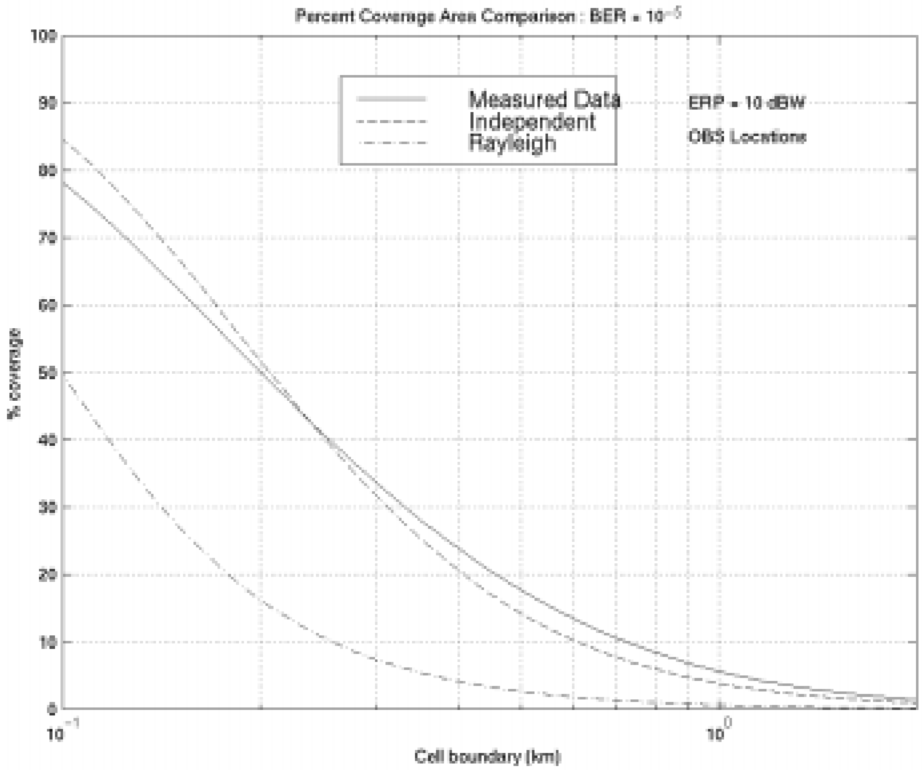

Figure 29: Percent coverage area comparison among the actual correlated K and receive power joint distribution, K and receive power independence and the Rayleigh channel assumption for OBS locations with BER = 10^{-5} and ERP = 10 dBW.

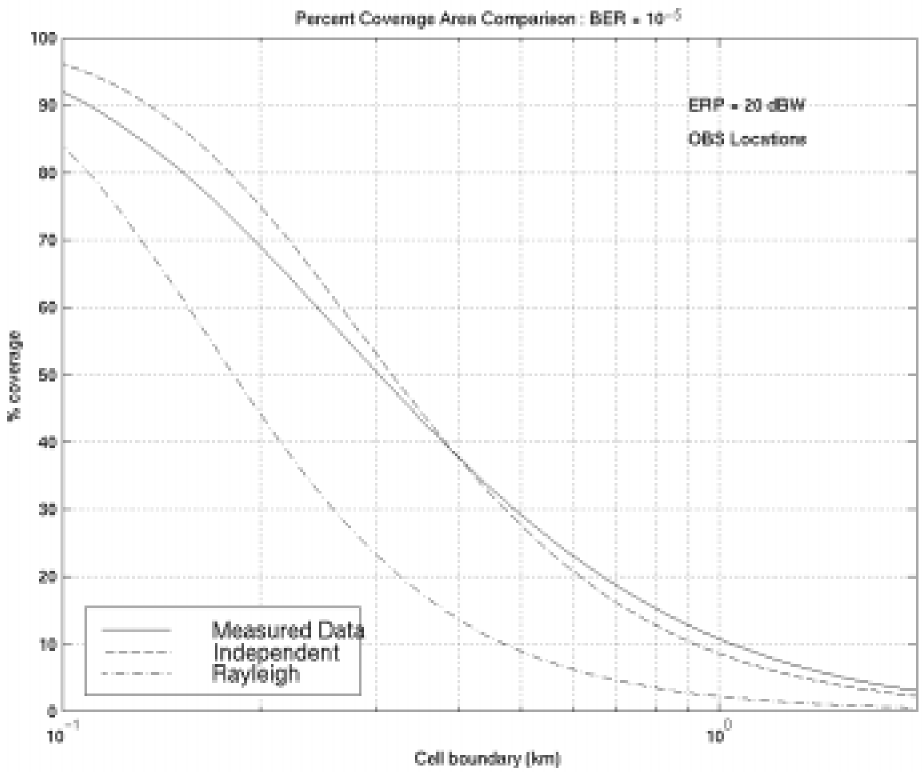

Figure 30: Percent coverage area comparison among the actual correlated K and receive power joint distribution, K and receive power independence and the Rayleigh channel assumption for OBS locations with BER = 10^{-5} and ERP = 20 dBW.

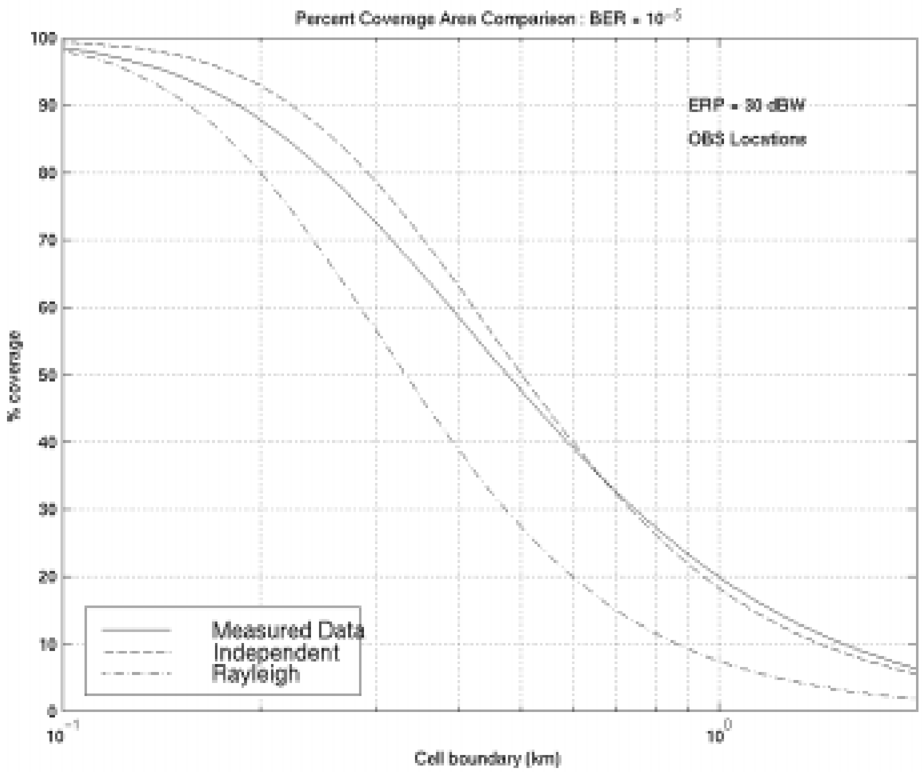

Figure 31: Percent coverage area comparison among the actual correlated K and receive power joint distribution, K and receive power independence and the Rayleigh channel assumption for OBS locations with BER = 10^{-5} and ERP = 30 dBW.

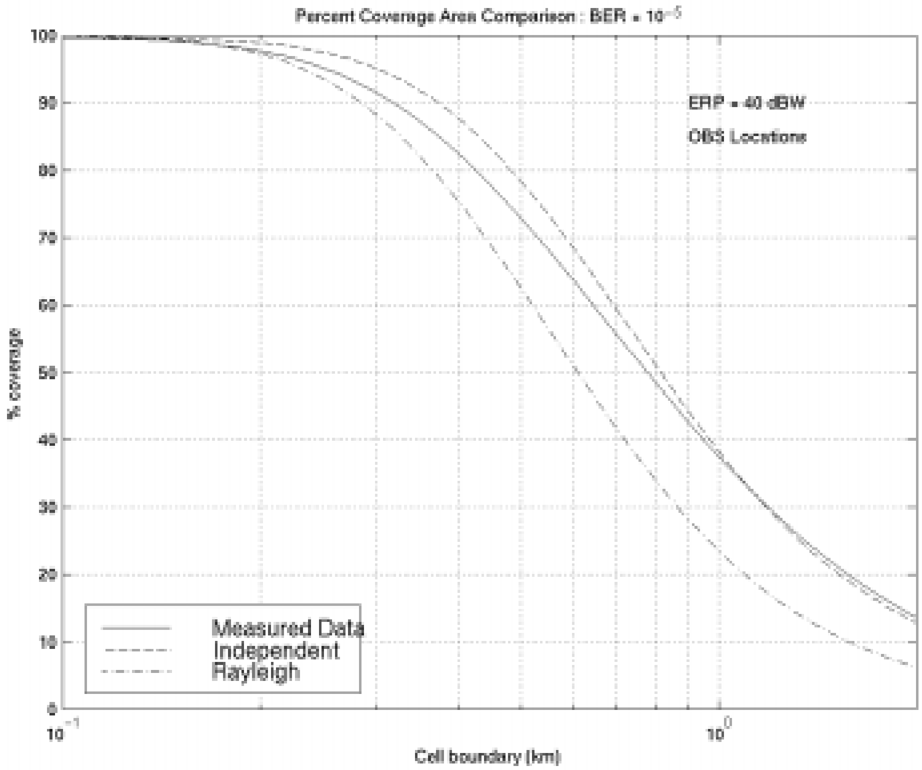

Figure 32: Percent coverage area comparison among the actual correlated K and receive power joint distribution, K and receive power independence and the Rayleigh channel assumption for OBS locations with BER = 10^{-5} and ERP = 40 dBW.

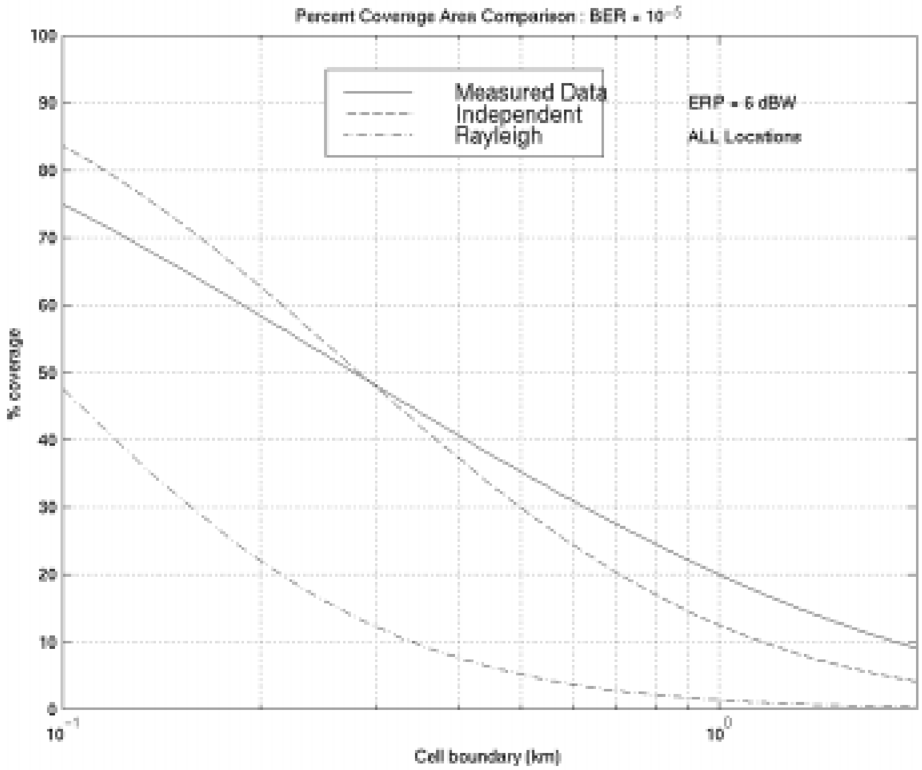

Figure 33: Percent coverage area comparison among the actual correlated K and receive power joint distribution, K and receive power independence and the Rayleigh channel assumption for ALL locations with BER = 10^{-5} and ERP = 6 dBW.

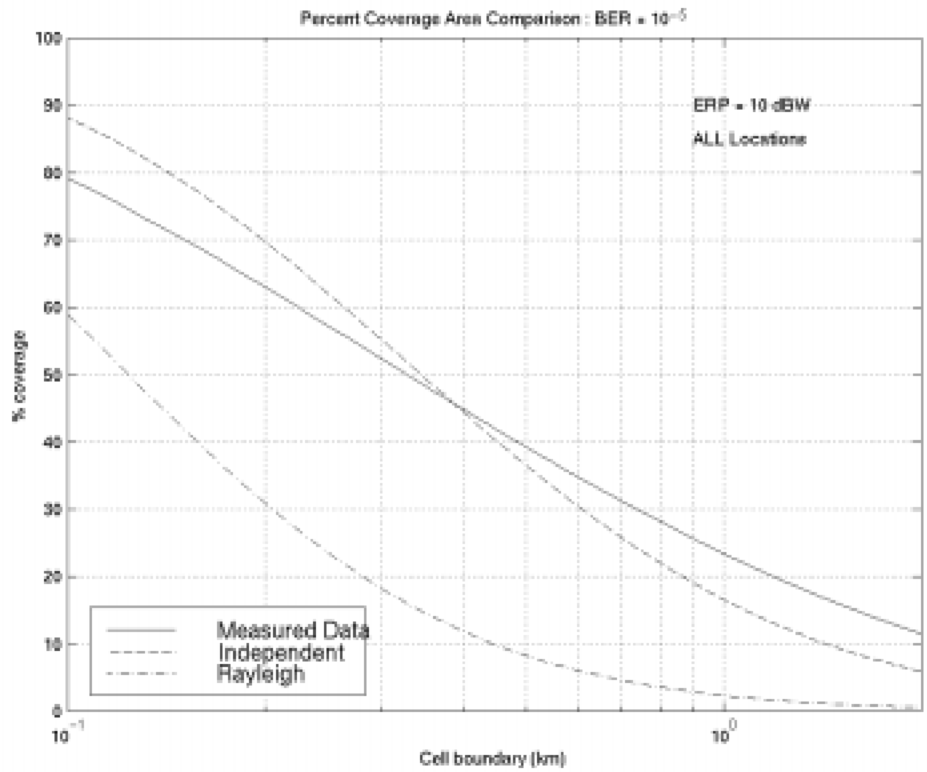

Figure 34: Percent coverage area comparison among the actual correlated K and receive power joint distribution, K and receive power independence and the Rayleigh channel assumption for ALL locations with BER = 10^{-5} and ERP = 10 dBW.

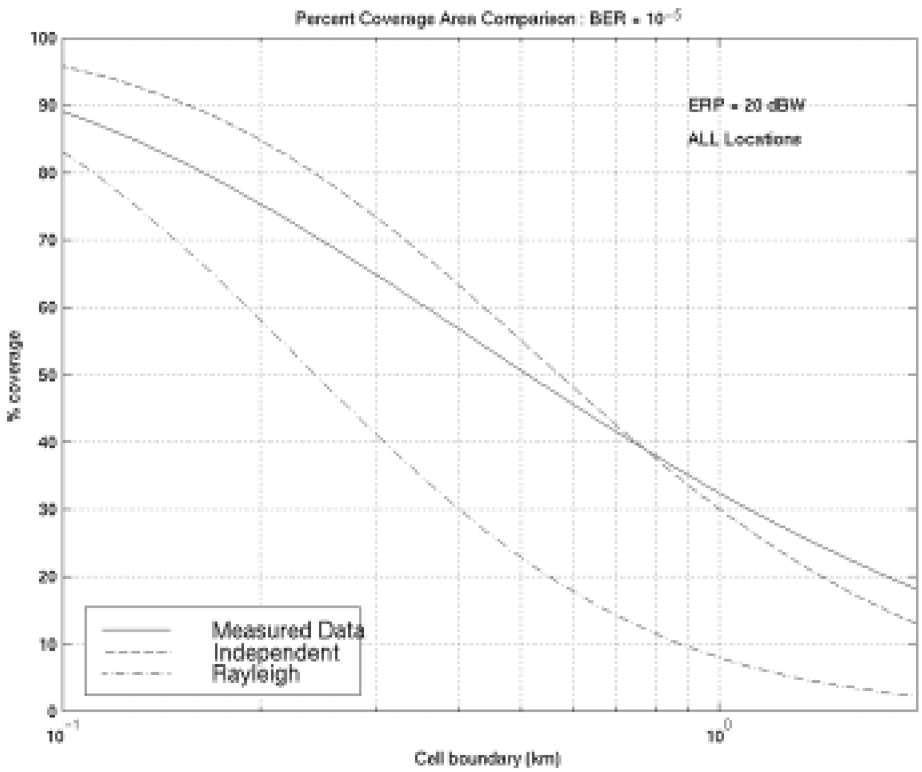

Figure 35: Percent coverage area comparison among the actual correlated K and receive power joint distribution, K and receive power independence and the Rayleigh channel assumption for ALL locations with BER = 10^{-5} and ERP = 20 dBW.

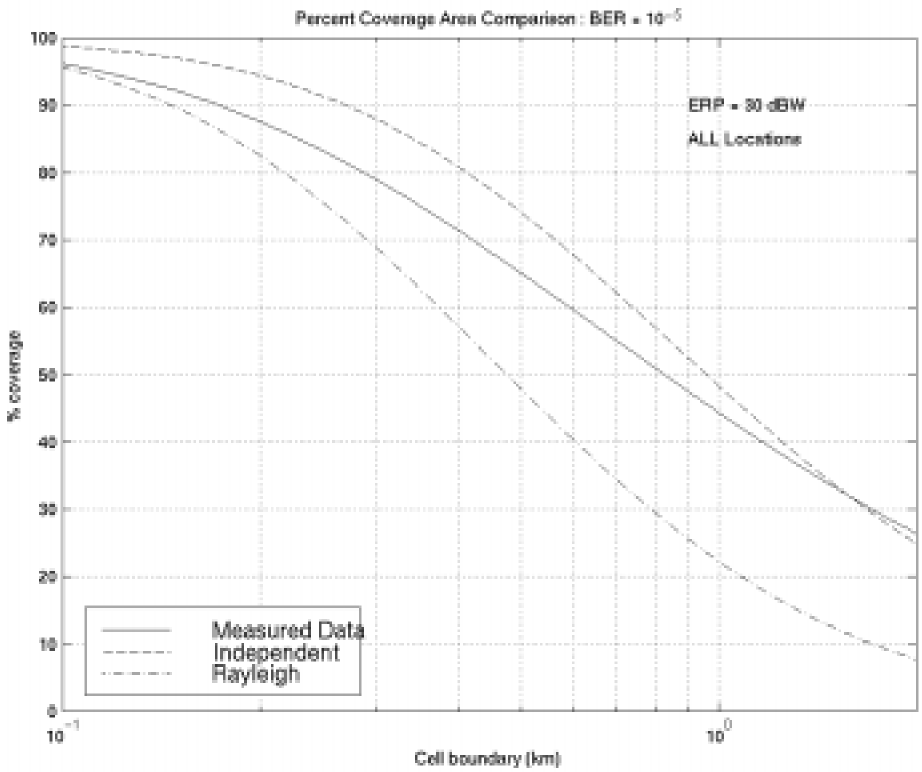

Figure 36: Percent coverage area comparison among the actual correlated K and receive power joint distribution, K and receive power independence and the Rayleigh channel assumption for ALL locations with BER = 10^{-5} and ERP = 30 dBW.

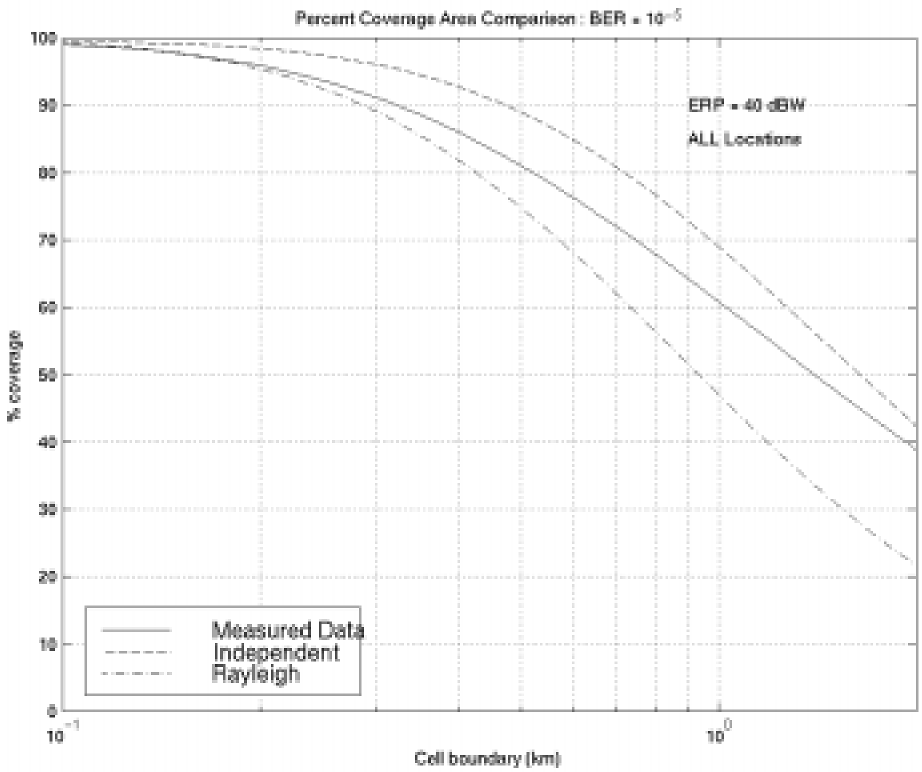

Figure 37: Percent coverage area comparison among the actual correlated K and receive power joint distribution, K and receive power independence and the Rayleigh channel assumption for ALL locations with BER = 10^{-5} and ERP = 40 dBW.

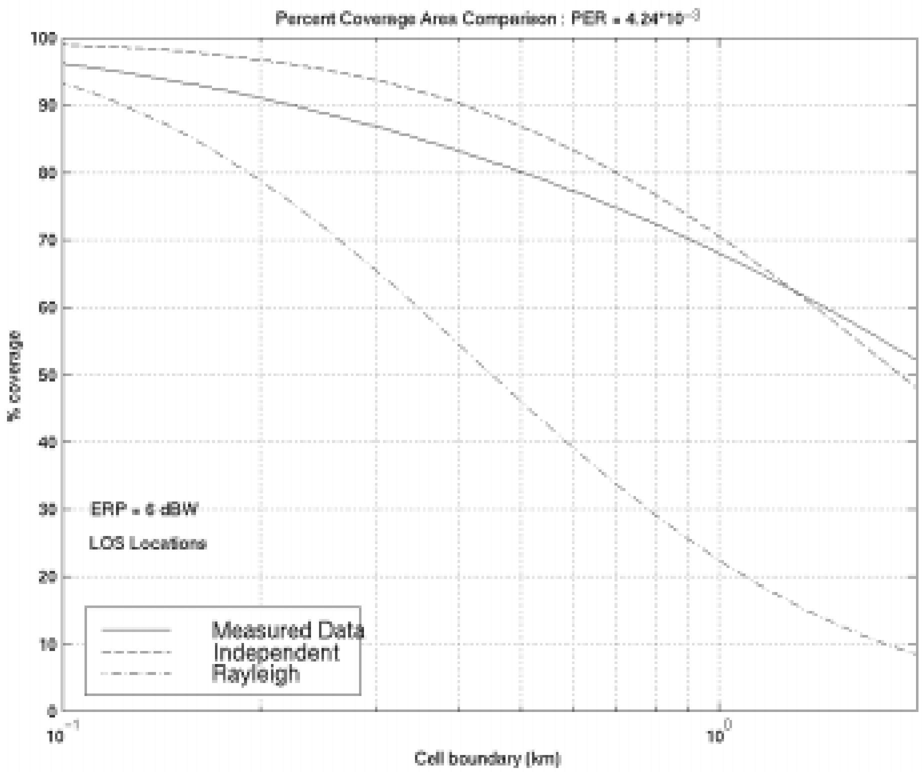

Figure 38: Percent coverage area comparison among the actual correlated K and receive power joint distribution, K and receive power independence and the Rayleigh channel assumption for LOS locations with PER = 0.00424 and ERP = 6 dBW.

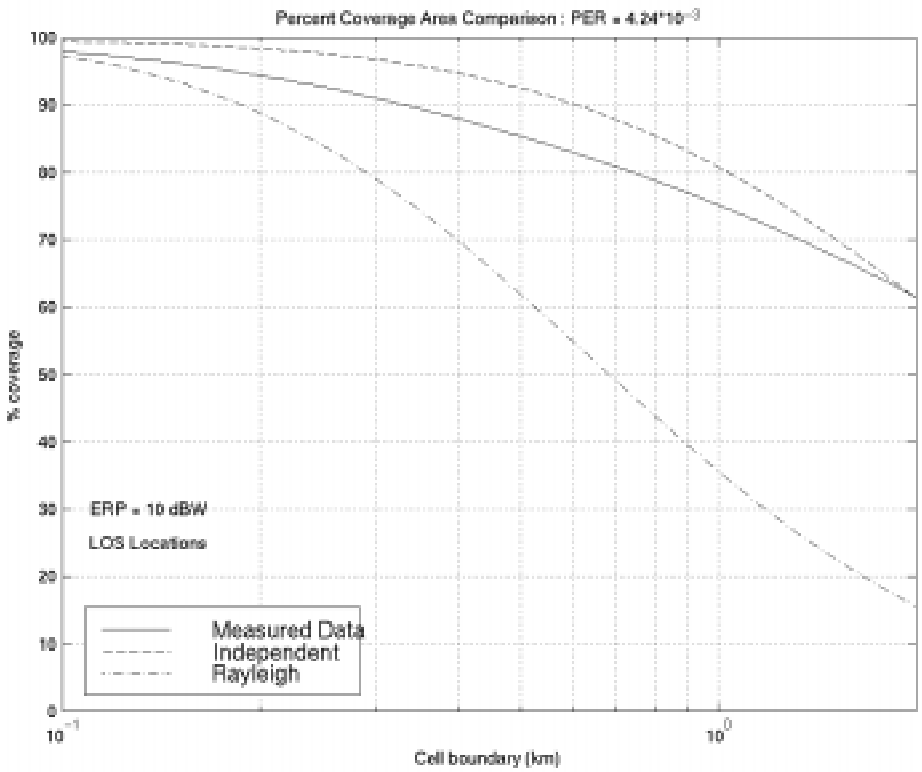

Figure 39: Percent coverage area comparison among the actual correlated K and receive power joint distribution, K and receive power independence and the Rayleigh channel assumption for LOS locations with PER = 0.00424 and ERP = 10 dBW.

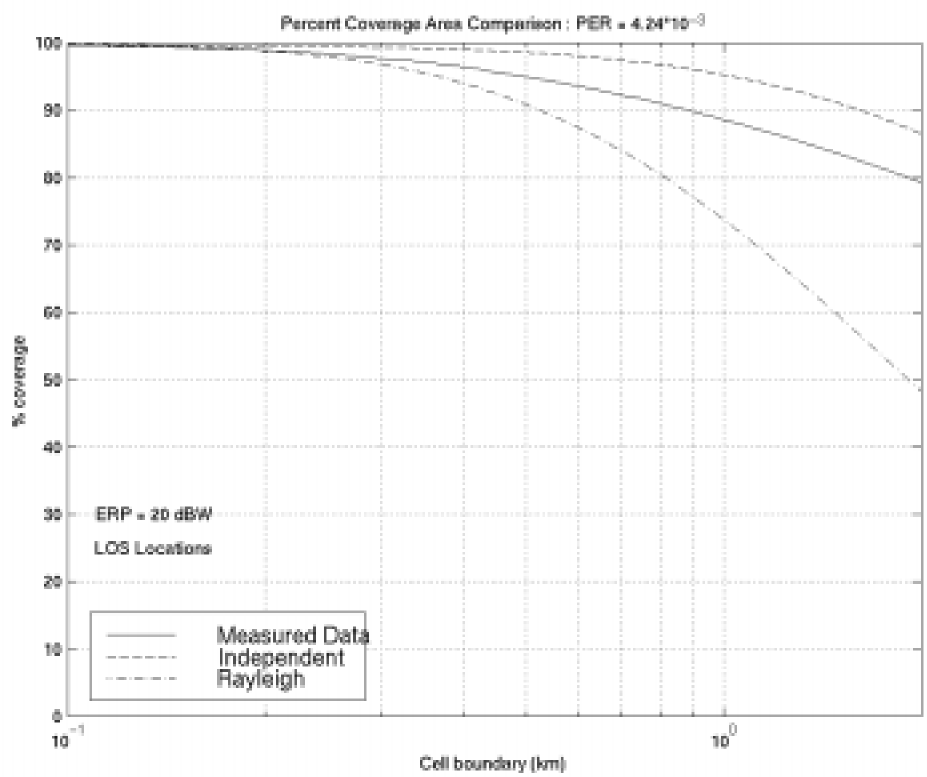

Figure 40: Percent coverage area comparison among the actual correlated K and receive power joint distribution, K and receive power independence and the Rayleigh channel assumption for LOS locations with PER = 0.00424 and ERP = 20 dBW.

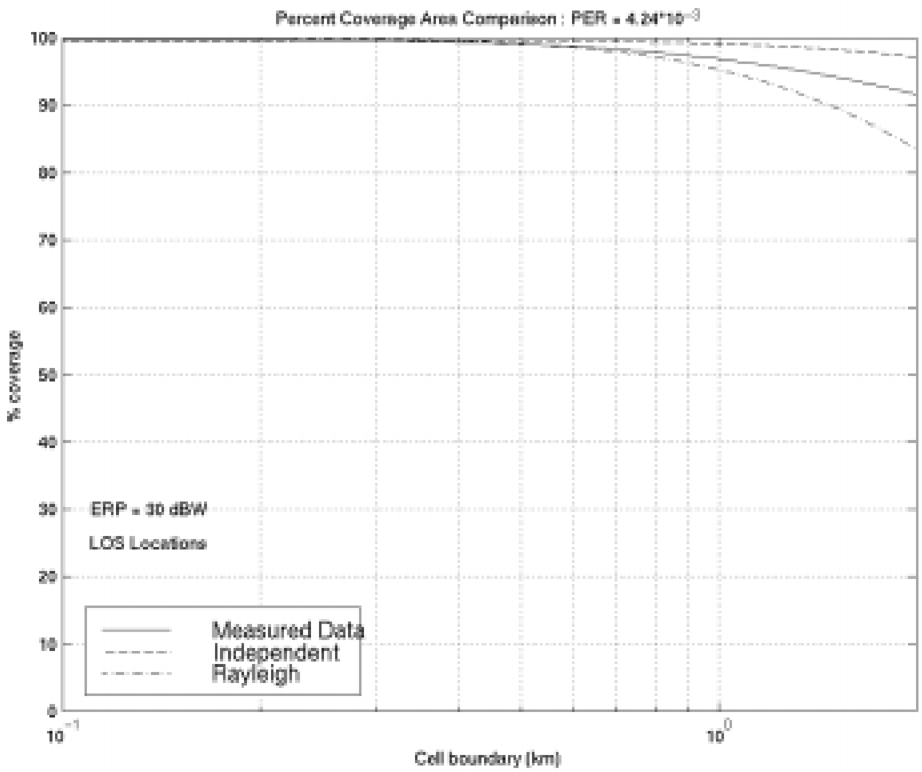

Figure 41: Percent coverage area comparison among the actual correlated K and receive power joint distribution, K and receive power independence and the Rayleigh channel assumption for LOS locations with PER = 0.00424 and ERP = 30 dBW.

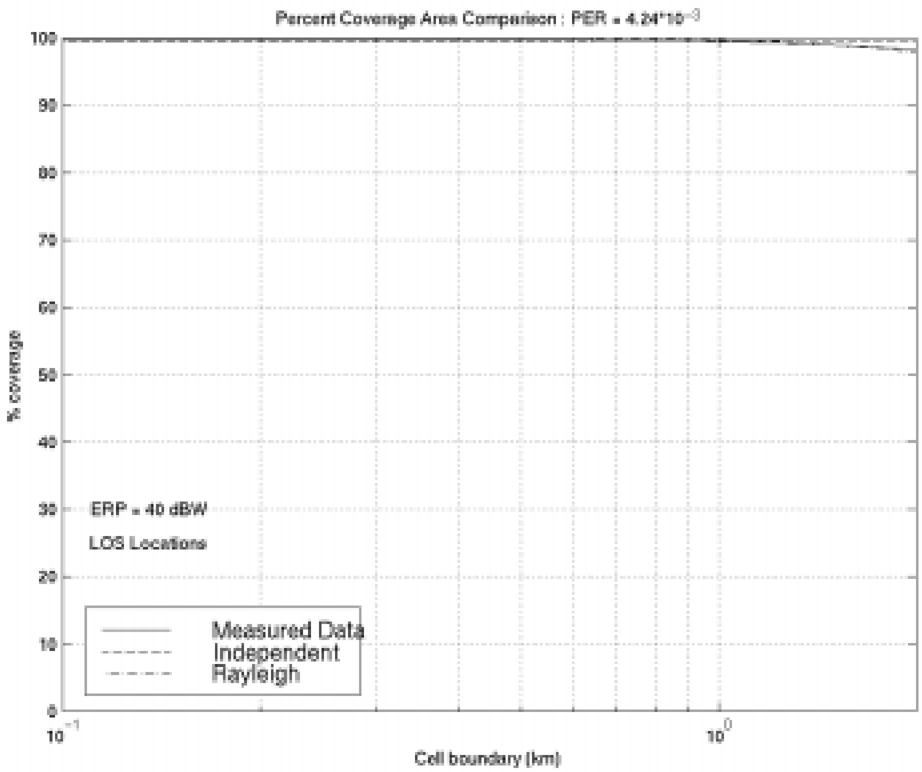

Figure 42: Percent coverage area comparison among the actual correlated K and receive power joint distribution, K and receive power independence and the Rayleigh channel assumption for LOS locations with PER = 0.00424 and ERP = 40 dBW.

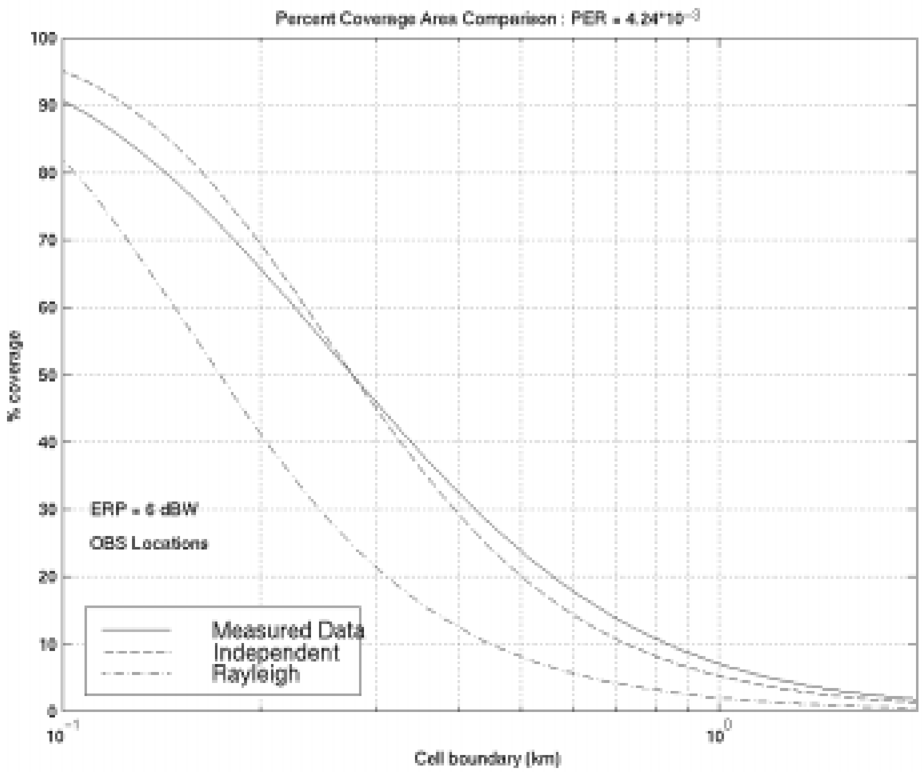

Figure 43: Percent coverage area comparison among the actual correlated K and receive power joint distribution, K and receive power independence and the Rayleigh channel assumption for OBS locations with PER = 0.00424 and ERP = 6 dBW.

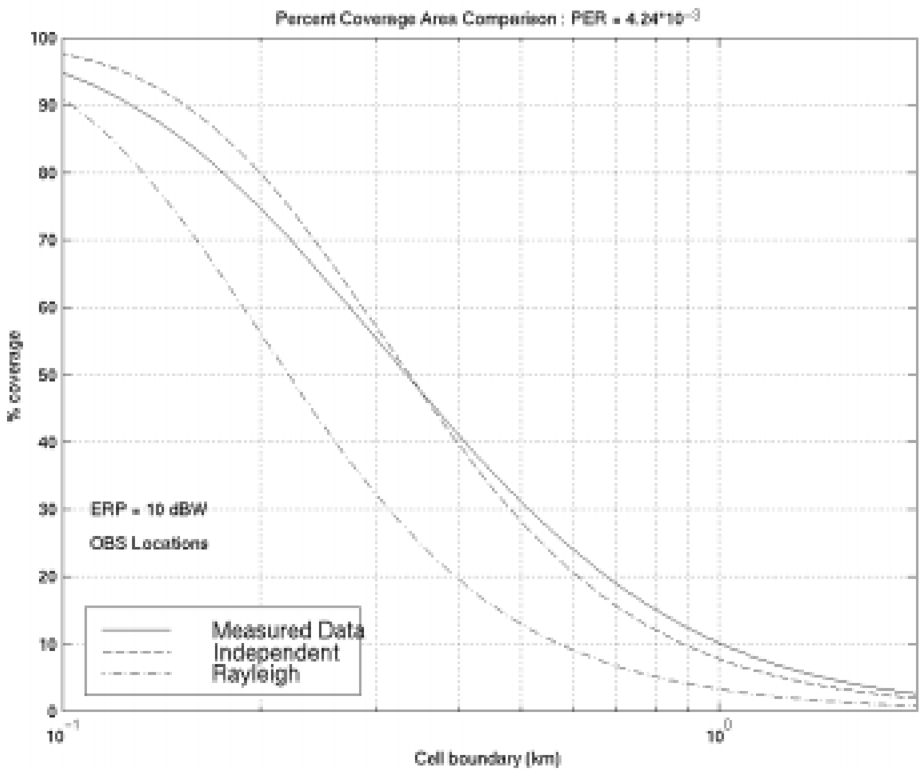

Figure 44: Percent coverage area comparison among the actual correlated K and receive power joint distribution, K and receive power independence and the Rayleigh channel assumption for OBS locations with PER = 0.00424 and ERP = 10 dBW.

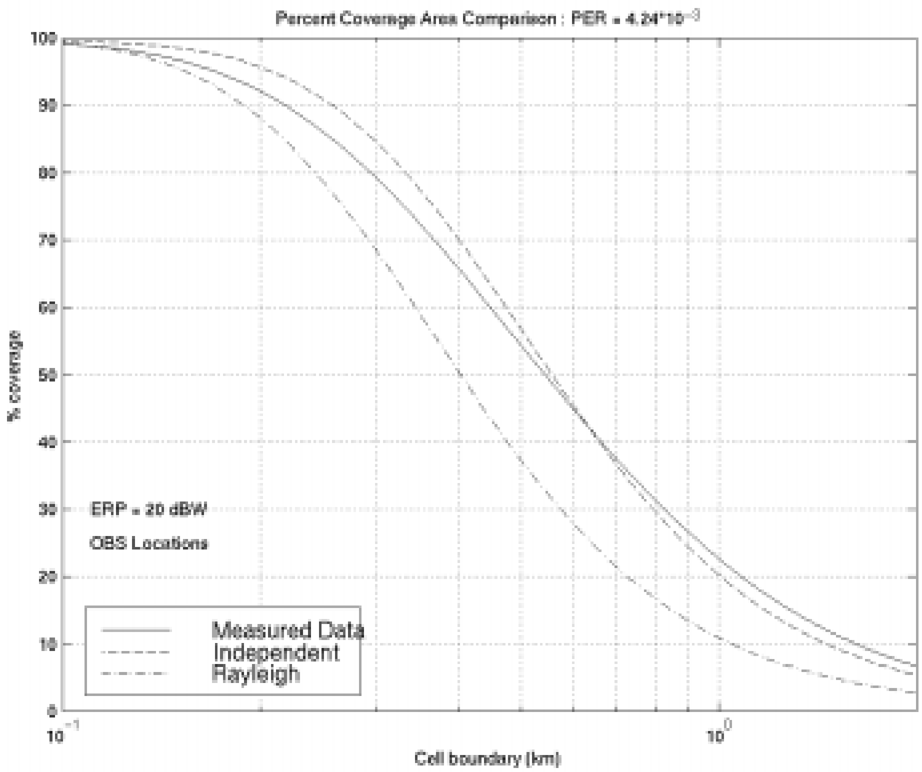

Figure 45: Percent coverage area comparison among the actual correlated K and receive power joint distribution, K and receive power independence and the Rayleigh channel assumption for OBS locations with PER = 0.00424 and ERP = 20 dBW.

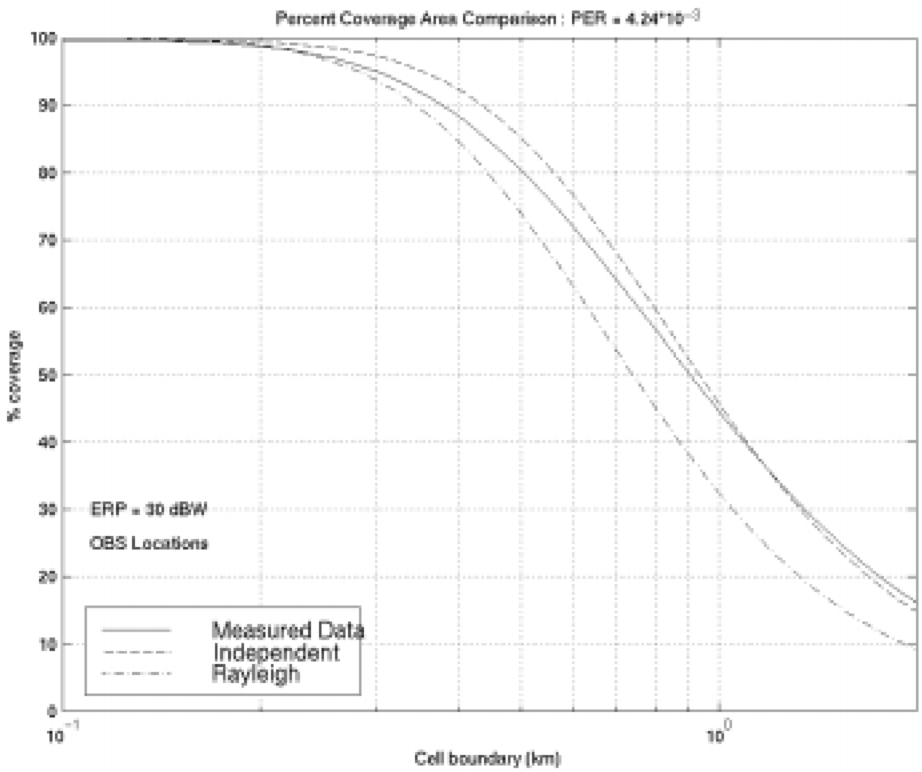

Figure 46: Percent coverage area comparison among the actual correlated K and receive power joint distribution, K and receive power independence and the Rayleigh channel assumption for OBS locations with PER = 0.00424 and ERP = 30 dBW.

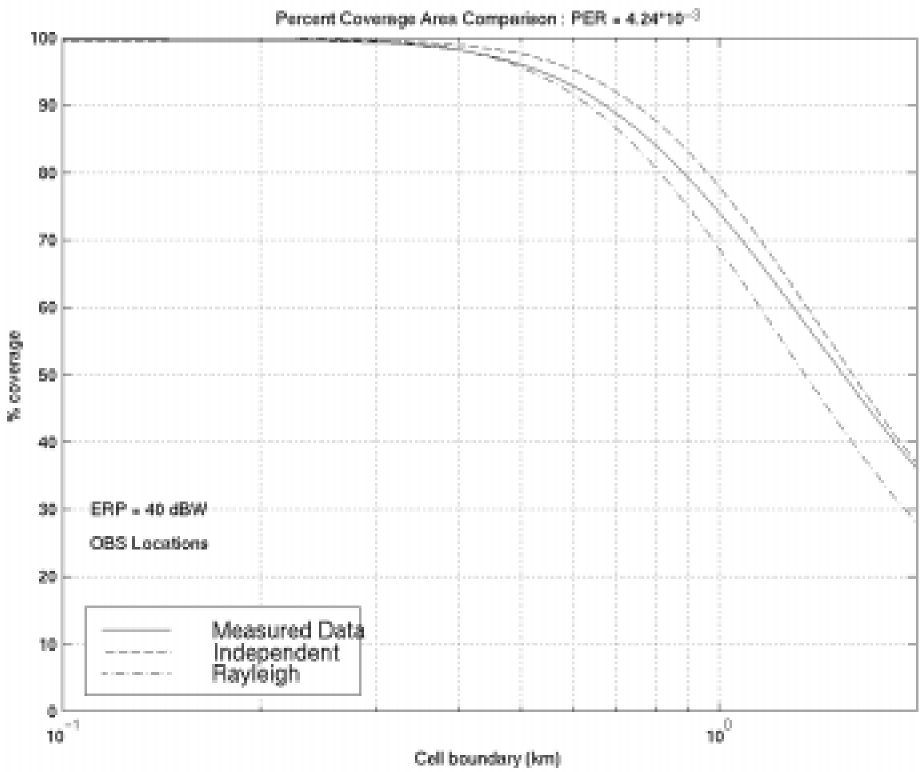

Figure 47: Percent coverage area comparison among the actual correlated K and receive power joint distribution, K and receive power independence and the Rayleigh channel assumption for OBS locations with PER = 0.00424 and ERP = 40 dBW.

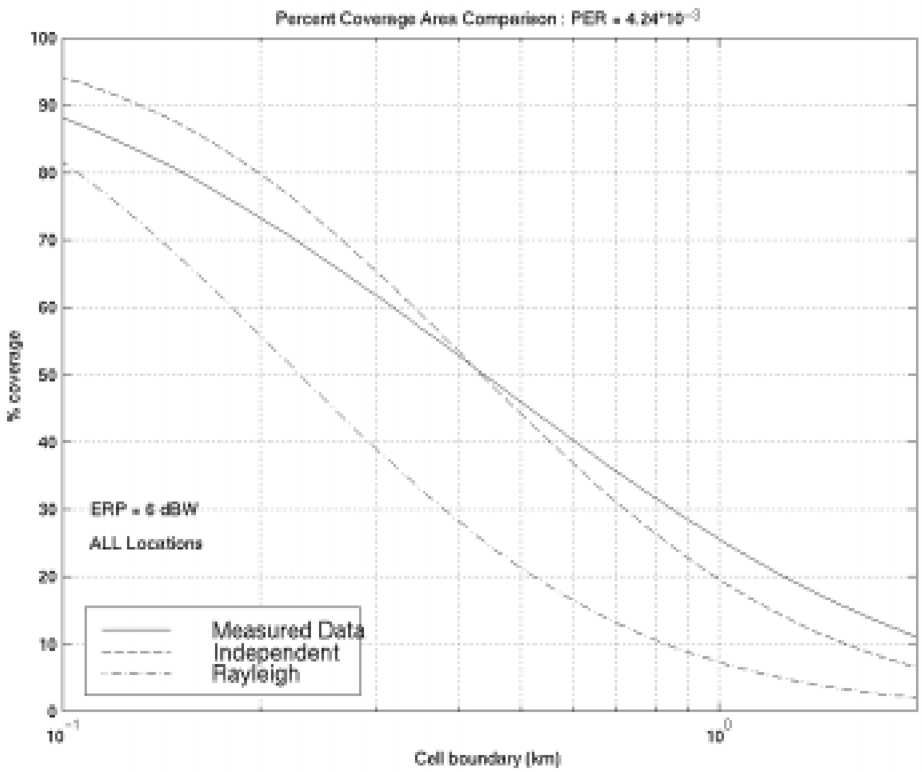

Figure 48: Percent coverage area comparison among the actual correlated K and receive power joint distribution, K and receive power independence and the Rayleigh channel assumption for ALL locations with PER = 0.00424 and ERP = 6 dBW.

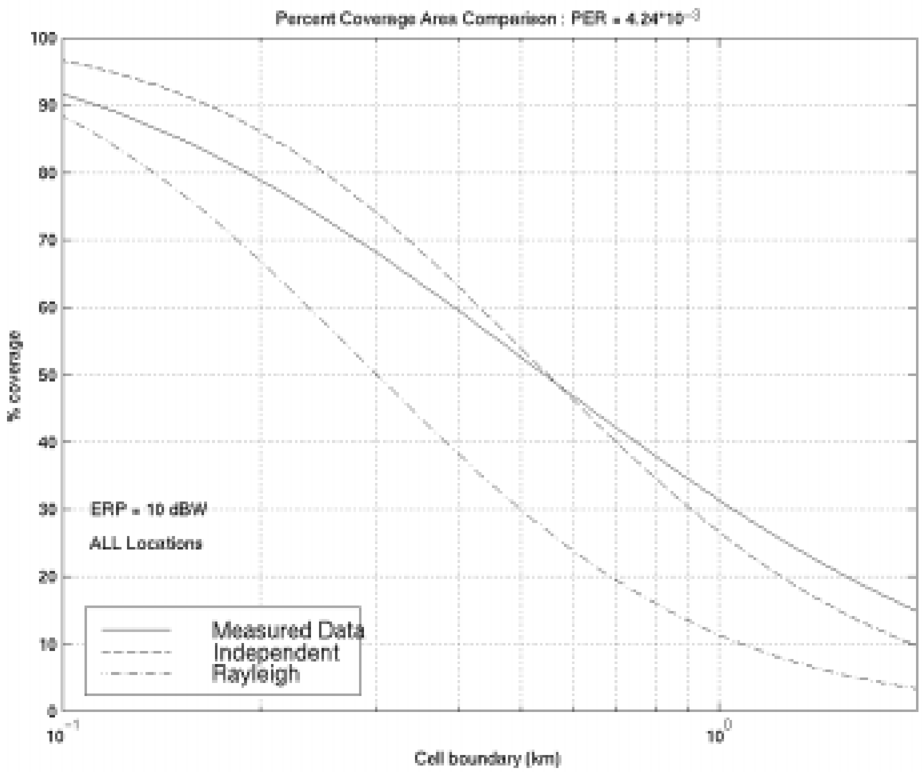

Figure 49: Percent coverage area comparison among the actual correlated K and receive power joint distribution, K and receive power independence and the Rayleigh channel assumption for ALL locations with PER = 0.00424 and ERP = 10 dBW.

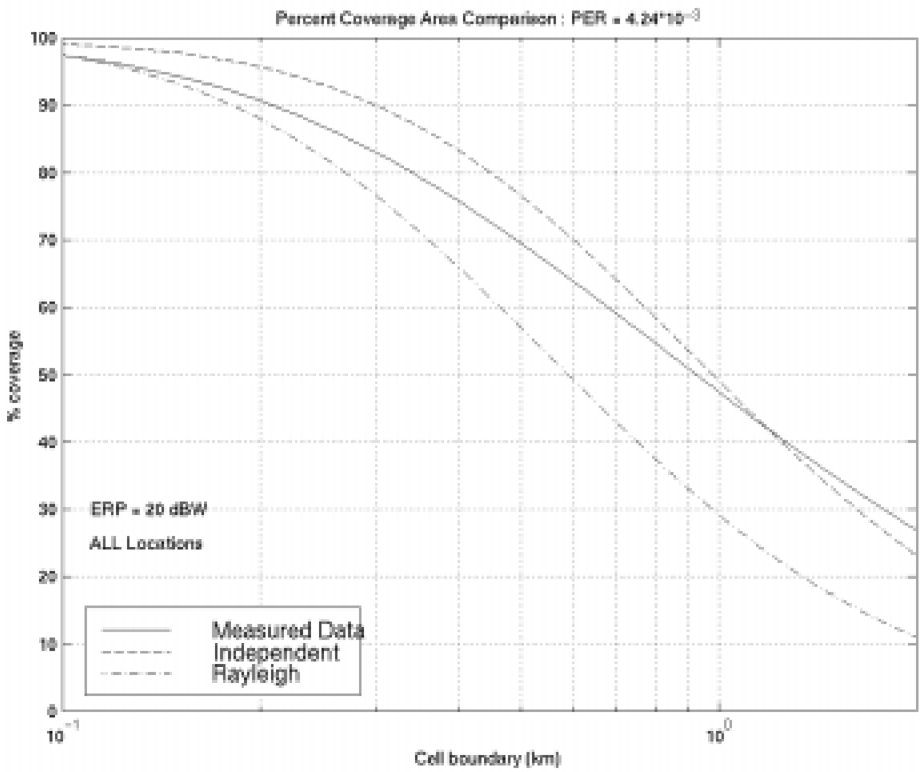

Figure 50: Percent coverage area comparison among the actual correlated K and receive power joint distribution, K and receive power independence and the Rayleigh channel assumption for ALL locations with PER = 0.00424 and ERP = 20 dBW.

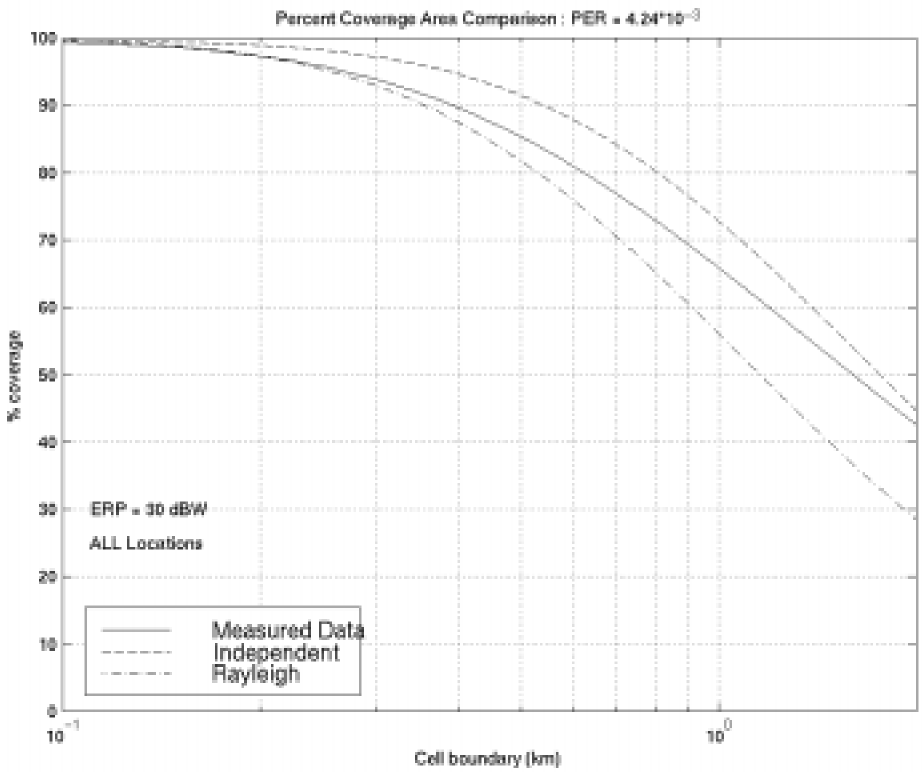

Figure 51: Percent coverage area comparison among the actual correlated K and receive power joint distribution, K and receive power independence and the Rayleigh channel assumption for ALL locations with PER = 0.00424 and ERP = 30 dBW.

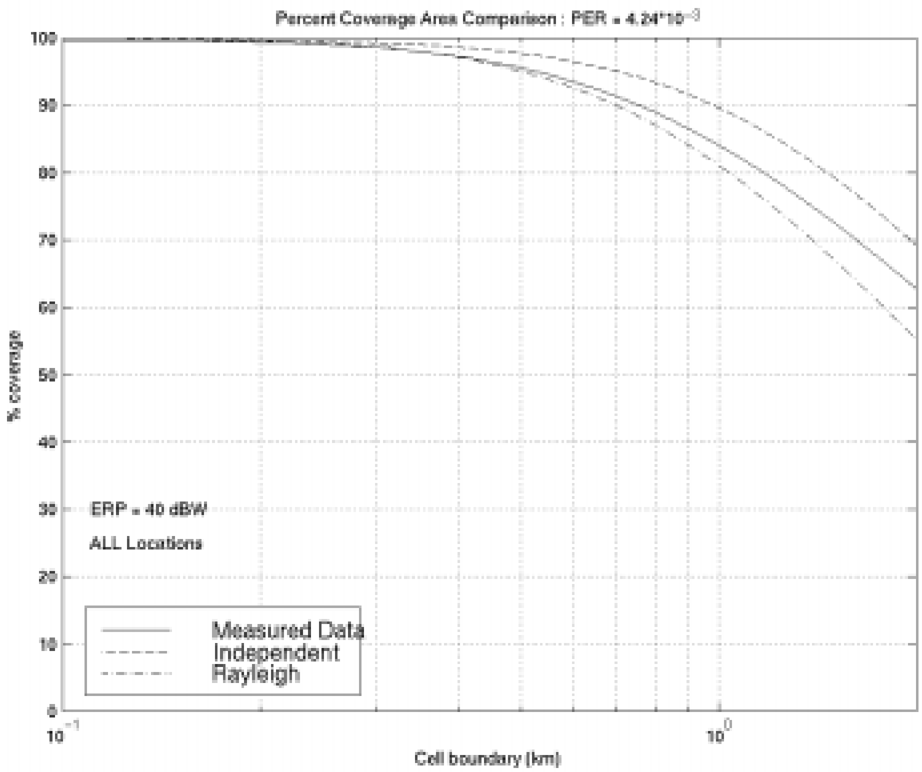

Figure 52: Percent coverage area comparison among the actual correlated K and receive power joint distribution, K and receive power independence and the Rayleigh channel assumption for ALL locations with PER = 0.00424 and ERP = 40 dBW.

4.7.2 Power Trade-Off

The results in the previous section were found by fixing the ERP for a specific set of locations and then determining the percent coverage area as a function of the cell boundary. In this way, the difference between the maximum cell boundaries for the actual correlated data and the Rayleigh channel assumption data can be found. The results in this section will be found by fixing the cell boundary and determining the percent coverage area as a function of the ERP. From these results, it will be possible to determine how much power can be saved if the percent coverage area and cell boundary are fixed. For example, if the percent coverage area is specified as 90 percent and the cell boundary is 100 m, Figure 54 shows that if the channel is assumed Rayleigh, 17 dBW is required. However, if the actual data is used, the requirement is only 11 dBW.

Figures 54-62 compare the actual data with the Rayleigh channel assumption for cell boundaries of 100 m, 500 m and 1 km assuming a BER = 10^{-5}. Figures 63-71 use the equivalent PER = 0.00424.

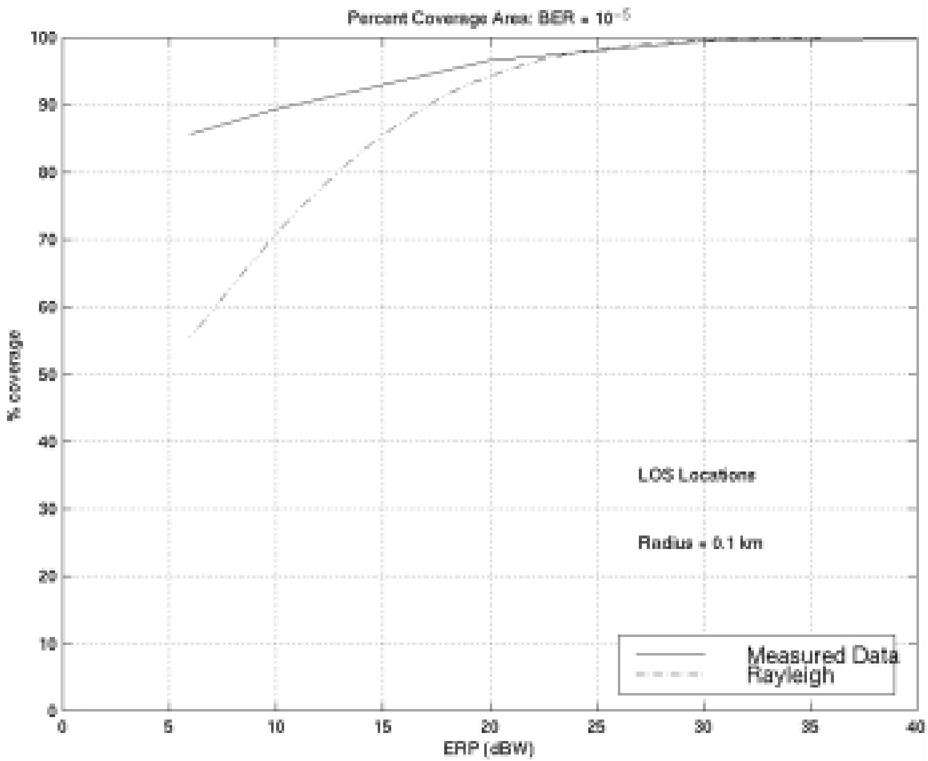

Figure 53: Percent coverage area for the actual correlated K and receive power distribution and the Rayleigh channel assumption for LOS locations with BER = 10^{-5} and a cell boundary of 100m.

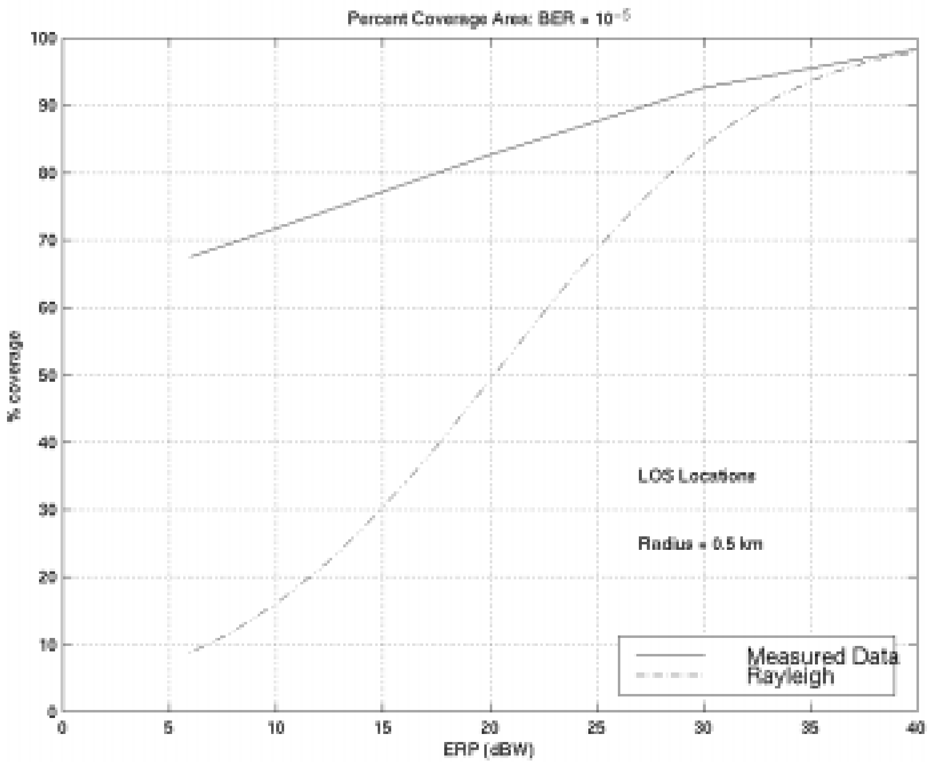

Figure 54: Percent coverage area for the actual correlated K and receive power distribution and the Rayleigh channel assumption for LOS locations with BER = 10^{-5} and a cell boundary of 500m.

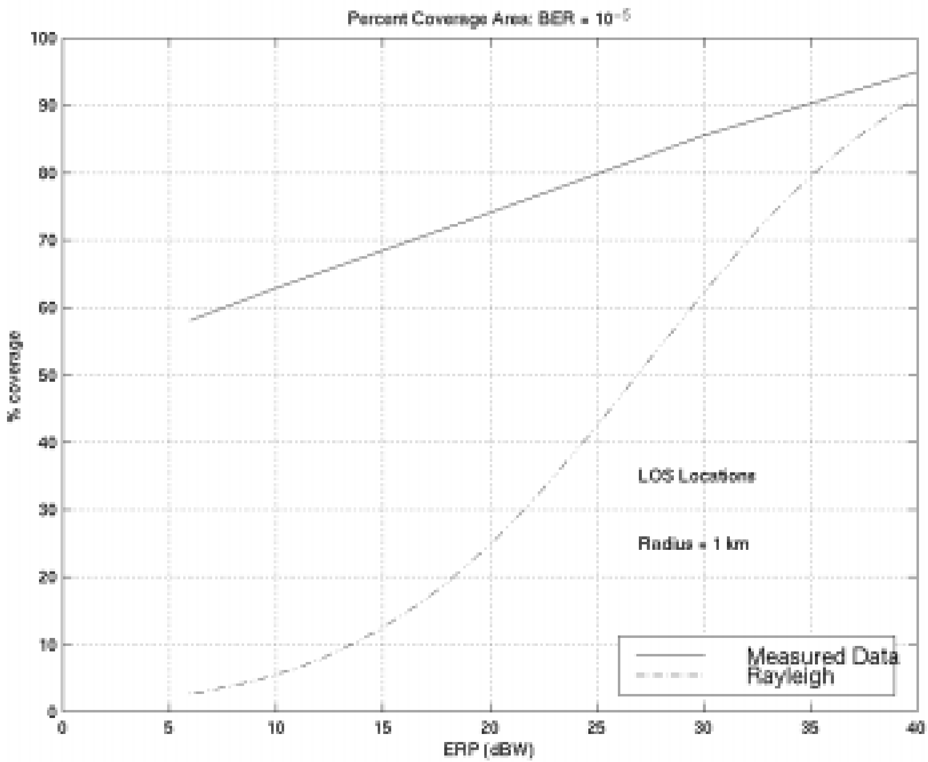

Figure 55: Percent coverage area for the actual correlated K and receive power distribution and the Rayleigh channel assumption for LOS locations with BER = 10^{-5} and a cell boundary of 1 km.

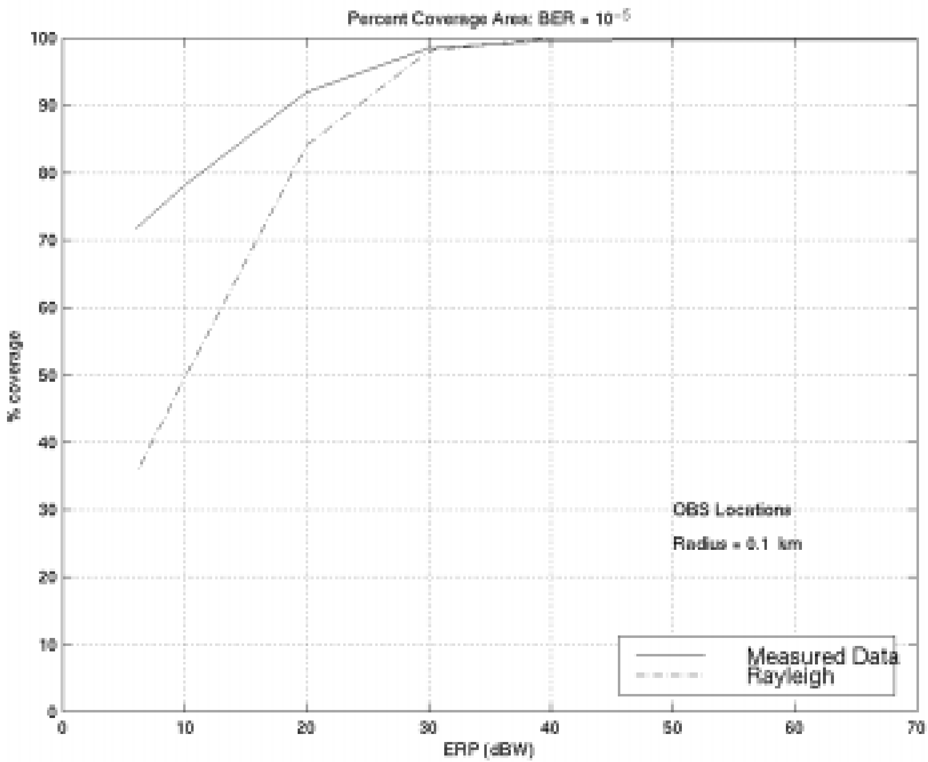

Figure 56: Percent coverage area for the actual correlated K and receive power distribution and the Rayleigh channel assumption for OBS locations with BER = 10^{-5} and a cell boundary of 100m.

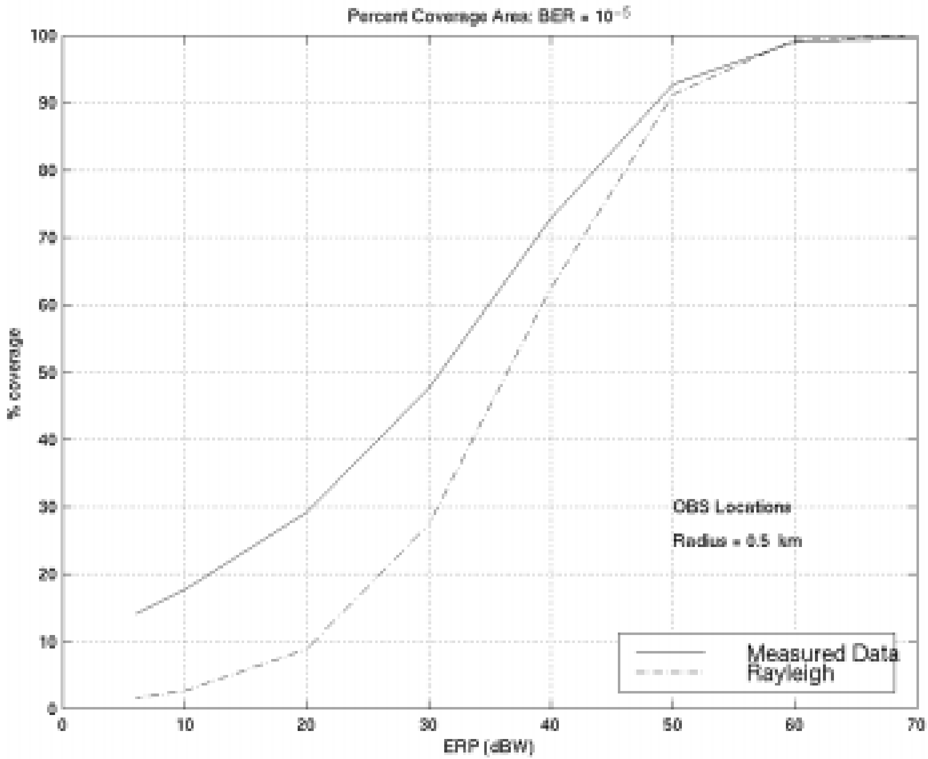

Figure 57: Percent coverage area for the actual correlated K and receive power distribution and the Rayleigh channel assumption for OBS locations with BER = 10^{-5} and a cell boundary of 500m.

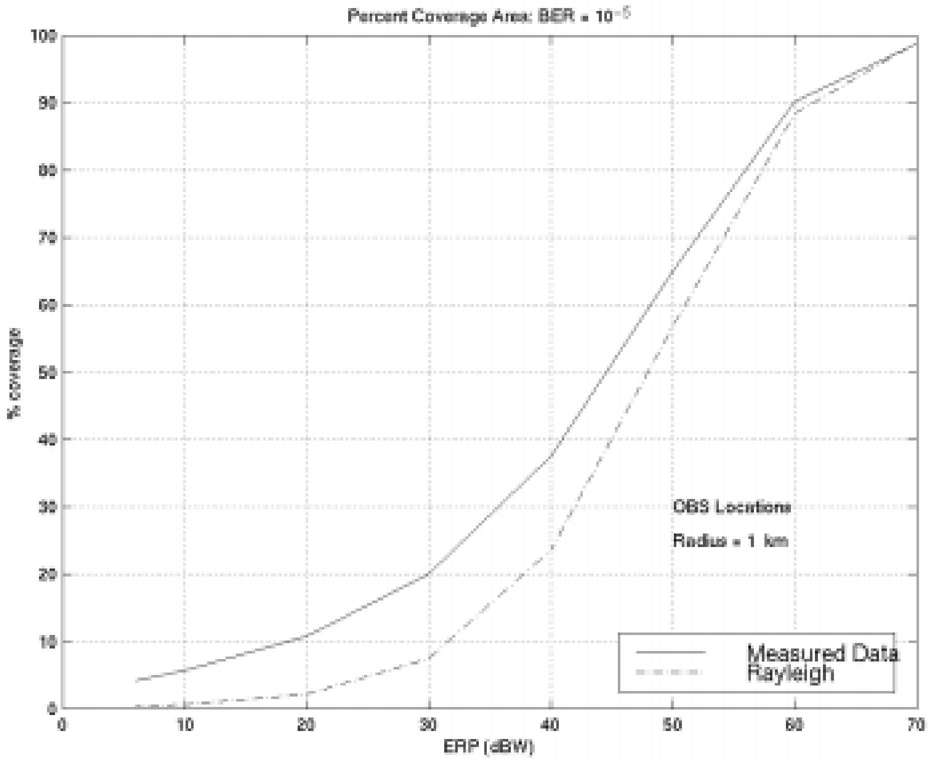

Figure 58: Percent coverage area for the actual correlated K and receive power distribution and the Rayleigh channel assumption for OBS locations with BER = 10^{-5} and a cell boundary of 1 km.

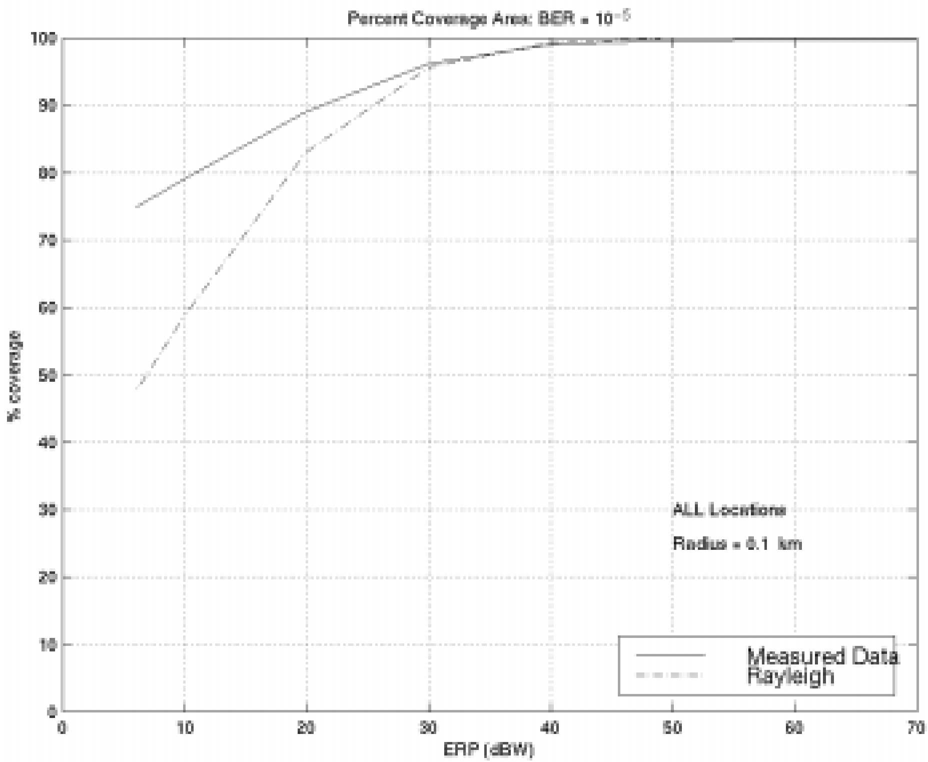

Figure 59: Percent coverage area for the actual correlated K and receive power distribution and the Rayleigh channel assumption for ALL locations with BER = 10^{-5} and a cell boundary of 100m.

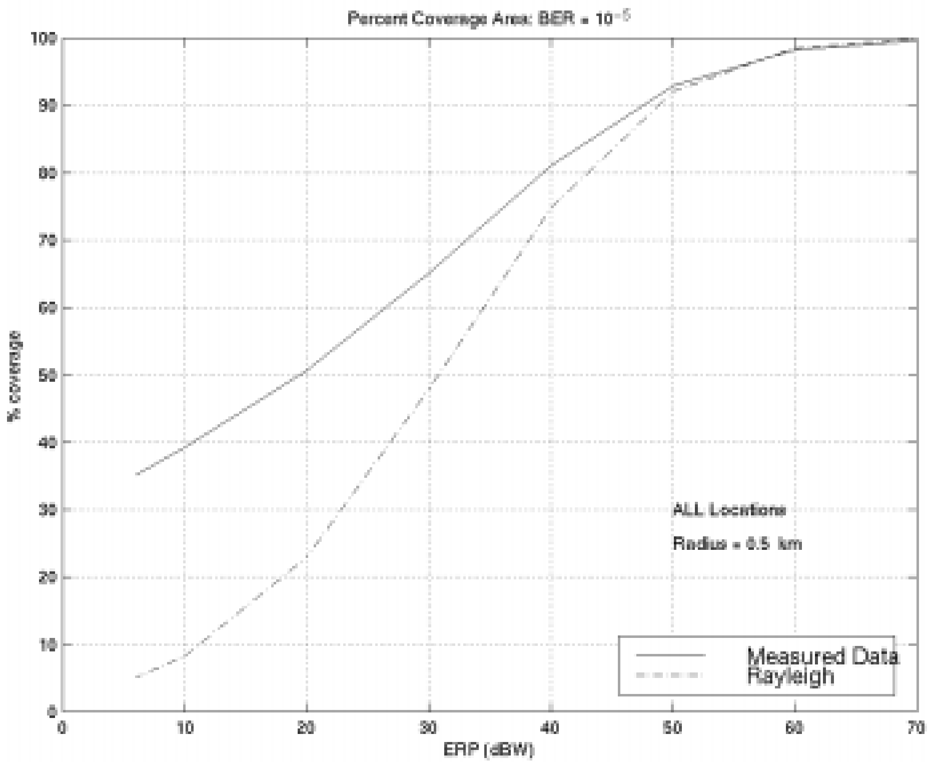

Figure 60: Percent coverage area for the actual correlated K and receive power distribution and the Rayleigh channel assumption for ALL locations with BER = 10^{-5} and a cell boundary of 500m.

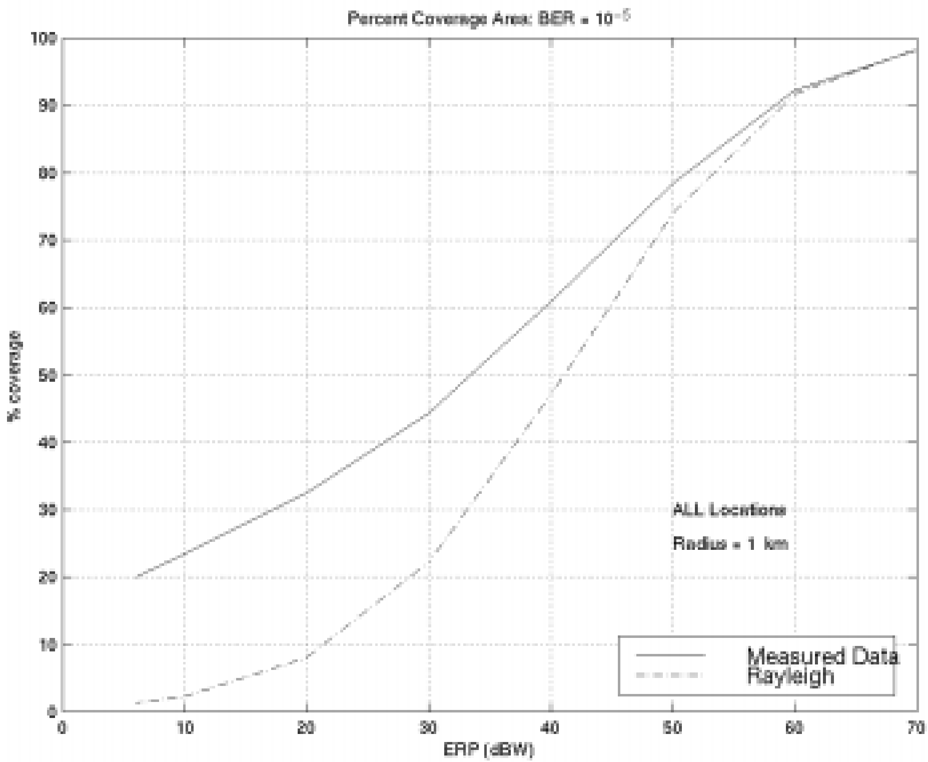

Figure 61: Percent coverage area for the actual correlated K and receive power distribution and the Rayleigh channel assumption for ALL locations with BER = 10^{-5} and a cell boundary of 1 km.

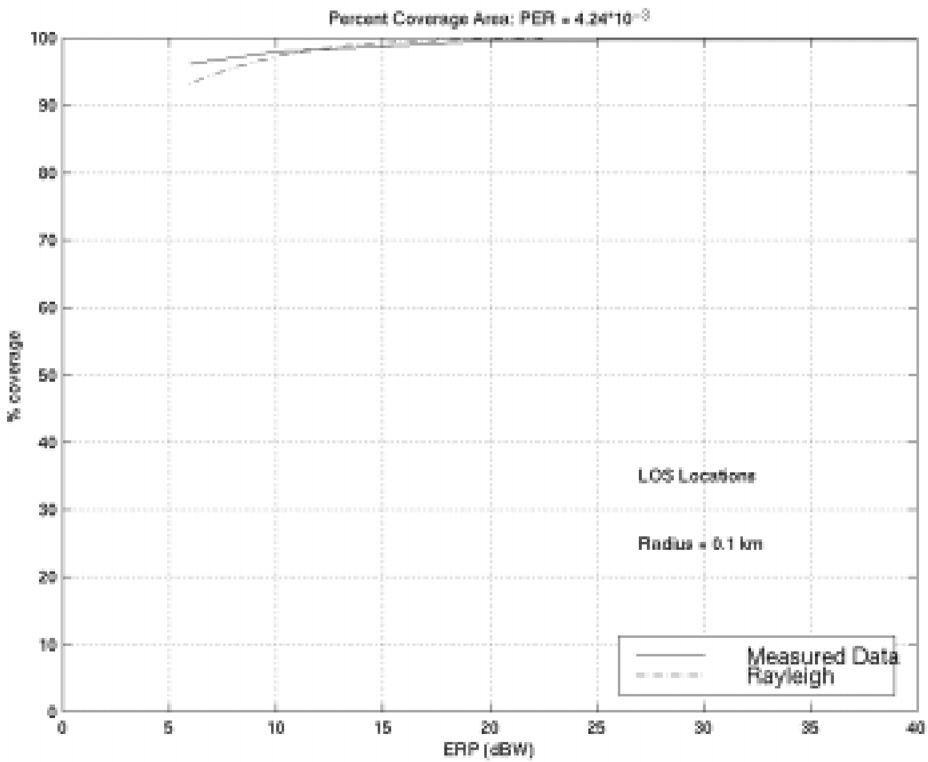

Figure 62: Percent coverage area for the actual correlated K and receive power distribution and the Rayleigh channel assumption for LOS locations with PER = 0.00424 and a cell boundary of 100m.

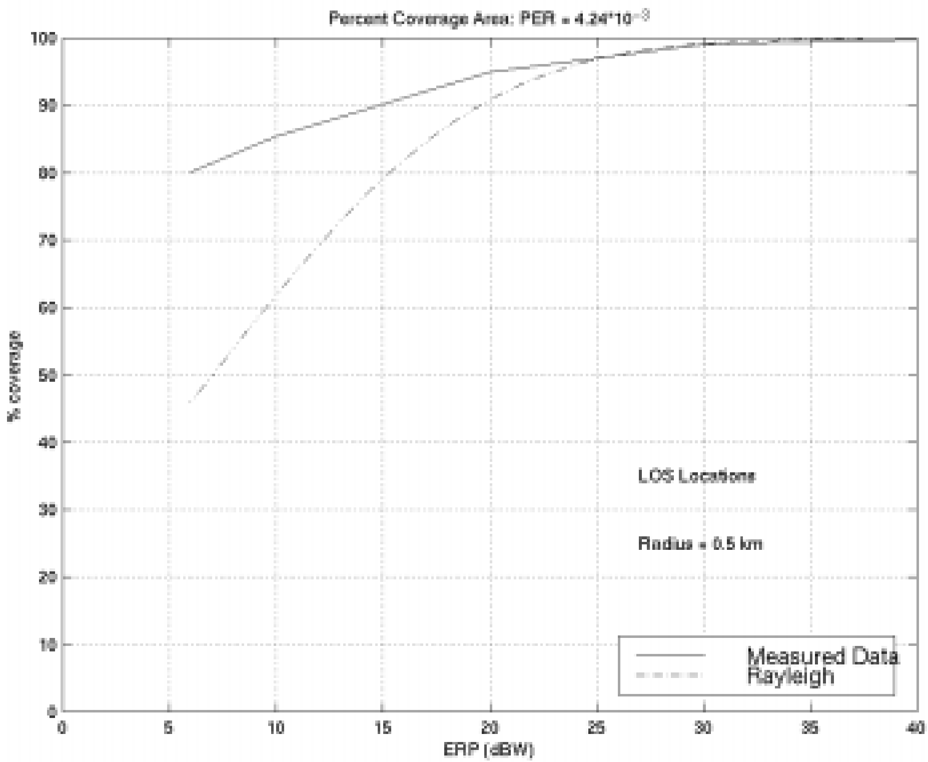

Figure 63: Percent coverage area for the actual correlated K and receive power distribution and the Rayleigh channel assumption for LOS locations with PER = 0.00424 and a cell boundary of 500m.

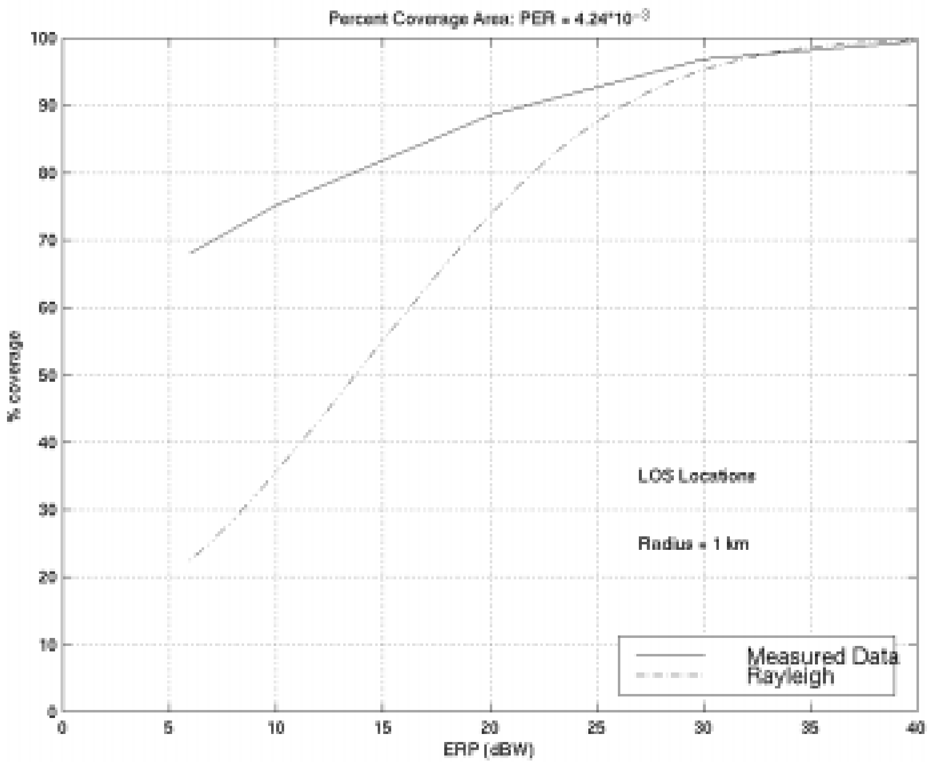

Figure 64: Percent coverage area for the actual correlated K and receive power distribution and the Rayleigh channel assumption for LOS locations with PER = 0.00424 and a cell boundary of 1 km.

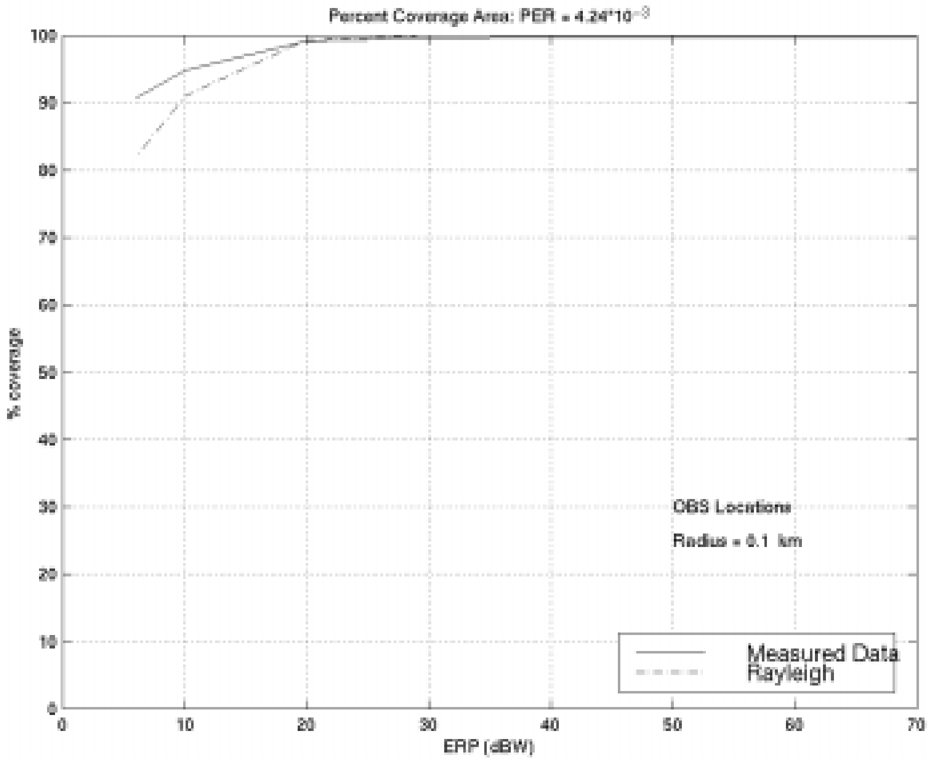

Figure 65: Percent coverage area for the actual correlated K and receive power distribution and the Rayleigh channel assumption for OBS locations with PER = 0.00424 and a cell boundary of 100m.

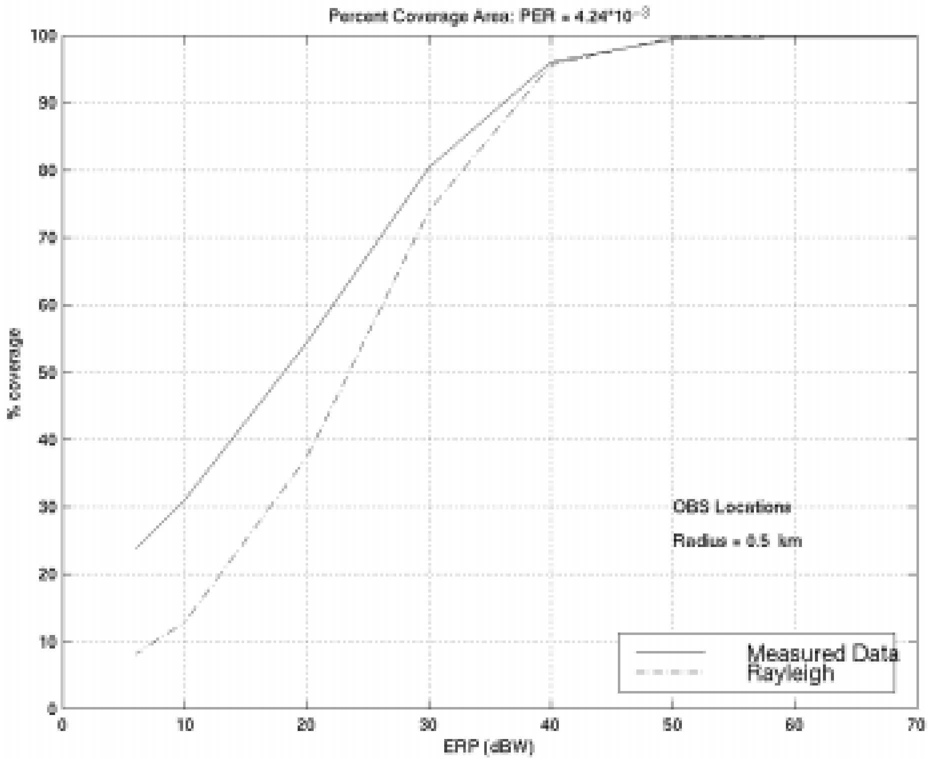

Figure 66: Percent coverage area for the actual correlated K and receive power distribution and the Rayleigh channel assumption for OBS locations with PER = 0.00424 and a cell boundary of 500m.

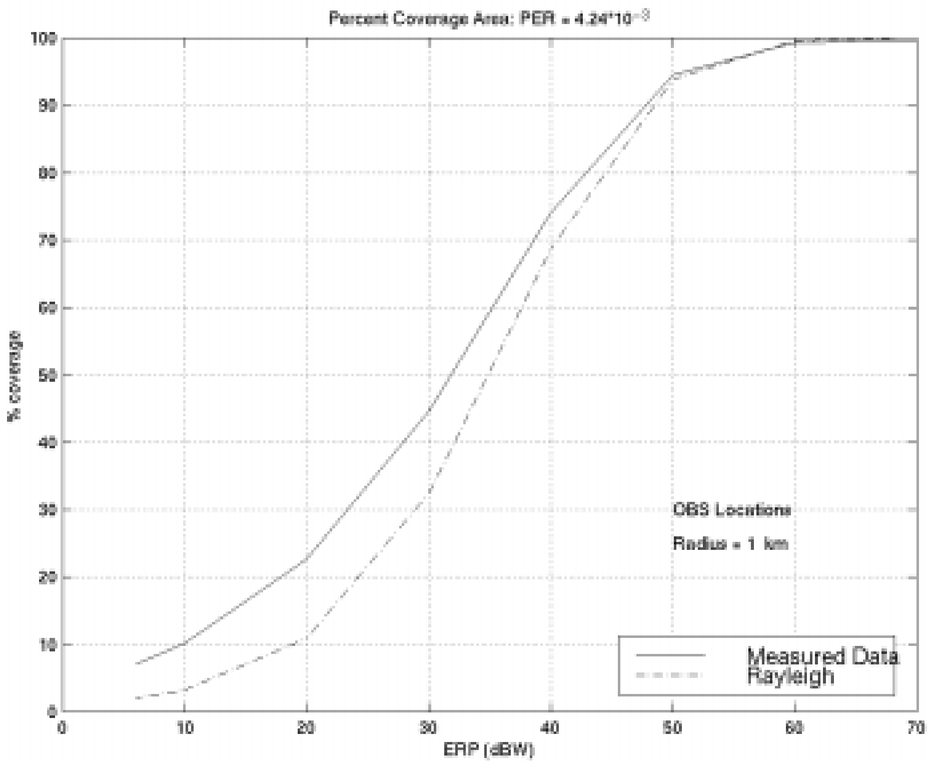

Figure 67: Percent coverage area for the actual correlated K and receive power distribution and the Rayleigh channel assumption for OBS locations with PER = 0.00424 and a cell boundary of 1 km.

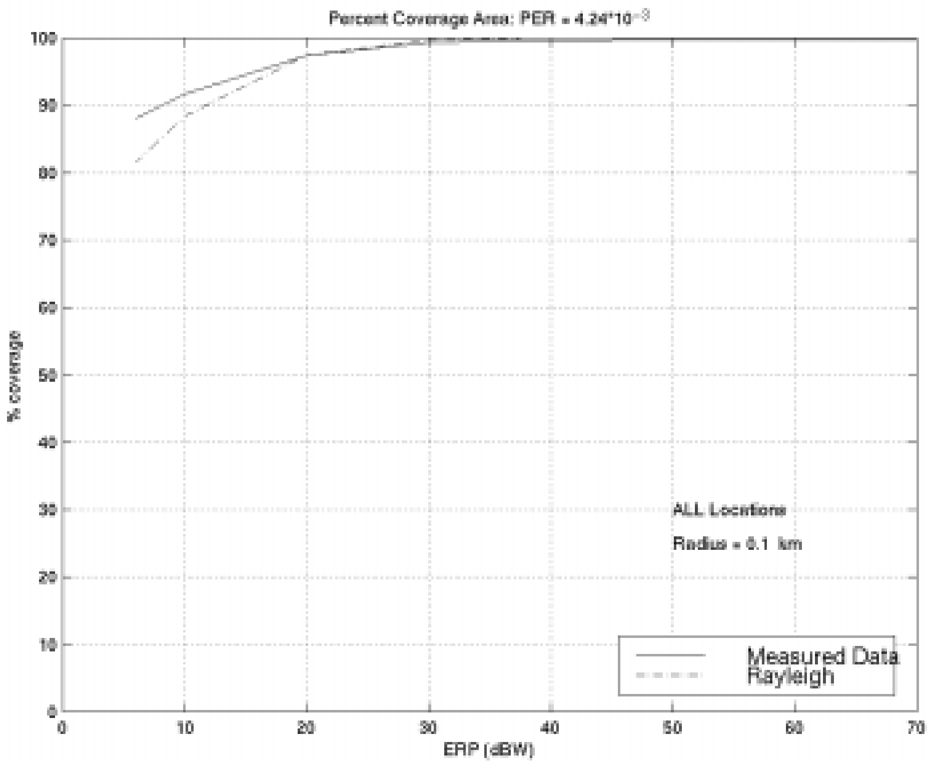

Figure 68: Percent coverage area for the actual correlated K and receive power distribution and the Rayleigh channel assumption for ALL locations with PER = 0.00424 and a cell boundary of 100m.

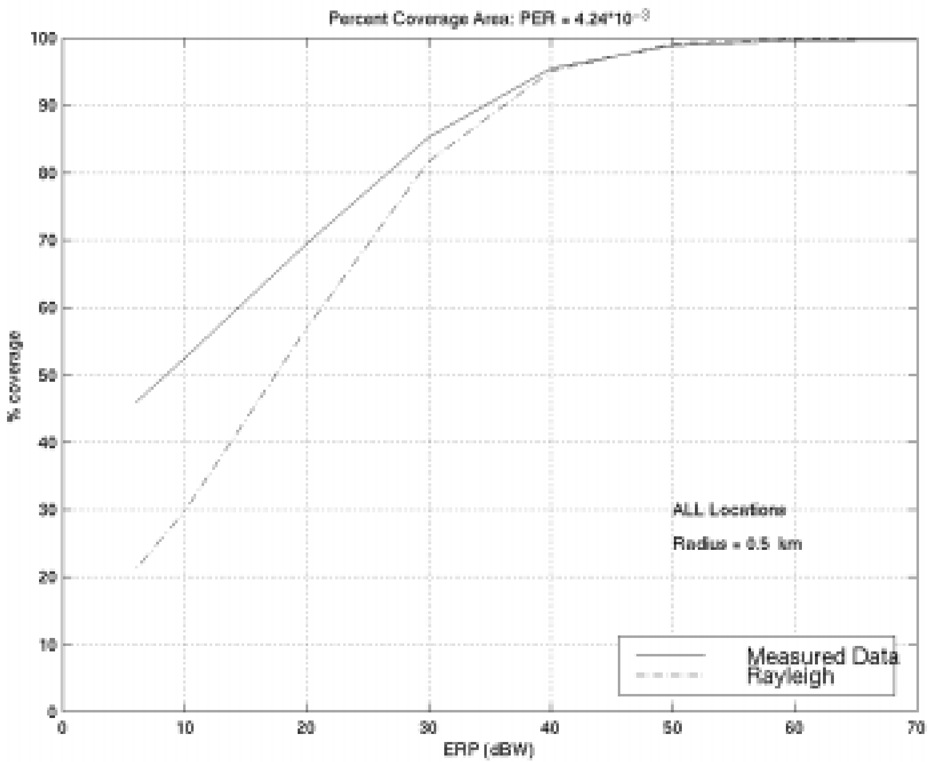

Figure 69: Percent coverage area for the actual correlated K and receive power distribution and the Rayleigh channel assumption for ALL locations with PER = 0.00424 and a cell boundary of 500m.

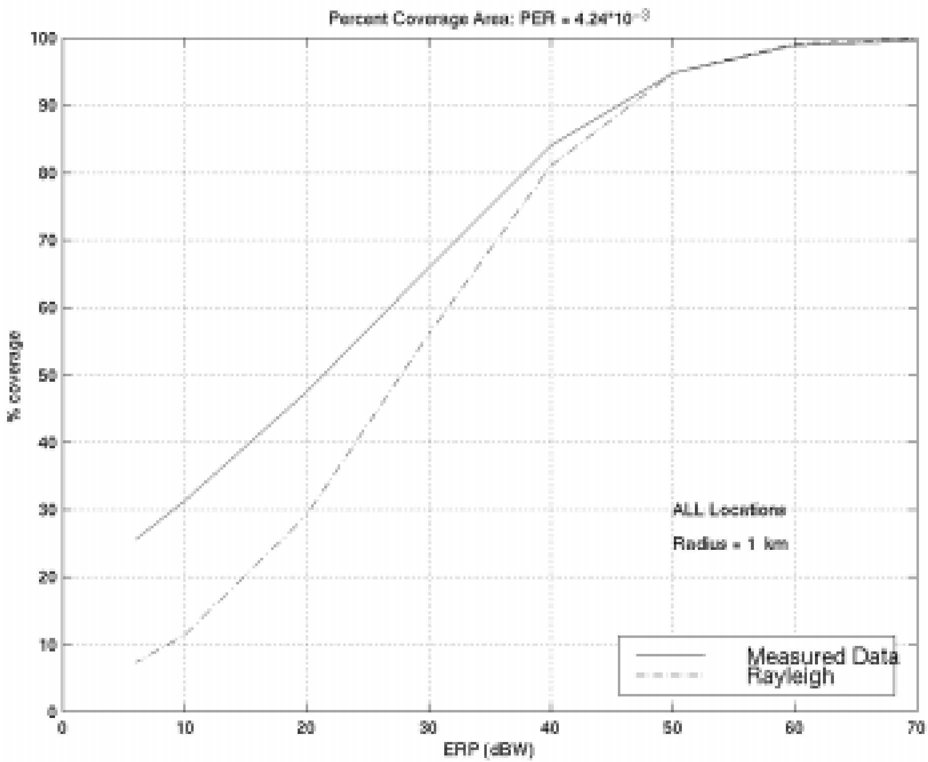

Figure 70: Percent coverage area for the actual correlated K and receive power distribution and the Rayleigh channel assumption for ALL locations with PER = 0.00424 and a cell boundary of 1 km.

4.8 Percent Coverage Area: BER vs PER Comparison

Figures 72-92 compare BER and PER approaches for each set of locations and for different values of ERP. The specified BER is 10^{-5} and equivalent PER for one ATM cell (424 bits) is $4.24 \cdot 10^{-3}$. The results were compiled for the parameters found for the collected data. As expected, sending data in packets increases the percentage of locations for which the link closes. For the LOS locations, an increase of 5 to 15 percent at 1 km is seen when a packet size of 424 bits is employed. The OBS locations show an increase of up to 35 percent at 1 km for a 424 bit packet.

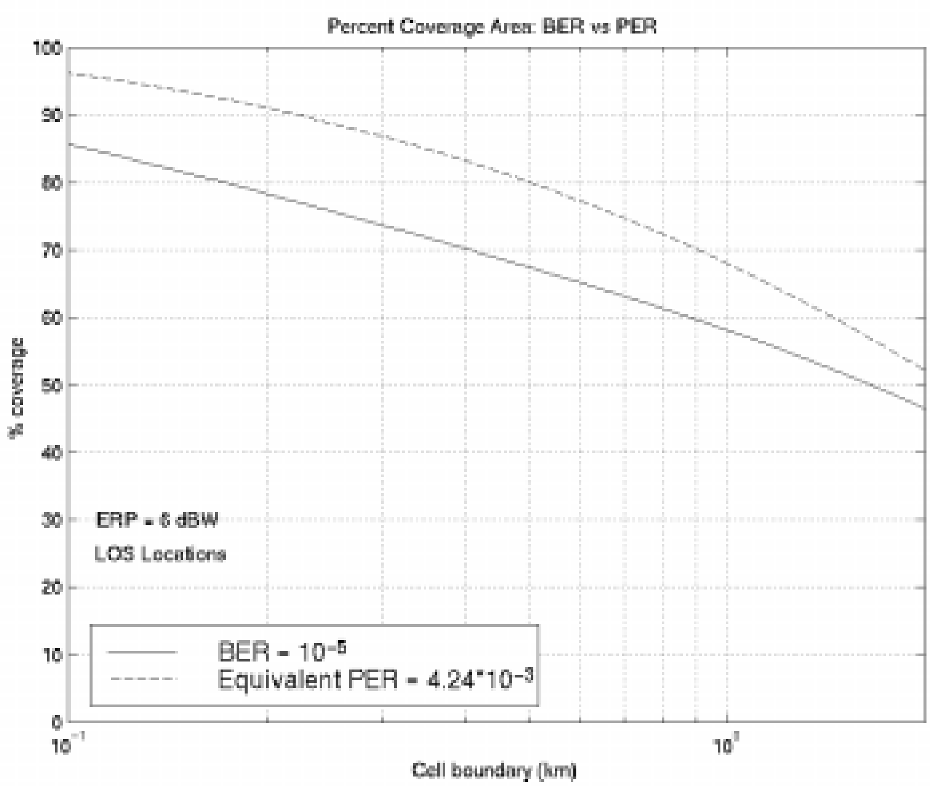

Figure 71: Percent coverage area comparison between BER and equivalent PER for an ATM cell for LOS locations and ERP = 6 dBW.

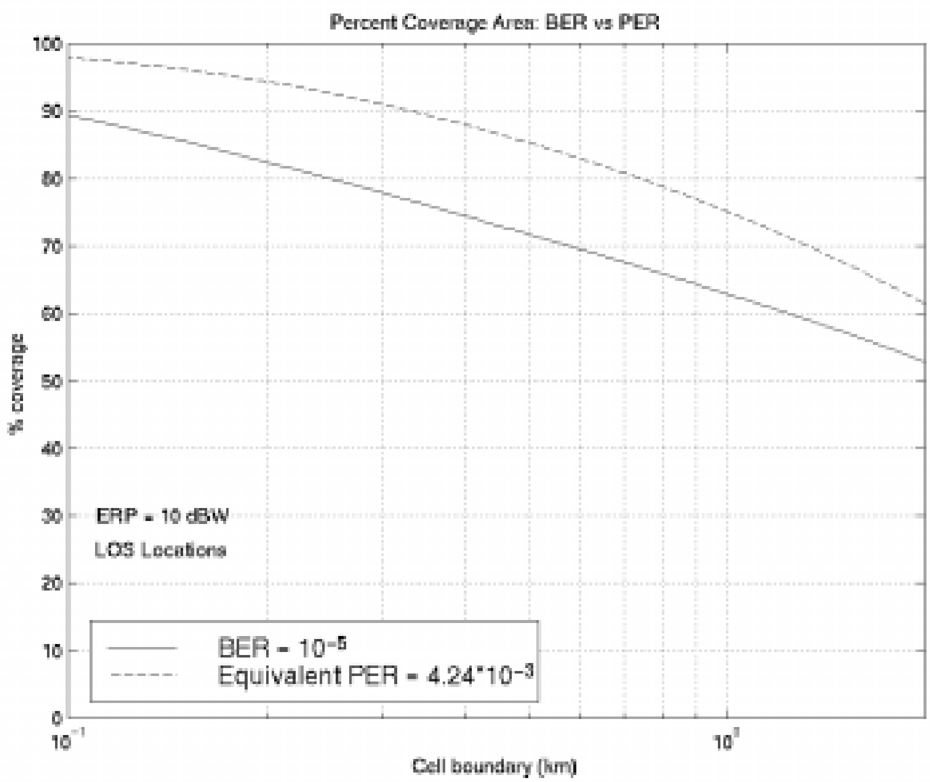

Figure 72: Percent coverage area comparison between BER and equivalent PER for an ATM cell for LOS locations and ERP = 10 dBW.

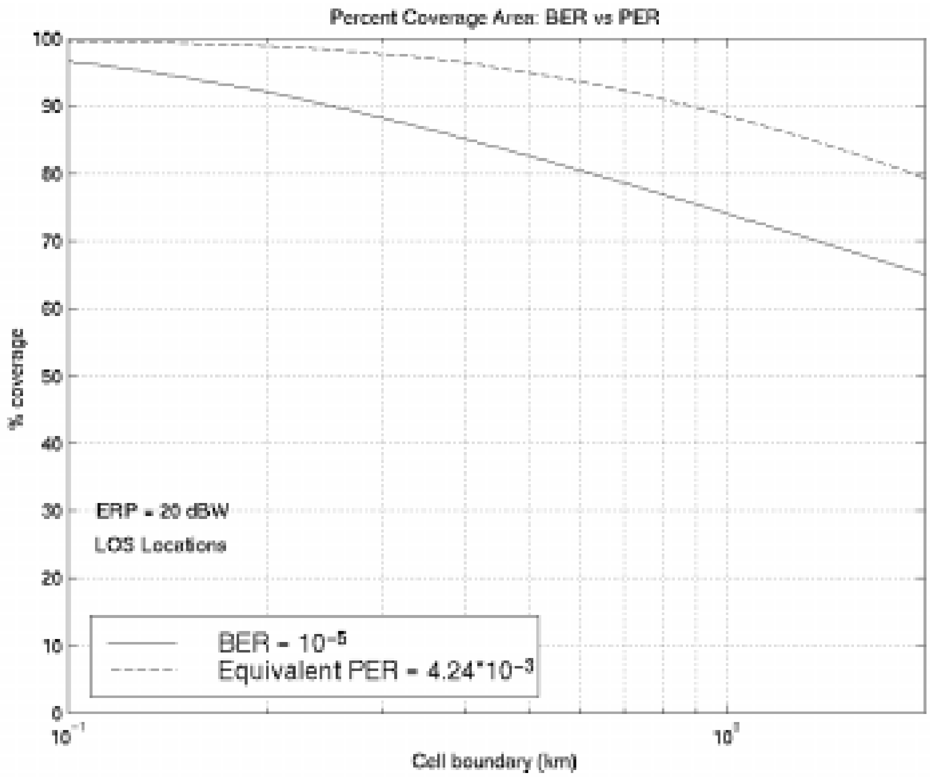

Figure 73: Percent coverage area comparison between BER and equivalent PER for an ATM cell for LOS locations and ERP = 20 dBW.

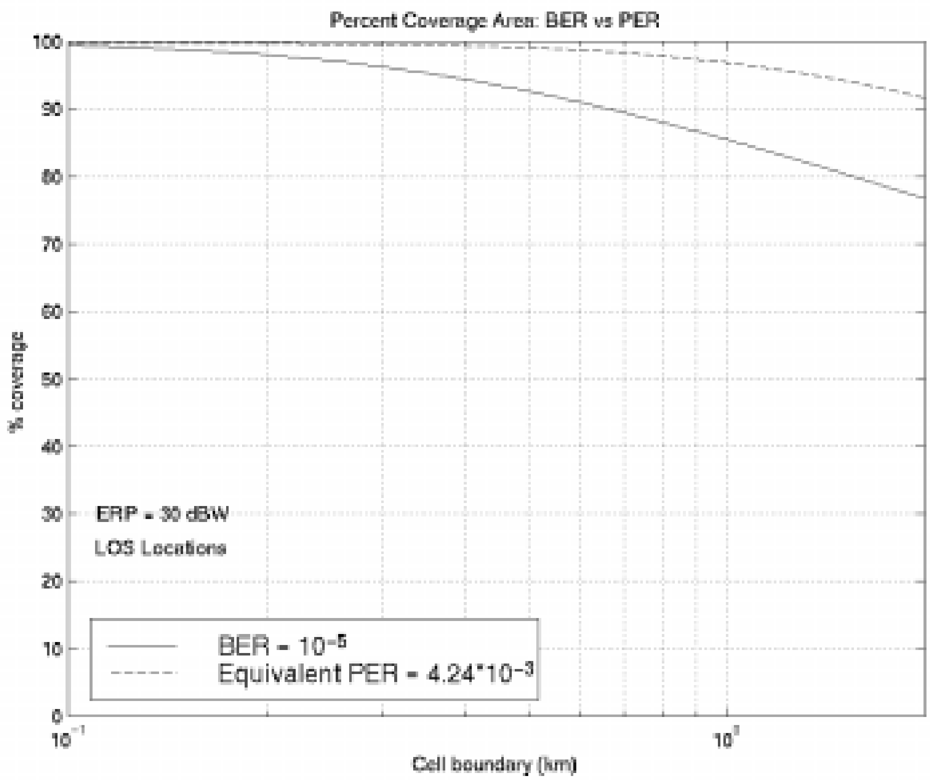

Figure 74: Percent coverage area comparison between BER and equivalent PER for an ATM cell for LOS locations and ERP = 30 dBW.

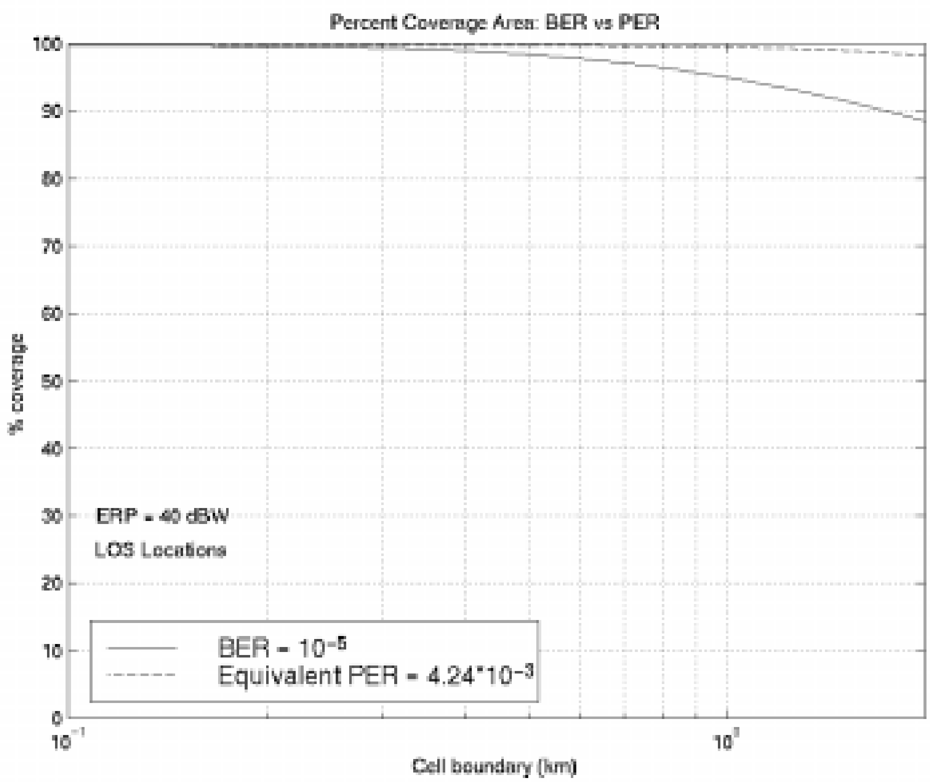

Figure 75: Percent coverage area comparison between BER and equivalent PER for an ATM cell for LOS locations and ERP = 40 dBW.

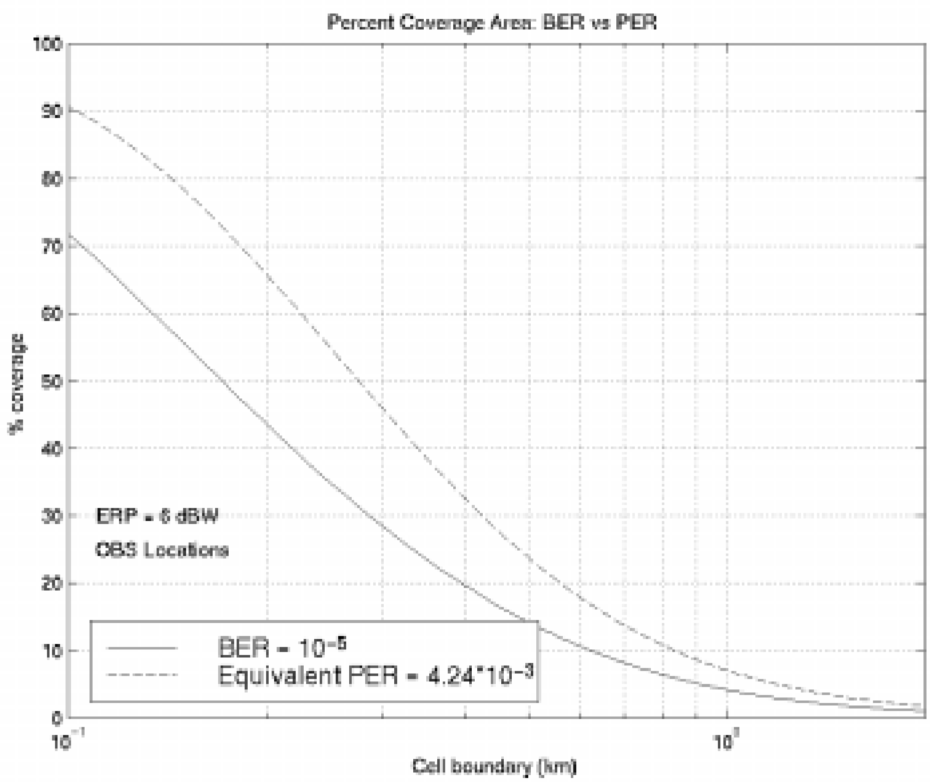

Figure 76: Percent coverage area comparison between BER and equivalent PER for an ATM cell for OBS locations and ERP = 6 dBW.

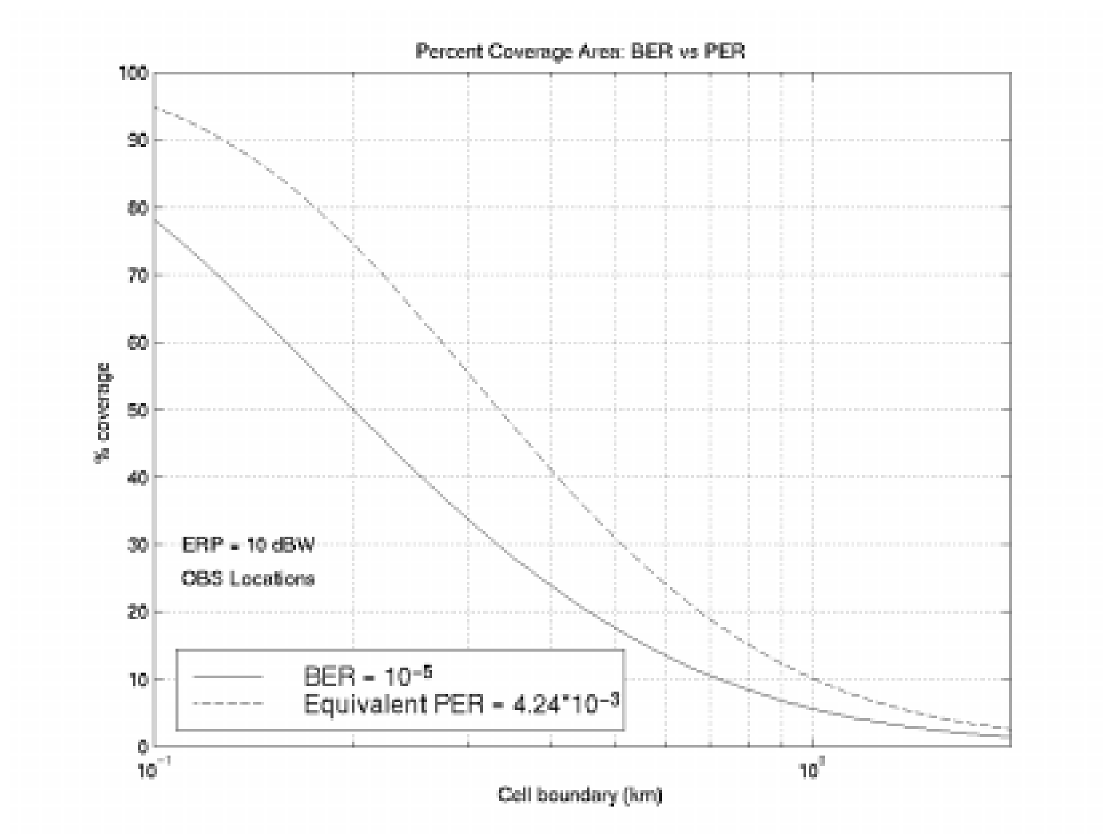

Figure 77: Percent coverage area comparison between BER and equivalent PER for an ATM cell for OBS locations and ERP = 10 dBW.

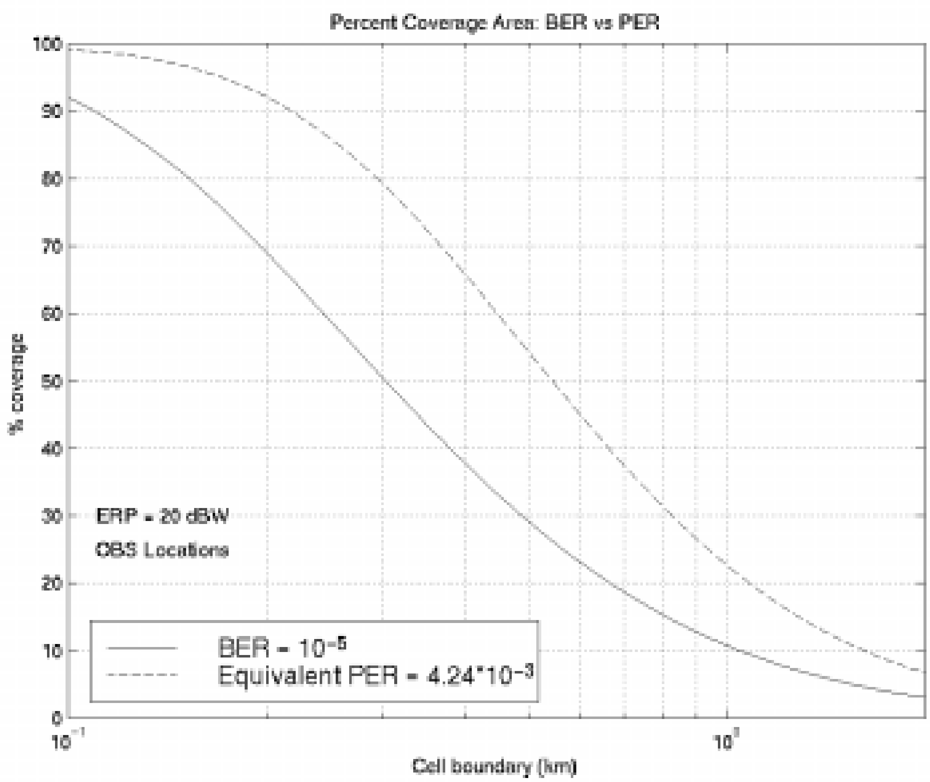

Figure 78: Percent coverage area comparison between BER and equivalent PER for an ATM cell for OBS locations and ERP = 20 dBW.

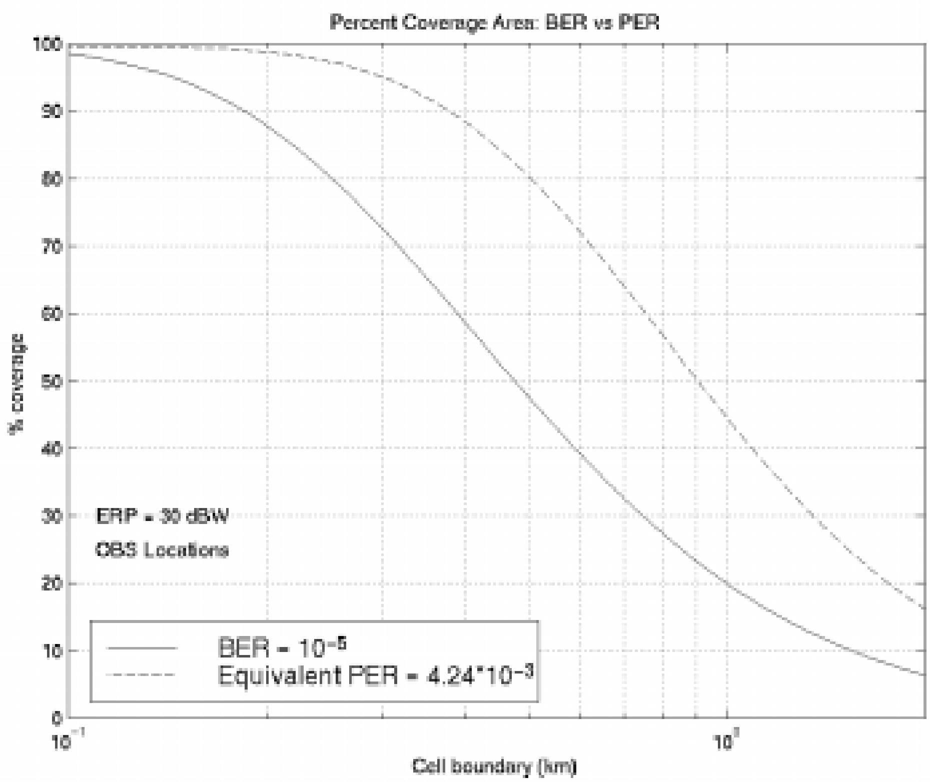

Figure 79: Percent coverage area comparison between BER and equivalent PER for an ATM cell for OBS locations and ERP = 30 dBW.

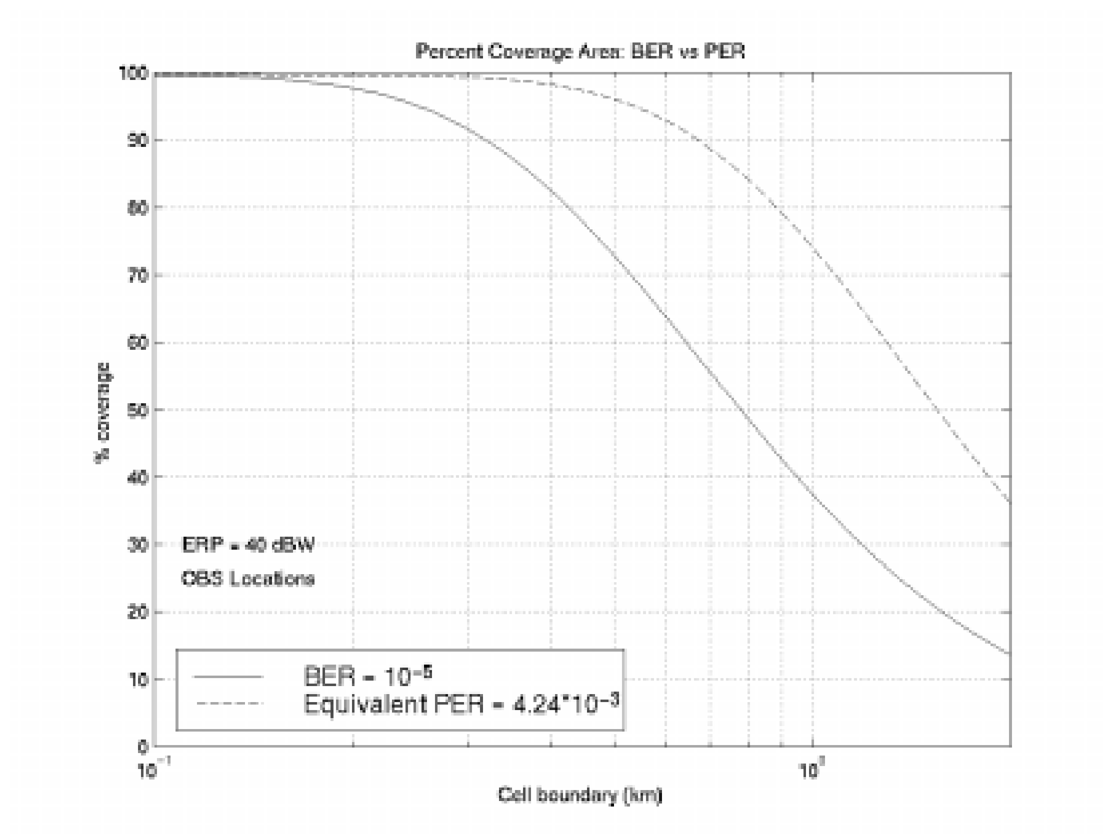

Figure 80: Percent coverage area comparison between BER and equivalent PER for an ATM cell for OBS locations and ERP = 40 dBW.

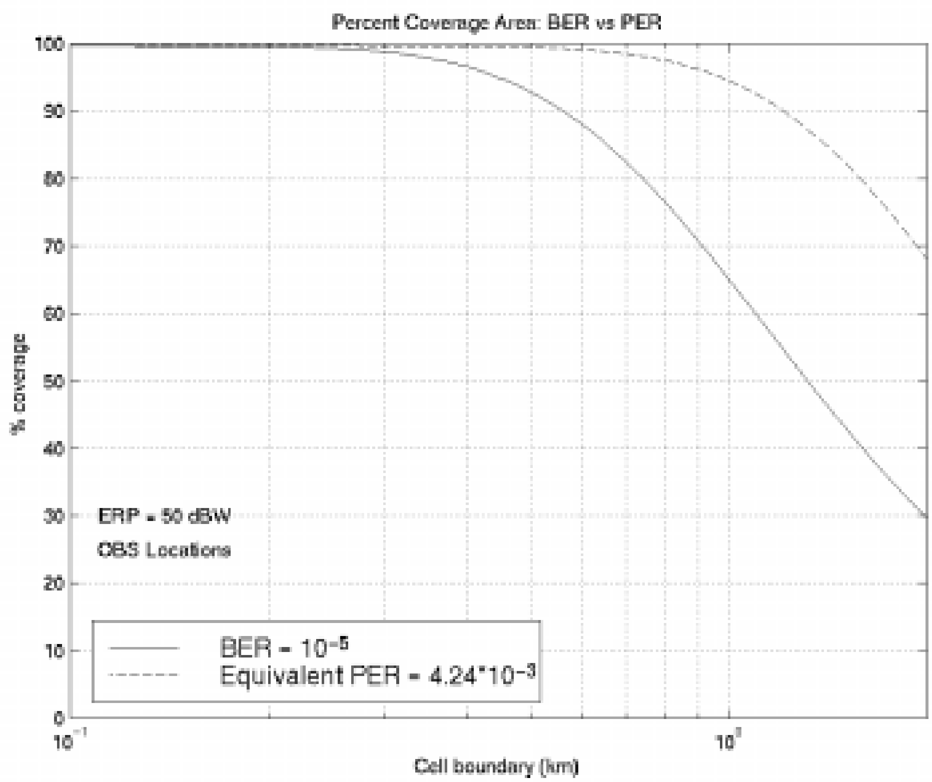

Figure 81: Percent coverage area comparison between BER and equivalent PER for an ATM cell for OBS locations and ERP = 50 dBW.

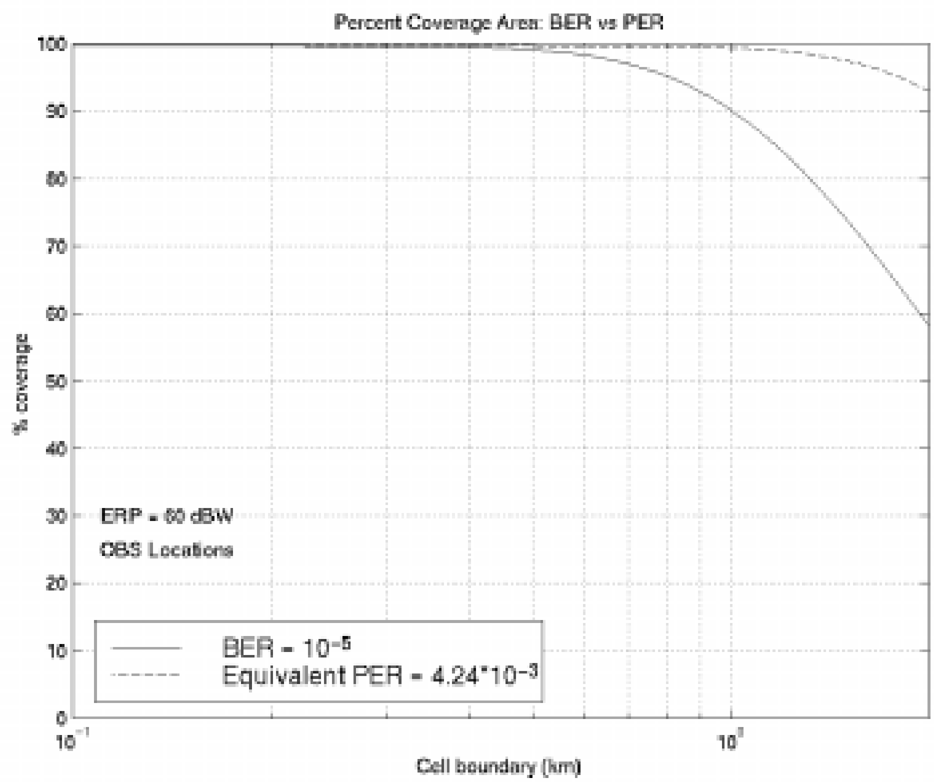

Figure 82: Percent coverage area comparison between BER and equivalent PER for an ATM cell for OBS locations and ERP = 60 dBW.

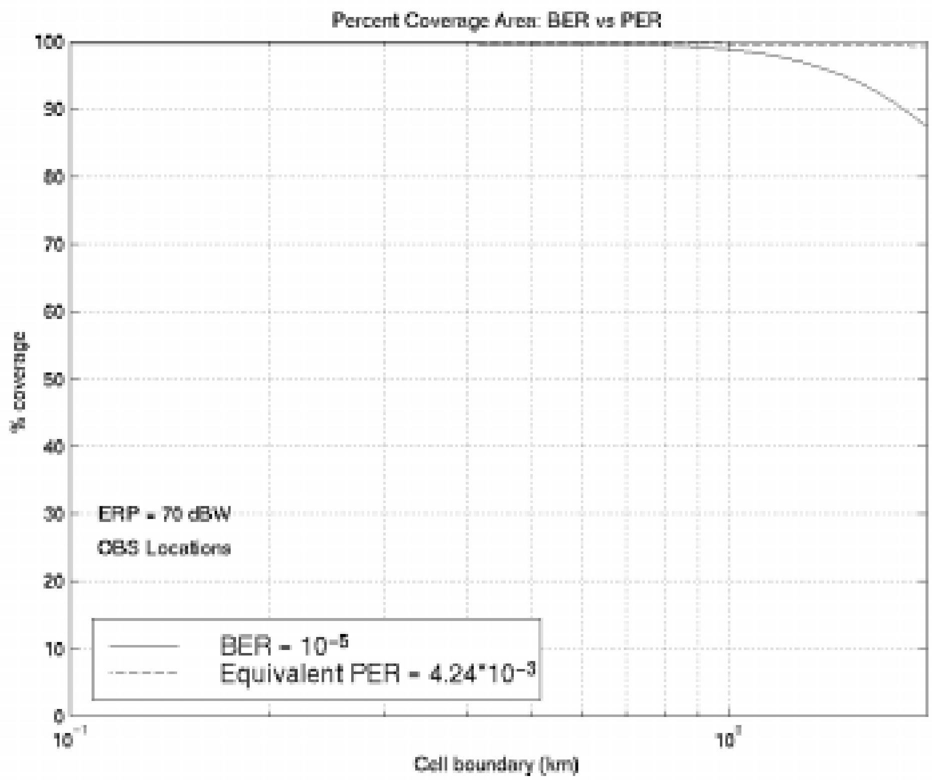

Figure 83: Percent coverage area comparison between BER and equivalent PER for an ATM cell for OBS locations and ERP = 70 dBW.

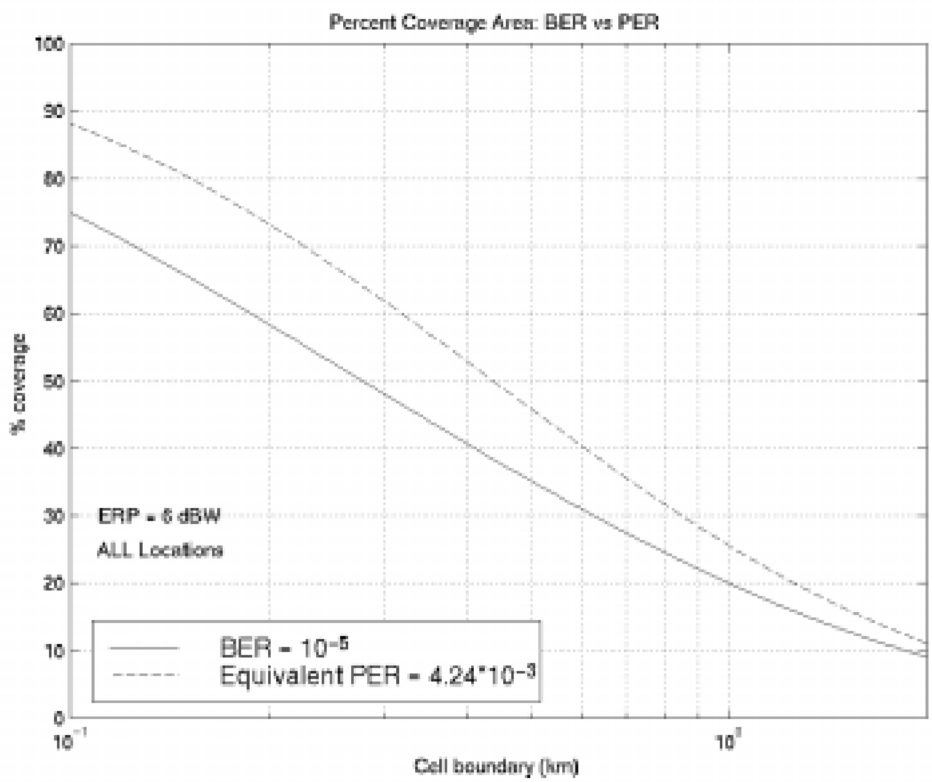

Figure 84: Percent coverage area comparison between BER and equivalent PER for an ATM cell for ALL locations and ERP = 6 dBW.

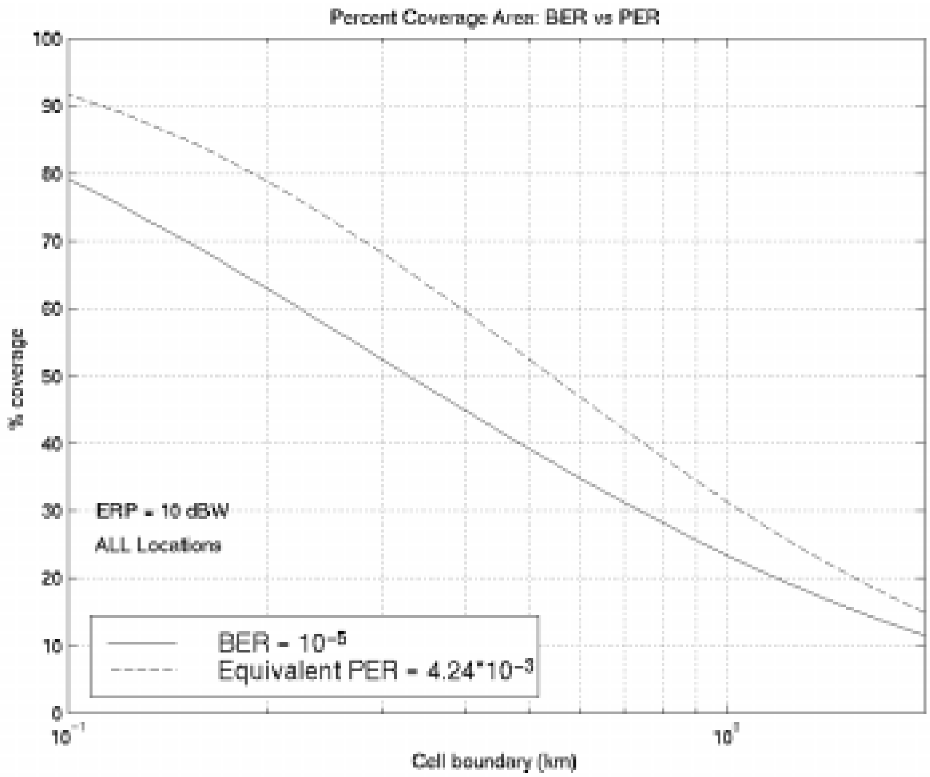

Figure 85: Percent coverage area comparison between BER and equivalent PER for an ATM cell for ALL locations and ERP = 10 dBW.

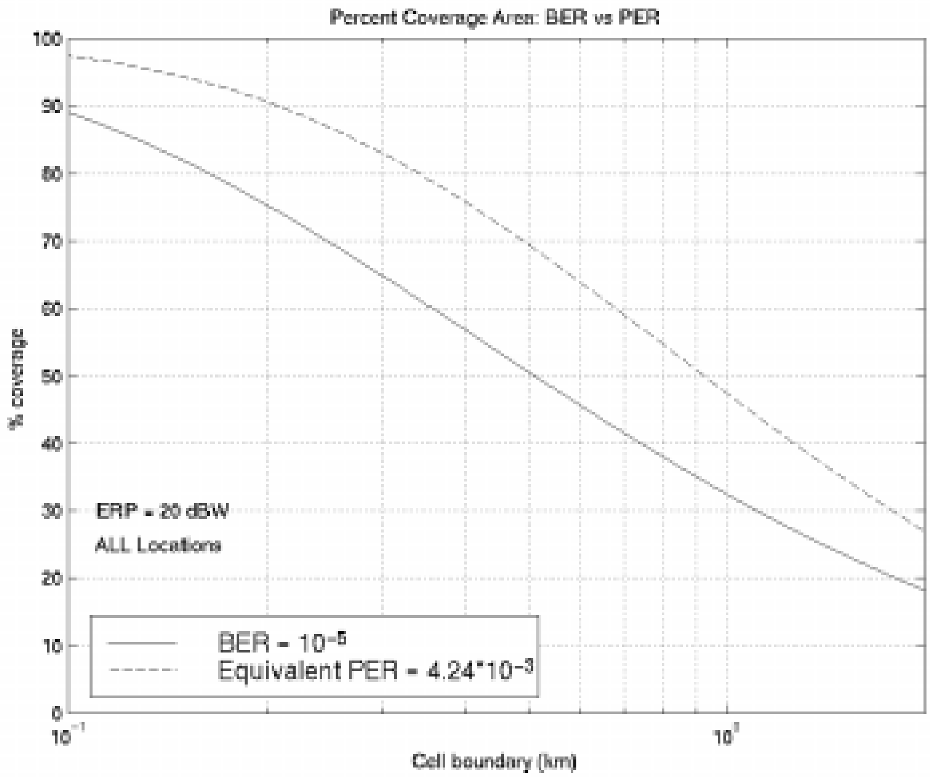

Figure 86: Percent coverage area comparison between BER and equivalent PER for an ATM cell for ALL locations and ERP = 20 dBW.

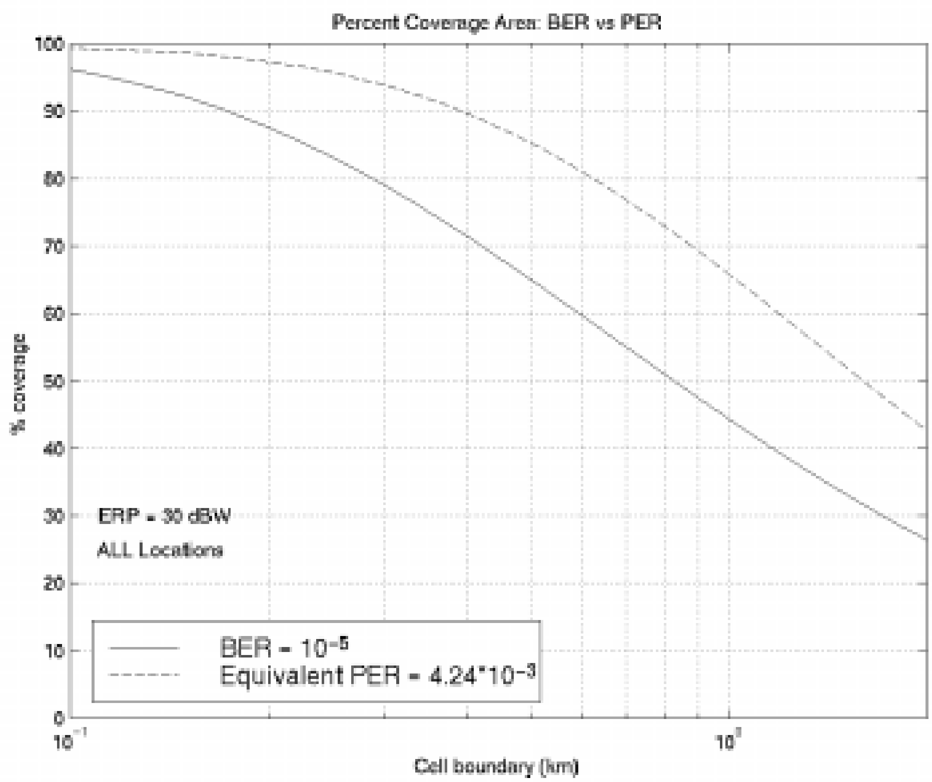

Figure 87: Percent coverage area comparison between BER and equivalent PER for an ATM cell for ALL locations and ERP = 30 dBW.

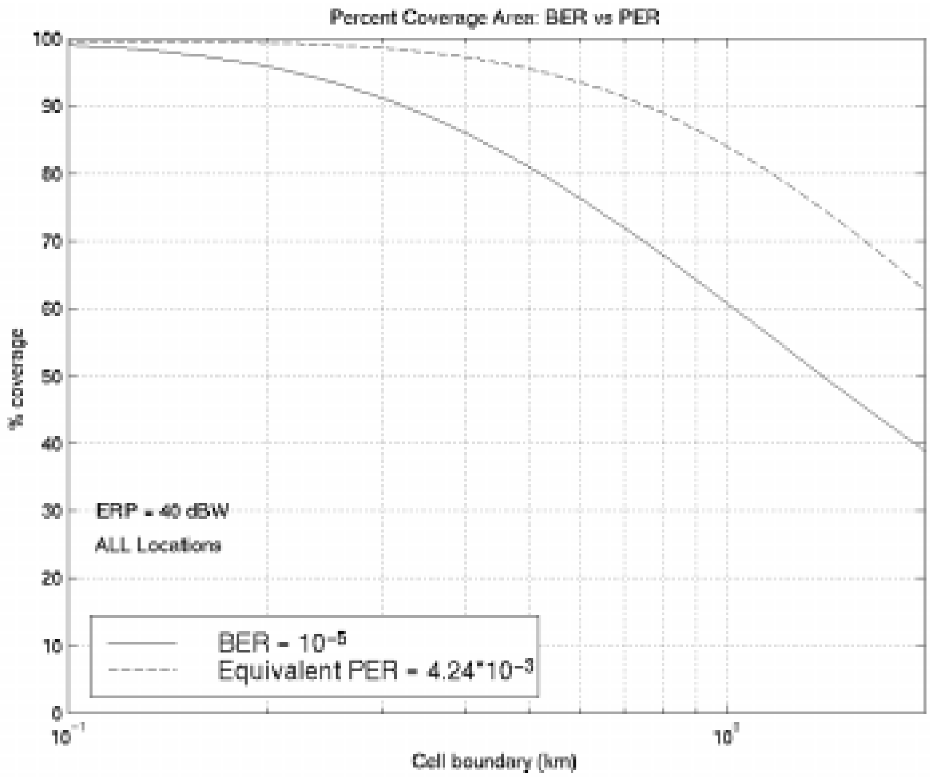

Figure 88: Percent coverage area comparison between BER and equivalent PER for an ATM cell for ALL locations and ERP = 40 dBW.

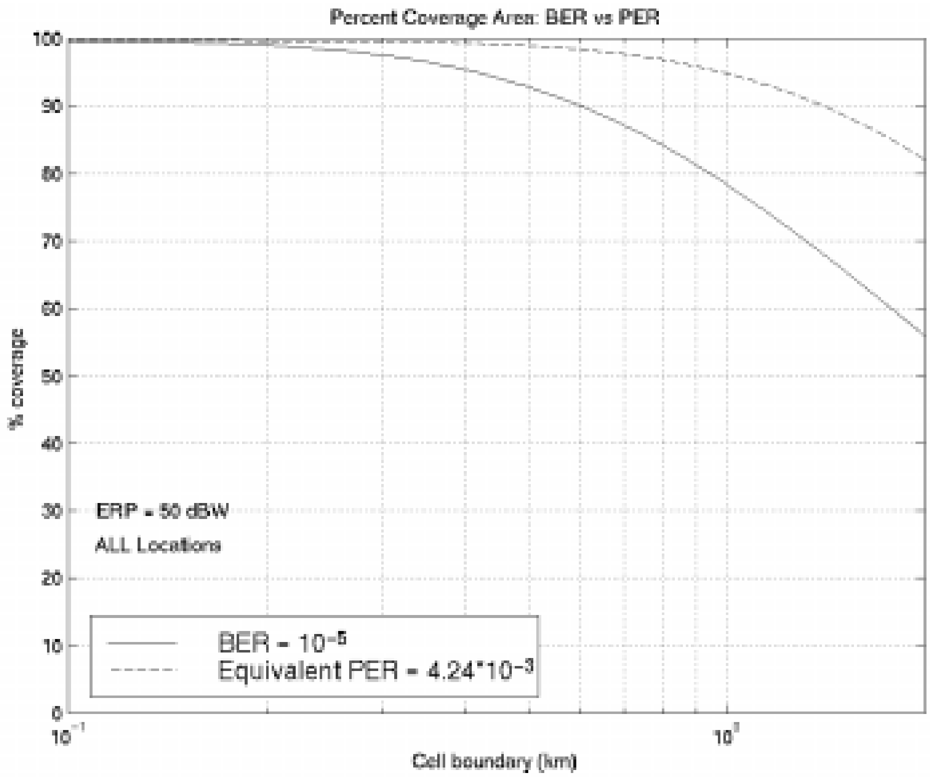

Figure 89: Percent coverage area comparison between BER and equivalent PER for an ATM cell for ALL locations and ERP = 50 dBW.

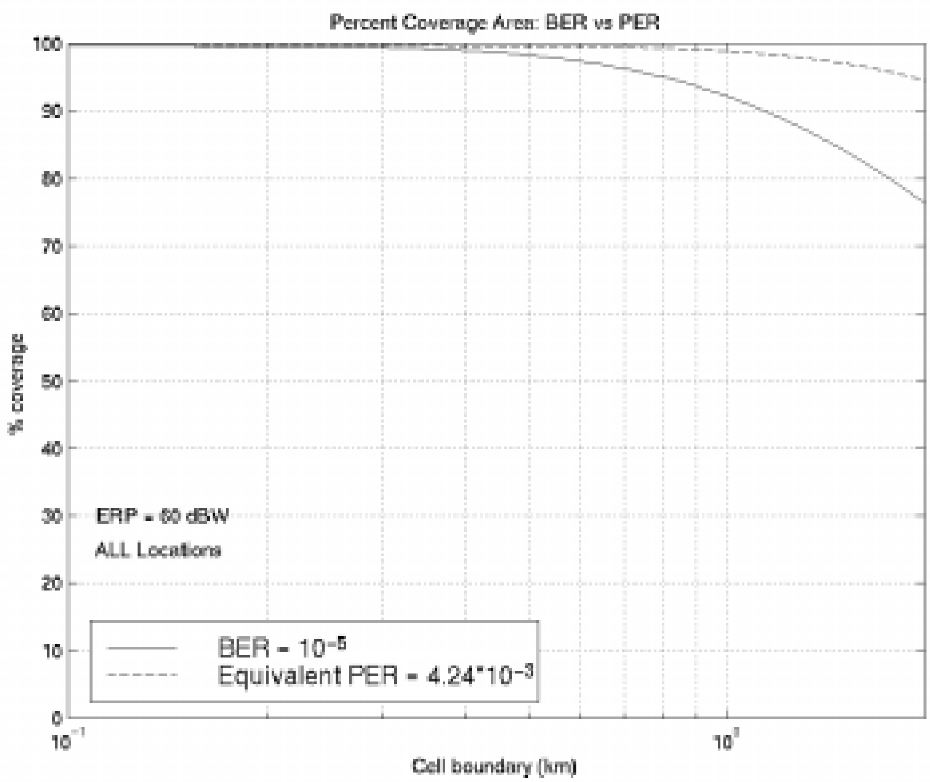

Figure 90: Percent coverage area comparison between BER and equivalent PER for an ATM cell for ALL locations and ERP = 60 dBW.

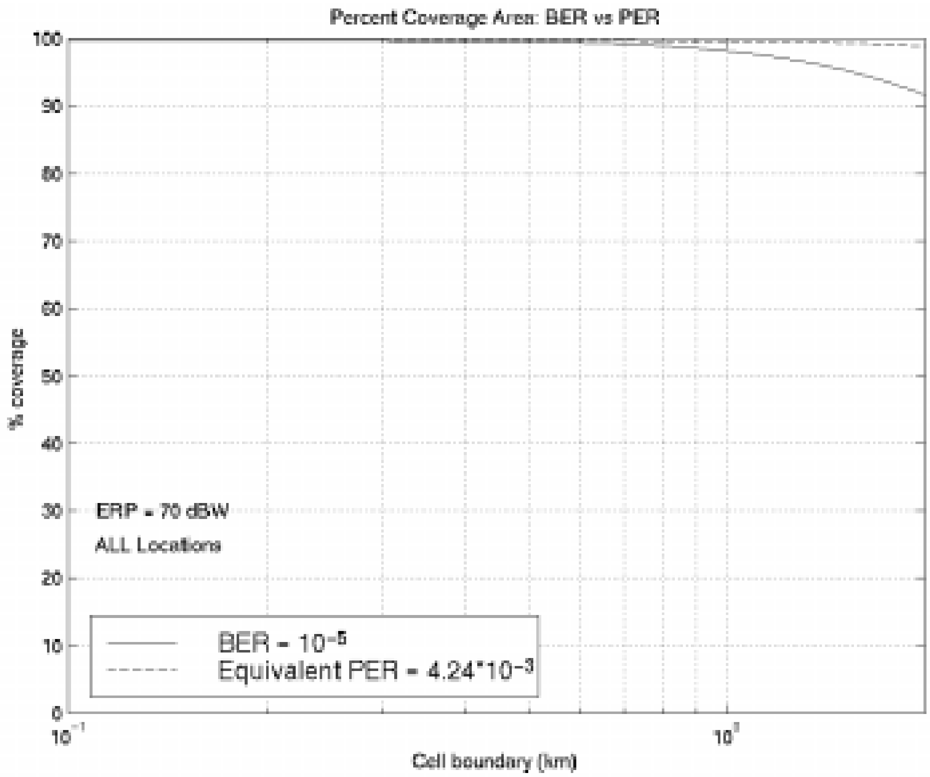

Figure 91: Percent coverage area comparison between BER and equivalent PER for an ATM cell for ALL locations and ERP = 70 dBW.

4.9 Percent Coverage Area: ERP Family of Curves

Figures 93-98 show a family of curves of different ERPs for the cases when BER and PER (424 bits) approaches are employed and also for LOS, OBS and ALL location cases.

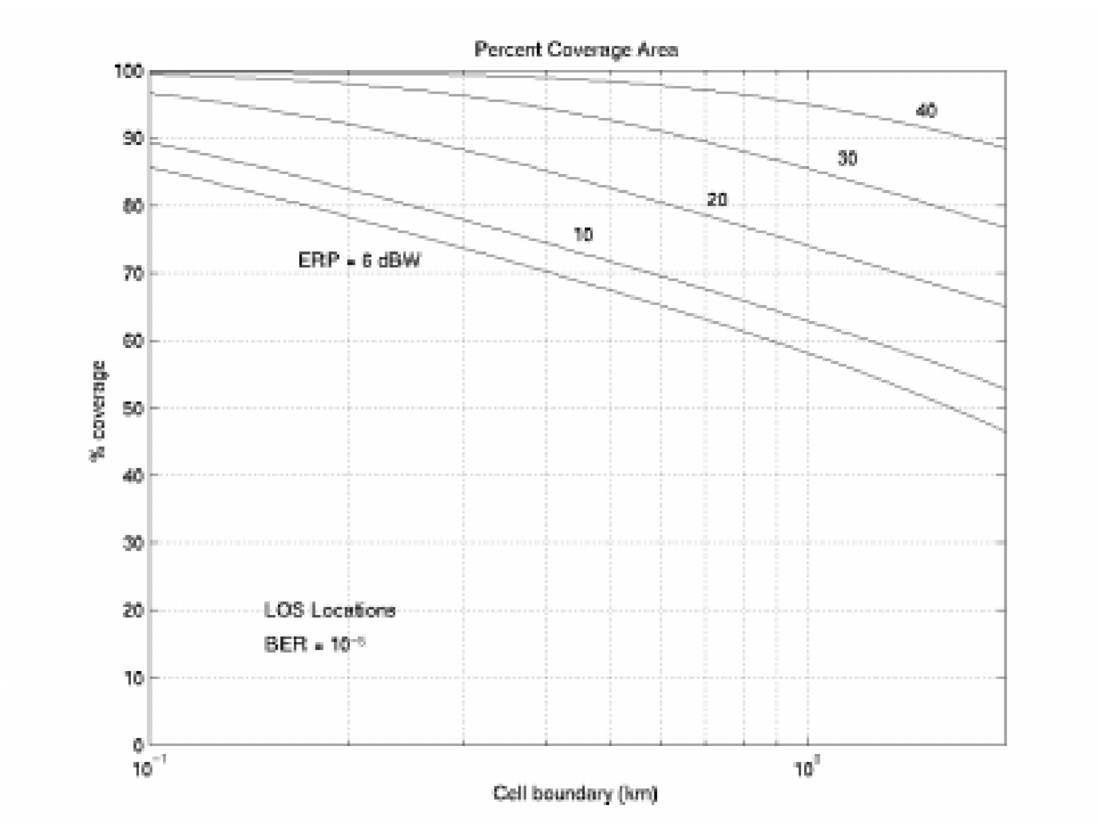

Figure 92: Percent coverage area over distance for various values of ERP using LOS locations and BER = 10^{-5}.

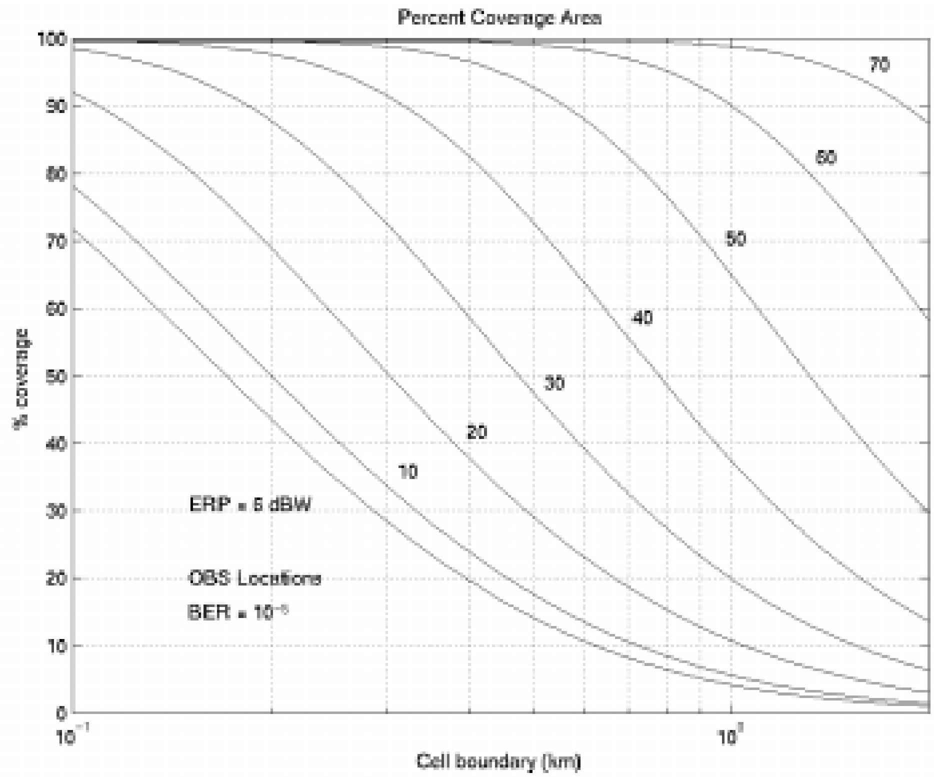

Figure 93: Percent coverage area over distance for various values of ERP using OBS locations and BER = 10^{-5}.

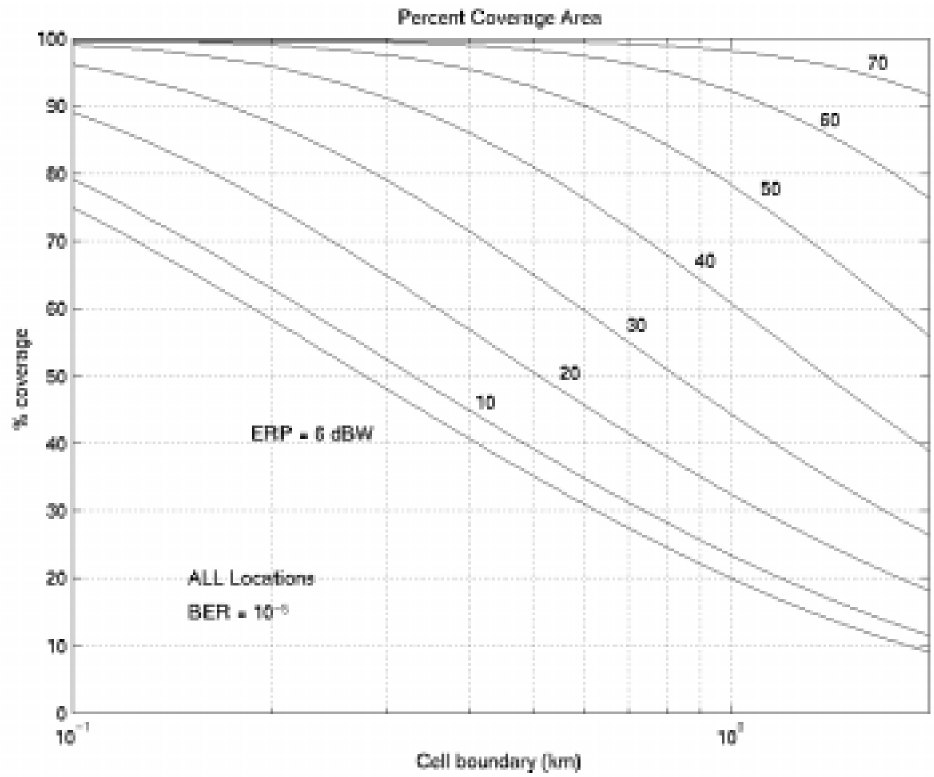

Figure 94: Percent coverage area over distance for various values of ERP using ALL locations and BER = 10^{-5}.

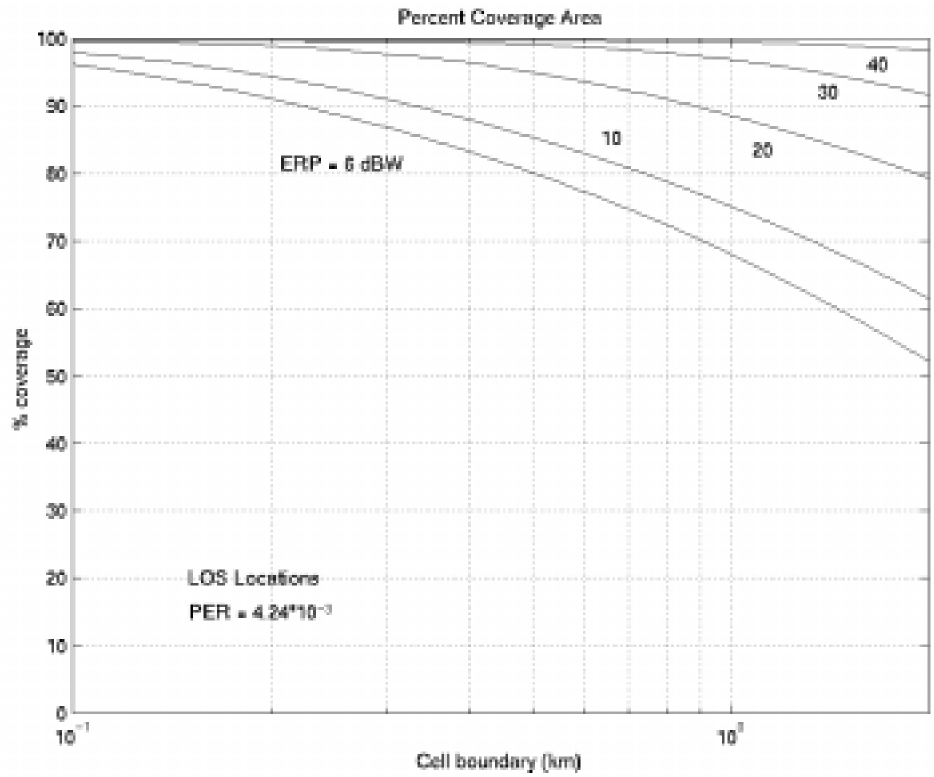

Figure 95: Percent coverage area over distance for various values of ERP using LOS locations and PER = $4.24*10^{-3}$ with packet size = 424 bits (ATM cell).

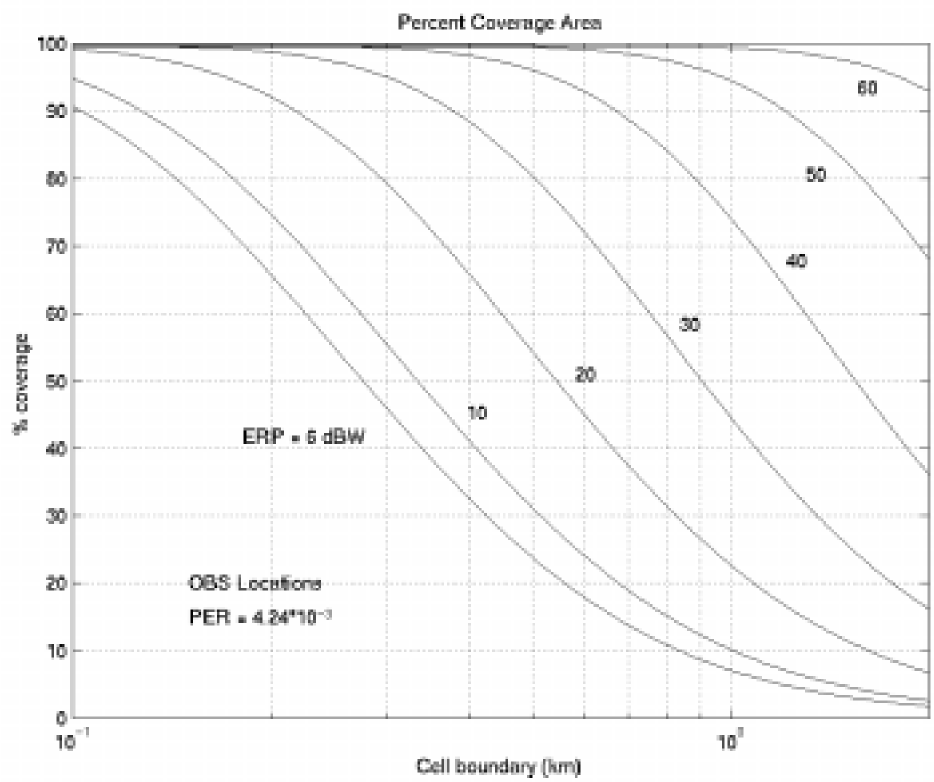

Figure 96: Percent coverage area over distance for various values of ERP using OBS locations and PER = $4.24*10^{-3}$ with packet size = 424 bits (ATM cell).

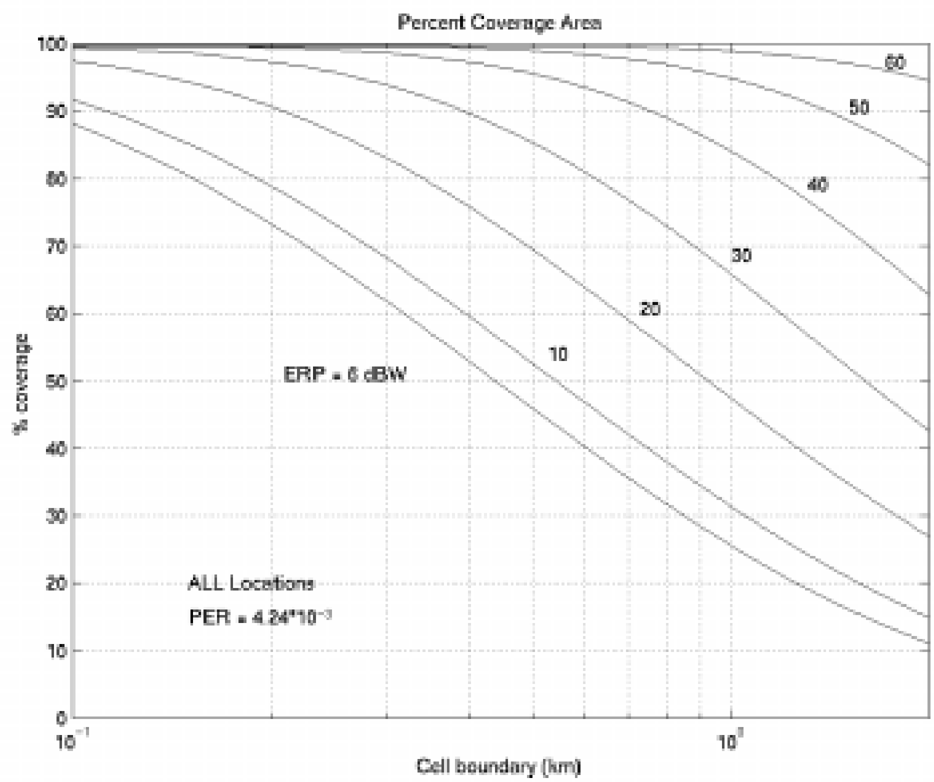

Figure 97: Percent coverage area over distance for various values of ERP using LOS locations and PER = $4.24*10^{-3}$ with packet size = 424 bits (ATM cell).

4.10 Percent Coverage Area: LOS, OBS, and ALL Case Comparison

Figures 99-108 compare the percent coverage area obtained using each location case LOS, OBS and ALL and for BER = 10^{-5} and the equivalent PER for one ATM cell (424 bits) = $4.24 \cdot 10^{-3}$. It can be seen that the difference in percent coverage can be vastly different between the LOS and OBS cases especially when the ERP is low. This is true for both the BER and PER approaches. At 1 km for an ERP = 6 dBW, the percent coverage area difference between LOS and OBS locations is 53 percent for the BER approach and 60 percent for the PER approach.

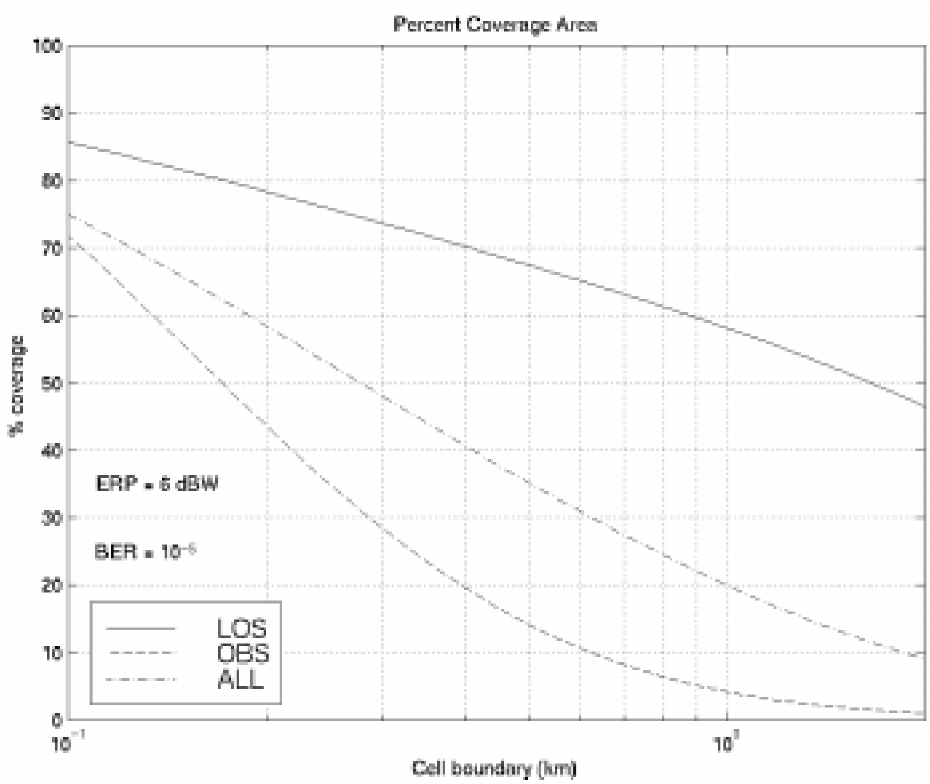

Figure 98: Percent coverage area for LOS, OBS and ALL locations with BER = 10^{-5} and ERP = 6 dBW.

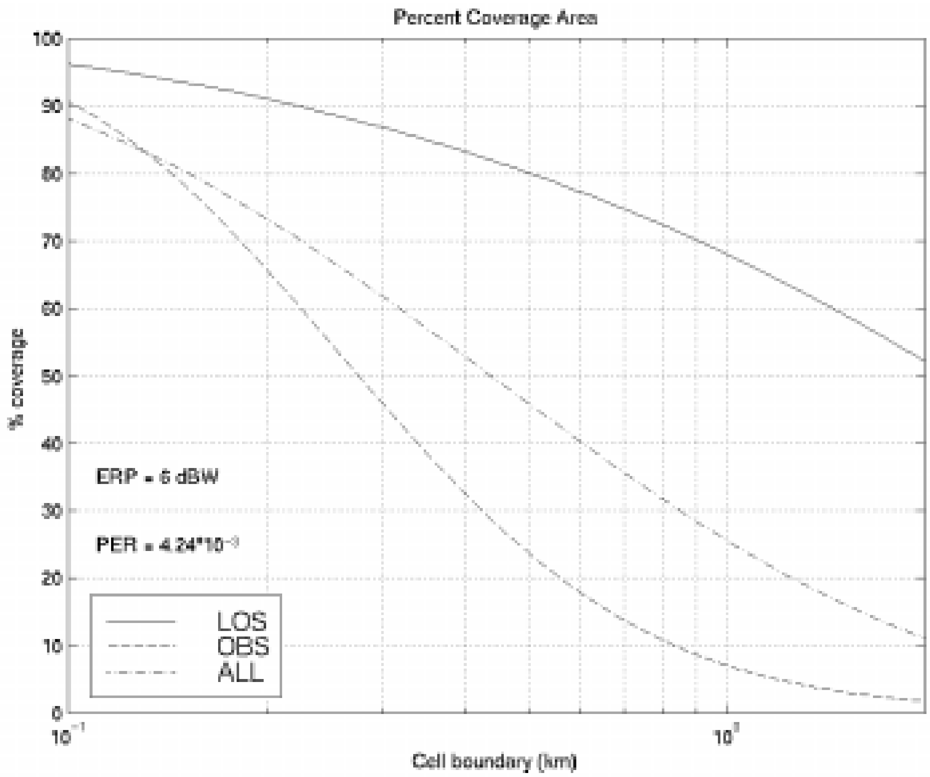

Figure 99: Percent coverage area for LOS, OBS and ALL locations with PER = 0.00424 and ERP = 6 dBW.

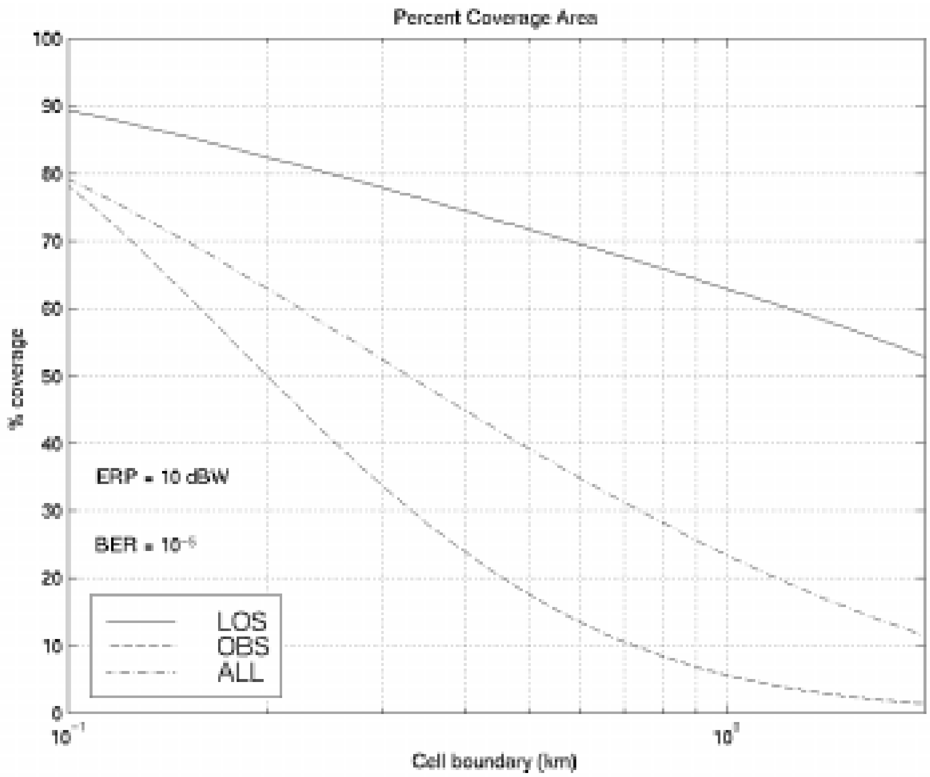

Figure 100: Percent coverage area for LOS, OBS and ALL locations with BER = 10^{-5} and ERP = 10 dBW.

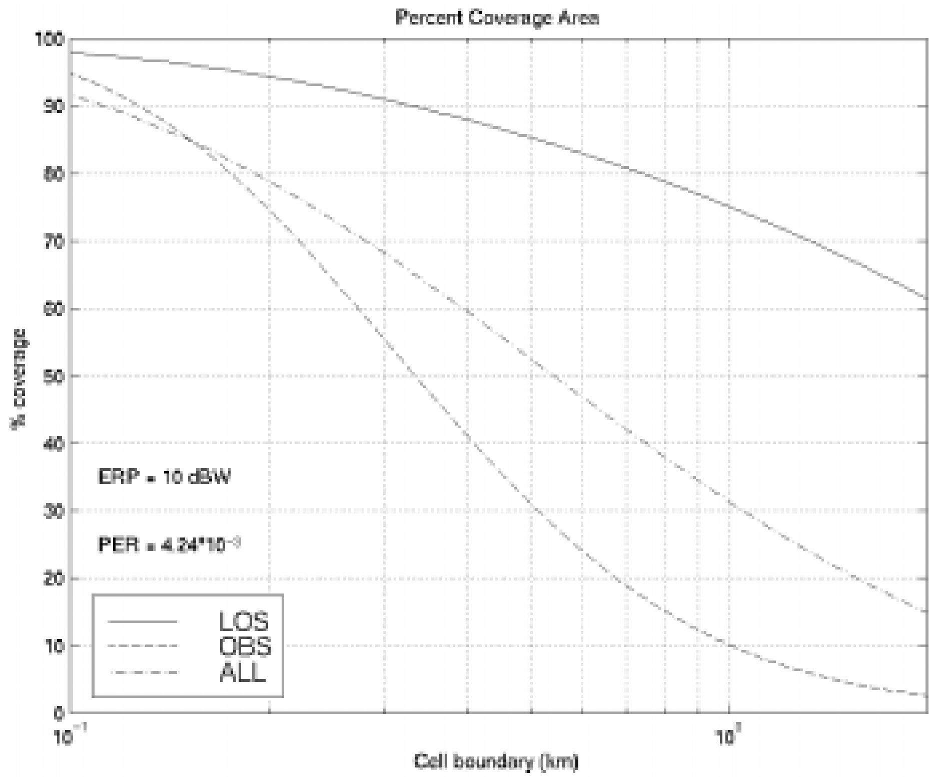

Figure 101: Percent coverage area for LOS, OBS and ALL locations with PER = 0.00424 and ERP = 10 dBW.

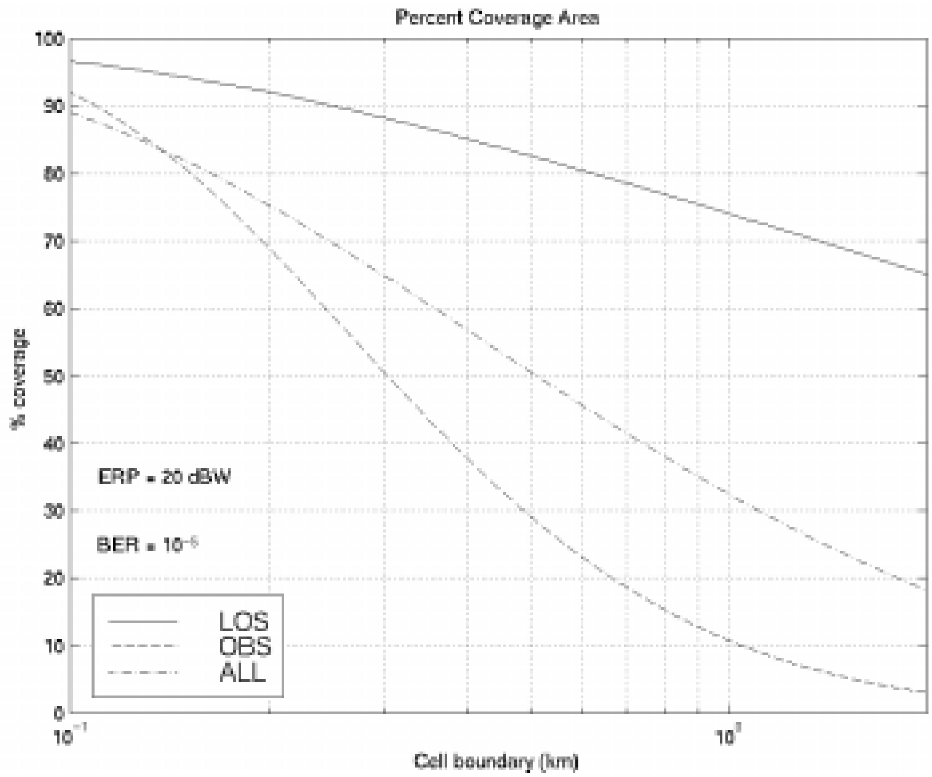

Figure 102: Percent coverage area for LOS, OBS and ALL locations with BER = 10^{-5} and ERP = 20 dBW.

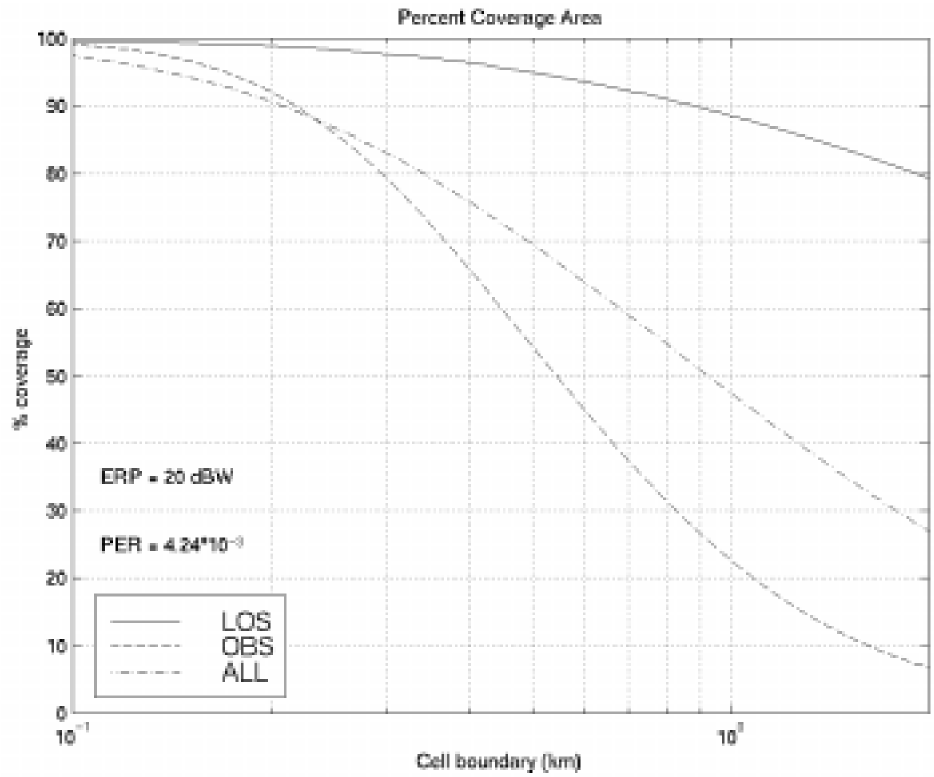

Figure 103: Percent coverage area for LOS, OBS and ALL locations with PER = 0.00424 and ERP = 20 dBW.

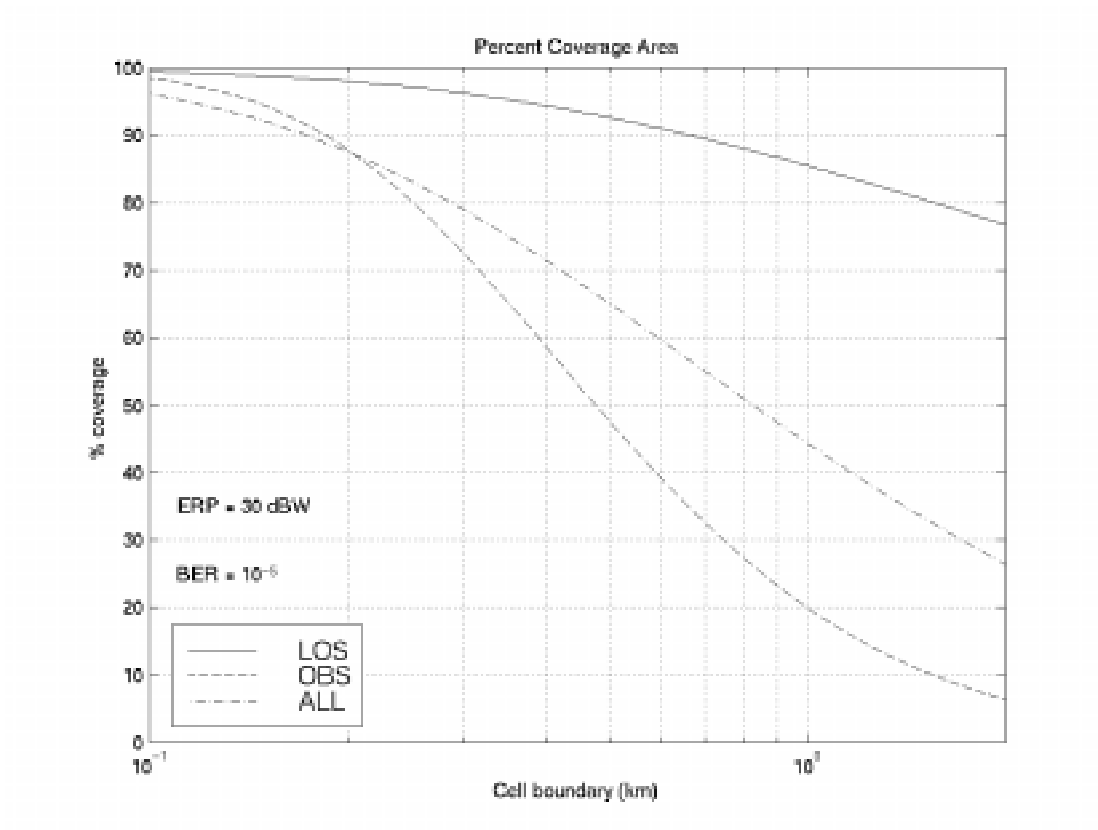

Figure 104: Percent coverage area for LOS, OBS and ALL locations with BER = 10^{-5} and ERP = 30 dBW.

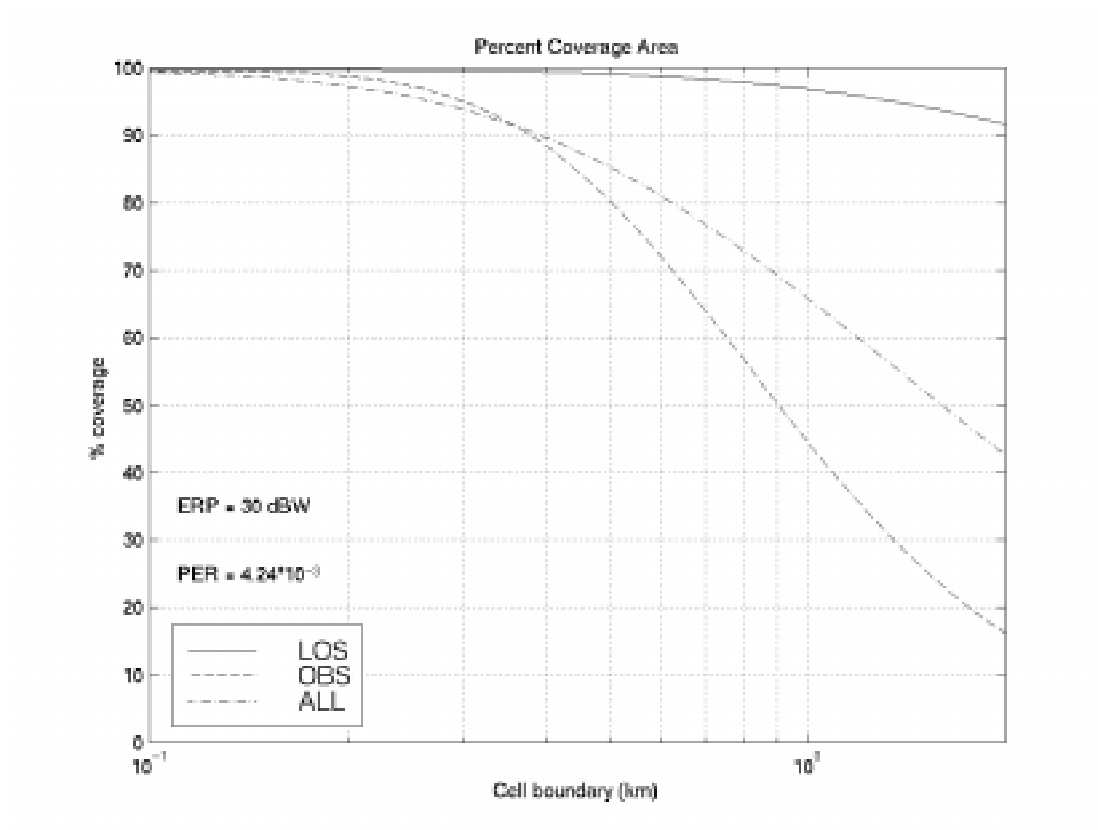

Figure 105: Percent coverage area for LOS, OBS and ALL locations with PER = 0.00424 and ERP = 30 dBW.

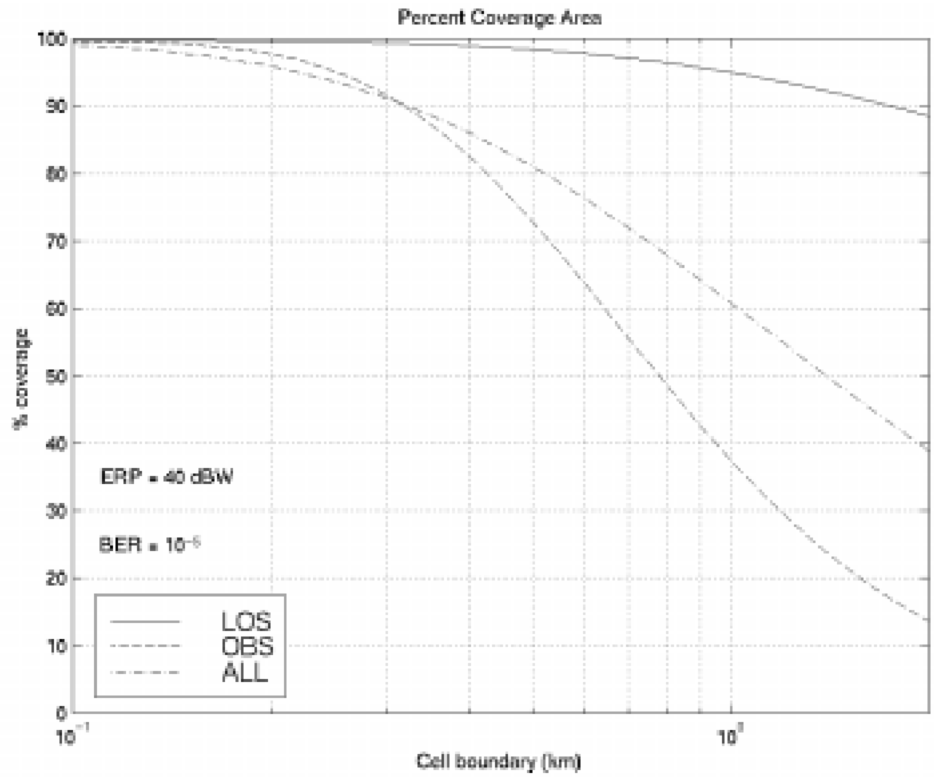

Figure 106: Percent coverage area for LOS, OBS and ALL locations with BER = 10^{-5} and ERP = 40 dBW.

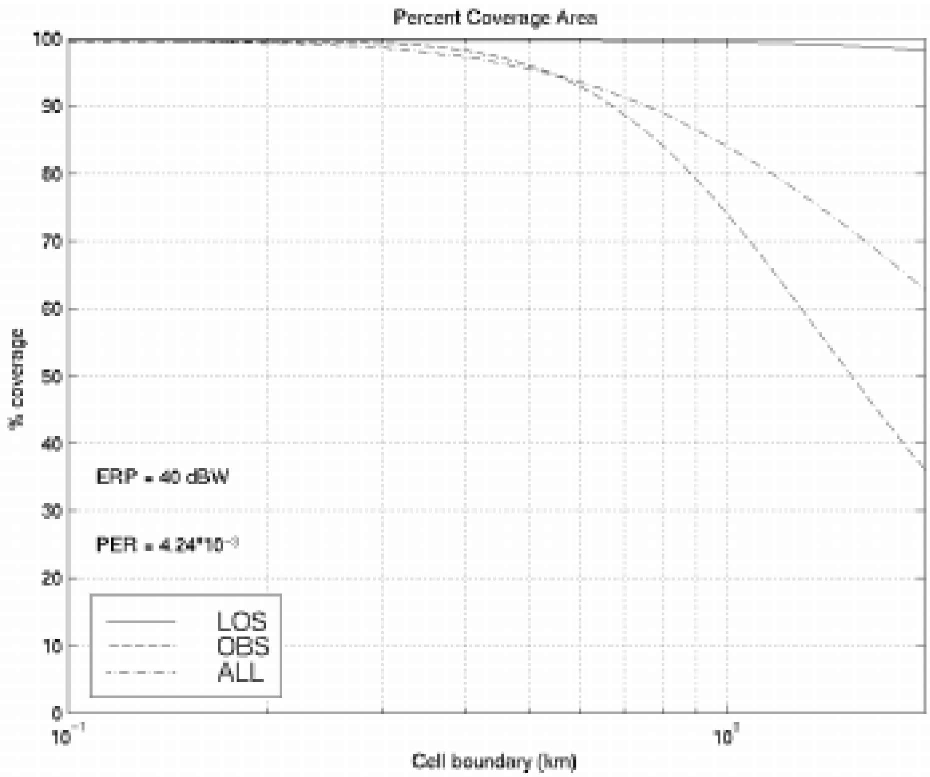

Figure 107: Percent coverage area for LOS, OBS and ALL locations with PER = 0.00424 and ERP = 40 dBW.

5 Conclusions

The use of the log-normal distribution as a model for the variation in the Rician parameter, K, is a significant result of this investigation. Although others have shown experimentally that the distribution of K may be log-normal [11], no theoretical basis for this distribution has been postulated. In addition, the other investigators found no practical use for modeling K as a log-normal random variable. The results show in this investigation that transmit power can be saved by using this information about K when the specification of interest is the percent coverage area.

The results obtained from considering the Rician parameter, K, show that K is log-normal about a distance dependent mean. It was shown that the average value of K decreases with distance, which is consistent with the findings of investigators who determined that Rayleigh fading was a good theoretical model for large transmitter-receiver separations. For the LOS case, the received signal was nearly specular, which would be expected since the definition of LOS used in this investigation was that a direct path existed between the transmitter and receiver. The OBS location case and the ALL location case show a rapid descent towards Rayleigh fading as the transmitter-receiver separation increases. This decrease in K as distance increases can be accounted for by noting that as the separation between the receiver and transmitter increases, there is more of a chance that something in the channel between them will change causing the signal level at the receiver to change as well.

There is nothing surprising about the average path loss results when the power law model was used. As with the Rician parameter, K, the average receive power was shown log-normal about a distance dependent mean. The separation of the locations into LOS and OBS locations show the path loss exponent to be vastly different for each case, but still reasonable considering the nature of the two groups. The interesting result concerns the use of the two-ray model to describe the path loss. Although the OBS and ALL cases do not match the two-ray model as well as the power law, the LOS case does. This is interesting because the two-ray model is normally considered as an introduction to the concept of multipath because the geometry is tractable. The results for LOS show that when designing an LOS point-to-point link, a more accurate path loss estimation can be made by using the two-ray model as opposed to the power law model.

A strong correlation was shown to exist between the Rician parameter, K, and receive power. This should be expected since the two are functions of the same quantities. It is then left to determine whether this correlation should be taken into account when determining the percent coverage for an area. The results show that the differences between the correlated and uncorrelated cases are no more than 10 percent over the cell radius range under investigation. Assuming K and shadowing are independent allows for a less complex percent coverage area calculation. It was found, however, that the program used to calculate the results did not run significantly faster for the independent case than the correlated case. Since K and receive power were found to be correlated and the analysis does not take much longer to accomplish using this information, it is recommended that when calculating the percent coverage for an area the joint pdf of the two random variables used should be a bivariate Gaussian with a non-zero correlation coefficient. What is also clear from the results is that assuming the entire cell to be a Rayleigh fading channel will greatly underestimate the percent coverage area.

The BER and PER results show about a 10 percent gain in percent coverage area for an ATM cell (424 bits) over the equivalent BER. This should be expected since the errors that occur in a fading channel tend to be bursty rather than random. Packetizing the data takes advantage of the bursty nature of the errors, thus increasing the percent coverage area.

The results showing the percent coverage area for different values of ERP show that to cover a significant area, a great deal of power is needed. Even for the LOS case, to cover 95 percent of the area of a circle with a radius of 1 km, 40 dBW is needed. This is well above the FCC limitation of 6 dBW. The major reason for this is the nature of the path loss at 5.8 GHz. In free space, path loss increases as the square of the frequency. The 5.8 GHz band is such a high frequency that the path loss incurred just from using this frequency is enough to make it difficult to close the link over any significant distance. To combat this enormous path loss, it will be necessary to use techniques, such as diversity combing and directional antennas, that will increase the performance of the system.

6 References

[1] T. S. Rappaport. *Wireless Communications: Principles and Practice* Prentiss Hall, Upper Saddle River, New Jersey, 1996.

[2] Richard L. Schaefer and James T. McClave. *Probability and Statistics for Engineers* Duxbury Text, Belmont, California, 1995.

[3] Milton Abramowitz and Irene Stegun. *Handbook of Mathematical Functions* Dover Publishing, New York, 1972.

[4] Erwin Kreyszig. *Advanced Engineering Mathematics 6th ed.* John Wiley & Sons, New York, 1988.

[5] M. A. Stevens, "EDF Statistics for Goodness of Fit and Some Comparisons," *Journal of the American Statistical Association*, vol 69, (September 1974), 730-737.

[6] D. C. Cox, R.R. Murray, A.W. Norris, "800 MHz Attenuation Measured In and Around Suburban Houses," *AT&T Bell Laboratories Technical Journal*, vol 63, No. 6, (July-August 1984), 921-954.

[7] Richard C. Bernhardt, "Macroscopic Diversity in Frequency Re-Use Radio Systems," *IEEE Journal on Selected Areas In Communications*, vol SAC-5, No. 5, (June 1987), 862-870.

[8] R. W. E. McNicol, "The Fading of Radio Waves of Medium and High Frequencies," *Data Communications Via Fading Channels*, ed. Kenneth Brayer, IEEE Press, New York, 1975.

[9] Vijay K. Garg and Joseph E. Wilkes. *Wireless and Personal Communications Systems,* Prentiss Hall PTR, Upper Saddle River, New Jersey, 1996.

[10] David Parsons, "Characterisation of Fading Mobile Radio Channels," *Personal and Mobile Radio Systems*, ed. R.C.V. Macario, Peter Peregrinus Ltd., London, 1991.

[11] Raymond Steele ed., *Mobile Radio Communications*, IEEE Press, New York, 1994.

[12] S. O. Rice, "Mathematical Analysis of Random Noise," *Bell Systems Technical Journal*, vol 23, (July 1944), 282-332.

[13] Sherman K. Stein and Anthony Barcellos. *Calculus and Analytic Geometry 5th ed.,* McGraw-Hill, Inc, New York, 1992.

[14] John G. Proakis. *Digital Communications 3rd ed.,* McGraw-Hill, Inc, New York, 1995.

[15] I. S. Gradshteyn and I.M. Ryzhik, *Table of Integrals, Series, and Products*, Academic Press, New York, 1965.

[16] K. S. Shanmugan and A. M. Breipohl, *Random Signals: Detection, Estimation and Data Analysis,* John Wiley and Sons, New York, 1988.

[17] R. H. Clarke, "A Statistical Theory of Mobile Radio," *Bell Systems Technical Journal*, vol 47, No. 6, (July-August 1968), 957-1000.

[18] W. B. Davenport and W. L. Root, *An Introduction to the Theory of Random Signals and Noise,* IEEE Press, New York, 1987.

[19] Gary C. Hess, *Handbook of Land-Mobile Radio System Coverage,* Artech House Publishers, Boston, 1998.

[20] M. T. Marsan, et al, "Shadowing Variability in an Urban Land-Mobile Environment at 900MHz," *Electronic Letters,* vol 26, No. 10, (July-August 1990), 646-648.

[21] Douglas O. Reudink, "Properties of Mobile Radio Propagation Above 400 MHz," *IEEE Transactions on Vehicular Technology,* vol VT-23, No. 4, (November 1974), 143-159.

[22] Finn I. Meno, "Mobile radio Fading in Scandinavian," *IEEE Transactions on Vehicular Technology,* vol VT-26, No. 4, (November 1977), 335-340.

[23] Michael J. Gans, "A Power Spectral Density Theory of Propagation in the Mobile-Radio Environment," *IEEE Transactions on Vehicular Technology,* vol VT-21, No. 1, (February 1972), 27-38.

[24] Robert J. C. Bultitude and Keith Bedal, "Propagation Characteristics on Microcellular Urban Mobile Radio Channels at 900 MHz," *IEEE Journal on Selected Areas in Communications,* vol 7, No. 1, (January 1989), 31-39.

[25] George L. Turin et al, "A Statistical Model of Urban Multipath Propagation," *IEEE Transactions on Vehicular Technology,* vol VT-21, No. 1, (February 1972), 1-9.

[26] Donald C. Cox, "910 MHz Urban Mobile radio propagation: Multipath Characteristics in New York City," *IEEE Transactions on Communications,* vol COM-21, No. 11, (November 1973), 1188-1194.

[27] Robert J. C. Bultitude, "Measurement, Characterization and Modeling of Indoor 800/900 MHz Radio Channels for Digital Communications," *IEEE Communications Magazine,* vol 25, No. 6, (June 1987), 5-12.

[28] T. S. Rappaport, "Characterization of UHF Multipath Radio Channels in Factory Buildings," *IEEE Transactions on Antennas And Propagation,* vol 37, No. 8, (August 1989), 1058-1069.

[29] W. R. Young, "Comparison of Mobile Radio Transmission at 150, 450, 900, 3700 Mc," *Bell System Technical Journal,* vol 31, No. 6, (November 1952), 1068-1085.

[30] W. C. Jakes and D. O. Reudink, "Comparison of Mobile Radio Transmission at UHF and X Band," *IEEE Transactions on Vehicular Technology,* vol VT-16, (October 1967), 10-14.

[31] Gary C. Hess, *Land-Mobile Radio System Engineering,* Artech House Publishers, Boston, 1993.

[32] The Federal Communications Commission World Wide Web Page, "The FCC Makes Spectrum Available for New Unlicensed," ET Docket No. 96-102, NEWSReport No. DC 97-3NEWS, January 9, 1997, http://www.fcc.gov/Bureaus/Engineering_Technology/News_Releases/1997/nret7002.txt

[33] James A. Roberts, *Wireless Communication*, Course number EECS 865, University of Kansas, Class notes, 1997.

[34] Kaveh Pahlavan et al," Wideband Local Access Wireless LAN and Wireless ATM," IEEE Communications Magazine, vol 35, no 11, (November 1977), 34.

[35] D. M. Balston, "Cellular Radio Principles," *Cellular Radio Systems*, ed. R.C.V. Macario, Artech House, Inc., Boston, 1993.

[36] Asha Mehrota, *Cellular Radio Performance Engineering*, Artech House, Inc., Boston, 1994.

7 Appendix A: CDFs for Tested Locations

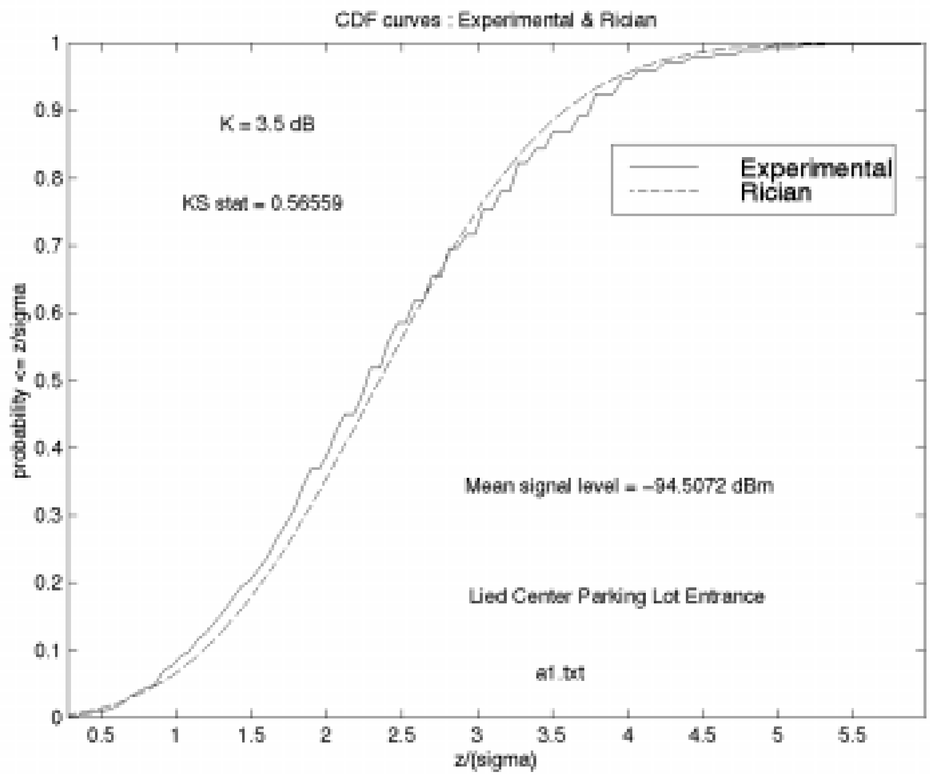

Figure 108: CDF for Lied Center parking lot entrance (a1) and the theoretical Rician CDF.

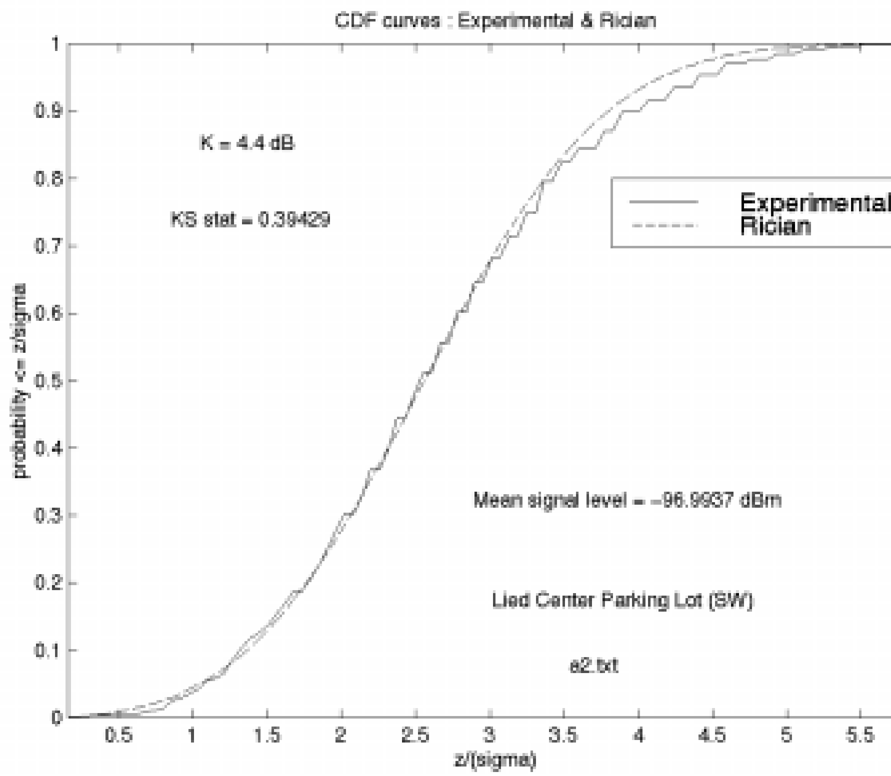

Figure 110: CDF for Lied Center SW parking lot (a2) and the theoretical Rician CDF.

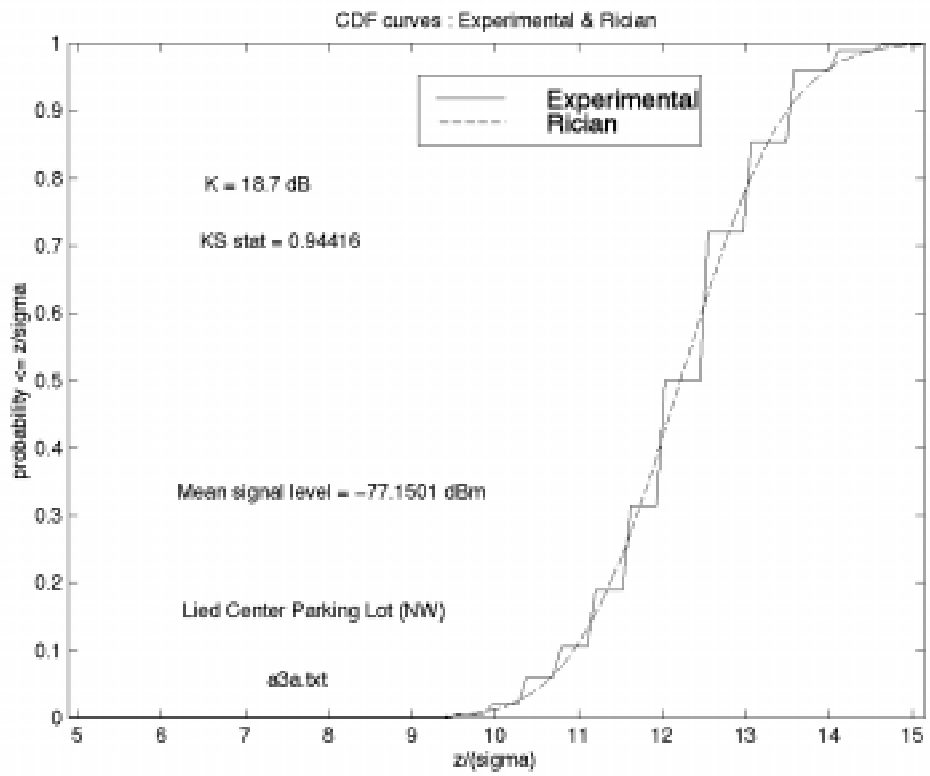

Figure 110: CDF for Lied Center NW parking lot (a3a) and the theoretical Rician CDF.

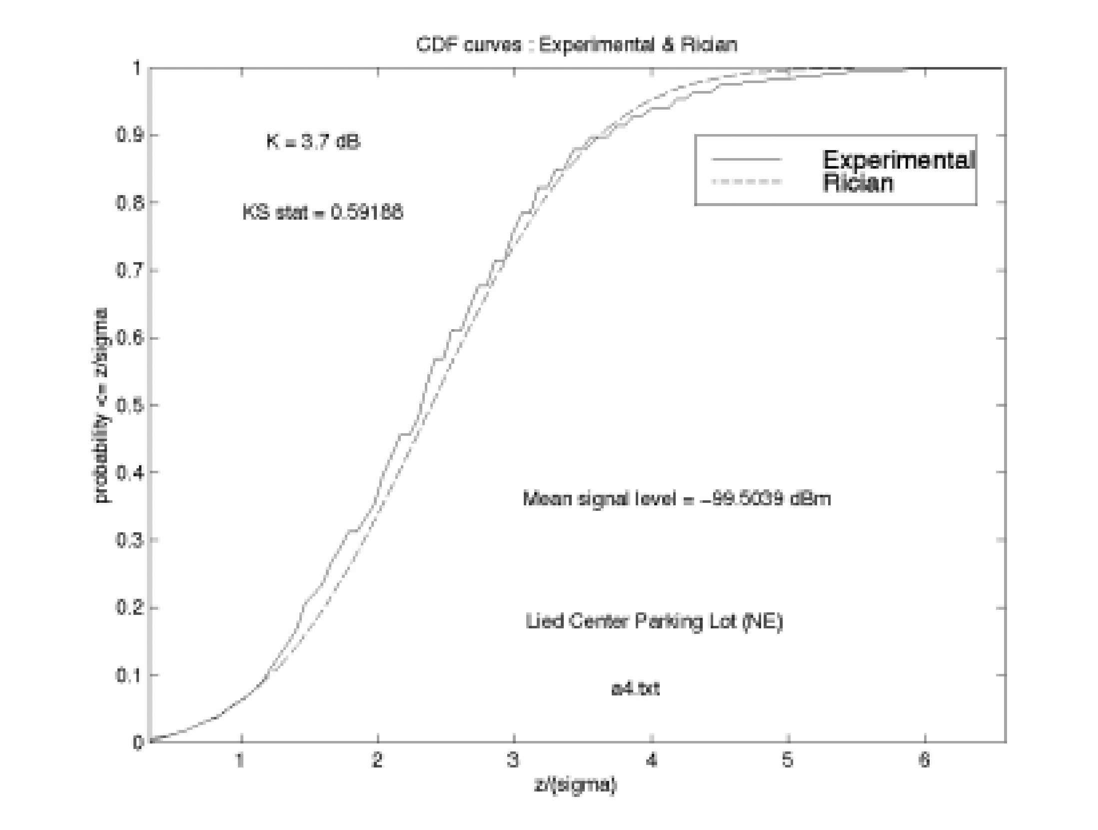

Figure 111: CDF for Lied Center NE parking lot (a4) and the theoretical Rician CDF.

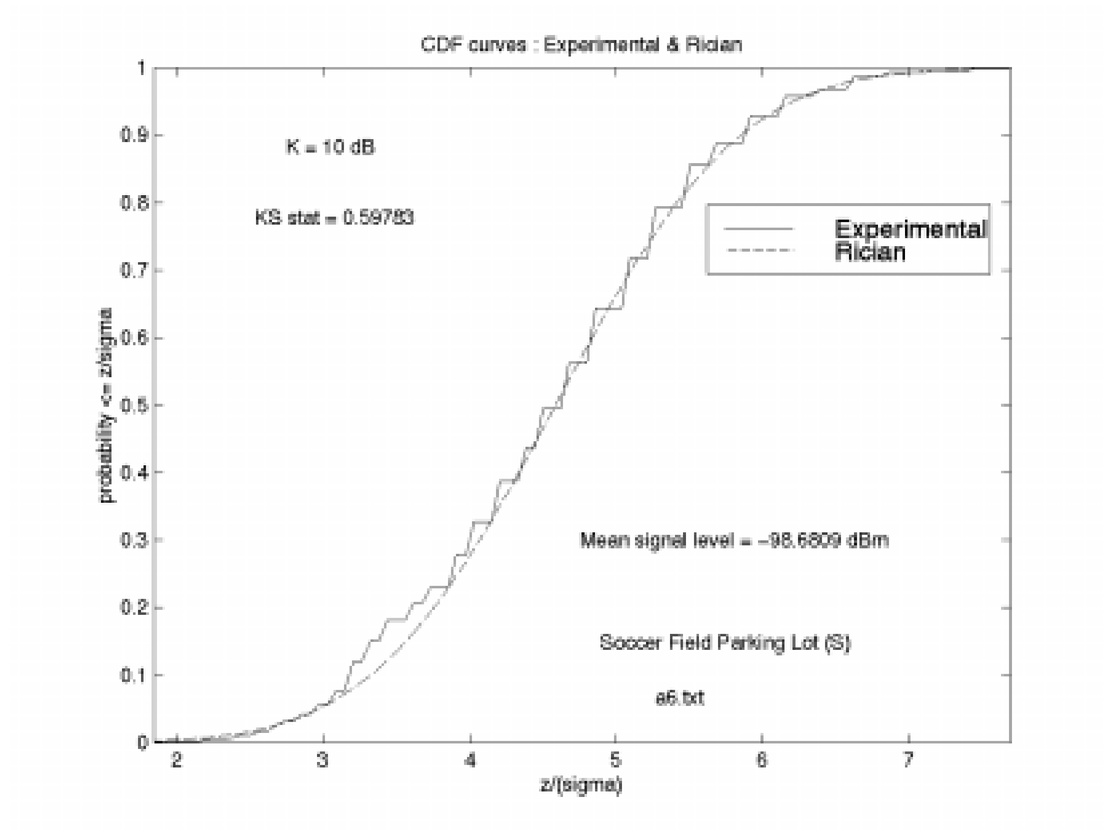

Figure 112: CDF for soccer field S parking lot (a6) and the theoretical Rician CDF.

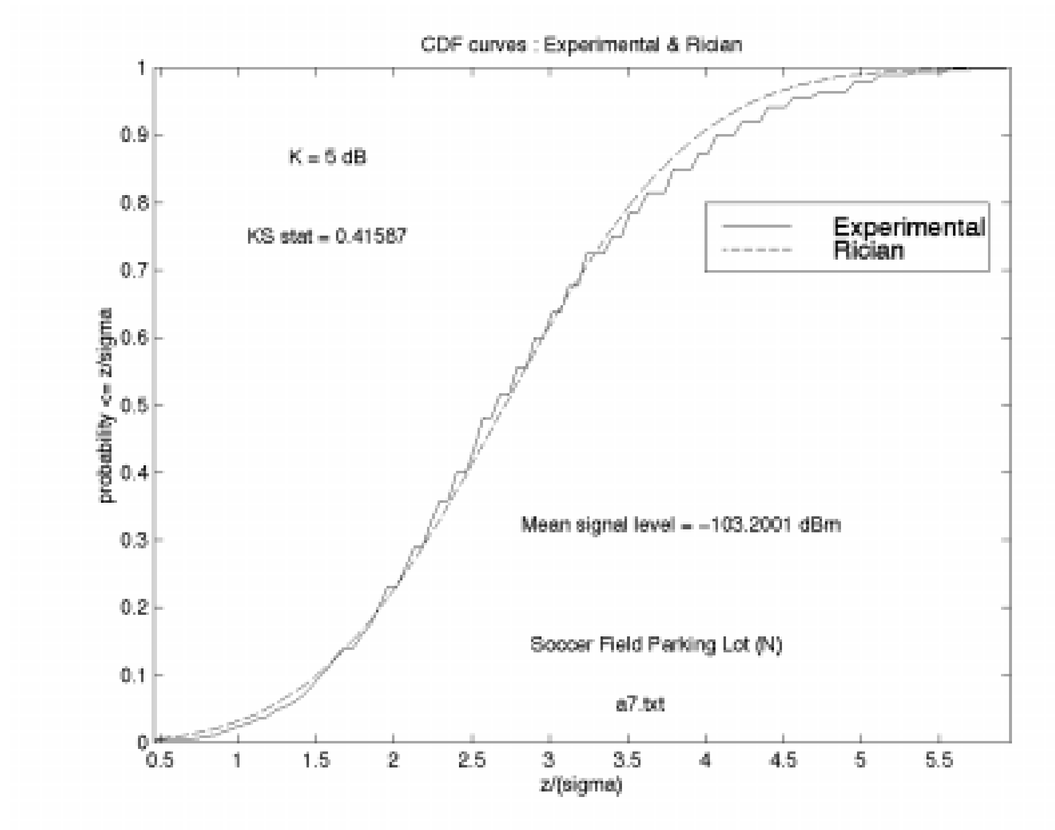

Figure 113: CDF for soccer field N parking lot (a7) and the theoretical Rician CDF.

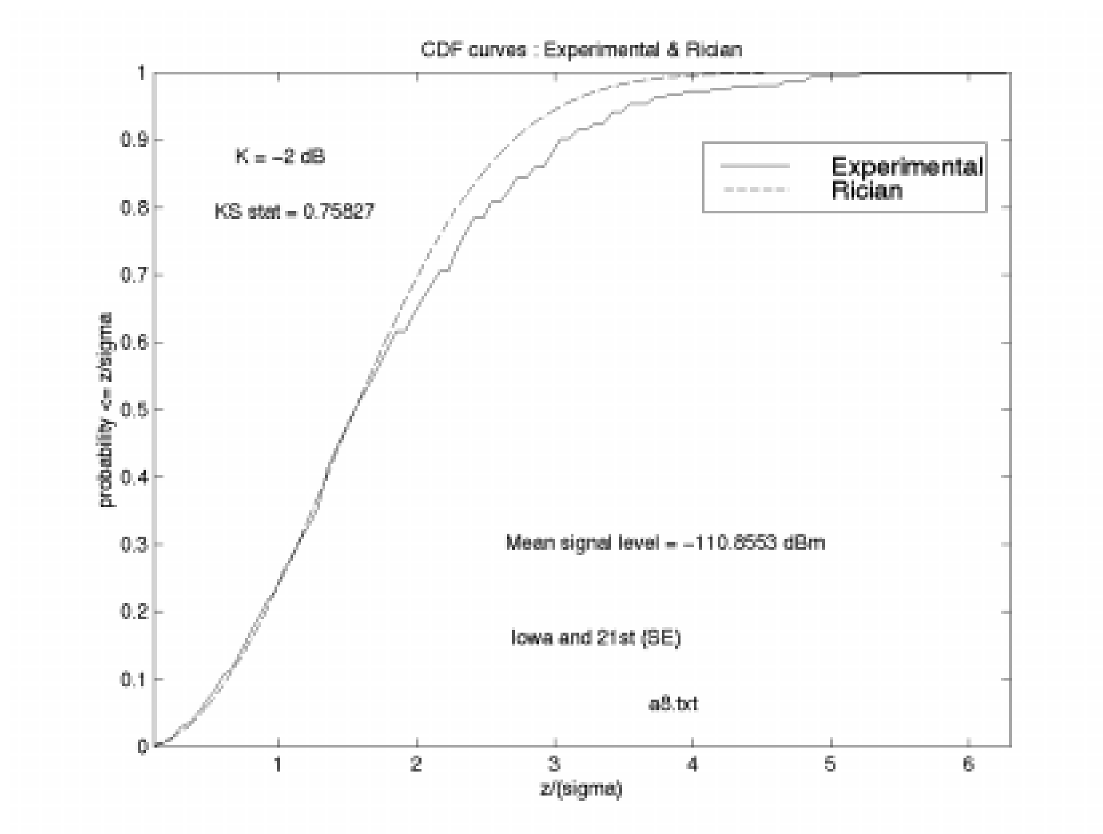

Figure 114: CDF for SE corner Iowa and 21st (a8) and the theoretical Rician CDF.

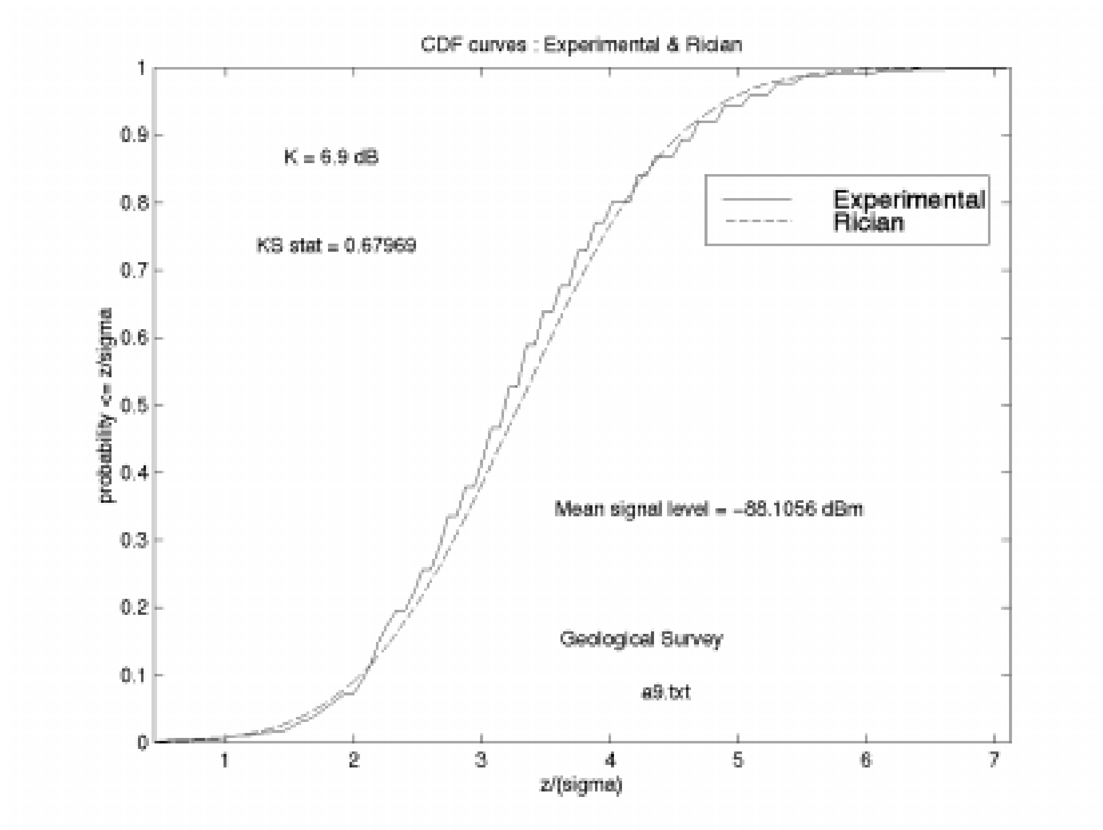

Figure 115: CDF for Geological Survey building (a9) and the theoretical Rician CDF.

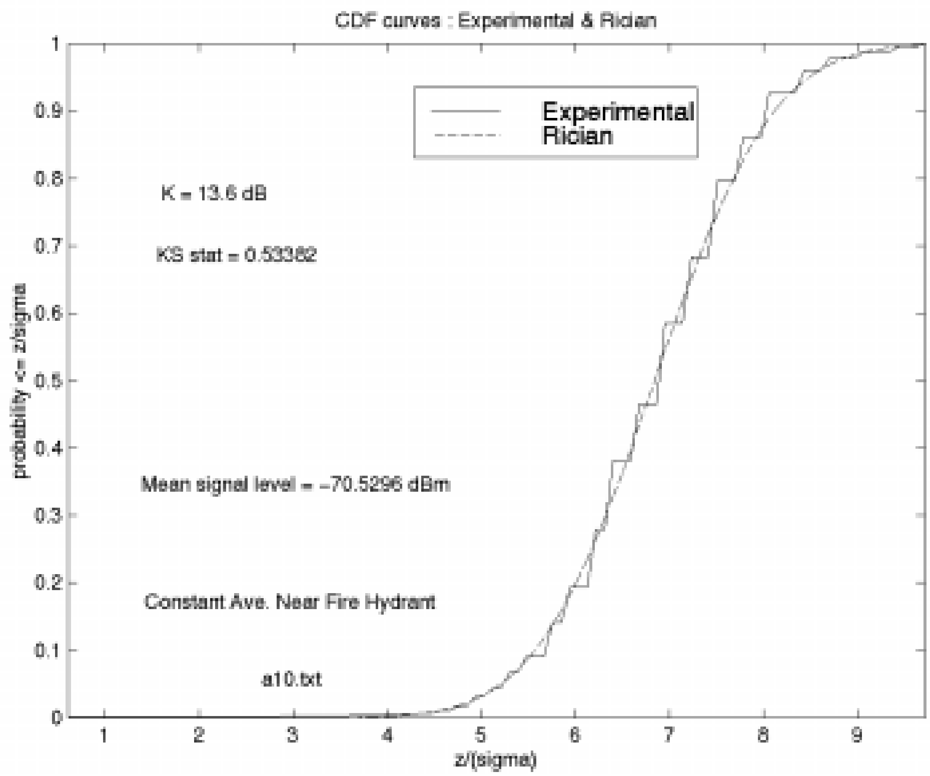

Figure 116: CDF for Constant Ave. near fire hydrant (a10) and the theoretical Rician CDF.

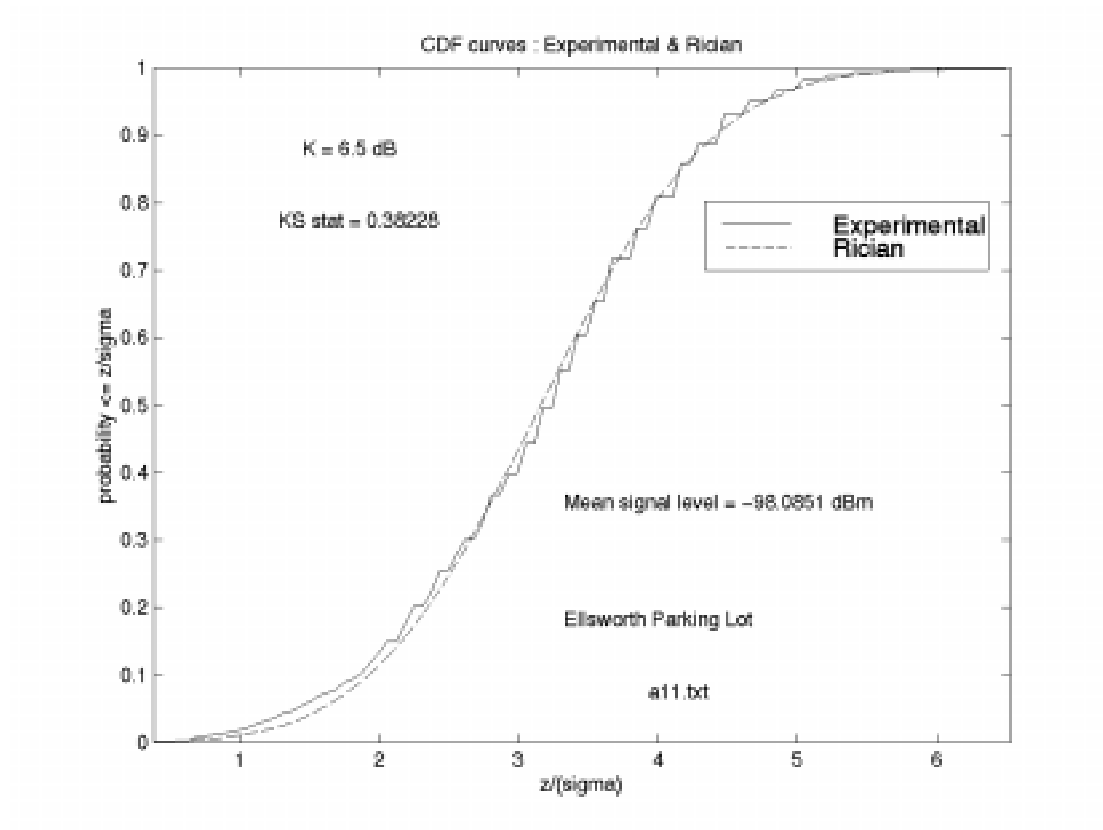

Figure 117: CDF for Ellsworth dormitory parking lot (a11) and the theoretical Rician CDF.

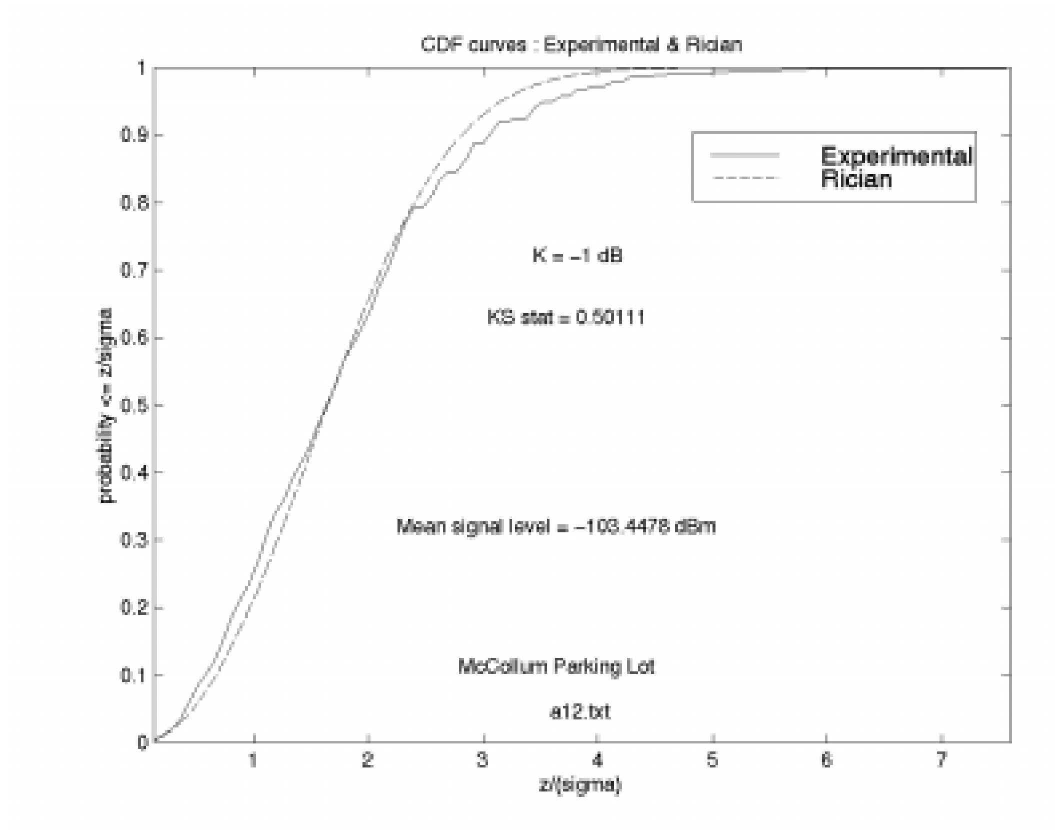

Figure 118: CDF for McCollum dormitory parking lot (a12) and the theoretical Rician CDF.

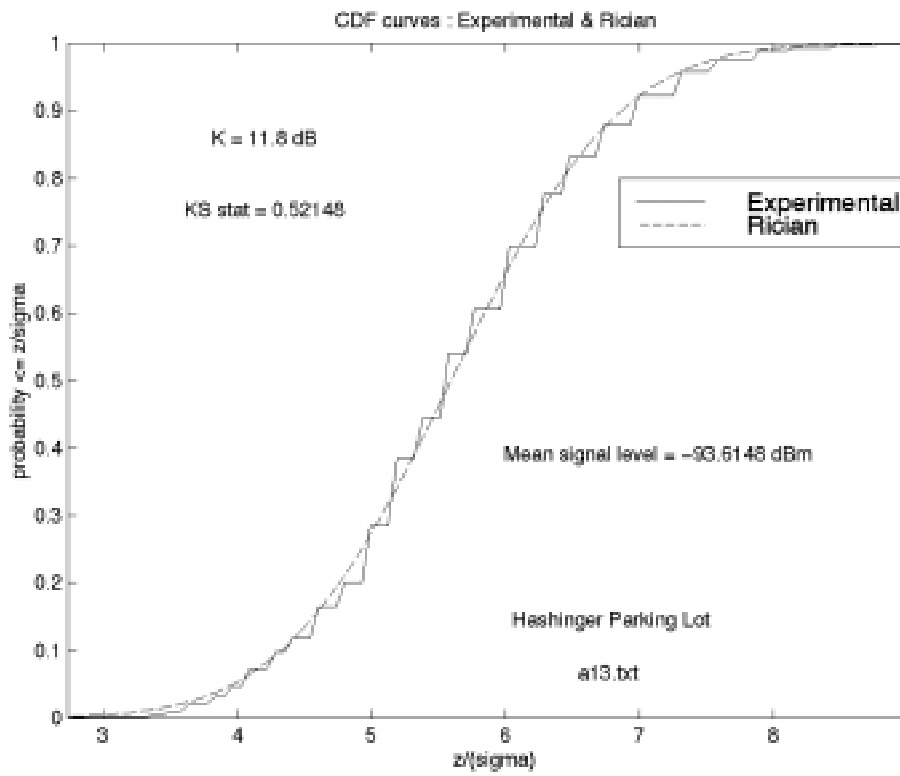

Figure 119: CDF for Hashinger dormitory parking lot (a13) and the theoretical Rician CDF.

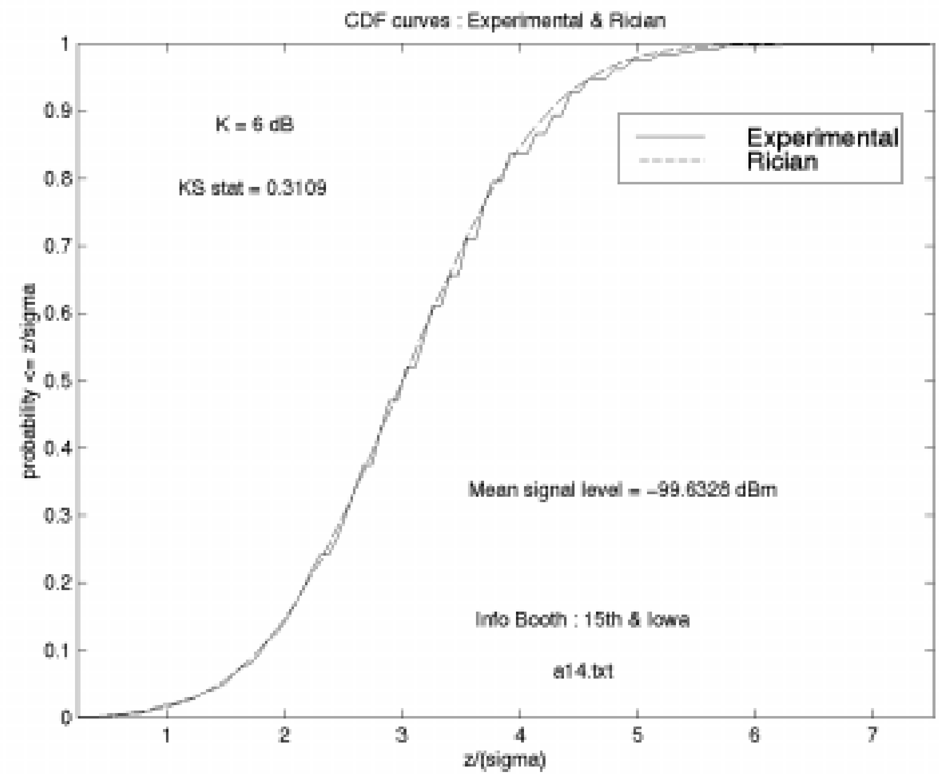

Figure 120: CDF for Information Booth on Iowa and 15th (a14) and the theoretical Rician CDF.

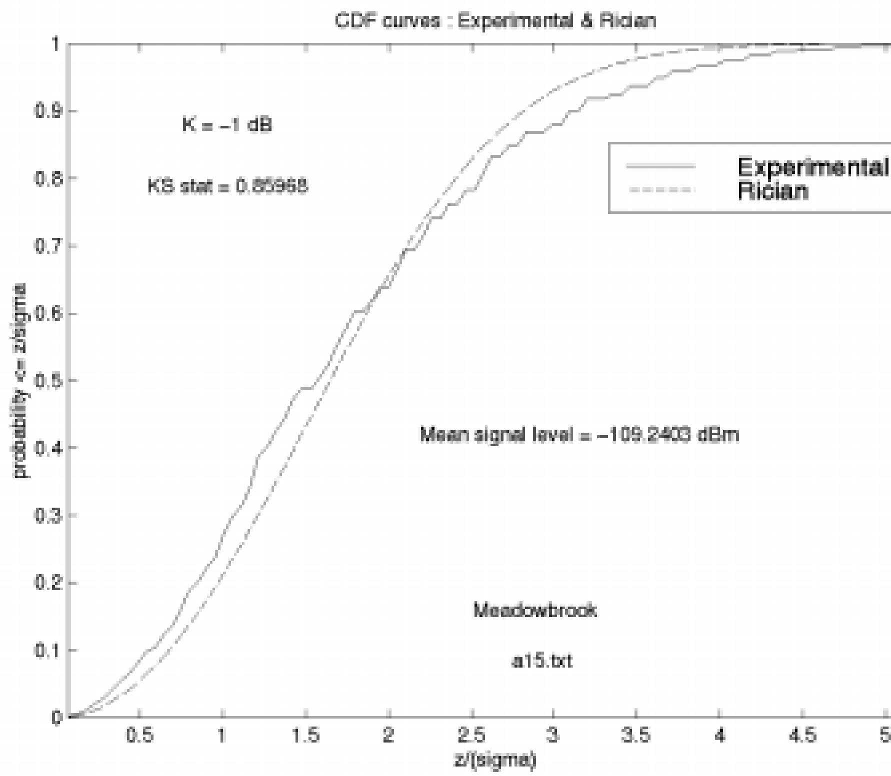

Figure 121: CDF for Meadowbrook Apartments (a15) and the theoretical Rician CDF.

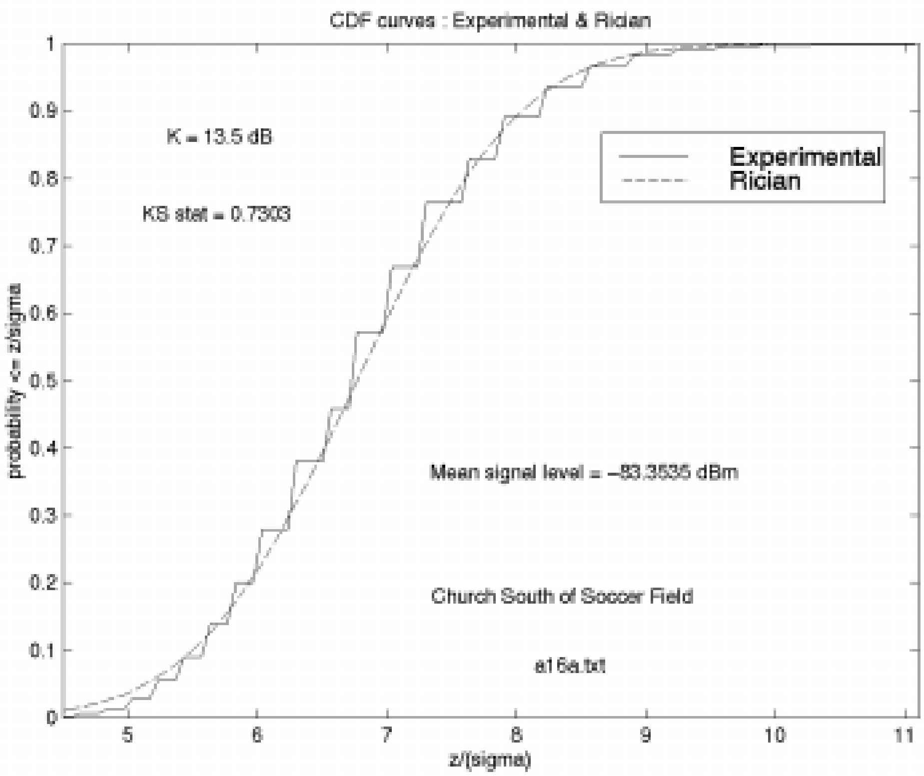

Figure 122: CDF for church south of soccer field (a16a) and the theoretical Rician CDF.

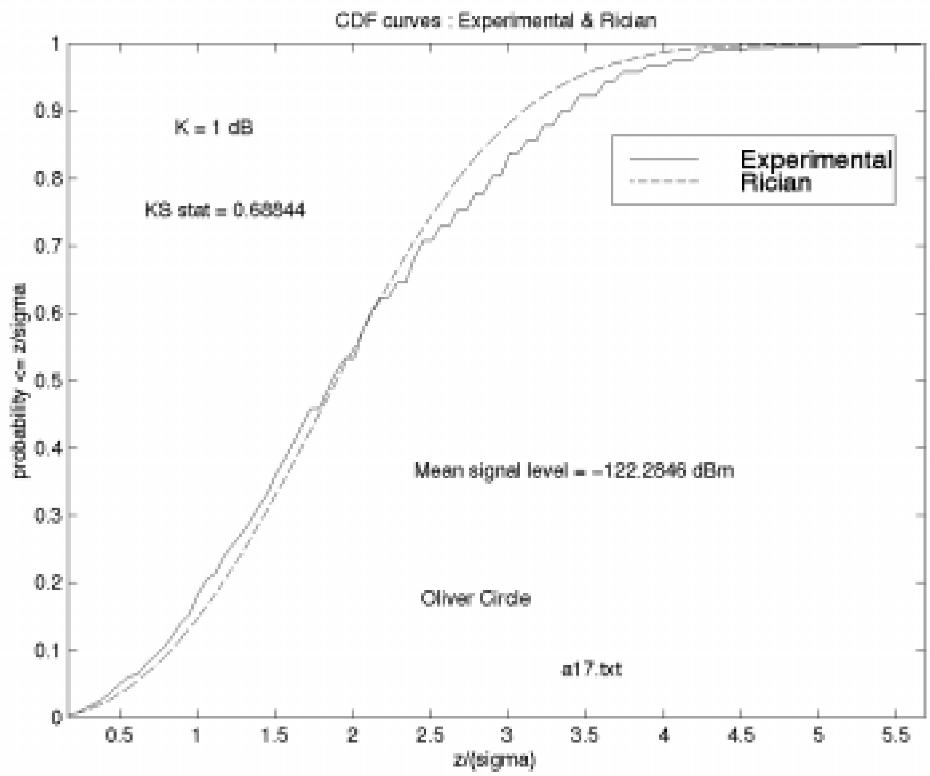

Figure 123: CDF for Oliver dormitory circle (a17) and the theoretical Rician CDF.

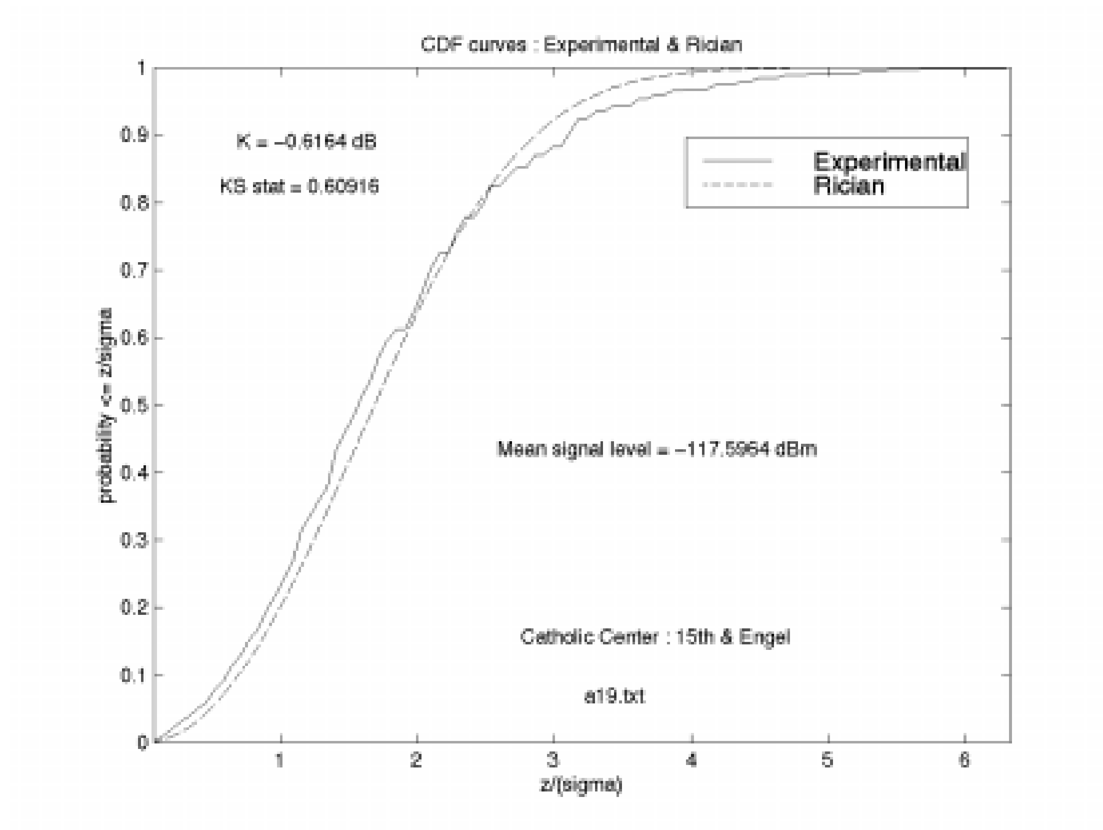

Figure 124: CDF for Catholic Center parking lot (a19) and the theoretical Rician CDF.

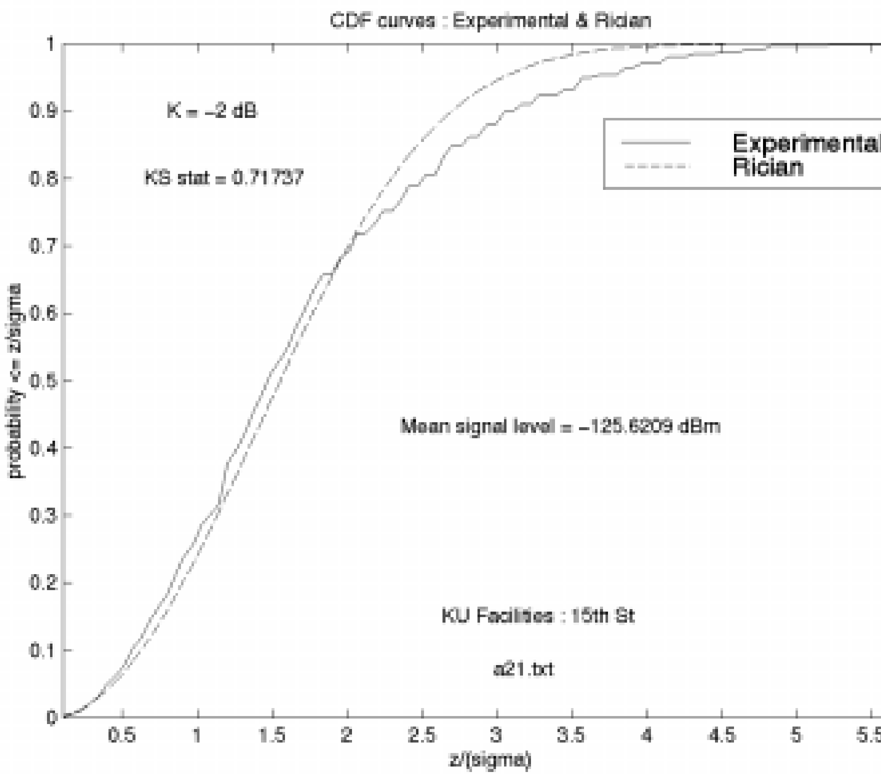

Figure 125: CDF for KU Facilities: 15th St (a21) and the theoretical Rician CDF.

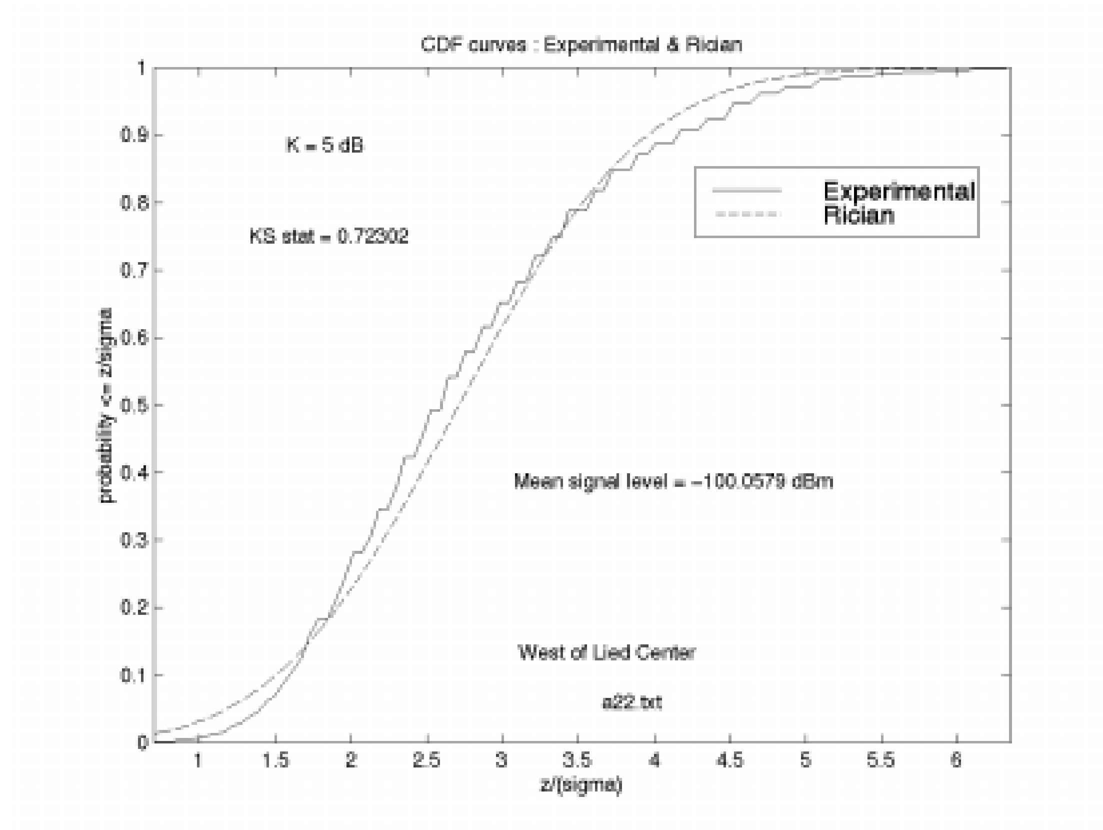

Figure 126: CDF for dirt road west of Lied Center (a22) and the theoretical Rician CDF.

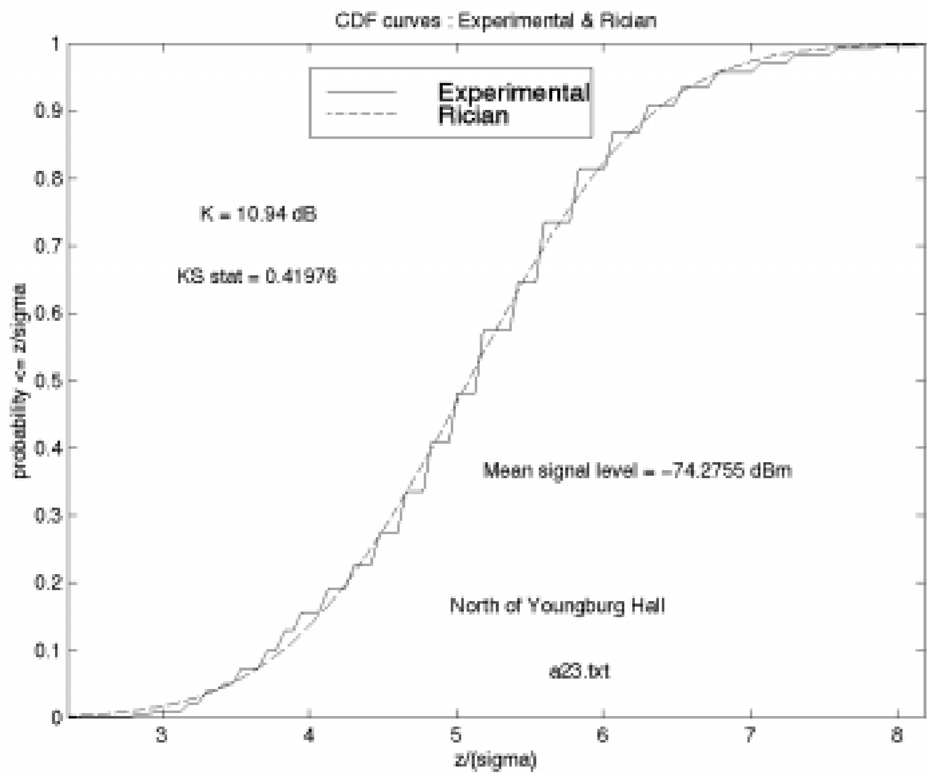

Figure 127: CDF for Youngberg Hall (a23) and the theoretical Rician CDF.

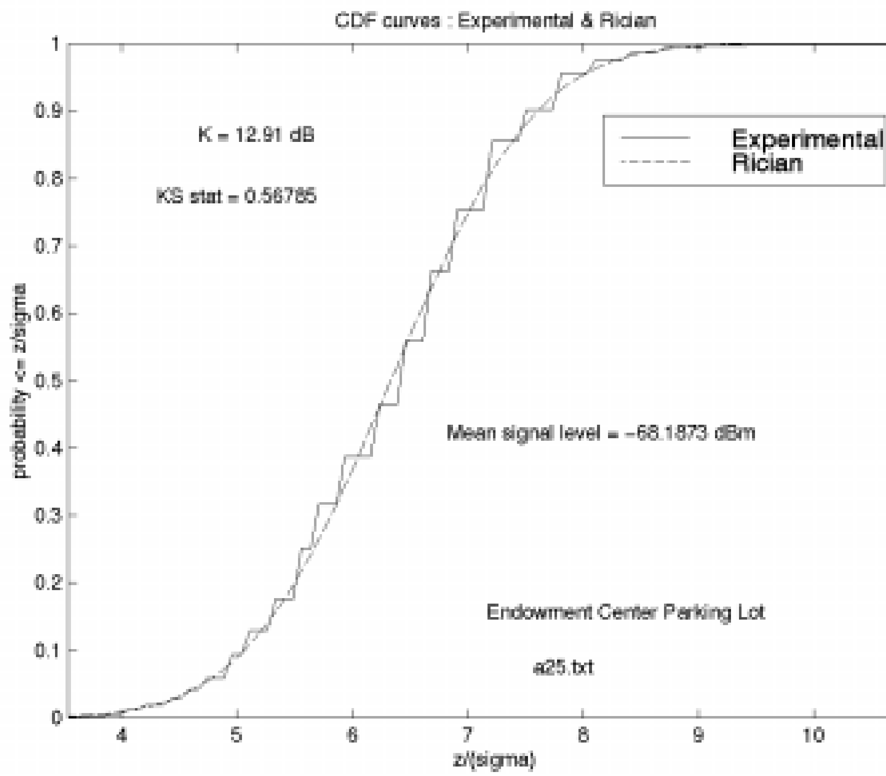

Figure 128: CDF for Endowment Center parking lot (a25) and the theoretical Rician CDF.

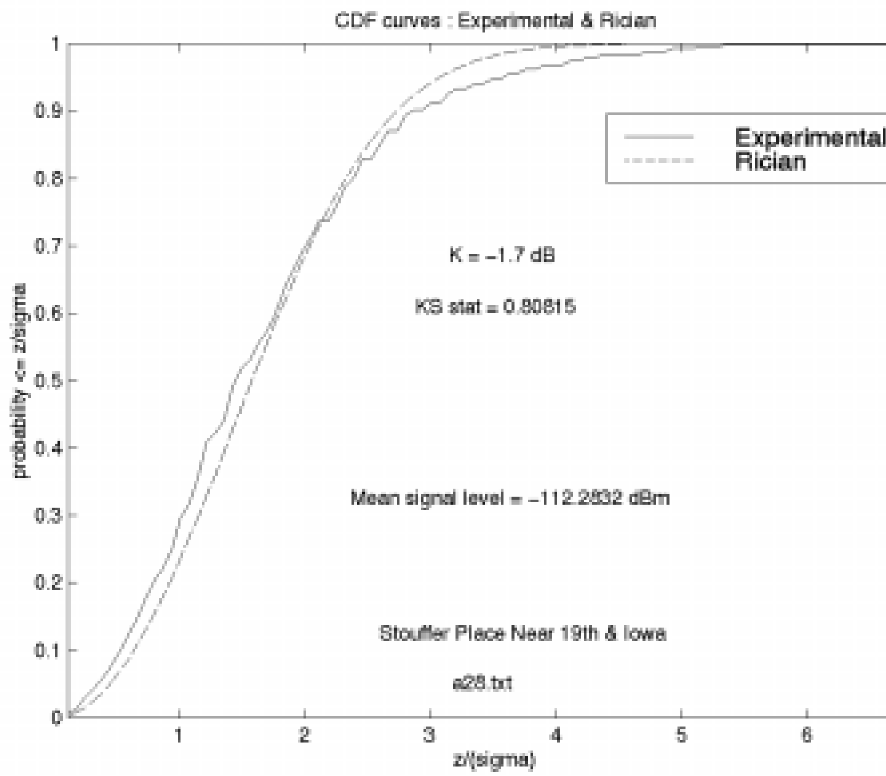

Figure 129: CDF for Stouffer Place near 19th and Iowa (a28) and the theoretical Rician CDF.

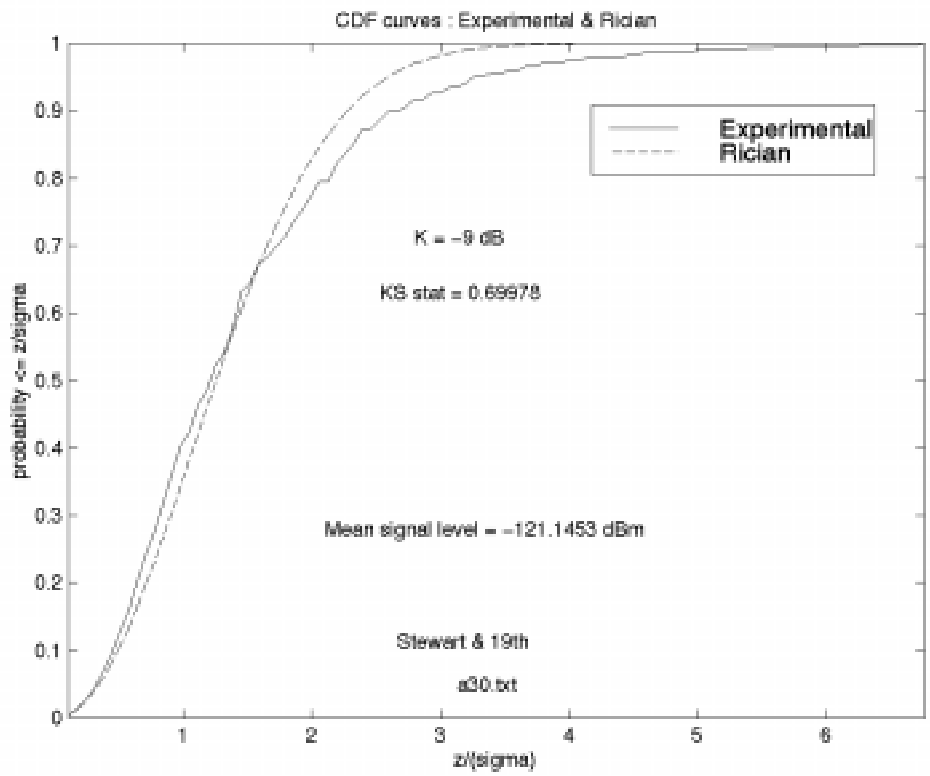

Figure 130: CDF for Stewart and 19th (a30) and the theoretical Rician CDF.

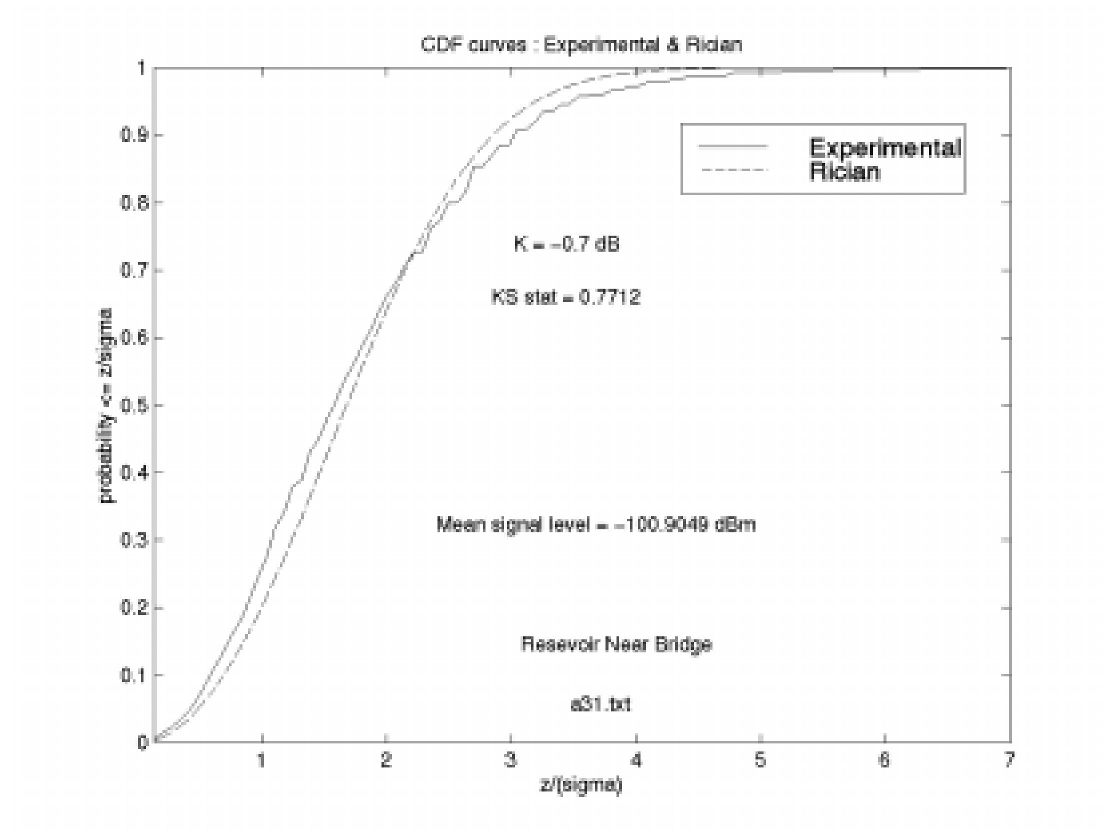

Figure 131: CDF for resevoir near bridge (a31) and the theoretical Rician CDF.

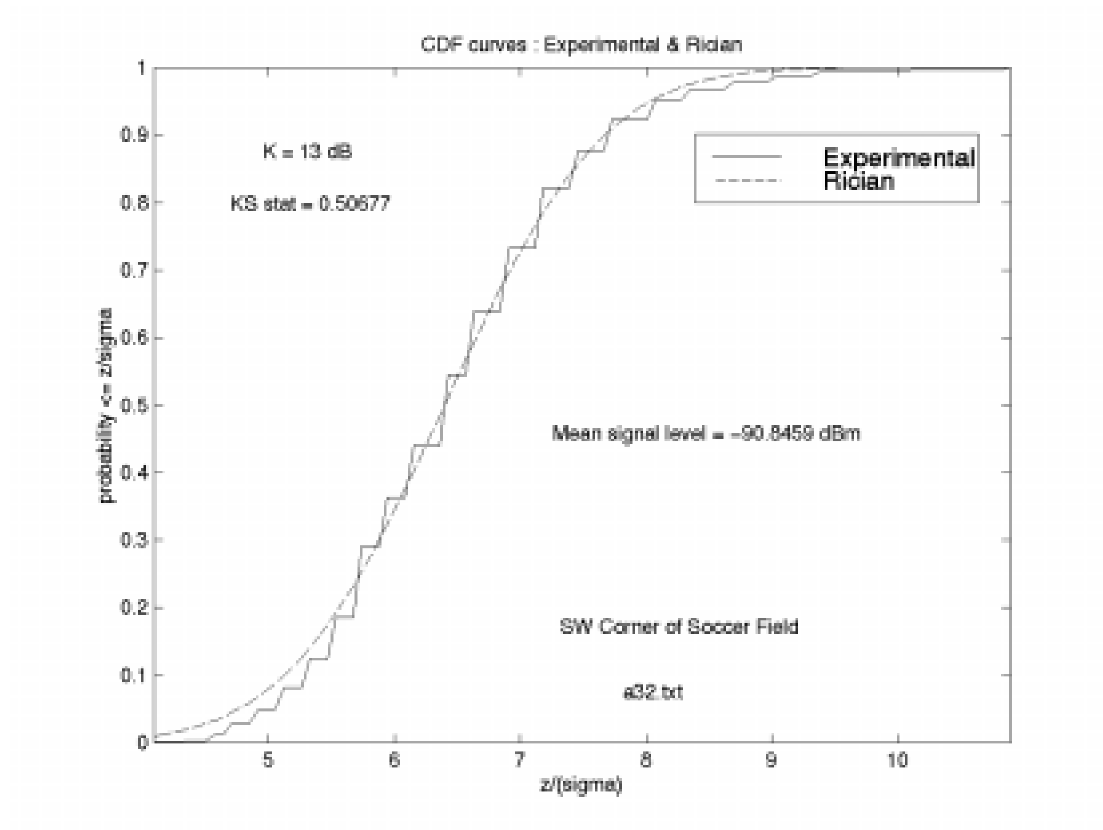

Figure 132: CDF for SW corner of soccer field (a32) and the theoretical Rician CDF.

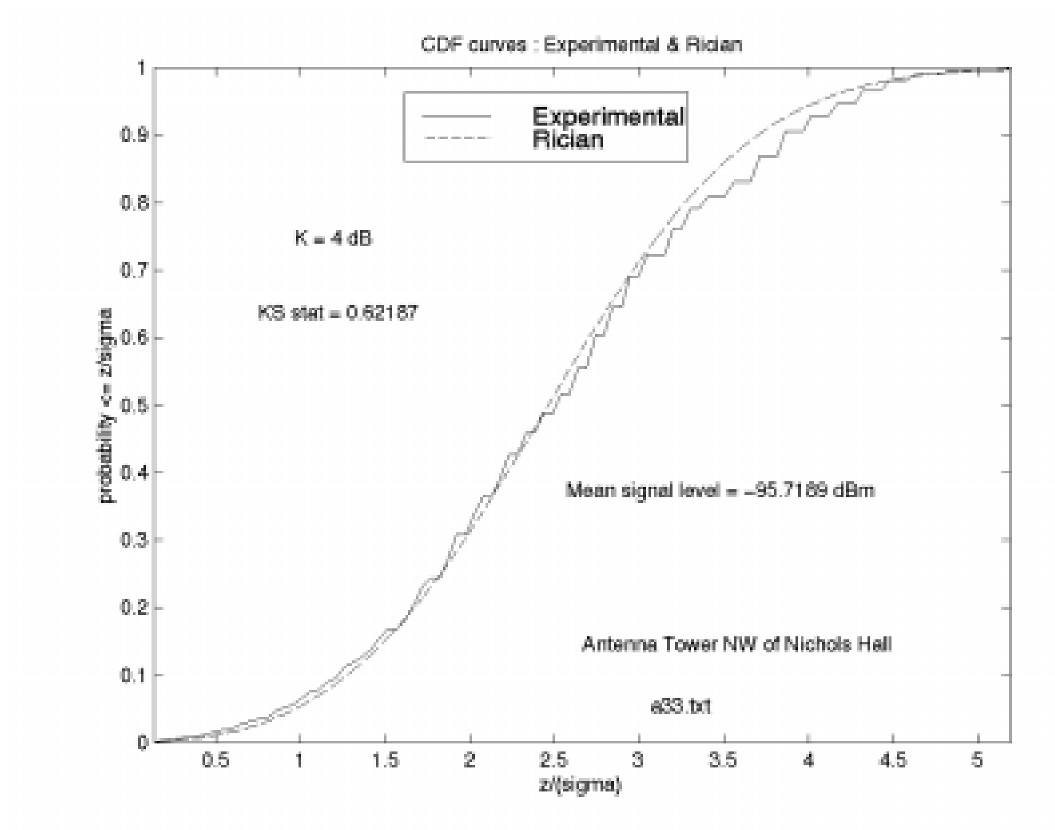

Figure 133: CDF for antenna tower NW of Nichols Hall (a33) and the theoretical Rician CDF.

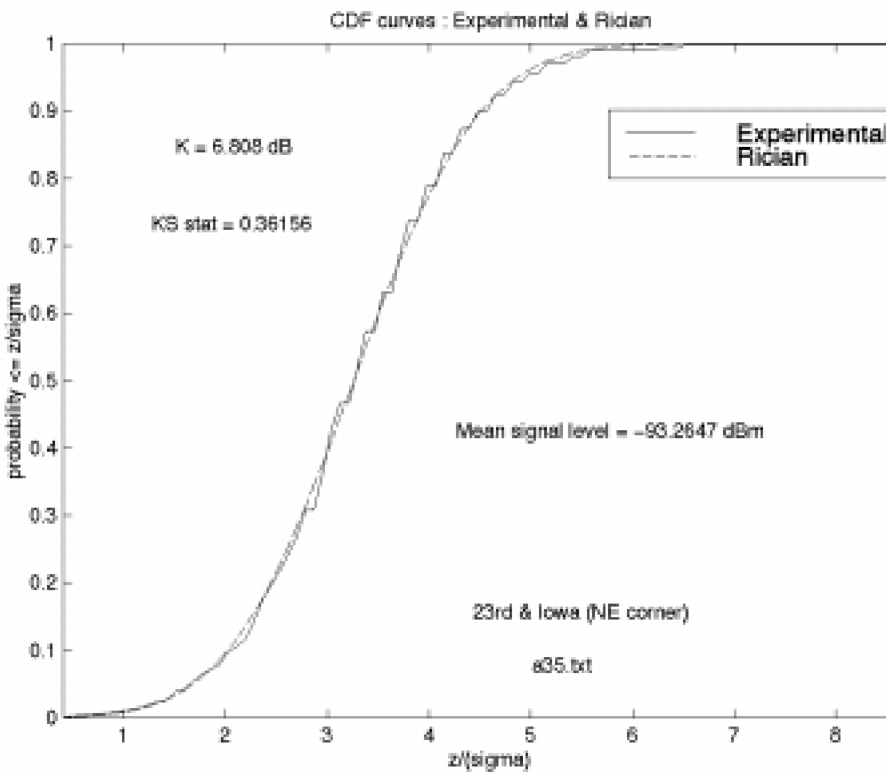

Figure 134: CDF for NE corner 23rd and Iowa (a35) and the theoretical Rician CDF.

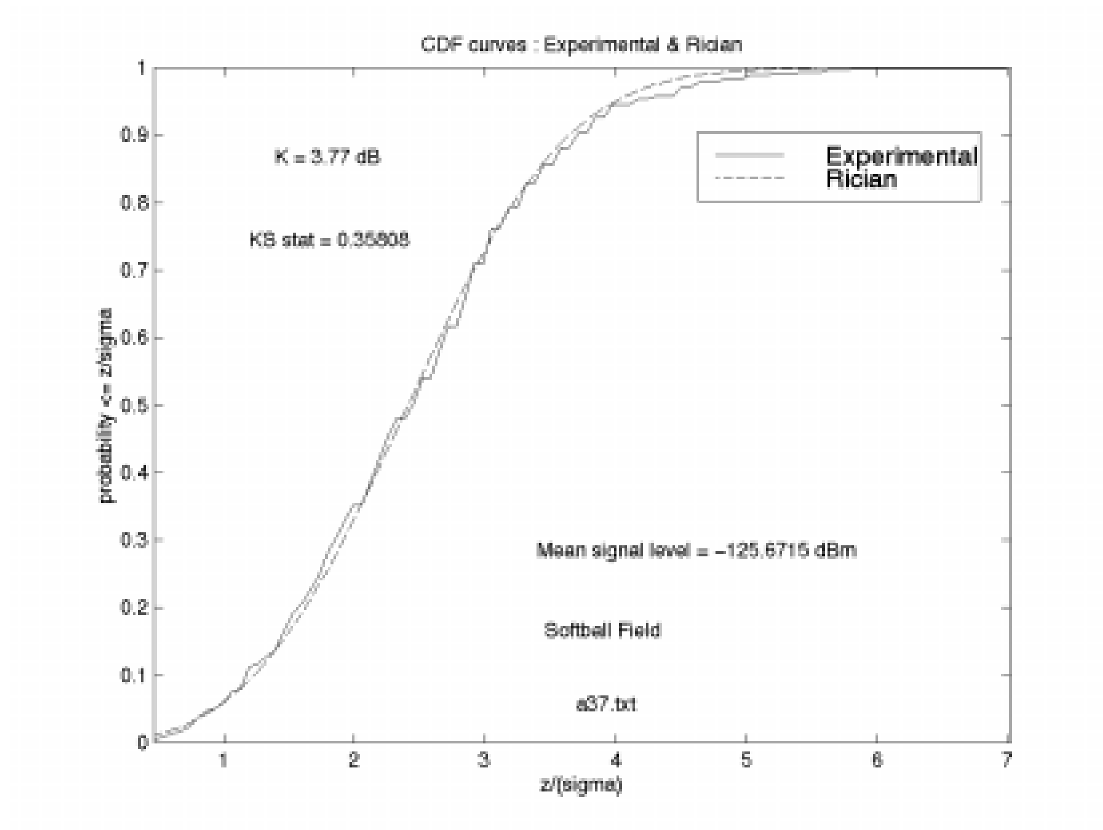

Figure 135: CDF for KU softball field (a37) and the theoretical Rician CDF.

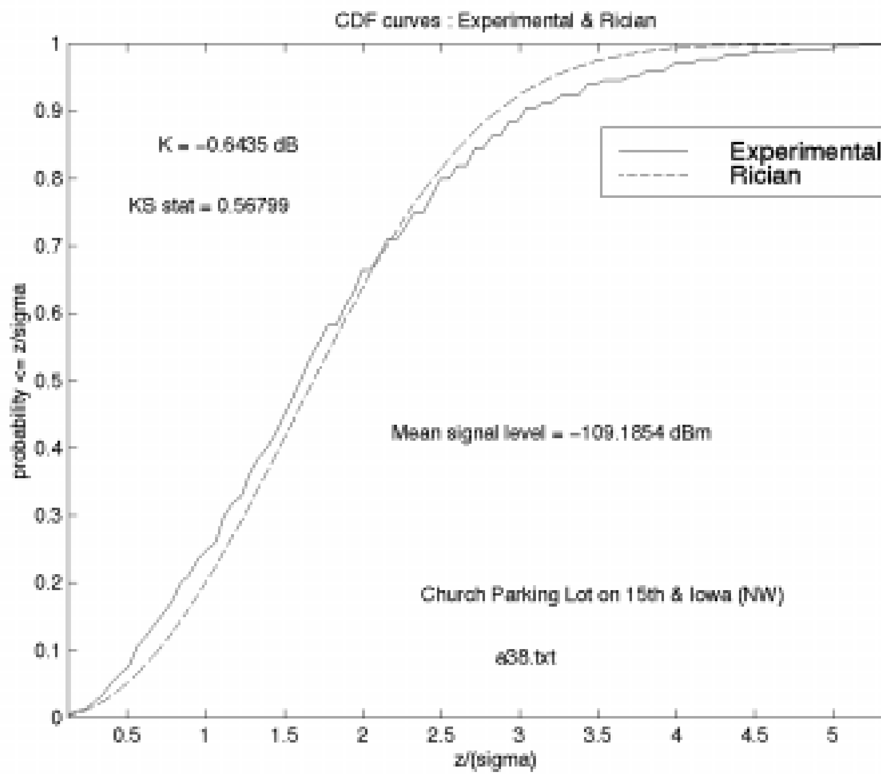

Figure 136: CDF for NW corner 15th and Iowa (a38) and the theoretical Rician CDF.

8 Appendix B: Data Collection Code (Borland C++)

```
/*******************************************************************

            Program Name      COMBO1.CPP

            Purpose           Gathers GPS and Spectrum Analyzer data

*******************************************************************/

#define STRICT

#include <windows.h>
#pragma hdrstop
#include <ddeml.h>
#include <dde.h>
#include <windowsx.h>
#include <string.h>
#include <iostream.h>
#include <dos.h>
#include <bios.h>
#include <math.h>
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include "dde_tim.h"
#include "decl-32.h"


HANDLE         hInst;         /*  Current instance of application */
struct time       timep;


/*
                  The DDE variables
*/

DWORD          idInst = 0L;      /*  Instance of app for DDEML  */
FARPROC        lpDdeProc;        /*  DDE callback function       */
HSZ            hszService;
HSZ            hszTopic;
HSZ            hszItem;
HCONV          hConv = (HCONV)NULL;  /*Handle of established
                                          conversation*/
HDDEDATA       hData;
DWORD          dwResult;
WORD           wFmt = CF_TEXT;    /*  Clipboard format      */
char           szDDEString[80]; /*Local allocation of data buffer */
char           szDDEData[80];   /*  Local receive data buffer   */
```

```
//  Spectrum analyzer variables


int spec_an;  //  file descriptor for spectrum analyzer
int k,j,l;  //  loop variables

char * out = _argv[1];  //  read output file name from command line
char cf_str[30];        //  center frequency
char sp_str[30];        //  span
char rl_str[30];        //  reference level
char ampl[30];
char freq[30];

FILE *outfile = fopen(out,"w+");

/*****************************************************************/

void main()
{

//----------  Initialize GPS ------------------------------

      lpDdeProc = MakeProcInstance ( (FARPROC) DDECallback, hInst );
      if ( DdeInitialize ( (LPDWORD)&idInst,

      (PFNCALLBACK)lpDdeProc,

      APPCMD_CLIENTONLY, 0L ) )
            {
            HandleOutput ( "Client DDE initialization failure." );
            }

      hszService= DdeCreateStringHandle(idInst,"Retki",CP_WINANSI );
      hszTopic= DdeCreateStringHandle ( idInst, "GPS", CP_WINANSI );
     hszItem=DdeCreateStringHandle(idInst,"TxtPosition",CP_WINANSI);

      hConv = DdeConnect ( idInst, hszService,
                              hszTopic,(PCONVCONTEXT) NULL );
      if ( hConv == (HCONV)NULL )
            {
                  HandleError ( DdeGetLastError ( idInst ) );
                  HandleOutput ( "Unsuccessful connection." );
            }
      else
            {
                  hData = DdeClientTransaction ( NULL,

                  0,
```

```
                  hConv,

                  hszItem,

                  wFmt,

                  XTYP_ADVSTART,

                  2000,

                  &dwResult );

                  if ( hData )
                        HandleOutput ( "Successful connection." );
                  else
                  HandleOutput ( "Unsuccessful connection." );
            }
 //---------------------------------------------------


 //------------  Initialize Spectrum analyzer  --------

if((spec_an = ibdev(0,18,0,13,1,0))<0)
      {
        fprintf(stderr,"ERROR%d:%s\n",iberr,"Could not open Spectrum
Analyzer\n");
        if (spec_an != -1)
          {
                  ibonl(spec_an,1);
          }
        exit(1);  //  abort program
      }

ibwrt(spec_an,"IP;",3);     //  instrument preset
ibwrt(spec_an,"TDF P;",6);  //  set output to real numbers
ibwrt(spec_an,"SNGLS;",6);  //  set continuous sweep mode

//  set spectrum analyzer parameters

strcpy(cf_str,"CF ");
strcat(cf_str,_argv[2]); //  get center frequency from command line
strcat(cf_str,"MHZ;");
strcpy(sp_str,"SP ");
strcat(sp_str,_argv[3]); //  get intial span from command line
strcat(sp_str,"MHZ;");
strcpy(rl_str,"RL ");
strcat(rl_str,_argv[4]); //  get reference level from command line
strcat(rl_str,"DB;");
```

```
ibwrt(spec_an,cf_str,strlen(cf_str));     //  set center frequency
//ibwrt(spec_an,sp_str,strlen(sp_str));  //  set span (from command
line)
ibwrt(spec_an,"AT 0DB;",7);                      //  set input
attenuation
ibwrt(spec_an,rl_str,strlen(rl_str));  //  set reference level

//----------------------------------------------------------------
//          Uncomment this section when viewing signal below -105dBm
//----------------------------------------------------------------
/*
ibwrt(spec_an,"CONTS;",6);
ibwrt(spec_an,"VAVG 200;",9);                    //  turn on video
averaging
for (j=1; j<100000 ; j++)
      {
      for (k=1; k<10000; k++)
            {
         }
   }
ibwrt(spec_an,"MKPK HI;",8);  //  mark high amplitude
ibwrt(spec_an,"MKCF;",5);  // make freq at marker the new center
freq
ibwrt(spec_an,"VAVG OFF;",9); //  turn off video averaging
ibwrt(spec_an,"SNGLS;",6);          //  return to single sweep mode
*/
//------------------------------------------------------------------
//------------------------------------------------------------------
/*
ibwrt(spec_an,"TS;",3);       //  take sweep
ibwrt(spec_an,"MKPK HI;",8);  //  mark high amplitude
ibwrt(spec_an,"MKCF;",5);  // make freq at marker the new center
freq
ibwrt(spec_an,"SP 100KHZ;",10);  //  set span

ibwrt(spec_an,"TS;",3);
ibwrt(spec_an,"MKPK HI;",8);  //  mark high amplitude
ibwrt(spec_an,"MKCF;",5);  // make freq at marker the new center
freq
ibwrt(spec_an,"SP 20KHZ;",9); //  set new span          */

ibwrt(spec_an,"SP 20KHZ;",9);  //  set new span
ibwrt(spec_an,"TS;",3);
ibwrt(spec_an,"MKPK HI;",8);
ibwrt(spec_an,"MKCF;",5);  // make freq at marker the new center
freq
ibwrt(spec_an,"SP 10KHZ;",9); //  set new span
```

```
ibwrt(spec_an,"TS;",3);
ibwrt(spec_an,"MKPK HI;",8);
ibwrt(spec_an,"MKCF;",5);  // make freq at marker the new center
freq

ibwrt(spec_an,"SP 2KHZ;",8);  //  set new span

ibwrt(spec_an,"TS;",3);
ibwrt(spec_an,"MKPK HI;",8);
ibwrt(spec_an,"MKCF;",5);  // make freq at marker the new center
freq
//ibwrt(spec_an,"MKPK HI;",8);
/*for (j=1; j<100000 ;j++)
      {
      for(k=1; k<1000; k++)
            {
         }
   }       */
ibwrt(spec_an,"MKA?;",5);            //  ask for amplitude
ibrd(spec_an,ampl,20);        //  read amplitude into a variable
ibwrt(spec_an,"MKF?;",5);            //  ask for frequency
ibrd(spec_an,freq,20);        //  read frequency into a variable

//  write power amplitude and frequency into output file
fprintf(outfile,"amplitude is %f\n",atof(ampl));
fprintf(outfile,"frequency is %f\n",atof(freq));


ibwrt(spec_an,"SP 0MHZ;",7);  //  set new span
ibwrt(spec_an,"ST 20US",7);  //  set sweep time

//---------------------------------------------------------

//-------   loop a given number of measurements

for ( k=1 ; k<=1000 ; k++)
{

//-----  Get time stamp

      gettime(&timep);

      fprintf(outfile,"%d:%d:%d.%d\n",

      timep.ti_hour,timep.ti_min,timep.ti_sec,timep.ti_hund);

//-----  Get GPS data

      hData = DdeClientTransaction ( NULL,
```

```
      0,

      hConv,

      hszItem,

      wFmt,

      XTYP_REQUEST,

      1000,

      &dwResult );

      if ( dwResult == DDE_FNOTPROCESSED )
            HandleOutput ( "Data not available from server." );
      else
            {
                  DdeGetData ( hData, (LPBYTE) szDDEData, 80L, 0L );

                  if ( szDDEData != NULL )
                        HandleOutput ( szDDEData );
                  else
                        HandleOutput ( "Message from server is
null." );
            }

//-----  Get Spectrum analyzer data

      ibwrt(spec_an,"TS;",3);
      ibwrt(spec_an,"MKPK HI;",8);  //  mark high amplitude
      ibwrt(spec_an,"MKA?;",5);           //  ask for amplitude
      ibrd(spec_an,ampl,20);        //  read amplitude into a
variable
      ibwrt(spec_an,"MKF?;",5);           //  ask for frequency
      ibrd(spec_an,freq,20);        //  read frequency into a
variable

//  write power amplitude and frequency into output file
      fprintf(outfile,"amplitude is %f\n",atof(ampl));
      fprintf(outfile,"frequency is %f\n",atof(freq));

}    //----  End of for loop



//----   Close and clear all pointers and handles

if ( hConv != (HCONV)NULL )
```

```
      {
      DdeDisconnect ( hConv );
      hConv = (HCONV)NULL;
      HandleOutput ("Disconnected from server");
   }

DdeFreeStringHandle ( idInst, hszService );
DdeFreeStringHandle ( idInst, hszTopic );
DdeFreeStringHandle ( idInst, hszItem );

FreeProcInstance ( lpDdeProc );
ibonl(spec_an,1);  //  finish remote control of spectrum analyzer

printf("All done -- Press any key to stop program\n");
getch();  //   wait until I can see the output

}   //  End of Main

#pragma warn .eff


/********************************************************************
********/

#pragma argsused
HDDEDATA EXPENTRY _export DDECallback ( WORD wType,
                                        WORD wFmt,
                                        HCONV hConvX,
                                        HSZ hsz1,
                                        HSZ hsz2,
                                        HDDEDATA hData,
                                        DWORD dwData1,
                                        DWORD dwData2 )
{
    switch ( wType )
    {
        case XTYP_DISCONNECT:
            hConv = (HCONV)NULL;
            HandleOutput ( "Retki forced a disconnect." );
            return ( (HDDEDATA) NULL );

        case XTYP_ERROR:
            break;

        case XTYP_XACT_COMPLETE:
            break;

        case XTYP_ADVDATA:
            DdeGetData ( hData, (LPBYTE) szDDEData, 80L, 0L );
            if ( szDDEData != NULL )
```

```
            {
                HandleOutput ( szDDEData );
            }
            else
                HandleOutput ( "Message from server is null." );
            break;

    }

    return ( (HDDEDATA) NULL );
}

/*****************************************************************/

void HandleError ( DWORD DdeError )
{
    switch ( DdeError )
    {
        case DMLERR_DLL_NOT_INITIALIZED:
            HandleOutput ( "DLL not initialized." );
            break;

        case DMLERR_INVALIDPARAMETER:
            HandleOutput ( "Invalid parameter." );
            break;

        case DMLERR_NO_CONV_ESTABLISHED:
            HandleOutput ( "No conversation established." );
            break;

        case DMLERR_NO_ERROR:
            HandleOutput ( "No error." );
            break;
    }
}

/****************************************************************/

void HandleOutput ( char *szOutputString )
{
        fprintf(outfile,"%s\n", szOutputString );
}

/*********************************************************************/

// End of Combo1.cpp

/*********************************************************************/
```

```
/***************************************************************/
//          dde_tim.h
/***************************************************************/

/*
                  Forward References
*/

LRESULT CALLBACK MainWndProc ( HWND, UINT, WPARAM, LPARAM );

HDDEDATA EXPENTRY DDECallback ( WORD, WORD, HCONV, HSZ,
                               HSZ,HDDEDATA, DWORD,  DWORD );
void HandleError ( DWORD );
void HandleOutput ( char * );
/***************************************************************************/
```

// End of dde_tim.h

```
/***************************************************************************/


/*
 *
 *
 *                     Win32 include file
 *          for accessing the 32-bit GPIB DLL (gpib-32.dll)
 *
 *
 *         Contains user variables (ibsta, iberr, ibcnt, ibcntl),
 *         function prototypes and useful defined constants for
 *         calling NI-488 and NI-488.2 routines from a Microsoft
 *         C/C++ Win32 application.
 *
 *
 *            Copyright 1995 National Instruments Corporation
 *
 */

#ifndef DECL_32_H       // ensure we are only included once
#define DECL_32_H

#ifdef __cplusplus
extern "C" {
#endif

/*****************************************************************/
```

```
/*    HANDY CONSTANTS FOR USE BY APPLICATION PROGRAMS ...
*/
/*****************************************************************/
#define UNL  0x3f  /* GPIB unlisten command                */
#define UNT  0x5f  /* GPIB untalk command                  */
#define GTL  0x01  /* GPIB go to local                     */
#define SDC  0x04  /* GPIB selected device clear           */
#define PPC  0x05  /* GPIB parallel poll configure         */
#define GET  0x08  /* GPIB group execute trigger           */
#define TCT  0x09  /* GPIB take control                    */
#define LLO  0x11  /* GPIB local lock out                  */
#define DCL  0x14  /* GPIB device clear                    */
#define PPU  0x15  /* GPIB parallel poll unconfigure       */
#define SPE  0x18  /* GPIB serial poll enable              */
#define SPD  0x19  /* GPIB serial poll disable             */
#define PPE  0x60  /* GPIB parallel poll enable            */
#define PPD  0x70  /* GPIB parallel poll disable           */

/* GPIB status bit vector :                                */
/*       global variable ibsta and wait mask               */

#define ERR     (1<<15) /* Error detected                  */
#define TIMO    (1<<14) /* Timeout                         */
#define END     (1<<13) /* EOI or EOS detected             */
#define SRQI    (1<<12) /* SRQ detected by CIC             */
#define RQS     (1<<11) /* Device needs service            */
#define CMPL    (1<<8)  /* I/O completed                   */
#define LOK     (1<<7)  /* Local lockout state             */
#define REM     (1<<6)  /* Remote state                    */
#define CIC     (1<<5)  /* Controller-in-Charge            */
#define ATN     (1<<4)  /* Attention asserted              */
#define TACS    (1<<3)  /* Talker active                   */
#define LACS    (1<<2)  /* Listener active                 */
#define DTAS    (1<<1)  /* Device trigger state            */
#define DCAS    (1<<0)  /* Device clear state              */

/* Error messages returned in global variable iberr        */

#define EDVR  0  /* System error                           */
#define ECIC  1  /* Function requires GPIB board to be CIC */
#define ENOL  2  /* Write function detected no Listeners   */
#define EADR  3  /* Interface board not addressed correctly*/
#define EARG  4  /* Invalid argument to function call      */
#define ESAC  5  /* Function requires GPIB board to be SAC */
#define EABO  6  /* I/O operation aborted                  */
#define ENEB  7  /* Non-existent interface board           */
#define EDMA  8  /* Error performing DMA                   */
#define EOIP 10  /* I/O operation started before previous  */
                 /* operation completed                    */
#define ECAP 11  /* No capability for intended operation   */
```

```
#define EFSO 12  /* File system operation error            */
#define EBUS 14  /* Command error during device call       */
#define ESTB 15  /* Serial poll status byte lost           */
#define ESRQ 16  /* SRQ remains asserted                   */
#define ETAB 20  /* The return buffer is full.             */
#define ELCK 21  /* Address or board is locked.            */

/* EOS mode bits                                           */

#define BIN  (1<<12) /* Eight bit compare                  */
#define XEOS (1<<11) /* Send END with EOS byte             */
#define REOS (1<<10) /* Terminate read on EOS              */

/* Timeout values and meanings                             */

#define TNONE    0   /* Infinite timeout (disabled)        */
#define T10us    1   /* Timeout of 10 us (ideal)           */
#define T30us    2   /* Timeout of 30 us (ideal)           */
#define T100us   3   /* Timeout of 100 us (ideal)          */
#define T300us   4   /* Timeout of 300 us (ideal)          */
#define T1ms     5   /* Timeout of 1 ms (ideal)            */
#define T3ms     6   /* Timeout of 3 ms (ideal)            */
#define T10ms    7   /* Timeout of 10 ms (ideal)           */
#define T30ms    8   /* Timeout of 30 ms (ideal)           */
#define T100ms   9   /* Timeout of 100 ms (ideal)          */
#define T300ms  10   /* Timeout of 300 ms (ideal)          */
#define T1s     11   /* Timeout of 1 s (ideal)             */
#define T3s     12   /* Timeout of 3 s (ideal)             */
#define T10s    13   /* Timeout of 10 s (ideal)            */
#define T30s    14   /* Timeout of 30 s (ideal)            */
#define T100s   15   /* Timeout of 100 s (ideal)           */
#define T300s   16   /* Timeout of 300 s (ideal)           */
#define T1000s  17   /* Timeout of 1000 s (ideal)          */


/*  IBLN Constants                                         */
#define NO_SAD   0
#define ALL_SAD -1


/*  The following constants are used for the second parameter of the
 *  ibconfig function.  They are the "option" selection codes.
 */
#define  IbcPAD        0x0001      /* Primary Address
*/
#define  IbcSAD        0x0002      /* Secondary Address
*/
#define  IbcTMO        0x0003      /* Timeout Value
*/
```

```
#define  IbcEOT        0x0004      /* Send EOI with last data byte?
*/
#define  IbcPPC        0x0005      /* Parallel Poll Configure
*/
#define  IbcREADDR     0x0006      /* Repeat Addressing
*/
#define  IbcAUTOPOLL   0x0007      /* Disable Auto Serial Polling
*/
#define  IbcCICPROT    0x0008      /* Use the CIC Protocol?
*/
#define  IbcIRQ        0x0009      /* Use PIO for I/O
*/
#define  IbcSC         0x000A      /* Board is System Controller?
*/
#define  IbcSRE        0x000B      /* Assert SRE on device calls?
*/
#define  IbcEOSrd      0x000C      /* Terminate reads on EOS
*/
#define  IbcEOSwrt     0x000D      /* Send EOI with EOS character
*/
#define  IbcEOScmp     0x000E      /* Use 7 or 8-bit EOS compare
*/
#define  IbcEOSchar    0x000F      /* The EOS character.
*/
#define  IbcPP2        0x0010      /* Use Parallel Poll Mode 2.
*/
#define  IbcTIMING     0x0011      /* NORMAL, HIGH, or VERY_HIGH
timing.   */
#define  IbcDMA        0x0012      /* Use DMA for I/O
*/
#define  IbcReadAdjust 0x0013      /* Swap bytes during an ibrd.
*/
#define  IbcWriteAdjust 0x014      /* Swap bytes during an ibwrt.
*/
#define  IbcSendLLO    0x0017      /* Enable/disable the sending of
LLO.      */
#define  IbcSPollTime  0x0018      /* Set the timeout value for
serial polls. */
#define  IbcPPollTime  0x0019      /* Set the parallel poll length
period.    */
#define  IbcEndBitIsNormal 0x001A  /* Remove EOS from END bit of
IBSTA.       */
#define  IbcUnAddr         0x001B  /* Enable/disable device
unaddressing.     */
#define  IbcSignalNumber   0x001C  /* Set UNIX signal number -
unsupported */
#define  IbcBlockIfLocked  0x001D   /* Enable/disable blocking for
locked boards/devices          */
#define  IbcHSCableLength  0x001F  /* Length of cable specified for
high speed timing.*/
```

```
#define  IbcIst        0x0020     /* Set the IST bit.
*/
#define  IbcRsv        0x0021     /* Set the RSV byte.
*/

/*
 *    Constants that can be used (in addition to the ibconfig
constants)
 *    when calling the ibask() function.
 */

#define  IbaPAD              IbcPAD
#define  IbaSAD              IbcSAD
#define  IbaTMO              IbcTMO
#define  IbaEOT              IbcEOT
#define  IbaPPC              IbcPPC
#define  IbaREADDR           IbcREADDR
#define  IbaAUTOPOLL         IbcAUTOPOLL
#define  IbaCICPROT          IbcCICPROT
#define  IbaIRQ              IbcIRQ
#define  IbaSC               IbcSC
#define  IbaSRE              IbcSRE
#define  IbaEOSrd            IbcEOSrd
#define  IbaEOSwrt           IbcEOSwrt
#define  IbaEOScmp           IbcEOScmp
#define  IbaEOSchar          IbcEOSchar
#define  IbaPP2              IbcPP2
#define  IbaTIMING           IbcTIMING
#define  IbaDMA              IbcDMA
#define  IbaReadAdjust       IbcReadAdjust
#define  IbaWriteAdjust      IbcWriteAdjust
#define  IbaSendLLO          IbcSendLLO
#define  IbaSPollTime        IbcSPollTime
#define  IbaPPollTime        IbcPPollTime
#define  IbaEndBitIsNormal IbcEndBitIsNormal
#define  IbaUnAddr           IbcUnAddr
#define  IbaSignalNumber     IbcSignalNumber
#define  IbaBlockIfLocked    IbcBlockIfLocked
#define  IbaHSCableLength    IbcHSCableLength
#define  IbaIst              IbcIst
#define  IbaRsv              IbcRsv

#define  IbaBNA              0x0200   /* A device’s access board.
*/


/* Values used by the Send 488.2 command. */
#define  NULLend 0x00  /* Do nothing at the end of a transfer.*/
#define  NLend   0x01  /* Send NL with EOI after a transfer.  */
#define  DABend  0x02  /* Send EOI with the last DAB.         */
```

```
/* Value used by the 488.2 Receive command.
 */
#define  STOPend     0x0100


/* Address type (for 488.2 calls) */

typedef short Addr4882_t;       /* System dependent: must be 16 bits
*/

/*
 *  This macro can be used to easily create an entry in address list
 *  that is required by many of the 488.2 functions. The primary
address goes in the
 *  lower 8-bits and the secondary address goes in the upper 8-bits.
 */
#define  MakeAddr(pad, sad)   ((Addr4882_t)(((pad)&0xFF) |
((sad)<<8)))

/*
 *  This value is used to terminate an address list.  It should be
 *  assigned to the last entry.
 */
#ifndef NOADDR
#define  NOADDR    (Addr4882_t)((unsigned short)0xFFFF)
#endif

/*
 *  The following two macros are used to "break apart" an address
list
 *  entry.  They take an unsigned integer and return either the
primary
 *  or secondary address stored in the integer.
 */
#define  GetPAD(val)    ((val) & 0xFF)
#define  GetSAD(val)    (((val) >> 8) & 0xFF)

/* iblines constants */

#define  ValidEOI   (short)0x0080
#define  ValidATN   (short)0x0040
#define  ValidSRQ   (short)0x0020
#define  ValidREN   (short)0x0010
#define  ValidIFC   (short)0x0008
#define  ValidNRFD  (short)0x0004
#define  ValidNDAC  (short)0x0002
#define  ValidDAV   (short)0x0001
#define  BusEOI     (short)0x8000
#define  BusATN     (short)0x4000
```

```
#define  BusSRQ     (short)0x2000
#define  BusREN     (short)0x1000
#define  BusIFC     (short)0x0800
#define  BusNRFD    (short)0x0400
#define  BusNDAC    (short)0x0200
#define  BusDAV     (short)0x0100

/****
 **** typedef for ibnotify callback ****
 ****/
typedef int (__stdcall * GpibNotifyCallback_t)(int, int, int, long,
PVOID);

#define IBNOTIFY_REARM_FAILED    0xE00A003F


/******************************************************************/
/*
*/
/*  iblockx and ibunlockx definitions --- for use with GPIB-ENET
only !! */
/*
*/
/******************************************************************/
#define  TIMMEDIATE                -1
#define  TINFINITE                 -2
#define  MAX_LOCKSHARENAME_LENGTH   64

#if defined(UNICODE)
   #define iblockx iblockxW
#else
   #define iblockx iblockxA
#endif

extern int __stdcall iblockxA (int ud, int LockWaitTime, PCHAR
LockShareName);
extern int __stdcall iblockxW (int ud, int LockWaitTime, PWCHAR
LockShareName);
extern int __stdcall ibunlockx (int ud);


/*******************************************************************/
/*          IBSTA, IBERR, IBCNT, IBCNTL and FUNCTION PROTOTYPES
*/
/*      ( only included if not accessing the 32-bit DLL directly )
*/
/******************************************************************/
#if !defined(GPIB_DIRECT_ACCESS)

/*
```

```
 *  Set up access to the user variables (ibsta, iberr, ibcnt,
ibcntl).
 *  These are declared and exported by the 32-bit DLL.  Separate
copies
 *  exist for each process that accesses the DLL.  They are shared
by
 *  multiple threads of a single process.
 */

extern int  ibsta;
extern int  iberr;
extern int  ibcnt;
extern long ibcntl;


#if defined(UNICODE)
   #define ibbna  ibbnaW
   #define ibfind ibfindW
   #define ibrdf  ibrdfW
   #define ibwrtf ibwrtfW
#else
   #define ibbna  ibbnaA
   #define ibfind ibfindA
   #define ibrdf  ibrdfA
   #define ibwrtf ibwrtfA
#endif

/*
 *  Extern 32-bit GPIB DLL functions
 */

/*  NI-488 Function Prototypes  */
extern int __stdcall ibfindA   (LPCSTR udname);
extern int __stdcall ibbnaA    (int ud, LPCSTR udname);
extern int __stdcall ibrdfA    (int ud, LPCSTR filename);
extern int __stdcall ibwrtfA   (int ud, LPCSTR filename);

extern int __stdcall ibfindW   (LPCWSTR udname);
extern int __stdcall ibbnaW    (int ud, LPCWSTR udname);
extern int __stdcall ibrdfW    (int ud, LPCWSTR filename);
extern int __stdcall ibwrtfW   (int ud, LPCWSTR filename);

extern int __stdcall ibask    (int ud, int option, PINT v);
extern int __stdcall ibcac    (int ud, int v);
extern int __stdcall ibclr    (int ud);
extern int __stdcall ibcmd    (int ud, PVOID buf, long cnt);
extern int __stdcall ibcmda   (int ud, PVOID buf, long cnt);
extern int __stdcall ibconfig (int ud, int option, int v);
extern int __stdcall ibdev    (int ud, int pad, int sad, int tmo,
int eot, int eos);
```

```
extern int __stdcall ibdiag   (int ud, PVOID buf, long cnt);
extern int __stdcall ibdma    (int ud, int v);
extern int __stdcall ibeos    (int ud, int v);
extern int __stdcall ibeot    (int ud, int v);
extern int __stdcall ibgts    (int ud, int v);
extern int __stdcall ibist    (int ud, int v);
extern int __stdcall iblines  (int ud, PSHORT result);
extern int __stdcall ibln     (int ud, int pad, int sad, PSHORT
listen);
extern int __stdcall ibloc    (int ud);
extern int __stdcall ibnotify (int ud, int mask,
GpibNotifyCallback_t Callback, PVOID RefData);
extern int __stdcall ibonl    (int ud, int v);
extern int __stdcall ibpad    (int ud, int v);
extern int __stdcall ibpct    (int ud);
extern int __stdcall ibpoke   (int ud, long option, long v);
extern int __stdcall ibppc    (int ud, int v);
extern int __stdcall ibrd     (int ud, PVOID buf, long cnt);
extern int __stdcall ibrda    (int ud, PVOID buf, long cnt);
extern int __stdcall ibrpp    (int ud, PCHAR ppr);
extern int __stdcall ibrsc    (int ud, int v);
extern int __stdcall ibrsp    (int ud, PCHAR spr);
extern int __stdcall ibrsv    (int ud, int v);
extern int __stdcall ibsad    (int ud, int v);
extern int __stdcall ibsic    (int ud);
extern int __stdcall ibsre    (int ud, int v);
extern int __stdcall ibstop   (int ud);
extern int __stdcall ibtmo    (int ud, int v);
extern int __stdcall ibtrg    (int ud);
extern int __stdcall ibwait   (int ud, int mask);
extern int __stdcall ibwrt    (int ud, PVOID buf, long cnt);
extern int __stdcall ibwrta   (int ud, PVOID buf, long cnt);

// GPIB-ENET only functions to support locking across machines
extern int __stdcall iblock   (int ud);
extern int __stdcall ibunlock (int ud);

/************************************************************/
/*  Functions to access Thread-Specific copies of the GPIB global
vars */

extern int  __stdcall ThreadIbsta (void);
extern int  __stdcall ThreadIberr (void);
extern int  __stdcall ThreadIbcnt (void);
extern long __stdcall ThreadIbcntl (void);


/*******************************************************************/
/*  NI-488.2 Function Prototypes  */
```

```
extern void __stdcall AllSpoll      (int boardID, Addr4882_t *
addrlist, PSHORT results);
extern void __stdcall DevClear      (int boardID, Addr4882_t addr);
extern void __stdcall DevClearList  (int boardID, Addr4882_t *
addrlist);
extern void __stdcall EnableLocal   (int boardID, Addr4882_t *
addrlist);
extern void __stdcall EnableRemote  (int boardID, Addr4882_t *
addrlist);
extern void __stdcall FindLstn      (int boardID, Addr4882_t *
addrlist, PSHORT results, int limit);
extern void __stdcall FindRQS       (int boardID, Addr4882_t *
addrlist, PSHORT dev_stat);
extern void __stdcall PPoll         (int boardID, PSHORT result);
extern void __stdcall PPollConfig   (int boardID, Addr4882_t addr,
int dataLine, int lineSense);
extern void __stdcall PPollUnconfig (int boardID, Addr4882_t *
addrlist);
extern void __stdcall PassControl   (int boardID, Addr4882_t addr);
extern void __stdcall RcvRespMsg    (int boardID, PVOID buffer, long
cnt, int Termination);
extern void __stdcall ReadStatusByte(int boardID, Addr4882_t addr,
PSHORT result);
extern void __stdcall Receive       (int boardID, Addr4882_t addr,
PVOID buffer, long cnt, int Termination);
extern void __stdcall ReceiveSetup  (int boardID, Addr4882_t addr);
extern void __stdcall ResetSys      (int boardID, Addr4882_t *
addrlist);
extern void __stdcall Send          (int boardID, Addr4882_t addr,
PVOID databuf, long datacnt, int eotMode);
extern void __stdcall SendCmds      (int boardID, PVOID buffer, long
cnt);
extern void __stdcall SendDataBytes (int boardID, PVOID buffer, long
cnt, int eot_mode);
extern void __stdcall SendIFC       (int boardID);
extern void __stdcall SendLLO       (int boardID);
extern void __stdcall SendList      (int boardID, Addr4882_t *
addrlist, PVOID databuf, long datacnt, int eotMode);
extern void __stdcall SendSetup     (int boardID, Addr4882_t *
addrlist);
extern void __stdcall SetRWLS       (int boardID, Addr4882_t *
addrlist);
extern void __stdcall TestSRQ       (int boardID, PSHORT result);
extern void __stdcall TestSys       (int boardID, Addr4882_t *
addrlist, PSHORT results);
extern void __stdcall Trigger       (int boardID, Addr4882_t addr);
extern void __stdcall TriggerList   (int boardID, Addr4882_t *
addrlist);
extern void __stdcall WaitSRQ       (int boardID, PSHORT result);
```

```
#endif

#ifdef __cplusplus
}
#endif

#endif   // DECL_32_H
```

/***/

// End of decl-32.h

/***/

9 Appendix C: CDF Analysis: Determining K (Matlab™ code)

```
%  This program reads the data from an input
%  file and extracts the spectrum analyzer data
%  The amplitude data is then used to construct
%  a cdf and compare it to a Rician.
%  The Kolmogorov-Smirnov test is used
%  to test for goodness-of-fit.

clear

fid1 = fopen('dfiles/a35.txt'); %  open data file

%  parse input file
connect1 = fgetl(fid1);

k = 0;

while ((feof(fid1)==0))
      time1 = fgetl(fid1);
      if (time1(1) ~= 'D' & feof(fid1)==0)
            k = k+1;
            [t1h,temp] = strtok(time1,[':']);
            [t1m,temp] = strtok(temp,[':']);
            t1s = strtok(temp,[':']);

            loc1 = fgetl(fid1);
            amp1 = fgetl(fid1);

            while(amp1(1) ~= 'a')
                  loc1 = amp1;
                  amp1 = fgetl(fid1);
            end

            [lt1d,temp] = strtok(loc1);
            lt1d = str2num(lt1d(1:length(lt1d)-1));
            [lt1m,temp] = strtok(temp);
            lt1m = str2num(lt1m(1:length(lt1m)-1));
            [lt1s,temp] = strtok(temp);
            lt1s = str2num(lt1s(1:length(lt1s)-1));
            [trash,temp] = strtok(temp);
            [ln1d,temp] = strtok(temp);
            ln1d = str2num(ln1d(1:length(ln1d)-1));
            [ln1m,temp] = strtok(temp);
            ln1m = str2num(ln1m(1:length(ln1m)-1));
            [ln1s,temp] = strtok(temp);
            ln1s = str2num(ln1s(1:length(ln1s)-1));
            [trash,temp] = strtok(temp);
            alt1 = strtok(temp);
```

```
            alt1 = str2num(alt1);

            lat1(k) = lt1d + lt1m/60 + lt1s/3600;
            lon1(k) = ln1d + ln1m/60 + ln1s/3600;
            alt(k) = alt1;

            [amp1,temp] = strtok(amp1);
            [amp1,temp] = strtok(temp);
            a1(k) =  str2num(temp);

            freq1 = fgetl(fid1);
            [freq1,temp] = strtok(freq1);
            [freq1,temp] = strtok(temp);
            f1(k) = str2num(temp);

      end
end
fprintf('finished with file\n')
fclose(fid1);

power = 10.^(a1/10)*10^(-3); % change to volts
a = sqrt(power*50);
[n,x] = hist(a,100);
n_norm = n/length(a);
cdf(1) = n_norm(1);
for m = 2:length(n_norm)
      cdf(m) = cdf(m-1) + n_norm(m);
end

x_sigma = std(a);
x_var = x_sigma^2;
x_sqrd = x.^2;
x2_norm = x_sqrd/(x_var);
x_norm = sqrt(x2_norm);

fprintf('finished with data\n')


% Plot Rician curve
KdB = 6.808;  %  Rician parameter in dB
K = 10^(KdB/10);
z = x;
sigma = x_sigma;
alpha = sqrt(2*K);
omega = z/sigma;
coeff = exp(-(omega.^2+alpha^2)/2);


for m = 1:length(z)
      temp = 0;
```

```
        for k = 0:40
                temp = temp + coeff(m)*(alpha/omega(m))^k * ...
                        besseli(k,alpha*omega(m));
        end
        errdiff = 1;
        l = k;
        oldtemp = temp;
        newtemp = temp;
        while( (errdiff > 10^(-6)) )
                l = l+1;
                newtemp = oldtemp + coeff(m)*(alpha/omega(m))^l * ...
                        besseli(l,alpha*omega(m));
                errdiff = abs(newtemp-oldtemp);
                oldtemp = newtemp;
        end
        Q(m) = newtemp;
        if Q(m) > 1
                Q(m) = 1;
        end
        P(m) = 1-Q(m);
        iter(m) = l;
        fprintf('# of iter = %d\n',l)
        fprintf('m = %d\n',m)
end
z_norm = x_norm;

%  Find average location

avglat = sum(lat1)/length(lat1)
avglon = sum(lon1)/length(lon1)
avgalt = sum(alt)/length(alt)



plot(x_norm,cdf,'k',z_norm,P,'k--')
axis([min(x_norm) max(x_norm) 0 1])
ylabel('probability <= z/sigma')
title('CDF curves : Experimental & Rician')
legend('Experimental','Rician')
gtext(['K = ' num2str(KdB) ' dB'])
gtext('a35.txt')
gtext('23rd & Iowa (NE corner)')
gtext(['Mean signal level = ' num2str(mean(a1)) ' dBm'])

%  Kolmogorov-Smirnov test
%     D is based on max distance between the
%     experimental cdf and the theoretical cdf

D = max(abs(cdf-P));
samples = length(P);
```

```
KStest = D*(sqrt(samples) + 0.12 + 0.11/sqrt(samples))
gtext(['KS stat = ' num2str(KStest)])

print -dtiff graphs/cdf/a35
```

10 Appendix D: Linear Regression: Path Loss and K (Matlab™ code)

```
%     Linear regression

clear

Tx  = 30 ;   %  transmit power in dBm
conf = 90;   % choose only 80, 90, 95, 98 or 99
if conf == 80
      ind = 1;
elseif conf == 90
      ind = 2;
elseif conf == 95
      ind = 3;
elseif conf == 98
      ind = 4;
else
      ind = 5;
end

%  distances in km
xin = ([0.15 0.39 0.43 0.34      0.85 ...
      0.80 0.74 0.44 0.22 0.36 0.38 ...
      0.45 0.64 0.80 1.14 1.01      ...
      0.84      0.80 0.30 0.12      ...
      0.13           0.49      0.56 ...
      0.40 0.91 0.40      1.04      ...
      0.93 0.68]);

x = 10*log10(xin);

%  Received powers in dBm  ( use K in dB values for K test )
Rx = [-94.51 -96.99 -77.15 -99.5        -98.51 ...
     -103.2 -110.9 -88.11 -70.53 -98.09 -103.4 ...
     -93.61 -99.63 -109.2 -83.35 -122.3 ...
     -117.6        -125.6 -100.1 -74.28...
     -68.19               -112.3        -121.1 ...
     -100.9 -90.85 -95.72         -93.26     ...
     -125.7 -109.2];

%  y is the path loss

y = Tx-Rx;  %  y = Rx when determining K


x2 = x.^2;
y2 = y.^2;
xy = x.*y;
```

```
tot = length(x);
x_avg = mean(x);
y_avg = mean(y);

term1 = sum(x2);
term2 = sum(x)^2/tot;
s1_2 = (term1-term2)/(tot-1);

term3 = sum(xy);
term4 = sum(x)*sum(y)/tot;
s_xy = (term3-term4)/(tot-1);

%  slope of the line is b
b = s_xy/s1_2 ;

%  now find confidence interval

dof = length(x)-2;
if dof <= 30
      t = [   3.078 6.314 12.706 31.821 63.657;
            1.886 2.920  4.303  6.965  9.925;
            1.638 2.353  3.182  4.541  5.841;
            1.533 2.132  2.776  3.747  4.604;
            1.476 2.015  2.571  3.365  4.032;
            1.440 1.943  2.447  3.143  3.707;
            1.415 1.895  2.365  2.998  3.499;
            1.397 1.860  2.306  2.896  3.355;
            1.383 1.833  2.262  2.821  3.250;
            1.372 1.812  2.228  2.764  3.169;
            1.363 1.796  2.201  2.718  3.106;
            1.356 1.782  2.179  2.681  3.055;
            1.350 1.771  2.160  2.650  3.012;
            1.345 1.761  2.145  2.624  2.977;
            1.341 1.753  2.131  2.602  2.947;
            1.337 1.746  2.120  2.583  2.921;
            1.333 1.740  2.110  2.567  2.898;
            1.330 1.734  2.101  2.552  2.878;
            1.328 1.729  2.093  2.539  2.861;
            1.325 1.725  2.086  2.528  2.845;
            1.323 1.721  2.080  2.518  2.831;
            1.321 1.717  2.074  2.508  2.819;
            1.319 1.714  2.069  2.500  2.807;
            1.318 1.711  2.064  2.492  2.797;
            1.316 1.708  2.060  2.485  2.787;
            1.315 1.706  2.056  2.479  2.779;
            1.314 1.703  2.052  2.473  2.771;
            1.313 1.701  2.048  2.467  2.763;
            1.311 1.699  2.045  2.462  2.756;
            1.310 1.697  2.042  2.457  2.750];
      c = t(dof,ind);
```

```
else
      t = [1.282 1.645 1.960 2.326 2.576];
      c = t(ind);
end


s2_2 = (sum(y2) - sum(y)^2/tot)/(tot-1);
q0 = (tot-1)*(s2_2 - b^2*s1_2);

%  Note :  c increases the one-sided interval linearly
k = c*sqrt(q0/((tot-2)*(tot-1)*s1_2));

lower = b - k;
upper = b+k;

% set up line object for calculated slope

y_int = y_avg - b*x_avg
L_x = [min(x) max(x)];
L_y(1) = b*L_x(1) + y_int;
L_y(2) = b*L_x(2) + y_int;

% Determine standard deviation

q_0 = y - y_int - b*x;  %  q_0 is the deviation from the mean
q = sum(q_0.^2);
sigma = sqrt(q/(length(x)-1));

semilogx(xin,y,'k*',10.^(L_x/10),L_y,'k')
axis([0.1  max(xin)+1 min(y)-10 max(y)+10])
title('Determination of Path Loss Exponent (ALL)')
xlabel('Distance (km)')
ylabel('Path Loss (dB)')
gtext(['n = ' num2str(b)])
gtext(['sigma = ' num2str(sigma) ' dB'])
gtext(['[ ' num2str(lower) ', ' num2str(upper) ' ]'])
gtext(['Confidence level = ' num2str(conf) '%'])

print -dtiff graphs/pathloss/linreg
```

11 Appendix E: Determination of $\frac{E_b}{N_o}$ for Specific Average BER (Matlab™ code)

```
%-------------------------------------------------------------
%  This program builds a table of required Eb/No for each K
%  given a bit error rate requirement
%-------------------------------------------------------------


clear

lower = -10;  % Must be an integer
upper = 29.99;
K_step = 0.01;

KdB = lower:K_step:upper;

BERreq = 10^(-5);  %  required BER

K = 10.^(KdB/10);
delta_eb = 1;

last1 = 1;
last2 = 0;

oldrow = 0;

for p = 1: length(K)
    if p == 1
       Eb_No_dB =  44;  % initial Eb/No estimate
      last1 = Eb_No_dB;
    elseif p == 2
       Eb_No_dB =  last1 - delta_eb; % initial Eb/No estimate
    else
       Eb_No_dB =  last1 -  delta_eb; % initial Eb/No estimate
       delta_eb = abs(last1 - last2);
    end
            accrcy = BERreq*10^(-3); % BER accuracy
            delta = 0.1;
            diff = 1;
            olddir = 1;
            newdir = 1;
            z = 0;
            while (abs(diff) > accrcy)
                  z = z+1;
                  Eb_No = 10^(Eb_No_dB/10);
                  mu_k = sqrt((Eb_No)/(1+K(p)+Eb_No));
```

```
            alpha = sqrt(K(p)/2)*(1-mu_k);
            omega = sqrt(K(p)/2)*(1+mu_k);
            coeff = exp(-(omega^2+alpha^2)/2);

            temp = 0;
            for k_cntr = 0:50
                  temp=temp+(alpha/omega)^k_cntr...
                    * besseli(k_cntr,alpha*omega);
            end

            Q = coeff*temp;
            temp1 = 0.5*(1+mu_k)* ...
                  exp(-K(p)/2*(1+mu_k^2));
            temp2 = temp1* ...
                  besseli(0,K(p)/2*(1-mu_k^2));
            avgBER = Q - temp2;
            diff = avgBER - BERreq;
            if (newdir ~= olddir)
                  delta = delta/2;
            end
            olddir = newdir;
            if( abs(diff) > accrcy )
                  if (diff > 0)
                        Eb_No_dB=Eb_No_dB+delta;
                        newdir = 1;
                  else
                        Eb_No_dB=Eb_No_dB-delta;
                        newdir = -1;
                  end

            end
      end

      fprintf('Number of iterations = %d\n', z)
      fprintf('Eb_No (dB) = %d\n',10*log10(Eb_No))
      fprintf('average BER = %d\n',avgBER)



row = floor(round(KdB(p)*100)/100) + abs(lower) + 1
tempcol = mod(round(abs(KdB(p))*100),100) ;
if KdB(p) < 0
      if tempcol == 0
            tempcol = 100;
      end
      col = round(101 - tempcol)
else
      col = round(tempcol + 1)
end
```

```
      newrow = row;
      if newrow == oldrow & col ==1
            fprintf('newrow = oldrow : row = %d\n',row)
            pause
      end
      oldrow = newrow;

      ebno_a(row,col) = 10*log10(Eb_No);
      last2 = last1;
      last1 = ebno_a(row,col);

end

save ebno_BER.mat ebno_a
```

12 Appendix F: Determination of $\frac{E_b}{N_o}$ for Specific Average PER (Matlab™ code)

```
%----------------------------------------------------
%  This program builds a table of required Eb/No values
%   for each K given a required BER which is translated into
%   an equivalent packet error rate
%------------------------------------------------------------


clear

lower = -10;  % Must be an integer
upper = 29.99;
K_step = 0.01;

KdB = lower:K_step:upper;

BERreq = 10^(-5);  %  required bit error rate
p_size = 424;  %  packet size in bits
PERreq = 1-(1-BERreq)^p_size;  % equivalent packet error rate

K = 10.^(KdB/10);
delta_eb = 1;

last1 = 1;
last2 = 0;

oldrow = 0;

for p = 1: length(K)
    if p == 1
       Eb_No_dB =  30.828;    % initial Eb/No estimate
      last1 = Eb_No_dB;
    elseif p == 2
       Eb_No_dB =  last1 - delta_eb; % initial Eb/No estimate
    else
       Eb_No_dB =  last1 -  delta_eb; % initial Eb/No estimate
       delta_eb = abs(last1 - last2);
    end
      accrcy = PERreq*10^(-3); % PER accuracy
      delta = 0.1;
      diff = 1;
      olddir = 1;
      newdir = 1;
      z = 0;
      while (abs(diff) > accrcy)
```

```
            z = z+1;
            Eb_No = 10^(Eb_No_dB/10);

%------------------------------------------------------------------
      glower = 0;
      gupper = 20;

      Z1 = [num2str((1+K(p))/Eb_No) '*'];
      Z2 = ['exp(-(gama*' num2str(1+K(p)) '+' num2str(K(p)*Eb_No) ];
      Z3 = [')/' num2str(Eb_No) ').*'];

      if KdB < 230
            Z4 = ['besseli(0,sqrt((' num2str(4*(1+K(p))*K(p))
'*gama)/'];
            Z5 = [num2str(Eb_No) ')).*'];
      else
            b_arg = ['sqrt((' num2str(4*(1+K(p))*K(p)) '*gama)/'
num2str(Eb_No) ')'];
            Z4 = ['exp(' b_arg ')./sqrt(2*pi*' b_arg ').*'];
            t2 = ['(1 + 1./(8*' b_arg ') + 9./(2*8^2*' b_arg '.^2)
+'];
            t3 = ['9*25./(2*3*8^3*' b_arg '.^3) +'];
            t4 = ['9*25*49./(2*3*4*8^4*' b_arg '.^4) ).*'];
            Z5 = [ t2 t3 t4];
      end


      Z6 = '(1 - (1 - erfc(sqrt(gama))).^424)';

      Z = inline([Z1 Z2 Z3 Z4 Z5 Z6],'gama');
      result = quad8(Z,glower,gupper);

%------------------------------------------------------------------
            avgPER = result;
            diff = avgPER - PERreq;
            if (newdir ~= olddir)
                  delta = delta/2;
            end
            olddir = newdir;
            if( abs(diff) > accrcy )
                  if (diff > 0)
                        Eb_No_dB=Eb_No_dB+delta;
                        newdir = 1;
                  else
                        Eb_No_dB=Eb_No_dB-delta;
                        newdir = -1;
                  end

            end
      end
```

```
        fprintf('Number of iterations = %d\n', z)
        fprintf('Eb_No (dB) = %d\n',10*log10(Eb_No))
        fprintf('average BER = %d\n',avgPER)



        row = floor(round(KdB(p)*100)/100) + abs(lower) + 1
        tempcol = mod(round(abs(KdB(p))*100),100) ;
        if KdB(p) < 0
              if tempcol == 0
                    tempcol = 100;
              end
              col = round(101 - tempcol)
        else
              col = round(tempcol + 1)
        end

        newrow = row;
        if newrow == oldrow & col ==1
              fprintf('newrow = oldrow : row = %d\n',row)
              pause
        end
        oldrow = newrow;

        ebno_a(row,col) = 10*log10(Eb_No);
        last2 = last1;
        last1 = ebno_a(row,col);

  end


  save ebno_PER.mat ebno_a
```

13 Appendix G: Closing the Link for Average K and Path Loss (Matlab™ code)

```
%-------------------------------------------------------------------
%  This program calculates the required transmit power
%  necessary to close the link for average receive power
%  and average K value as a function of distance
%-------------------------------------------------------------------

clear

BERreq = 10^(-5);  % required average BER
x = [0.1:0.01:5]; %  distance in km
xdB = 10*log10(x);      %  log distance
awgn = 10*log10((erfinv(1-2*BERreq))^2)*ones(size(x));
rayl = 10*log10((1-2*BERreq)^2/(1-(1-2*BERreq)^2))*...
      ones(size(x));

region = 2;
      if region == 1
            x_avg = -3.0701;
            y_avg = 10*log10(4.9092);
            slope = -0.57032;
            ebno_a(1) = 20;
      elseif region == 2
            x_avg = -3.6645;
            y_avg = 10*log10(10.2758);
            slope = -0.11166;
            ebno_l(1) = 15;
      else
            x_avg = -2.7573;
            y_avg = 10*log10(2.0847);
            slope = -0.67997;
            ebno_o(1) = 20;
      end

      KdB = round((slope*(xdB-x_avg)+y_avg)*100)/100;
%-----------------------------------------------------
%  Determine required Eb/No for each K

      load ebno_BER.mat;
      for j_cntr = 1:length(KdB)
            found = 0;
            if KdB(j_cntr) < -10
                  ebno(j_cntr) = 44;  % 30.8 for 1 ATM cell PER
equiv
                  found = 1;
            end
            if KdB(j_cntr) > 29.99
                  ebno(j_cntr) = 9.6;
```

```
                  found = 1;
            end
            if found == 0

                  row = floor(round(KdB(j_cntr)*100)/100) + 11;
                  tempcol = mod(round(abs(KdB(j_cntr))*100),100) ;
                  if KdB(j_cntr) < 0
                        if tempcol == 0
                              tempcol = 100;
                        end
                        col = round(101 - tempcol);
                  else
                        col = round(tempcol + 1);
                  end

            ebno(j_cntr) = ebno_a(row,col);
            end

      end
fprintf('Got the Eb/No reqs\n')

%-------------------------------------------------------
%  Find required transmit power
%-------------------------------------------------------
Rd = 1*10^6;            %  data rate
Rd_dB = 10*log10(Rd);
M = 0;            %  link margin (dB)
Gt = 8;                 %  Tx antenna gain (dB)
k = -228.6;       %  Boltzman's constant (dB)
Gr = 0;                 %  Rx antenna gain (dB)
Lc = 3;                 %  Cable loss (dB)
lc = 10^(Lc/10);
Fr = 5;                 %  Rx noise figure (dB)
fr = 10^(Fr/10);
Ta = 290;         %  Antenna temp
Tl = 290;         %  Cable temp

Tr = 290*(fr-1);
Ts = 10*log10(Ta + Tl*(lc - 1) + Tr*lc);

x_los = -3.6645;
y_los = 115.2740;
los_slope = 2.232;

Ls_los = los_slope*(xdB-x_los)+y_los;

ERP_req = ebno + Rd_dB + M  + Ls_los + k - Gr + Ts;

% Multi-path gain for 2-ray model
```

```
h1 = 20;
h2 = 2;
dR = sqrt((h2+h1)^2+x.^2) - sqrt((h2-h1)^2+x.^2);
%psi = atan((h1+h2)/x);
gamma = -1; %  perfect reflection
rho = abs(gamma);
phi = -angle(gamma);
lambda = 3^8/5.8^9;
gm = 1 + 2*rho.*cos((2*pi*dR)/lambda+phi) +rho.^2;
Gm = 10*log10(gm);

Pt_2ray = ERP_req-Gm;

%  Determine available ERP
elements = 6;
PperElement = 30 ; % dBm
Pavail = 10*log10((elements * 10.^(PperElement/10))/1000) + Gt;  %
dBW



semilogx(x,ERP_req,'k--',x,Pt_2ray,'k')
axis([min(x) max(x) min(Pt_2ray - 5) max([max(Pt_2ray +5)
Pavail])+5])
title('Required Effective Radiated Power (LOS) : BER = 10^-^5')
xlabel('Distance (km)')
ylabel('ERP (dBW)')

line([min(x) max(x)], [Pavail Pavail])

text(0.12,35,['Available ERP = ' num2str(Pavail) ' dBW'])
text(0.15,30,['Power/element = ' num2str(PperElement) ' dBm'])
text(0.15,25,'6 antenna elements')
text(0.15,20,['Gt = ' num2str(Gt) ' dB'])

legend('Power Law Model','Two-Ray Model')

text(0.2,70,['Rd = ' num2str(Rd/10^6) ' Mbps'])
text(0.2,65,['Fr = ' num2str(Fr) ' dB'])
text(0.2,60,['Gr = ' num2str(Gr) ' dB'])


print -dtiff graphs/closelink/c_lnk_ex
```

14 Appendix H: Determination of Correlation Coefficient (Matlab™ code)

```
%-----------------------------------------------------------
%  Calculate correlation coefficient for K and shadowing
%-----------------------------------------------------------

%  column 1 is K, column 2 is shadowing

all_mat = [ 3.5   12.7574;  % 1
            4.4   1.0372;   % 2
            18.7  -20.2539; % 3
            3.7   5.5862;   % 4
            10    -9.0212;   % 6
            5     -3.4303;  % 7
            -2    5.4284;   % 8
            6.9   -9.6355;  % 9
            13.6  -16.9144; % 10
            6.5   3.3267;   % 11
            -1    7.8332;   % 12
            11.8  -4.4695;  % 13
            6     -3.6840;  % 14
            -1    2.5697;   % 15
            13.5  -28.5438; % 16
            1     12.2056;  % 17
            -0.6164  10.2447; % 19
            -2    18.9697;  % 21
            5     8.0463;   % 22
            10.94 -4.1564;  % 23
            12.91 -11.4359; % 25
            -1.7  12.9549;  % 28
            -9    19.7704;  % 30
            -0.7  4.5709;   % 31
            13    -17.6949; % 32
            4     -0.6091;  % 33
            6.808 -17.2694; % 35
            3.77  16.8320;  % 37
            -0.6435     4.9850 ]; % 38


allcorr = corrcoef(all_mat);  %  correlation coefficient
plot(all_mat(1:29,1),all_mat(1:29,2),'k*')
title('Determination of Correlation - ALL')
xlabel('K (dB)')
ylabel('shadowing (dB about mean)')
gtext(['rho = ' num2str(allcorr(2,1))])

print -dtiff graphs/correlation/allcor
```

15 Appendix I: Determination of Percent Coverage Area (Matlab™ code)

```
%------------------------------------------------
%   Determination of percent coverage area
%------------------------------------------------


clear

%-----------------------------
%  ERPs to be checked

ERP_v = [10 20 30 40]; %  dBW

for erp_cntr = 1:length(ERP_v)   %  For loop for erps

%--------------------------------
%  distances to be checked

r = 0.01:0.01:2.01;
r_dB = 10*log10(r);
%---------------------------------

%-------------------------------------
%  Parameters for link calculation

Rd = 10^6;        %  data rate
Rd_dB = 10*log10(Rd);
M = 0;            %  link margin (dB)
k_boltz = -228.6; %  Boltzman’s constant (dB)
Lc = 3;                 %  Cable loss (dB)
lc = 10^(Lc/10);
Fr = 5;                 %  Rx noise figure (dB)
fr = 10^(Fr/10);
Ta = 290;         %  Antenna temp
Tl = 290;         %  Cable temp

Tr = 290*(fr-1);
Ts = 10*log10(Ta + Tl*(lc - 1) + Tr*lc);
%-----------------------------------------

%########################################################
%  Section to be changed for each case (LOS, OBS, ALL)

%----------------------------------------------------
%  Find average receive power at the given distances

ERP = ERP_v(erp_cntr);
pl_exp = 3.422;  % (LOS 2.232) (OBS 3.8108)(ALL 3.422);
```

```
pl_int =  139.9465;  % (LOS 123.4530) (OBS 147.4042) (ALL 139.9465)
sP = 12.4788;  % (LOS 9.0272) (OBS 8.1661) (ALL 12.4788);
muP = ERP - (pl_exp*r_dB + pl_int);
%-----------------------------------------------------

%-------------------------------------------
%  Find average K at the given distances

K_exp =  -0.57032;  % (LOS -0.11166)  (OBS -0.67997) (ALL -0.57032)
K_int =   3.1583;  % (LOS 9.8666)  (OBS 0.2099) (ALL 3.1583);
sK =  6.0958;  % (LOS 5.5547)  (OBS 4.4164) (ALL 6.0958);
muK = K_exp*r_dB + K_int;
%-------------------------------------------

%------------------------------------------
%  Correlation coefficient

rho = 0.8427;  % (LOS  0.88327) (OBS 0.6622)  (ALL 0.8427)

%###########################################################

for i_cntr = 1:length(r)  %  loop for radius

fprintf('\nCorrelated  --  ALL\n\n')
fprintf('ERP is %d\n',ERP_v(erp_cntr))
fprintf('radius is %d\n',r(i_cntr))

%-----------------------------------------------------
%  Determine Ks to be used at a specific distance

delta_K = 1;

KdB =  floor(100*(muK(i_cntr) - 3*sK))/100 :delta_K: ...
      ceil(100*(muK(i_cntr)+3*sK))/100;

fprintf('Got the Ks\n')
%------------------------------------------------------

%------------------------------------------------------
%  Determine required Eb/No for each K

      load ebno_PER.mat;
      for j_cntr = 1:length(KdB)
            found = 0;
            if KdB(j_cntr) < -10
                  ebno(j_cntr) = 30.8;  % 30.8 for 1 ATM cell PER
equiv
                  found = 1;
            end
            if KdB(j_cntr) > 29.99
```

```
                  ebno(j_cntr) = 9.6;
                  found = 1;
            end
            if found == 0

                  row = floor(round(KdB(j_cntr)*100)/100) + 11;
                  tempcol = mod(round(abs(KdB(j_cntr))*100),100) ;
                  if KdB(j_cntr) < 0
                        if tempcol == 0
                              tempcol = 100;
                        end
                        col = round(101 - tempcol);
                  else
                        col = round(tempcol + 1);
                  end

            ebno(j_cntr) = ebno_a(row,col);
            end

      end
fprintf('Got the Eb/No reqs\n')
%-------------------------------------------------------

%----------------------------------------------------------------
%  Determine required receive power level

gamma_req = ebno + Rd_dB + M + k_boltz + Ts;
%----------------------------------------------------------------

%-------------------------------------------------
%  for loop for integration over each K

fprintf('Number of integrations is %d\n',length(gamma_req))
for j_cntr = 1:length(KdB)

if (mod(j_cntr,100) == 0)
fprintf('j_cntr is %d\n',j_cntr);
end

      Prupper = muP(i_cntr) + 3*sP;
      Prlower = gamma_req(j_cntr);

      if Prupper > Prlower

            Z1 = ['exp((-1/(2*(1-' num2str(rho) '^2)))*'];

            Z2 = ['(((Pr-' num2str(muP(i_cntr)) ').^2/' ];

            Z3 = [num2str(sP) '^2)-((2*' num2str(rho) '*'];
```

```
            Z4 = ['(Pr-' num2str(muP(i_cntr)) ').*('];

            Z5 = [ num2str(KdB(j_cntr)) '-' num2str(muK(i_cntr))];

            Z6 = [ '))/(' num2str(sP) '*'];

            Z7 = [ num2str(sK) '))+((' num2str(KdB(j_cntr)) '-' ];

            Z8 = [num2str(muK(i_cntr))];

            Z9 = [ ').^2/(' num2str(sK) '^2))))'] ;

            Z10 = ['/(2*pi*' num2str(sP) '*' num2str(sK) ];

            Z11  = ['*(sqrt(1-' num2str(rho) '^2)))'];


            Z = inline([Z1 Z2 Z3 Z4 Z5 Z6 Z7 Z8 Z9 Z10 Z11],'Pr');

            result = quad8(Z,Prlower,Prupper);

      else
            result = 0;
      end

      f_PandK(j_cntr) = result;
end

%------ End of for integration for each K interval--------------

fprintf('Done integrating\n')

%----------------------------------------------
%  Calulate probability of closing the link for a
%   specific distance

P_closed(erp_cntr,i_cntr) = delta_K*sum(f_PandK);

end
%--------- End of for closing link at each distance ----------


%-------------------------------------------------------
%  Calculate percent coverage area using each distance
%   as a cell boundary

term(1) = r(1)^2*P_closed(erp_cntr,1);
for m_cntr = 2:length(r)
  term(m_cntr) = (r(m_cntr)^2 - r(m_cntr-
1)^2)*P_closed(erp_cntr,m_cntr);
```

```
end
for n_cntr = 1:length(r)
  fraction(erp_cntr,n_cntr) = sum(term(1:n_cntr))/r(n_cntr)^2;
end

end % end of ERP loop

Pc_P_A_corr = P_closed;
Frac_P_A_corr = fraction;

save p_p_a_c.mat Pc_P_A_corr
save f_p_a_c.mat Frac_P_A_corr



semilogx(r,fraction(1,:)*100,r,fraction(2,:)*100, …
r,fraction(3,:)*100,r,fraction(4,:)*100)


grid on
title('Percent Coverage Area : ALL')
xlabel('Cell boundary (km)')
ylabel('% coverage')
gtext('K and Pr correlated')
gtext(['ERP = ' num2str(ERP_v(1)) ' dBW'])
gtext([num2str(ERP_v(2)) ' dBW'])
gtext([num2str(ERP_v(3)) ' dBW'])
gtext([num2str(ERP_v(4)) ' dBW'])

%print -deps -tiff graphs/cov2
```

16 Appendix J: Test for Normality (Matlab™ code)

```
%  Cramer-Von Mises Test for Goodness of fit

clear

% x is the vector of values ( K deviation or shadowing)

x = [ -4.3572  -1.0905  13.4513  -2.1304  6.43982  1.2890 ...
      -5.9041  1.7082  6.6914  0.8112  -6.5549  6.6639 ...
      1.7363  -4.7110  10.6662  -2.1336  -4.2065  -5.7110 ...
      -1.404  2.5301  4.6983  -6.6252  -13.5944  -6.1278 ...
      9.6081  -1.4278  3.7469  0.4320  -4.7570 ];


x = sort(x);
n = length(x);
m = mean(x);
s = std(x);
w = (x-m)/s;

cdf(1) = 1/n;
for k = 2:n
      cdf(k) = cdf(k-1)+1/n;
end

z = 1 - 0.5*erfc(w/sqrt(2));

W2 = 0;
for k = 1:n
      W2 = W2 + (z(k) - (2*k-1)/(2*n))^2;
end
W2 = W2 + 1/(12*n);
W2mod = W2*(1+0.5/n);

x1 = -15:0.1:15;
w1 = (x1-m)/s;
z1 = 1 - 0.5*erfc(w1/sqrt(2));

plot(x,cdf,'k*',x1,z1,'k')
title('CDF of K deviation')
xlabel('K deviation about the mean (dB)')
ylabel('probability <= abscissa')
text(-10,0.9,['C-vM stat = ' num2str(W2mod)])
text(-10,0.8,['Mean = ' num2str(m) ' dB'])
text(-10,0.7,['Sigma = ' num2str(s) ' dB'])

%print -dtiff graphs/logn_Kdev
```

17 Appendix K: Construction of Bivariate Gaussian Graph (Matlab™ code)

```
%-------------------------------------------------------------
%  This program plots the joint pdf of K and receive power
%-------------------------------------------------------------

clear

r = 0.4;  %  distance to be checked in km
r_dB = 10*log10(r);

%-------------------------------------
%  Parameters for link calculation

Rd = 10^6;        %  data rate
Rd_dB = 10*log10(Rd);
M = 0;            %  link margin (dB)
k_boltz = -228.6; %  Boltzman’s constant (dB)
Lc = 3;                 %  Cable loss (dB)
lc = 10^(Lc/10);
Fr = 5;                 %  Rx noise figure (dB)
fr = 10^(Fr/10);
Ta = 290;         %  Antenna temp
Tl = 290;         %  Cable temp

Tr = 290*(fr-1);
Ts = 10*log10(Ta + Tl*(lc - 1) + Tr*lc);
%-----------------------------------------

%#########################################################
%  Section to be changed for each case (LOS, OBS, ALL)

%-----------------------------------------------------
%  Find average receive power at the given distances

ERP = 10;
pl_exp = 3.422;  % (LOS 2.232) (OBS 9.0272)(ALL 3.422);
pl_int =  139.9465;  % (LOS 123.4530) (OBS 147.4042) (ALL 139.9465)
sP = 12.4788;  % (LOS 9.0272) (OBS 8.1661) (ALL 12.4788);
muP = ERP - (pl_exp*r_dB + pl_int);
%-----------------------------------------------------

%-------------------------------------------
%  Find average K at the given distances
```

```
K_exp =  -0.57032;  % (LOS -0.11166)  (OBS -0.67997) (ALL -0.57032)
K_int =   3.1583;  % (LOS 9.8666)  (OBS 0.2099) (ALL 3.1583);
sK =  6.0958;  % (LOS 5.5547)  (OBS 4.4164) (ALL 6.0958);
muK = K_exp*r_dB + K_int;
%-------------------------------------------

%------------------------------------------
%  Correlation coefficient

rho = 0;  % (LOS  0.88327) (OBS 0.6622)  (ALL 0.8427)

%###########################################################
%----------------------------------------------------
%  Determine Ks to be used at a specific distance

delta_K = 1;

KdB =  floor(100*(muK - 3*sK))/100 :delta_K: ...
      ceil(100*(muK+3*sK))/100;

fprintf('Got the Ks\n')
%-----------------------------------------------------
%-----------------------------------------------------
%  Determine required Eb/No for each K

      load ebno_BER.mat;
      for j_cntr = 1:length(KdB)
            found = 0;
            if KdB(j_cntr) < -10
                  ebno(j_cntr) = 44;  % 30.8 for 1 ATM cell PER
equiv
                  found = 1;
            end
            if KdB(j_cntr) > 29.99
                  ebno(j_cntr) = 9.6;
                  found = 1;
            end
            if found == 0

                  row = floor(round(KdB(j_cntr)*100)/100) + 11;
                  tempcol = mod(round(abs(KdB(j_cntr))*100),100) ;
                  if KdB(j_cntr) < 0
                        if tempcol == 0
                              tempcol = 100;
                        end
                        col = round(101 - tempcol);
                  else
                        col = round(tempcol + 1);
                  end
                  ebno(j_cntr) = ebno_a(row,col);
```

```
            end

      end
fprintf('Got the Eb/No reqs\n')
%-------------------------------------------------------

%----------------------------------------------------------------
%  Determine required receive power level

gamma_req = ebno + Rd_dB + M + k_boltz + Ts;
%----------------------------------------------------------------

%----------------------------------------
%  Determine range of random variables

 x = -20:0.5:30;
 x = x(ones(1,length(x)),:);
 y = x*2-110;
 y = y';

sx = sK;
sy = sP;
mx = muK(1);
my=muP(1);

%------------------------------------------------
%  calulate bivariate gaussian pdf for each point

z = 1/(2*pi*sx*sy*sqrt(1 - rho^2))*exp(-1/(2*(1 - rho^2)) ...
      *(((x - mx)/sx).^2 + ((y - my)/sy).^2 - 2*rho*(x - mx).*(y -
my)/sx/sy));

%--------------------------------------
%   3-D plot the pdf

mesh(x,y,z)
set(gca,'xlim',[-20 30],'ylim',[-150 -60])

%--------------------------------------
%  Plot required receive power

no_of_lines = 10;
for j = 1:no_of_lines
      hold on
      plot3(KdB,gamma_req,ones(1,length(KdB))*0.00175*(j-
1)/no_of_lines,'k','linewi',2)
end
```

```
title('Bivariate Gaussian PDF : rho = 0')
xlabel('K (dB)')
ylabel('Receive Power (dB)')
zlabel('f(K,Receive Power)')
view(-15,30)
text(-20,-65,0.0023,'All Locations')
text(-20,-65,0.0020,'Distance = 400 m')
text(-20,-65,0.0017,'ERP = 10 dBW')
text(-20,-65,0.0014,'Data Rate = 1 Mbps')
text(-20,-65,0.0011,'Fr = 5 dB')
text(-20,-65,0.0008,'BER = 10^-^5')
text(17,-125,0.0018,'Receive Threshold')

print -dtiff graphs/jointpdf/clr_in400

colormap(gray)
print -dtiff graphs/jointpdf/bw_in400
```

18 Appendix L: Difference of Two Independent Gaussians

Let $Y = X_1 + X_2$, where X_1 and X_2 are independent, Gaussian random variables with means μ_1, μ_2 and variances σ_1^2, σ_2^2, respectively. It is well known that the distribution for Y is a Gaussian with mean $\mu_1 + \mu_2$ and variance $\sigma_1^2 + \sigma_2^2$. To find the difference of two independent Gaussians, it is only necessary to make a simple substitution.

Let $X_2 = -X_3$. Since only the signs of the possible outcomes have changed, the variance remains the same and the mean changes sign. The distribution of X_3 is then Gaussian with $\mu_3 = -\mu_2$ and $\sigma_3^2 = \sigma_2^2$. Since $Y = X_1 + X_2 = X_1 - X_3$, Y is Gaussian with mean $\mu_Y = \mu_1 + \mu_2 = \mu_1 - \mu_3$ and variance $\sigma_Y^2 = \sigma_1^2 + \sigma_2^2 = \sigma_1^2 + \sigma_3^2$,

The pdfs of Y can then be written as:

$Y = X_1 + X_2$,

$$f_Y(x_1, x_2) = \frac{1}{\sqrt{2 \cdot \pi \cdot (\sigma_1^2 + \sigma_2^2)}} \cdot \exp\left\{ -\frac{[y - (\mu_1 + \mu_2)]^2}{2 \cdot (\sigma_1^1 + \sigma_2^2)} \right\}$$

$Y = X_1 - X_2$,

$$f_Y(x_1, x_2) = \frac{1}{\sqrt{2 \cdot \pi \cdot (\sigma_1^2 + \sigma_2^2)}} \cdot \exp\left\{ -\frac{[y - (\mu_1 - \mu_2)]^2}{2 \cdot (\sigma_1^1 + \sigma_2^2)} \right\}$$

19 Appendix M: Abbreviations and Acronyms

BER	Bit Error Rate
CDF	Cumulative Distribution Function
DSP	Digital Signal Processing
ERP	Effective Radiated Power
FCC	Federal Communications Commission
GPS	Global Positioning System
HIPERLAN	High Performance Local Area Network
ISM	Industrial, Scientific and Medical
LAN	Local Area Network
LOS	Line Of Sight
OBS	Obstructed
pdf	Probability Density Function
PER	Packet Error Rate
QoS	Quality of Service
RDRN	Rapidly Deployable Radio Network
U-NII	Unlicensed National Information Infrastructure
WATM	Wireless Asynchronous transfer Mode
WLAN	Wireless Local Area Network
WLL	Wireless Local Loop

20 Appendix N: Measurement System Operation

Start Retki Land Navigation from the program group. The GPS navigation window will appear. The map software was never loaded so don't be concerned that the window consists only of the toolbar with nothing in the workspace. Click on the GPS enable/disable icon. Once the receiver is enabled, click on the GPS information icon. This will open the window that gives all of the status information for the receiver. When the receiver is first enabled, the status window will show 0D Acquiring Satellites. As the satellites are located, the status will change. Once the receiver locates and is able to track four satellites, the status will become 3D Navigating. This means that the location data is valid for latitude, longitude and altitude. Once the receiver is navigating, close the information window and make the main window an icon. Run from the command line, *combo1.exe out.txt cfreq span reflevel*, where *combo1.exe* is the data collection program, *out.txt* is the output file name, *cfreq* is the center frequency, *span* is the initial span used by the spectrum analyzer and *reflevel* is the reference level to be used by the spectrum analyzer. When the program is finished collecting data, the user will be prompted with *All done-press any key to finish program*. The window will close and the data can be found in the output file specified on the command line.

The estimated time for the program to run is about one sample/second. This can be increased to four samples/second if the spectrum analyzer is not run in zero span. Although the spectrum analyzer can actually operate faster than this, the time it takes for the spectrum analyzer to send the data to the laptop is about 200 mS.