Shadows and Tall Trees

Path Loss due to Terrain Diffraction[1]

Kenneth Steeves
ECE 3065 – A
04/27/2006

INTRODUCTION

New generations of wireless networks are coming to realization around the country. Users of these new networks will expect better service, which means fewer dropped calls, and more areas of coverage. New, after all, corresponds to better in the minds of everyday users. It is therefore important that the new network is modeled appropriately for received signal strength in the areas surrounding the base stations. With an accurate propagation, or path loss model, service providers will be able to determine where the optimal locations for base stations are, thus being able to more accurately design their network for better signal coverage.

In addition to an increase in customer satisfaction, there are other applications of better propagation models. The more accurate signal strength predictions are, the easier it will be to locate the position of a user placing a 911 call. When the signal strength measurement is taken, it can be matched up to a location that corresponds to a received signal strength predicted by the path loss model. This would allow emergency vehicles to find the caller's location quicker.

PROPAGATION MODEL: TOP LEVEL

The propagation model uses the logarithmic form of the link budget equation, shown in Equation 1. In this equation, P_R is the received signal strength in dB. This is the number to be calculated

$$P_R = EIRP - Loss \qquad \text{Equation 1}$$

by the model. When this value falls below a certain value, known as the noise floor, a call is dropped. When it is sufficiently above this noise floor, the call goes through. EIRP is an abbreviation for the Effective Isotropic Radiated Power. This is just the sum of the power radiated from the transmitter, the gain of the transmitter, and the gain of the receiver. Finally, there is the Loss term. This term is determined from the various modes of propagation, which include the following:

- Line-of-Sight in Free Space
- Reflection/Refraction
- Scattering
- Diffraction

Line-of-sight is just a free space line between the transmitter and receiver, with no obstructions. Reflection and refraction is just the way the signals bounce off, or penetrate surfaces. Scattering occurs with tiny irregularities in surfaces that cause the wave incidence angle to not equal the reflection angle. Finally diffraction occurs when waves bend around objects, such as mountains.

Each of these propagation modes have some loss associated with them. However, some of these loss terms dominate over others. There will always be line-of-sight loss due to the dominant nature of that particular loss term. Due to the large variations in terrain elevations, diffraction is Lodue to vegetation close to the ground. This dominance arises because variations in terrain due to large changes in elevation are considerably larger than variations in terrain due to vegetation.

PROPAGATION MODEL: COMPONENT LEVEL

Reference Path Loss

When solving the Poynting Vector for spherical waves, it is found that there is a dependence on the square of the wavelength of the transmitted wave. The loss due to this dependence is known as the reference path loss, and is defined by Equation 2.

$$PL_{REF} = 20\log_{10}\left(\frac{4\pi c}{f}\right) \qquad \text{Equation 2}$$

In this equation, c is the speed of light in a vacuum, which is taken to be $3x10^8$ m/s, and f is the frequency at which the power is being transmitted.

Free-Space Path Loss

Free-space refers to the unobstructed line-of-sight between the transmitter and the receiver. When solving the Poynting Vector for spherical waves, it is found that there is an inverse dependence on the square of the transmitter-receiver separation distance when dealing with an isotropic radiator. The loss due to this dependence is known as the free-space loss, and is defined by Equation 3:

$$PL_{FS} = 20\log_{10}(r) \qquad \text{Equation 3}$$

In this equation, r is the separation distance between the transmitter and the receiver.

Diffraction Path Loss

Diffraction occurs when waves bend around objects or obstructions to a line-of-sight. In the case of terrain variations, loss due to diffraction becomes a dominant term. In November, 2002, a study was performed to determine field strength due to irregular terrain, and gave way to the Point-to-Point model.

The data was taken at 93.1 MHz in Springfield, Massachusetts. The Point-to-Point model incorporates four determinants of propagation over irregular terrain. These four determinants include the clearance or blockage of a line-of-sight between the transmitter and receiver, the position of a terrain obstacle in the line-of-sight, the roundness associated with the terrain obstacle, and atmospheric refraction. The model utilized in this network service includes two most dominant determinants; clearance and blockage, and obstacle position.[2]

Path Clearance/Blockage

Path clearance or blockage is determined by the distance between the elevation of the direct path between the transmitter and the receiver, and the elevation of the obstacle in the path. Denoted by F in Figure 1, if there is clearance, then F is a positive number. If there is blockage, then F is a negative number. Clearance occurs when the line-of-sight is above the obstacle, and blockage occurs when the line-of-sight passes through the obstacle.

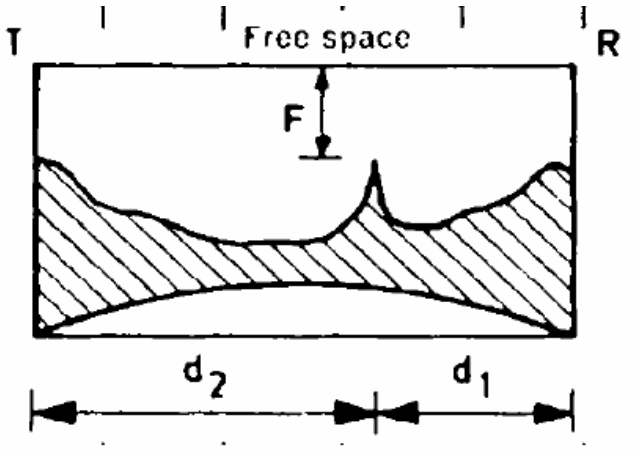

Figure 1. Cross-sectional view of an elevation showing clearance [3].

Obstacle Position

When analyzing the line-of-sight between the transmitter and receiver, the obstacle position along that line-of-sight needs to be determined. From there, the first Fresnel radius is calculated from Equation 4:

$$F_1 = \sqrt{\frac{c \times d_1 \times d_2}{f \times (d_1 + d_2)}} \qquad \text{Equation 4}$$

In this equation, c is again the speed of light, d_1 is the distance between the transmitter and the position of the obstacle, d_2 is the distance between the position of the obstacle and the receiver, as shown in Figure 2, and f is the frequency of the power being transmitted.

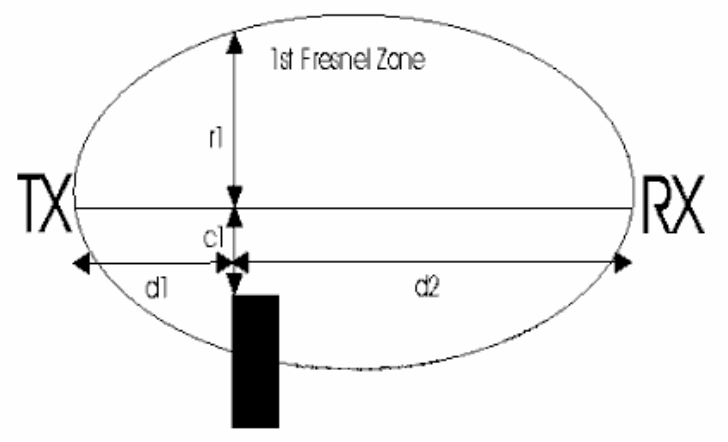

Figure 2. Representation of the first Fresnel Zone around an object [4].

Diffraction Loss Calculation

The information above is combined into a relation with loss from diffraction, known as Knife-Edge loss. This is described by Equation 5:

$$PL_{DIFF} = 1.377\left(\frac{F}{F_1}\right)^2 - 11.31\left(\frac{F}{F_1}\right) + 6 \qquad \text{if } \frac{F}{F_1} \geq -0.5$$

$$PL_{DIFF} = -\frac{50.4}{1.6 - \frac{F}{F_1}} + 36 \qquad \text{if } \frac{F}{F_1} \leq -0.5$$

Equation 5

The coefficients were derived empirically from the data collected during the experiment.

CONCLUSION

When putting this model together, Equation 6 results from expanding the link budget in Equation 1. By adding the diffraction component to the Loss term, a more accurate prediction of the

$$P_R = EIRP - 20\log\left(\frac{4\pi c}{f}\right) - 20\log(r) - PL_{DIFF}$$

Equation 6

signal coverage can be made. Better signal coverage corresponds to happier customers, which corresponds to more business for Wimaxorbust, Inc. One of the most important applications of having such an accurate propagation model is in determining optimal base station locations. If such locations are known, less money would need to be spent to enhance dead zones, because it is already known that signal coverage is at a maximum for the given terrain in the region.

SUMMARY OF DATA COLLECTED

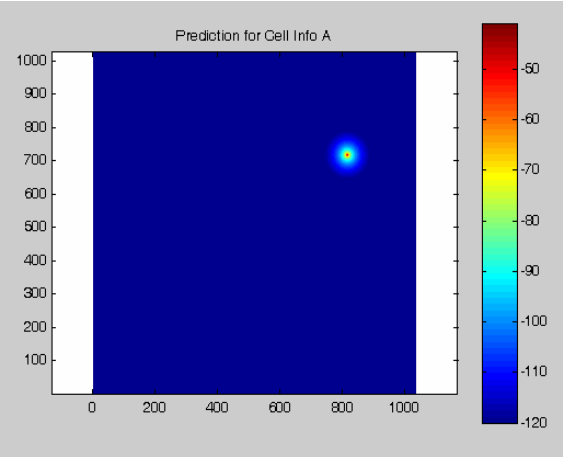

Figure 3. Cell A Prediction.

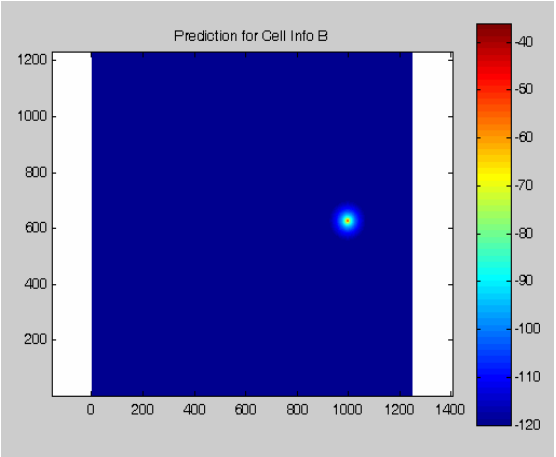

Figure 4. Cell B Prediction.

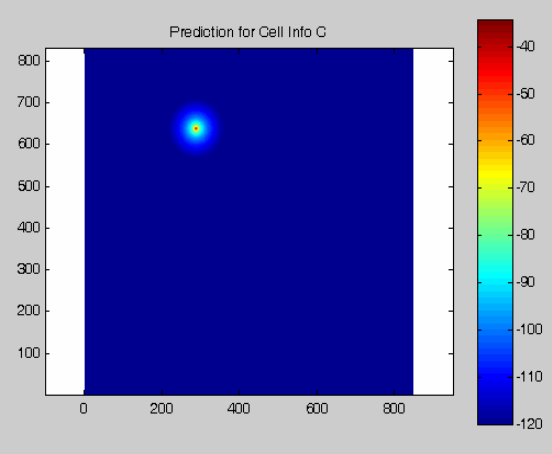

Figure 5. Cell C Prediction.

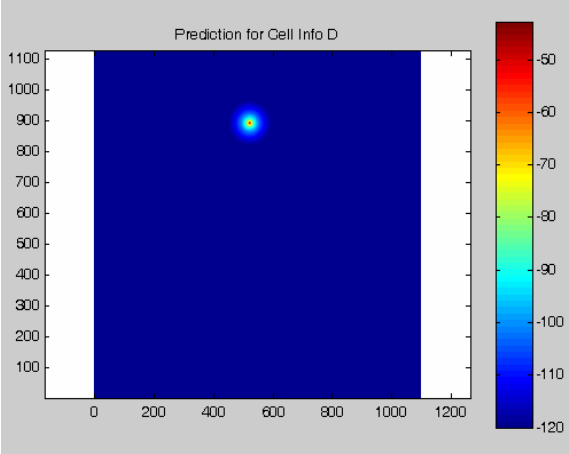

Figure 6. Cell D Prediction.

Figures 3-6 show the prediction maps generated by the propagation model. When compared with actual measurements, the mean, and standard deviation was found for each cell. That data is summarized in Table 1.

Cell	Mean	Standard Deviation
Cell A	-17.2 dB	9.81 dB
Cell B	-11.15 dB	10.63 dB
Cell C	-8.78 dB	9.05 dB
Cell D	-15.32 dB	10.07 dB

Table 1. Summary of Mean and Standard Deviation for each cell.

REFERENCES/NOTES

[1] The title “Shadows and Tall Trees” refers to the shadow boundaries caused by diffraction at edges, and the “knife edge” diffraction associated with tall trees. The title is taken from the name of 5th chapter in William Golding’s novel “The Lord of the Flies,” © 1953. It is also the title of the final track on U2’s debut album “Boy,” © 1980.

[2],[3] *Field Strength Prediction in Irregular Terrain – the PTP Model*, 2002, Wong, Harry.

[4] *Wireless – Fresnel Zones and Their Effect*, http://www.zytrax.com/tech/wireless/fresnel.htm

APPENDIX: MATLAB Code

Function: Signal_Strength_Prediction

```
function OutputMap = Signal_Strength_Prediction(TerrainMap, Azimuth, EIRP, BS_x, BS_y, Frequency, CellSize,
        Height)
%   Function Definition:
%      Signal_Strength_Prediction      Produces the raster map containing
%                                      the predicted values of signal
%                                      strength at each point on the
%                                      raster map.
%   Input Definitions:
%      TerrainMap           Map of the terrain that will be
%                           used to predict the signal
%                           strength.
%      Azimuth              Direction in which the antenna is
%                           pointing. This is an angle from 0
%                           to 360 degrees.
%      EIRP                 Effective Isotropic Radiated Power.
%                           Transmitted power plus the antenna
%                           gains.
%      BS_x                 Column coordinate in the raster map
%                           of the transmitter location.
%      BS_y                 Row coordinate in the raster map of
%                           the transmitter location.
%      Frequency            Radiation Frequency in MHz of the
%                           transmitter.
%      CellSize             Size of each map pixel in meters.
%      Height               Hight of the Base Station's antenna
%                           in meters above the ground.
%   Output Definition:
%      OutputMap                 Map of predicted signal strength.
%

% Contants.
c = 3*10^8;                              % Speed of Light in a vaccuum.
Noise_Floor = -120;                          % Sets the Noise Floor to -120 dB.

% Map Processor.
[Map_Rows, Map_Columns] = size(TerrainMap);          % Gets the size of the matrix Map.

Tx = ((CellSize*BS_x) - CellSize);                   % Transmitter x position in meters.
Ty = ((CellSize*BS_y) - CellSize);                   % Transmitter y position in meters.
Tz = (double(TerrainMap(BS_y, BS_x)) + Height);      % Transmitter z position in meters.

PL_Reference = 20*log((4*pi*c)/(Frequency*10^6));    % Reference Path Loss.

for i = 1:Map_Rows
   Ry = ((CellSize*i) - CellSize);                   % Receiver y position in meters.
   for j = 1:Map_Columns
      Rx = ((CellSize*j) - CellSize);                % Receiver x position in meters.
      Rz = double(TerrainMap(i,j));                  % Receiver z position in meters.
      r = sqrt((Tx-Rx)^2 + (Ty-Ry)^2 + (Tz-Rz)^2);   % Distance between Transmitter and Receiver.
      PL_FreeSpace = 20*log(r);                      % Free Space Path Loss.
```

```
      [OutputHigh, OutputX, OutputY] = Line_of_Sight(TerrainMap, BS_x, BS_y, i, j);
      Hx = ((CellSize*OutputX) - CellSize);
      Hy = ((CellSize*OutputY) - CellSize);
      DistanceMax = sqrt((Hx-Tx)^2 - (Hy-Ty)^2);
      PL_Diffraction = DiffractionCalculation(DistanceMax, r, Tz, Rz, OutputHigh, Height, Frequency);

      Loss = PL_Reference + PL_FreeSpace + PL_Diffraction;    % Loss between Transmitter and Receiver in dBm.
      Power_Received = EIRP - Loss;                % Received Power in dBm.

      if Power_Received > Noise_Floor
         SignalMap(i,j) = Power_Received;          % Sets the current pixel to the received signal strength.
      else
         SignalMap(i,j) = Noise_Floor;             % Sets the Noise Floor for the signal strength.
      end
   end
end

OutputMap = SignalMap;
```

Function: DiffractionCalculation

```
function PL_Diffraction = DiffractionCalculation(DistanceMax, TR_Separation, Tz, Rz, HighPoint, Height,
          Frequency, TerrainVector)

% Constants.
c = 3*10^8;

if DistanceMax > 0
   if TR_Separation > 0
      % Path Clearance Ratio.
      Zmax = ((Tz - Rz)/TR_Separation)*(DistanceMax - TR_Separation) + Rz;
      F = Zmax - double(HighPoint);
      D1 = DistanceMax;
      D2 = TR_Separation - DistanceMax;

      F1 = sqrt((c*D1*D2)/(Frequency*(D1+D2)));

      x = F/F1;

      % Knife Edge Loss Calculation.
      if x > -0.5
         KnifeEdgeLoss = 1.377*x^2 - 11.31*x + 6;
      else
         KnifeEdgeLoss = -(50.4/(1.6 - x)) + 36;
      end

      % Roundness Factor.
%     Slope = (double(HighPoint) - Tz + Height)/DistanceMax;
%     r = [1:round(DistanceMax)];
%     Z = Tz + (Slope*r) - Height;
%     StandardDeviation = 0;
%     for i = 1:length(Z)
%        StandardDeviation = StandardDeviation + (Z(i) - TerrainVector(i))^2;
%     end
%     H = 0.9*StandardDeviation;
```

```
%       R = 75/(H + 75);
   end
else
   KnifeEdgeLoss = 0;
%    R = 0;
end
% Diffraction Path Loss
PL_Diffraction = KnifeEdgeLoss;
```

Function: Line_of_Sight

```
function [OutputHigh, OutputX, OutputY] = Line_of_Sight(TerrainMap, BS_x, BS_y, i, j)

if j ~= BS_x
   Slope = (i - BS_y)/(j - BS_x);
end

HighPoint = TerrainMap(BS_y, BS_x);
HighX = BS_x;
HighY = BS_y;
%TerrainVector = HighPoint;

%[Rows, Columns] = size(TerrainMap);
%MapDebug = zeros(Rows,Columns);

if BS_x < j
   m = [(BS_x+1):j];
   Ceiling = ceil((Slope*(m-j)+i));
   Floor = floor((Slope*(m-j)+i));
%    TerrainVector = zeros(2*length(m));

%    for w = 1:2*length(m)
%       if mod(w,2) == 0
%          TerrainVector(w) = Ceiling(w/2);
%       else
%          TerrainVector(w) = Floor(ceil(w/2));
%       end
%    end

   for x = 1:length(m)
      if TerrainMap(Ceiling(x),m(x)) > HighPoint
         HighPoint = TerrainMap(Ceiling(x),m(x));
         HighX = m(x);
         HighY = Ceiling(x);
      end
      if TerrainMap(Floor(x),m(x)) > HighPoint;
         HighPoint = TerrainMap(Floor(x),m(x));
         HighX = m(x);
         HighY = Floor(x);
      end
%       MapDebug(Ceiling(x),m(x)) = TerrainMap(Ceiling(x),m(x));
%       MapDebug(Floor(x),m(x)) = TerrainMap(Floor(x),m(x));
   end
elseif BS_x > j
```

```
   m = [(j+1):BS_x];
   Ceiling = ceil((Slope*(m-j)+i));
   Floor = floor((Slope*(m-j)+i));
%    TerrainVector = zeros(2*length(m));

%    for w = 1:2*length(m)
%       if mod(w,2) == 0
%          TerrainVector(w) = Ceiling(w/2);
%       else
%          TerrainVector(w) = Floor(ceil(w/2));
%       end
%    end

   for x = 1:length(m)
      if TerrainMap(Ceiling(x),m(x)) > HighPoint
         HighPoint = TerrainMap(Ceiling(x),m(x));
         HighX = m(x);
         HighY = Ceiling(x);
      end
      if TerrainMap(Floor(x),m(x)) > HighPoint;
         HighPoint = TerrainMap(Floor(x),m(x));
         HighX = m(x);
         HighY = Floor(x);
      end
%       MapDebug(Ceiling(x),m(x)) = TerrainMap(Ceiling(x),m(x));
%       MapDebug(Floor(x),m(x)) = TerrainMap(Floor(x),m(x));
   end
end

%imagesc(MapDebug);
%axis xy;
%colorbar;
%MapDebug = zeros(Rows, Columns);

OutputHigh = HighPoint;
OutputX = HighX;
OutputY = HighY;
```