REFERENCES

ActivePerl (2000), Free binary distribution of Perl for Linux, Solaris and Windows. http://www.ActiveState.com/Products/ActivePerl/docs/index.html

Andrew F.S., and John A.M. (1999), *Scenario Management in Web-based Simulation*. In Proceedings of the 1999 Winter Simulation conference. 1430-1437.

Ann M. M (1999), *Web-Based Simulations For Computer-Assisted Learning In The Higher Education Sector*. Engineering Science and Educational Journal, June 1999. 107-114.

BATR (2000). Broadband and ATM Research Group, Faculty of Engineering, Universiti Putra Malaysia. http://cc.eng.upm.edu.my

Beca, L., Cheng, G. and Fox, G.C. *Web Technologies for Collaborative Visualization and Simulation.* Northeast Parallel Architectures Center, 111 College Place, Syracuse, New York 13244. http://www.npac.syr.edu

Berger, M. and Leiner, U. (1997), *Remote Visualization, Manipulation and Simulation in Internet*. In Proceedings of Simulation and Animation '97 Magdeburg, 1-11, Ghent, Belgium; SCS Europe.

Biran, A. and Breiner, M. (1999), *MATLAB 5 for Engineers 2nd Edition,* Addison Wesley, Dorset, 1999. 90, 104.

Crespo, A. and Bier, E.A. (1996), *Webwriter: A Browser-Based Editor for Constructing Web Applications*. http://www5conf.intria.fr/fich_html/papers/P35/Overview.html

Dincer, K. and Fox, G.C. (1997), *Using Java and JavaScript in the Virtual Programming Laboratory: A Web-Based Parallel Programming Environment.* NPAC Technical Report, SCCS.
http://www.npac.syr.edu/pub/by_index/sccs/papers/html/0750/abs-0788.html

Dorwarth, H., Lorenz, P., and Schriber, T.J. (1997). *Towards a Web Based Simulation Environment"*. 1997 Winter Simulation Conference Proceedings pp.1338-1344. http://isgwww.cs.uni-magdeburg.de/~pelo/publikat/wsc/wsc97.html

Elder, M.D. *Simulation Course Handbook- An Overview of How to do Simulation Now.* http://130.159.224.178/mark/sim/simguide.htm

Ernest, H.P., Robert, L.M., and Sean, P.G. *Web-based Simulation In Simjava Using Remote Method Invocation*. Proceedings of the 1997 Winter Simulation Conference, pp. 468-474, Atlanta, GA, 7-10 December 1997.
http://ms.ie.org/page/papers/wsc/wsc97/wsc97.html

Eugene, E.K. (1996), *CGI Developer's Guide*. Sams.net Publishing, Indianapolis. Chapter 9.

Ferscha, A. and M. Richter. (1997), *Java Based Conservative Distributed Simulation*. In Proceedings of the Winter Simulation conference. 381-388.

Fishwick, P. (1995). *What is Simulation?* http://www/cis.ufl.edu/~fishwick/introsim/node1.html

Fishwick, P. (1998). *Digital Circuit Simulation on the Web.* http://www/cis.ufl.edu/~fishwick/dig/

Fishwick, P. *Web-based Simulation: Some Personal Observations*. Dept. Of Computer and Information science and Engineering, University of Florida.

Flanagan, D. (1998), *Javascript: The Definitive Guide, Third Edition*, O'reilly & Associates, Sebastopol, CA.

GNUPlot (2000), *The GNUPlot Home Page.* The Department of Computer Science, Dartmouth College. http://www.cs.dartmouth.edu/gnuplot_info.html

Healy, K.J. and Kilgore, R.A. (1998), *Introduction to Silk and Java-based Simulation*. In Proceedings of the 1998 Winter Simulation Conference. 1649-1654.

Hoffer, J.A., George, J.F., Valacich, J.S. *Modern Systems Analysis and Design*. The Benjamin/Cummins Publishing Company, Inc. 54-60.

Howell, F.W. (1999), *The Simjava Home Page*. http://www.dcs.ed.ac.uk/home/hase/simjava/

IIS (2000), Microsoft Internet Information Server. Microsoft Corporation. http://www.microsoft.com/iis/

Javasim (1999), Javasim Home Page at the University of Newscastle. http://javasim.ncl.ac.uk/

Kian, K.C. and Yung, W.T. (1998), *Simulation Environment and Tools*. Term paper for Advanced Modeling and Simulation Techniques Course. Nanyang University of Singapore. http://www.comp.nus.edu.sg/~teoym/ic52z1/webbase.htm

Leong, M.W., Guan, T.S. (1998), *Web-based Simulation.* Term paper for Advanced Modeling and Simulation Techniques Course. Nanyang University of Singapore. http://www.comp.nus.edu.sg/~teoym/ic52z1/webbase.htm

McNab, R. and Howel, F.W. (1996), *Using Java for Discrete Event Simulation*. Department of Computer Science, University of Edinburgh.

Medeiros, D.J., Watson, E., Carson, J. and Manivannan, M. (1998). *The Rise Of Web-Based Simulation: Implications For The High Level Architecture.* Proceedings of the 1998 Winter Simulation Conference, Eds., pp. 1663-1668, Washington, DC.

MIL3 (1997), *OPNET Tutorial Manual*, Washington DC, 1997.

Omnicron (2000), OmniHTTPd Web Server. Omnicron Technologies Corporation. http://www.omnicron.ab.ca/httpd/

O'Reilly (2000), *Practical Extraction and Report Language*.
http://www.perl.com/pub/doc/manual/html/pod/perl.html

Orfali, R. and Harkey, D. (1997), *Client/Server Programming with Java and CORBA*, Wiley Computer Publishing, New York, 1997.

Phillips, P. (1998), Safe CGI Programming.
http://www.go2net.com/people/paulp/cgi-security/safe-cgi.txt

Pollatschek, M.A. (1999). *Introduction To Simulation*.
http://iew3.technion.ac.il:8080/~bani/simint.html

Rational (1997), *Analysis and Design with UML*. Rational Software Corporation
http://rational.com/

Seibt, F., Schumann, M., Beikirch, J. *Concept and Components for a Web-based Simulation Environment (WBSE)*. http://www.isima.fr/scs/wbms/d6/seibtf.html

Sliman, Y. (2000), *Data Link Control Layer Performance For Wireless ATM Network*, Universiti Putra Malaysia, 2000.

Sol, S. (2000), *Security Issues When Installing and Customizing Pre-Built Web Scripts*
http://Stars.com/Authoring/Scripting/Security/

Stein, L. (2000), *CGI.pm - a Perl5 CGI Library*.
http://stein.cshl.org/WWW/software/CGI/cgi_docs.html

Stein, L. (2000b), The World Wide Web Security FAQ
http://www.w3.org/Security/Faq/www-security-faq.html

Stroustrup, Bjarne (2000), *The C++ Programming Language*
http://akpublic.research.att.com/~bs/C++.html

Swing (2000), Creating a GUI with JFC/Swing.
http://java.sun.com/docs/books/tutorial/uiswing/index.html

VRML (2000), A Beginner Guide to VRML.
http://www.netscape.com/eng/live3d/howto/vrml_primer_body.html

Wall., L. (1999), *Perl, The First Postmodern Computer Language*.
http://www.perl.com/pub/1999/03/pm.html

Whatis (2000), whatis.com - The IT Specific Encyclopedia
http://whatis.com

Yucesan, E., Chen, C.H., Lee, I. *Web-based Simulation Experiments*. In Proceedings of the 1998 Winter Simulation Conference. 1649-1654.

APPENDICES

APENDIX A-1

```
/*
A diode circuit
A Program to generate and show the characteristic of semiconductor
circuit.

This program proves that for any negative voltage, a very small
negative current will flow, while for positive voltages the current
will be positive and increase rapidly with v. We use the range1 and
range2 variables represent the voltage domains.

      1. range1     minimum voltage
      2. range2     maximum voltage

(taken from [Example 2.13], MATLAB5 for Engineers, page 104)

   C/C++ Program written by:
     Tan Kee Leong (GS 3656)
     ATM and Broadband Research Group
     Dept. of Computer and Communication System,
     Engineering Faculty, Universiti Putra Malaysia
*/


#include <iostream.h>
#include <fstream.h>
#include <math.h>
#include <stdlib.h>

int main(int argc, char *argv[])
{    double io, v, i, range1, range2;

     if (argc != 3)
     {    cout << "Usage: diode [range1] [range2]\nExample: fr 1.0
2.0 0.001";
          exit(0);
     }

     ofstream fo("output.txt");
     range1 = atof (argv[1]);
     range2 = atof (argv[2]);

     io = 1.0e-6;

     for (v=range1; v<=range2; v+=0.005)
     {    i = io*(exp(40*v)-1);
          fo << v << "\t" << i << endl;
     }

     fo.close();
     return 0;
}
```

APENDIX A-2

```
/*
     Fourier Series
     A Program to generate a truncated Fourier series given:
      1. A     amplitude of wave
      2. f     angular frequency, where w = f * pi
      3. p  increment points

     (taken from Example 2.5, MATLAB5 for Engineers, page 90)

     C/C++ Program written by:
     Tan Kee Leong (GS 3656)
     ATM and Broadband Research Group
     Dept. of Computer and Communication System,
     Engineering Faculty, Universiti Putra Malaysia
*/

#include <iostream.h>
#include <fstream.h>
#include <math.h>
#include <stdlib.h>

#define pi 3.142

int main(int argc, char *argv[])
{  double p, A, f, t, x, x1, x2, x3, omega;

//   to convert a string to float
//   A=1.0; f=2.0; p = 0.001;

     if (argc != 4)
     {    cout << "Usage: fourier [amplitude] [frequency]
[points]\nExample: fr 1.0 2.0 0.001";
          exit(0);
     }

     ofstream fo("output.txt");
     A = atof(argv[1]);   //amplitude of wave
     f = atof(argv[2]);   //angular frequency
     p = atof(argv[3]);   //no of points

     omega = f * pi;

     for (t=-2; t<=2; t+=p)
     {    x1 = cos (omega*t);
          x2 = -cos (3*omega*t)/3;
          x3 = cos (5*omega*t)/5;
          x = 4*A*(x1+x2+x3)/pi;
          fo << t << "\t" << x << endl;
     }

fo.close();
return 0;
}
```

APENDIX A-3

```
/*     A Program to find the effect of timeout on GBN throughput given:
       1. percentage of packet losses
       2. percentage of check-sum error; and
       3. timeout range

       A Master Thesis simulation program entitled
       "Data Link Control Layer Performance for Wireless ATM Network"
       by Sliman KA.A.Yaklaf (UPM 2000),
       ATM and Broadband Research Group
       Dept. of Computer and Communication System,
       Engineering Faculty, Universiti Putra Malaysia

Simulation program in C written by Dr. Veeraraghavan Prakash (Nov,1999).
Modified to support Websim by Tan Kee Leong (August,2000).
*/

#include<stdio.h>
#include<time.h> /*for giving the time as the the random seed */
#include<stdlib.h> /*for rand() */
#include<math.h>

int main(int argc, char *argv[])
{
            FILE *fp,*fp1;
            float ave=0;
            int run;
            int badpacket,badpacketsize;
            float avebadpacket,avebadpacketsize,avegood;
            avegood=0;
            avebadpacket=0;
            avebadpacketsize=0;

// Added by websim
            int given_timeout2;
            fp = fopen("output.txt", "w");
//          fp=fopen("gbnfixed.txt","a");
//          fp1=fopen("gbnfixed1.txt","a");

              int
timeout,resend,i,max_seq,total_packet_generated,j,temp,temp1;
              int ack, good_packet_generated,payload, rq, timer;
              int given_timeout,check_sum,packetloss;
              float effective_packet, effective_packet_loss,
effective_check_sum;
              struct packet{
                              int size;
                        int type;
                              };
            struct packet sq[8]; /* transmitter queue with 2^n-1 elements
*/
//    Added by websim
            given_timeout2 = atoi(argv[1]);
            packetloss  = atoi(argv[2]);
            check_sum = atoi (argv[3]);

/* Input from the User */
/*        printf("\n*****************************************\n");
              printf("*      please input the following data    *\n");
```

```
             printf("*****************************************\n");
        printf("\n Timeout in msec:");
        scanf("%d", &given_timeout2);
        printf("\n Packet loss in percentage:");
        scanf("%d", &packetloss);
        printf("\n Check-sum error in percentage:");
        scanf("%d", &check_sum);
        srand((unsigned)time(NULL)); /* randomizing the generator*/
              /*end of input from the user
        printf("\n Simulation has started........\n");
 */

for (given_timeout=0; given_timeout<=given_timeout2; given_timeout++)
{

for(run=1;run<=20;run++)
{
             /* Intitialization steps */
      ack=0; /* no ack received */
      rq=1; /* no packet at the receive buffer */
      good_packet_generated=0; /* no good packets yet */
      total_packet_generated=0; /* no packets genetated yet */
      payload=0; /* no payload accepted yet */
      timeout=0; /* timer has started yet */
      resend=0; /* not in the resend mode */
      max_seq=0; /* initially sq is empty */
      timer=0; /*clock is at rest */
      badpacket=0;/*count for bad packet */
      badpacketsize=0;/*size count for bad packets */

      for (i=0; i <=7; i++)
            {
                  sq[i].size=54;
                  sq[i].type=1;
            }
      while(total_packet_generated <= 50)
            {
                /* trasmitter thread */
                if(ack==1) /* received an acknowledgement */
                        {
                                    /*debug mode*/
 //                                 printf("\nAn Ack received");
                              good_packet_generated=good_packet_generated+1;
                              payload=payload+sq[0].size;
                              /* pushing the rest if there are many*/
                                    if(max_seq > 1)
                                    {
                                          for(i=0;i<=max_seq-2;i++)
                                                  sq[i]=sq[i+1];
                                          ack=0; /*removing the ack */
                                          timer=0;
                                          if(resend == 1)j=j-1;
                                                max_seq = max_seq - 1;
                                    }
                                    else
                                    {
                                    /* to generate the payload */
 //                                 printf("\n Generating the packet");
                                          temp = 54;
                                          sq[0].size = temp;
 //                                       printf("size=%d",sq[0].size);
```

```
                        effective_packet=(int)(temp/54);
                        effective_packet_loss=effective_packet*packetloss;
                        effective_check_sum=effective_packet*check_sum;
                                        /*desiding good or bad */
                        temp=rand()%101;
                        temp1 = rand()%101;

      if((temp<=effective_packet_loss)||(temp1<=effective_check_sum))
            sq[0].type=1;
      else sq[0].type=0;
      //    printf("type=%d\n",sq[0].type);

      total_packet_generated=total_packet_generated+1;

      if (sq[0].type == 1)    {
            badpacket = badpacket+1;
            badpacketsize=badpacketsize+sq[0].size;
      }
            rq = sq[0].type;
            ack = 0;
            timer = 0;
            timeout = 0;
            max_seq = 1;
            goto ti;
            }
      }
       /* If an timeout has occured */
      if (timeout==1)  {
            /* enter into retransmission mode */
            /*degugging */
//          printf("\nTimeout has occured ");
            resend = 1;
            j = 0;
            /* regenerate the packet type (good or bad) */
            effective_packet=(int)( sq[0].size/54);
            effective_packet_loss=effective_packet*packetloss;
            effective_check_sum=effective_packet*check_sum;
            /* desiding good or bad */
            temp=rand()%101;
            temp1 = rand()%101;

      if((temp<=effective_packet_loss)||(temp1<=effective_check_sum))
            sq[0].type=1;
      else sq[0].type=0;
            ack=0;

      rq=sq[0].type; /*retransmission*/

      if (sq[0].type == 1)    {
            badpacket = badpacket+1;
            badpacketsize=badpacketsize+sq[0].size;
      }
      /*debug info*/
//    printf("\nregenerating the packet of size=%d,
      type=%d",sq[0].size,sq[0].type);
      total_packet_generated=total_packet_generated+1;
      j=j+1;
      timeout=0;
      timer=0;
 }
 else  {
```

```
      if (resend==1) {
            /* regenerate the packet type (good or bad) */
            effective_packet=int( sq[j].size/54);
            effective_packet_loss=effective_packet*packetloss;
            effective_check_sum=effective_packet*check_sum;
            /* desiding good or bad packet */
            temp=rand()%101;
            temp1 = rand()%101;

      if((temp<=effective_packet_loss)||(temp1<=effective_check_sum))
            sq[j].type=1;
      else sq[j].type=0;

      if (sq[j].type == 1) {
            badpacket = badpacket+1;
            badpacketsize=badpacketsize+sq[j].size;
      }
      /*debugging info */
//    printf("\nregenerating packet of
      size=%d,type=%d",sq[j].size,sq[j].type);

      total_packet_generated=total_packet_generated+1;

      if (j==0)   {
            rq=sq[j].type;
            ack=0;
            timer=0;
            timeout=0;
      }
      j = j+1;
      if (j >= max_seq) resend=0;
      goto ti; /* force to go to timer*/
      }

      if (max_seq <= 6) /* transmitter queue not full */
      {
            /* to generate the payload */
//          printf("\n Generating the packet");
            temp = 54;
            sq[max_seq].size = temp;
            printf("size=%d",sq[max_seq].size);
            /* to generate whether this is a good or a bad packet */
            effective_packet=(int)(temp/54);
            effective_packet_loss=effective_packet*packetloss;
            effective_check_sum=effective_packet*check_sum;
            /*desiding good or bad */
            temp=rand()%101;
            temp1 = rand()%101;

      if((temp<=effective_packet_loss)||(temp1<=effective_check_sum))
            sq[max_seq].type=1;
      else sq[max_seq].type=0;

      if (sq[max_seq].type == 1)    {
            badpacket = badpacket+1;
            badpacketsize=badpacketsize+sq[max_seq].size;
      }
      //    printf("type=%d\n",sq[max_seq].type);

      total_packet_generated=total_packet_generated+1;
```

```
      if (max_seq == 0) {
            rq=sq[max_seq].type;
            timer=0;
            ack = 0;
            timeout=0;
      }

      max_seq = max_seq + 1;
      }else goto ti;
      } /*end of transmitter part */

      /* timer module */
      ti:timer = timer + 1;
      if (timer >= given_timeout+1) {
            timeout=1;
            timer=0;
      }
      /* receiver module */
      if ((rq == 0)&&(timeout==0)) {
            ack = 1;
      }
      else {ack = 0;}

      } /*end of while*/

      /* receiving the packets in transit */
      for(i=0;i<=max_seq;i++)
            {
                  if(sq[i].type==1) break;
//                printf("\nAck received");
                  payload=payload+sq[i].size;
                  good_packet_generated = good_packet_generated+1;
            }

//Modified by Websim
//          printf("\n Number of Good packets
Generated:%d",good_packet_generated);
//          printf("\n Payload in bytes:%d",payload);
//          printf("\n Total packet %d", total_packet_generated-1);
//          fprintf(fp,"\n%6d %10d %7d %7d %6d %10d
%10d",given_timeout,packetloss,check_sum,good_packet_generated,payload,badp
acket,badpacketsize);
            ave=ave+payload;
            avebadpacket=avebadpacket+badpacket;
            avebadpacketsize=avebadpacketsize+badpacketsize;
            avegood=avegood+good_packet_generated;
}

ave=(int)ave/20;
avebadpacket=(int)avebadpacket/20;
avebadpacketsize=(int)avebadpacketsize/20;
avegood=(int)avegood/20;

//    Added by Websim
//fprintf(fp1,"\n%6d %10d %7d %7.0f %7.0f %7.0f
%7.0f",given_timeout,packetloss,check_sum,avegood,ave,avebadpacket,avebadpa
cketsize);
fprintf(fp,"\n%6d %7.0f",given_timeout,avegood*54);
}
return 0;
}
```

BIODATA OF AUTHOR

The author was born on the 7th December 1975 in Kota Bharu, Kelantan. In 1998, he graduated in Bachelor of Engineering (Electronic/Computer) from Universiti Putra Malaysia, Serdang, Malaysia.

Presently the author is attending the M.Sc. programme specializing in telecommunication at Faculy of Engineering, Universiti Putra Malaysia. His research interest is in web-based application development and computer networking. He is receiving a scholarship award from the government of Malaysia under the PASCA scheme. Listed below is author's email and homepage address.

Email : kltan@engineer.com

Homepage: keeleong.tripod.com