MEMORANDUM

To: Frank Heart/IMP Guys

From: W. C. Crowther, D. C. Walden, N. W. Mimno

Subject: The New Routing Algorithm

Date: February 11, 1972

The following draft describes the new improved routing algorithm you've been waiting for. We have a working simulation that produces nice results; we believe that the algorithm should be implemented on a test network with the anticipation of some real world problems. This paper is a first draft - questions and comments are invited.

/le

Attachment

THE NEW ROUTING ALGORITHM: CONTENTS

I. GENERAL DESCRIPTION

The new routing algorithm essentially uses a base of information to make two decisions: where to send traffic (routing), and how much to allow (metering).

In making these decisions the algorithm tries to: maximize network throughput; optimize routing for any given set of conditions; maintain fairness among users; smooth out sudden increases or decreases in traffic; prevent network lockup.

With respect to the above, the central aspect of the algorithm is the ability to schedule, and therefore control traffic, to the various destinations along the various lines. Control includes rejection of traffic as well as choice of route, smoothing, amount of traffic allowed, and bias shown towards or against particular traffic. Scheduling in turn requires information about the network. For this purpose, the algorithm defines a value called excess capacity, or the amount of line bandwidth available for scheduling traffic to a particular destination. Scheduled traffic and excess capacity are inter-dependent and in some sense complementary values with the former representing line use and the latter line non-use. Each, however, responds to an additional set of constraints: the complement is rarely exact.

Excess capacity is calculated such that the value indicates how much can be scheduled, while the gradient of values among adjacent nodes indicates direction towards (or from) the destination. Excess capacity values are a function of both network traffic and network geometry and will respond to both dynamically.

In addition to limits on line capacity, the routing algorithm must consider limits on buffer space at the nodes. Each line at a node thus has a queue of available spaces for packets corresponding to its share of buffers. Traffic may be rejected on account of full queues as well as full lines.

The routing algorithm performs the continuous function of dispatching (or rejecting) packets based on the information contained in tables of excess capacities and scheduled traffic, and in the queues. Periodically, the tables are updated by parameters passed between the nodes and revised as a function of the actual traffic encountered. In the simulation, the routing code checks the entire network for traffic thirty-two times (each small tick) between each update of the tables (each large tick, or "half-second time out"). Timing of these two functions is generally a trade-off between currency of information used for routing and the overhead involved in passing such information through the network. In addition, the balance between the functions probably affects the tendency of the network to oscillate as a result of changing state either too often or by too much.

In sum, therefore, the new scheme regulates traffic in both quantity and direction as a function of excess capacity, or the ability of the network lines to carry traffic, and queue length, or the ability of the nodes to store traffic. Significant aspects of this algorithm include the ability to reject traffic and to route traffic for the same destination on several lines simultaneously.

II. ROUTING ALGORITHM

Information Base

Normal bookkeeping operations require information such as network geometry, source and destination of traffic. This information does not relate directly to routing decisions and will not be discussed here. The major tables of interest are: queues, best route excess capacity, actual and scheduled traffic. In addition, some small tables are saved relating to fairness and other such constraints on the system; these will be discussed later.

Each modem is associated with a queue indicating the packets waiting to be sent from that modem to its neighbor. Available buffer space limits the queue length, with equal sharing among modems indicating a length of four. (The algorithm must have some way of determining whether an entry is on a queue the first time.)

The excess capacity table is used by routing decisions but changed only by information update. Every node has such a table containing values for every destination; each node sends its excess values to its neighbors as routing messages. The table, called E maximum, contains the excess capacity and identity of the neighbor best able to take traffic for each destination. The best route is chosen by information updating each cycle and is considered the route with the greatest excess capacity. In an empty net, the best route is also the shortest; in a net with traffic, the best route may have shifted from the shortest if that route is already heavily used.

Scheduled traffic values are saved at each node for every destination along every line. It is possible, for example, to have traffic for a certain destination scheduled on each of the output lines, or alternatively, traffic for every destination scheduled on any one line. Exactly parallel to this table of allowable traffic is a table of actual traffic. They interact in the following manner: at the end of information update, the actual traffic table is set to all zeroes; routing proceeds, and actual traffic is built up towards scheduled as traffic requires; information update reduces scheduled traffic values if actual traffic is less. Scheduled traffic thus remains if used, and gradually diminishes if not used. Routing may increase scheduled traffic for any destination only on a single line at once, in particular, the best route to that destination. To maintain traffic on several lines, therefore, the best line must have shifted and traffic must be sufficient to use the scheduled amounts. Both of these conditions, it may be noted, occur in a heavily used network.

Routing Decisions

Faced with a packet to send along, the routing algorithm chooses one of three general options. First, the packet may be sent to a neighbor on a line with available scheduled traffic. Second, if scheduled traffic is non-existent or has been already used on all lines, the algorithm attempts to increase the scheduled traffic on the best line (best route) to the destination. Third, if traffic cannot be increased for any reason, the packet is either rejected or sent anyway on the best line if traffic is light. In each of these cases, a full queue (lack of buffer space) causes failure. The above conditions apply to a packet at its source as well as in the network.

As a first option, routing attempts to find already scheduled, or allowable, traffic. Lines are checked for scheduled traffic in the order of best line first, then all lines in some arbitrary order. A line with scheduled traffic will be ignored if the output queue for the line is full. In addition, routing builds actual traffic up to the scheduled limit smoothly over time. Because the actual traffic value is periodically set to zero by information update, this last check is necessary to avoid bunching at the beginning of each cycle. A line is thus ignored if the actual traffic sent to a destination has reached the proportion of the scheduled limit indicated by the time elapsed this cycle. In the simulation, for example, a line must have:

$$A_d \le \frac{t}{32} S_d$$

where t equals the number of small ticks elapsed and 32 is the total number of small ticks this cycle. The first line that can pass the above tests receives the packet on its queue. The actual traffic for the given destination on that line is increased, and routing is free for the next packet. If no line can pass, routing tries its next option.

When scheduled traffic is insufficient to handle a request, routing exercises its second option and attempts to schedule more. Scheduled traffic increases subject to a number of constraints which control the rate of increase, competition among users, and direction of increase (which line). Most generally, a node may add no more than two packets to scheduled traffic in any cycle, or big tick. This potential increase must be shared among all destinations over all lines. The amount of increase remains small to prevent sudden changes in the routing information base. To insure a measure of fairness, the line-destination combination that takes the first available packet increase must wait at least half-way through the cycle (big tick) before trying for the second and then must have twice the normal excess. In other words, number one must try harder. The probability of one line-destination pair using both packets therefore declines as other traffic increases. For any given destination, an increase in scheduled traffic may occur only on the best line to that destination. The E maximum table at every node indicates the best route to each destination and also the excess capacity of the neighbor on that best line. As a further condition, the excess capacity of the best line neighbor must be at least one packet (or two, as above). Given all of the above conditions, and that the best line queue has space, routing adds one packet worth of scheduled traffic, transfers the packet to the best line queue, and increments the actual traffic value by one packet. Routing is then free for the next request. Given all of the above conditions, but the best line queue is full, routing rejects the packet by not acknowledging it within the network, or by holding it at a source. If any of the conditions above do not exist, scheduled traffic for the given destination may not increase. If in addition the queue is full, the packet is rejected. If however the queue is not full, the packet may possibly be sent anyway as unscheduled traffic.

Option three allows unscheduled traffic to be sent on the best line under certain conditions. Option three occurs when scheduled traffic is insufficient to handle a packet and cannot be increased, but the best route to the destination has room on its queue. Because scheduled traffic is constrained to increase and decrease slowly, insufficient scheduled traffic does not necessarily imply a full net. Likewise, inability to schedule more traffic may result from fairness or smoothing constraints as well as real congestion. Room on the next queue indicates at least some space in the network. Two cases should be considered: packets already in the net, and packets at a source trying to enter the net. In the first case, routing automatically sends the packet on the best line to its destination if that queue has room. This provides an escape mechanism from system constraints and keeps traffic moving on the best route to each destination if at all possible. Unscheduled traffic does not increase actual traffic values and has no direct affect on the information base. In the second case of packets entering the net, routing becomes more selective in allowing unscheduled traffic. In addition to room on the best line queue, traffic over the net must be light. This condition essentially allows a fast buildup of scheduled traffic from an empty net. Every node keeps a value which is decremented by each unscheduled packet from a source and incremented once each cycle when the excess capacity at the node (E table) for every destination is at least half of a full line capacity (25KB). In the simulation, the value is not allowed to be greater than ten (i.e., ten packets). Since each node has only one such value, the total number of unscheduled packets from all sources remains small and occurs only when traffic has been light over the entire net for several cycles.

Information Update

Information update occurs each big tick or half second time-out. Between each time-out, a node receives new excess capacity values from each of its neighbors. Two primary functions are performed: recalculation of the excess capacity and best route to each destination, and smoothing of scheduled traffic values as a function of actual traffic. Several minor adjustments also occur for the purpose of fairness, shifting traffic to the best route, and allowing unscheduled traffic to start.

Excess capacity values, or the available bandwidths to each destination, are kept in a table at every node called E maximum. For any particular destination, a node compares the capacities of its neighbors to accept traffic for that destination; the node chooses the line with the largest value as the best route and saves the value as E maximum. This value minus one (but not less than zero) is in turn sent to the node's neighbors which perform similar calculations. By definition, a node that is a destination receives a value of 50 kilobits (i.e., the maximum) as its excess capacity to itself. In an empty net, the E maximum values for a destination gradually decrease with distance from the destination. Note that the technique of choosing the best neighbor will always result in the shortest path to a destination in the empty case.

Traffic introduces further constraints to excess capacity values. In the empty case, the only limit on traffic is the ability of a neighbor to accept that traffic; the line itself is free. Once traffic exists on a line, however, the limit on traffic is either the available unscheduled bandwidth of the line or the ability of the neighbor to accept traffic, whichever is smaller. This constrained, or line, excess capacity is the value actually used to determine the E maximum or best route to a destination. The line excess capacity may be less than or equal to the neighbor's original excess, but never greater; the values maintain a gradient with respect to a destination. Traffic on a line to any destination affects the line excess capacities of all destinations. The system thus responds to the overall use of the lines.

As traffic builds from zero on a line, the excess line capacity to the neighbor diminishes. In many cases, an alternate line would then be chosen as the best route (E maximum), only to change again as traffic builds up on the new line. There are a number of reasons for preventing such rapid changes. First, as a general principle the network should change state slowly to prevent oscillation. Second, a route is designated best for some good reason; in the empty net, for example, the best route is the shortest. Since the most likely reason for line excess capacity to decrease is the growing use of the line rather than congestion, the best route should probably stay the same for awhile. Third, and most important, because nodes perform their time-out calculations asynchronously, several cycles may occur before all adjacent nodes reflect a correct gradient to the destination. If a change in the best route were to occur immediately following a reduction in excess capacity, a spur or line in the wrong direction might be chosen as the "best" route. Note that an immediate shift may happen if another line is simply better. For these reasons, the algorithm incorporates a mechanism called hold-down. When the excess capacity on the best route decreases, the best line must not change and the excess value must not increase for two additional cycles. During this time, however, the excess capacity value may further decrease, resetting the hold-down counter to two more cycles. As an example, in a simulated network with four alternate paths to the same destination, the algorithm allowed maximum traffic to build up on the shortest route before shifting to an alternate.

Smoothing of scheduled traffic with respect to actual traffic occurs smoothly and insures that scheduled traffic remains only if used. If actual traffic (A) on a given line to a given destination is greater than the corresponding scheduled traffic (S), the value of S remains constant. If A is either zero or less than S by some amount, S is replaced by 7/8S + 1/8A. (The simulation requires this difference between A and S to be greater than 9, where A and S have maximum values of 255 decimal.) If none of the above, that is A and S are close in value, S is replaced by S-1. These conditions slowly decrease unused scheduled traffic over the network.

Fairness also causes a reduction in scheduled traffic. When the sum of scheduled traffic on any line is greater than the capacity of the line less one packet, the scheduled traffic of the largest user (destination) is reduced by an amount sufficient to bring the sum down to the limit. On a full line, fairness opens a small amount of scheduled traffic to competition from other destinations. On a non-full line, fairness has no effect.

A further correction tends to switch traffic onto the best line for a given destination. If the excess capacity for that destination is greater than two packets and the scheduled traffic over the node did not increase by the maximum allowed during the last cycle, a packet of scheduled traffic is switched to the best route from some other line, if any exists. This best route correction concentrates traffic on the best line to each destination when traffic is in steady state or decreasing.

Finally, information update attempts to increase the unscheduled start-up allowance at the node. Discussed earlier, this value may increase by one only if the excess capacity for all destinations is large (≥25 kilobits), and in any case can be no greater than 10 decimal. This "rubber band" value allows more traffic to start in an empty net. Just before exiting, information update sets all actual traffic to zero, and resets the counter that allows an increase in scheduled traffic.

III. THE SIMULATION

To test the algorithm, a simulation has been written, debugged and run on the PDP-1 computer. Actual and scheduled traffic and excess capacity are stored as eight bit values scaled to 50 kilobits. The simulation provides the option of printing numerous system characteristics such as: throughput achieved by each source, total throughput, hop counts, the information base, number of retransmissions. Simulation is generally a non-exact version of the real case; the routing simulation is limited in the following ways, not necessarily listed in the order of importance.

1. Available core space limits the number of nodes to six in a simulated network.

2. Simulated lines never break; geometry does not change dynamically.

3. Simulated lines are all the same speed (50 kilobits). At some point, the real network will probably contain both faster and slower lines.

4. All simulated messages are the same length, one packet.

5. Simulated acknowledgments take no time to return and are never lost.

6. In the real world, traffic from a source host will be scheduled, or metered, similarly to traffic from within the net. Available core did not permit implementation of this feature in the simulation. As a result, certain geometries which give an unmetered source an advantage cannot be adequately tested. An obvious example is a straight line network, where a source close to a destination can grab more line capacity than those further away.

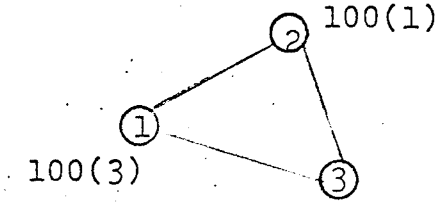
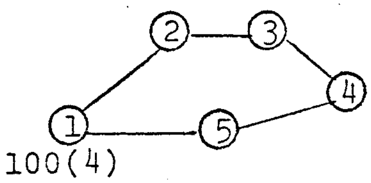
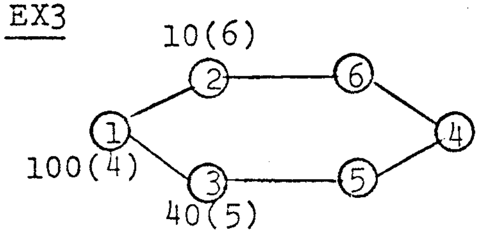
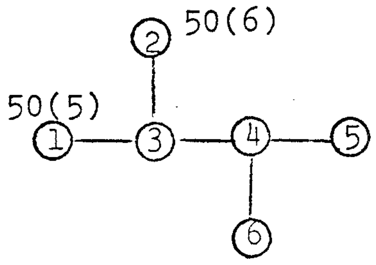
7. The simulation generates steady state traffic rather than dynamically changing traffic.

8. In the simulation, nodes perform information update in a random but synchronous manner, that is, all at once before routing decisions continue. In the real network, information update will be asynchronous.

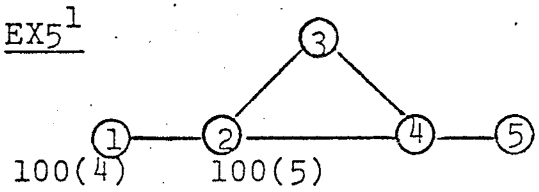
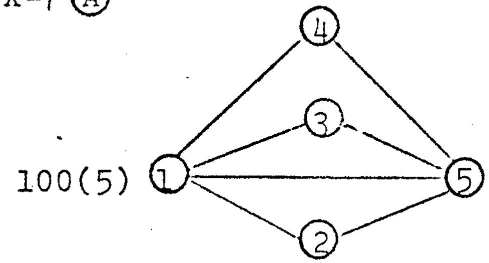
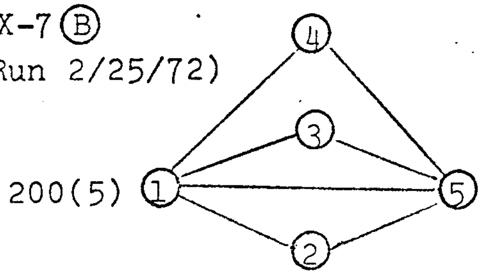
9. The simulation operates independently of the IMP software.

IV. CONCLUSIONS

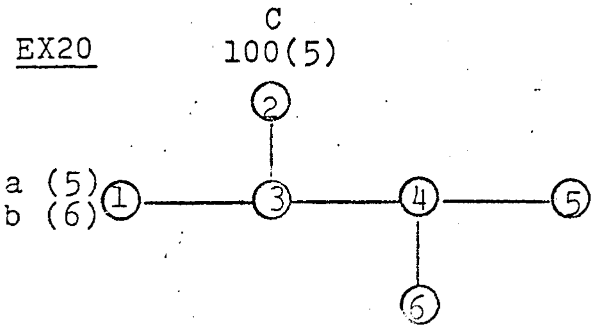
Varied and repeated runs of the simulator indicate that the routing algorithm works well and does indeed maximize bandwidth and provide a rather steady state over time. At this point the limitations on the simulation itself suggest the most significant unknowns. As a next step the algorithm should be implemented in a test network and modified as necessary under real conditions.

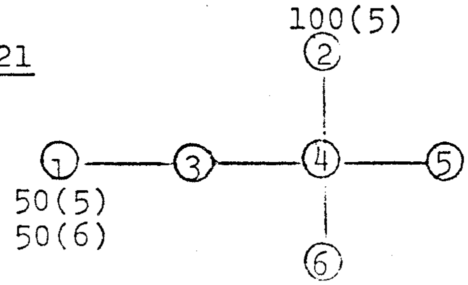
EXAMPLE		RISE TIME	THRUPUT		COMMENTS
		(From Beginning of Arrivals to Ave.)	Ave. (Max.)	Range	
EXØ		22 sec. 1500 ticks	147 KB (150)	146 min. 148 max.	Steady state; fair (slight favoring of traffic→1; from 2→3 has bigger Σ' than 2→1 (long way)
EX2		24 sec. 1600 ticks	94 (100)	93-94	Steady state; sched traffic on longer route low (232) - cut down by fairness, never again best route so can't build up
EX3		24 sec. 1600 ticks	99/100 (100)	97-102	Steady-state; fair; (see data for further breakdown)
EX4		6 sec. 400 ticks	50 (50)	49-50	Steady state; fair; varies slightly back and forth

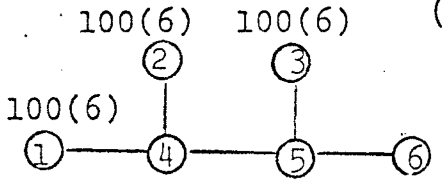
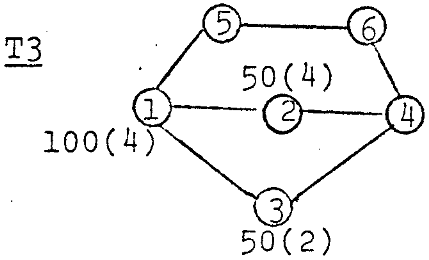
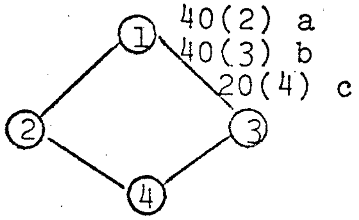
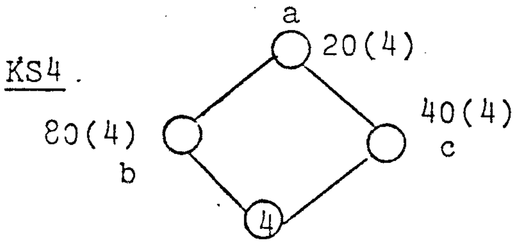
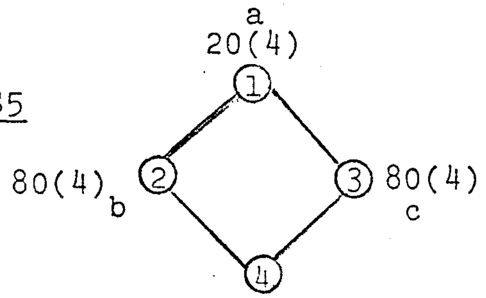
All Runs 2/23/72 RSIM, 57

EXAMPLE	RISE TIME	THRUPUT		COMMENTS
	(From 1st Arrival to Min. Value)	Ave. (Max. Poss.)	Range	
EX5[1] 	16 sec. 1100 ticks	95/96 (100)	90-99	Traffic from 2 ave. 49; traffic from 1 varies somewhat; host metering should help
EX-7 Ⓐ 	21 sec. 1400 ticks	98/99 (100)	96-101	Uses all four lines; traffic shifts around
EX-7 Ⓑ (Run 2/25/72) 	52 sec. 3500 ticks	188 (200)	186-190	Uses all four lines; builds up each to max. in sequence

EXAMPLE	RISE TIME (1st Arrivals to Values)		THRUPUT (for 1st 5000 ave. and last 5000 ave.)					COMMENTS
EX20								Most of these are steady state; all have progression in one direction.
	(suspect)		a	b	c	tot		
A. a = 10 b = 90	12 sec. 900 ticks	1st--- last---	4 1 or 2	44 48	45 48	93 99	min. 90 max. 99	
B. a = 50 b = 50	10 sec. 700 ticks		21 11	27 37	29 39	77 87	min. 73 max. 89	
C. a = 60 b = 40	9 sec. 600 ticks		21 12	27 33	29 38	77 83	min. 70 max. 86	
D. a = 70 b = 30	15 sec. 1000 ticks		22 17	25 25	28 33	75 75	min. 69 max. 81	
E. a = 80 b = 20	7 sec. 500 ticks		23 20	16 19	27 29	66 69	min. 59 max. 75	Steady-state after 5000
F. a = 90 b = 10	7 sec. 500 ticks		25 22	8 9	25 28	58 59	min. 53 max. 65	Reasonably steady-state after 5000
EX21	18+ sec. 1100+ ticks		13 6	29 39	36 44	78 89	min. 71 max. 92	





EXAMPLE	RISE TIME (1st Arr. to Ave.)	THRUPUT	COMMENTS
EX38 	6 sec. 400 ticks	50 (50)	Steady state: distrib 11(1) - 11(2) - 28(3)
T3 	30-36 sec.	range 143-168 ave. 159±9	Steady state with oscillation on order of ±9 from ave. (see data for further) get circular paths
KS3 	31 sec.	range 77-90 after 20,000; ave. 86 a(34); b(33); c(17)	Paths superimpose and flip together - phasing prob; fair but overall low; not using best routes
KS4 	22 sec.	range 92-100 ave. roughly a(16); b(47); c(32)	Substantial circular traffic (see data sheets) oscillates between sources; on ave. fair
KS5 	9 sec.	ave. 98 $\underbrace{a(17); b(33)}_{share}$; c(48)	Steady state; source 3 gets all of one line