Simulation Analysis of TWACS System-10 and Nertec Automatic Meter Reading Networks

BY

L. Scott Rankin

A Thesis Submitted to the Faculty of Graduate Studies in Partial Fulfillment of the Requirements for the Degree of

MASTER OF SCIENCE

Department of Mechanical and Industrial Engineering

University of Manitoba

Winnipeg, Manitoba

ISBN 0-612-16243-5

Name ____________________

Dissertation Abstracts International and *Masters Abstracts International* are arranged by broad, general subject categories. Please select the one subject which most nearly describes the content of your dissertation or thesis. Enter the corresponding four-digit code in the spaces provided.

Industrial Engineering — SUBJECT TERM

0546 — SUBJECT CODE

UMI

Subject Categories

THE HUMANITIES AND SOCIAL SCIENCES

COMMUNICATIONS AND THE ARTS
Architecture 0729
Art History 0377
Cinema 0900
Dance 0378
Fine Arts 0357
Information Science 0723
Journalism 0391
Library Science 0399
Mass Communications 0708
Music 0413
Speech Communication 0459
Theater 0465

EDUCATION
General 0515
Administration 0514
Adult and Continuing 0516
Agricultural 0517
Art 0273
Bilingual and Multicultural 0282
Business 0688
Community College 0275
Curriculum and Instruction 0727
Early Childhood 0518
Elementary 0524
Finance 0277
Guidance and Counseling 0519
Health 0680
Higher 0745
History of 0520
Home Economics 0278
Industrial 0521
Language and Literature 0279
Mathematics 0280
Music 0522
Philosophy of 0998
Physical 0523
Psychology 0525
Reading 0535
Religious 0527
Sciences 0714
Secondary 0533
Social Sciences 0534
Sociology of 0340
Special 0529
Teacher Training 0530
Technology 0710
Tests and Measurements 0288
Vocational 0747

LANGUAGE, LITERATURE AND LINGUISTICS
Language
General 0679
Ancient 0289
Linguistics 0290
Modern 0291
Literature
General 0401
Classical 0294
Comparative 0295
Medieval 0297
Modern 0298
African 0316
American 0591
Asian 0305
Canadian (English) 0352
Canadian (French) 0355
English 0593
Germanic 0311
Latin American 0312
Middle Eastern 0315
Romance 0313
Slavic and East European 0314

PHILOSOPHY, RELIGION AND THEOLOGY
Philosophy 0422
Religion
General 0318
Biblical Studies 0321
Clergy 0319
History of 0320
Philosophy of 0322
Theology 0469

SOCIAL SCIENCES
American Studies 0323
Anthropology
Archaeology 0324
Cultural 0326
Physical 0327
Business Administration
General 0310
Accounting 0272
Banking 0770
Management 0454
Marketing 0338
Canadian Studies 0385
Economics
General 0501
Agricultural 0503
Commerce-Business 0505
Finance 0508
History 0509
Labor 0510
Theory 0511
Folklore 0358
Geography 0366
Gerontology 0351
History
General 0578
Ancient 0579
Medieval 0581
Modern 0582
Black 0328
African 0331
Asia, Australia and Oceania 0332
Canadian 0334
European 0335
Latin American 0336
Middle Eastern 0333
United States 0337
History of Science 0585
Law 0398
Political Science
General 0615
International Law and Relations 0616
Public Administration 0617
Recreation 0814
Social Work 0452
Sociology
General 0626
Criminology and Penology 0627
Demography 0938
Ethnic and Racial Studies 0631
Individual and Family Studies 0628
Industrial and Labor Relations 0629
Public and Social Welfare 0630
Social Structure and Development 0700
Theory and Methods 0344
Transportation 0709
Urban and Regional Planning 0999
Women's Studies 0453

THE SCIENCES AND ENGINEERING

BIOLOGICAL SCIENCES
Agriculture
General 0473
Agronomy 0285
Animal Culture and Nutrition 0475
Animal Pathology 0476
Food Science and Technology 0359
Forestry and Wildlife 0478
Plant Culture 0479
Plant Pathology 0480
Plant Physiology 0817
Range Management 0777
Wood Technology 0746
Biology
General 0306
Anatomy 0287
Biostatistics 0308
Botany 0309
Cell 0379
Ecology 0329
Entomology 0353
Genetics 0369
Limnology 0793
Microbiology 0410
Molecular 0307
Neuroscience 0317
Oceanography 0416
Physiology 0433
Radiation 0821
Veterinary Science 0778
Zoology 0472
Biophysics
General 0786
Medical 0760

EARTH SCIENCES
Biogeochemistry 0425
Geochemistry 0996
Geodesy 0370
Geology 0372
Geophysics 0373
Hydrology 0388
Mineralogy 0411
Paleobotany 0345
Paleoecology 0426
Paleontology 0418
Paleozoology 0985
Palynology 0427
Physical Geography 0368
Physical Oceanography 0415

HEALTH AND ENVIRONMENTAL SCIENCES
Environmental Sciences 0768
Health Sciences
General 0566
Audiology 0300
Chemotherapy 0992
Dentistry 0567
Education 0350
Hospital Management 0769
Human Development 0758
Immunology 0982
Medicine and Surgery 0564
Mental Health 0347
Nursing 0569
Nutrition 0570
Obstetrics and Gynecology 0380
Occupational Health and Therapy 0354
Ophthalmology 0381
Pathology 0571
Pharmacology 0419
Pharmacy 0572
Physical Therapy 0382
Public Health 0573
Radiology 0574
Recreation 0575
Speech Pathology 0460
Toxicology 0383
Home Economics 0386

PHYSICAL SCIENCES
Pure Sciences
Chemistry
General 0485
Agricultural 0749
Analytical 0486
Biochemistry 0487
Inorganic 0488
Nuclear 0738
Organic 0490
Pharmaceutical 0491
Physical 0494
Polymer 0495
Radiation 0754
Mathematics 0405
Physics
General 0605
Acoustics 0986
Astronomy and Astrophysics 0606
Atmospheric Science 0608
Atomic 0748
Electronics and Electricity 0607
Elementary Particles and High Energy 0798
Fluid and Plasma 0759
Molecular 0609
Nuclear 0610
Optics 0752
Radiation 0756
Solid State 0611
Statistics 0463

Applied Sciences
Applied Mechanics 0346
Computer Science 0984
Engineering
General 0537
Aerospace 0538
Agricultural 0539
Automotive 0540
Biomedical 0541
Chemical 0542
Civil 0543
Electronics and Electrical 0544
Heat and Thermodynamics 0348
Hydraulic 0545
Industrial 0546
Marine 0547
Materials Science 0794
Mechanical 0548
Metallurgy 0743
Mining 0551
Nuclear 0552
Packaging 0549
Petroleum 0765
Sanitary and Municipal 0554
System Science 0790
Geotechnology 0428
Operations Research 0796
Plastics Technology 0795
Textile Technology 0994

PSYCHOLOGY
General 0621
Behavioral 0384
Clinical 0622
Developmental 0620
Experimental 0623
Industrial 0624
Personality 0625
Physiological 0989
Psychobiology 0349
Psychometrics 0632
Social 0451

THE UNIVERSITY OF MANITOBA

FACULTY OF GRADUATE STUDIES

COPYRIGHT PERMISSION

SIMULATION ANALYSIS OF TWACS SYSTEM-10 AND NERTEC AUTOMATIC METER READING NETWORKS

BY

L. SCOTT RANKIN

A Thesis/Practicum submitted to the Faculty of Graduate Studies of the University of Manitoba in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

ABSTRACT

Computer simulation is a straightforward and cost effective method of analyzing the performance of communications networks such as automatic meter reading (AMR) systems. AMR networks are a relatively new technology quickly gaining interest with major utility providers. Manitoba Hydro, as a major utility provider, has undertaken a study to evaluate the suitability of several AMR systems to operations in the City of Winnipeg. As part of this study, Manitoba Hydro desired that a simulation analysis be performed.

The TWACS System-10 and Nertec AMR systems will be modeled using the Simscript II.5 simulation language. The intention is to determine, through simulation, if either system meet basic performance criteria set out by Manitoba Hydro. Further, it is hoped that simulation modeling will demonstrate the strengths and weaknesses of each system within the City of Winnipeg context.

The componentry and logical operation of the two systems will be described. A description of the two models constructed will be provided along with a verification of each. The performance of each system, as well as the sensitivity of each to variations in system parameters, is evaluated through simulation. The results of sensitivity analyses and performance experiments with each model are presented. These results provide insight into the performance which could be expected from either system should they be implemented in the City of Winnipeg. It is determined that, according to simulation results, the TWACS System-10 will meet the basic performance requirements of Manitoba Hydro while the Nertec System will not.

ACKNOWLEDGMENTS

I would like to thank Dr. A. S. Alfa of the Department of Mechanical and Industrial Engineering for his guidance and instruction throughout both my undergraduate and graduate programs. Also, Ted Cotton of Manitoba Hydro for his ongoing efforts with the AMR project.

On an personal level, thanks and appreciation are also due to friends and acquaintances made during my somewhat extended stay at the U of M. The past years have been a pleasure.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

1.1 Background

Automatic meter reading (AMR) systems are an emerging technology quickly gaining interest among public utility providers throughout North America. The efficiency and abilities of such systems offer an attractive alternative to the conventional meter reading process. Current customer information can now be retrieved quickly and easily from a central location over a communications network. AMR technology offers intuitive advantages since meters need not be physically inspected by an operator.

Manitoba Hydro, as a major utility provider, initiated a study to evaluate the suitability of AMR technology to its needs. In the interest of estimating expected performance of AMR systems should they be implemented in the City of Winnipeg, Hydro desired that a simulation analysis of several AMR systems be performed as a part of this study. Three systems of particular interest to Hydro were evaluated, namely the Iris System 2020, the TWACS System-10 and the Nertec AMR system.

Simulation analysis of these systems was initiated in 1992, at which point the Iris System 2020 was modeled and evaluated by Richard Stone. Complete details of this portion of the study may be found in Stone [1]. Later, it was decided to expand the simulation study to include simulation of both the TWACS and Nertec systems, the analysis resulting in this thesis.

1.2 Simulation Study of AMR

Due to the accommodating cost of computer time, simulation analysis has become a common and practical means of evaluating the performance of real world systems. Telecommunications networks, such as AMR networks, are frequently analyzed by means of computer simulation where the underlying system makes analytic solutions difficult to develop.

Here, the performance analysis of two AMR systems by means of computer simulation will be undertaken, namely the TWACS System-10 and Nertec AMR systems. As this work is so closely related to that of Mr. Stone, an introduction to the Iris System 2020 will also be provided. It will become apparent throughout this thesis that these systems are so different in logical operation and componentry, that it is not possible to make direct comparisons of the performance of each. Therefore, it is not the intention of this study to determine which of the three systems is the best, so to speak. Rather, the aim of the overall project is to determine which of the three meet performance requirements specified by Manitoba Hydro and to highlight the abilities and shortcomings of each through individual simulation analysis. The final judgment is left to those at Manitoba Hydro. It is hoped that the information presented here will provide a useful tool in the decision making process.

The remaining chapters of this thesis are summarized as follows: First, a thorough description of the componentry and logical operation necessary for the simulation of the TWACS System-10 and Nertec AMR systems will be presented, along with that of the Iris System 2020 for completeness. The Nertec and TWACS simulation models constructed from this information are then introduced, including a description and verification of each. Theory necessary for proper output analysis of the types of simulation models constructed is reviewed in the following chapter. Next, an analysis of the sensitivity of each system to variation of modifiable parameters is undertaken along with relevant discussion and plots of results. In the subsequent chapter, the simulation results for the Nertec and TWACS models, based on system and traffic parameters obtained from the respective manufacturers and Manitoba Hydro, are presented and discussed. The final two chapters contain a discussion of information generated about each system through the simulation analysis and concluding remarks, respectively.

2. DESCRIPTION OF AMR SYSTEMS

2.1 Introduction

The componentry and logical operation of the AMR systems studied for Manitoba Hydro will be described here. For completeness, an introduction to the Iris System 2020 will provided, along with more detailed descriptions of the Nertec and TWACS systems. The chapter will begin with a description of the service requirements that Manitoba Hydro would like each system to accommodate.

2.2 Manitoba Hydro AMR Service Requirements

Manitoba Hydro requires that each system be modeled under implementation in the City of Winnipeg. According to Hydro this will require each system serve approximately 150 000 customers spread over roughly 100 square miles.

Ideally, each system should report readings three times per day and handle multiple message priority levels. Where applicable, two message priority levels will be included in the study, one to handle standard meter reading, and another for additional functions such as demand readings. Given the characteristics of a specific AMR system, it may or may not be possible to satisfy the service requirements of Manitoba Hydro.

2.3 Nertec AMR System

The Nertec system, produced by Nertec Design, Inc. of Granby Quebec, is the most straightforward of all the AMR systems studied. The entire system consists of only the meter retrofit, (known as the Telereader), and the AMR central server connected by the customer's own phone line. As shown schematically in Figure 1, a Nertec system would consist of a Central Server (CS) and N ($1 \leq N < \infty$) Telereaders, (one for each meter).

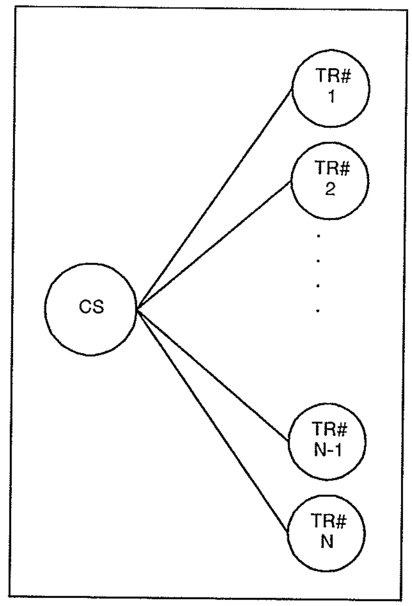

Figure 1: Schematic of Nertec System

Metering information is collected by the Telereader. The Telereader will collect and store a predetermined number of reads each day and, once weekly at a scheduled time, it will transmit the information to the Central Server via a modem dial-in using the customer's own phone line. There is no buffering of data between the Telereader and the central server.

The NERTEC AMR is defined as an inbound system since the communication process is initially triggered by the meter retrofit. In otherwords the decision of when to initiate the data transmission process is made at the Telereader.

The Nertec system is perhaps best described as a retrial queuing system with N servers in parallel, each with a buffer size of zero. As described later, arrivals who fail to receive service from the CS make further attempts according to a retrial schedule.

Expected Arrival Rate

The NERTEC AMR system operates on a seven day cycle. The Telereader collects and stores three reads per day and, once weekly, forwards them to the CS. Thus we expect 1/7 of all meters to dial in each day.

Message Scheduling and Arrival

When a Telereader connects to the CS, in addition to accepting information, the CS will schedule the Telereader's next transmission time. The Nertec system assigns each Telereader a call-in window in an attempt to minimize the probability of a call being blocked, i.e. no free modem available at the CS. Transmission times are scheduled evenly over a primary call-in period, typically eight hours in length from midnight until 8:00 AM.

CS Capacity

The CS can accommodate up to 40 modems. According to information provided by NERTEC, Manitoba Hydro would require 28 modems, using a seven day cycle, to meet its requirements.

CS Service Rate

Each Telereader transmits at the same rate, and messages are of constant length, thus the rate at which the CS serves each Telereader is deterministic. According to Nertec, each Telereader transmission will take approximately 24 seconds.

Retrial Schedule

Should the Telereader fail to transmit its data to the CS successfully, (no free line at CS, customer picks up phone, etc.), the Telereader will reschedule the transmission. The logic of this process, which will be termed the retrial schedule, is depicted in Figure 2. As shown, the Telereader selects a call back interval of 5, 15, 60, or 360 minutes. Should the initial attempt fail, the Telereader will attempt to retransmit after the call back interval has elapsed. The Telereader will repeat this process, after waiting for the same interval, up to four times. If transmission is still unsuccessful, the Telereader will wait for a random period, select a new call back interval, and begin the cycle again. If, in the unlikely event that after four complete cycles no successful transmission has yet taken place, the Telereader will wait eight days and begin the process again.

Issues Relating to Manitoba Hydro AMR Service Requirements

As stated earlier, the Nertec system operates on a seven day cycle, reading approximately 1/7 of all meters each day. Though the Telereader will gather three reads per day, it is only intended to transmit them once a week.

The Nertec system is also unable to handle multiple message priorities. Since each call is initiated by the meter, and each meter must phone in directly to the CS, the CS itself has no way of distinguishing between message types until a connection is actually established. Thus, the CS has no way of giving preferential treatment to a subset of incoming messages.

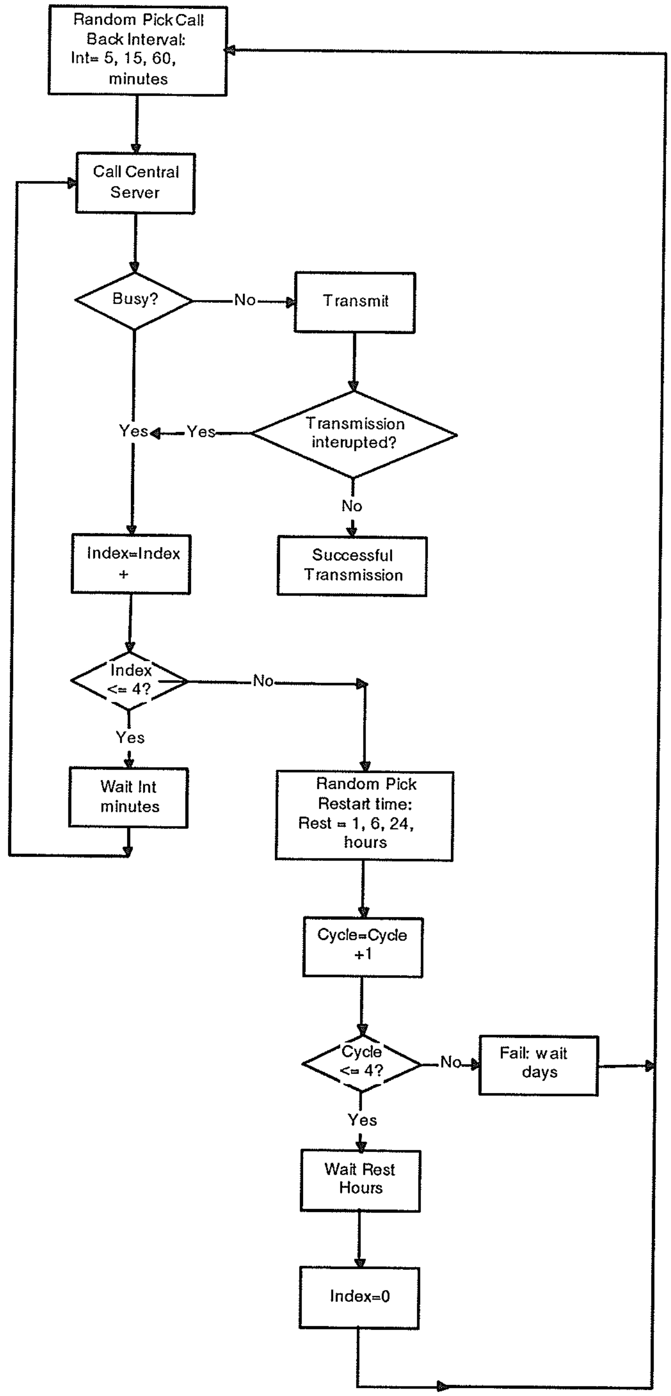

Figure 2: Nertec Retrial Schedule

2.4 TWACS System-10

The TWACS System-10, produced by Distribution Control systems, Inc. (DCSI) of Hazelwood, Missouri, consists of three tiers in the communications chain, they are:

1. Central Control Equipment (CCE)
2. Substation Communication Equipment (SCE)
3. Remote Communication Equipment (RCE)

The hierarchy of the TWACS network is shown in Figure 3.

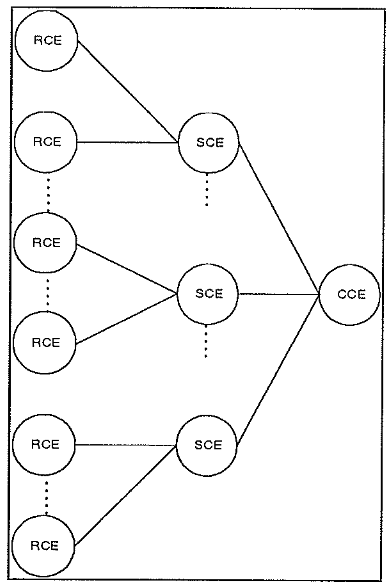

Figure 3: TWACS System-10 Hierarchy

The TWACS system is known as an outbound AMR system, since the decision to send data from the RCE to the central server is initialized by the CCE itself. However the physical act of reading the meter is carried out by the RCE meter retrofit.

TWACS is a hybrid system and thus employs different communication media between the three tiers. Communication between the RCE and the SCE takes place over the existing power network while the SCE and the CCE are typically connected via a telephone link.

To collect metering information, the CCE will transmit an outbound command to the SCE. Each outbound command carries the addresses of 16 meters from which the CCE wishes information. The SCE breaks up the command and forwards a request to each of the 16 meters. Each of the sixteen will then return a response to the SCE. Should the SCE fail to receive a response from any of the sixteen meters, it must send a retry to each failed meter in sequential order. Once all the 16 responses have returned successfully, the SCE assembles them into an inbound command, which will be transmitted to the CCE when the SCE is polled for data.

The focus of the TWACS study will be the relationship between the CCE and the SCEs which can be modeled as a polling system. Henceforth, this reduced system will be known as the TWACS SCE-CCE Network.

2.4.1 TWACS SCE-CCE Network

Generally, the TWACS SCE-CCE network consists of 1 central server and N stations, (where the upper bound of N is determined by the existing power transmission infrastructure.) The central server represents the CCE while the each SCEs is represented by a station with a priority queue for each possible message priority level. According to DCSI, the polling discipline of the SCE-CCE network can be modeled as round robin with negligible switchover time. A schematic of the TWACS SCE-CCE Network with round robin polling is shown in Figure 4. Fundamental parameters of the TWACS SCE-CCE Network will now be explained.

SCE Requirement

An SCE may be placed at any or all of the 81 Manitoba Hydro substations in Winnipeg. No information was available form DCSI specifying a recommended number of SCEs for an implementation in the City of Winnipeg.

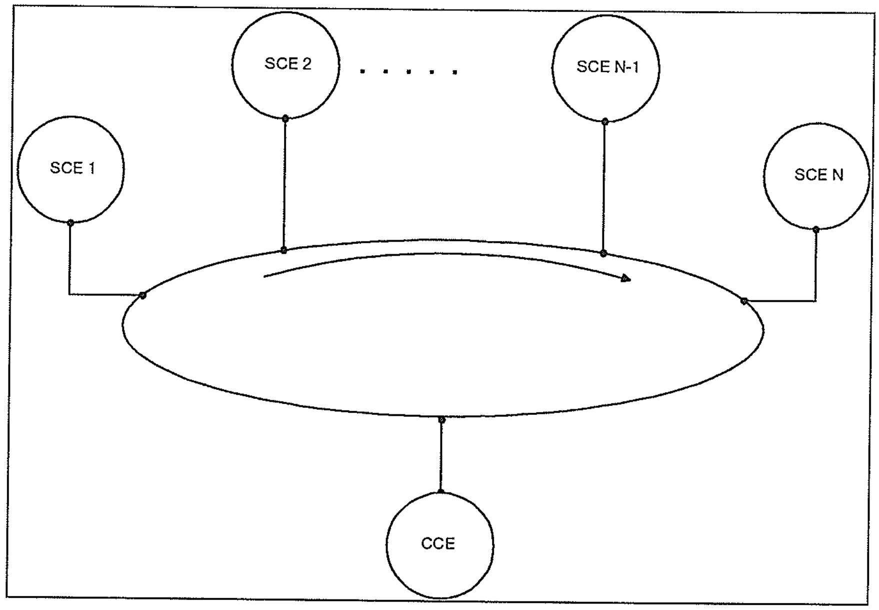

Figure 4: TWACS SCE-CCE Round Robin Polling Discipline

SCE buffers

As mentioned earlier, commands flow through each SCE in both directions, thus each SCE essentially contains a unique queue for inbound and outbound commands. A conceptual representation of the SCE is illustrated in Figure 5. Each SCE has storage capacity for a combined total of 10 outbound and inbound commands.

Command Processing by the SCE

Once an outbound command is sent out to the 16 meters in question, the SCE will have the corresponding inbound command assembled and ready for transmission in 18.2 seconds under ideal circumstances, (i.e.: all 16 meters respond successfully). For each failed response, a retry message must be sent, this requires 6-8 seconds per failed response. Retry messages are sent one at a time. According to DCSI, we can assume that 95% of the initial attempts are successful, thus a typical failed response probability is 5%.

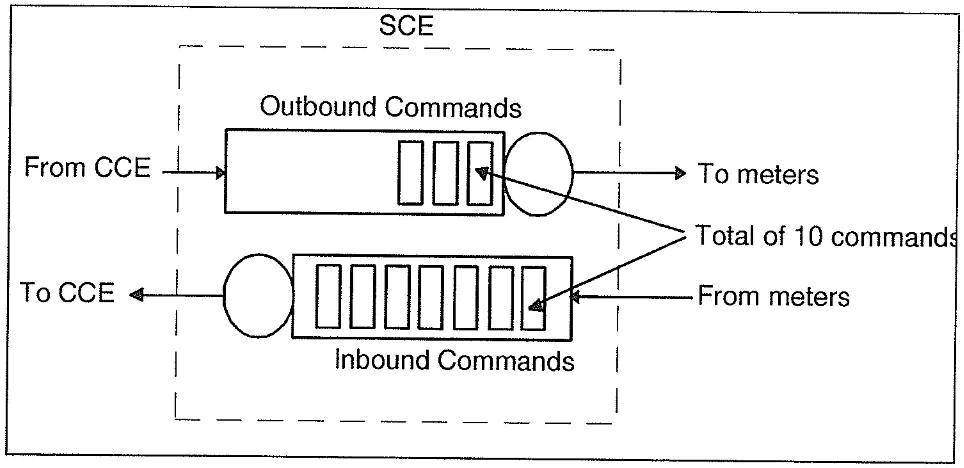

Figure 5: Schematic of SCE Queues

For clarity, the transmission of commands by the SCE to the meters, and the subsequent response, will be referred to as *command processing by the SCE.* This terminology is appropriate since each SCE can be viewed as a server processing commands as they arrive. Only one command, of any type, may be in process by the SCE at a given time.

CS service rate

Data exchange between the CS and the stations takes place at a rate of 2400 bps. Incoming and outgoing commands from the CCE are of fixed length at 238 and 78 bytes respectively, including all data overhead. Thus, when polled, the service time required by each SCE is a function only of the number of inbound and outbound commands transmitted at each SCE.

CCE Polling and Service Discipline

The CCE will poll the SCEs in a sequential, round robin, fashion with negligible switchover time. Any inbound commands buffered by the SCE at the outset of a poll will first be transmitted back to the CCE. The CCE will then transmit enough outbound commands to refill the buffer. The CCE will transmit such that buffer overflows do not occur, i.e. it will first accept all inbound commands from the SCE and then replace them with a like number of outbound commands. The CCE service discipline is illustrated in Figure 1.

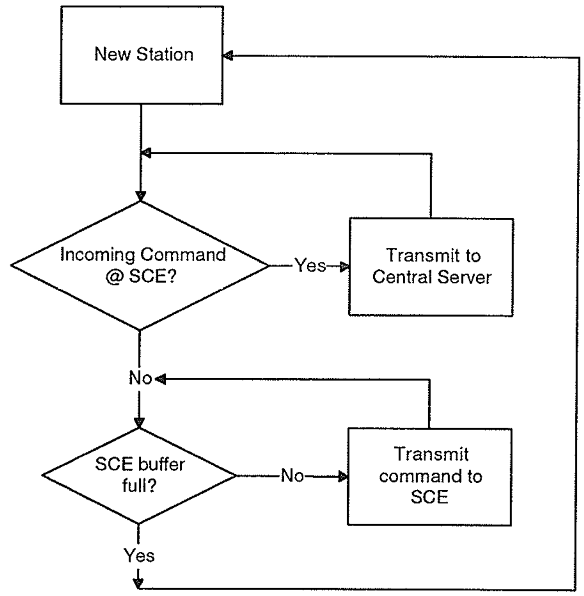

Figure 6: CCE Service Discipline

Priority Messages

The TWACS system can accommodate up to 255 priority levels. In this study, only two priority levels will be considered. Low priority commands are those designated for standard AMR, while high priority is reserved for demand readings. These designations reflect those used by DCSI, and result from the manner in which the TWACS system operates.

According to DCSI, the TWACS system performs the meter reading function by a batch process, all meter reading commands are generated and then sent out continually until all readings have been collected. Conceptually, the system queues all meter reading commands at the CCE, and sends each as soon as transmission is possible. Additional commands, for demand readings for instance, are generated on an as needed basis. When generated during the meter reading cycle, these additional commands must be marked as higher priority so that they needn't wait behind enqueued meter reading commands.

Priority message handling follows intuitive rules. High priority messages are transmitted first during a given poll. If all high priority messages have been sent and buffer space remains at the SCE, low priority commands may then be transmitted. Once at the SCE, high priority commands are processed first and subsequently queued first for transmission back to the CCE. High priority commands may not, however, interrupt a low priority command currently in transmission or processing.

Since high priority commands typically target a single meter rather than a group of 16, the processing and transmission requirements of these commands differ from those of the standard command. High priority commands outbound from the CCE have the same bit-length as their low priority counterparts. Once at the SCE, the high priority message requires the same processing time as the retry messages described above, i.e. 6-8 seconds with the same success rate as standard commands. When returning, high priority commands carry the same digital overhead but approximately 1/16 of the data carried by standard, low priority commands. It will be assumed that transmission time for returning high priority messages is 1/4 the length of low priority commands.

Message Generation

As described in the previous section, standard meter reading commands (low priority) are generated all at once at the outset of the meter reading cycle. High priority commands, to be employed for demand readings, will be generated at a rate of 1500 per hour. This figure is arrived at according to data provided by Manitoba Hydro. Since the distribution of the high priority arrivals is not known it will be assumed that interarrival times follow an exponential distribution.

Issues Relating to Manitoba Hydro AMR Service Requirements

Unlike other AMR systems where the meter retrofit can store several reads and transmit them all at once, the TWACS system is only able to gather one read at a time. This read will represent the usage since the RCE was last accessed. Since Manitoba Hydro wishes three reads per day, a TWACS system would have to perform three complete cycles per day.

2.5 Iris System 2020

Introduction

Stone [1] performed a simulation analysis of the Iris System-2020 for Manitoba Hydro which was completed in 1994. The goal was to judge the suitability of the system to the needs of Manitoba Hydro and, primarily, to determine which of several possible service disciplines provided the best service for an Iris system implemented in the City of Winnipeg. An overview of both the Iris System-2020 and the service disciplines analyzed will be provided here. For complete details and results, the reader is referred to Stone [1].

The Iris System 2020 is produced by Iris Systems, Inc. of Winnipeg, Manitoba, and consists of four communication tiers. At the lower level of the hierarchy is the meter retrofit, known as the Network Service Module (NSM). The NSM collects metering information and then transmits messages through the 1st level repeaters (RPT1s) to the 2nd level repeaters (RPT2s). Messages arriving at RPT2s are stored for eventual transmission to the Central Server (CS). A schematic of the network hierarchy is shown in Figure 7.

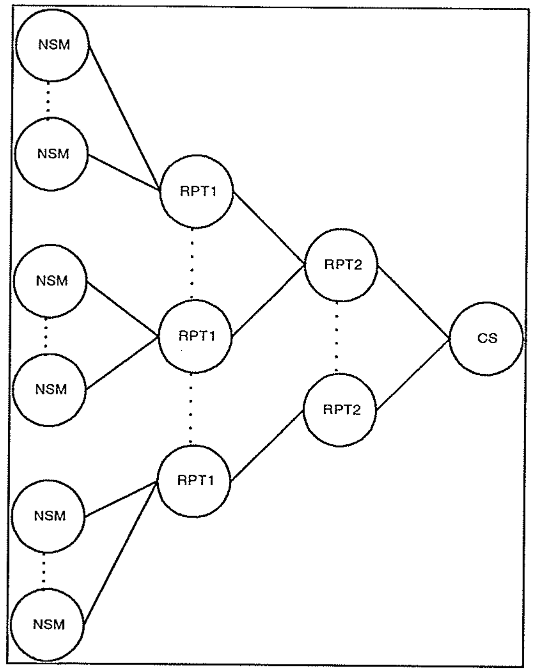

Figure 7: Iris System 2020 Hierarchy

Like the TWACS System-10, the IRIS System 2020 is a hybrid communication network. Communication between the lower three tiers is accomplished via radio wave while the RPT2s and the CS are connected by a telecommunication link. Unlike the TWACS network, System 2020 is an inbound system in that the decision to transmit a message from the meter to the CS is initialized at the meter itself.

Like the TWACS system, the focus of the Iris System 2020 study will be on the upper two tiers of the network hierarchy to be known henceforth as the Iris RPT2-CS Network.

Iris RPT2-CS Network

The Iris RPT2-CS Network can be considered a polling system with N $(1 \leq N < \infty)$ stations and one central server. Each station represents an RPT2 with a priority queue for each level of message priority. An overview of the fundamental parameters of the Iris RPT2-CS Network follows.

RPT2 Switchover Time

Like the TWACS SCE-CCE Network, it was assumed that the Iris RPT2-CS Network is symmetric and consists of identical stations. In other words the switchover and connect time is identical at, and between, each station. Therefore, the switchover time between any two stations is 20 seconds in length and deterministically distributed.

Message Arrival Process

Two message priority levels were considered. Since the true distribution for arrivals of each message type into to RPT2s was unavailable, each was assumed to follow a Poisson distribution.

RPT2 Requirement

Each RPT2 may cover an area of approximately 4 square miles. Therefore, assuming the City of Winnipeg covers an area of approximately 100 square miles, a minimum of 25 RPT2s will be required.

RPT2 Buffers

The RPT2 buffers are of finite capacity, however they are large enough that overflow is unlikely. Therefore, for simplicity, RPT2 buffers were assumed to be infinite.

CS Service Rate

According to Iris Systems Inc., all messages are of constant length at 200 bits. It was assumed that each station transmits at a rate of 9600 bps. Thus message transmission time to the CS is deterministic and identical for each message at 1/48 of a second.

Priority Messages

Two priority message types were included in the analysis performed by Stone. Standard meter reading commands were designated a high priority while, demand readings were designated low priority. These designations resulted from information provided by Iris systems.

CS Service Discipline

The performance of the Iris RPT2-CS Network was evaluated for two CS service disciplines, namely Exhaustive and Time Limited. For each, the CS was assumed to poll the RPT2s in a round robin fashion. A description of each service discipline follows:

Exhaustive Service Discipline:

Once the CS polls a station, it will remain at the station until all messages of both priority types have been transmitted. This service algorithm, assuming a single priority type, is represented in Figure 8.

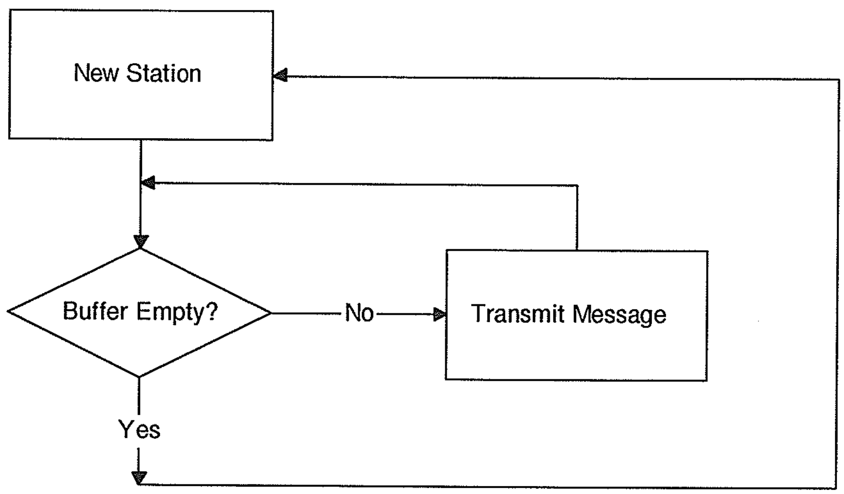

Figure 8: Exhaustive Service Discipline

Time Limited Service Discipline

A representation of the Time Limited Service Discipline is shown in Figure 9. Once the CS polls a station, it will remain at the station until one of the following conditions are met:

1. While serving station i, the station occupation period, T_i, reaches the maximum allowed length t_{max}. The CS will cease service, leaving remaining messages for transmission during a subsequent poll, and switch to the next station in sequence.

2. All messages have been received, and the station occupation period for the current station has exceeded the minimum required length, t_{min}.

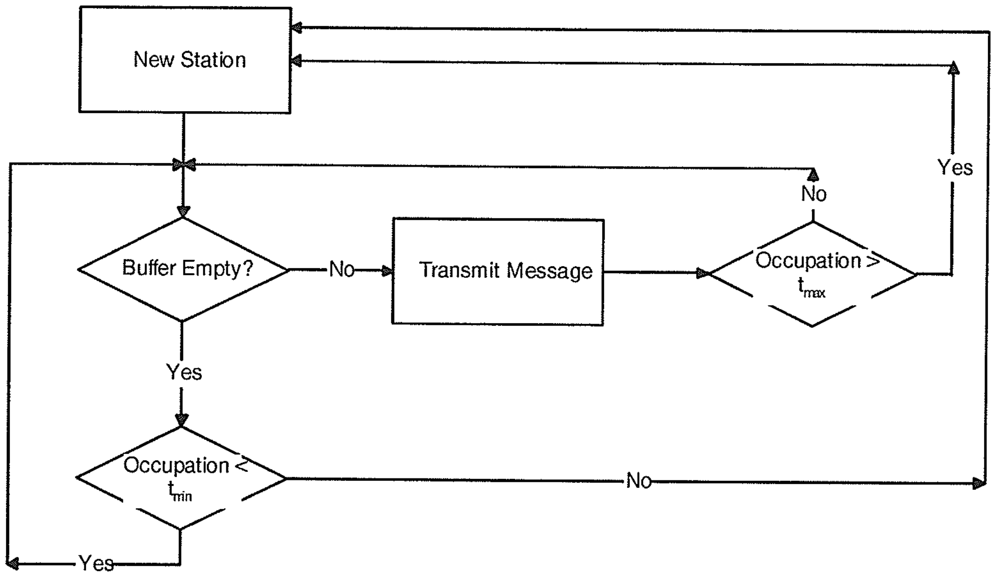

Figure 9: Time Limited Service Discipline

3. SIMULATION MODELS OF AMR SYSTEMS

3.1 Introduction

Each model has been coded in the Simscript II.5 simulation language. This chapter will begin with a brief introduction of Simscript itself. An overview of the models constructed to evaluate each AMR system then follows. Included for each system is a description of the model itself along with the input data required. Verification of each model is then provided by means of a debug trace and exogenous variable check.

3.2 Simscript II.5

Simscript II.5 is a general purpose simulation language conspicuous for its English-like syntax. This characteristic makes the logic of models coded in Simscript easy to follow and understand, even for those unfamiliar with the program. The language has found wide ranging application in the simulation of real world systems.

The current version, II.5, is the product of three decades of development of the original version. This version, which is the proprietary product of CACI, Inc., is available for use on several servers on the University of Manitoba system.

Further information on the structure, syntax, and modeling concepts of Simscript II.5 can be found in Russell [2] and Kiviat [3], a brief introduction is also included in Law and Kelton [4] .

3.3 Nertec Simulation Model

3.3.1 Input Data

The Nertec simulation model requires the following data input:

- Length of primary call-in period in hours, i.e. the typical primary call-in period is from 12:00 AM to 8:00 AM giving 8 hours
- Mean service time
- Number of modems at CS
- Number of days per cycle
- Total meters to be read per cycle

3.3.2 Model Description

General

The Nertec model performs 5 independent replications, resetting all system variables to zero and allowing a runup period for each. The model estimates two quantities, the mean delay and blocking probability, and provides confidence intervals for each. The delay in this case is defined as the elapsed time from the initial attempt until successful connection, it does not include dialing time or the resulting service time.

Message Arrival Scheduling

The Nertec system schedules a specific time for each Telereader to call in to the central server so as to minimize blocked calls. In order to model the arrival process in the Nertec simulation model it is assumed that the interarrival time follows an normal distribution with low variability.

The mean interarrival time is as follows:

$$Mean\ Interarrival\ Time = \frac{(Days\ /\ Cycle) \times (Length\ of\ Call\ in\ Period)}{Total\ Customers}$$

That is, the product of the cycle length, typically 7 days, and the length of the call-in period, typically 8 hours, divided by the total number of meters to be read. New connection attempts continue to arrive until the end of the call in period, while those previously blocked may arrive at any time according to the retrial schedule. After the last day of the cycle is complete, we have:

$$E[New\ Calls] = Total\ Customers\,.$$

Priority Messages

For the reasons described in chapter 2, priority messages are not modeled in the Nertec simulation. Only standard meter reading transmissions are included in the model.

Retrial Schedule

The retrial schedule is modeled as described in section 2.3. However it is assumed that if the Telereader is unable to connect to the CS after four complete cycles, the message transmission is aborted.

Customer Interrupted Transmission

As described in chapter 2, the Telereader transmits over the customers own phone line. Therefore, transmission may be aborted should a customer happen to pick up their phone. The probability of this occurring is arbitrarily set to 2%.

3.3.3 Model Verification

Exogenous Variable Check

Checks were performed to insure that the intended input distributions for quantities such as interarrival times were actually being achieved. These checks were performed on interarrival and service times. The intended mean and standard deviation for interarrival time, for the trial that produced the output in Figure 10, were 1.34 and 0.1 seconds respectively. The service times, on the otherhand were fixed at 24 seconds. As shown in the sample output, the interarrival and service times generated within the model have the intended parameters for their respective distributions.

Exogenous variable check:	Mean	S.D.
-Mean interarrival time:	1.34	.10
-Mean service time:	24	0

Figure 10: Exogenous Variable Check in Nertec Model Output

Debug Trace

In order to verify that the logic of the retrial queue was implemented correctly, a debug trace was performed for the Nertec model. The output of the trace is found in

Table **1** and

Table **2**. In order to illustrate the logic, the CS is given a capacity of zero to ensure blocking. Two messages are generated and the paths of each are traced separately.

As expected, each message is blocked four times per cycle. Upon the first block of each cycle the Telereader, or customer, randomly selects a call back interval. On all subsequent blocks of the same cycle, the telereader waits for the same call back interval before making another attempt. After four attempts, the telereader waits a random period of time before the next cycle begins. A total of four cycles are performed until, since no connection has yet taken place, the message is aborted.

```
Customer #1 created at time=  0.
cust #1 blocked for 1st time at time=  0.  , given call back int= 360 mins
customer #1 blocked for 2 time of cycle 1 at time=   360.00 min
customer #1 blocked for 3 time of cycle 1 at time=   720.00 min
customer #1 blocked for 4 time of cycle 1 at time=  1080.00 min
Cust #1: no cnnct on cycle 1, nxt cyc bgns in  1 hours w/new CB int= 360 mins

customer #1 blocked for 1 time of cycle 2 at time=  1140.00 min
customer #1 blocked for 2 time of cycle 2 at time=  1500.00 min
customer #1 blocked for 3 time of cycle 2 at time=  1860.00 min
customer #1 blocked for 4 time of cycle 2 at time=  2220.00 min
Cust #1: no cnnct on cycle 2, nxt cyc bgns in  6 hours w/new CB int=   5 mins

customer #1 blocked for 1 time of cycle 3 at time=  2580.00 min
customer #1 blocked for 2 time of cycle 3 at time=  2585.00 min
customer #1 blocked for 3 time of cycle 3 at time=  2590.00 min
customer #1 blocked for 4 time of cycle 3 at time=  2595.00 min
Cust #1: no cnnct on cycle 3, nxt cyc bgns in  6 hours w/new CB int=   5 mins

customer #1 blocked for 1 time of cycle 4 at time=  2955.00 min
customer #1 blocked for 2 time of cycle 4 at time=  2960.00 min
customer #1 blocked for 3 time of cycle 4 at time=  2965.00 min
customer #1 blocked for 4 time of cycle 4 at time=  2970.00 min

cust #1 failed on all cycles, destroying cust at time=  2970.00
```

Table 1: Nertec Debug Trace Output for First Message

```
Customer #2 created at time=  2.27
cust #2 blocked for 1st time at time=  2.27, given call back int=   5 mins
customer #2 blocked for 2 time of cycle 1 at time=     5.04 min
customer #2 blocked for 3 time of cycle 1 at time=    10.04 min
customer #2 blocked for 4 time of cycle 1 at time=    15.04 min
Cust #2: no cnnct on cycle 1, nxt cyc bgns in  1 hours w/new CB int= 360 mins

customer #2 blocked for 1 time of cycle 2 at time=    75.04 min
customer #2 blocked for 2 time of cycle 2 at time=   435.04 min
customer #2 blocked for 3 time of cycle 2 at time=   795.04 min
customer #2 blocked for 4 time of cycle 2 at time=  1155.04 min
Cust #2: no cnnct on cycle 2, nxt cyc bgns in 24 hours w/new CB int= 360 mins

customer #2 blocked for 1 time of cycle 3 at time=  2595.04 min
customer #2 blocked for 2 time of cycle 3 at time=  2955.04 min
customer #2 blocked for 3 time of cycle 3 at time=  3315.04 min
customer #2 blocked for 4 time of cycle 3 at time=  3675.04 min
Cust #2: no cnnct on cycle 3, nxt cyc bgns in 48 hours w/new CB int= 360 mins

customer #2 blocked for 1 time of cycle 4 at time=  6555.04 min
customer #2 blocked for 2 time of cycle 4 at time=  6915.04 min
customer #2 blocked for 3 time of cycle 4 at time=  7275.04 min
customer #2 blocked for 4 time of cycle 4 at time=  7635.04 min

cust #2 failed on all cycles, destroying cust at time=  7635.04
```

Table 2: Nertec Debug Trace for Second Message

3.4 TWACS SCE-CCE Network

3.4.1 Input Data

The TWACS CCE-SCE simulation model requires the following data input:

- Number of meters to read
- Number of SCEs in AMR network
- Failed response probability
- Switchover time
- High priority arrival rate (commands/hour)

3.4.2 Model Description

General

The SCE-CCE model performs 10 independent replications and calculates estimates and confidence intervals of two quantities which should be interpreted within the TWACS context. The first quantity is the mean command delay, for both high and low priority messages. Recall however that a command represents the metering information of 16 meters. Therefore for a given command, we will have 16 meters with identical message delays occurring in parallel.

Since the TWACS network can only transmit one read per message it will be necessary to read each meter three unique times per day in order to meet Manitoba Hydro Service requirements. This requirement leads to the second estimate of interest, the mean completion time, i.e. the mean time required to read all meters once. If the mean completion time is short enough, it may be possible to finish three cycles within a 24 hour period.

Simultaneity Issues Regarding SCE Polling

There are two simultaneity issues regarding SCE polling. Specifically, when an SCE is polled, the following potentialities must be addressed:

1. Are commands which complete processing by the SCE during a poll also transmitted back to the CCE?
2. Do commands currently in process by the SCE during a poll take up room in the buffer since they have technically been sent out to the individual meters to request information?

In response to these questions, the CCE-SCE simulation model makes the following assumptions. First, only those commands ready for transmission back to the CCE at the outset of the poll will be transmitted, in other words the buffer is said to be gated. Further, a command is assumed to take up one unit of buffer space from the time of its initial arrival until its transmission back to the SCE. Essentially, the model senses the number of completed commands at the outset of a poll, accepts them and transmits an equal number back to the SCE.

Priority Messages

As described in chapter 2, the TWACS System-10 can accommodate multiple message priority levels. Two priority levels will be included in the model, one for standard meter reading transmission, and a second for demand readings.

SCE Switchover Time

The dedicated telecommunications links between the CCE and each SCE are assumed identical and employ identical modems. According to DCSI switchover time is negligible, thus minimal and identical switchover times between successive SCEs are assumed.

Command Processing Reliability

The number of retries required per command is assumed to follow a binomial distribution with n=16 and p=0.95. This distribution will result in a mean of .8 failed reads of the 16 reads in each command and a standard deviation of .87. The time required for either a retry or a high priority command is assumed uniformly distributed between six and eight seconds. Retries and high priority commands are assumed to have a 100% success rate.

Network Symmetry

For simplicity, it is assumed that the load on the network is symmetric. That is to say that each SCE is responsible for an equal share of the total commands to be processed. The model can however, be easily extended to accommodate asymmetric loads.

3.4.3 Model Verification

Exogenous Variable Check

Checks were performed to insure that the intended input distributions for quantities such as interarrival times were actually being achieved. These checks were performed on the following processes:

Process	Intended Distribution	Expected Mean	Expected Mean
Mean LP Reads Lost	Binomial	0.80	0.87
HP Interarrival Time	Exponential	2.40	2.40
LP command processing	Deterministic	18.2	0.00
HP command processing	Uniform	7.00	0.50

Table 3: Intended Distributions and Parameters for TWACS SCE-CCE Model

As shown in Figure 11, the sample output confirms that the four processes have the intended mean and standard deviation for the respective distributions.

```
Exogenous Vars:
   -Mean # Reads lost per Cmnd:        .800         S.D.:       .871
   -High Prior. Iarr.:                2.40 (sec)    S.D.:      2.39 (sec)
   -Mean LP SCE proc. time:          18.20 (sec)    S.D.:       .00 (sec)
   -Mean HP SCE proc. time:           7.00 (sec)    S.D.:       .58 (sec)
```

Figure 11: Exogenous Variable Check of TWACS SCE-CCE Model Output

Debug Trace

In order to verify the logic of the polling algorithm within the simulation model, a debug trace was performed. A portion of the trace output is found in Table **4**. The terms used in the output are explained as follows. The out.q contains the commands enqueued for transmission back to the CCE while the in.queue contains those commands awaiting processing by the SCE itself. The quantities n.out.q and n.in.q represent the number of commands contained in the out.q and the in.q respectively.

For exemplary purposes, the network in this test consists of two SCEs. The switchover time is set to 100 seconds so that each SCE will be sure to have completed commands to transmit at each poll. Both SCEs are empty at the outset of execution.

The output in Table 4 represents the first two cycles in the polling sequence, and verifies that the polling logic is modeled accurately. During the first cycle through the two SCEs, the CCE senses the empty buffer and fills the in.q of each. Upon its return to SCE#1, the CCE finds 9 completed commands and one remaining in the in.q. (Note that, within the model, commands remain in the in.q until processing is completed.) The 9 commands are accepted and replaced by the CCE.

As shown the CCE leaves SCE 1 with a full in.q, specifically 10 commands, and an empty out.q. Similarly, the SCE arrives to find 8 completed commands in the out.q of SCE 2. Upon completion of the poll, the CCE leaves SCE 2 with a full in.q and an empty out.q.

```
Next SCE with work to do is SCE #1 at time=  100.0
CS polls SCE #1 at time= 100.0, with  0 comnds in out.q and  0 in in.q
CS poll SCE #1, all out.commands received, n.out.q(SCE#1)= 0
CS poll SCE #1, all in.commands transmitted, n.in.q(SCE#1)=10
Poll for SCE #1 completed at time=  102.6

Next SCE with work to do is SCE #2 at time=  202.6
CS polls SCE #2 at time= 202.6, with  0 comnds in out.q and  0 in in.q
CS poll SCE #2, all out.commands received, n.out.q(SCE#2)= 0
CS poll SCE #2, all in.commands transmitted, n.in.q(SCE#2)=10
Poll for SCE #2 completed at time=  205.2

Next SCE with work to do is SCE #1 at time=  305.2
CS polls SCE #1 at time= 305.2, with  9 comnds in out.q and  1 in in.q
CS poll SCE #1, all out.commands received, n.out.q(SCE#1)= 0
CS poll SCE #1, all in.commands transmitted, n.in.q(SCE#1)=10
Poll for SCE #1 completed at time=  314.7

Next SCE with work to do is SCE #2 at time=  414.7
CS polls SCE #2 at time= 414.7, with  8 comnds in out.q and  2 in in.q
CS poll SCE #2, all out.commands received, n.out.q(SCE#2)= 0
CS poll SCE #2, all in.commands transmitted, n.in.q(SCE#2)=10
Poll for SCE #2 completed at time=  423.1
```

Table 4: TWACS SCE-CCE Model Debug trace Output

4. REVIEW OF THEORY

4.1 Introduction

Computer simulations may be classified as either terminating or steady state simulations. This classification typically results from the underlying system being modeled. The two types of simulations will be described along with appropriate statistical analyses for each. In the final section, a method for addressing problems related to the presentation of multiple confidence intervals will be described.

4.2 Statistical Analysis for Terminating Simulations

The nature of systems such as the TWACS SCE-CCE network dictate that the resulting model be classified as a terminating simulation. By definition, terminating simulations begin at time 0 with initial conditions I, and continue until a predetermined event E occurs at time T_E. All performance measures of interest are therefore gathered over the interval $(0, T_E)$. It is important to note that performance measures for terminating simulations are dependent upon the initial conditions, thus these conditions must be chosen to reflect those encountered by the actual system.

The method used to construct a confidence interval for a terminating simulation is known as the fixed sample size procedure, Law and Kelton [4]. This method requires that a fixed number, say n, independent and identically distributed (IID) estimates of the performance measure under investigation be obtained. This is accomplished by performing n independent replications of the simulation each beginning with the same initial conditions and terminating with the same event E. Independent replications are easily achieved by evaluating each over different portions of the random number streams.

Each estimate of the performance measure, X_j (j=1,...,n), is therefore independent and identically distributed and standard statistical analysis can be employed to construct the confidence interval for $\mu = E(X)$. Based on the assumption that the estimates X_j's are normally distributed, the approximate $100(1-\alpha)$ % confidence interval can be constructed as follows:

$$\overline{X}(n) \pm t_{n-1,1-\frac{\alpha}{2}} \sqrt{\frac{s^2(n)}{n}}$$

where:

$$\overline{X}(n) = \frac{\sum_{j=1}^{n} X_j}{n} \qquad \text{and} \qquad s^2(n) = \frac{\sum_{j=1}^{n} \left[X_j - \overline{X}(n)\right]^2}{n-1}$$

The term *approximate* confidence interval is used since its construction is based on the assumption of normally distributed data points. In practice, however, it is rare that this assumption is strictly true, the deviation from normalcy being dependent on the underlying model and the sample size n. Due to the existence of central limit theorems, it is expected that if the X_j's are the average of a large number of data points, the departure from normalcy should be acceptably low.

Literature suggests that simulation models which generate asymmetrically distributed data points produce the greatest departure from normalcy, see Johnson [5] and Law [6]. However the experience of Law and Kelton [4], for instance, suggests that many real-world simulation models produce output data upon for which the assumption of normalcy is acceptable provided that the X_j's are the product of a large number of observations.

4.3 Statistical Analysis for Non-Terminating Simulations

A non-terminating simulation, by definition, is one with no natural ending point. In other words, there is no naturally occurring event that signals the end of the process. The Nertec model is one such simulation since the actual system operates 24 hours a day on a repeating weekly cycle. For such systems, it is important to evaluate performance measures once the system has reached steady state.

Steady state may be defined statistically as follows: Given a set of initial conditions I, a time index i, a real number y, and an output process Y_i, if $F_i(y|I) = P(Y_i \leq y|I) \rightarrow F(y)$ as $i \rightarrow \infty$ for all y and any initial conditions I, then F(y) is called the steady state distribution of the output process Y_1, Y_2, ... A process which has achieved *steady state* is governed by a steady state distribution. Note that the steady state distribution is independent of the initial conditions.

$F_i(y|I)$, on the other hand, is known as the *transient distribution* of the output process at time i. The transient distribution changes with the index i, and is dependent on the initial conditions I. A system governed by a transient distribution is said to be in a transient state. In this state, the output process Y_1, Y_2, ..., remains dependent on the initial conditions.

All non-terminating simulations begin in the transient state. Thus the analysis methods presented in the previous section cannot be applied here since observations obtained in the initial, transient state, are not accurate estimates of the steady state behavior. In other words, suppose we wish to estimate a steady state mean $E(Y)=\nu$, in the transient state $E(Y_i)\neq\nu$, while $E(Y_i)=\nu$ only once the system has reached steady state. This stumbling block is known as the *problem of the initial transient.*

The method commonly employed for analysis of steady state parameters for a non-terminating simulation is known as the replication/deletion approach. In order to measure only the steady state behavior of the system, the model is allowed a *warm-up* period of length ℓ to reach an approximate steady state. In other words, at each replication, the system is allowed to run for a predetermined amount of time and data gathered previous to ℓ is discarded. Only those observations obtained after the warm up period has expired are included to determine the estimate.

In a strict statistical sense, one can only reach steady state in the limit as $i \rightarrow \infty$. In practice, however, steady state can be said to begin after the system has operated for a reasonable period. The fundamental problem is determining how long the system must warm up to be able to consider it to have achieved steady state, i.e. that $E(Y_i) \approx \nu$.

A simple, graphical, technique to determine an appropriate period warm up l is proposed by Welch [7] and [8]. The technique involves plotting moving averages, themselves averaged over a fixed number of independent replications. The result is a transient mean curve identical to that of the original process, but with relatively low variability. Since the curve must converge to the steady state mean, the system can be considered to have reached steady state once the curve flattens out. The period preceding this point can be considered the transient state and employed as the warm up period l. Simulation can then be repeated, using only those observations after l to estimate the steady state ν.

Once an appropriate warm-up period has been determined, the method to construct estimates and confidence intervals is essentially the same as that for terminating simulations except that only those observations occurring after the warm-up period are used to construct estimates. Using a discrete time model simulated over the interval (o,m) as an example, the method is as follows. Given a total of n replications and a warm-up period l, the estimate at the j^{th} replication is determined as follows:

$$X_j = \frac{\sum_{i=l+1}^{m} Y_{ji}}{m-l} \quad for\ j = 1,2,\ldots,n$$

Each X_j is obtained from observations occurring when the system is assumed to be at steady state since those observations occurring during the warm-up period are not included. Since each replication is performed over a different section of the random number streams, the X_j's are IID random variables with $E(X_j) \approx \nu$. Therefore $\overline{X}(n)$ is an approximately unbiased estimator of ν, and an approximate 100(1-α) confidence interval is determined as:

$$\overline{X}(n) \pm t_{n-1,\,1-\alpha/2}\sqrt{\frac{S^2(n)}{n}}$$

Other approaches for constructing estimates and confidence intervals for steady state distributions can be found in Law and Kelton [4]. Included is a review of fixed sample size procedures such as the batch means and regenerative methods, as well as a survey of sequential procedures.

4.4 Problems with Multiple Comparisons

In most situations, more than one measure of performance is of interest to the experimenter. Although multiple performance measures can be easily determined simultaneously from the same set of replications, the presentation of multiple confidence intervals for such measures results in an intuitive, though important consequence.

Given as set of k 100(1-α)% confidence intervals, I_s, corresponding to a set of performance measures μ_s, the probability that all I_s cover their corresponding performance measure is given as follows:

$$P(\mu_s \in I_s) \textit{ for all } s = 1,2,\ldots,k \geq 1 - \sum_{s=1}^{k} \alpha_s$$

The result is known as the Bonferroni inequality. Intuitively, depending on the confidence of the intervals I_s and the number of performance measures estimated, a serious erosion in the overall confidence of the results may occur.

The problem may be easily corrected according to a method presented by Law and Kelton [4]: Simply choose the individual levels α_s such that the overall desired level of confidence α is given as follows:

$$\sum_{s=1}^{k} \alpha_s = \alpha$$

5. SENSITIVITY ANALYSIS

5.1 Introduction

In order to gauge the response of the two systems to variation of there input parameters, sensitivity analysis of the Nertec and TWACS SCE-CCE models was performed. Parameters that were deemed to be controllable, such as the number of SCEs in a TWACS implementation for instance, were manipulated and the model response plotted for both models.

5.2 Nertec

Specific performance measures of the system were evaluated, through simulation, for varying cycle lengths. Specifically, the mean connect time and the blocking probability were estimated. As described earlier, the Nertec system typically operates on a seven day cycle, with 1/7 of the Telereaders scheduled to call-in during each day. The cycle length was varied, assuming an eight hour primary call-in period, for a 28 modem system and a 40 modem system. Results for the mean connect time and blocking probability are plotted in Figure 12 and Figure 13 respectively.

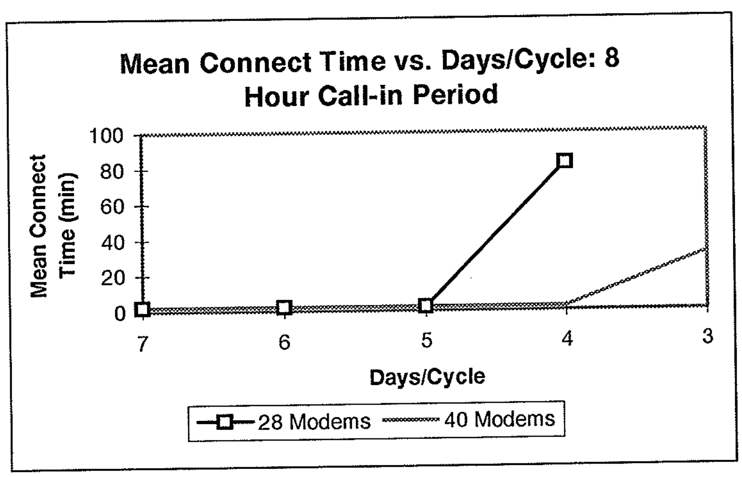

Figure 12: Mean Connect Time vs. Days/Cycle: 8 Hour Call-in Period

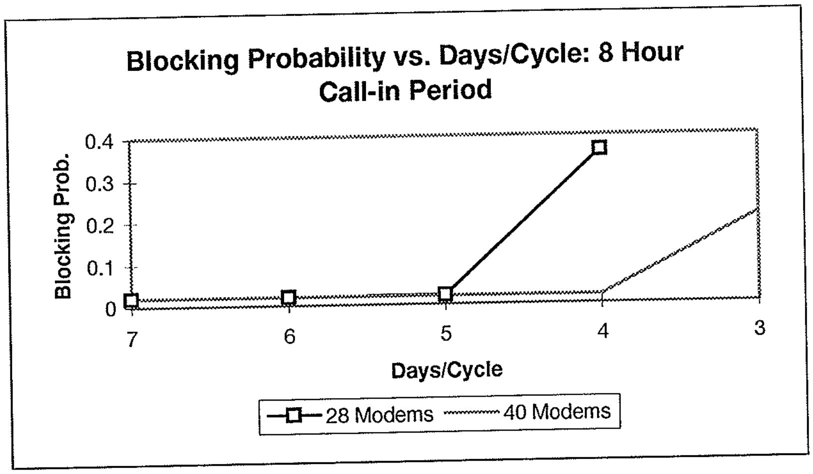

Figure 13: Blocking Probability vs. Days/Cycle: 8 Hour Call-in Period

The mean connect time for both the 28 and 40 modem models remained acceptably low, at just over 2 minutes, for cycle lengths of greater than five days. The 40 modem system can manage the load down to a cycle length of four days while the 28 modem system is overloaded for cycle lengths of 5 days of less. Further reduction in cycle lengths, in the respective configurations, lead to a sudden and marked reduction in performance. By changing the cycle length of the 40 modem model from four days to three, the mean connect time increases from approximately 2 minutes to roughly half an hour. Obviously, this configuration is unable to cope with the increased load. The 28 modem model experiences and even more pronounced collapse when the cycle length is reduced from five to four days.

The obvious cause of the collapse, in each configuration, is the increase in traffic intensity brought upon by reducing the cycle length. The traffic intensity is defined as the product of the service time and the arrival rate, divided by the number of modems. Once this quantity surpasses unity by enough of a margin, the system is unable to cope. Even though there are no further arrivals for sixteen hours after the primary call-in period during which the system will attempt to catch up according to the retrial schedule, it is initially so overloaded that the service level becomes unacceptable.

The blocking probabilities for these two models follow the same pattern. This is the expected result since a larger blocking probability increases the likelihood of multiple attempts before a successful connection. Note that the blocking probability is constant at approximately 2% even before the system becomes overloaded. This reflects the modeled probability of a customer picking up the phone during transmission, set arbitrarily at 2%. Thus most of the mean message delay observed in these experiments can be attributed to the potentiality of customer interruptions, rather than overload at the central server. It is interesting to note that a 2% chance of customer interruption can lead to a mean delay of over two minutes.

The same simulation experiments were repeated assuming that the system operated on a 16 hour primary call-in period. In other words, Telereader calls scheduled for each day of the cycle are spread out over a 16 hour period rather than the standard eight. As shown in Figure 14 and Figure 15, both the mean connect time and the blocking probability follow the same pattern as in the previous experiment. However, both the 28 and 40 modem models are now able to handle shorter cycle periods, as low as two days in the 40 modem case.

It was attempted to model a 40 modem Nertec system with a cycle length of one day, again assuming a 16 hour primary call-in period. As such the Nertec system would approach Manitoba Hydro's requirement that a given system collect and transmit three reads per day. However, configured as such, the model became so overloaded that it was not possible to obtain results. Thus, even though the cycle length can be reduced from the intended seven days, according to experimental observations, the Nertec system will not be able to accommodate the relevant stipulation of the Manitoba Hydro AMR service requirements.

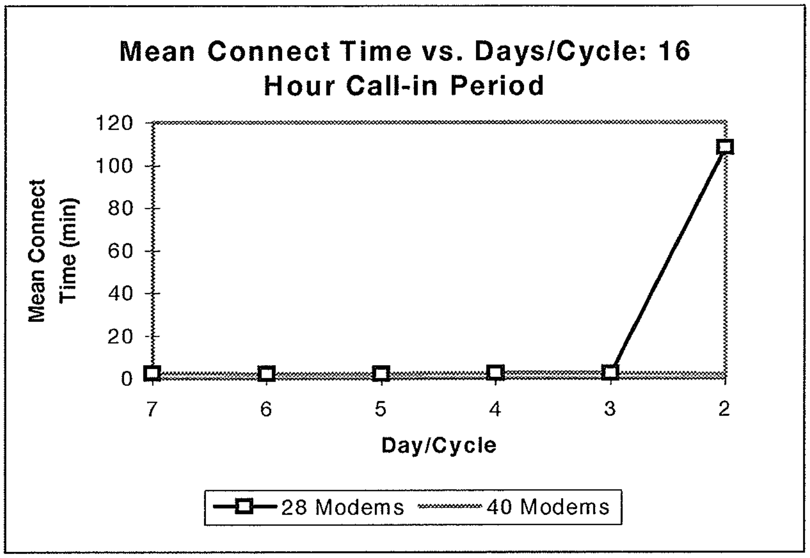

Figure 14: Mean Connect Time vs. Days/Cycle: 16 Hour Call-in Period

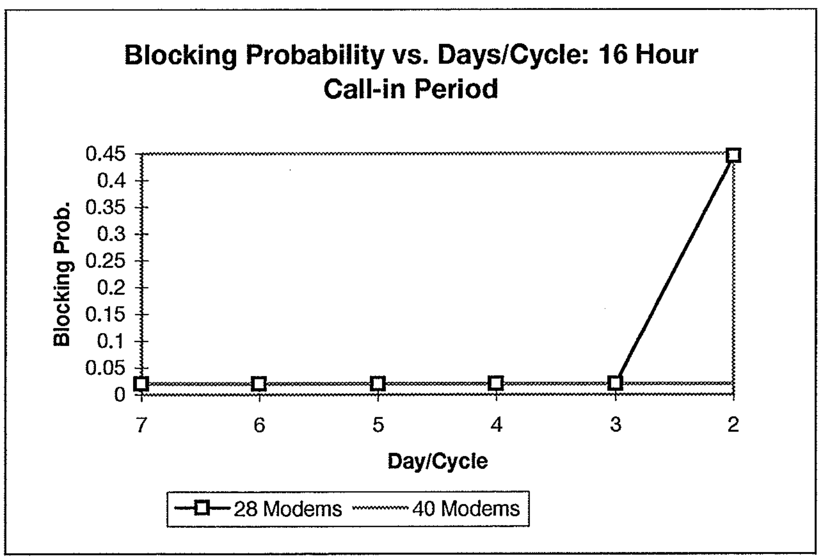

Figure 15: Blocking Probability vs. Days/Cycle: 16 Hour Call-in Period

5.3 TWACS SCE-CCE Network

The TWACS SCE-CCE network was modeled as described with the measures of interest being the mean command delay, for high and low priority commands, and the mean completion time. Specific model parameters were manipulated in order to gauge the sensitivity of the model. Unless otherwise specified, the system is assumed to have 25 SCEs, a switchover time of 0.1 seconds, and a high priority arrival rate of 1500 per hour.

Generally, performance measure are affected by a specific quantity, the CCE cycle time, i.e. the time required for the CCE to complete one circuit through all connected SCEs. If the cycle time for the CCE to poll all SCEs is rapid enough, it will essentially cycle through all SCEs stripping and replacing one or two high priority commands at each. Since the cycle time is brief, few low priority commands are able to complete processing between successive polls and thus few are ready for transmission at a given poll. This leads to the expected results for a multiple priority system, namely long mean delays for low priority commands relative to that of high priority commands.

However, once the CCE cycle time increases past a certain limit, observed performance becomes less intuitive. More low priority commands are able to complete processing between polls and thus more are ready for transmission at a given poll . A cumulative effect then develops: more low priority commands ready for processing further slows down the cycle time allowing for the completion of yet more low priority commands. The end result is that, when polled, each SCE has completed processing all or most of its commands and must dump and refill its entire buffer. Thus, the main component of message delay becomes the interval between successive polls. Queue position is now of little importance and the advantage of high priority commands over low priority commands is greatly diminished. The result is a negligible difference in mean message delay between the two priority types.

The first parameter to be manipulated was the number of SCEs comprising the system. As shown if Figure 16, the expected gap in mean delays diminishes to a negligible difference with 26 SCEs or greater. Below 26 SCEs, the CCE is able to poll all SCEs at a rate such that at each SCE it will generally find only high priority commands, and the odd low priority command, ready for transmission. This leads to the high delay for low priority commands, while high priority commands have a low delay since they obtain service shortly after completing processing. Past 26 SCEs, the time required to poll all SCEs is long enough that the CCE finds most messages ready for transmission at each SCE. Adding an additional SCE increases the time required to cycle through all SCEs by the approximate time required to drain and fill an SCE buffer, the mean delay times are affected accordingly.

Completion time reaches a minimum with 26 SCEs and remains essentially constant thereafter. Past this point, the CCE generally finds each SCE buffer full at each poll. Because the CCE polls the SCEs in cyclical order, essentially a fixed total number of SCEs must be polled to complete a fixed number of commands. Thus the completion time remains essentially constant.

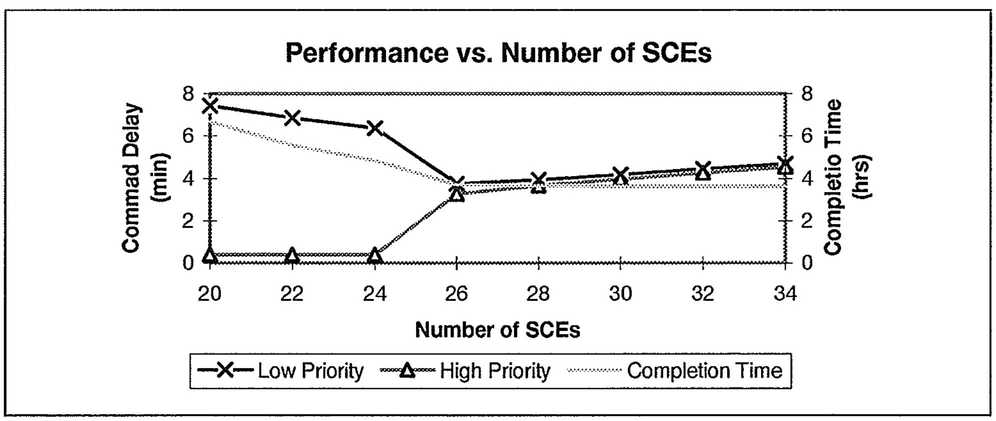

Figure 16: TWACS SCE-CCE Network Performance vs. Number of SCEs

The second parameter to be manipulated was the switchover time. Increasing switchover time produces a similar effect to increasing SCEs since the essential effect is also to increase CCE cycle time by a deterministic amount. As shown in Figure 17, at a switchover time of 0.2 the difference in mean delay for high priority and low priority commands closes. Past 0.2 seconds, the gap between the two curves steadily narrows. Of course, the completion time steadily increases since it is directly affected by switchover time.

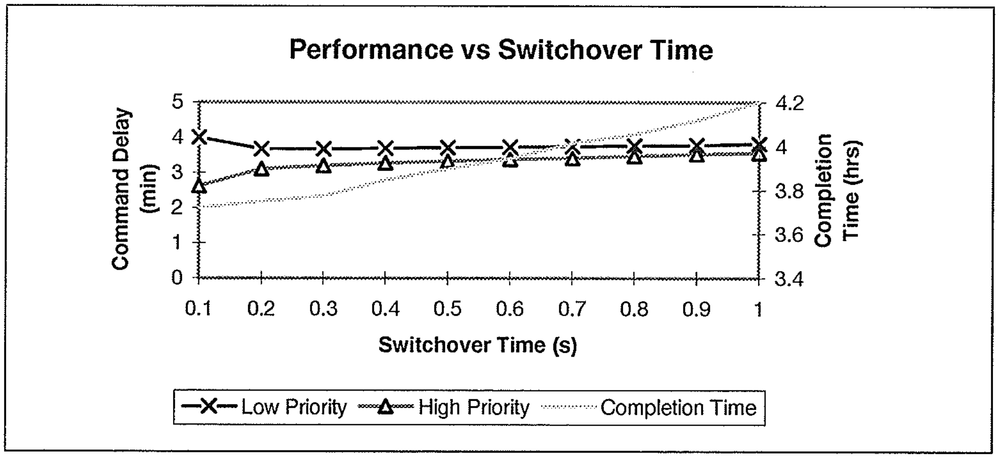

Figure 17: TWACS SCE-CCE Network Performance vs. Switchover Time

Finally, the response of the model to the rate of arrival of high priority commands was examined and the output plotted in Figure 18. As expected, higher rates of high priority command arrivals leads to higher mean low priority delay and completion time. The mean delay of high priority commands, however, falls moderately until the arrival rate reaches 2500. The initial drop results from the fact that more high priority arrivals means more high priority commands, on average, in the SCE buffers. Since the high priority commands are processed first, it becomes less likely for a low priority command to complete processing before the next poll. Since the high priority commands require little transmission time, the polling cycle time reduces leading to the decrease in high priority command delay. Past a rate of 2500 per hour the model begins to become overloaded, as evidenced by the steepening curves of the mean low priority command delay, the completion time and a moderate increase in high priority command delay.

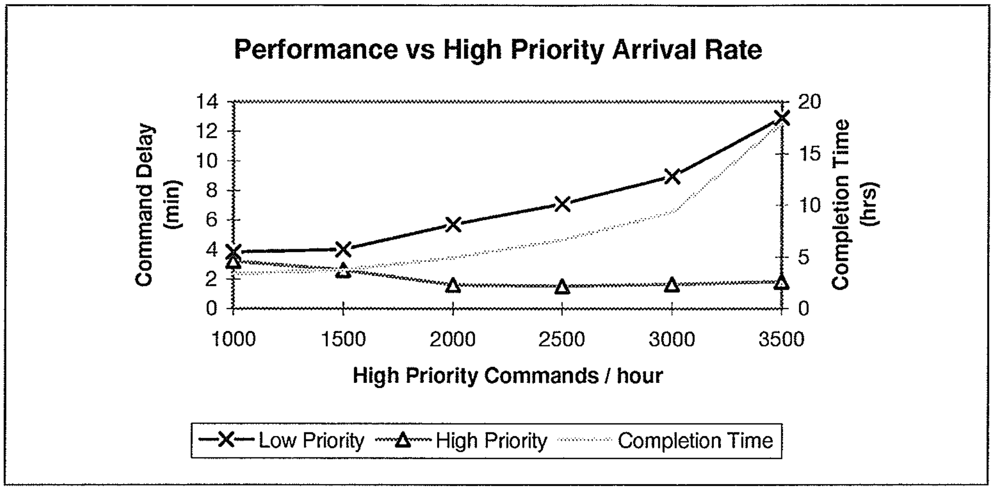

Figure 18: TWACS SCE-CCE Network Performance vs. HP Arrival Rate

6. SIMULATION ANALYSIS OF AMR SYSTEM PERFORMANCE

6.1 Introduction

In this chapter, the performance of the TWACS and Nertec systems, as each would normally be implemented in the City of Winnipeg, will be analyzed via the respective simulation models. This presents immediate problems with the Nertec system due to discrepancies between the manufacturer's intended service parameters and those desired by Manitoba Hydro. For instance, the Nertec System is unable to operate on a one day cycle. Rather, it is the intention of the manufacturer that the system to store 3 reads per day and transmit them once weekly. In the analyses that follow, where the system's intended operation departs from that desired by Manitoba Hydro, the specifications of the former will be modeled.

6.2 Nertec Model

The Nertec system was modeled as described in chapter 2. According to information provided by Nertec and Manitoba Hydro, the following system parameters were specified as model input:

- 8 hour primary call-in period in hours.
- 7 day cycle
- 28 modems at CS
- 24 second service time (fixed)
- 150 000 meters to be read per cycle

A Nertec system, configured as such would collect three reads per day as required by Manitoba Hydro, however reads would be submitted only on a weekly basis. Note also that since only one message priority level is possible with this system, only one is modeled.

The performance measures of interest were the mean connect time, (the mean time for a Telereader to successfully connect to the CS), and the blocking probability, (the proportion of unsuccessful calls). These measures were chosen as they give the clearest indication of system performance under a given load. Mean connect time demonstrates the efficiency of message collection, while blocking probability illustrates the ability of the server to cope with a given load.

Actual output from the model, given the above inputs, is shown in Figure 19. Though not subject to thorough output analysis, additional output data was collected for checking purposes and for insight into the estimates of interest described above.

```
---Simulation of Nertec AMR---
#
Steady state simulation of Nertec AMR with  5 independent replications
and
runup. The measures of interest are the mean message delay and blocking
probability.
#
-------------------------------------------------------------------------

System and model parameters:
        -Mean interarr. time:                        1.34    (exponential)
        -Service time:                              24.00    (deterministic)
        -Call-in period Length                       8.0     hours
        -No. of modems:                              28
        -Expected new calls per day:              21429
        -Number of Days per cycle                     7
        -Total custs. to be served/cycle         150000
        -Number of days simulated:                    5

-------------------------------------------------------------------------

Exogenous variable check:                  Mean              S.D.
        -Mean interarrival time:           1.34              .10
        -Mean service time:                  24                0

Av. new calls per day                             21426
Mean number served per day:                     21426.1
Mean number blocked calls per day:                436.5
Mean number of busy modems:                         6.0
                        ---Final results---
#
R.V              Average                90% CI
-----            -------                ------
Connect time     135.374(sec)     ( 130.324 ,  140.425)
Blocking Prob      .0200                 (    .0195 ,     .0204)
```

Figure 19: Output of Nertec Simulation Model

As shown, the estimated blocking probability was found to be 2%. Recall that, a 2% interruption probability was built in to the model to account for the possibility of customers picking up the phone during transmission. Therefore the estimated blocking probability can be, for the most part, attributed to customer interruption rather than system overload. It is interesting to note that such a low probability of customer pickup can be responsible for most of the relatively large mean connect time.

The mean connect time was found to be 135.374 seconds for the recommended implementation of the Nertec system. The 90% confidence interval was somewhat wider than might be expected considering that 5 repetitions of significant evaluation length were performed. This is due to the large variability in connection time which can vary, according to the retrial schedule, from zero to several days. If necessary, confidence intervals could easily be shortened by increasing the number of replications or extending the evaluation period.

This inherent variability in connect time is also the cause of the high mean connect time. Considering that 98% of calls placed are successful, a like number of initial calls will also be successful and thus have a connect time of zero. Thus all delay may be attributed to the 2% of calls that are unsuccessful and are then delayed according to the retrial schedule.

6.3 TWACS SCE-CCE Model

The TWACS SCE-CCE Network was modeled as described in chapter 2. Unlike the Nertec system, there is no recommended network implementation. In other words, an implementation of a TWACS network may consist of anywhere from one to 81 SCEs, since an SCE may be placed at any or all of the 81 substations in Winnipeg. However, according to observations made in chapter 5, a system with 24 SCEs appeared to provide the best trade off between completion time and high and low command delay. Based on this observation, and information provided by both DCSI and Manitoba Hydro, the following input parameters were specified for the model:

- 24 SCEs in network
- 5% Failed response probability
- 0.1 second switchover time
- High priority arrival rate of 1500 commands/hour
- 150 000 meters to read per cycle

```
---Simulation of the TWACS CCE-SCE Network---
#
Terminating simulation of the TWACS SCE-CCE network, with 10 replications.
System assumed empty at outset of each trial.

Measures investigated:  Mean HP and LP command delay
                        Mean completion time

Model Inputs:
Total Customers/cycle:        150000
Number of SCE's:                24.0
SCE buffer size:                10.0
SCE Switchover time:              .1    (sec)
Command processing time:        18.2    (sec)
Failed Reply Prob:               .050
Mean high prior. iarr. time:    2.40    (sec)

                         ---Final Results---
Exogenous Vars:
        -Mean # Reads lost per Cmnd:       .800         S.D.:       .872
        -High Prior. Iarr.:               2.39 (sec)    S.D.:      2.38 (sec)
        -Mean LP SCE proc. time:         18.20 (sec)    S.D.:       .00 (sec)
        -Mean HP SCE proc. time:          6.98 (sec)    S.D.:       .56 (sec)
#
Performance measures:
Command delay:
Low Priority (AMR):
        -Average= 6.384 min             95%CI=( 6.345 ,  6.423)
High Priority (Load Survey):
        -Average=  .393 min             95%CI=(  .391 ,   .394)
Completion Time:
        -Average: 4.828 hrs             95%CI=( 4.774 ,  4.882)
```

Figure 20: Output of TWACS SCE-CCE Model

As shown, estimated mean command delays were found to be 6.384 and 0.393 minutes for low and high priority commands respectively. Thus high priority commands experienced approximately 1/20 the delay of their low priority counterparts. The model, on average, completed all low priority commands in 4.828 hours, given an high priority arrival rate of 1500 messages per hour.

Confidence intervals were deemed to be acceptably tight. 95% confidence intervals were used in this model so that the overall confidence in presented results remained acceptably high at 85%. Since run time was quite reasonable, further tightening of the presented intervals could be easily achieved by extending the evaluation interval or increasing the number of replications.

With an estimated mean completion time of 4.828 hours, an implementation of the TWACS System-10 could complete three full cycles per day. In order to further validate this proposition, some knowledge of the underlying distribution of the completion time is required. To this end 100 estimates of the completion time were generated and their relative frequencies plotted in the histogram in Figure 21. As shown, most of the probability is collected in the range between 4.65 and 4.95 hours and no estimate exceeded 5.2 hours. Therefore it is expected that the TWACS System-10 could collect three reads per meter per day, and thus meet Manitoba Hydro AMR service requirements.

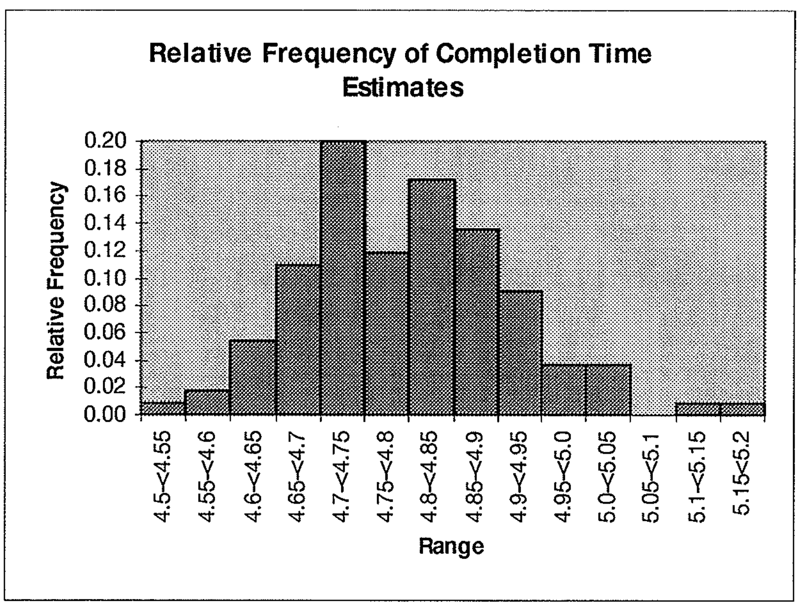

Figure 21:Relative Frequency of TWACS Completion Time Estimates

7. DISCUSSION OF AMR SIMULATION ANALYSIS

It is initially apparent from the descriptions of the three AMR systems that they are each unique systems. Each gather and transmit data in entirely different manners. One system may store data for eventual transmission, another may not. One system may be an inbound system, while the other is outbound. Due to the varied nature of the three systems, a straight comparison of the three, based on simulation analyses, is akin to the proverbial comparison of apples and oranges. Indeed, the measures by which the performance of each system was judged, were themselves dependent on the system in question.

Therefore, rather than attempt to determine which system performed "best", the intention of the Manitoba Hydro study was to present an analysis of the performance of each system, to be judged individually. By contrasting the results of the simulation analysis with the Manitoba Hydro AMR service requirements, it is possible only to determine which systems will perform acceptably from the point of view of Manitoba Hydro.

It is quite apparent that the Nertec System will not meet the needs of Manitoba Hydro. Though the system will provide the necessary three reads per meter per day, the information is intended only to be transmitted once weekly. Though the length of the transmission cycle can be manipulated, results from the chapter 5 suggest that a cycle length of one day would be unrealistic under the expected message load expected for a City of Winnipeg implementation. Thus there would be an erosion in the currency of the information at hand to Manitoba Hydro. Further, the system is unable to handle multiple message priority levels, reducing the ability of the system to be customized to the specific requirements of Manitoba Hydro.

TWACS System-10, on the otherhand, appears to meet all requirements of Manitoba Hydro. According to results of the TWACS SCE-CCE model presented in chapter 6, the TWACS system will be able to collect three reads per day. The system also accommodates multiple message priority levels and can collect meter readings efficiently under the expected load of demand readings.

Note, however, that care should be exercised when choosing the number of SCEs to included in a TWACS System-10 implementation. From the sensitivity analysis in chapter 5, it is apparent that both lower and upper thresholds exist regarding the number of SCEs. While too few SCEs result in higher cycle times, too many may result in a rapid erosion in the advantage of high priority commands over low priority commands.

According to the work done by Stone, the Iris system 2020, will also meet the needs of Manitoba Hydro. The system is able to gather and transmit three reads per day, can handle multiple message priority levels and does so under expected message load. Again, complete details are available in his thesis.

Therefore, one system falls short of required performance while the TWACS and IRIS systems perform to a standard acceptable to Manitoba Hydro. These two remaining systems, must be judged by Hydro on an individual basis. While a final purchase decision will certainly be partly based on factors such as the purchase cost and technical support for each system, the analysis presented here will provide an insight into the performance that can be expected of each should it be implemented.

8. CONCLUSION

As stated, the intention of the overall AMR simulation study for Manitoba Hydro was to determine which of the three systems met the performance requirements of Manitoba Hydro. In addition, through simulation analysis, the strengths and weaknesses of the three systems can be highlighted, and thus judged, on an individual basis. The TWACS System-10 and Nertec AMR systems were therefore modeled here so that their abilities could be judged along side those of the previously evaluated Iris System 2020.

The componentry and logical operation of the TWACS, Nertec, and Iris systems has been presented in order to provide background for the models developed. The two models developed for this portion of the study were introduced and verified according to standard practice. Through sensitivity and performance analysis of the models, the abilities and potential pitfalls of the two systems were highlighted. According to the results of the simulation models, it was determined that the Nertec system would not be able to perform to the standard required by Manitoba Hydro. The TWACS system, on the other hand, was able to meet these requirements, however care should be taken when choosing the number of SCEs to include.

As stated earlier, the intention here was not to determine which system will best suit the needs of Manitoba Hydro, this final judgment is left to those involved with the AMR project at Hydro. While their decision will certainly be based on other factors such as cost and vendor support, it is hoped that the information presented here will provide a useful tool in the decision making process.

9. REFERENCES

1. Stone, R. J.: *Simulation Analysis of Multiple-Priority Polling Systems with Application to the System 2020 Automatic-Meter-Reading Network*, University of Manitoba, Winnipeg, 1994.
2. Russel, E. C.: *Building Simulation Models with Simscript II.5*, CACI, Inc., Los Angeles, 1983.
3. Kiviat, P. J., Villaneuva, R., and Markowitz, H. M.: *Simscript II.5 Programming Language*, CACI, Inc., Los Angeles, 1975.
4. Law, A. M. and Kelton, W. D.: *Simulation and Modeling Analysis,* MacGraw-Hill Book Company, New York, 1990.
5. Johnson, N.J.: Modified t Tests and Confidence Intervals for Asymmetrical Populations, *Journal the American Statistical Association*, 73: 536-544 (1978).
6. Law, A. M.: Statistical Analysis of the Output Data from Terminating Simulations, *Naval Research and Logistics Quarterly*, 27: 131-143 (1980).
7. Welch, P. D.: On the Problem of the Initial Transient in Steady-State Simulation, IBM Watson Research Center, Yorktown Heights, New York, (1981).
8. Welch, P. D.: The statistical Analysis of Simulation Results, *The Computer Performance Modeling Handbook*, S. S. Lavenberg, ed., Academic Press, New York, (1983).

APPENDIX 1: SOURCE CODE FOR SIMULATION MODELS

1. NERTEC MODEL SOURCE CODE

```
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

'' NERTEC SIMULATION MODEL, PERFORMANCE MEASURES EVALUATED:
''     -      MEAN CONNECT TIME
''     -      MEAN BLOCKING PROBABILITY
''MODEL INPUTS LOCATED IN FILE "NERTEC.IN"
''MODEL OUTPUT LOCATED IN FILE "NERTEC.OUT"
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''



Preamble
Normally mode is undefined
Event notices include New.Day
      Every Callback has a ptr.to.cust
Processes include Generator
      Every Service has a ptr.to.cust
Temporary Entities
      Every Customer has a create.time, a call.time, a repnum, a type,
      and a cycnum
Resources include Server

The System has a CB random step variable
      Define CB as an integer variable
The System has a RT random step variable
      Define RT as an integer variable

Define delay.array as a real, 1-dimensional array
Define prob.array as a real, 1-dimensional array

Define .true to mean 1
Define .false to mean 0
Define reps to mean 5
Define arr.stream to mean 2
Define 24hrs to mean 86400

Define ptr.to.cust as an integer variable
Define retrials, and t as real variables
Define end.day as integer variable

Define iarr, serv, iarr.mean and serve.mean as real variables
```

```
Define rep.num, Modems, Runup, num.cust and cyc.days as integer
variables
Define end.time and period as real variables

Define trial.busy,trial.connect and trial.prob as real variables
Define create.time,call.time,repnum, cycnum and type as real variables
Define num.blocks, num.served, num.new as real variable
Define trial.new and trial.served as real variables
Define day.num and trial.blocks as real variable
Define pickup.prob as a real variable
Define connect.time and day.connect as real variables

Accumulate av.busy as the average of N.X.Server
Tally av.connect as the average of connect.time
Tally sim.av.connect as the average of trial.connect
Tally sim.av.prob as the average of trial.prob
Tally sim.av.busy as the average of trial.busy

Tally sim.av.serv as the mean and sim.sd.serv as the std.dev of serv
Tally sim.av.iarr as the mean and sim.sd.iarr as the std.dev of iarr
Tally sim.av.new as the mean of trial.new
Tally sim.av.blocks as the mean of trial.blocks
Tally sim.av.served as the mean of trial.served
End
```

```
Main
Open unit 3 for input, file name is "nertec.in"
Use unit 3 for input
Open unit 12 for output, file name is "nertec.out"
Use unit 12 for output
Call Get.Dat
Reserve delay.array(*) as reps
Reserve prob.array(*) as reps
Call header
For rep.num = 1 to reps do
      Let runup=.true
      Create Every Server(1)
      Let u.server(1) = modems
      Schedule a New.Day in 24hrs units
      Activate a Generator Now
      Let Pickup.prob=0.02
      Start Simulation
      Reset the totals of N.X.Server
      Reset the totals of connect.time
      Destroy every Server
      Let time.v=0
      Let end.time=period
      Let num.blocks=0
      Let num.served=0
      Let num.new=0
      Let day.num=0
Loop
Call report
End

Process Generator
Define start as a real variable
Until Time.v > end.time do
      Let start=time.v
      Wait normal.f(iarr.mean,0.1,arr.stream) units
      Let iarr=time.v-start
      Add 1 to num.new
      Create a customer
      Let create.time(customer)=time.v
      Activate a service giving customer now
Loop
End
```

```
Process Service given customer
Define Customer as an integer variable
Define begin.serv as a real variable
Define retry.time as a real variable
If ((n.x.server = modems)or(Random.f(1) < pickup.prob))
      Add 1 to num.blocks
      Add 1 to repnum(customer)
      If type(customer)=0
            call.time(customer)=CB
            Schedule a Callback giving customer
                  in call.time(customer) units
      else
            if repnum(customer)<=3
                  Schedule a Callback giving customer
                        in call.time(customer) units
            else
              add 1 to cycnum(customer)
              if cycnum(customer)<4
                  let repnum(customer)=0
                  Retry.time=RT
                  call.time(customer)=CB
                  Schedule a Callback giving customer
                        in Retry.time units
               else
                  Let connect.time=time.v-create.time(customer)
                  Destroy this customer
               Always
            Always
      Always
Else
      Add 1 to num.served
      Request 1 server(1)
            Let begin.serv=time.v
            Let connect.time=time.v-create.time(customer)
            Work Serve.mean units
            Let serv=time.v-begin.serv
      Relinquish 1 server(1)
      Destroy this customer
Always
End
```

```
Event Callback given customer
Define customer as an integer variable
Let type(customer)=1
Activate a Service giving customer now
End
```

```
Event New.Day
Add 1 to day.num
If runup=.false
      If day.num < end.day
            Schedule a New.Day in 24hrs units
            Let end.time=time.v+period
            Activate a Generator now
      Else
            Call Snap
      Always
Else
      Reset the totals of connect.time
      Reset the totals of N.X.Server
      Let num.blocks=0
      Let num.served=0
      Let num.new=0
      If day.num=1
            Let day.num=0
            Let runup=.false
      Always
      Schedule a New.Day in 24hrs units
      Let end.time=time.v+period
      Activate a Generator now
Always
End
```

```
Routine Snap
      let trial.busy=av.busy
      Let trial.new=num.new/end.day
      Let trial.served=num.served/end.day
      Let trial.blocks=num.blocks/end.day
      Let trial.connect=av.connect
      Let trial.prob=num.blocks/(num.blocks+num.served)
      Let delay.array(rep.num)=trial.connect
      Let prob.array(rep.num)=trial.prob
End


Routine Get.Dat
Let runup=.true
Let end.day=5
Read period
Let period=period*3600
Let end.time=period
Read serve.mean, modems
Read cyc.days
Read num.cust
Let iarr.mean=(cyc.days*period)/num.cust
Read CB
Read RT
End
```

```
Routine Header
Use unit 12 for output
Skip 2 lines
Print 7 lines with reps thus
            ---Simulation of Nertec AMR---
#
Steady state simulation of Nertec AMR with ** independent replications
and
runup. The measures of interest are the mean message delay and blocking
probability.
#
----------------------------------------------------------------------
-
Skip 1 line
Print 10 lines with iarr.mean, serve.mean, period/3600, modems,
period/iarr.mean, cyc.days, num.cust and end.day thus
System and model parameters:
      -Mean interarr. time:                 *.** s      (normal)
      -Mean std.dev for iarrs.              0.1
      -Service time:                        *.**  (deterministic)
      -Call-in period Length                 *.*  hours
      -No. of modems:                        ***
      -Expected new calls per day:         *******
      -Number of Days per cycle               **
      -Total custs. to be served/cycle       *******
      -Number of days simulated:             ***
skip 1 line
End


Routine Report
Define numerc, numerp, widthc, and widthp as real variables
Define i as an integer variable

Print 1 line thus
----------------------------------------------------------------------
-
skip 1 lines
Print 3 lines with sim.av.iarr, sim.sd.iarr, sim.av.serv
and sim.sd.serv thus
Exogenous variable check:            Mean               S.D.
      -Mean interarrival time:       *.**               *.**       (sec)
```

```
      -Mean service time:               **                  **      (sec)
skip 1 line
Print 4 line with sim.av.new,sim.av.served, sim.av.blocks
and  sim.av.busy thus
Av. new calls per day                        ******
Mean number served per day:               *******.*
Mean number blocked calls per day:        *******.*
Mean number of busy modems:                   **.*

''''''''''''''''''''''''''''
''CALCULATE C.I. HALF WIDTH
''''''''''''''''''''''''''''

For i=1 to reps do
      Let numerc=numerc+(sim.av.connect-delay.array(i))**2
      Let numerp=numerp+(sim.av.prob-prob.array(i))**2
Loop
Let numerc=numerc/(reps-1)
Let numerp=numerp/(reps-1)
Let widthc=2.132*((numerc/reps)**(1/2))
Let widthp=2.132*((numerp/reps)**(1/2))

Print 6 lines with  sim.av.connect, sim.av.connect-widthc,
sim.av.connect+widthc, sim.av.prob, sim.av.prob-widthp and
sim.av.prob+widthp thus
                  ---Final results---
#
R.V         Average                  90% CI
-----       -------                  ------
Connect time      ****.*** (sec)         (****.*** , ****.***)
Blocking Prob       *.****           (  *.**** ,    *.****)



end
```

2. TWACS SCE-CCE NETWORK MODEL SOURCE CODE

```
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
''TWACS SCE-CCE SIMULATION MODEL, PERFORMANCE MEASURES EVALUATED:
''     -      MEAN HP AND LP COMMAND DELAY
''     -      MEAN CYCLE COMPLETION TIME
''MODEL INPUTS LOCATED IN FILE "TWACS.IN"
''MODEL OUTPUT LOCATED IN FILE "TWACS.OUT"
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''


Preamble
Normally mode is undefined
Processes include Generator
      Every Seize.SCE has a ptr.to.SCE
      Every Poll.SCE has a ptr.to.SCE
      Every command.exec has a ptr.to.SCE
Event notices
      Every Generate.high has a mrk.time
Permanent entities
      Every SCE has a state, owns an arrive.queue, owns an in.queue,
      and owns an out.queue
Temporary Entities
      Every command has a go.time, a priority and
      may belong to the in.queue and may belong to the out.queue
      and may belong to the arrive.queue

Define arrive.queue as a FIFO set ranked by high priority
Define in.queue as a FIFO set ranked by high priority
Define out.queue as a FIFO set ranked by high priority

Define low.array as a real, 1-dimensional array
Define high.array as a real, 1-dimensional array
Define finish.array as a real, 1-dimensional array

Define state and priority as integer variables
Define ptr.to.SCE and ptr.to.command as integer variables
Define mrk.time, go.time, reply.high and reply.low as real variables
Define day.time,trial.high and trial.low as real variable
Define exog.proc, exog.hpproc and exog.high as real variables
Define initialize.SCEs, Num.SCE and Num.cust as integer variables
Define high.iarr, error.prob and poll.time as real variables
```

```
Define trial.buff, mark.buff and buff.time as real variables
Define mean.lost as real variables

Define high.stream to mean 4
Define retrials to mean 10
Define buffer.size to mean 10
Define command.proc to mean 18.2
Define send.time to mean .26
Define return.time.low to mean .79
Define return.time.high to mean .20
Define .true to mean 1
Define .false to mean 0
Define .done to mean 2
Define .busy to mean 1
Define .idle to mean 0

Tally av.low as the average of reply.low
Tally sim.av.low as the average of trial.low
Tally av.high as the average of reply.high
Tally sim.av.high as the average of trial.high
Tally av.exog.proc as the mean and sd.exog.proc as the std.dev of
      exog.proc
Tally av.exog.hpproc as the mean and sd.exog.hpproc as the std.dev of
      exog.hpproc
Tally av.exog.high as the average and sd.exog.high as the std.dev of
      exog.high
Tally sim.av.time as the average of day.time
Tally sim.mean.lost as the mean and sim.sd.lost as the std.dev of
mean.lost
Tally av.buff as the mean of buff.time
Tally sim.av.buff as the mean of trial.buff
End
```

```
Main
Define i as an integer variable
Open unit 3 for input, file name is "twacs.in"
Use unit 3 for input
Open unit 12 for output, file name is "twacs.out"
Use unit 12 for output
Reserve low.array(*) as retrials
Reserve high.array(*) as retrials
Reserve finish.array(*) as retrials
Call get.dat
For i=1 to retrials do
      Create every SCE(num.SCE)
      Activate a Generator now
      Schedule a Generate.high in exponential.f(high.iarr,high.stream)
units
      Start Simulation
      Let trial.low=av.low/60
      Let trial.high=av.high/60
      Let low.array(i)=av.low/60
      Let high.array(i)=av.high/60
      Let day.time=time.v/3600
      Let finish.array(i)=time.v/3600
      Let trial.buff=av.buff
      Reset the totals of buff.time
      Let time.v=0
      Let mark.buff=time.v
      Reset the totals of reply.low
      Reset the totals of reply.high
      Destroy Every SCE

Loop
Call Report
End
```

```
Process Generator
Define i and command as an integer variable
Define index as an integer variable
For every SCE do
      For i=1 to int.f(num.cust/(num.sce*16)) do
            Create a command
            Let priority(command)=1
            File the command in the arrive.queue(SCE)
      Loop
Loop
Let index=1
Activate a Poll.SCE giving index now
End


Event Generate.high given mark.high
Define mark.high as a real variable
Define index as an integer variable
Let exog.high=time.v-mark.high
Let mark.high=time.v
Let index=randi.f(1,num.sce,8)
Create a Command
Let Priority(Command)=2
File the Command in the Arrive.queue(index)
Schedule a Generate.High giving mark.high
      in exponential.f(high.iarr, high.stream) units
End
```

```
Process Seize.SCE given index
Define index, command, and i as integer variables
Define snap.in.buff and snap.out.buff as integer variables
Let mark.buff=time.v
Let snap.in.buff=n.in.queue(index)
Let snap.out.buff=n.out.queue(index)
For i=1 to snap.out.buff do
      Remove the first command from the out.queue(index)
      If priority(command)=1
            Work return.time.low units
      Else
            Work return.time.high units
      Always
      If priority(command)=1
            Let reply.low = time.v - go.time(command)
      Else
            Let reply.high = time.v - go.time(command)
      Always
      Destroy the command
Loop
For i=1 to (buffer.size-snap.in.buff) do
      If n.arrive.queue(index) > 0
            Remove the first command from the arrive.queue(index)
            Let go.time(command)=time.v
            Work send.time units
            File the Command in the in.queue(index)
      Else
            If n.in.queue(index)=0
                  Let state(index)=.done
            Always
      Always
Loop
If state(index) NE .done
      Let buff.time=time.v-mark.buff
Always
If state(index)=.idle
      Activate a command.exec given index now
Always
If index < num.SCE
      Let index=index+1
Else
      Let index=1
```

```
Always
Activate a poll.SCE giving index now
End
```

```
Process Poll.SCE given index
Define index and test as an integer variable
For every SCE do
      If state(SCE) ne .done
            test=.true
      Always
Loop
If test=.true
Wait Poll.time units
      Until n.out.queue(index) > 0 or n.in.queue(index) < buffer.size do
            If index < num.SCE
                  Let index=index+1
            Else
                  Let index=1
            Always
            Wait Poll.time units
      Loop
      Activate a Seize.SCE giving index now
Else
      Cancel the Generate.High
      Destroy the Generate.High
Always
End
```

```
Process command.exec given index
Define i, index and command as integer variables
Define temp and mark.serv as a real variable
Let state(index)=.busy
While n.in.queue(index) > 0 do
      If priority(f.in.queue(index)) =1
            Let mark.serv=time.v
            Work command.proc units
            Let exog.proc=time.v-mark.serv
            Let temp=0
            For i = 1 to 16 do
                  If random.f(1) < error.prob
                        add 1 to temp
                        Work (uniform.f(6,8,2)) units
                  always
            Loop
            Let mean.lost=temp
            Remove the first command from the in.queue(index)
            File the command in the out.queue(index)
      Else
            Let mark.serv=time.v
            Work uniform.f(6,8,2) units
            Let exog.hpproc=time.v-mark.serv
            Remove the first command from the in.queue(index)
            File the command in the out.queue(index)
      Always
Loop
Let state(index)=.idle
End


Routine get.dat
Define hpmessg.per.hour as an real variable
Read num.SCE, Num.cust, Poll.time, hpmessg.per.hour, error.prob
Let high.iarr=3600/hpmessg.per.hour
end


Routine Report
Define i as an integer variable
Define numerl, numerh, numerf, samp.stdl, samp.stdh, samp.stdf,
widthl, widthh and widthf as real variables
```

```
Print 4 lines with retrials thus
            ---Simulation of the TWACS CCE-SCE Network---
#
Terminating simulation of the TWACS SCE-CCE network, with **
replications.
System assumed empty at outset of each trial.
skip 1 line
Print 2 lines thus
Measures investigated:  Mean HP and LP command delay
                  Mean completion time
Skip 1 line

Print 8 lines with Num.cust, Num.SCE, buffer.size, poll.time, and
command.proc, error.prob, high.iarr thus
Model Inputs:
Total Customers/cycle:        ******
Number of SCE's:        **.*
SCE buffer size:        **.*
SCE Switchover time:          **.*  (sec)
Command processing time:      **.*        (sec)
Failed Reply Prob:              .***
Mean high prior. iarr. time:   ***.**      (sec)
Skip 1 line

''''''''''''''''''''''''
''CALC. C.I. HALF LENGTH
''''''''''''''''''''''''

      For i=1 to retrials do
            numerl=numerl+(sim.av.low-low.array(i))**2
            numerh=numerh+(sim.av.high-high.array(i))**2
            numerf=numerf+(sim.av.time-finish.array(i))**2
      loop
      samp.stdl=(numerl/(retrials-1))**(1/2)
      samp.stdh=(numerh/(retrials-1))**(1/2)
      samp.stdf=(numerf/(retrials-1))**(1/2)
      widthl=2.262*(samp.stdl/(retrials**(1/2)))
      widthh=2.262*(samp.stdh/(retrials**(1/2)))
      widthf=2.262*(samp.stdf/(retrials**(1/2)))


Print 15 lines with sim.mean.lost, sim.sd.lost,
```

```
av.exog.high, sd.exog.high, av.exog.proc, sd.exog.proc,
av.exog.hpproc, sd.exog.hpproc,
sim.av.low,sim.av.low-widthl, sim.av.low+widthl,
sim.av.high,sim.av.high-widthh, sim.av.high+widthh,
sim.av.time, sim.av.time-widthf and sim.av.time+widthf  thus
                  ---Final Results---
Exogenous Vars:
-Mean # Reads lost per Cmnd:    *.***           S.D.:   *.***
-High Prior. Iarr.:           ***.** (sec)      S.D.:  ***.** (sec)
-Mean LP SCE proc. time:       **.** (sec)      S.D.:  ***.** (sec)
-Mean HP SCE proc. time:       **.** (sec)      S.D.:  ***.** (sec)
#
Performance measures:
Command delay:
Low Priority (AMR):
      -Average=**.*** min           95%CI=(**.*** , **.***)
High Priority (Load Survey):
      -Average=**.*** min           95%CI=(**.*** , **.***)
Completion Time:
      -Average:**.*** hrs           95%CI=(**.*** , **.***)

End
```