IDN-4

Process Organization

Part 2

Eric Rosen

September 1981

I want to bring up a few more considerations that are relevant to the process structure issues I discussed in my previous message. In my previous message on the gateway process structure, I left out a necessary process, which we might call "Select GO" (SLGO). This is needed to handle the case in which a particular neighboring gateway or destination host can be reached through either of two networks. Thus the Dispatch process would, by looking in the routing table, determine the next gateway to which a transit packet should be sent. The SLGO process would decide which network that packet should use as a Pathway to the chosen neighbor, and would pass the packet to the GO process which corresponds to that network. The GO process would choose a particular access line for that network, and pass the packet to the appropriate PAP Level 3 Output process.

Note that IDP has a similar need to pass packets to SLGO, since a destination host which is directly accessible from this gateway may be accessible from any of several networks to which this gateway is connected.

This hierarchy of processes also has implications for the way delay measurements are done and reported to the routing algorithm, and for the way up/down determinations are made.

The process structure relates to delay measurements as follows. Each GI/GO pair of processes "controls" the I/O to a particular network. The delay to EACH other gateway on that network must be computed separately for each access line to that network. This means, of course, that the GI/GO processes must be able to tell which access line a packet came from. A single value of delay to each neighbor must be computed from the delays on the various access lines. (For example, the single value of delay might just be the minimum over all access lines, or it might be a weighted average.) This generates a value of delay to each neighbor over the network "controlled" by that pair of GO/GI processes. This information must be fed up to some higher level process (SLGO?), which computes a single value of delay to each neighbor, based on the delay values to that neighbor over the individual networks that can be used to reach it. It is this final value that will be put into the routing updates.

If a packet must be sent to a neighboring gateway, and there are several different networks, or several different access lines to the same network, that can be used, how do we decide which one to choose? This is not quite clear yet, which means that we are going to have to be able to experiment with a number of different algorithms, which means that our software has to be properly structured to make easy changes in the selection algorithm. Possible selection criteria are:

1 - Choose the network or access line which provides the least delay. If we use this criterion, we will probably want to report the minimum delay of all the lines and networks as the Pathway delay to that neighbor. (Even though we will be sending all data on one access line to one network, we can use test traffic to measure delays on other access lines or networks.)

2 - Round-robin the traffic among the access lines which

have roughly the same delays (assuming that several lines have comparable delays, but several others have larger delays).

3 - Split the traffic in proportion to the delays.

4 - Use the shortest-delay access line or network until its queues exceed a certain threshold, then use the next shortest-delay line, etc. A variant: use the shortest-delay access line until the number of outstanding packets on it passes a threshold. This variant might be useful if we are dealing with HDLC lines, or in general with lines controlled by a low-level protocol which limits the number of unacknowledged packets that can be sent.

5 - Base the decision on the type of service requested by the user.

Note that these criteria are by no means mutually exclusive; rather we will probably end up using some combination o them, depending on the circumstances or the configuration. Furthermore, some of these criteria require the higher level processes to be aware of the resource utilization of the lower level processes. I think it will be quite a challenge to produce a software system which makes it easy for us to experiment with a number of different algorithms.

On occasion, a destination gateway may have to choose among several access lines or even several networks in deciding how to send a packet to a destination host. We would like to use the same sort of criteria as we use in deciding how to send a packet to a neighboring gateway, except that we don't know the delay to the hosts. In some cases, perhaps the network will tell us its delay to a given host, in which case we can use that as a guide. However, in such a case, we might want to plan for the contingency of the network giving us incorrect information, in which case it has to be easy to change the selection algorithm on the fly.

Just as delay information needs to be passed up the levels of protocol and consolidated into a single quantity (or at least, fewer quantities than we had to begin with), the same is true of up/down information. First, the PAP Level 2 processes must make their up/down determinations for the individual access lines. PAP Level 2 up/down information is generally restricted to saying whether the access line itself is up or down. That is, PAP Level 2 might be able to tell us that NO host or gateway at all is reachable over that access line, but will not be able to tell us that any particular host or gateway is reachable over that access line. Examples of PAP Level 2 up/down might be checking the Ready Line (1822), or performing the VDH line up/down protocol. I assume that HDLC also has something it uses to determine whether a line is up or down. PAP Level 3 up/down might also tell us that no node at all can be reached over the access line. For example, SATNET host access protocol has a line up/down determination at level 3 in addition to the VDH line up/down determination at level 2, and I understand from Dale and Peter that it is not at all unheard of for level 3 to declare the line down while level 2 declares it up. (Apparently this is usually due to some bug in the host software implementing the protocol, which gets into a dead state.) PAP Level 3 might also be able to say that particular destinations are not reachable over this access line. For example, an 1822 "destination dead" message might be received. Both the Level 2 and Level 3 up/down determinations would correspond to what I called "low level up/down protocols" in my IENs. The "high level up/down protocols" would be at the level of the internet protocol, and would be executed by GI/GO and by IDP (for neighboring gateways or hosts, respectively.) Note that Pathway down determinations have to percolate up to higher level processes, so that the GO and SLGO processes do not select networks or access lines which cannot reach a given destination. The GO and SLGO processes must also AND together the down determinations of the processes "beneath", to determine whether a particular destination is completely unreachable. When this determination is made, it must be reflected in a routing update (if there is no longer a Pathway to a particular neighboring gateway), or in the address translation tables (so that DNA messages can be returned for messages that can't be delivered).

One thing we need to decide is which processes really need to be independent entities, and which can be subroutines called by others. (A process is an "independent entity" if it is independently schedulable at a priority which may be different from that of the process which pokes it.) In order to make such decisions, we need to know not only what the ordinary data packet path is, but also what sorts of additional information (like delay measurements and line up/down determinations) ned to be passed among these processes. I find this issue of process structure somewhat confusing.