Project	**IEEE 802.16 Broadband Wireless Access Working Group <http://ieee802.org/16>**
Title	**Inter-cell interference in mesh networks**
Date Submitted	**2002-11-11**
Source(s)	Nico van Waes Nokia Wireless Routers
Re:	Ballot 8a
Abstract	This document provides discussion and derivation of an analysis model for inter-cell interference in mesh networks. It also provides a Monte Carlo simulation tool Lastly, it provides proposed text for P802.16.2a with simulation results
Purpose	Adoption of proposed text for P802.16.2a
Notice	This document has been prepared to assist IEEE 802.16. It is offered as a basis for discussion and is not binding on the contributing individual(s) or organization(s). The material in this document is subject to change in form and content after further study. The contributor(s) reserve(s) the right to add, amend or withdraw material contained herein.
Release	The contributor grants a free, irrevocable license to the IEEE to incorporate material contained in this contribution, and any modifications thereof, in the creation of an IEEE Standards publication; to copyright in the IEEE's name any IEEE Standards publication even though it may include portions of this contribution; and at the IEEE's sole discretion to permit others to reproduce in whole or in part the resulting IEEE Standards publication. The contributor also acknowledges and accepts that this contribution may be made public by IEEE 802.16. The above release shall hold only for pages 16 through 21. Pages 1 through 15 remain the sole property of Nokia Inc. The contents of these pages may be used freely for co-existence evaluation purposes and may be freely distributed and reproduced in whole, but shall not be sold, in whole or in part and shall not be included in any commercial product, in whole or in part without the express approval of Nokia Inc.
Patent Policy and Procedures	The contributor is familiar with the IEEE 802.16 Patent Policy and Procedures <http://ieee802.org/16/ipr/patents/policy.html>, including the statement "IEEE standards may include the known use of patent(s), including patent applications, provided the IEEE receives assurance from the patent holder or applicant with respect to patents essential for compliance with both mandatory and optional portions of the standard." Early disclosure to the Working Group of patent information that might be relevant to the standard is essential to reduce the possibility for delays in the development process and increase the likelihood that the draft publication will be approved for publication. Please notify the Chair <mailto:chair@wirelessman.org> as early as possible, in written or electronic form, if patented technology (or technology under patent application) might be incorporated into a draft standard being developed within the IEEE 802.16 Working Group. The Chair will disclose this notification via the IEEE 802.16 web site <http://ieee802.org/16/ipr/patents/notices>.

Inter-cell interference in mesh networks

Dr. Ir. Nico van Waes
Nokia Wireless Routers

1. Introduction

The study of inter-cell interference in mesh networks is substantially different from that of classical PMP architectures. As there is no elevated BS (or CS), and the SSs (or CPEs) are worst case using omni-directional antennas instead of highly directional antennas aimed at the cell-center, the main source of inter-cell interference is no longer the BS's high duty-cycle transmission. Instead, in a mesh network, each node within a cell is a potential source of interference to adjacent cells.

In this document, a simple simulation model is introduced which illustrates the concept of mesh inter-cell analysis.

2. Simulation scenario

The scenario consists of a mesh network consisting of nodes spaced randomly within the sector supported by a Mesh Gateway. It is assumed that each Mesh Gateway site contains a number of sectors, and that the coverage area of these sectors is circular with a certain radius. This principle is shown in Figure 1.

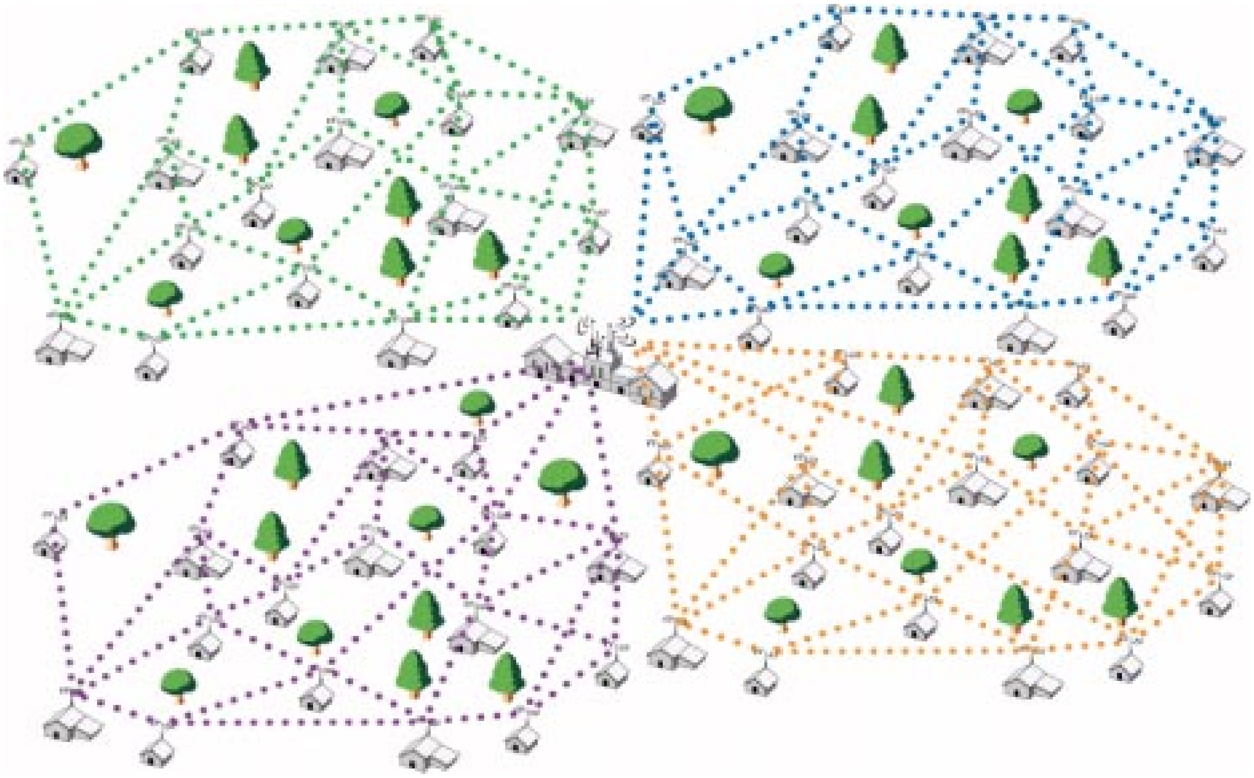


Figure 1—Multi-sector mesh deployment

It is assumed that a classical re-use pattern is used for different Mesh Gateway sites.

3. Scenario analysis

The probability of interference from a node is dependent on its Tx activity. For mesh nodes, three types of transmissions occur:

1) Data transmissions that originate from the node itself;
2) Data re-transmissions for other nodes;
3) Control information transmissions.

Data transmissions from a node depend on the activity profile of the user behind a node. It is considered that the network is used for residential deployments, where there typically is only one active user.

The number of retransmissions depends on the topology of the network.

3.1 User activity profile

Table 1 shows a profile for a user, which is substantially more active than typically seen in active networks. The profile shows the number of sources transferred and the number of transfers for non-streaming applications, as well as the packet size and number of packets transferred for streaming and interactive applications. For the purpose of residential deployments, the usage per user is equivalent to the bandwidth requirements per deployed system, as typically only one user simultaneously uses the system (i.e. there is no demand aggregation per deployed system as common in business deployments).

Table 1—User activity profile

Service	average transaction size (kB)	average frequency during busy hour	bandwidth required (kB)	notes
Email	3	2	6	
Instant messaging	0.1	5	0.5	
HTTP (browsing)	200	20	4000	
audio streaming	9600	0.5	4800	5 minutes at 2x128 kbps (server -> client
voice (over IP)	2880	0.5	2880	5 minutes at 64 kbps (peer <-> peer)
IRC	0.1	600	60	60 packets/min, 10 minutes (server <-> client)
text based gaming	0.2	2500	2000	250 packets/min, 10 minutes (server <-> client)
graphics based gaming	0.2	10000	2000	1000 packets/min, 10 minutes (server <-> client)
audio files	3500	1	3500	
video / software files	100000	0.02	2000	
		Total bandwidth demand (B_D)	20	MB

The usage probability is the factor consisting of the likelihood that a particular user will use the connection during the busy hour, and the duration of that connection. Assuming that the average user will use the connection half an hour every workday of the week provides a reasonable assumption.

3.2 Tx activity computation

Computing the Tx activity of a mesh node is not entirely straightforward and somewhat dependent on the implementation. Due to the overhead associated with each data burst, it is necessary to compute the average data burst size and from there the number of OFDM symbols (assuming the use of the WirelessMAN-OFDM or HIPERMAN PHY).

Using the packet distribution approximation shown in Table 2 (derived from measurement data from CAIDA), the average data burst size may be computed as (ignoring granularity effects):

$$\text{average data burst size} = \frac{\sum_{\text{packet distribution}} \text{percentage of traffic} \times \text{packet size}}{\text{average data bytes per OFDM symbol}} \tag{1}$$

Table 2—Packet distribution approximation

% of traffic	Packet size (Bytes)	Notes
15	1500	Max. size TCP with path MTU discovery
22	560	Mix of 552 and 572 bytes max. size TCP without path MTU discovery
50	48	Min. size TCP
13	400	Mean of remaining traffic

The resultant number of OFDM symbols required per node, ignoring granularity, can be computed as:

$$\begin{aligned}\text{OFDM symbols} = B_D \times \frac{\text{average data burst size + OFDM preamble size}}{\text{average data burst size}} \times \text{usage probability} \times \text{average number of hops} + \\ \frac{\text{MSH\_CTRL\_SLOTS} \times \text{MSH\_CTRL\_SLOT\_SIZE}}{\text{NODES}} \times \frac{3600}{\text{frame duration}}\end{aligned} \tag{2}$$

The first part of the expression reflects the number of OFDM symbols required for the data-transfer, whereas the second reflects the worst-case overhead added by the mesh control sub-frame during the busy hour.

The Tx activity may then be computed as:

$$\text{Tx activity} = \frac{\text{OFDM symbols} \times \text{FFT size} \times (1 + T_g/T_b)}{\text{BW} \times (F_s/\text{BW})} \tag{3}$$

where BW is the channel bandwidth, F_s/BW is the oversampling ratio and T_g/T_b is the cyclic prefix ratio.

It will be assumed that all nodes have the same average Tx activity, which is worst-case when computing inter-cell interference, as the nodes on the cell-boundary in practice always have the lowest Tx activity.

3.3 Routing

The routing strategy, i.e. the strategy how to transfer traffic from a node to the BS, has a substantial impact on the number of retransmissions. In practice, routing strategies can be extremely complex and real-time dynamic.

For the purpose of this document, two static routing strategies will be considered:

Hop minimization, Modulation maximization

Using this strategy, the number of hops is minimized for each node, after which a path is sought from each node to a node with lower hopcount, which has the maximum modulation for that link.

This strategy typically leads to the use of very long links to the Mesh Gateway with low modulation orders using the maximum transmission power. From an intercell interference perspective, this results in a worst-case scenario.

Energy/bit minimization

Using this strategy,. each node seeks to minimize its Xmt energy/bit to the Mesh Gateway, regardless of the number of hops. For WirelessMAN / HIPERMAN compliant devices, this parameter is distributed through the MSH-NCFG message.

This strategy typically leas to the use of short links using very high orders of modulation, but tends to result in a fairly high hop-count to reach the Mesh Gateway. From an intercell interference perspective, this results in a favorable scenario.

4. Simulation setup

For a cell, a number of nodes is randomly established within the coverage area of the Mesh Gateway. For each of these links, the burst profile used is maximized within the limits of the linkbudget. Subsequently, each node’s power is reduced such that there is no power for each link in excess of what is necessary to achieve the desired fading margin.

Subsequently, a route is sought for each node to the Mesh Gateway using either of the routing protocols described in 3.3.

Assuming that each node on average has the same total bandwidth demand B_D, the average burst profile (in terms of bytes/OFDM symbol) is computed, tracing the route from each node to the mesh Gateway. From this, using the equations developed in Clause 3, the average Tx activity per node is computed.

For the interfered cell, the interference is computed for the nearest omni-directional node, but not the mesh gateway of that cell, as the mesh gateway will benefit substantially from the front-to-back ratio of its sector antenna. To compute the interference, a transmission is traced from each node to the Mesh Gateway, and the interference from each link recorded. As no two nodes transmit at the same time within this simple setup, the cumulative interference probability density function is easily computed.

5. Simulation source code

5.1 Header file

```
/*
    Scenario parameters
*/

#define FILENAME "mesh.m"
/*  antenna gain (dB) */
```

```
#define ANTGAIN     9
/* mean power at the antenna port (dBm) */
#define XMTPWR     21
/* fading margin (dB) */
#define FM 6
/* Tolerated I/N */
#define INTERFERENCE_TO_NOISE_RATIO -6
/*  pathloss exponents for different parts of the desired path */
#define PATHLOSS_EXPONENT_DESIRED {2,2.3}
/* Distances over which each of the desired pathloss components is valid */
#define PATHLOSS_DISTANCE_DESIRED {1,50,100000}
/* pathloss exponents for different parts of the interfering path */
#define PATHLOSS_EXPONENT_INTERFERENCE {2,3,4}
/* Distances over which each of the interfering pathloss components is valid */
#define PATHLOSS_DISTANCE_INTERFERENCE {1,50,500,100000}
/* Standard Deviation */
#define PL_STD_DEV 7
/* Center frequency (Hz) */
#define FC 350000000
/* Number of nodes/sector */
#define NODES 100
/* Cell radius (m) */
#define CELLRADIUS  3200
/* Interfered node distance (should be 2*CELLRADIUS for large scale network computation) */
#define INT_DIST 6400
/* cell sector size */
#define SECTOR M_PI_2
/* channel bandwidth (Hz) */
#define BW 7000000
/* user data during busy hour (MB) */
#define USER_DATA   20
/* user average active time during busy hour (fraction of hour) */
#define USER_ACTIVE_TIME 0.5
/* number of realizations of links to interfered nodes */
#define REALIZE_INTERFERENCE 10
/* number of realizations of deployment */
#define REALIZE_CELL 10
/* Either MINIMIZE_EPB or MINIMIZE_HOPS routing*/
#define ROUTING_TYPE MINIMIZE_HOPS
/* Either CO_CHANNEL, ADJ_CHANNEL or ALT_ADJ_CHANNEL */
#define CHANNEL_REJECTION CO_CHANNEL

/*
    WIRELESSMAN-OFDM / HIPERMAN parameters
*/

/*  preamble size (OFDM symbols) */
#define PREAMBLE_SIZE 2
/* cyclic prefix size (OFDM symbol duration fraction) */
#define CP 1.0/32.0
/* bandwidth oversampling fraction */
#define OVERSAMPLING 8.0/7.0
/* FFT size */
#define FFT_SIZE 256
```

```
/* bytes per OFDM symbol */
#define BYTES_PER_SYMBOL { 24, 36, 48, 72, 96, 108 }
/* mesh control slots */
#define MSH_CTRL_SLOTS 6
/* mesh control slot size (OFDM symbols) */
#define MSH_CTRL_SLOT_SIZE 7
/* frame duration (ms) */
#define FRAME_DURATION 20
/* receiver sensitivity (7 MHz) (dB)  for QPSK-1/2, QPSK-3/4, QAM16-1/2, QAM16-3/4, QAM64-2/3, QAM64-
3/4 */
#define RECEIVER_SENSITIVITY    {-84,-82,-77,-75,-71,-69}
/* channel rejection (dB) */
#define CO_CHANNEL 0
#define ADJ_CHANNEL 29
#define ALT_ADJ_CHANNEL 48
```

5.2 Source file

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <time.h>
#include <limits.h>
#include "mesh.h"

/* Some physical constants
*/

/* Receiver Thermal Noise (dBmW) */
#define RCV_TN -108.0 +10*log10(BW/1000000.0)
/* Speed of light (m/s) */
#define C   29800000

/* Some internal definitions
*/
#define DOUBLE_MAX  (double)INT_MAX
#define DOUBLE_MIN  (double)INT_MIN
#define EPS 0.0000001
#define FALSE 0
#define TRUE 1
#define DESIRED 0
#define INTERFERENCE 1
#define MAXHOPS 10

typedef struct location
    {
        double x,y;
    } Location;

#define MINIMIZE_HOPS 1
#define MINIMIZE_EPB  2
static double RCVSENS[]   = RECEIVER_SENSITIVITY;
static double PL_EXP_D[]  = PATHLOSS_EXPONENT_DESIRED;
```

```
static double PL_EXP_I[]  = PATHLOSS_EXPONENT_INTERFERENCE;
static double PL_DIST_D[] = PATHLOSS_DISTANCE_DESIRED;
static double PL_DIST_I[] = PATHLOSS_DISTANCE_INTERFERENCE;
static double INR         = INTERFERENCE_TO_NOISE_RATIO;
static int    BPS[]       = BYTES_PER_SYMBOL;
double RefPathLoss;

double dmax(double x, double y) { return (x>y)?x:y;}
double dmin(double x, double y) { return (x>y)?y:x;}

static int newGap(int gap) {
/* A subroutine for the combsorting algorithm.
*/
    gap = (gap * 10) / 13;
    if (gap == 9 || gap == 10)
        gap = 11;
    if (gap < 1)
       gap = 1;
    return gap;
}

static void combsort(double a[], int aSize) {
/* Combsort, a derivative of bubble-sort but much better ;^)*/
    int i,j;
    double tmp;
    int gap, swapped;

    gap = aSize;
    for (;;) {
    gap = newGap(gap);
        swapped = FALSE;
    for (i = 0; i < aSize - gap; i++) {
            j = i + gap;
            if (a[i] > a[j]) {
                tmp = a[i];
                a[i] = a[j];
                a[j] = tmp;
                swapped = TRUE;
            }
        }
        if (gap == 1 && !swapped)
            break;
    }
}


double randGaussian() {
/*
  The polar form of the Box-Muller transformation.
  Box, G.E.P., Muller M.E., 'A Note on the generation of drand48om normal deviates,
  Annals Math. Stat, V.29, pp. 610-611, 1958.
*/

    double x1,x2,w;
```

```
     do {
          x1= 2.0*drand48() -1.0;
          x2= 2.0*drand48() -1.0;
          w = x1*x1+x2*x2;
     } while (w >= 1.0);
     w = sqrt( (-2.0*log(w) ) / w);

     return  x1*w;
}

Location randLocInCircle() {
     Location newLoc;
       double a,r;

       a = SECTOR*drand48();
       r = CELLRADIUS*drand48();
       newLoc.x = r*sin(a);
     newLoc.y = r*cos(a);

     return newLoc;
}

double distance(Location x, Location y) {
/* The distance between two locations */
     return sqrt( (x.x-y.x)*(x.x-y.x)+(x.y-y.y)*(x.y-y.y) );
}

double pathloss_dB(double Distance, int dORi) {
/* Multi-component pathloss calculation */
     int i;
       double pl;

       pl = RefPathLoss;
     if (dORi== DESIRED) {
          for ( i=0;i<sizeof(PL_EXP_D)/8; i++ ) {
               if( Distance >= PL_DIST_D[i] )
                    pl += 10*PL_EXP_D[i]*log10( dmin(PL_DIST_D[i+1],Distance)/PL_DIST_D[i] );
          }
     } else {
          for ( i=0;i<sizeof(PL_EXP_I)/8; i++ ) {
               if( Distance >= PL_DIST_I[i] )
                    pl += 10*PL_EXP_I[i]*log10( dmin(PL_DIST_I[i+1],Distance)/PL_DIST_I[i] );
          }
     }
       return pl + PL_STD_DEV*randGaussian();

}

int dijkstraIteration(int Picked[], int Route[], int Hops[], int Links[][NODES]) {
     int i,j;
     int bestNode;
     int bestHops;
```

```
    bestHops = INT_MAX;

    for ( i=0; i<NODES; i++ ) {
        if (( !Picked[i] ) && (Hops[i] < bestHops )) {
            bestNode = i;
            bestHops = Hops[i];
        }
    }
    if ( bestHops == INT_MAX) {
        return 0;
    }

    i = bestNode;
    Picked[i] = 1;

/* Update all neighbors of the picked node */

    for ( j=0; j<NODES; j++ ) {
        if ( i!=j && Links[i][j] && Hops[j]>Hops[i]+1 ) {
                Hops[j] = Hops[i]+1;
                Route[j] = i;
        }
    }

    return 1;
}

void powercontrol(double pathLoss[][NODES], int links[][NODES], double xmtPwr[][NODES] ) {
/* Minimizes the power on each link given the selected modulation for that link*/
    int i,j;

    for ( i=0;i<NODES; ++i ) {
        for ( j=0;j<NODES; ++j ) {
                xmtPwr[i][j] = DOUBLE_MIN;
                if(i!=j && links[i][j])
                    xmtPwr[i][j] = pathLoss[i][j] - 2*ANTGAIN + RCVSENS[links[i][j]-1] + FM;
        }
    }
}

double modulation_avg( int route[],int links[][NODES] ) {
/* Compute the average modulation being used, keeping in mind that some links are more equal than others
   (i.e. some links are used for re-transmissions for other links */
    int i,j,k,l;
      int modulation[sizeof(RCVSENS)/8];

    memset(modulation,0,sizeof(RCVSENS)/8*sizeof(int));

    for( l=0,j=0,i=0;i<NODES;++i ) {
        k = i;
        do {
            j = route[k];
            if(j!=k) {
                modulation[links[k][j]-1]++;
```

```
                    l++;
                }
            } while(k=j);
        }

        for ( j=0,i=0;i<sizeof(RCVSENS)/8;++i)
            j += modulation[i]*BPS[i];

        return ((double)j)/((double)l);
}

double avgNumberOfHops(int hops[]) {
/* Compute the average number of hops for a burst */

        int hopCnt[MAXHOPS];
        int i;
        double avgNrHops;

        memset(hopCnt,0,MAXHOPS*sizeof(int));

        for ( i=1; i<NODES; i++ ) {
            if ( hops[i] < MAXHOPS )
                hopCnt[hops[i]]++;
            else
                hopCnt[0]++;
        }

        for ( avgNrHops=0.0,i=1;i<MAXHOPS; i++ )
            avgNrHops += hopCnt[i]*i;
        avgNrHops /= (double) NODES;

        return avgNrHops;
}

double txActivity(int hops[],double avgMod,double avgNrHops) {
/* Computes the TxActivity for each node, taking the overhead of the WirelessMAN / HIPERMAN standard into
account */
        double avgBurstSize,OFDMsymbols;

/* The average burst size (in bytes). The packet distributed is approximated from CAIDA
   measurement data.
*/
        avgBurstSize = (0.15*1500+0.22*560+0.5*48+0.13*400)/avgMod;

/* The number of OFDM symbols of the average node during the busy hour.
   Start by adding the preamble overhead (in bytes).
   Then the overhead caused by multiple hops, the user active time and convert from bytes to
   OFDM symbols.
   Lastly, add the control subframe overhead to it.
*/
        OFDMsymbols  = USER_DATA*1024.0*1024.0/avgBurstSize*(avgBurstSize+PREAMBLE_SIZE);
        OFDMsymbols *= avgNrHops*USER_ACTIVE_TIME/avgMod;
        OFDMsymbols         +=         MSH_CTRL_SLOTS*MSH_CTRL_SLOT_SIZE/NODES*3600000/
FRAME_DURATION;
```

```
/* The Tx activity  as fraction of the busy hour.
*/
     return OFDMsymbols/(BW*OVERSAMPLING/FFT_SIZE*(1+CP))/3600;

}


long intercell_interference( int route[], double xmtPwr[][NODES], Location pos[],
          Location posI, double intLevel[],long intLevel_index ) {
/* Computes the interference from each node in the cell to the interfered node, tracing along the route from each
node to the Mesh Gateway */
     int i,j,k;
       double pl;

     for( j=0,i=0;i<NODES;++i ) {
          k = i;
          do {
               j = route[k];
               if(j!=k) {
                    pl = pathloss_dB(distance(pos[i],posI),INTERFERENCE);
                    intLevel[intLevel_index++]= xmtPwr[k][j] + 2*ANTGAIN - pl - CHANNEL_REJECTION;
               }
          } while(k=j);
     }
     return intLevel_index;
}

void routing(int route[],int hops[],int links[][NODES],double xmtPwr[][NODES],int type) {
/* Provides the option between the routing strategy based on “minimum hop, maximum modulation” and “energy
per bit minimization (it actually uses energy/byte, but it’s the same difference) */
     int picked[NODES];
     int i,j;
     int changes;
     double xmtEpB[NODES];
     double t;

     if ( type == MINIMIZE_HOPS ) {
/* Routing based on hop-minimization, followed by modulation-maximization.
*/
          memset(picked,0,NODES*sizeof(int));
          i =1;
          while (i)
               i = dijkstraIteration(picked,route,hops,links);

          for (i=0; i<NODES; i++ ) {
               for ( j=0; j< NODES; j++ ) {
                    if (hops[j] == hops[route[i]] && links[i][j] > links[i][route[i]])
                         route[i] = j;
               }
          }
     }

     if (type == MINIMIZE_EPB) {
```

```
/* Routing based on energy-minimization.
*/
        memset(xmtEpB,0,NODES*sizeof(double));

        for (i=0; i<NODES; ++i)
            xmtEpB[i] = DOUBLE_MAX;
        xmtEpB[0] = 0; /* Mesh Gateway */

        for (;;) {
            changes = FALSE;
            for( i=0;i<NODES;i++ ) {
                for( j=1;j<NODES;j++ ) {
                    if( i!=j && links[i][j]) {
                        t= pow(10.0,xmtPwr[j][i]/10.0)/BPS[links[j][i]-1];
                        if(xmtEpB[i] + t + EPS< xmtEpB[j] ) {
                            hops[j] = hops[i]+1;
                            route[j] = i;
                            xmtEpB[j] = xmtEpB[i] + t;
                            changes = TRUE;
                        }
                    }
                }
            }
            if (changes == FALSE)
                break;
        }
    }
}

void filedump(double intLevel[],int intLevel_index,double TxActivity,double avgNrHops) {
/* Dumps the results of the simulation to a MatlabTM formatted file. Running this file from Matlab plots the resulting interference cdf. */
    FILE *fp;
    int i;
    char fn[20];
    double t;

#if (ROUTING_TYPE == MINIMIZE_HOPS)
    sprintf(fn,"%s\0","meshHops.m");
#elif (ROUTING_TYPE == MINIMIZE_EPB)
    sprintf(fn,"%s\0","meshEpB.m");
#endif
    if((fp = fopen(fn,"wr"))== NULL) {
        printf("Failed to open file: %s.\n",FILENAME);
        exit(-1);
    } else {
#if (ROUTING_TYPE == MINIMIZE_HOPS)
    sprintf(fn,"%s\0","Hops");
#elif (ROUTING_TYPE == MINIMIZE_EPB)
    sprintf(fn,"%s\0","EpB");
#endif
        fprintf(fp,"Iv%s = [\n",fn);
        for (i=0;i<intLevel_index;i++)
            fprintf(fp,"% 4.3f\n",intLevel[i]);
```

```
        fprintf(fp,"];\nTxActivity%s = % 2.5f;\n",fn,TxActivity);
        fprintf(fp,"Realizations%s = %d;\n",fn,REALIZE_INTERFERENCE);
        fprintf(fp,"Cells%s = %d;\n",fn,REALIZE_CELL);
        fprintf(fp,"AvgNrHops%s = % 2.5f;\n",fn,avgNrHops);
        t = TxActivity/REALIZE_CELL/REALIZE_INTERFERENCE/avgNrHops;
        fprintf(fp,"semilogy(Iv%s,fliplr([1:size(Iv%s,1)])*% 2.8e)\n",fn,fn,t);
        fprintf(fp,"axis([-140 -90 0.0001 1]);\n");
        fprintf(fp,"hold on;\n");
        fprintf(fp,"line([% 2.3f % 2.3f],[0.0001 1]);\n",RCV_TN+INR,RCV_TN+INR);
        fclose(fp);
    }
}

void filedumpcell(Location pos[],Location posI,int links[][NODES],int route[], int cell) {
/* Dumps the cell-scenario to a Matlab™ file. Running this file from Matlab will provide a graphic representation
of the routing tree. The thickness of the lines indicates the modulation used (higher modulatio, thicker line).
    FILE *fp;
    int i;
    char fn[20];

#if (ROUTING_TYPE == MINIMIZE_HOPS)
    sprintf(fn,"scenarioHops%d%s\0",cell,FILENAME);
#elif (ROUTING_TYPE == MINIMIZE_EPB)
    sprintf(fn,"scenarioEbP%d%s\0",cell,FILENAME);
#endif
    if((fp = fopen(fn,"wr")) == NULL) {
        printf("Failed to open file: %s.\n",fn);
        exit(-1);
    } else {
        fprintf(fp,"figure;\n");
        fprintf(fp,"axis([0 % 2.3f 0 %d]);\n",dmax(CELLRADIUS,INT_DIST),CELLRADIUS);
        fprintf(fp,"hold on;\n");
        for(i=0;i<NODES;i++) {
            fprintf(fp,"plot(% 2.3f, % 2.3f,'o');\n",pos[i].x,pos[i].y);
            if(i!=route[i]) {
                fprintf(fp,"line([% 2.3f % 2.3f],[% 2.3f % 2.3f],'LineWidth',% 1.2f);\n",
                pos[i].x,pos[route[i]].x,pos[i].y,pos[route[i]].y,(double)links[i][route[i]]/4.0);
            }
        }
        fprintf(fp,"plot(% 2.3f, % 2.3f,'x');\n",posI.x,posI.y);
        fprintf(fp,"Coverage = [");
        for(i=0;i<CELLRADIUS;i++)
            fprintf(fp,"% 2.3f ",sqrt((double)(CELLRADIUS*CELLRADIUS-i*i)));
        fprintf(fp,"];\nplot([0:%d],Coverage,'-.');\n",CELLRADIUS-1);

        fclose(fp);
    }
}


int main() {

    int hops[NODES];
    int links[NODES][NODES];
```

```
    double pathLoss[NODES][NODES];
    int route[NODES];
    Location pos[NODES];
    double xmtPwr[NODES][NODES];
    double intLevel[MAXHOPS*NODES*REALIZE_INTERFERENCE*REALIZE_CELL];
    double avgMod,avgNrHops,TxActivity;
    Location posI;

    int i,j,k,cell;
    long intLevel_index,changes;

    srand( (unsigned) time(NULL));
    memset(intLevel,0,MAXHOPS*NODES*REALIZE_INTERFERENCE*REALIZE_CELL*sizeof(double));
    posI.x = INT_DIST;
    posI.y = 0.0;
    intLevel_index = 0;
    RefPathLoss = 20*log10(4*M_PI*FC/C);

    for ( cell=0;cell<REALIZE_CELL;cell++ ) {
        memset(pos,0,NODES*sizeof(Location));
        memset(hops,0,NODES*sizeof(int));
        memset(route,0,NODES*sizeof(int));
        memset(xmtPwr,0,NODES*NODES*sizeof(double));
        memset(links,0,NODES*NODES*sizeof(int));
        memset(pathLoss,0,NODES*NODES*sizeof(double));


        for (i=0; i<NODES; ++i) {
            pos[i] = randLocInCircle();
            hops[i] = INT_MAX;
        }
        hops[0] = 0; /* Mesh Gateway */

        for (i=0; i<NODES; i++) {
           pathLoss[i][i] = DOUBLE_MAX;
            for ( j=0; j<i; j++ ) {
                pathLoss[i][j] = pathloss_dB( distance(pos[i],pos[j]),DESIRED );

                for ( k=0;k<sizeof(RCVSENS)/8;k++ ) {
                    if ( XMTPWR + 2*ANTGAIN - RCVSENS[k] - FM >  pathLoss[i][j] )
                        links[i][j] = k+1;
                }
/* Link symmetry is assumed
 */
                pathLoss[j][i] = pathLoss[i][j];
                links[j][i] = links[i][j];
            }
        }

/* Set the powercontrol for each link
*/
        powercontrol(pathLoss,links,xmtPwr);

/* Set up the routing table and compute the hop-number
```

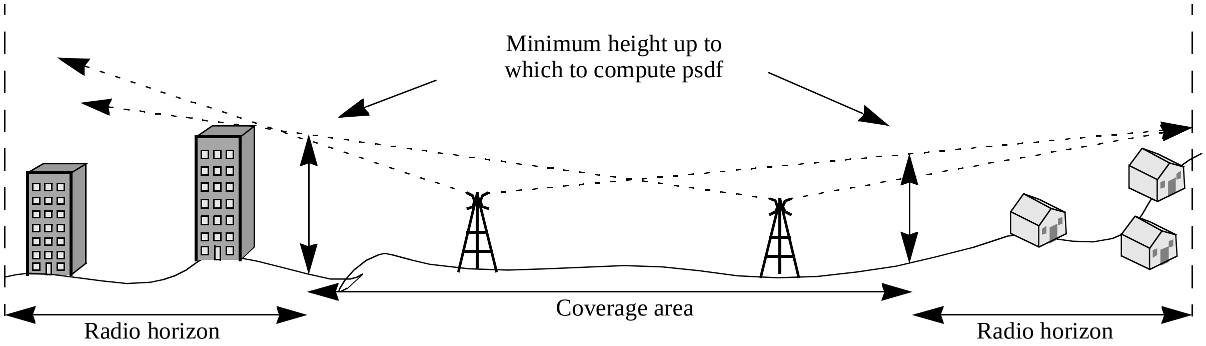
```
*/
        routing(route,hops,links,xmtPwr,ROUTING_TYPE);

/* The average number of hops per burst.
*/
        avgNrHops = avgNumberOfHops( hops );

/* The modulation average produces the average number of bytes/OFDM symbol.
*/
        avgMod = modulation_avg( route,links );

/* The Tx activity as fraction of the busy hour.
*/
        TxActivity = txActivity( hops,avgMod,avgNrHops );

/* Compute the interference to the interfered node from each transmission
*/
        for (i=0; i< REALIZE_INTERFERENCE; i++) {
            intLevel_index = intercell_interference(route,xmtPwr,pos,posI,
                intLevel,intLevel_index);
        }

/* Dump the cell layout (chosen format is Matlab)
*/
    filedumpcell(pos,posI,links,route,cell);
    }

/* Sort the interference values for cdf plotting
*/
    combsort(intLevel,intLevel_index);
/* Dump the results for plotting (chosen format is Matlab)
*/
    filedump(intLevel,intLevel_index,TxActivity,avgNrHops);
}
```

6. Proposed text to be included in P802.16.2a

20.1.1 Methodology

Coordination is recommended between licensed service areas where both systems are operating co-channel, i.e., over the same fixed BWA frequencies, and where the service areas are in close proximity, e.g., the shortest distance between the respective service boundaries is less than the coordination trigger. The operators are encouraged to arrive at mutually acceptable sharing agreements that would allow for the provision of service by each licensee within its service area to the maximum extent possible. Under the circumstances where a sharing agreement between operators does not exist or has not been concluded, and where service areas are in close proximity, a coordination process should be employed.

Fixed BWA operators should calculate the power spectral flux density (psfd) at their own service area boundary. Power spectral flux density should be calculated using good engineering practices, taking into account such factors as propagation loss, atmospheric loss, antenna directivity toward the service area boundary, and the curvature of Earth. The psfd level at the service area boundary should be evaluated for heights up to which reasonably interference to potential devices located within the radio horizon could be expected.



Aggregation may in some cases be needed if the flux contributed by potential interference sources differs less than 3 dB (which generally indicates possible joint direct main beam to main beam coupling between those interference sources and the potential victim system).

Deployment of facilities which generate a psfd, averaged over any 1 MHz at their own service area boundary, less than or equal to that stated in Table 33, should not be subject to any coordination requirements.

Table 33—Maximum psfd limits

Frequency band (GHz)	psfd (dBW/m^2)/MHz
3.5	-125
10.5	-126

20.1.2.2 MP

For MP deployments, generally no LOS exists over the service area boundary. The PMP trigger defined in REF 20.1.2.1 hence needs to be refined for MP deployments. Observing that the tolerated psdf at the receiver should exceed the aggregate psdf produced by all transmitters (including unspecified pathlosses), and assuming for simplicity that all nodes contribute equally to the interference, provides the worst case relation:

$$pathloss > P_{\mathrm{Tx}} - 10\log(BW) + G_{\mathrm{Tx}} + G_{\mathrm{Rx}} - 10\log(kT_0) + N_{\mathrm{F}} + (I/N) + \log(Nodes) \qquad dB \tag{4}$$

where:

kT_0 = -144 dBW/MHz = Equipartition Law
N_{F} = Receiver noise figure
P_{Tx} = Mean power at the antenna port
BW = Occupied bandwidth
$G_{\mathrm{Tx}} = G_{R\mathrm{x}}$ = Antenna gain (Tx/Rx)
I/N = Tolerated Interference to Noise ratio
$Nodes$ = Nodes served on this channel (near this service area boundary)

The mean *pathloss* is composed of several components. The first component is the reference pathloss which is defined as $20\log(4\pi/\lambda)$ dB, where λ is the wavelength. The remaining components follow the propagation model. In the mesh case, Table 1 specifies the first 50 m LOS, followed by d^3 for the next 500 m, followed by d^4 for any excess distance. Hence:

$$\begin{aligned} pathloss(d) &= 20\log(4\pi/0.09) + 20\log(50) + 30\log(500/50) + 40\log(d/500) \\ &= 40\log(d) - 1 \end{aligned} \qquad dB \qquad \forall d > 500m \tag{5}$$

Combining Eq.4 and Eq.5, using the parameters listed in Table 1, hence results in a coordination trigger of 6 km (in comparison, using this analysis for PMP would result in a coordination trigger of 80 km for a single BS, similar to the radio horizon). However, should a mesh deployment be installed substantially above the clutter (which is not recommended), then the coordination trigger as specified for PMP should be applied.

22. Coexistence of Mesh networks

22.1Co-channel intercell interference in a large scale network

In a multi-cellular mesh network, the interference into cells using the same frequency consists of the joint interference from all nodes in the cell. The generated interference depends on the network topology of the cell and the Tx activity of each node within this cell. The logical links that are established by a node determine the transmission power of this node for each of those links (assuming power control is used). When the distance of a logical link is long, the power used, and hence the interference caused, will be higher. On the other hand, if the distance of a logical link is short, it requires more hops to reach the Mesh Gateway, which increases the number of retransmissions and hence the Tx activity of the average node.

The establishment of logical links is directed by the routing algorithm of nodes within the network, which is typically a complex time-variable algorithm. For the purpose of evaluation, the routing algorithm is restricted to the following two algorithms:

Hop minimization, Modulation maximization

Using this strategy, the number of hops is minimized for each node, after which a path is sought from each node to a node with lower hopcount, which has the maximum modulation for that link.

This strategy typically leads to the use of very long links to the Mesh Gateway with low modulation orders using the maximum transmission power. From an intercell interference perspective, this results in a bad-case scenario.

Energy/bit minimization

Using this strategy,. each node seeks to minimize its Xmt energy/bit to the Mesh Gateway, regardless of the number of hops. For WirelessMAN / HIPERMAN compliant devices, this parameter is distributed through the MSH-NCFG message.

This strategy typically leas to the use of short links using very high orders of modulation, but tends to result in a fairly high hop-count to reach the Mesh Gateway. From an intercell interference perspective, this results in a favorable scenario.

In Figure 2 and Figure 3, a typical 100 node scenario, derived using the parameters listed in REF Table 31, is shown using each of the routing methods. Derivation of these scenarios, in which no synchronization between the Mesh Gateway sites is assumed, as well as the simulation tool to compute these scenarios, is provided in [H3.27]. The thickness of the lines represents the modulation order.

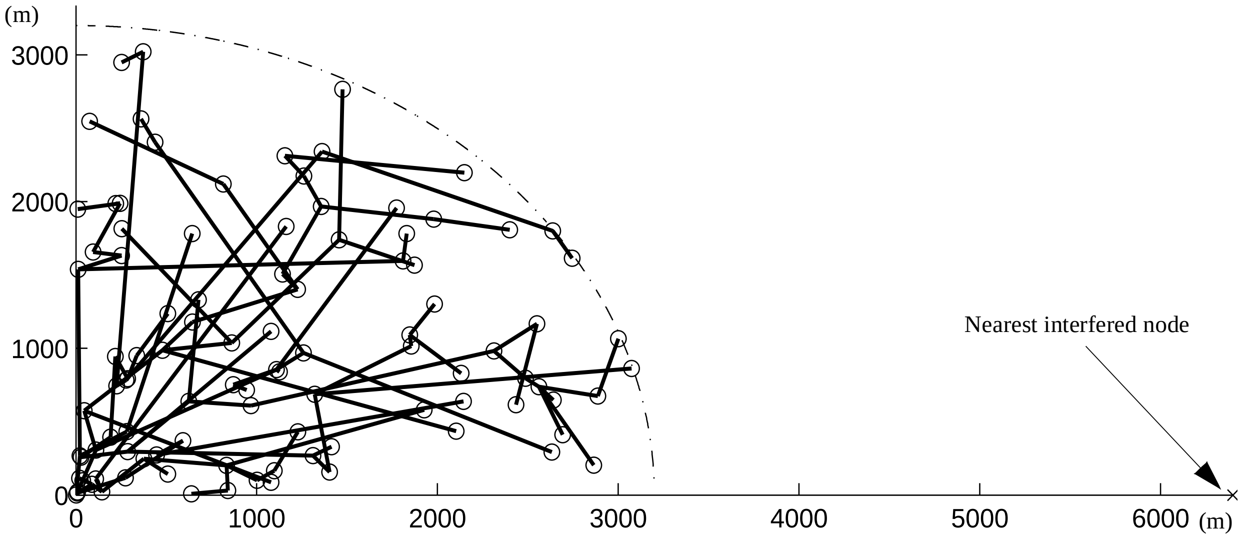


Figure 2—Example mesh scenario using minimized energy-per-bit routing

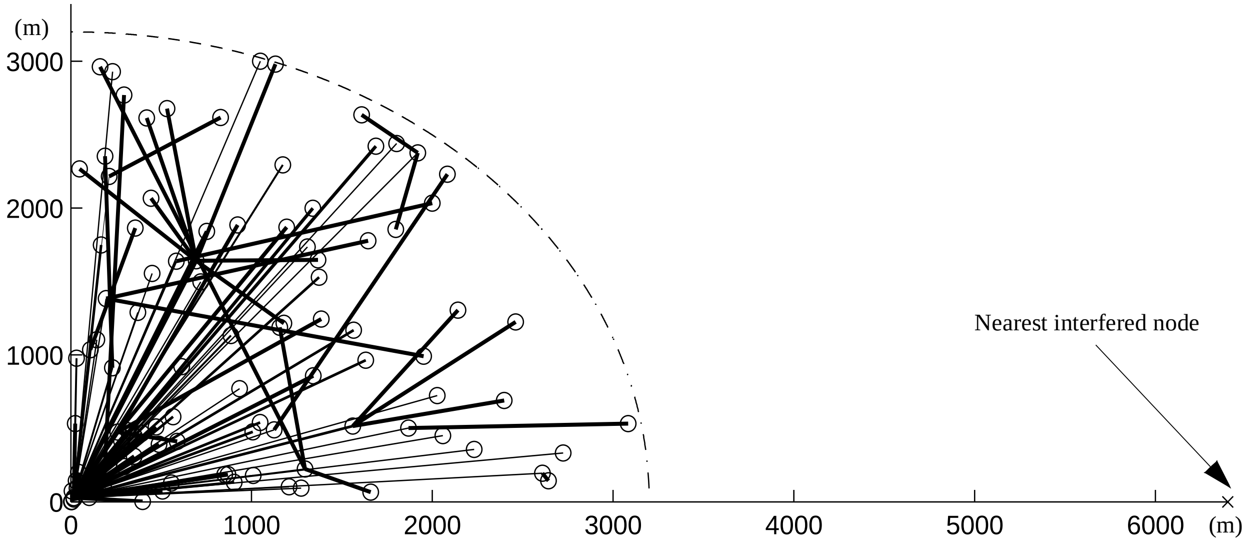


Figure 3—Example mesh scenario using minimized hopcount, maximized modulation routing

Performing a set of Monte Carlo simulations over random scenarios such as those in Figure 2 and Figure 3, the interference to the nearest node in the nearest co-channel cell is computed. For this, it is assumed that a classical frequency re-use pattern of four is used. The nearest co-channel node is hence twice the cell-radius away.

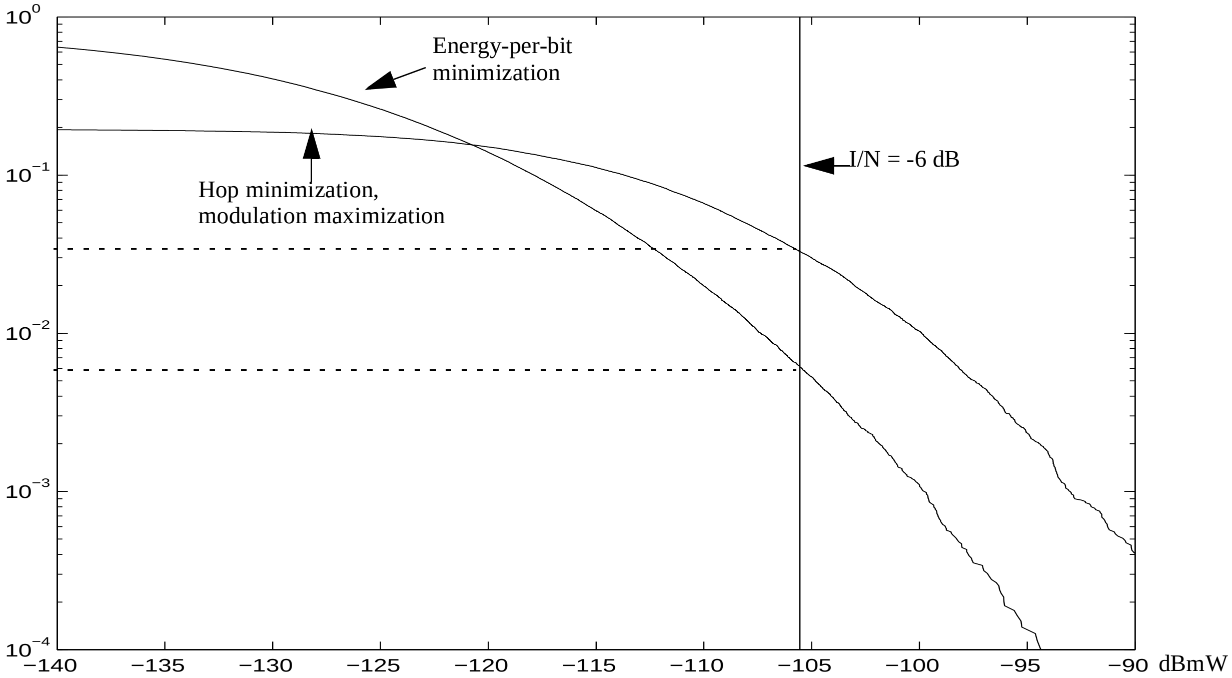


Figure 4—Interference cdf for nearest co-channel node

In Figure 4, the effect of the different routing strategy on the interference to the nearest co-channel node can be seen. It should be noted that the probability of interference (i.e. the probability of the interference power exceeding the *I*/*N* margin, which decreases the link-budget by 1 dB) is generally relatively low, in the order of 2.5% to 0.6%.

Energy-per-bit minimization results in a substantially lower interfererence probability due the typically lower link distances (and related power levels). It does however result in higher hop-counts, resulting in higher Tx Activity factors, which is observable in Figure 4 from the higher probabilities for low interference levels.

22.2 Co-channel inter-network interference between adjacent areas

When two operators serve adjacent areas, co-ordination of channel allocations is recommended within the co-ordination trigger area as described in 20.1.1. The probability of interference into the adjacent operator's co-channel cell is computed in an identical fashion as for large scale networks, with the exception that the distance to the nearest node of the adjacent operator now varies. The resulting curve is hence purely a function of the pathloss model. Using the pathloss model described in REF Table 31 leads to a decrease of 1 dB for every 280 m. The probability of a decrease in linkbudget by 1 dB can hence easily be read from Figure 4 by observing the probability for the x-axis value of -105.6+ (*distance of interferer to cell-center - cell radius*)/280 dB.

22.3 Adjacent channel intercell interference in a large scale network

When deploying Mesh Gateway sites with cell sectors and a classical frequency re-use pattern, adjacent sectors may be using adjacent channels. Performing a Monte Carlo simulation over scenarios similar to Figure 2 and Figure 3, except with the nearest interfered node at 3.4 km (using the depicted 3.2 km cell radius), the probability of interference shown in Figure 5 is derived. It should be noted that the probability of the interference power exceeding the I/N margin, which decreases the link-budget by 1 dB, is generally relatively low, in the order of 0.5% to 0.2%.

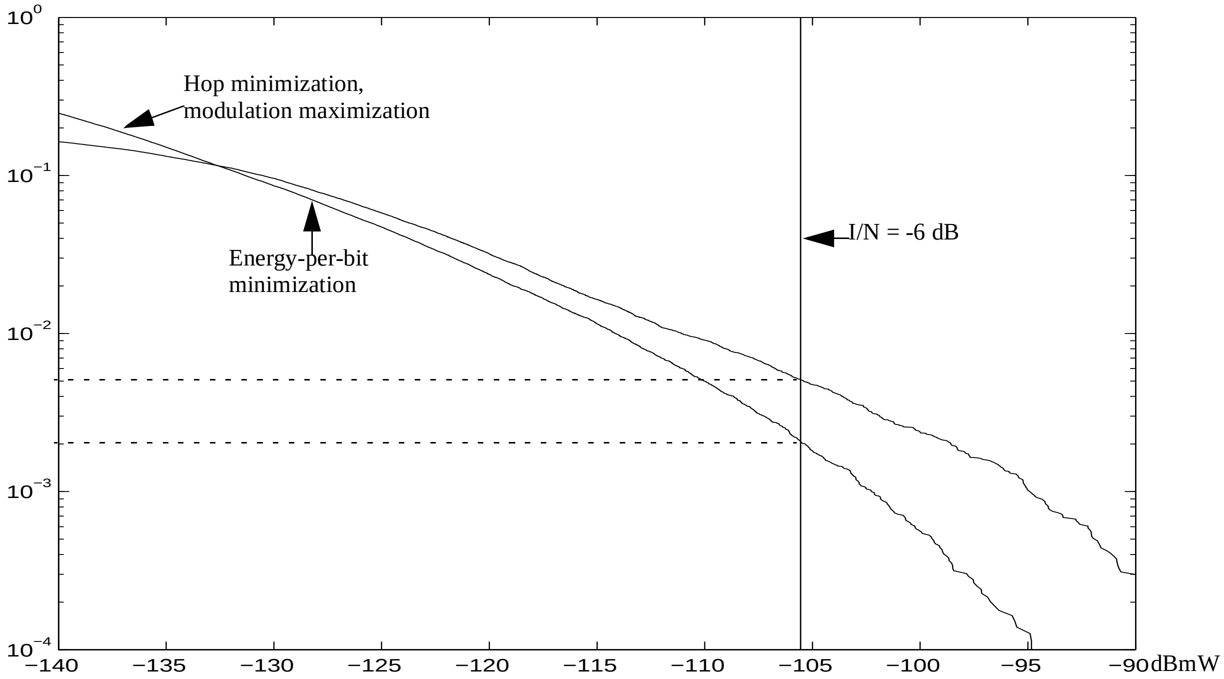


Figure 5—Interference cdf for nearest adj. channel node

22.4 Adjacent channel inter-network interference between adjacent areas

The probability of adjacent channel interference between adjacent areas for networks of different operators is identical to that for multi-cell networks of one operator as derived in 22.3.

22.5 Adjacent channel inter-network interference within the same area

When two mesh networks operate on adjacent channels within the same area, potential for interference will depend largely on the location of the nodes in each of the networks. It is however to be expected that each network will suffer from bursty interference from the other network, as the isolation between nodes will only be in the order of 90 dB. For this case, operation on alternate adjacent channels would in general be recommended.

Annex N: Bibliography

[H3.26] G. Durgin, T.S. Rappaport and H. Xu "Measurements and Models for Radio Path Loss and Penetration Loss In and Around Homes and Trees at 5.85 GHz." IEEE Transactions on Communications, Vol. 46, No. 11

[H3.27] N. van Waes, "Inter-cell interference in mesh networks", C802162a-02_xx

[H3.28] D. Beyer, Fundamental Characteristics and Benefits Of Wireless Routing ('Mesh') Networks ", 8th Annual WCA Technical Symposium, Jan. 15, 2002.