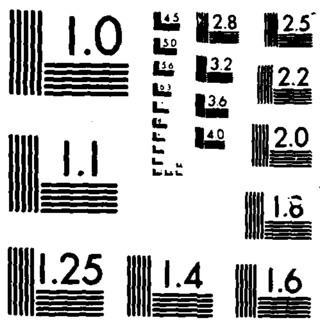
AD-A163 748 A SYSTEM DYNAMICS MODEL FOR COMMUNICATIONS NETWORKS(U) 1/1
ROYAL SIGNALS AND RADAR ESTABLISHMENT MALVERN (ENGLAND)
A J AWCOCK ET AL. SEP 85 RSRE-MEMO-3866 DRIC-BR-97971

UNCLASSIFIED F/G 17/2 NL

END
FILMED
DTIC




MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

AD-A163 748

RSRE

MEMORANDUM No. 3866

# ROYAL SIGNALS & RADAR ESTABLISHMENT

## A SYSTEM DYNAMICS MODEL FOR COMMUNICATIONS NETWORKS

Authors: [illegible] and T E G King

PROCUREMENT EXECUTIVE,
MINISTRY OF DEFENCE,
RSRE MALVERN,
WORCS.



MEMORANDUM No. 3866

ROYAL SIGNALS AND RADAR ESTABLISHMENT

Memorandum 3866

TITLE: A SYSTEM DYNAMICS MODEL FOR COMMUNICATIONS NETWORKS

AUTHORS: M J Awcock and T E G King

DATE: September 1985

## SUMMARY

An abstract model of a communications network in system dynamics terminology is developed and an implementation of this model by a Fortran program package developed at RSRE is discussed. The result of this work is a high-level simulation package in which the performance of adaptive routing algorithms and other network controls may be assessed for a network of arbitrary topology.

CONTENTS

## 1. INTRODUCTION

The present Report describes the implementation of some of the ideas proposed in an earlier work [1] which was concerned with the performance evaluation of adaptive controls in communication networks.

The earlier Report concluded that analytical techniques were not sufficiently powerful to deal with the modelling of adaptive controls in networks, whilst a full scale simulation posed serious problems both in its implementation and in the interpretation of the results. The techniques of System Dynamics were proposed for the construction of a high-level network simulation in which the traffic flow through the network was considered to be continuous (as distinct from consisting of individual packets).

In such an approximation only the amount of traffic at each network switch and the traffic rates between switches are considered. This level of approximation is appropriate for the modelling of routing algorithms, but data-link-control techniques such as retransmission time-out cannot be represented explicitly because knowledge of the identities of individual packets is not available. The effects of such controls can be included only in an approximate manner.

The justification for the above approach rests upon the reasonable hypothesis that the general behaviour of network controls does not depend upon the detailed behaviour of the network; for example, the time taken for a routing algorithm to stabilise traffic distribution after a sudden increase in the traffic injected into the network should not change remarkably if there are 8 rather than 9 packets in the input queue for a particular switch at the time the shock is applied. The suppression of unnecessary detail is vital for the assessment of higher-level behaviour: it is hoped that the simulation technique described in this Report provides a way of accomplishing this.

The Report divides conveniently into two parts: the first part provides an abstract description of an idealised network in System Dynamics terminology and the second part describes the implementation of the abstract model by a Fortran program. The result of this work is a high-level simulation package in which the performance of adaptive routing algorithms and other network controls may be assessed for a network of arbitrary topology.

N.B.- In order to avoid unnecessary duplication, the reader is expected to have read Ref.1: this contains explanations of the meanings of various terms used in the present report.

## 2. THE SYSTEM DYNAMICS MODEL

### 2.1 The Level of Approximation

A System Dynamics model is defined by its rates, levels, and the relationships between them. The level of detail appropriate for a network model which is to investigate the performance of a routing algorithm would perhaps differentiate between traffic streams belonging to different source-destination pairs, but not differentiate between individual packets. Any feature of the network (such as retransmission time-out at the data-link-control level) which requires a more detailed description of the network must of course be represented in an approximate manner.

The choice of the time increment DT which governs the duration of each time step in the simulation also introduces an element of approximation. The value of DT must be adjusted until convergence of the simulation results is attained.

### 2.2 The Basic Switch Structure

Figure 1 depicts a simplified influence diagram for a network switch. The switch is assumed to be connected to a host and to

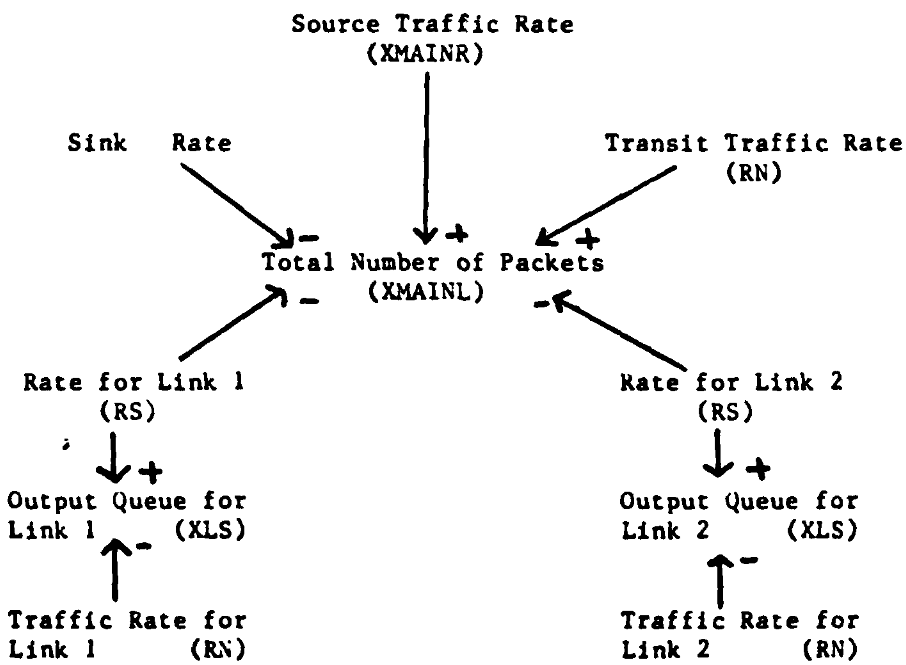



Figure 1 Influence Diagram for the Basic Switch Structure

two network links. The quantities enclosed in brackets denote the Fortran names given to the various rates and levels within the simulation program.

The Source Traffic Rate originates from the host and increases the number of message packets held by the switch, as does the Transit Traffic Rate from neighbouring network switches. The Sink Rate depletes the switch of traffic intended for the host connected to that switch. The remaining packets that are held by the switch are intended for other destinations within the network.

The routing algorithm (whose influence is not shown explicitly in Figure 1 for reasons of clarity) determines the acceptance rates for links 1 and 2 respectively. These rates deplete the Total Number of Packets held by the switch and augment the Output Queue for each link. Finally the Traffic Rate for each link depletes the output queues.

Figure 1 represents a first attempt at constructing a System Dynamics model of a network. It is obviously incomplete; adhering to the System Dynamics methodology, we next enhance the model by incorporating the influences brought about by the routing procedure.

### 2.3 The Routing Algorithm

For the purposes of constructing a more detailed influence diagram, we assume that the routing algorithm is of an adaptive decentralised kind which takes into account the output queue for each link in the network; i.e. the distance matrix depends in some way on the queue sizes.

A standard mechanism for a routing algorithm of this type involves the maintenance of a table by each switch which records the current estimated minimum "distance" from that switch to each destination. The table is constructed by requesting access to the (old) tables of each nearest-neighbour switch and then adding the output queue for the link to the nearest neighbour to the distance from the nearest neighbour to the destination. The minimum distance to the destination is then obtained by taking the minimum over the distances via each nearest-neighbour switch.

A qualitative representation of this process is given in Figure 2. This figure is essentially an expansion of Figure 1, although the Transit Traffic Rate, Sink Rate, and Link Rates have been omitted for clarity: they do not directly affect the functioning of the routing algorithm. Two destinations are assumed; thus a partition of the traffic according to destination is necessary. The top half of the Figure is essentially the same as Figure 1, which must be reproduced twice in order to accommodate the effects of the traffic partition.

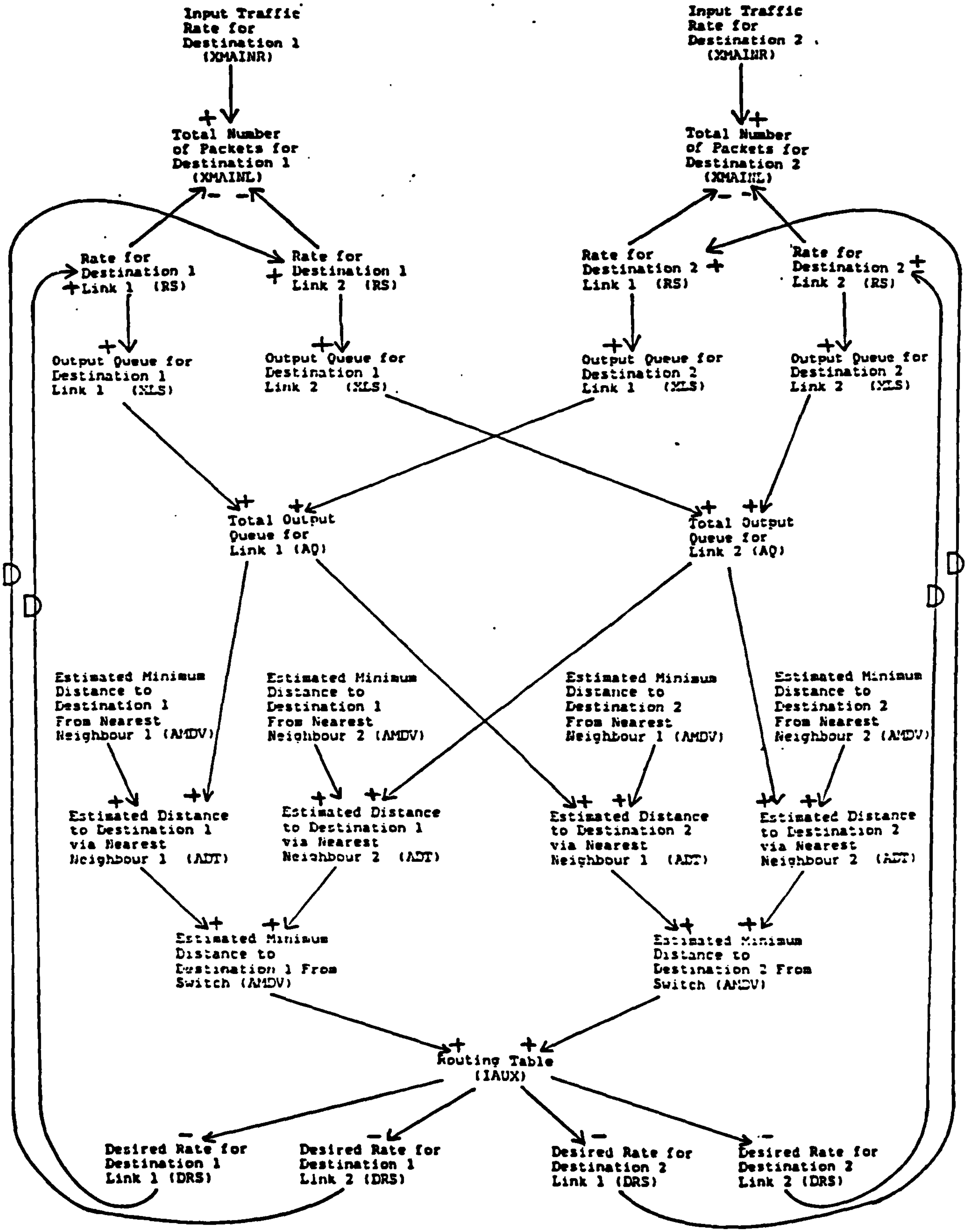



FIGURE 2. INFLUENCE DIAGRAM FOR THE ROUTING ALGORITHM

The rates at which the output queues for the links are increased are governed by the decisions of the routing algorithm, which are in turn influenced by (among other things) the sizes of the link output queues. The completion of the corresponding feedback control loops is portrayed by the bottom half of Figure 2: notice that delays D will in general be introduced between the establishment of a desired rate and the time at which the actual rate is affected.

## 2.4 The System Equations

Figure 3 shows the equations that are implemented in the simulation.

The levels XLS and XMAINL are evaluated by integrating with respect to time the rates that feed or deplete them; such equations are standard in System Dynamics.

The auxiliary variables are concerned with the mechanism of the routing algorithm. The distance tables are calculated from the current queue length and the old minimum distance tables. (Note that a liberty is taken with System Dynamics at this point in that the old value of AMDV is not delayed by an intervening rate: rather, ADT is updated before AMDV, so that the earlier value of AMDV is used in the calculation of ADT.) A bias "BIAS" is added to ADT in order to control the sensitivity of the routing algorithm to changes in the network state.

The rate equations are determined by the control policy. The rate RS feeds the packets held by the switch s into the output queues, depending on the decision of the routing algorithm. The decision is implemented by the variable DRS, which takes the value 0 or 1 (i.e. "off" or "on"). The equation for the link traffic rate RN must take into account the finite link capacity. If the transmission of the complete output queue over the next time increment would not exceed the capacity of the link, then the whole queue is sent: otherwise the link rates RN are scaled down according to the proportion of traffic for each destination that is currently in the link queue. This guarantees that the link capacity is never exceeded and provides a high-level approximation of a data-link-control protocol.

N.B.- The time argument for each rate should be of the form (t,t+DT) or (t-DT,t), as a rate holds its value for a time interval, rather than at a specific time t as in the case of levels. The notation in Figure 3 should be interpreted as follows: RATE(t) means RATE(t,t+DT), etc.

## FIGURE 3: THE EQUATIONS OF THE SYSTEM

AUXILIARIES

Link queue length: $AQ_{s,ns}(t) = \sum_{d} XLS_{s,ns,d}(t)$

Distance table: $ADT_{s,ns,d}(t) = AQ_{s,ns}(t) + AMDV_{ns,d}(t-DT) + BIAS$

Minimum distance table: $AMDV_{s,d}(t) = \min_{ns} ADT_{s,ns,d}(t)$

RATES

Internal rate governed by routing decision:

$$RS_{s,ns,d}(t) = DRS_{s,ns,d}(t-DT) * XMAINL_{s,d}(t) / DT$$

Link Traffic Rate:

$$RN_{s,ns,d}(t) = \begin{cases} XLS_{s,ns,d}(t) / DT & \text{if } AQ_{s,ns}(t) / DT \leq CAP_{s,ns} \\ XLS_{s,ns,d}(t) * CAP_{s,ns} / AQ_{s,ns}(t) & \text{otherwise} \end{cases}$$

Source traffic rate: $XMAINR_{h,d}(t)$ (an exogenous rate)

Delayed routing decision:

$DRS_{s,ns,d}$ (determined by the minimum distance table)

LEVELS

Link queue length for each destination:

$$XLS_{s,ns,d}(t) = XLS_{s,ns,d}(t-DT) + DT * RS_{s,ns,d}(t-DT) - DT * RN_{s,ns,d}(t-DT)$$

Packets held by switch:

$$XMAINL_{s,d}(t) = XMAINL_{s,d}(t-DT) + DT * XMAINR_{h,d}(t-DT) + DT * \sum_{ns} RN_{ns,s,d}(t-DT) - DT * \sum_{ns} RS_{s,ns,d}(t-DT)$$

NOTATION s denotes a specific switch
ns denotes a nearest-neighbour switch of s
"t" denotes the current simulation time
h denotes the host connected to s
d denotes a destination
DT denotes the time increment
$CAP_{s,ns}$ denotes the capacity of the link connecting s to ns

## 2.5 Performance Evaluation

The simulation provides access to the number of packets held by each switch and the link traffic rates at each time step. Thus a wide range of performance functions may be defined, depending upon the wishes of the user. For example, if the time taken to regain stability of the network afer an external shock is of interest, then a possible performance function PF could be

$$PF(t) = \left\{ \sum_{\substack{\text{all output queues} \\ (s,ns)}} \left[ AQ_{s,ns}(t) - AQ_{s,ns}(t-DT) \right]^2 \right\}^{1/2}$$

If the average packet delay is of interest, then the performance function may be obtained by calculating the average total queue length for all source-destination routes defined by the routing table (bearing in mind that the routes will vary with time because of the adaptive nature of the routing algorithm).

## 3. THE IMPLEMENTATION

### 3.1 General Structure of the Program

The simulation program NETSIM is written in FORTRAN-IV and runs on a VAX 11/780 under the VMS operating system. The investigation of different types of routing algorithm or performance functions necessitates the replacement or modification of the appropriate FORTRAN subroutines.

The high-level program structure follows the System Dynamics methodology, i.e.:

1. Initialise levels and external rates.
2. Calculate auxiliary variables from levels. (Subroutine RALG)
3. Calculate rates from levels and auxiliary variables. (Subroutine RATE)
4. Output levels and rates.
5. Calculate new levels from rates. (Subroutine LEVEL)
6. If not end of simulation, go to Step 2

Program listings may be found in Appendix 1; definitions of the Fortran variables may be found in Appendix 2.

### 3.2 Questions of Storage

In general the connectivity of a network is such that every switch is not connected to every other switch. Thus the two-dimensional representation of a link as (switch,neighbouring switch) (as given in Figure 3) is wasteful of storage: a realistic network of N nodes will possess far less than the possible maximum of N(N-1) links (assuming that each link is unidirectional). The data structure employed in the program "flattens" this two-dimensional representation into a one-dimensional representation, as shown in Appendix 3. As the one-dimensional representation requires storage proportional only to the number of actual links, a considerable saving of spac is accomplished.

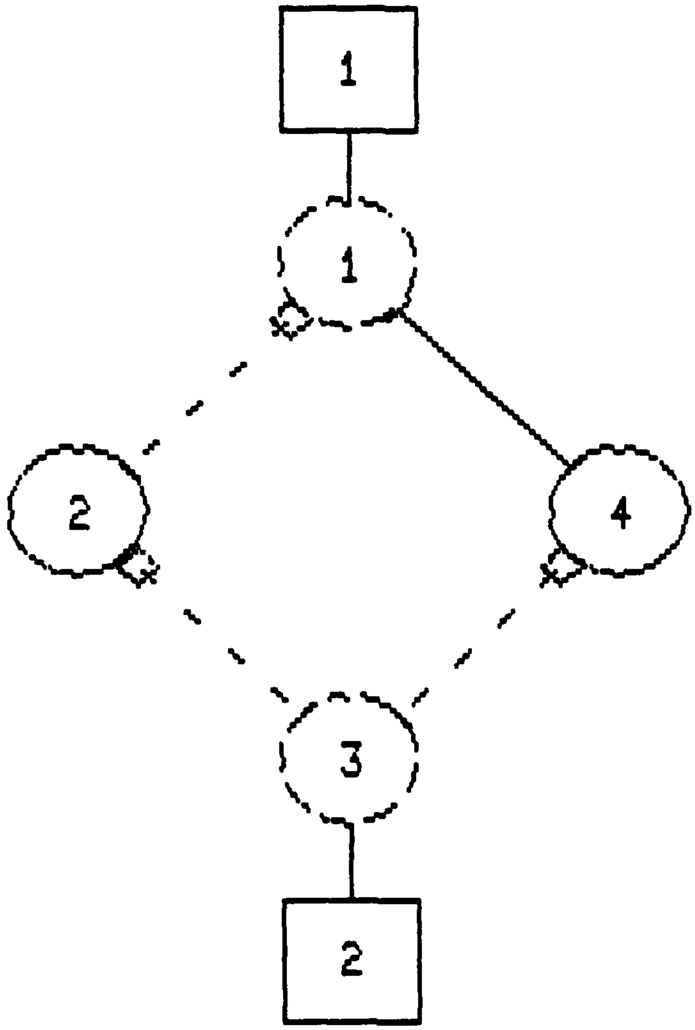
### 3.3 The Graphics Display

The program's graphical output can take two forms; either as a display showing the network (as explained in this Section) or as a plot of a performance function against time. Both of these make use of terminals with REGIS graphics capabilities.

Initially the display gives a picture of the network containing the labelled switches, hosts, and links. Switches connected to hosts may be drawn with broken lines: this convention allows us to omit the hosts themselves in order to provide a less cluttered display.

As the simulation runs, representations of the current queue lengths (in the form of a box with length proportional to the queue length - one box for each output queue) and of the current link loadings (in the form of lines of varying solidity - the less solid the line, the more load there is) are continually updated to provide a dynamic overview of the network performance. A more objective measure of the performance may be obtained by requesting a graph of the performance function with respect to time.

FIGURE 4: EXAMPLE OF GRAPHICS OUTPUT



APPENDIX 1; LISTINGS OF SIMULATION FILES

```
      SUBROUTINE RALG
C subroutine for the general routing algorithm
      INCLUDE 'PARCOM.FOR'
      DIMENSION ADT(NSMAX,NHMAX),AMDV(NSMAX,NHMAX)
      DATA ((ADT(I,J),I=1,NSMAX),J=1,NHMAX)/N2*0.0/
      DATA ((AMDV(I,J),I=1,NSMAX),J=1,NHMAX)/N2*0.0/
C initialisation
      IF(TIME.GT.SMALL) GO TO 10
        DO 20 I=1,NS
          DO 30 L=1,NH
          ADT(I,L)=0.0
          AMDV(I,L)=0.0
   30     CONTINUE
   20   CONTINUE
   10 CONTINUE
C queue lengths found
        DO 40 I=1,NS
          DO 50 J=NP(I),NP(I+1)-1
          AQ(J)=0.0
          IF(ISS(J).GT.NS) GO TO 50
            DO 60 L=1,NH
            AQ(J)=AQ(J)+XLS(J,L)
   60       CONTINUE
   50     CONTINUE
   40   CONTINUE
C average distance tables found
        DO 70 I=1,NS+NH
          DO 80 J=NP(I),NP(I+1)-1
          K=ISS(J)
          IF(K.GT.NS) GO TO 80
            DO 90 L=1,NH
            ADT(J,L)=AQ(J)+AMDV(K,L)+BIAS
   90       CONTINUE
   80     CONTINUE
   70   CONTINUE
C average minimum distance vector reset to allow recalculation
        DO 100 IB=1,NS
          DO 110 LB=1,NH
            DO 120 JB=NP(IB),NP(IB+1)-1
            IF(ISS(JB).EQ.LB+NS) GO TO 110
  120       CONTINUE
          AMDV(IB,LB)=XLARGE
  110     CONTINUE
  100   CONTINUE
C minimum distance vector and next switch found
        DO 130 IC=1,NS
          DO 140 JC=NP(IC),NP(IC+1)-1
          KC=ISS(JC)
          IF(KC.GT.NS) GO TO 140
            DO 150 LC=1,NH
            IF(ADT(JC,LC).GE.AMDV(IC,LC)) GO TO 150
            AMDV(IC,LC)=ADT(JC,LC)
```

```
            IF(AMOD(TIME,TUP).GE.SMALL) GO TO 150
            IAUX(IC,LC)=JC
  150       CONTINUE
  140     CONTINUE
  130   CONTINUE
C check that routing not via a host
        DO 160 L=NS+1,NS+NH
        L1=L-NS
          DO 170 J=NP(L),NP(L+1)-1
          I=ISS(J)
            DO 180 J1=NP(I),NP(I+1)-1
            IF(ISS(J1).LE.NS) GO TO 180
            IAUX(I,L1)=J1
  180       CONTINUE
  170     CONTINUE
  160   CONTINUE
C control returned to simpri/simdis/qfunc
      RETURN
C end statement ************************************************
      END
```

```
      SUBROUTINE RATE
C rate equations for network simulation
      INCLUDE 'PARCOM.FOR/LIST'
        DO 10 I=1,NS
          DO 20 J=NP(I),NP(I+1)-1
            DO 30 L=1,NH
C rate set to 1 if along route 0 if not
            DRS=0
            IF(J.EQ.IAUX(I,L)) DRS=1
            IF(ISS(J).LE.NS) THEN
C internal rates found using delay function
              RS(J,L)=DELAY(DRS,J,L)*XMAINL(I,L)/DT
            ELSE IF(I.EQ.L) THEN
C internal rate for switch connected to host does not use delay
              RS(J,L)=XMAINL(L,L)/DT
            ELSE
              CONTINUE
            END IF
   30       CONTINUE
C traffic rate initialised
          TR=0.0
C loops to find external rates
            DO 40 L1=1,NH
            TR=TR+XLS(J,L1)/DT
   40       CONTINUE
C check for link down
          IF(CAP(J).LT.SMALL) THEN
            TRC=0.0
C check for traffic rate > capacity
          ELSE IF(TR.GT.CAP(J)) THEN
            TRC=CAP(J)/TR/DT
          ELSE
            TRC=1.0/DT
          END IF
C network (external) rates found
            DO 50 L2=1,NH
            RN(J,L2)=XLS(J,L2)*TRC
   50       CONTINUE
   20     CONTINUE
   10   CONTINUE
C control returned to simpri/simdis/simplt
      RETURN
C end statement ************************************************
      END
```

```
      FUNCTION DELAY(RNEW,J,L)
C delay function
      INCLUDE 'PARCOM.FOR/LIST'
      DIMENSION DEL(NCMAX,NHMAX,2)
      DATA (((DEL(I,J,K),I=1,NCMAX),J=1,NHMAX),K=1,2)/N7*0.0/
C check for zero delay
      IF(MDEL.EQ.0) THEN
        DELAY=RNEW
C control returned to rate if zero delay
        RETURN
C check for unit delay
      ELSE IF(MDEL.EQ.1) THEN
        DELAY=DEL(J,L,1)
        DEL(J,L,1)=RNEW
C control returned to rate if unit delay
        RETURN
C delay>1
      ELSE
        DELAY=DEL(J,L,1)
          DO 10 L1=1,MDEL-1
          DEL(J,L,L1)=DEL(J,L,L1+1)
   10     CONTINUE
        DEL(J,L,MDEL)=RNEW
      END IF
C control returned to rate if delay > 1
      RETURN
C end statement ************************************************
      END
```

```
                 SUBROUTINE LEVEL
C subroutine for level equations used in network simualtion
      INCLUDE 'PARCOM.FOR/LIST'
        DO 10 I=1,NS
          DO 20 J=NP(I),NP(I+1)-1
            DO 30 L=1,NH
            IF(ISS(J).GT.NS) GO TO 30
C source destination output queue
            XLS(J,L)=XLS(J,L)+DT*(RS(J,L)-RN(J,L))
C decrease in input queue due to internal rate
            XMAINL(I,L)=XMAINL(I,L)-DT*RS(J,L)
   30       CONTINUE
   20     CONTINUE
   10   CONTINUE
C traffic from neighbouring switches
        DO 40 I=1,NS
          DO 50 J=NP(I),NP(I+1)-1
          K=ISS(J)
          IF(K.GT.NS) GO TO 50
            DO 60 L=1,NH
C increase in input queue due to external rate
            XMAINL(K,L)=XMAINL(K,L)+DT*RN(J,L)
   60       CONTINUE
   50     CONTINUE
   40   CONTINUE
C traffic from neighbouring hosts
        DO 70 LA=NS+1,NS+NH
          DO 80 JA=NP(LA),NP(LA+1)-1
          KA=ISS(JA)
            DO 90 LB=NS+1,NS+NH
C increase  in  input  queue due to traffic from host x is dummy
C variable
          X=DT*XMAINR(LA-NS,LB-NS)
            XMAINL(KA,LB-NS)=XMAINL(KA,LB-NS)+X
   90       CONTINUE
   80     CONTINUE
   70   CONTINUE
C control returned to simpri/simdis/simplt
      RETURN
C end statement ************************************************
      END
```

APPENDIX 2; VARIABLES USED IN THE GENERAL SIMULATION ROUTINES

## ARRAYS

In INCLUDE FILE PARCOM.FOR,called in all routines:

```
ISS    ; Auxiliary containing the connections for each switch
         and host.
NP     ;  Auxiliary defining the regions in ISS belonging to
         each switch or host.{See notation sheet}
XMAINR ; Rate between host and switch.
XMAINL ; Level containing total number of packets in switch.
RS     ; Rate in switch used to split XMAINL depending on the
         best route availiable to the destination required.
XLS    ; Level in switch containing number of packets to be
         sent along the connection defined in the notation.
RN     ; Rate along the links between switches.
IAUX   ; Auxiliary containing the point at which the ISS array
         gives the best connection for the route to the
         destination.
```

Used only in RALG:

```
AQ     ; Auxiliary containing the output queue length along the
         defined connection.
AMDV   ; Auxiliary containing the estimated minimum distance
         from switch to destination.
ADT    ; Auxiliary containing the estimated distance from switch
         to destination over all connections.
```

Used only in DELAY:

```
DEL    ; Auxiliary containing the last n values of DRS (see
         below) for each connection,where n denotes the
         size of delay
```

## VARIABLES AND CONSTANTS

Used in all routines:

```
NS     ; Number of switches.
NH     ; Number of hosts.
LP     ; The channel number for output.
XLARGE ; Contains a large number.
DT     ; Time interval used in calculations.Rates,levels and
         auxiliaries are updated after each time interval DT.
```

Used only in RATE:

```
DRS    ; Acts as a switch with value 1.0 or 0.0 depending
         upon whether the connection under consideration
         is part of the selected route to the destination
         under consideration or not.
MDEL   ; Denotes the size of delay required.
```

```
Used only in DELAY:
M      ; As MDEL in RATE
RNEW   ; As DRS in RATE
```

NB.

All other variables beginning with N are used for initialisation purposes. All I,J,K,or L variables are used as loop counters to define array positions.

## APPENDIX 3

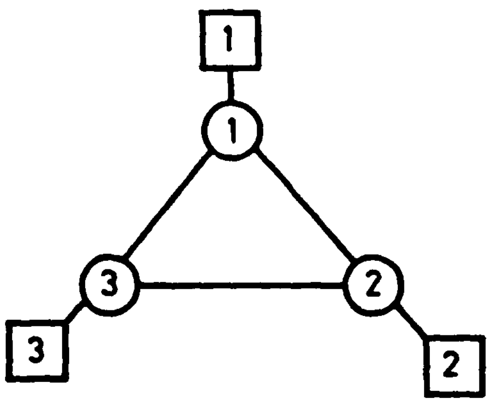
```
NP       1   4   7  10   11   12   13

         >    1    <>    2    <>    3    <>1<>2<>3<

ISS      2   3   4   3   1   5   1   2   6   1   2   3

         >                               <>        <
                   switches                 hosts
```

The array NP implicitly lists the number of entities (i.e switches and hosts) connected to each switch or host.

The array ISS lists the actual entities connected to each switch and host. Thus the entries of NP demarcate the regions in ISS appropriate for each switch or host.

The entities connected to switch I are held in positions NP(I) to NP(I+1)-1 of ISS ; e.g. in the illustrative example above, switch 2 is connected to entities 3,1, and 5. The first two are switches, whilst entity 5 corresponds to host 2 (i.e. the number of switches NS is equal to 3, and thus entity 5 is host (5-NS) = host 2).

When ISS is decoded for the above example, it yields the following network:



## REFERENCES


[1] T.E.G.King "System Dynamics and Network Control", R.S.R.E. Memorandum No. 3755 (1984)

[2] M.J.Awcock "The Simulation of Communications Networks Using System Dynamics" CC1 Divisional Memorandum No.14 (Unpublished) (1985)

UNLIMITED

DOCUMENT CONTROL SHEET

Overall security classification of sheet ........UNCLASSIFIED........

(As far as possible this sheet should contain only unclassified information. If it is necessary to enter classified information, the box concerned must be marked to indicate the classification eg (R) (C) or (S) )

| 1. DRIC Reference (if known) | 2. Originator's Reference<br>Memorandum 3866 | 3. Agency Reference | 4. Report Security Classification<br>U/C |
|---|---|---|---|

| 5. Originator's Code (if known) | 6. Originator (Corporate Author) Name and Location<br>ROYAL SIGNALS AND RADAR ESTABLISHMENT |
|---|---|
| 5a. Sponsoring Agency's Code (if known) | 6a. Sponsoring Agency (Contract Authority) Name and Location |

7. Title

A system dynamics model for communications networks

7a. Title in Foreign Language (in the case of translations)

7b. Presented at (for conference papers) Title, place and date of conference

| 8. Author 1 Surname, initials | 9(a) Author 2 | 9(b) Authors 3,4... | 10. Date | pp. ref. |
|---|---|---|---|---|
| Awcock M J | King T E G | | 8.1985 | |

| 11. Contract Number | 12. Period | 13. Project | 14. Other Reference |
|---|---|---|---|
| | | | |

15. Distribution statement

UNLIMITED

Descriptors (or keywords)

continue on separate piece of paper

Abstract

An abstract model of a communications network in system dynamics terminology is developed as implementation of this model by a fortran program package developed at RSRE is discussed. The result of this work is a high-level simulation package in which the performance of adaptive routing algorithms and other network controls may be assessed for a network of arbitrary topology.

S80/48

END

FILMED

3-86

DTIC