Routing and Metering
Status Report

Robert E. Kahn
Robert Sittler
15 September 1971

PART I: A STUDY OF METERING

SUMMARY AND CONCLUSIONS

This draft document reports primarily on the current status of the metering study. Its purpose is to document, in rough form, the state of progress but is not intended to be a full polished exposition.

Our conclusions, as of 14 September, are summarized below:

1) The original aims of routing and metering still seem to be valid; to increase throughput and efficiency on the lines both routing and metering are needed.
2) The original proposed metering scheme performed poorly.
3) A new metering scheme was developed and appears to work quite well.
4) Testing of the metering was done by simulation and with fixed routing. The interaction with dynamically changing routes still ought to be evaluated for applicability as well as for technical operation.
5) The estimate of needed core memory has been deferred, but a comparision with the old scheme is possible. The new scheme will undoubtedly allow a savings relative to the old scheme.

Our primary recomendation is to proceed with an implementation of the new algorithm, for purposes of testing but not yet for phasing over. A more concrete approach is now appropriate. The study will probably be beneficial to continue at low key at least until the implementation is finished and available for testing.

TABLE OF CONTENTS

I. INTRODUCTION

This draft document describes the current status of our study on routing and metering in the ARPA network. The routing establishes paths through the network along which packets are allowed to flow. The metering regulates the flow of packets along these paths.

During the course of this study, we focused our attention on the definition and evaluation of a metering procedure. A simulation program was constructed to aid in the process of algorithm definition and evaluation. Using this simulation, a metering proposal considered in an earlier study was found to perform poorly. In section II, we describe a new metering procedure which has proven to work quite well in numerous tests. A brief description of the simulation program used in the testing is given in section III. Some experimental results are presented in section IV along with an evaluation of those experiments. Some conclusions are contained in section V.

The earlier study described a routing procedure that was used as a starting point in this study. To concentrate entirely on metering, the dynamics of the routing procedure were inhibited and metering was confined to operate with a set of routes fixed in advance for each case. A study of the relation between dynamic routing and the metering was deferred.

II. TECHNICAL DESCRIPTION OF METERING

The flow of traffic is metered by the IMP's to stabilize the buildup of flow and to allow the spacial distribution of traffic to be slowly adjusted for increased flow, if possible. Metering regulates the maximum traffic to each destination allowed to pass through each IMP.

Two distinct and independent meters are maintained by each IMP, one specifically for combined Host input and a second meter for combined output to lines. There are four cases to consider:

1) traffic arriving from an IMP and headed to another IMP is regulated only by the line meter.
2) traffic arriving from the Host and headed to another IMP is is regulated by the Host meter and then by the line meter.
3) traffic from another IMP headed to a Host at the current IMP is not metered in the destination IMP.
4) traffic from a host at the current IMP and headed to another Host at the current Imp is not metered (i.e. the Host meter is deactivated in this special case.)

The relation of these two meters is illustrated schematically in figure 1 below.

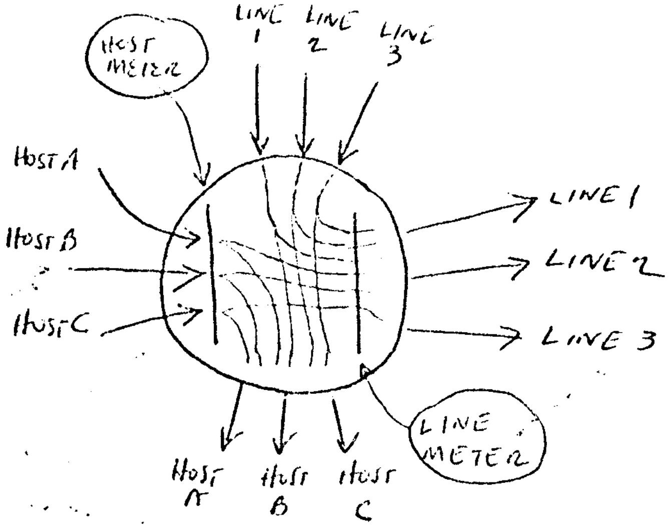


Figure 1, Relation of Host and Line Meters

Each meter operates on a threshold crossing principle using a pair of counters C and D per destination. For clarity, we will focus on traffic to a single destination in this section and only consider the corresponding two counters for that destination.

The counter D is a slowly adjustable threshold. The counter C provides a time base by regularly counting up. When C equals or exceeds D, traffic is allowed to flow past the meter. Otherwise traffic is inhibited from flowing. The limits on C are $\phi \leq C \leq 2D$. For each packet to a given destination that passes the meter and is placed on a queue, C is decremented by D, to reset the time base. Every unit of time (defined to be 20 ms. in this memo) C is incremented by a constant , unless the upper limit 2D has already been attained. If C = 2D and the D limit decreases to D'. C is also decreased to 2D'.

The counter D is initialized to zero, and in steady state with no congestion both C and D remain at zero. When congestion begins to occur, the threshold D is allowed to rise thus permitting the input traffic rate to be reduced. The action of these counters is illustrated schematically in figure 2 below. The shaded sections on the D line, just below the clipped peaks, indicate the periods when traffic is allowed to flow. With a constant and heavy offered traffic load, C linearly counts up until it reaches threshold D at which time one packet is permitted to pass and C is reset to zero.

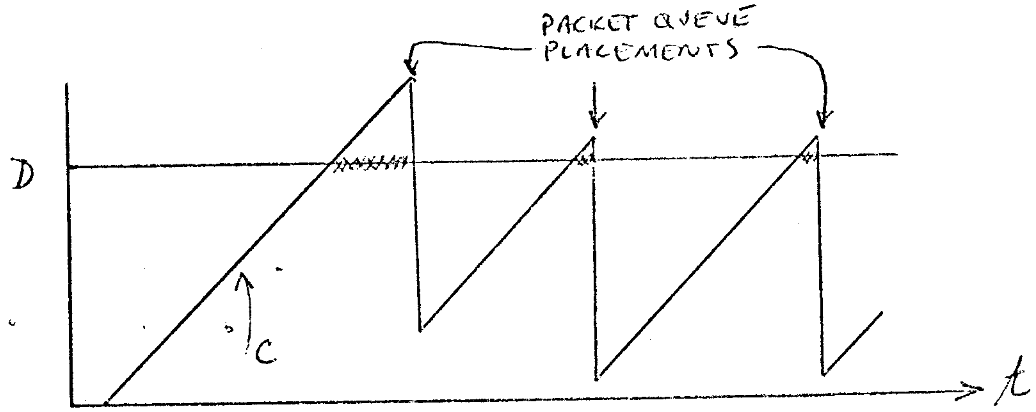


Figure 2. Threshold Metering with C and D

The level of D regulates the flow. When D is zero, packets may pass the meter freely. When D is sufficiently large, packets may rarely pass the meter. Various choices of constants have been tried in the simulation. In the latest version, C is incremented by 5Ø every unit of time. The limit on D is 6ØØ, thus C can count up full scale to 12ØØ in about a half second.

The threshold D is incremented only when packets are "negatively acknowledged". Three types of acknowledgments - positive, null, and negative - are introduced, with only the latter one being used to affect D.

For the moment, let us focus on packets arriving from a neighboring IMP. The situation is only slightly different for packets arriving from a Host, as we shall see. Each arriving packet which passes the line meter and is placed on a queue is positively acknowledged. Each packet which arrives correctly but fails to achieve a queue placement (two possibilities - no room for buffers or a line meter rejection) is discarded and negatively acknowledged. For each packet which arrives in error a null acknowledgement is returned and the packet will be retransmitted.

The D counter is incremented by 1Ø for each received negative acknowledgement, but is unaffected by positive and null acknowledgements. The D counter is decremented by one every unit of time. When D equals 6ØØ (the upper limit) one packet is allowed to pass at most every quarter second. For n equal to 50, one packet may pass at most every 20 ms. The corresponding maximum data rates are approximately 4 kilobits/sec and 50 kilobits/second respectively.

The use of negative acknowledgements (nacks) is essential. Consider for example, the following situation which illustrates how a poor flow of traffic can result if negative acknowledgements are not used.

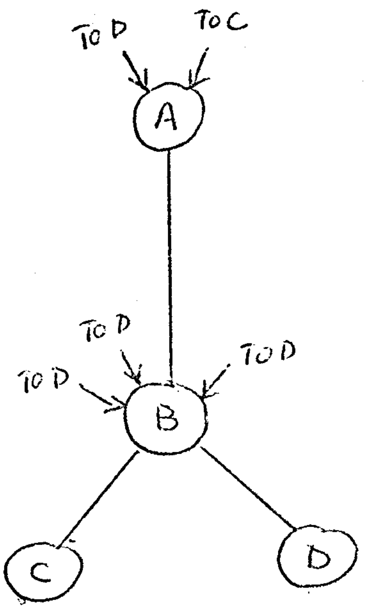


Figure 3. The Need for Negative Acknowledgments

The offered traffic to C is 50 kilobits/sec from one Host. The offered traffic to D is 50 kilobits/second from each of four Hosts. A full queue will build up at B headed to D. A queue will not build up at B headed to C. The IMP at A should direct about 3/4 traffic headed to C on the route to B, since at most only /4 of its traffic will be able to reach C. This distribution is possible using nack's from B, but not in time by using information local only to A. In the other case, traffic to C would be lowered to 1/4 of the circuit bandwidth.

The percentage of steady state nack'd traffic must be reasonably small for efficient use of the circuits, yet not so small as to be useless for regulation. A 1Ø% nack rate at full capacity appears to be an acceptable upper bound. Incrementing D by 1Ø for each nack and decrementing D by one per unit of time corresponds to a 1Ø% steady state nack rate at 5Ø kilobits/second

The grain(or quantization) on traffic regulation is related to the nack rate. To regulate down to say 5 kilobits/sec requires that a nack every 1Ø time units just maintain steady state. Thus, if D is incremented by 1Ø per nack and decremented by 1 per unit of time, only data rates above 5 kilobits/sec can be regulated. For this choice, with 5Ø kilobits/sec of traffic metering will occur when at least 1Ø% of the traffic is negatively acknowledged. With under 5 kilobits /sec of traffic metering is inibited regardless of the nack rate. In this case, the net typically reverts quickly to minimum hop routing. For these values, a full scale traverse of D takes slightly over a second. It takes about 1Ø seconds for D to drop full scale.

Decreasing D by one per unit time allows the flow to stabilize. Decreasing D by one per acknowledgement (rather than by one per unit time) makes the regulation independent of rate and therefore causes no regulation to be possible. Decreasing D every n units of time (provided that one or more nack's has occurred over that period of time) is equivalent to decreasing D every unit of time. The corresponding increment for D in this case may be made equal to 10 (or any other number) by a suitable selection of n.

Traffic from the Host is metered in a similar way using "pseudo nacks". For every packet from the host or hosts that fails to obtain queue placement and must therefore be held by the host routine, the host D counter is appropriately incremented. The host meter in tandem with the line meter is equivalent to a single meter with a threshold equal to the sum of the two D counters.

The purpose of the host meter is to prevent the host from unfairly obtaining an undesirablly large portion of circuit bandwidth as illustrated in Figure 4 below.

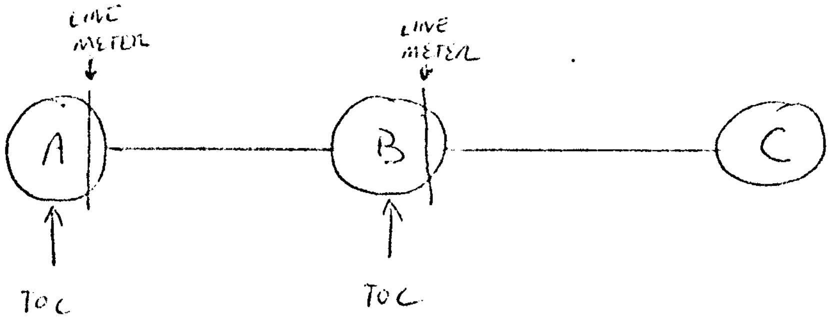


Figure 4, The Need for Host Metering

When the hosts at A and B are sending to C at maximum rate, the queue at B will fill. Host A will be regulated to a trickle of traffic by the line meter at A via nack's from B; but host B will not be regulated (since no nacks are returned from C) and will therefore gain an unfair competitive advantage in vieing for the circuit to C. Thus, host B must be regulated and a host meter is used for this purpose. The reservation of buffer space by the flow control algorithm cannot apportion the two flows equally since, buffer space will typically always exist at the destination.

Only one pair of counters per destination per IMP and only the total traffic flow per destination through the IMP is metered. However, the meters purposely do not determine onto which output line to route the packet. How then is the output line selected? The particular choice is affected by the fact the no queue is ever allowed to get very large.

We presuppose that a set of routes have been opened from each IMP to a given destination. This procedure involves assigning a direction to each line to that destination and only allowing traffic to flow one way. In this study, we assume these directions may not be changed dynamically. The lines are assigned priority according to the initial order of establishment. A sample network with fixed routes to destination Imp 7 is shown in Figure 5 below.

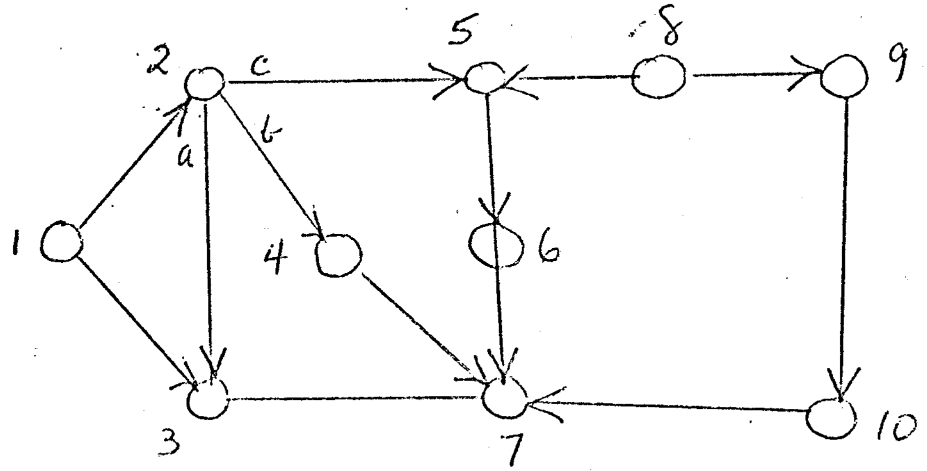


Figure 5. A Sample Network with Fixed Routes.

Let us concentrate on the output lines for Imp 2 in the figure. Traffic passing IMP 2 heading to IMP 7 has three possible output lines labeled a, b, and c respectively in the figure. Let us suppose a has priority 1, b has priority 2 and c has priority 3. If an arriving packet passes the metering at 2, an attempt will first be made to place the packet on the output queue for a; if no buffer placement is allowed on a, an attempt will be made to place the packet on the output queue for line b. If no buffer placement is allowed on b, an attempt will be made to place the packet on the output queue for c. If all attempts fail, the packet is discarded and a negative acknowledgment is returned. if a nack is then received on b, the priority order reverts to a b c. With priority order b a c to begin, a nack on line a will yield order b c a and so forth.

A cyclic permutation of the priorities whenever a nack is received on the high priority line is known to approximate Kirchoffs law for current flow. That is, if the nack rate on each line (nack's per unit time) is given by Ra Rb and Rc respectively, the percentage of packets sent out line a is given by

$$\frac{\frac{1}{Ra}}{\frac{1}{Ra}+\frac{1}{Rb}+\frac{1}{Rc}}$$

just as for current flow in a resistor network where, in that case, the R's correspond to the resistances. The pairwise interchange of priorities is not as simple to describe analytically. Various other schemes that achieve a smoothed probabilistic flow over the various output lines have been considered and may provide some incremental improvements. However, each is a variant of the scheme presented here , which appears to be fundamentally sound.

III. SIMULATION

In view of the general difficulty of a satisfactory mathematical analysis of the behavior of proposed metering algorithms, a simulation tool has been developed to aid in the study of this area. The simulation program (IBM 1130 - FORTRAN) generates routes and meters messages on a per packet basis through an IMP network. Various statistics on message flows are accumulated and metering counter variables are observed.

The program currently operates with fixed (arbitrary) routing which allows alternate paths. Extension of the simulation to incorporate dynamic routing algorithms would be feasible but is not attempted at present.

The basis of the simulation is to treat time in 20 ms segments, each assumed just sufficient to transmit or input a single packet message. Each packet is independent; reassembly is not simulated. Packets which reach their destinations are immediately removed. Packets enroute are kept on queues and processed on a first in/first out basis. Each output line has a reserved store and forward buffer for five packets.

The simulation assumes the availability of negative acknowledgements (nacks). The flow of acks or nacks is not simulated (they are detected instantly).

The following is a summary of some of the pertinant characteristics and parameters of the program.

1. Input (input cards) consists of:
   a. IMP connections (16 IMPS, 4 lines/IMP (max))
   b. Routing
   c. Message traffic (intensity to each destination)
   d. Metering algorithm parameters

2. Output (printout at selected 20 ms time points) consists of:
   a. Metering counter contents.
   b. Store and forward queue contents (all packets).
   c. Cumulative counts of number of packets:
      (1) Attempting/succeeding input from host.
      (2) Attempting/succeeding transmission to adjacent IMP.
      (3) Delivered at destination.

   (Counts are subdivided on a destination basis.)

3. Host/IMP connections and timing.
   a. Each IMP has a single host supplying it with traffic for the net.
   b. IMP to IMP lines transmit one packet (or none) or 20 ms basic time (50 kb).
   c. Host to IMP lines input two packets (or none or one) per 20 ms (100 kb).
   d. Thus for each 20 ms there are three processes defined at each imp (line transmission-host input (1) - host input (2)). The order of these three processes at all IMP's is randomly scrambled anew each 20 ms interval.

4. Packets/queues.
   a. Packets carry three designations (time of entry, origin IMP, destination IMP).
   b. Packets are kept on store and forward queues assigned to an output line. Maximum queue length per line is 5.
   c. Queue processing is first in/first out.
   d. No reassembly at destination is done. Packet is removed.

5. Routing/metering.
   a. Routing is fixed, allows alternates.
   b. Metering is modeled in detail, based on metering counters and priority shifts.

c. Routing and metering of packets is driven by instantaneous acks/nacks.

d. Failure to input packet from Host results in hanging host.

e. Failure to transmit packet successfully to adjacent IMP results in internal packet reassignment to alternate route (if any).

6. Traffic generation.

a. Fraction of Host capacity (100 kb) assigned to each destination is specified.

b. Packets are generated independently at random according to this distribution of their respective destinations. (Host has no memory for failed input.)

c. Unused capacity is specified by a dummy (null) destination.

7. Program data.

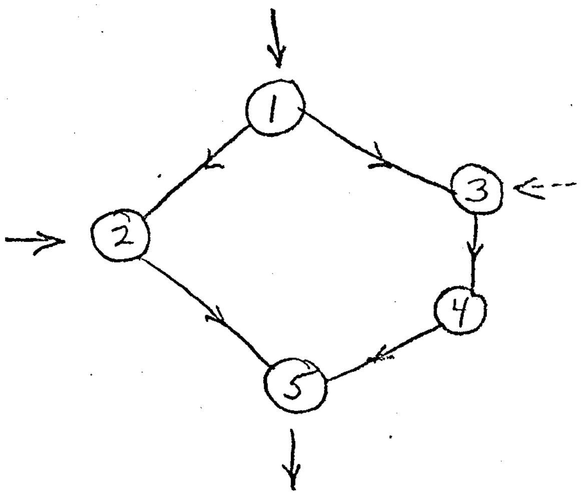
Size is approximately 400 FORTRAN statements compiling to 3800 program instructions exclusive of utilities, routines, and requires 8900 table locations. Running time is about 10 x real time, but depends on net dimensions.

IV. SOME EXPERIMENTAL RESULTS

Experiments using the packet flow simulation program have progressed through two phases. In the first phase we investigated the metering algorithms which were originally proposed. The results of these first experiments were uniformly unsatisfactory, indicating that the original algorithm could not work.

Phase I

The following network was used in this series of experiments.



Traffic is generated at IMPs 1 and 2 for IMP 5. IMP 1 has a first priority route through 2 and a second priority through 3-4-5. IMP 2 has only the direct route to 5. (Additional traffic for IMP 5 may be injected at IMP 3.)

The first observation made of this system was that IMP 2 input got as much line capacity as it needed and IMP 1 input traffic took the rest regardless of the competing IMP 3 input. IMP 2 was always dominant.

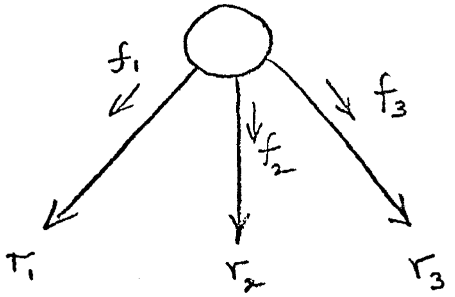
The explanation is that since 5 is the destination, no nacks are generated on attempts at 2-5 transmissions. Thus metering counters at 2 do not inhibit assignment to queues there. Host input at 2 always gets through if space is available on queue, and it is not deterred by the occurrence of refusals. On the other hand, meters at 1 are activated by such refusals (they produce nacks) and this metering reduces the assignment to queues at IMP 1.

Thus it became evident that the process of metering as originally proposed was incomplete and an additional metering of host inputs were required. Subsequent experiments were carried out with this additional metering.

We next determined that although this improvement produced a more equitable use of the line 2-5 by inputs at 1 and 2, we could not get the metering to properly divert the input at 1 from the route 1-2-5 to the alternate 1-3-4-5 without simultaneously inhibiting input at 2.

The reason for this behavior is that the both packet flow from 1 to 2 and input at 2 encounter the same average occupancy conditions while competing for space on the output queue at 2. Thus the nack probabilities are similar. Since it is the nacks that drive the metering system (a given nack rate leads to a specific metered flow), the metered level of these two packet flows is adjusted to an approximate equality regardless of conditions on the alternate route 1-3-4-5.

A simple mathematical result (discovered after the experiment) shows that this defect is quite general in metering algorithms of the initially proposed type. For example, focus attention of a given IMP with traffic for a single destination and three alternate output lines.



On line i let f_i be the metered packet rate (say per 20 ms) and r_i the rejection (nack) probability at the adjacent IMP. Let α be the D counter increment when a nack is received and β the decrement in real time (per 20 ms). Then if meter i is regulating, the up and down rates are in balance. That is,

$$\alpha f_i r_i = \beta$$

or

$$f_i = \frac{\beta}{\alpha r_i}$$

which shows that f_i is independent of conditions on other lines (i.e., the other r's).

Now attempt a modification in which a nack on line i increments D_i by α but also decrements other D's by an amount δ. We would have at equilibrium,

$$\begin{aligned} \alpha f_1 r_1 - \delta f_2 r_2 - \delta f_3 r_3 &= \beta \\ -\delta f_1 r_1 + \alpha f_2 r_2 - \delta f_3 r_3 &= \beta \\ -\delta f_1 r_1 - \delta f_2 r_2 + \alpha f_3 r_3 &= \beta \end{aligned}$$

The obvious solution of this set of equations is

$$f_1 r_1 = f_2 r_2 = f_3 r_3 = \frac{\beta}{\alpha - 2\delta}$$

a result not significantly different from the above.

In fact. we see that any set of three equilibrium relations in terms of nack rates $f_i r_i$ can be solved to provide

$$f_i r_i = \text{constant}_i$$

which is inadequate for multiline metering. (No solution or multiple solutions are equally bad, for then the metering is physically as well as mathematically indeterminate or equilibrium is impossible.)

By a series of intuitive judgments, a new metering algorithm was proposed. This algorithm

1. Replaced the fixed priority order of lines (per destination) by a freely changing order controlled by the received nacks,
2. Deletes all but a single joint metering C/D counter for all output lines (per destination),
3. Retains the host input metering (per destination) as explained above.

Output meter D counters are increment by nacks on any line. Host input D counters are incremented by failure to place input packets on queue.

All D counters are decremented in real time. The priority order of any line receiving a nack is interchanged with the next lowest order line (if any).

Phase II

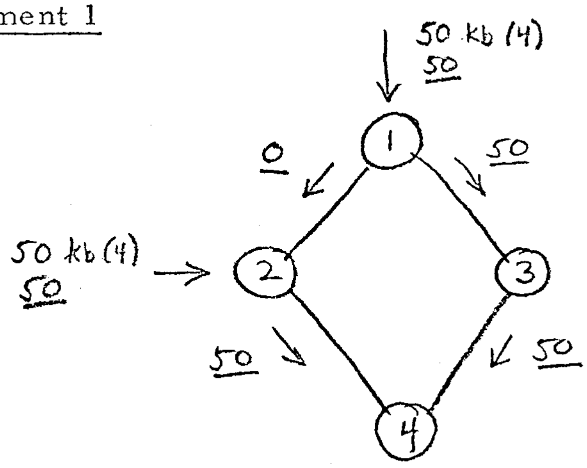
Experiments done with the new algorithm have been quite successful. It does the proper things in each situation that has been examined so far. Although the rather erratic behavior of the priority switching causes traffic "banging" to occur, we have as yet not been able to devise a test which shows a detrimental effect of this behavior or a reflection in the external world.

The experiments were done with the following parameters which proved generally satisfactory.

C (increment)	=	1	
D (increment)	=	1/5	(α)
D (decrement)	=	1/50	(β)
D (maximum)	=	12	
D (minimum)	=	0	
C (maximum)	=	2D	
C (minimum)	=	0	

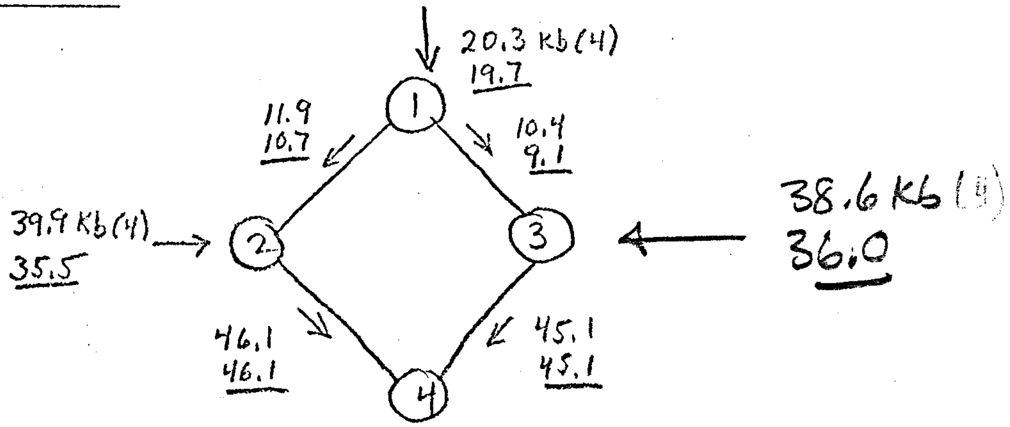
In the following diagrams the lines are all of 50 kb capacity, the input traffic rates and line rates are indicated in kb. Underlined figures are successful flow rates; not underlined indicates attempted flow (may have been nacked or refused input). Paranthesis indicate destination of flow. Initially line 1-2 has priority over 1-3 alternate on the diamond nets.

Experiment 1

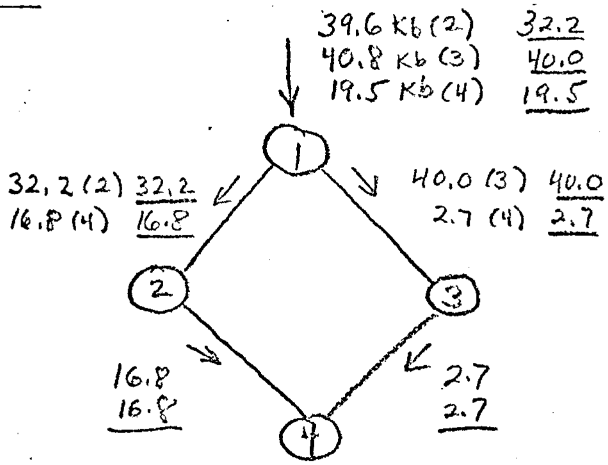


First nack on 1-2 switches the first priority to 1-3 which then continues indefinitely.

Experiment 2

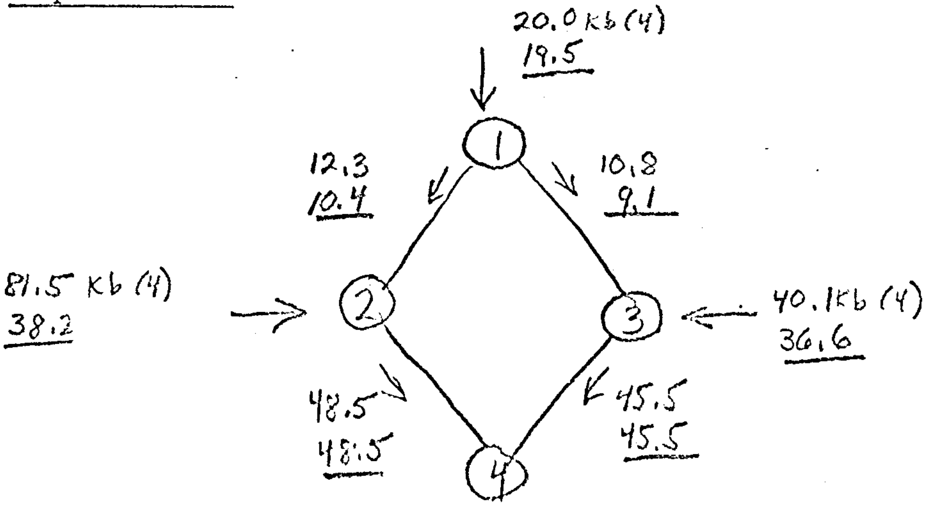


Experiment 3

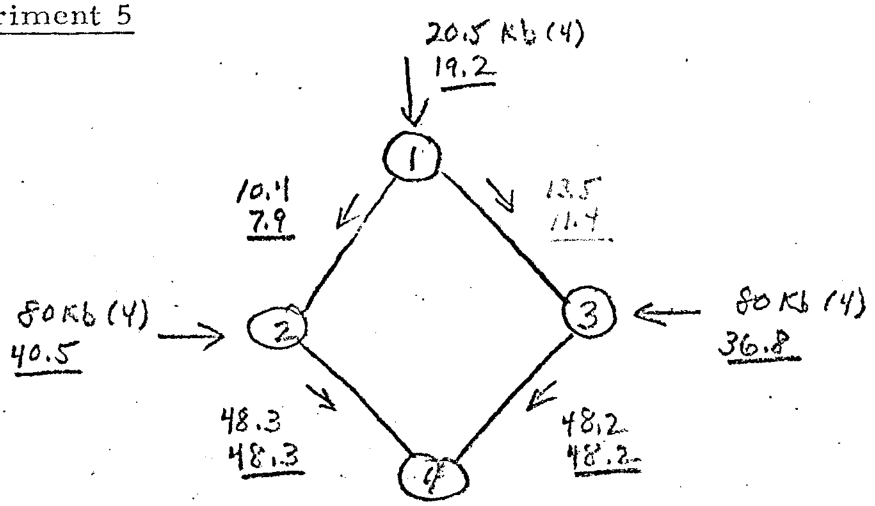


In this case priority never switched.

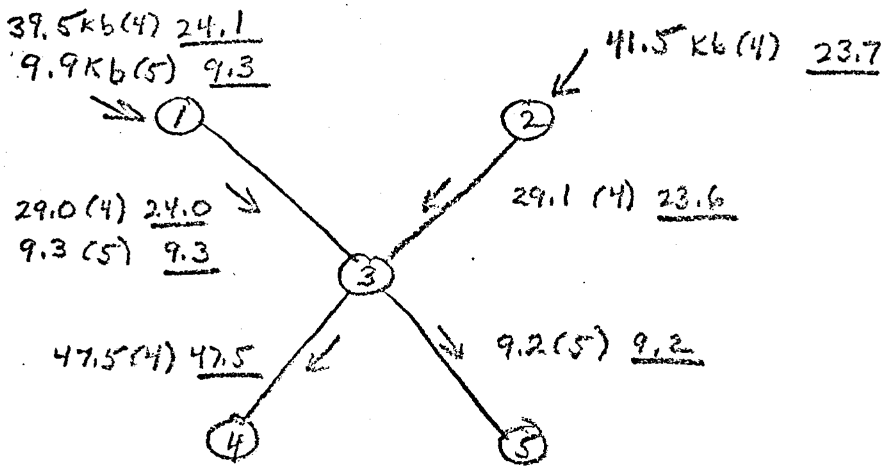
Experiment 4



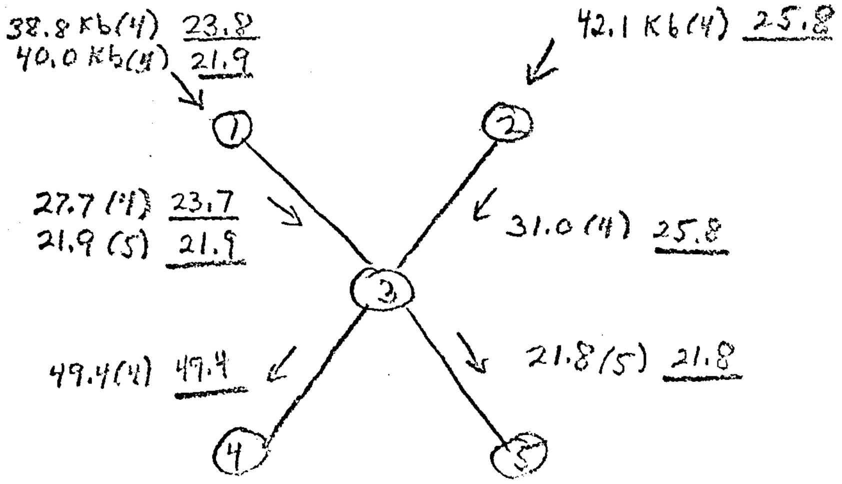
Experiment 5



Experiment 6

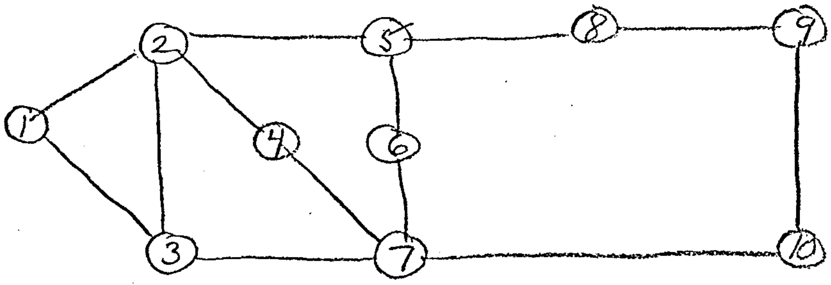


Experiment 7



Experiment 8

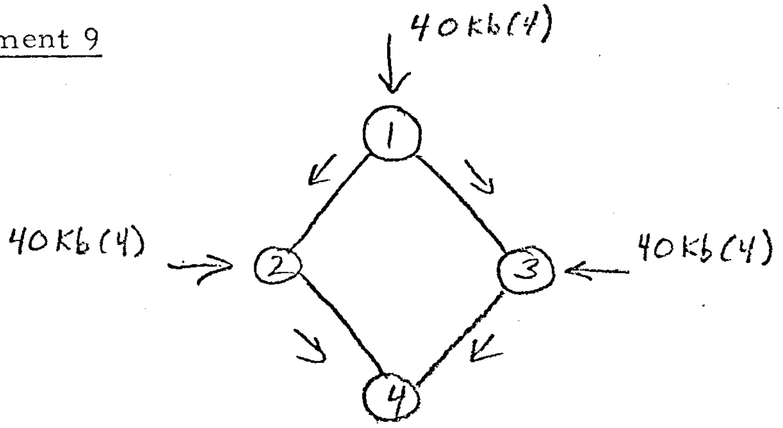
The 10 IMP network was as follows:



Traffic was generated from every IMP equally to every other IMP. Only single shortest routes were allowed.

It was found that nothing much happened below a traffic level of 36 kb total input at each IMP. Almost all transmissions and inputs succeeded and metering did not operate. At 45 kb and above, metering did operate (with $\alpha/\beta = 10$ but not with $\alpha/\beta = 3$). The results at 45 kb with metering disabled did not appear much different than with metering turned on as might be expected from the fact that no alternate routes were available.

Experiment 9



In this experiment a closer look was taken of the shift in route priorities at IMP 1. Initially the 1-2 route had first priority. We found that the metering had the following "banging" tendency after an initial settling period:

1. Traffic is assigned with priority on the 1-2 path (say) colliding with input at 2.
2. The queue at 2 builds, the queue at 3 unloads.
3. Finally a nack is returned from 2 to 1 which shifts priority.
4. Now the queue at 3 builds, the queue at 2 unloads, etc.

The period of this variation is random but averages about 10-15 packet times (≈ 250 ms). The input metering at 2 and 3 is not entirely stable, andalso shows some oscillation. The metering at 1 is quite stable.

SOME CONCLUSIONS

1. The current metering algorithm appears to fulfill two main functions assigned to it:

   a. It proportions traffic properly among alternate routes in inverse relation to the resistance as perceived through the nack rate. (Shown by the diamond experiments.)

   b. It suppresses high volume traffic at the source so as to allow low volume traffic for other destinations to use the net. When, however, the low volume traffic increases to match the high level, a proper competition occurs. (Shown by the cross experiment.)

2. Metering is not effective where there is a dispersed traffic flow (all Hosts to all Hosts) even though the total traffic is of high volume. This results from metering on a per destination basis only. Metering is most effective when only a few users try to dominate the net with high volume traffic to a limited number of destinations. (Shown by the large, 10 IMP net experiment.)

3. The current algorithm causes traffic "banging". However this does not appear to affect flow efficiency. The effect of banging on the proposed routing algorithm, however, is more difficult to assess. It is becoming clear that the metering and routing algorithms may interact. Smoothing of the priority switching may be required for satisfactory metering/routing algorithm operation. We have at least one proposed way to accomplish this. (Banging is illustrated in the last experiment.)

NOTES ON THE SELECTION OF α, β

The selection of α, the D counter nack increment, and β, its real time decrement, is influenced by the following considerations.

1. Since nacks occur randomly, α, β should both be small in order that the resulting D level be stable and smooth.

2. If α, β are too small, the D counter will not be able to respond rapidly to changing traffic conditions.

3. The ratio β/α controls the nack rate upon which the metering operates. β/α should be small in order that nacks be infrequent. This leads to good line efficiency, and operates the queues mostly in an unloaded condition. The smaller β/α is selected, the smaller the volume of traffic that can be successfully regulated. Otherwise, for sufficiently small traffic flow, the nack rate will be too small to support a nonzero D level.

4. However if β/α is too small, the nack becomes a rare event; β and α must then be made individually small yielding heavy smoothing and a very sluggish response.