MURI Network Layer Simulation and Optimization

Hua Wang

Department of Electrical Engineering and Computer Science

University of Michigan

Ann Arbor, MI 48109

May 26, 1999

Contents

1 Overview of Simulation and Optimization **2**

2 Network Layer Simulation and Optimization **2**

2.1 Network Scenario . . . 3

2.1.1 Drift Model . . . 4

2.1.2 Broadcast Model . . . 4

2.1.3 Antenna Model . . . 5

2.1.4 Propagation Model . . . 5

2.1.5 Estimation Model . . . 16

2.1.6 Energy Constraint . . . 17

2.1.7 Optimization Problem Formulation . . . 17

2.2 Modeling using *OPNET* Software . . . 18

2.3 Simulated Annealing . . . 20

2.4 Integration of Simulation and Optimization . . . 20

3 Preliminary Simulation Results **22**

4 Analysis of Simplified Network Scenario **26**

5 Conclusions **29**

6 Proposed Power Adjustification **29**

7 Future Work **31**

List of Figures

1 MURI Network Simulation Scenario . . . 3

2 Bit Error Rate as a Function of Distance . . . 8

3 Packet Error Rate as a Function of Distance . . . 8
4 Bit Error Rate as a Function of Distance . . . 9
5 Packet Error Rate as a Function of Distance . . . 9
6 Bit Error Rate as a Function of Distance . . . 10
7 Packet Error Rate as a Function of Distance . . . 10
8 Bit Error Rate as a Function of Distance . . . 11
9 Packet Error Rate as a Function of Distance . . . 11
10 Bit Error Rate as a Function of Distance . . . 12
11 Packet Error Rate as a Function of Distance . . . 12
12 E_b/N_0 as a Function of Distance . . . 13
13 E_b/N_0 as a Function of Distance . . . 13
14 E_b/N_0 as a Function of Distance . . . 14
15 E_b/N_0 as a Function of Distance . . . 14
16 E_b/N_0 as a Function of Distance . . . 15
17 Binary Phase Shift Keying (bpsk) BER . . . 15
18 Network Level Modeling . . . 19
19 Node Level Modeling . . . 19
20 Process Level Modeling . . . 20
21 MURI Network Simulation and Optimization Setup . . . 21
22 MURI Network Simulation and Optimization Using Parallel Structure . . . 22
23 drop zone $1km \times 1km$, estimated position error versus energy constraint . . . 23
24 drop zone $1km \times 1km$, estimated position error versus energy constraint . . . 24
25 With drop zone $1km \times 1km$ and energy constraint $2 \times 10^{-2}J$ . . . 25
26 With $T = 8s$, drop zone $1km \times 1km$, and energy constraint $2 \times 10^{-2}J$ . . . 25

List of Tables

1 **Symbols Used in Packet Error Rate Calculation** 6

MURI Network Layer Simulation and Optimization

Abstract

The Multidisciplinary University Research Initiative (MURI) is a simulation and optimization study of future army communication system from a low energy perspective. The overall communication system can be partitioned into three different layers, namely, network layer, processing layer, and device layer. The ultimate goal of MURI is to achieve global optimization of these three layers and demonstrate the methodology of approaching problems of such large scale.

At present, local simulation and optimization has been applied at each layer individually. At network layer in particular, we chose *OPNET* as our network simulation tool and *simulated annealing* as our optimizer. A very simple scenario has been implemented for the initial approach, where several mobile nodes, initially deployed in a drop zone, travel towards the same destination and broadcast position awareness packets under certain energy constraint. Each node gives estimates of all other nodes' positions periodically. The performance criteria of the system is chosen to be the root mean squared error between a node's estimated position and its real position, and is minimized over the parameter set (T, q, P), where T is the interval between successive transmissions, q is the retransmission probability, and P is the transmission power.

We have got some preliminary results of performance versus energy constraint curve, which agree with what we have expected.

Keywords: MURI, OPNET, Simulated Annealing

1 Overview of Simulation and Optimization

The Multidisciplinary University Research Initiative (MURI) undertaken at the University of Michigan is to investigate the various algorithms and protocols of future army communication systems from a low energy perspective. The system we are investigating may be partitioned into three layers, namely network layer, processing layer, and device layer. The ultimate goal of MURI is the global simulation and optimization of the overall communication system to achieve a performance versus energy consumption curve that may involve different layers altogether at the same time and the demonstration of methodology of approaching problems of such large scale.

To find a mathematical expression of performance measure as a function of parameters for the overall communication system is generally not possible, we therefore have to rely on the integration of simulation and optimization. Even so, it is still extremely hard to integrate simulation and optimization over all these three layers at once since it usually involves too huge dimension of freedom on the parameters and it takes much too long to run simulation. As a consequence, an integration of optimization and simulation has to be taken at each different layers individually at the moment in order to achieve some understanding of performance versus energy consumption.

In the following sections, more details of simulation and optimization at the network layer will be given. As we shall see, the simulation and optimization at network layer alone is already very time consuming.

2 Network Layer Simulation and Optimization

As with any simulation and optimization problem, we face the choice of simulation and optimization tools that we should use. We use *OPNET* package as our simulation tool and *simulated annealing* program as our optimizer. One of the reasons we chose the above software is that *OPNET* is becoming dominant in industry and military, and simulated annealing has great potential in finding global minimum. A very simple network scenario has been implemented and the integration of *OPNET* simulation and the *simulated annealing* has been applied to this network model to measure the performance of the communication protocols at the network layer.

2.1 Network Scenario

At the network layer, we started with a simple scenario which can be defined as a position awareness task as shown in Figure 1.

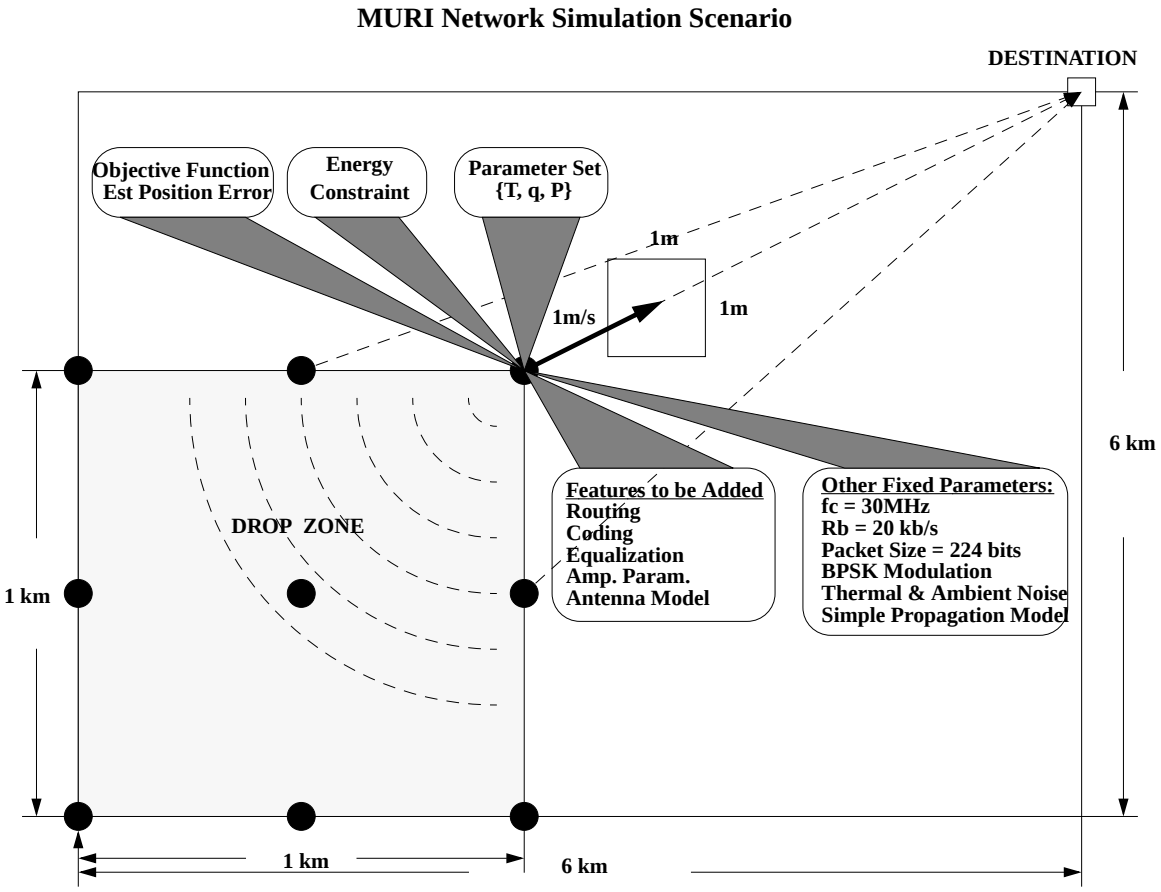

Figure 1: MURI Network Simulation Scenario

As shown in the figure, multiple mobile nodes are initially deployed in a drop zone of area $1km \times 1km$ and they move towards the same destination according to a drift model at a speed of $1m/s$. The destination is currently set at coordinates $(gx, gy) = (6000, 6000)m$. While moving towards the destination, each node broadcasts its position awareness packets periodically. The packet, undergoing a propagation model, may or may not reach all other mobile nodes. A node also gives position estimates of all other nodes periodically. The average root mean squared error between the a node's estimate and its actual position is obtained as the performance measure of the network layer. In the following subsections, the network model is further divided into drift model, broadcast model, propagation model, estimation model, and energy constraint for detailed explanation.

2.1.1 Drift Model

Each mobile node moves towards the destination according to a drift model described below.

Each node moves every T_m, where $T_m = 1s$ in the current network setup. Suppose node 1's position at time kT_m is defined by a vector (x_k, y_k), then node 1's position at time $(k+1)T_m$ is calculated as follows:

Node 1 first calculates a normalized direction vector from its position at time kT_m toward the destination. Let us denote the normalized direction vector by (dx, dy) and use the above notation (gx, gy) for the destination, then the direction vector is calculated by

$$\begin{cases} dx = \frac{gx - x_k}{d} \\ dy = \frac{gy - y_k}{d} \end{cases} \tag{1}$$

where,

$$d = \sqrt{(gx - x_k)^2 + (gy - y_k)^2} \tag{2}$$

Each node travels at a certain speed v, where $v = 1m/s$ in current network model. Besides calculating the direction vector, a node also generates a uniform random vector (rx, ry), where rx, ry are outcomes of uniform random generator with range $(-0.5, 0.5)$.

At time index $(k+1)T_m$ node 1 will move to

$$\begin{cases} x_{k+1} = vT_m(dx + rx) \\ y_{k+1} = vT_m(dy + ry) \end{cases} \tag{3}$$

OPNET has the capability of modeling continuous movement, however, in our current network model, each node jumps from one position to another position every T_m.

2.1.2 Broadcast Model

The current communication network model uses time division multiple access (TDMA) and Binary Phase Shift Keying (BPSK) modulation, where each node is assigned a fixed time slot to broadcast its position awareness packets. The packet has length $L = 224bits.$ and data rate $R_b = 20kb/s$. Therefore the duration of each bit of the packet $T_b = 1/R_b$. There is also a predetermined retransmission probability q and battery capacity E imposed on each node.

Suppose it comes to node 1's time slot to broadcast its position awareness packets, node 1 first checks whether its cumulative consumed energy is greater than the battery capacity and only transmits packet if its consumed energy is less than its battery capacity. Node 1 transmits the packet with a fixed power P, which does not depend on the distances separating node 1 from all other nodes. It increases its consumed energy by PT_bL whenever a packet is transmitted.

After its first transmission, as long as its consumed energy is less than the battery capacity and there is still enough time left in its time slot for the duration of one packet, node 1 generates a random number r uniformly distributed on the interval $(0, 1)$. If $r < q$, it retransmits the same packet again and increased its consumed energy by PT_bL after each retransmission. Node 1 stays in above loop until any one of the conditions breaks.

2.1.3 Antenna Model

A dipole antenna (whip antenna) is used at the current network layer. Each node has an antenna of height $1m$. Since all the nodes are on the same altitude and each node transmits packets using a carrier frequency of $30MHz$, the corresponding antenna efficiency is 14.814%, and the directivity gain is $1.775dB$. The overall antenna gain is therefore $-6.518dB$, which means that the actual radiated power out of antenna is $6.518dB$ less than the input power into antenna.

2.1.4 Propagation Model

We assume that we have a propagation model that takes into account a direct path from transmitter to receiver, and a reflection path with 180 degree phase change at the reflection point from transmitter to receiver as well. The cumulative effect of both paths give us an relation between received power and transmitted power, which is given in Eqn (6). The meaning of all the parameters are given in Table 1.

We need to get a rough idea of how packet error rate is related to the transmitted power before we can set up reasonable energy constraint on each mobile node. Based on our assumptions of the propagation model, we can calculate bit error rate (BER) and packet error rate (PER) as a function of the transmitted power (the power radiated from antenna) and the propagation distance between the transmitter and the receiver.

Since we are only considering thermal noise at the moment without counting any other noise source and thermal

noise level is very low, we don't actually need too much transmitted power to get a very low packet error rate.

Table 1: **Symbols Used in Packet Error Rate Calculation**

Symbol	Meaning	Value	Unit
k	Boltzmann constant	$1.379 \cdot 10^{-23}$	$W/Hz/K$
C	propagation speed	$3.0 \cdot 10^{8}$	m/s
T	background temperature	290	K
N_0	one-sided power spectral density	Eqn (4)	W/Hz
f_c	carrier frequency	30	MHz
λ_c	carrier wavelength	Eqn (5)	m
bw	bandwidth	10	kHz
h_t	height of transmitter antenna	1	m
h_r	height of receiver antenna	1	m
G_t	gain of transmitter antenna	1	$none$
G_r	gain of receiver antenna	1	$none$
R_b	data rate	20	kb/s
T_b	data symbol duration	Eqn (7)	s
L	packet length	224	bit
P_t	transmitted power	$[10^{-7}, 1]$	W
d	propagation distance	$(0, 9000)$	m
P_r	received power	Eqn (6)	W
BER	bit error rate	Eqn (9)	$none$
PER	packet error rate	Eqn (10)	$none$

$$N_0 = kT = 1.379 \cdot 10^{-23} \cdot 290 = 3.9991 \cdot 10^{-21} \quad W/Hz \tag{4}$$

$$\lambda_c = \frac{C}{f_c} = \frac{3.0 \cdot 10^8}{30 \cdot 10^6} = 10 \quad m \tag{5}$$

$$P_r = P_t 4 \left(\frac{\lambda_c}{4\pi d}\right)^2 G_t G_r sin^2 \left(\frac{2\pi h_t h_r}{\lambda_c d}\right) = P_t 4 \left(\frac{10}{4\pi d}\right)^2 sin^2 \left(\frac{2\pi}{10d}\right) \tag{6}$$

$$T_b = \frac{1}{R_b} = \frac{1}{20 \cdot 10^3} = 0.00005 \quad s \tag{7}$$

$$E_b = P_r T_b = 0.00005 Pr \quad J \tag{8}$$

$$BER = Q\left(\sqrt{\frac{2E_b}{N_0}}\right) = Q\left(\sqrt{\frac{2 \cdot 0.00005 Pr}{3.9991 \cdot 10^{-21}}}\right) = Q\left(\sqrt{2.5006 \cdot 10^{16} Pr}\right) \tag{9}$$

Since the communication system is uncoded, we have

$$PER = 1 - (1 - BER)^L \tag{10}$$

Based on the above formulae, we could plot the following set of figures indicating the relationship of PER (or BER) as a function of transmitted power and propagation distance. We could use these theoretical results as a guide to set reasonable energy constraint of our interest region on each node and to design our future power allocation strategy.

Since BPSK modulation is used in our current network layer, Figure 17 shows BER versus E_b/N_0 of a BPSK system. From Figure 12 to Figure 16, which shows the relationship between E_b/N_0 at the receiver and propagation distance and Figure 17, we can derive the relationship of BER and PER versus propagation distance as shown from Figure 2 to Figure 10.

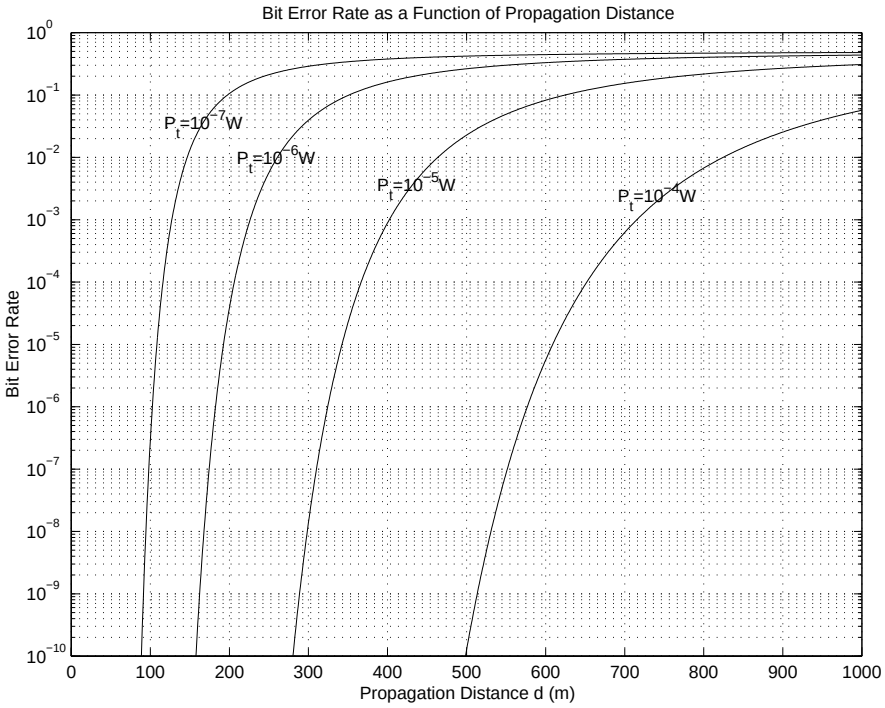

Figure 2: Bit Error Rate as a Function of Distance

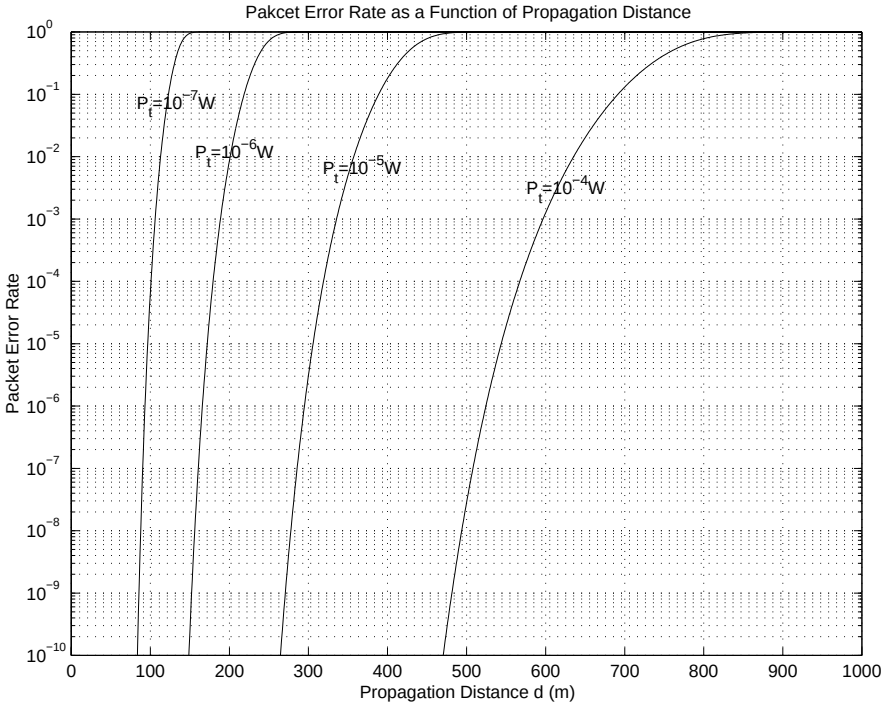

Figure 3: Packet Error Rate as a Function of Distance

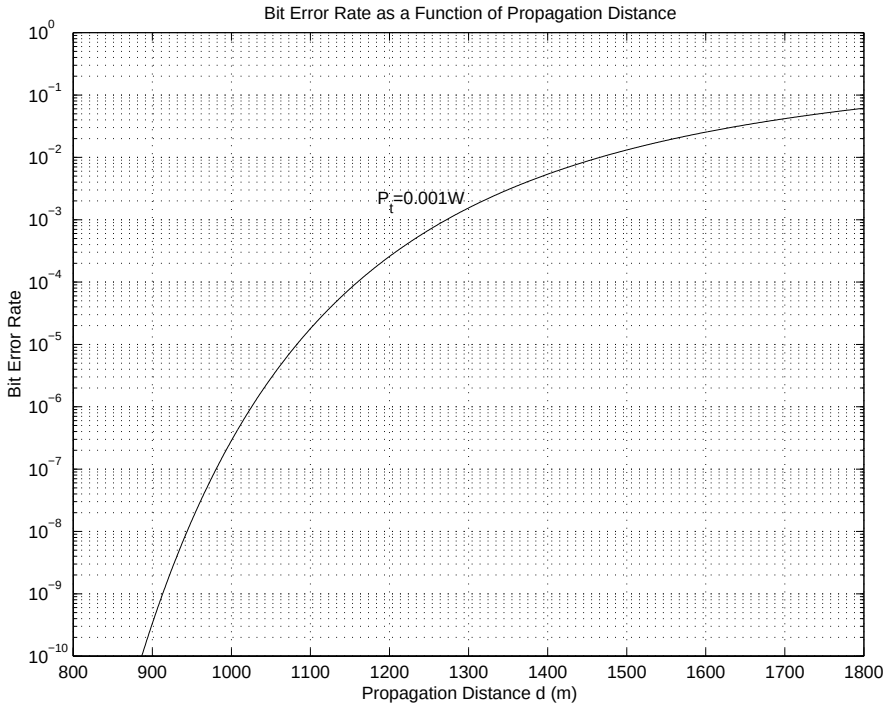

Figure 4: Bit Error Rate as a Function of Distance

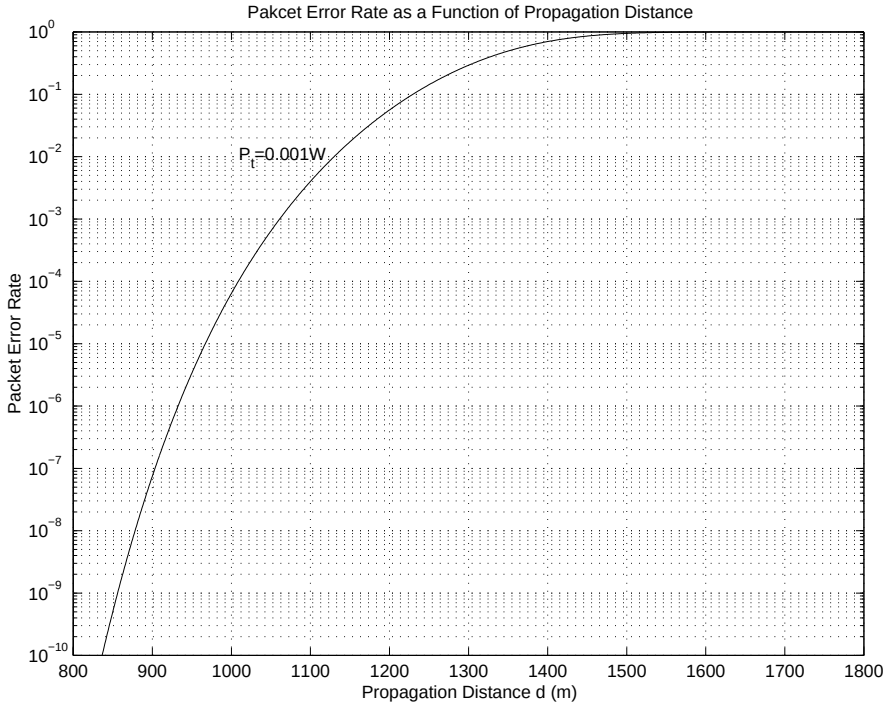

Figure 5: Packet Error Rate as a Function of Distance

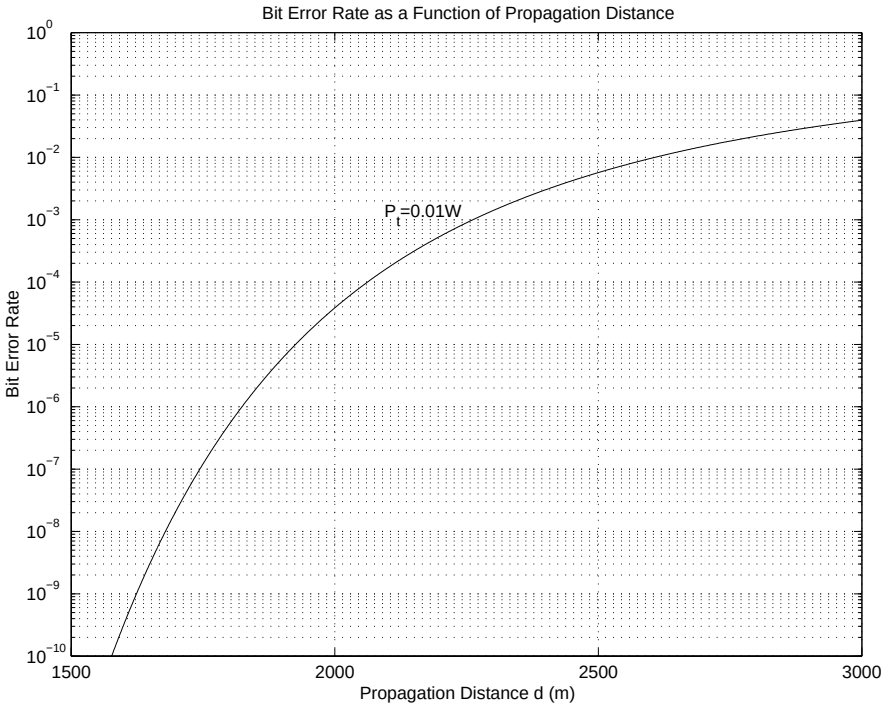

Figure 6: Bit Error Rate as a Function of Distance

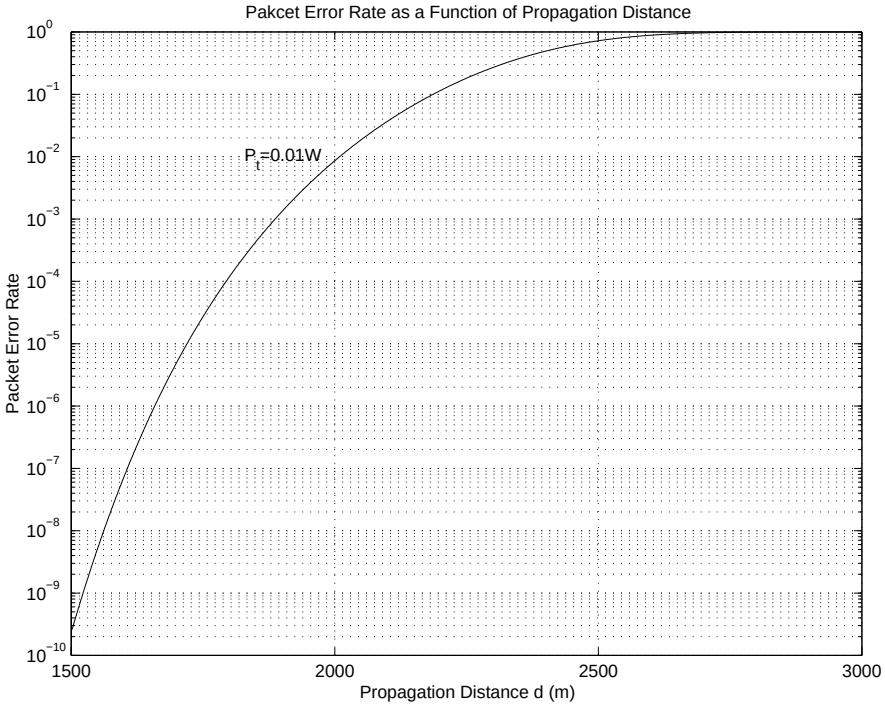

Figure 7: Packet Error Rate as a Function of Distance

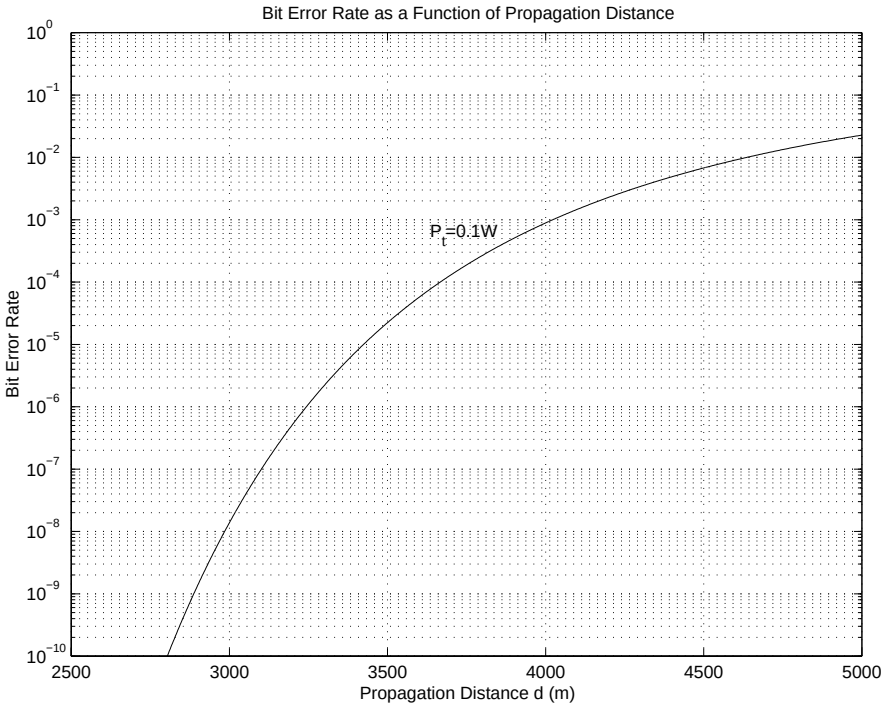

Figure 8: Bit Error Rate as a Function of Distance

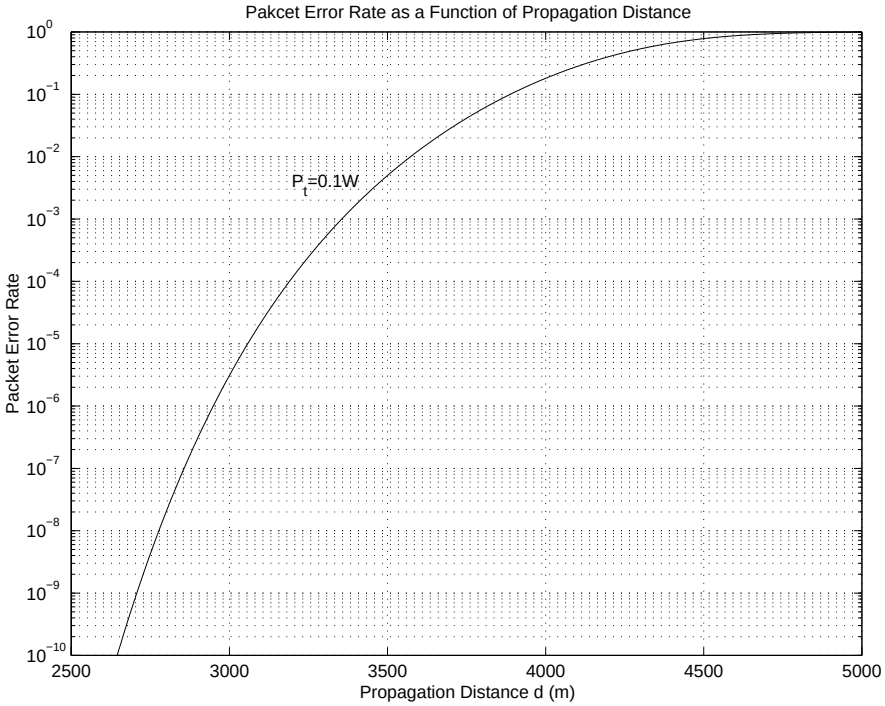

Figure 9: Packet Error Rate as a Function of Distance

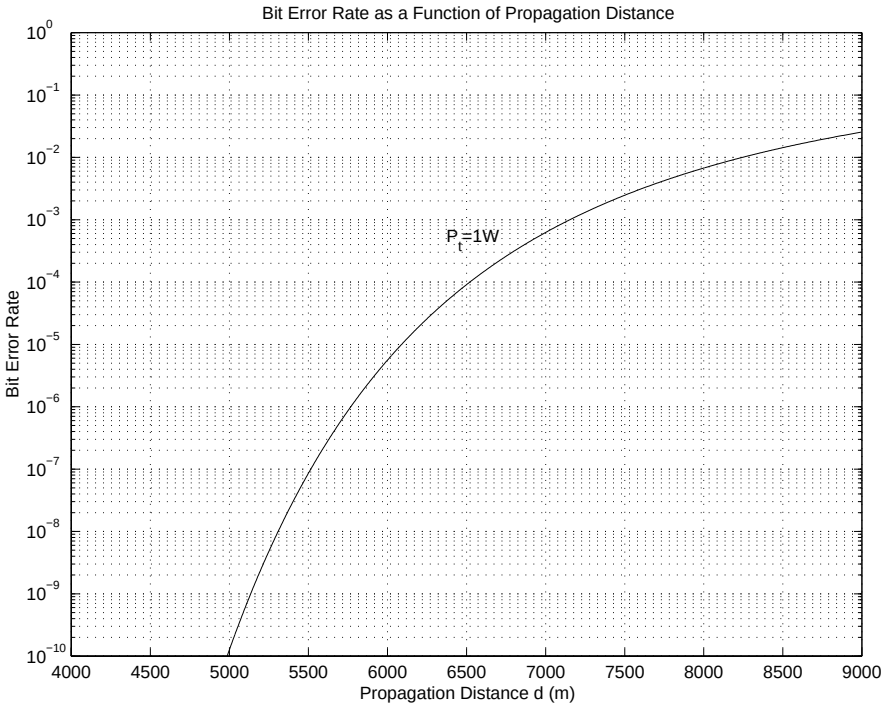

Figure 10: Bit Error Rate as a Function of Distance

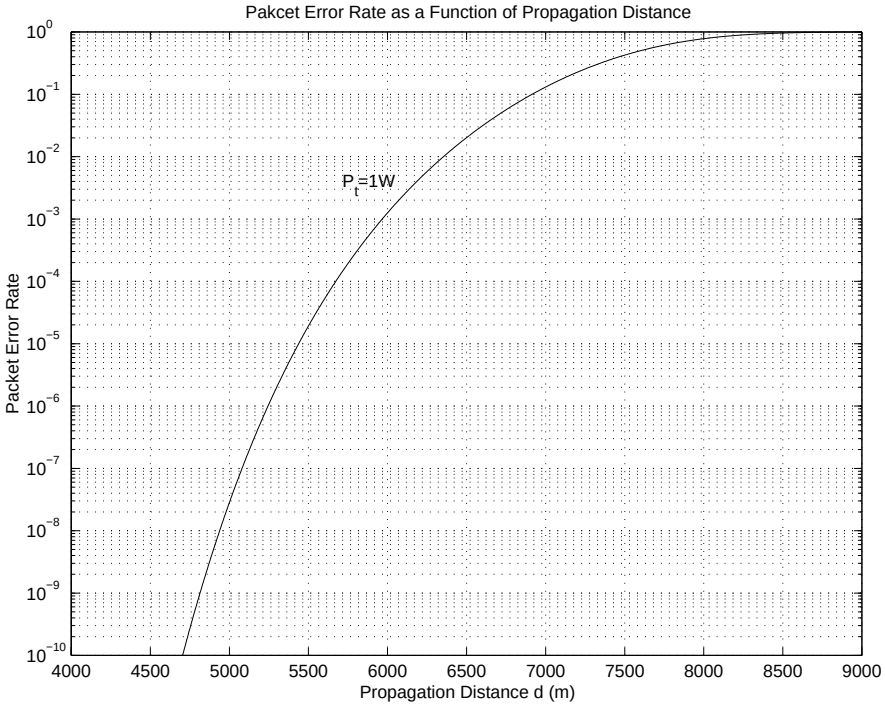

Figure 11: Packet Error Rate as a Function of Distance

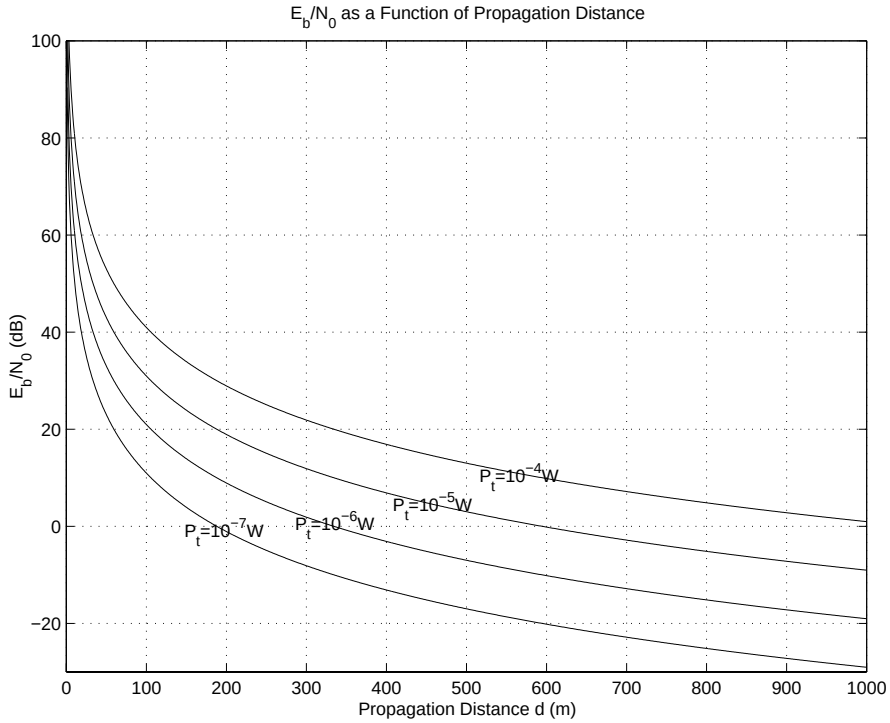

Figure 12: E_b/N_0 as a Function of Distance

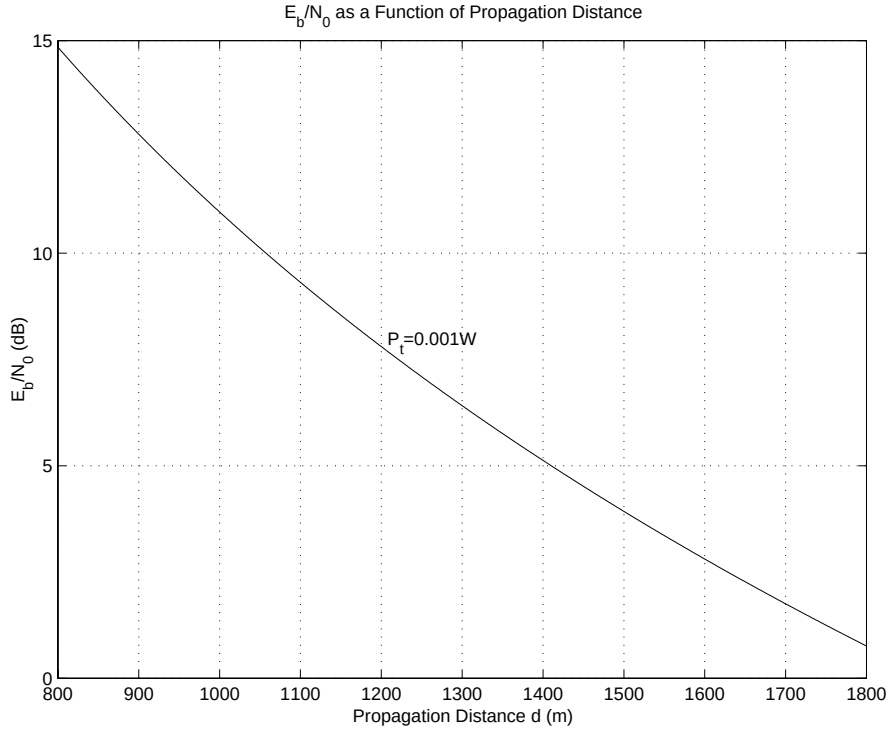

Figure 13: E_b/N_0 as a Function of Distance

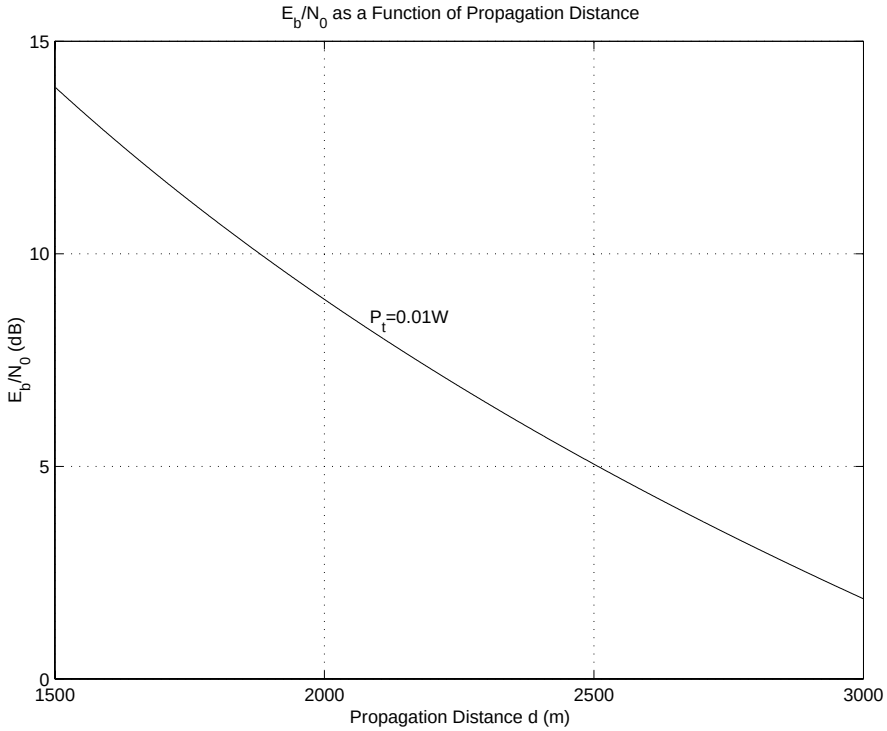

Figure 14: E_b/N_0 as a Function of Distance

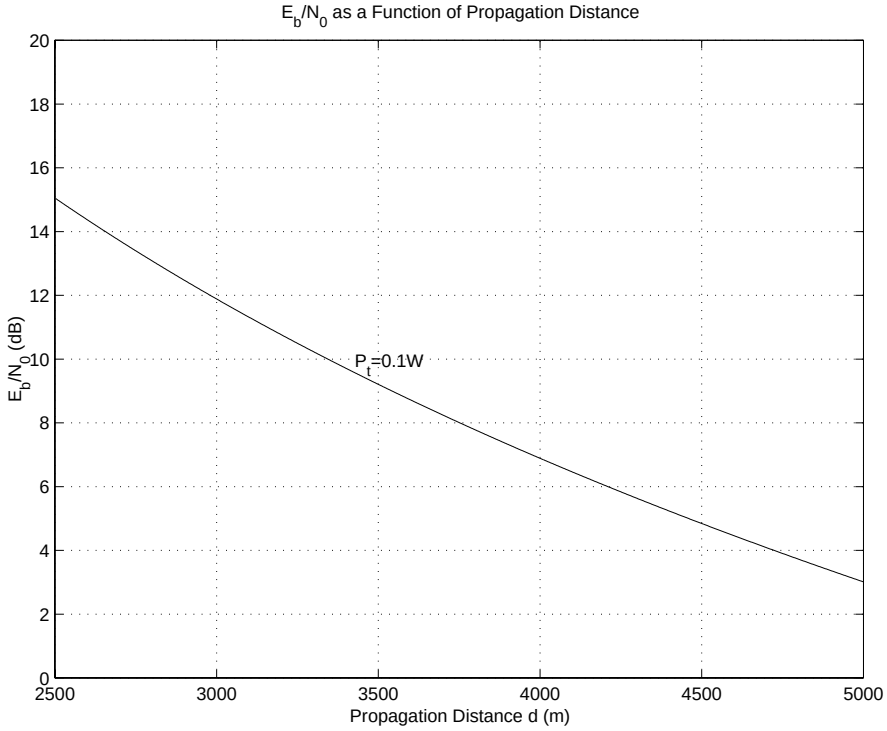

Figure 15: E_b/N_0 as a Function of Distance

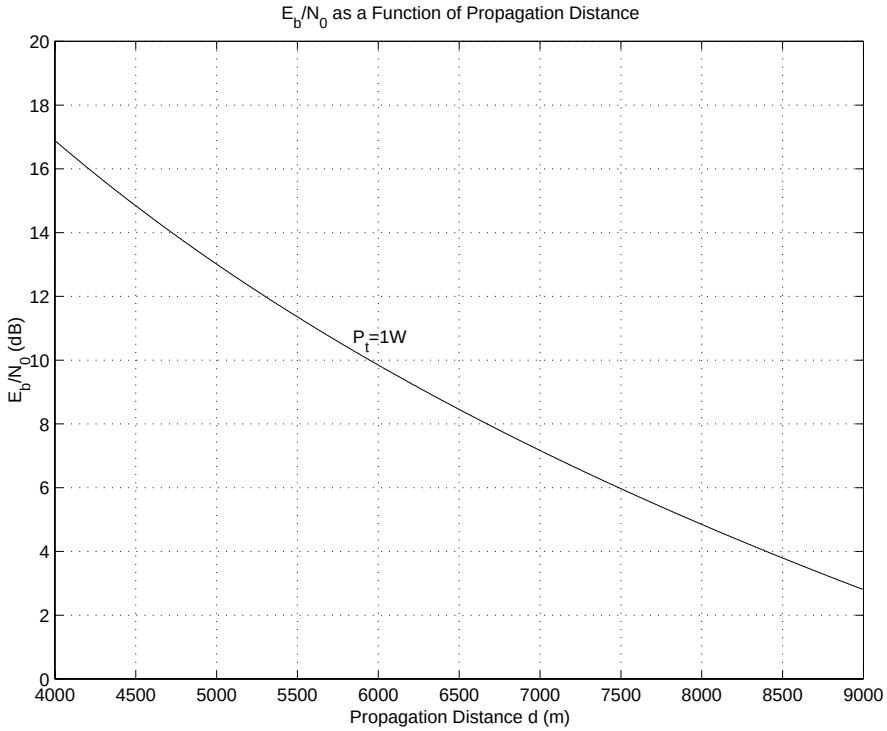

Figure 16: E_b/N_0 as a Function of Distance

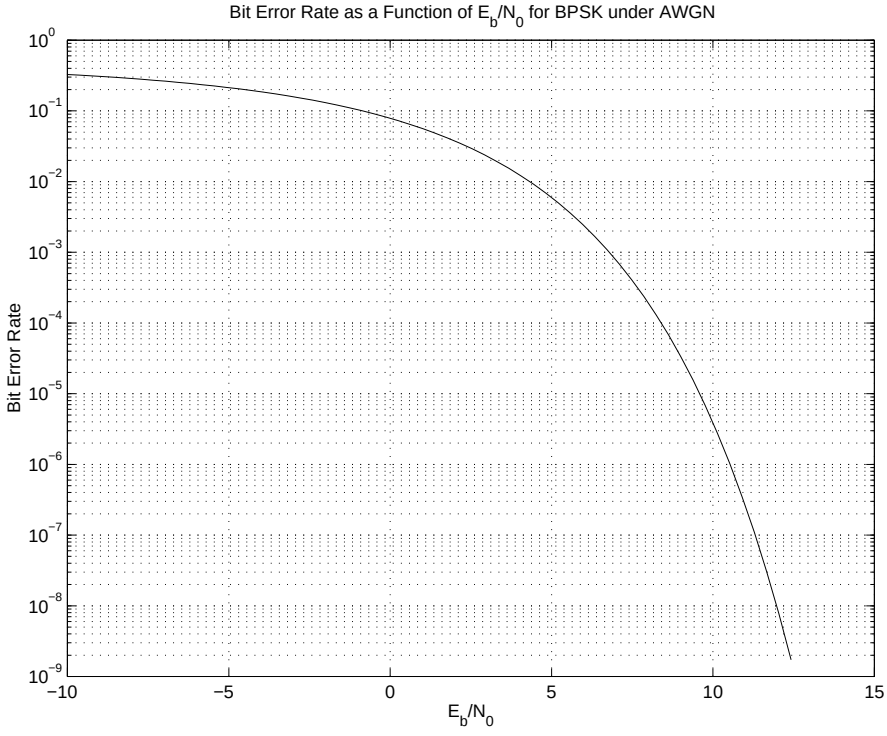

Figure 17: Binary Phase Shift Keying (bpsk) BER

2.1.5 Estimation Model

Each node updates its estimate on each other node's position every T_e, where $T_e = 2s$ for the current network model. Suppose it is time for node 1 to update its estimate on node 2's position, node 1 first gets node 2's last estimate from node 1's own database, suppose this estimate is $(\hat{x}_k, \hat{y}_k)$. This estimate may be recorded either at time t_k, where t_k is the time node 1 last receives a packet from node 2 between kT_e and $(k+1)T_e$, or at time kT_e, where kT_e is the time node 1 last gives an estimate on the position of node 2.

If node 1 ever receives a packet from node 2 between time kT_e and $(k+1)T_e$, the new estimate made by node 1 on node 2 at time $(k+1)T_e$ is determined by the following formula:

$$\begin{cases} \hat{x}_{k+1} = \hat{x}_k + d\hat{x}\,(v((k+1)T_t - t_k)) \\ \hat{y}_{k+1} = \hat{y}_k + d\hat{y}\,(v((k+1)T_t - t_k)) \end{cases} \tag{11}$$

where,

$$\begin{cases} d\hat{x} = \frac{gx - \hat{x}_k}{\hat{d}} \\ d\hat{y} = \frac{gy - \hat{y}_k}{\hat{d}} \end{cases} \tag{12}$$

and,

$$\hat{d} = \sqrt{(gx - \hat{x}_k)^2 + (gy - \hat{y}_k)^2} \tag{13}$$

Otherwise, if no packet from node 2 is received by node 1 between time kT_e and $(k+1)T_e$, the new estimate made by node 1 on node 2 at time $(k+1)T_e$ is determined by the following formula:

$$\begin{cases} \hat{x}_{k+1} = \hat{x}_k + d\hat{x} \cdot T_e \\ \hat{y}_{k+1} = \hat{y}_k + d\hat{y} \cdot T_e \end{cases} \tag{14}$$

The position estimation error at time kT_e made by node 1 on node 2 is defined as

$$e_k^{(1,2)} = \sqrt{(x_k - \hat{x}_k)^2 + (y_k - \hat{y}_k)^2} \tag{15}$$

Node 1 also gives estimates to the positions of all other nodes $3, \cdots, N$ in the same fashion at time kT_e, where N is the number of nodes in the network model. At the end of simulation, node 1 calculates the average of the position estimation error it made on all other nodes by

$$e^{(1)} = \frac{1}{M(N-1)} \sum_{i=2}^{N} \sum_{k=1}^{M} e_k^{(1,i)} \tag{16}$$

where M is the total number of estimates that node 1 made on each of other nodes.

Node $2, \cdots, N$ in the network make the estimates on the positions of all other $(N-1)$ nodes in the same fashion as does node 1. At the end of simulation, node $2, \cdots, N$ also calculate their average position estimation error $e^{(2)}, \cdots, e^{(N)}$ respectively.

The overall network performance measure is given by the average of the position estimation error made by all the nodes in the network:

$$e = \frac{1}{N} \sum_{k=1}^{N} e^{(k)} \tag{17}$$

2.1.6 Energy Constraint

As having been stated in the above sections, each mobile node has an imposed energy constraint, which is realized by means of battery capacity. If its consumed energy is more than its carried battery capacity, it can no longer transmit or receive any packet.

During simulation, caution had to be taken to set this energy constraint to a reasonable value in order to achieve meaningful results.

2.1.7 Optimization Problem Formulation

We first fixed several parameters to reduce the dimension of the problem as shown in Figure 1, i.e., in current network model, the carrier frequency is set to $30MHz$, data rate is $20kb/s$, each packet has size $224bits$, the communication system uses BPSK modulation without coding and equalization, and a simple propagation model is assumed as given in Eqn (6), etc.

Since it is very hard to find an analytical expression for average position estimation error e as a function of our chosen parameter set (T, q, P), we have to rely on simulation to get e as the value of the objective function for each parameter set (T, q, P) generated by the optimizer.

With the concern that each network simulation with different random seeds may give us different position estimation error e, we need to run simulation over the same parameter set (T, q, P) many times and get an average value of e.

Let $e(i)$ be the position estimation error of the network when $i-th$ random seed is used for simulation, then

$$\overline{e} = \frac{1}{30}\sum_{i=1}^{30} e(i) \tag{18}$$

is used as the value of the objective function for optimization.

The sample variance of $e(1), \cdots, e(30)$ can also be calculated by

$$Var(e) = \frac{1}{29}\sum_{i=1}^{30}(e(i) - \overline{e})^2 \tag{19}$$

As a summary, the problem of network layer simulation and optimization can be defined as follow:

- Energy Constraint: E

  Each node has a energy constraint E. It can no longer transmit or receive any packet if its consumed energy is over energy constraint E.

- Interval between Transmissions: T

  Each node broadcasts its position awareness packet with period T.

- Retransmission Strategy: q

  Each node rebroadcasts the same message after the first transmission based on q.

- Power Allocation: P

  Each node broadcasts or rebroadcasts the packet using the same power P that is irrelevant of the distance between nodes.

- Optimization Problem:

  Given E, optimize over (T, q, P) to get the lowest $\overline{e}$.

After defining our problem, we are ready to look at the simulator and the optimizer and integrate simulation and optimization together.

2.2 Modeling using *OPNET* Software

OPNET software is a very complicated, yet very powerful modeling tool. The basic *OPNET* modeling can be partitioned into three levels, namely, network level, node level, and process level. Please note that these three levels

have nothing to do with MURI layered structure. Network modelers can build models bottom up or top down, normally they have to go back and forth among different levels.

The network level modeling, node level modeling, and process level modeling for MURI are shown in Figure 18, Figure 19, and Figure 20 respectively.

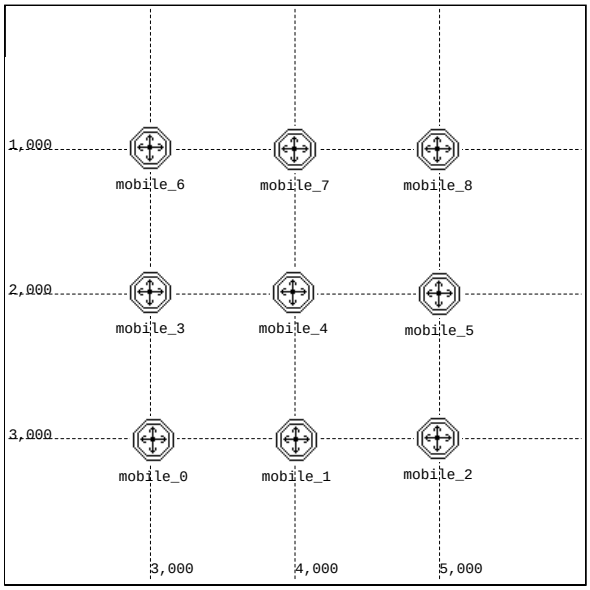

Figure 18: Network Level Modeling

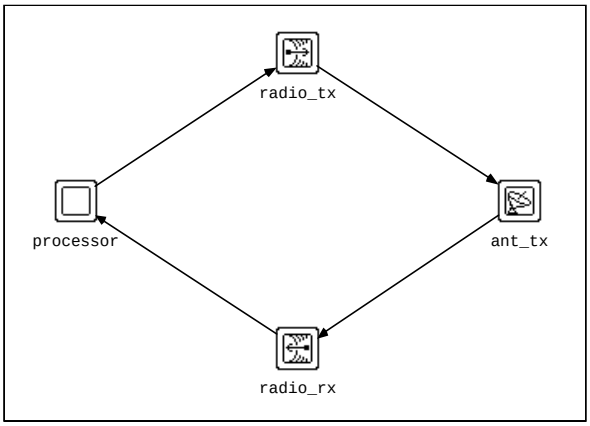

Figure 19: Node Level Modeling

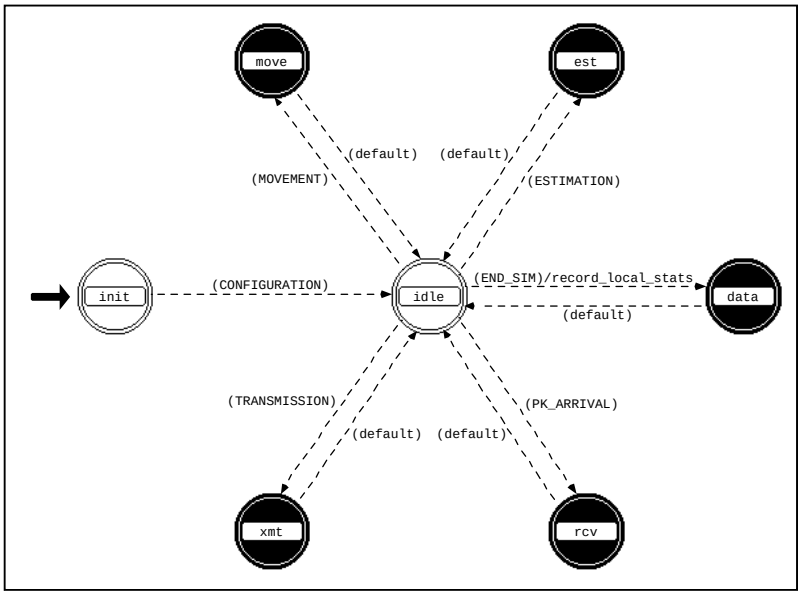

Figure 20: Process Level Modeling

2.3 Simulated Annealing

The method of *simulated annealing* is a technique that has attracted significant attention as suitable for optimization problems of large scale, especially ones where a desired global extremum is hidden among many, poorer, local extrema. This is one of the reasons *simulated annealing* was used as the optimization tool in MURI.

The *simulated annealing* program used in MURI is called *hidenseek* and was developed by students of Professor Robert Smith's at the IOE department of University of Michigan. The current version of *simulated annealing* can only deal with parameters that are continuous instead of discrete.

hidenseek requires a well-defined objective function on a parameter set, which forms the interface between optimization and simulation.

2.4 Integration of Simulation and Optimization

Since optimization program requires a well defined objective function, we want to provide the optimization program with an illusion that there existed an objective function by using our simulation values. The general setup of simulation and optimization is given in Figure 21.

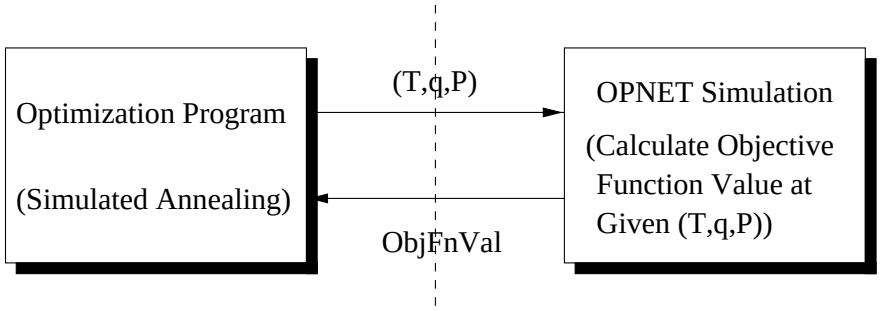

Figure 21: MURI Network Simulation and Optimization Setup

hidenseek relies on hundreds of iterations, or probably more, to converge to the global minimum. For each iteration, it feeds *OPNET* with a parameter set (T, q, P) and waits for *OPNET* to return the position estimation error. Since we want to average the position estimation error over 30 *OPNET* simulations on the same parameter set (T, q, P) but with different random seeds, and each *OPNET* simulation is quite time-consuming, it is natural and necessary to parallelize *OPNET* simulation on different machines, with each machine running an *OPNET* simulation on the same parameter set, but with a different random seed, as shown in Figure 22. The position estimation error returned by each machine is then averaged and the averaged value is returned to *hidenseek*, upon which a new parameter set is determined by *hidenseek* for another round of simulation by *OPNET*. This procedure keeps going until the number of iterations in *hidenseek* is reached and the best parameter set is given by *hidenseek*. By parallelizing *OPNET* simulation on different machines, the time span to get the final optimization result is greatly reduced.

As you may have noticed in Figure 22, *OPNET* simulations over the same parameter set are actually running on 40 different machines, however, *problem.c* only takes the first 30 returned position estimation error and returns the average of these 30 values to the optimizer without waiting the rest 10 simulations to finish. By this way, the overall speed of simulation and optimization would be not reduced even if some of simulation processes were killed by other users.

One example below may show how computationally intensive the integration of simulation and optimization can be. If *hidenseek* iterates 300 times and we are using parallel simulation and optimization structure as shown in Figure 22, one realization of integration of simulation and optimization took 28 hours for a network model with 9 nodes.

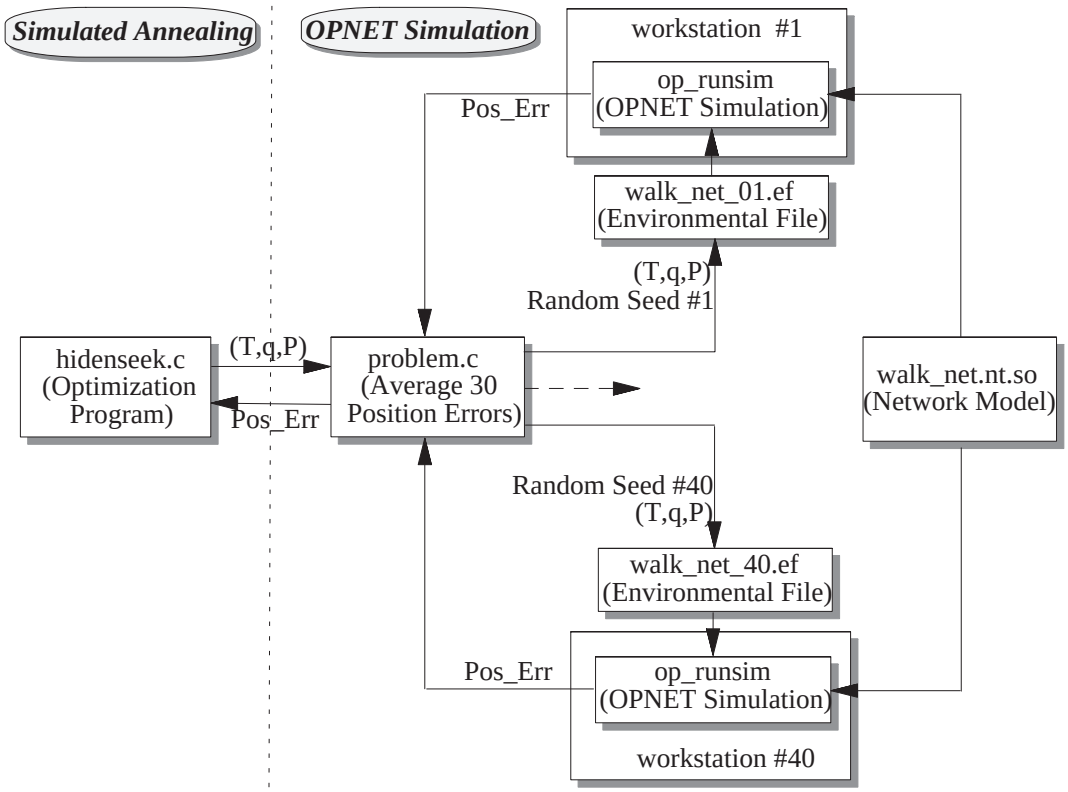

Figure 22: MURI Network Simulation and Optimization Using Parallel Structure

3 Preliminary Simulation Results

A lot of simulation has been done. Here we just show some relatively important results. Note that all the following results are obtained using our simple network scenario as shown in Figure 1 without coupling the result from other layers.

We have some preliminary results on the network performance (average position estimation error) versus energy constraint imposed on each node at network layer as shown in Figure 23 and Figure 24, where Figure 24 is a zoom-in plot of Figure 23 on the interval $(0, 0.04) Joules$. In these two figures, we considered the loss of the dipole antenna and each point on the curve requires 120 iterations in *hidenseek*. Figure 25 shows the relationship between the standard deviation of position estimation error versus the average of position estimation error, where the standard deviation is obtained by taking the squared root of Eqn (19). Figure 26 is a 3D plot of average position estimation error versus q and P with $T = 8s$, $E = 2 \times 10^{-2} Joules$, and without considering antenna loss.

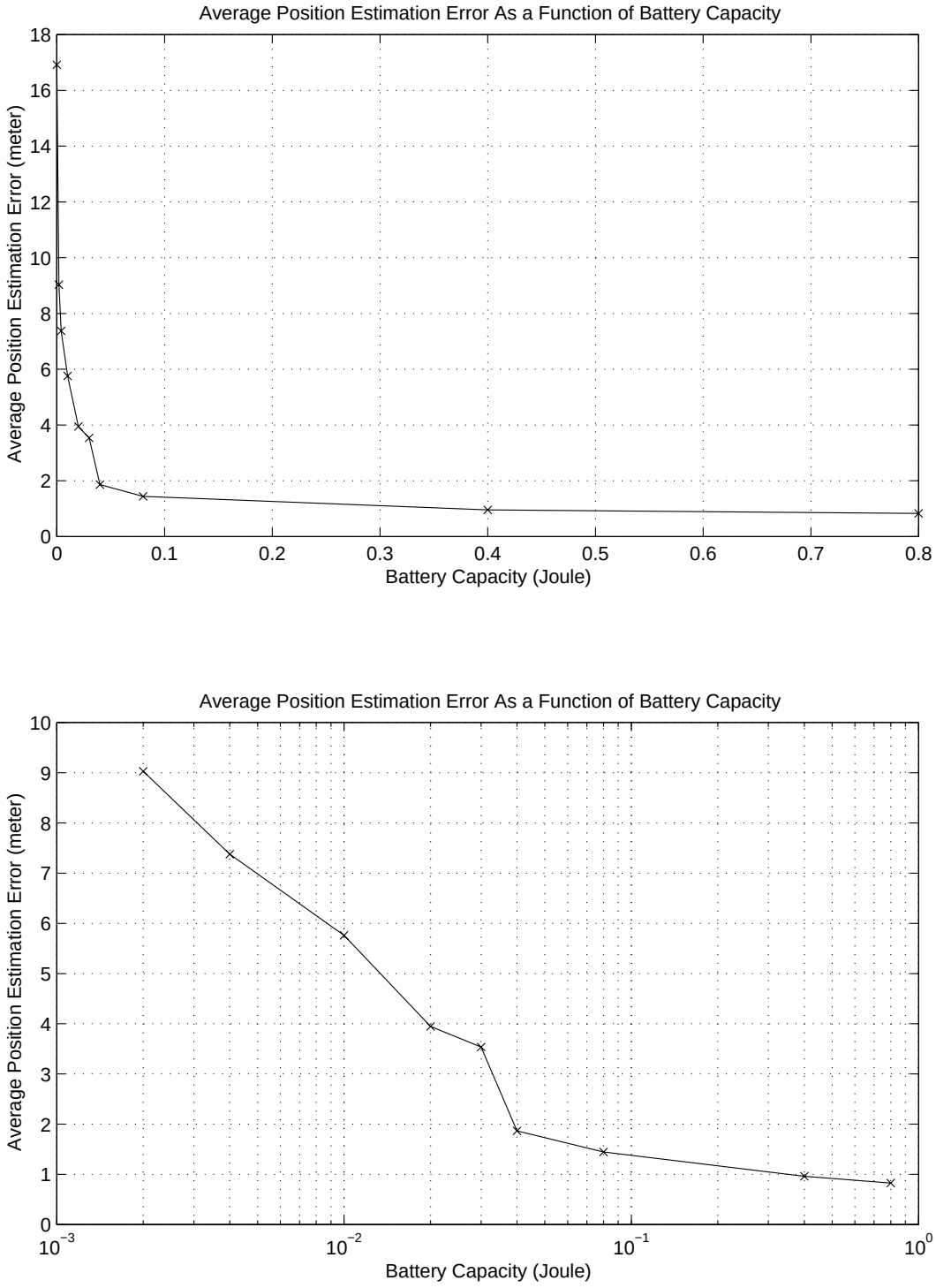

Figure 23: drop zone $1km \times 1km$, estimated position error versus energy constraint

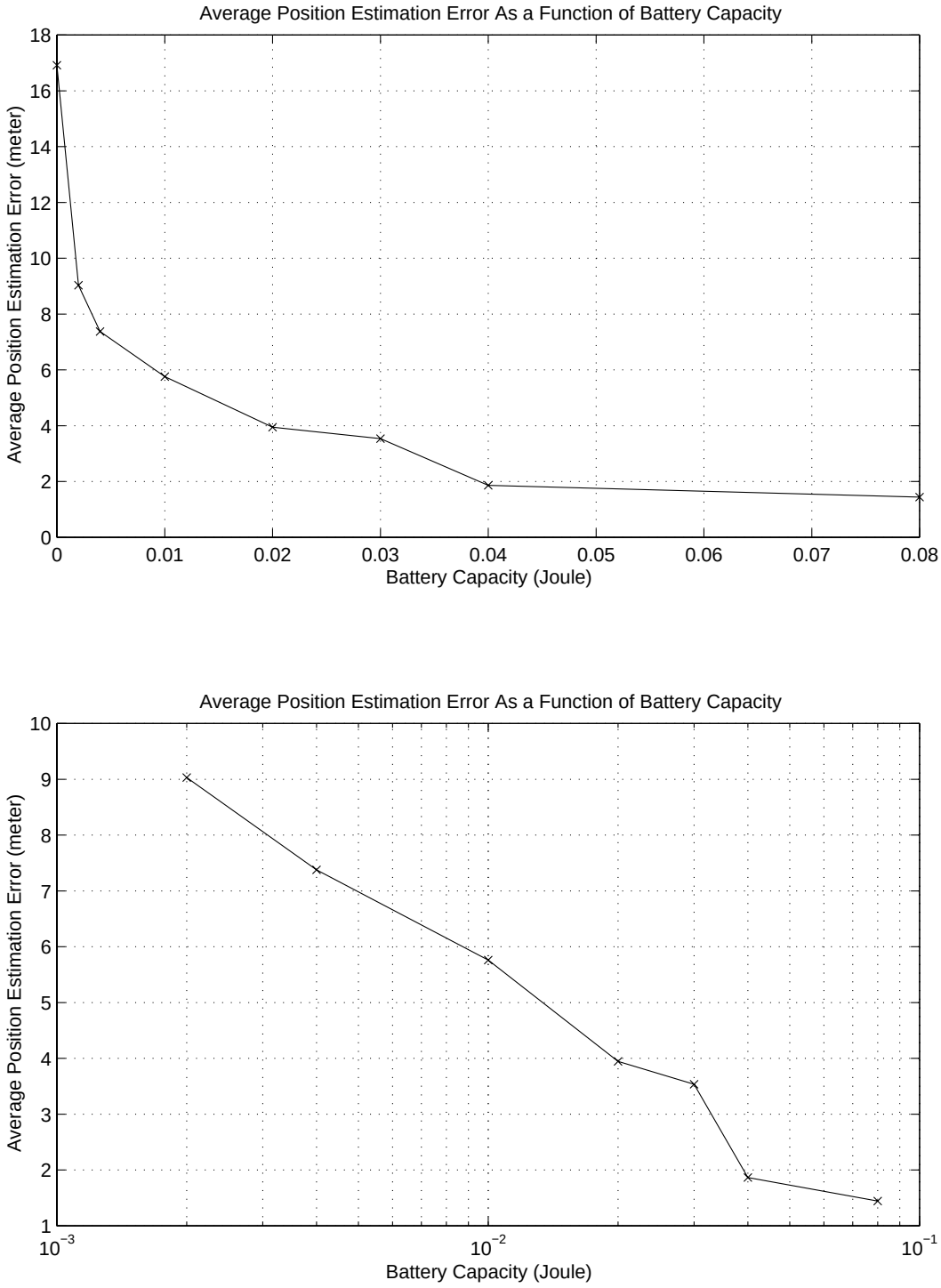

Figure 24: drop zone $1km \times 1km$, estimated position error versus energy constraint

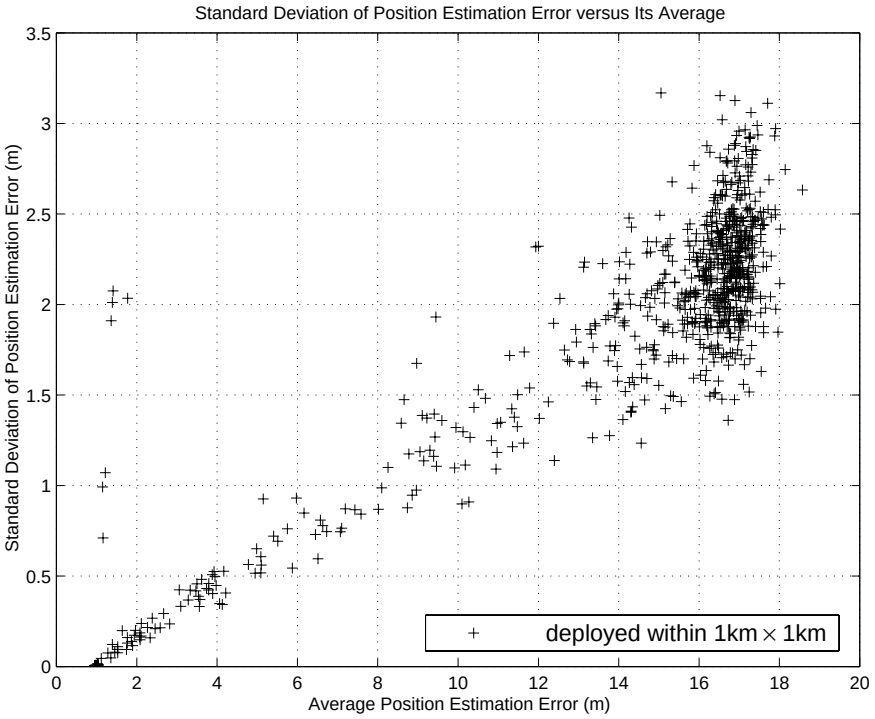

Figure 25: With drop zone $1km \times 1km$ and energy constraint $2 \times 10^{-2}J$

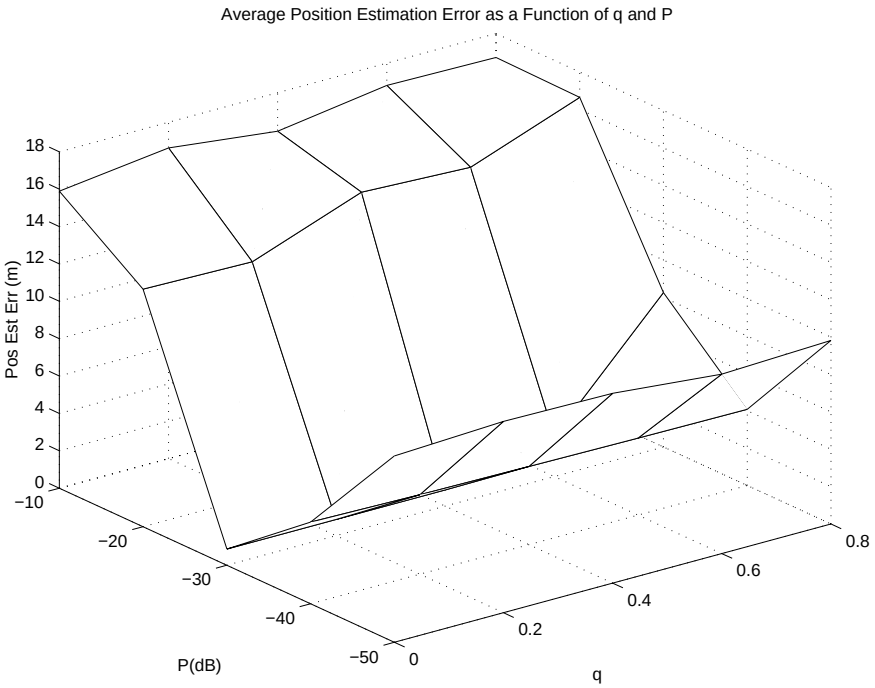

Figure 26: With $T = 8s$, drop zone $1km \times 1km$, and energy constraint $2 \times 10^{-2}J$

4 Analysis of Simplified Network Scenario

We want to do some analysis over an even more simplified network model than what we have implemented currently. Assume that the interval between successive transmissions T_t is a multiple of the interval between successive estimates T_e, that is, $T_t = KT_e$, where K is a positive integer.

We also assume that there is a perfect alignment of movement, transmission and estimation, that is, estimation always happens at time $0, T_e, 2T_e, \cdots$, and transmission always happens at time $0, T_t, 2T_t, \cdots$. When a transmission occurs, if a receiver ever receives the packet, it receives the packet at the same time the packet is transmitted, so we assume there is no transmission and propagation delay between the transmitter and the receiver.

Let (X_k, Y_k) be the position of a node at time kT_e. (Note that this is the time when estimation happens.) The simplified drift model of (X_k, Y_k) is given by:

$$\begin{cases} X_k = A_k + \alpha_k \\ Y_k = B_k + \beta_k \end{cases} \tag{20}$$

where A_k and B_k are known values for each k, α_k and β_k are disturbance random variables uniformly distributed on $(-0.5, 0.5)$. We assume α_i and β_j are independent and identically distributed for all i and j.

Therefore,

$$\begin{cases} E[\alpha_k] = E[\beta_k] = 0 \\ Var[\alpha_k] = Var[\beta_k] = \frac{1}{12} \end{cases} \tag{21}$$

We want to find a formula of the average mean squared position estimation error with respect to the key parameter set, which is (T, q, P) at the current network model, or in some way related to the key parameter set. This is a little different from the objective function of the simulation, which tries to minimize the average root mean squared position estimation error.

Let node 1 be observer and node 2 be observant. The position of node 1 at time kT_e is given by

$$\begin{cases} X_{1,k} = A_{1,k} + \alpha_{1,k} \\ Y_{1,k} = B_{1,k} + \beta_{1,k} \end{cases} \tag{22}$$

and the position of node 2 at time kT_e is given by

$$\begin{cases} X_{2,k} = A_{2,k} + \alpha_{2,k} \\ Y_{2,k} = B_{2,k} + \beta_{2,k} \end{cases} \tag{23}$$

where by our previous assumption,

$$\begin{cases} E[\alpha_{1,k}] = E[\alpha_{2,k}] = E[\beta_{1,k}] = E[\beta_{2,k}] = 0 \\ Var[\alpha_{1,k}] = Var[\alpha_{2,k}] = Var[\beta_{1,k}] = Var[\beta_{2,k}] = \frac{1}{12} \end{cases} \tag{24}$$

The estimate of the position made by node 1 on node 2 at time kT_e is given by the following formula:

$$\begin{cases} \hat{X}_{2,k} = X_{2,k}, \hat{Y}_{2,k} = Y_{2,k} & \text{if node 1 receives a packet from node 2 at time } kT_e \\ \hat{X}_{2,k} = A_{2,k}, \hat{Y}_{2,k} = B_{2,k} & \text{if node 1 does not receive a packet from node 2 at time } kT_e \end{cases} \tag{25}$$

The mean squared position estimation error made by node 1 on node 2 at time kT_e is thus given by:

$$\begin{aligned} & E[(\hat{X}_{2,k} - X_{2,k})^2 + (\hat{Y}_{2,k} - Y_{2,k})^2] \\ = \; & E[(\hat{X}_{2,k} - X_{2,k})^2 + (\hat{Y}_{2,k} - Y_{2,k})^2 | \text{packet received at } kT_e] \cdot Prob(\text{packet received at } kT_e) \\ + \; & E[(\hat{X}_{2,k} - X_{2,k})^2 + (\hat{Y}_{2,k} - Y_{2,k})^2 | \text{packet not received at } kT_e] \cdot Prob(\text{packet not received at } kT_e) \\ = \; & 0 \cdot Prob(\text{packet received at } kT_e) + E[\alpha_{2,k}^2 + \beta_{2,k}^2] \cdot Prob(\text{packet not received at } kT_e) \\ = \; & \tfrac{1}{6} \cdot Prob(\text{packet not received at } kT_e) \end{aligned} \tag{26}$$

Since node 2 only transmits packets at time $0, KT_e, 2KT_e, \cdots$, node 1 can not receive any packets at any other time from node 2. Therefore, at time kT_e, where k is not a multiple of K, we have $Prob(\text{packet not received at } kT_e) = 1$, and the corresponding mean squared position estimation error is $\frac{1}{6}$. On the other hand, at time kT_e, where k is a multiple of K, we have mean squared position estimation error $\frac{1}{6} Prob(\text{packet not received at } kT_e)$.

Let M be the total number of estimates that node 1 makes on node 2 when the simulation terminates. Then, the average mean squared position estimation error made by node 1 on node 2 is given by

$$e^{(1,2)} = \left(\sum_{i=0}^{M/K} \frac{1}{6} \cdot Prob(\text{packet not received at } iKT_e) + (M - M/K)\frac{1}{6} \right) / M \tag{27}$$

For any given trajectory, i.e., if we know A_k and B_k for all k, we can calculate $Prob(\text{packet not received at } kT_e)$ for all possible k by first calculating d_k, the distance between node 1 and node 2, where $k = iK$,

$$d_k = \sqrt{((A_{1,k} - A_{2,k}) + (\alpha_{1,k} - \alpha_{2,k}))^2 + ((B_{1,k} - B_{2,k}) + (\beta_{1,k} - \beta_{2,k}))^2} \tag{28}$$

In our network layer simulation,

$$\begin{aligned} |A_{1,k} - A_{2,k}| &>> |\alpha_{1,k} - \alpha_{2,k}| \\ |B_{1,k} - B_{2,k}| &>> |\beta_{1,k} - \beta_{2,k}| \end{aligned} \tag{29}$$

Therefore, we can approximate d_k by

$$d_k = \sqrt{(A_{1,k} - A_{2,k})^2 + (B_{1,k} - B_{2,k})^2} \tag{30}$$

Using Eqn (10) and the above d_k and transmitted power P, we can calculate the packet error probability PER_k for one transmission of the packet at time kT_e. Suppose that whether a packet is in error or not is independent of all other packets, and is independent of other retransmissions of the same packet. Then

$$\begin{aligned} & Prob(\text{packet not received at } kT_e) \\ = & \sum_{n=1}^{\infty} Prob(\text{packet not received } kT_e | \text{n transmission of the packet}) \cdot Prob(\text{n transmissions of the packet}) \\ = & \sum_{n=1}^{\infty} Prob(\text{packet not received at } kT_e | \text{n transmission of the packet}) \cdot Prob(\text{n-1 retransmissions of the packet}) \\ = & \sum_{n=1}^{\infty} PER_k^n \cdot q^{n-1}(1-q) \\ = & \frac{PER_k(1-q)}{1-PER_k \cdot q} \end{aligned} \tag{31}$$

Therefore, we have an equation that relates $Prob(\text{packet not received at } kT_e)$ with retransmission probability q. If we take the derivative of Eqn(31), we have

$$\frac{d}{dq} Prob(\text{packet not received at } kT_e) = \frac{PER_k^2 - PER_k}{(1 - PER_k \cdot q)^2} \leq 0 \tag{32}$$

because $0 \leq PER_k \leq 1$.

Therefore, Prob(packet not received at kT_e) is a monotone decrease function of q as expected.

Eqn(26) then becomes

$$\begin{aligned} & E[(\hat{X}_{2,k} - X_{2,k})^2 + (\hat{Y}_{2,k} - Y_{2,k})^2] \\ = & \frac{1}{6}\frac{PER_k(1-q)}{1-PER_k \cdot q} \end{aligned} \tag{33}$$

Substitute Eqn (33) into Eqn (27), we get the average mean squared position estimation error made by node 1 on node 2. We can apply the above procedure to any pair of nodes in the network, thus find the average mean squared position estimation error of the overall network.

Compared with our actual implemented *OPNET* network model, our assumptions made above may be not well justified, for example, there may not be perfect alignment of interval for transmission and interval for estimation and there are propagation and transmission delays in our actual network model. Nevertheless, our analysis above gives us some understanding of the average position estimation error of the network.

5 Conclusions

There are some conclusions that can be made from our simulation:

1. Since PER is so sensitive to distance, a good strategy of using power would be varying transmitting power based on the the propagation distance between nodes. As the distance gets smaller, such as the situation when almost all nodes are near the same destination, less power should be used for communication if we want to achieve the same PER as when the nodes are separated far apart. But then, we are optimizing network performance over a function $P(d)$, which definitely adds more complexity to our optimization problem.

2. Power control strategy should be incorporated with multi-hop routing protocol in the future in order to save more energy.

3. *Simulated annealing* does not guarantee the optimum solution when the number of iterations is small. However, we can tune the decision variables returned by *simulated annealing* based on theoretical PER versus propagation distance curve from Figure 2 to Figure 11 to get better performance.

4. Since at the moment the network model does not have a terrain constraint, retransmission almost does not affect the network performance at high signal-to-noise ratio. However, it affects the network performance at low signal-to-noise ratio. Therefore, there may be an optimal retransmission strategy that can be designed based on the signal-to-noise ratio.

6 Proposed Power Adjustification

Based on the PER versus propagation distance formula in AWGN, we can set our target of a particular PER that we want to achieve, then calculate the P_t based on the current estimated distance and inverse formula of Eqn (6).

Suppose BER is small, then Eqn (10) becomes

$$PER \simeq 1 - (1 - BER)^L = 1 - (1 - L \cdot BER) = L \cdot BER \tag{34}$$

which implies that

$$BER \simeq \frac{PER}{L} \tag{35}$$

Since BER is small, we can also approximate Eqn (9) by

$$BER = Q\left(\sqrt{\frac{2E_b}{N_0}}\right) \simeq e^{-\frac{E_b}{N_0}} \tag{36}$$

Then, By Eqn (34) and Eqn (36), we have

$$E_b \simeq N_0 ln\left(\frac{L}{PER}\right) \tag{37}$$

Based on Eqn (37) and Eqn (7), we have

$$P_r = \frac{E_b}{T_b} = E_b R_b \simeq N_0 R_b ln\left(\frac{L}{PER}\right) \tag{38}$$

Based on current propagation model of Eqn (6), we have

$$P_t = \frac{P_r}{4\left(\frac{\lambda_c}{4\pi d}\right)^2 G_t G_r sin^2\left(\frac{2\pi h_t h_r}{\lambda_c d}\right)} \tag{39}$$

Assuming that $d >> max(h_t, h_r)$, then

$$sin\left(\frac{2\pi h_t h_r}{\lambda_c d}\right) \simeq \frac{2\pi h_t h_r}{\lambda_c d} \tag{40}$$

Substitute Eqn (38) and Eqn (40) into Eqn (39), we have

$$P_t \simeq \frac{N_0 R_b ln\left(\frac{L}{PER}\right)}{4\left(\frac{\lambda_c}{4\pi d}\right)^2 G_t G_r \left(\frac{2\pi h_t h_r}{\lambda_c d}\right)^2} = \frac{N_0 R_b d^4 ln\left(\frac{L}{PER}\right)}{G_t G_r h_t^2 h_r^2} \tag{41}$$

Suppose we are targeting at $PER = 10^{-3}$ and assume that we use the value of parameters given in Table 1, then power allocation strategy for each packet transmission is

$$P_t \simeq 9.8533 \cdot 10^{-16} \cdot d^4 \quad W \tag{42}$$

We want the power allocation strategy to be simple because we want to save the processing power at each mobile nodes. It is worth noting that any multiplicative factor of power loss or gain can be thrown into the coefficient before d^4 in Eqn (42).

At the moment, the power allocation is not used in network scenario, because we don't want to add another degree of freedom to complicate our problem.

7 Future Work

The MURI network layer simulation and optimization is going on well at the moment. We are expecting to couple the results from different layers of the communication system and to obtain more interesting results.

We are going to build more features into current network model, such as power allocation strategy, retransmission strategy, and probably routing as well, and investigate the sensitivity of performance measure with respect to different parameters in order to improve the network performance.

Acknowledgment

I would like to thank Prof. Wayne Stark, Prof. Stéphane Lafortune, Prof. Demosthenis Teneketzis, Prof. Kimberly Wasserman, Karl Kluge and all the students George Barrett, Rami DeBouk, Christopher Lott and Tara Javidi for their helpful opinions during our weekly meeting, Olivier Boivineau for the contribution to MATLAB program of network simulation, J.D. Shumpert for the contribution to the antenna model.

References

[1] Mil 3, "OPNET Modeler Manual," 1997.

[2] Pablo Aghemo, George Barrett, Olivier Boivineau, Rami Debouk, Utkan Demirci, Christopher Lott, "Very High Level Scenario for the MURI Simulation," 1998

[3] Karl Kluge, "Wireless Network Simulation and Design Optimization Plan for the MURI 'Low Energy Electronics Design for Mobile Platforms', " January 7, 1999.

[4] H. Edwin Romeijn and Robert L. Smith, "Simulated Annealing for Constrained Global Optimization," Journal of Global Optimization 5:101-126, 1994.