IEN 104

Minutes of the Fault Isolation Meeting

12 March 1979

Virginia Strazisar

Bolt, Beranek, and Newman

20 March 1979

Minutes of the Fault Isolation Meeting held at BBN on March 12

Attendees:
Virginia Strazisar, BBN, chairman
Peter Sevcik, BBN
Dale McNeill, BBN
Noel Chiappa, MIT
Ray McFarland, DOD
Mike Wingfield, BBN
Jack Haverty, BBN
Bill Plummer, BBN
Mike Brescia, BBN

Ginny suggested that there are three situations in which fault isolation is needed: 1) the user at a terminal on the catenet who cannot reach some destination on the catenet, 2) a catenet control center that must decide what network or gateway in the catenet has failed, and 3) the gateway implementor who must decide what part of the gateway hardware or software has failed. These situations were put forth as a framework for discussing the types of fault isolation facilities that we need. Ginny stated that the object of the meeting was to draw up a list of fault isolation tools needed, giving special consideration to what situations each of these tools would be used in and what questions they could be used to answer. From the suggestions drawn up at the meeting, the detailed formats and protocols could be designed; this level of design was specifically avoided at the meeting.

The first situation discussed was the user at a catenet terminal, who discovers that he either cannot connect to a particular destination host or that he no longer gets any response from his previously working connection. At present no information is passed to the user in either of these cases. Everyone agreed that the user should receive some error reply. It was suggested that the user should receive a response indicating that either 1) the destination host is unreachable, 2) the local gateway or network is unreachable or 3) the catenet is inoperational. Most people agreed that the naive user does not care to know what the catenet problems are in any more detail than this. For example, an error messgage of the form "Can't reach destination network because gateway 3 is down" would be totally useless to the naive user. The user also wants to know when the service will be restored, either "within a short time" such that the user is willing to wait for the service to be restored; or "not for a long time" such that the user will quit trying to use the service at this time. Several people pointed out that a more sophisticated user may want to know exactly what component of the catenet failed. There was some discussion as to whether users should be given access to tools that would enable them to probe the catenet gateways to determine where the failure occurred.

The consensus of opinion was that the user should be given access to such tools, but that no user should be required to use such tools. Our model was that the naive user on receiving an error message would call a network or catenet control center, whereas the more sophisticated user may attempt to track down the problem before contacting the control center. We discussed in more detail what sort of message a gateway could return to the user. It was suggested that if the network returned an error message about a specific host that that error message (text) should be returned verbatim to the user. It was also suggested that error codes be defined for "common" failures, i.e. net down, host down, and that these be included in the error message. It was pointed out that the gateways currently return messages to the source host if they believe (based on their routing information) that the destination network is unreachable. These messages contain the source and destination addresses and the protocol field from the original datagram. Several people pointed out that this information is insufficient to return an error message to the source user and that the entire internet header of the original datagram should be returned in the error message. We discussed the problem of what to do in the case where datagrams are lost in a gateway or network in such a manner that no error message is generated and returned to the source. It was decided in this case that the source host should automatically probe the gateways in order to return a reasonable status message to the user. It was assumed that the user is running a program that implements some type of internet protocol, such as TCP, and that that program is capable of detecting long delays or mutiple retransmisssions and of generating some type of probe packet to attempt to track down the failure when this occurred. These probe packets are discussed in more detail below. Information obtained from such probing could also be sent to a monitoring center.

We discussed the concept of a monitoring or control center. The primary purpose of a monitoring or control center in terms of fault isolation is to isolate the component (network or gateway) that failed and to notify the proper authority to have it fixed. We felt that a control center was needed to avoid having all the users in the catenet calling any and all implementors they felt might be responsible for problems. The concept of a single control center was discussed and rejected for both technical and political reasons. From the technical point of view, it was pointed out that the catenet could become partitioned such that the control center was cut off from part of the catenet and thus could no longer handle faults in that portion of the catenet. On the political side, it was pointed out that organizations responsible for the individual networks may be unwilling to support one control center run by one organization. We agreed that the catenet control center should actually be multiple control centers. These could be either the existing network

control centers working in co-operation or separate catenet control centers, each of which was established by co-operating network groups. Tools that these control centers would need included a facility to probe gateways to determine why a particular destination was unreachable.

We elaborated slightly on the design of a facility for probing gateways. A host or control center sends its local gateway a message saying "poll the gateways in the catenet to determine why I can not get to destination X". The gateway then polls its neighbors, its neighbors' neighbors, etc., extracting routing tables, addresses of neighbor gateways, status of neighbor gateways and networks, etc. to determine why the destination is unreachable. The gateway would then formulate a response to the host; this response would be of the form: "the network connection between gateway 3 and net 2 is down", "gateway 5 and gateway 6 are down", etc. This mechansim would be an extension of the gateway-gateway protocol as defined in IEN #30. This probe facility would be used by the source host to generate a message to the user in the case where no response is recieved from the destination and no error message is returned by the gateways. The facility would also be used by catenet control centers to isolate the componenet of the catenet that has failed.

It was pointed out that we should be concerned not only with total failures, but also with system performance, especially delay. In this context, we were not concerned with cases where delay seemed slightly longer than usual, but rather cases in which traffic crossed the catenet with extrememly high delays, i.e several minutes. A facility was suggested to track this sort of problem: generate a packet from source A addressed to destination B; have this packet trace its route and timestamp it at each gateway on the route to B; at B, echo the packet; return the packet to the source, A, using source routing and the route stored in the packet via the trace mechanism; timestamp the packet on its route back to A. The timestamps in the packet could now be interpreted to yield transit times across each network as there would be a pair of timestamps for each gateway traversed.

The final stage of fault isolation is the situation in which the failure has been attributed to a particular gateway and the implementor of that gateway must debug it. This part of fault isolation was not discussed in detail. It was suggested that at this point, it would be very useful to be able to turn off timeouts in the catenet to avoid having the state of the catenet change in such a way that the problem can no longer be isolated.

In summary, the following list of tools and situations in which they would be used was suggested.

1) Error messages indicating whether the destination host, the local network or gateway, or the catenet had failed, and indicating the time at which service should be restored.

These are to be returned automatically to the catenet user whenever there is a failure in using a catenet service.

2) Gateway to gateway probing mechanism that can be initiated with a host to gateway message.

This mechanism would be used by a control center to isolate a component failure. It would also be available to the user. It would be used by source host protocol programs to formulate an error message for the user when no repsonse was received from the destination and no error message was received from the gateways.

3) Ability to trace, echo and source route packet with timestamping.

This facility would be used to determine where delays are occurring when a destination is reachable, but delays cannot be accounted for.

4) Ability to echo packets off any gateway.
5) Ability to trace packets.
6) Ability to source route packets.
7) Ability to dump gateway tables.
8) Ability to trace packets by sending replies from every gateway that handles the packet.

These capabilities would be used by control centers and gateway implementors to isolate failed components and determine the reasons for failure. These facilities were not discussed in detail. A description of mechanisms for tracing packets and source routing packets was given in IEN #30, although these have not yet been implemented.

The next step in developing fault isolation mechanisms for the catenet is to work out the detailed design for the mechanisms suggested above, and to implement these in hosts, gateways and control centers.