MEMORANDUM

To: IMP Guys

From: Steve

Subject: Rolling Routing

Date: December 15, 1972 IG #4

Rolling routing is a routing algorithm which dramatically reduces the time it takes routing information to propagate through the Net. It does not apply to the routing information itself, but only to its propagation; it can easily be used with both the present and proposed routing information. I will attempt to give some idea of how it works and some interesting things that have been learned about it.

The algorithm makes the sending of routing at any one IMP dependent on the arrival of its neighbors' routing. This is in contrast to the present system in which an IMP sends routing irrespective of when it receives routing. The difference in the propagation time of the two algorithms over one line is the present system's routing calculation and sending latencies.

Now for the details. Rolling routing has an IMP send routing out a line when it has received it on all its other lines. The next sending requires new arrivals on the other lines. For example, a three line IMP waits to receive routing on its lines 2 and 3 before it sends on line 1, and then waits to receive again before sending again. Lower and upper bound time controls are imposed to keep routing from rolling out of control and from rolling to a stop. They insure that routing messages are sent at intervals not less than some time, T/throttle, and not greater than another time, T/protect. When the throttle time control is invoked, the IMP must remember to send routing once T/throttle has elapsed. This assures that a gaiting IMP will cause routing to roll, and IMPs will not invoke the protect time control and send old routing. (Needless to say, weird and wontondrous things happen if the lower bound is set greater than the upper bound.) A stub IMP is treated as a special case since it has no other lines to receive routing on, and sends routing whenever it receives routing on its only line i.e. it reflects routing messages.

I found, with the help of a LISP simulation of the present net, that the algorithm sustained itself and rolled routing over the net once every T/throttle. Furthermore, the protect time control was never invoked (I did use it to start the simulation going but once all lines had sent routing, it was never used again.)

Some interesting things have been learned in trying to understand the algorithm's workings. When worrying about starting the simulation, I noticed that the algorithm is stable with all lines sending routing at the same time. In such a state, the throttle time control determines when routing is sent, and routing is sent out at T/throttle intervals. Given that this strongly resembles the present algorithm, rolling routing has no advantage if it can easily slip into such a state. One way to get, and stay out of this state is to have one IMP in the Net (e.g., lowest numbered IMP alive) have a larger throttle time. The rolling throughout the Net comes to wait for this one IMP and not for any other IMPs.

I tried the algorithm out on Will Crowther, and he came up with two interesting observations. One was that the protect time control is invoked when a line or IMP goes down, the exact time when routing does not want to be delayed. We found, however, no reason why the protect time could not be just a little (i.e. maximum expected sum of jitters) greater than the gaiting throttle time.

The other observation was that the propagation of path information is not improved as much as that of line and IMP information. For a path to be found routing must travel every line of the path in order; rolling routing will propagate along worst case (perverted) paths only two lines at a time and along the first and last lines singly. Best case paths, however, are found in one roll of the routing. Rolling routing loses to the present algorithm only for the shortest worst case paths (it will take the gaiting throttle time to propagate over each line whereas the present system will average half that time,) but averages out much better for all paths.

Before I talked with Will, I had intended to preface the presentation with the remark 'Think of this as a clay pigeon flung out to be shot down, remembering that even if you can't, don't, or won't, it will probably break when it comes down to earth.' Will's response was that its likely not to break. Good shooting.