ISSUES INVOLVING NON-ROUTING GATEWAYS

B. Bowman & P. Cook

Ford Aerospace & Communications Corporation

Colorado Springs, Colorado

(303-471-9110)

IEN # 196

September 11, 1981

INTRODUCTION

This note presents some proposed modifications to the gateway
routing algorithm and the protocol for exchanging routing
information described in IEN #109, "How To Build A Gateway". The
described algorithm modifications were defined as a result of

problems encountered in the implementation of the gateway to gateway control protocol specified in IEN # 109. The problems encountered in implementing the algorithm involve the mechanisms for incorporating non-routing gateways in the catenet. Therefore, this discussion focuses on modifications designed to facilitate use of non-routing gateways in concert with routing gateways in the catenet.

As indicated above, the issues discussed resulted from our implementation of the IEN # 109 algorithm. Some initial implementation questions concerning interpretation of the rules for assembling routing updates from non-routing gateways led to our analysis of the design of this routing strategy. From routing strategy design issues, our study progressed to questions regarding the intended/optimal role of non-routing gateways in the catenet.

As a result, this note presents specific modifications to IEN # 109 necessary for successful implementation and also includes discussion of the broader issues of the role of the non-routing gateway in the catenet.

Finally, we propose a modification to the routing information exchanged in the IEN # 109 algorithm, which we believe has significant advantages in reducing the logical complexity of the algorithm and traffic between gateways. Although we have looked at design approaches of other authors in this field, we have not done an exhaustive search to determine the "novelty" of our approach. We would also like to stress that our goal was not to design a new routing algorithm but simply to implement the IEN # 109 algorithm. In a very real sense, it can be said that we backed into these routing design issues as a natural consequence to our implementation efforts.

I. Non-Routing Gateways

It should be noted that any solution must be based on an understanding regarding the intended role of the non-routing gateway in the catenet. Unambiguous rules for incorporating non-routing gateways in the routing scheme can only be specified when the intended role is clearly defined. An optimal solution, then, would involve clarification of the role of the non-routing gateway followed by revision of the routing specification consistent with the defined role.

-2-

IEN # 109, "How to Build a Gateway", contains a section (Sect.9, Page 7) discussing the interface between routing gateways and non-routing gateways and, in particular, discussing the use of a non-routing gateway by a routing gateway for routing packets. Two areas of this section are problematic with respect to a successful implementation. These are, (1) initialization with respect to the non-routing gateway and (2) computation of the minimum distance vector.

A. Initialization

IEN # 109 gives the following instructions for initializing with respect to a non-routing neighbor gateway.

 * "For each non-routing neighbor gateway of gateway G1, compute the routing update that would be sent to G1 assuming that all

gateways and network connections are operational. These routing
updates are assembled in G1."

This rule can be interpreted in one of two ways. The first is to
assemble the non-routing gateway's routing update showing actual
distances to each network from the non-routing gateway. Figure 1
shows an example of this interpretation.

In this example, Gateway (B) is a non-routing gateway and Gateway
(A) is a routing gateway.   Initially, G(A) assembles G(B)'s
update as actual distances from G(B) to the 3 networks. Based on
this initialization, G(A) assumes that it is a distance of 1 from
Net.#1 because the non-routing gateway provides the only path to
Net.#1.   G(A) also assumes that it is directly (0) connected to
Nets.#2 and #3. G(A) builds its initial minimum distance vector
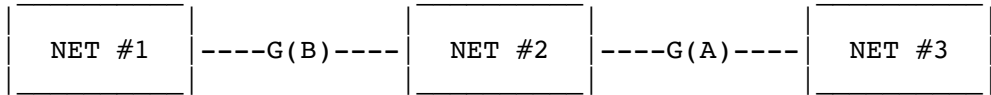reflecting these distances.

Assume, then, that G(A)'s link to Net.#3 goes down at some time.
When this happens, G(A) recomputes it's minimum distance vector.
Since G(A) is no longer connected to Net.#3, there is no path
available unless the non-routing gateway's path is used.   G(B)'s
update indicates a distance of 1 to Net.#3. Therefore, G(A)
assumes a path to Net.#3 of distance 2 through G(B). Of course,
there is no path to Net.#3 but G(B) does not accept or transmit
routing tables so any packets addressed to Net.#3 will shuttle
indefinitely (until the IP4 time to live field is decremented to
zero) from G(A) to G(B) to G(A), etc.   When this example is
extended to larger catenets, the shuttling problem persists and
simply involves higher distance shuttles.   Therefore, this
interpretation of the initialization instructions is unworkable.

                                -3-

        FIGURE 1:   INITIALIZATION OF NON-ROUTING GATEWAY UPDATE
                     "ACTUAL DISTANCE" METHOD

CATENET CONFIGURATION

 _____   _____   _____
| | | | | |
| NET #1 |----G(B)----| NET #2 |----G(A)----| NET #3 |
|_____| |_____| |_____|

where
    G(A) = Gateway A: Routing Gateway
    G(B) = Gateway B: Non-Routing Gateway

G(A) DISTANCE MATRIX AFTER INITIALIZATION

                          NETWORKS
      GATEWAYS   1   2   3

         A   1   0   0
         B   0   0   1

G(A) DISTANCE MATRIX (ASSUMING LOSS OF CONNECTIVITY TO NET.#3)

| | NETWORKS | | |
GATEWAYS	1	2	3
A	1	0	2
B	0	0	1

The other possible interpretation of the IEN # 109 instructions is to assemble the non-routing gateway's update showing actual distances to each network from the non-routing gateway only when the non-routing gateway is as close to or closer to the network than the gateway assembling the routing update. This interpretation incorporates a rule from Section 7, page 7 regarding computation of routing updates. Figure 2 shows an example of this interpretation.

In this example, Gateway (B is a non-routing gateway and Gateways A and C are routing gateways. The example is presented from the perspective of Gateway A. Gateway B's update is initialized showing 0 distance to Nets.#1 and #2. A distance of 1 is shown to Net.#3 because Gateway B is as close to Net.#3 as is Gateway A. An infinite distance is shown to Net.#4 because Gateway A is closer to Net.#4 than Gateway B. The example shows receipt of a routing update from Gateway C which shows Gateway C's distances calculated according to the rules for preparation of a routing update.

Gateway A calculates its minimym distance vector as 0 distance from Nets.#2 and #4 because they are directly connected, distance 1 from Net.#1 because of the path through Gateway B and distance 1 from Net.#3 because of the path through Gateway C.

Assuming that Gateway C loses connectivity to Network #3 at some time, Gateway C would send Gateway A a new routing update showing infinite distance to Network #3. Gateway A would then recalculate its minimum distance vector. G(A) would assume that the only path to Net.#3 is through the non-routing gateway (G(B)) and that this is a path of distance 2. This occurs because G(B) neither accepts nor transmits routing updates so it is unaware of the connectivity loss and G(A) is assembled with G(B)'s routing update so there is no provision for changes to this information in G(A). The situation is complicated by the fact that G(A), upon recalculation of its minimum distance vector, will transmit a routing update to G(C) showing the distance of 2 from G(A) to Network #3. G(C) will recompute its distance vector to show a path of distance 3 from G(C) to Network #3. Since no path actually exists and all 3 Gateways believe a path exists, packets addressed to Net.#3 will shuttle indefinitely (until the IP4 time to live field is decremented to zero).
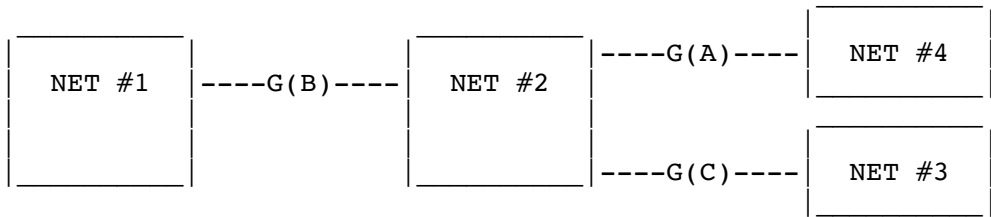
Thus it seems that Thus it seems that both interpretations of the IEN # 109 instruction for assembling the non-routing gateway's

update present serious problems.

-5-

FIGURE 2:   INITIALIZATION OF NON-ROUTING GATEWAY UPDATE
             "TAILORED UPDATE" METHOD

CATENET CONFIGURATION

```
                                                     _____
 _____                      _____          |         |
|         |                    |         |----G(A)----| NET #4  |
|  NET #1 |----G(B)----|  NET #2 |         |_____|
|         |                    |         |           _____
|         |                    |         |          |         |
|_____|                    |_____|----G(C)----| NET #3  |
                                                     |_____|
```


where
   G(A) = Gateway A: Routing Gateway
   G(B) = Gateway B: Non-Routing Gateway
   G(C) = Gateway C: Routing Gateway

G(A) DISTANCE MATRIX AFTER INITIALIZATION

	NETWORKS			
GATEWAYS	1	2	3	4
A	1	0	1	0
B	0	0	1	INF
C	1	0	0	INF

G(A) DISTANCE MATRIX (ASSUMING LOSS OF CONNECTIVITY TO NET.#3)

	NETWORKS			
GATEWAYS	1	2	3	4
A	1	0	2	0
B	0	0	1	INF
C	1	0	INF	INF

-6-

B. Computation of the Minimum Distance Vector

IEN # 109 goes on to provide the following instructions for
calculating the minimum distance vector when non-routing gateways
are included.

  * "The gateway, G(A), first computes its minimum distance
     vector using only the routing updates from neighbors that
     participate in routing. If the minimum distance to any
     network is infinity, i.e., the network is unreachable via
     any of the routing gateways, then the minimum distance to
     that network is re-computed using the routing update
     compiled for the non-routing neighbor gateway."

This rule appears inefficient if we look at the example shown in
Figure 3. In this example, Gateway B is a non-routing gateway
and Gateways A and C are routing gateways. The example is
presented from the perspective of Gateway A and the focus of the
example is on the distance to Network #1.

G(A) is assembled with the update of G(B) (the non-routing
gateway) and G(B) shows a distance of 0 from Net.#1 since it is
directly connected. In our example, G(A) has received an update
from G(C). G(C) shows a distance of 1 to Net.#1 based on its
path through G(B). According to the rule, G(A) has to prefer the
route through G(C) since G(C) is a routing neighbor and G(B) is
non-routing. Therefore, G(A) assumes a path of distance 2 to
Net.#1 and G(A) will route to G(C) any packets addressed to
Net.#1. On receipt of the packet addressed to Net.#1, G(C) will
route the packet to G(B) where it will be delivered to Net.#1.

This solution is not unworkable but it does cause delay and
inefficiency by forcing an additonal transfer with respect to the
optimal path. In a geographically dispersed network, this delay
and additional network traffic could become significant.

-7-

    FIGURE 3:   COMPUTATION OF THE MINIMUM DISTANCE VECTOR

CATENET CONFIGURATION

```
                                               _____
                                              |          |
 _____              _____  |----G(A)----| NET #4 |
|          |            |          | |       |          |
| NET #1   |----G(B)----| NET #2   | |        |_____|
|          |            |          | |        _____
|          |            |          | |       |          |
|_____|            |_____| |----G(C)----| NET #3 |
                                              |          |
                                              |_____|
```


where
   G(A) = Gateway A: Routing Gateway
   G(B) = Gateway B: Non-Routing Gateway
   G(C) = Gateway C: Routing Gateway

G(A) DISTANCE MATRIX

```
                        NETWORKS
     GATEWAYS           1    2    3    4


         A              2    0    1    0
         B              0    0    1    INF
         C              1    0    0    1
```

C. Proposed Solution to Non-Routing Gateway Problems

Role Clarification

As indicated above, a solution to the problems in incorporation
of non-routing gateways must address clarification of the role of
the non-routing gateway in the catenet. Three possible roles for
the non-routing gateway have been identified and are discussed in
this section. In general, we see potential for non-routing
gateways to be used to provide 1) the "only" route, 2)
"redundant" routes or 3) "parallel" routes. Use of the
non-routing gateway to provide "redundant" and "parallel" routes
is discussed in Section D.

Figure 4 illustrates use of the non-routing gateway to provide
the "only" route. This role basically restricts the use of
non-routing gateways to allow (i.e., recognize) them in the
catenet only when the non-routing gateway provides the "only"
connection to a network or network branches.

Although other possible roles will be discussed , this particular
role definition is the one we recommend. Using non-routing
gateways where they provide the "only" possible route seems the
most logical use of these gateways and also lends itself most
easily to unambiguous rule definition.   The mechanisms for
incorporating non-routing gateways consistent with the "only"
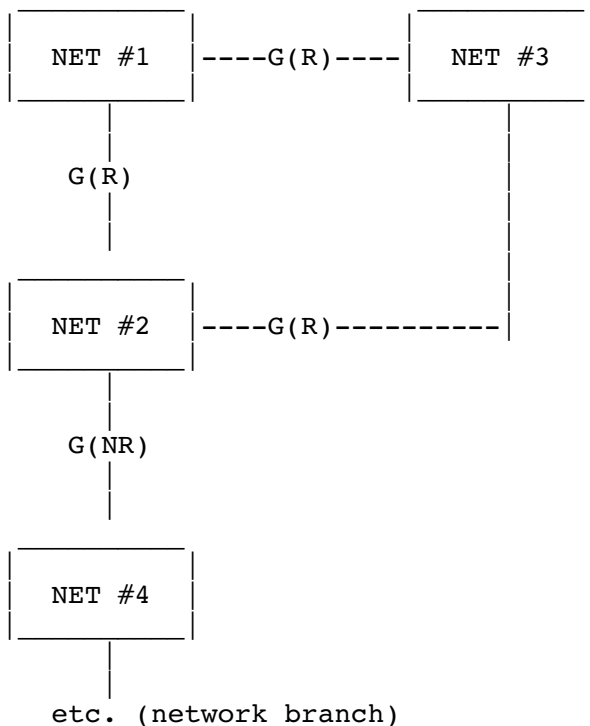route role are described in the following paragraphs.

```
        FIGURE 4:    NON-ROUTING GATEWAY AS "ONLY" ROUTE


CATENET CONFIGURATION


 _____                    _____
|                |                  |                |
|    NET #1      |----G(R)----|    NET #3       |
|_____|                  |_____|
        |                                   |
        |                                   |
      G(R)                                  |
        |                                   |
        |                                   |
 _____                           |
|                |                          |
|    NET #2      |----G(R)----------|
|_____|
        |
        |
      G(NR)
        |
        |
 _____
|                |
|    NET #4      |
|_____|
        |
        |
      etc. (network branch)



      G(R) = Routing Gateway
      G(NR) = Non-Routing Gateway


NOTE:    In this configuration, the non-routing gateway provides
         the only connection to Network #4.
```

Rules for "Only" Route Role

It appears that the initialization and route computation problems
associated with non-routing neighbor gateways described above,
could be solved by a few rule changes. The proposed new rules
are as follows:

 * IEN # 109 Rule:

    For   each non-routing neighbor gateway of gatewayG(A),
    compute the routing update that would be sent to G(A)
    assuming that all gateways and network connections are
    operational. These routing updates are assembled in G(A).

New Rule:

   F or each non-routing neighbor gateway of gateway G(A),
   compute the routing update that would be sent to G(A)
   assuming that all gateways and network connections are
   operational. In computing these routing updates, show
   actual (non-infinite) distances to networks which are
   reachable only through the non-routing gateway. Networks
   which are reachable by a routing gateway shall be
   initialized with distance equal infinity. These routing
   updates are assembled in G(A).

   IEN # 109 Rule:

   T he gateway, G(A), first computes its minimum distance
   vector using only the routing updates from neighbors that
   participate in routing. If the minimum distance to any
   network is infinity, (i.e., the network is unreachable via
   any of the routing gateways), then the minimum distance to
   that network is re-computed using the routing update
   compiled for the non-routing neighbor gateway.

   New Rule:

   The gateway, G(A), first computes its minimum distance
   vector using all the routing updates (including updates from
   both routing and non-routing gateways).

These proposed rule changes, taken together, cause non-routing
gateways to be used only when they provide the only path to a
network or string of networks. These rules also eliminate the
possibility of selecting any other path (i.e., through a routing
gateway) when a non-routing gateway provides the only path.
Thus, non-routing gateways are used for packet routing if and
only if they provide the only route to a destination.

Figure 5 shows how the new rules solve the problems identified in
Examples 1, 2, and 3. In this example, Gateway B is a
non-routing Gateway and Gateways A and C are routing gateways.
The example is presented from the perspective of Gateway A. The
G(A) Distance Matrix after initialization shows that the problem
depicted in Figure 3 (computation of the minimum distance vector)
is resolved through application of the new rules. Since the
minimum distance vector is calculated based equally on the
routing and non-routing gateway updates, the shortest path (and
only path) to Network #1 is selected by both routing gateways
(G(A) and G(C)).

The G(A) Distance Matrix resulting from loss of connectivity to
Net.#4 shows that the problem depicted in Figure 1
(Initialization of non-routing gateway update) is resolved
through use of the new rules. Since the assembled non-routing
gateway update no longer shows paths to networks reachable by
routing gateways, G(A) is able to determine that Net.#4 is now
unreachable and does not attempt to route traffic to Net.#4
through Gateway (B).

The G(A) Distance Matrix resulting from loss of connectivity to
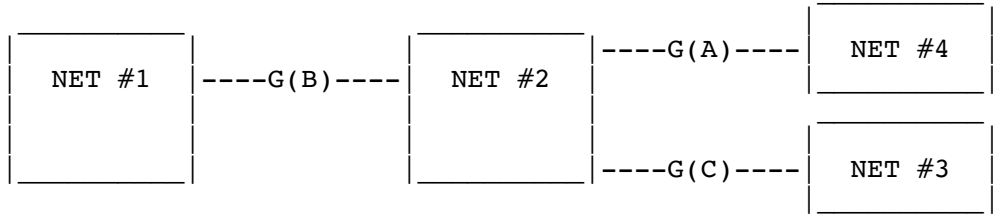Net.#3 shows that the problem depicted in Figure 2

(Initialization of non-routing Gateway update) is also resolved
through application of the new rules. This problem is resolved
because the assembled non-routing gateway update no longer show
paths to networks reachable by routing gateways. Thus, when
Net.#3 is declared unreachable by Gateway C, G(A) determines that
Net.#3 is unreachable by all gateways.

FIGURE 5:   PROPOSED SOLUTION

CATENET CONFIGURATION

```
 _____                      _____ |----G(A)----|  _____
|          |                    |          ||            | |          |
|  NET #1  |----G(B)----|  NET #2  |            |  NET #4  |
|          |                    |          ||            | |_____|
|          |                    |          ||
|_____|                    |_____||----G(C)----|  _____
                                                         | |          |
                                                         | |  NET #3  |
                                                         | |_____|
```

where

    G(A) = Gateway A: Routing Gateway
    G(B) = Gateway B: Non-Routing Gateway
    G(C) = Gateway C: Routing Gateway

G(A) DISTANCE MATRIX AFTER INITIALIZATION

	NETWORKS			
GATEWAYS	1	2	3	4
A	1	0	1	0
B	0	INF	INF	INF
C	1	0	0	INF

G(A) DISTANCE MATRIX (ASSUMING LOSS OF CONNECTIVITY TO NET.#4)

	NETWORKS			
GATEWAYS	1	2	3	4
A	1	0	1	INF
B	0	INF	INF	INF
C	1	0	0	INF

G(A) DISTANCE MATRIX (ASSUMING LOSS OF CONNECTIVITY TO NET.#3)

```
                        NETWORKS
      GATEWAYS           1    2    3    4


         A               1    0   INF   0
         B               0   INF  INF  INF
         C               1    0   INF  INF



                        -13-
```

D. Other Possible Approaches

Redundant Role

Restricting the use of non-routing gateways in the catenet to
locations where they provide the only configured path and
adoption of the corresponding rule changes is our recommended
solution to the implementation problems we've identified. In
reviewing IEN # 109 and it's predecessor IEN # 30, we find it
difficult to determine the intended role of the non-routing
gateways.

Both papers indicate that non-routing gateways are to be used to
forward traffic only when they provide the only route to a
network.   This seems to indicate that the initial intent was to
use non-routing gateways in the role defined above (i.e.   "only"
route role).   On the other hand, it's possible that the authors
actually intended to use the non-routing gateways to provide the
only configured path and to provide backup routes to networks
connected by routing gateways.   This role for non-routing
gateways is what we have called the "redundant " role. Figure 6
illustrates this use of non-routing gateways.   This role is
basically an addition to the "only" route role.

Here the non-routing gateway can also be configured in parallel
with a routing gateway but the non-routing path is only used if
connectivity through the routing gateway is lost.   Thus, the
non-routing gateway is used as a backup for the routing gateway.
As long as the routing gateway and its connections   are
functional, the non-routing gateway is not used to forward any
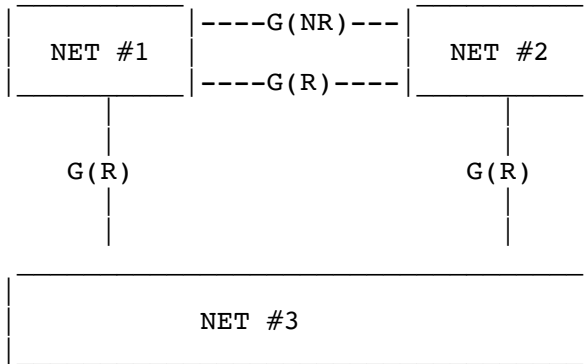traffic.

749/

                        -14-

FIGURE 6:   NON-ROUTING GATEWAY AS "REDUNDANT" ROUTE

CATENET CONFIGURATION


```
 _____                   _____
|          |----G(NR)---|    |          |
|  NET #1  |            |    |  NET #2  |
|_____|----G(R)----|    |_____|
      |                            |
      |                            |
    G(R)                         G(R)
      |                            |
      |                            |
 _____
|                                    |
|               NET #3               |
|_____|
```


WHERE
   G(R) = Routing Gateway
   G(NR) = Non-Routing Gateway

It seems that this second role interpretation is implementable.
Basically, the corresponding rules would be:

 1) Assemble routing update showing direct network connections
     and actual distances to networks reachable only by the
     non-routing gateway.

 2) Calculate minimum distance vector using non-routing gateway
     paths only when they provide the only route.

Implementing   this   role   concept, then, involves the added
complexity of the two stage minimum distance vector calculation
(with bias toward routing gateway paths). While this may be the
role anticipated by the authors of IEN # 109 and # 30, we feel
the backup role is of little added value for the additional
complexity involved. It seems unlikely to us that networks would
buy a processor which was to be idle the majority of the time.
Instead, if need existed for a backup, it seems a routing gateway
would be a better choice and cost efficient as it could increase
bandwidth as well as providing redundancy.

Parallel Role

At the other end of the spectrum, it is possible that some group might wish to use non-routing gateways in any configuration where routing gateways can be used. In this role, which we call the "parallel" role, non-routing gateways are used to provide additional (parallel) paths and thus effectively increase bandwidth. This role differs from the "redundant" route role in that non-routing gateways are used to forward traffic even when they do not provide the only path. It is this facility which allows the effective increase in real-time bandwidth.

Whether this intended role is implementable is an open question. Obviously, it would be more complex to implement and would basically require a redefinition of some of the ground rules. We rejected this possible role application because it significantly reduces catenet reliability and seems to defeat the whole idea behind the routing algorithm function. If non-routing gateways are used routinely to forward traffic in parallel with routing gateways, connectivity changes occuring with the non-routing gateways are unreported and therefore the non-routing gateways could easily become traffic sinks.

We felt that the only possibility for using non-routing gateways to increase bandwidth would be to implement the role outside the gateway-gateway control protocol. This function could be implemented as a host function where there was an understanding between hosts on two networks that traffic would be split on some basis between routing and non-routing gateways. Thus, the decision to sacrifice catenet reliability in favor of increased traffic flow would be made by the parties involved without involving any gateway control function awareness or participation.

II. Routing Update Calculations

In the process of designing the gateway routing software IAW IEN # 109, we determined that a subset of the routing algorithm rules dictated a logically cumbersome implementation. These rules, from IEN # 109 are as follows:

  A. "The gateway reports its distance to a network to a
      neighbor only if it is as close or closer to a network than
      its neighbor." (p.7)

  B. "The gateway maintains a copy of the most recent routing
      updates that it sent to each of its neighbors. The gateway
      computes the new routing updates to send to each of its
      neighbors. If any of these routing updates are different
      than the preceding updates, then the gateway sends new
      routing updates to its neighbors." (p.7)

Rule A calls for "tailoring" routing updates for each neighbor such that the transmitting gateway reports its distance relative to the connectivity of each recipient gateway. Thus, when a gateway calculates its routing update, it first calculates its minimum distance vector and then compares that vector to the last routing update from each neighbor. From this comparison, it

creates the "tailored" routing update for each neighbor which reflects not only the transmitting gateway's connectivity but also the connectivity of each neighbor. The comparison process is performed as follows:

```
        DO (For each neighbor)

            DO (For each network)

                IF (Transmitting gateway is as close or closer to

                    the network)

                      Report actual distance

                ELSE (Report infinity distance)

            ENDDO

        ENDDO
```

Rule B requires that, having performed the calculations based on Rule A resulting in routing updates for each neighbor, these updates be compared to the updates last sent to each neighbor. If any of the updates are different from those last sent, all the updates are transmitted. If there are no changes to any neighbor since the last transmittal, no updates are sent. Implementation of this rule requires that copies of the last updates sent be maintained at the gateway.

The basis for these rules, particularly Rule A, is prevention of shuttling which could occur if only one path existed to a network and that network became disconnected. If gateways were allowed to report actual distances rather than infinity, then two gateways could, because of their dependent or relative distance relationship, begin step increments of their distances to the disconnected network eventually approaching infinity and thus shuttle packets until the IP4 time to live field is decremented to zero.

This shuttling problem is, in fact, a natural outcome of a routing scheme which involves the exchange of distance information only. Thus, the problem can be solved by the addition of special rules such as Rule #1 above or can be solved by going to a routing scheme which exchanges connectivity information (distance and path).

After studying the current solution of exchanging relative
distance vectors, it seems that going to a routing scheme
involving exchange of both connectivity information and distance
provides a simpler and more efficient solution. The added
information exchange proposed is more than offset by the
computational simplicity and the reduction in routing update
traffic afforded.

The routing scheme proposed provides for including in routing
updates both the actual distance vector (as opposed to the
current tailored relative distance vector) and the routing table.

Figure 7 is a configuration used as the basis for illustration of
the updates.   To see how these routing updates are constructed,
examine the update from G3 in Figure 8 (Step 3). G3 is directly
connected to Nets.#2 and #26, so G3 shows a distance of 0 to each
with G3 as the "next" gateway address for routing. G3 is 1 away
from Net.#1 and this path to Net.#1 is through G2. Similarly, G3
is 1 away from Nets.#3 and #27 and these paths are through G5 and
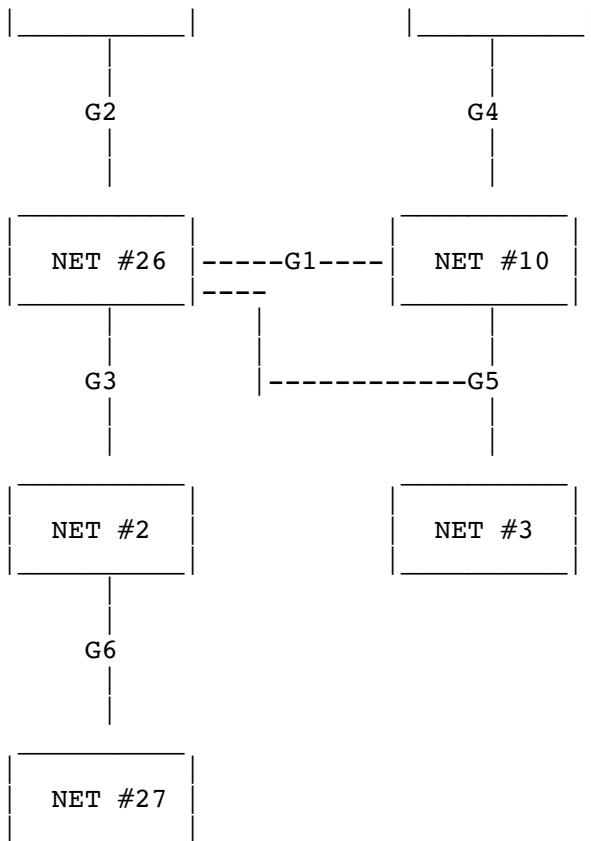G6 respectiviely. G3 is also a distance of 1 from Net.#10.

        FIGURE 7:   CATENET CONFIGURATION

 _____   _____
| | | |
| NET #1 | | NET #4 |

```
 |_____|              |_____|
      |                        |
      |                        |
     G2                       G4
      |                        |
      |                        |
 _____                _____
|         |              |         |
| NET #26 |-----G1----|  | NET #10 |
|_____|----        | |_____|
      |      |           |    |
      |      |           |    |
     G3      |-----------G5   |
      |                       |
      |                       |
 _____                _____
|         |              |         |
| NET #2  |              | NET #3  |
|_____|              |_____|
      |
      |
     G6
      |
      |
 _____
|         |
| NET #27 |
|_____|
```

NOTE: Gateways G1 - G6 are all routing gateways.

FIGURE 8:   ROUTING UPDATES BASED ON NEW ROUTING METHOD
            (SEE FIGURE 7 FOR CONFIGURATION)

NETWORK DISTANCES/"NEXT" NEIGHBOR ADDRESS(ES)

STEP	SOURCE	DEST.		1	2	3	4	10	26	27
1.	G1	ALL	Distance	INF	INF	INF	INF	0	0	INF
			"Next"					G1	G1	
2.	G2	ALL	Distance	0	1	1	2	1	0	2
			"Next"	G2	G3	G5	G1	G1	G2	G3
3.	G2	ALL	Distance	1	0	1	2	1	0	1
			"Next"	G2	G3	G5	G1	G1	G3	G6
							G5	G5		

4.	G4	ALL	Distance	2	2	1	0	0	1	3
			"Next"	G1	G1	G5	G4	G4	G1	G1
				G5	G5				G5	G5
5.	G5(10)	ALL	Distance	1	1	0	1	0	0	2
			"Next"	G2	G3	G5	G4	G5	G5	G3
6.	G5(26)	ALL	Distance	1	1	0	1	0	0	2
			"Next"	G2	G3	G5	G4	G5	G5	G3
7.	G1	ALL	Distance	1	1	1	1	0	0	2
			"Next"	G2	G3	G5	G4	G1	G1	G3

Assume loss of connectivity by G2 to Net.#1

8.	G2	ALL	Distance	INF	1	1	2	1	0	2
			"Next"		G3	G5	G5	G5	G2	G3
							G1	G1		
9.	G1	ALL	Distance	INF	1	1	1	0	0	2
			"Next"		G3	G5	G4	G1	G1	G3

NOTE: G5(10) is gateway 5 on network 10. A gateway which is a
         neighbor on two networks is treated as two gateways. Thus
         G5 is identified as G5(10) and G5(26) by Gateway 1.

-21-

FIGURE 9:   ROUTING UPDATES BASED ON IEN # 109 METHOD
            (SEE FIGURE 7 FOR CONFIGURATION)

NETWORK DISTANCES

SOURCE	DEST.	1	2	3	4	10	26	27
G1	ALL	INF	INF	INF	INF	0	0	INF
G2	G1	0	1	1	2	1	0	2
G1	G2	INF	INF	INF	INF	0	0	INF
G1	G3	1	2	2	3	0	0	3
G1	G4	1	2	2	3	0	0	3
G1	G5(10)	1	2	2	3	0	0	3
G1	G5(26)	1	2	2	3	0	0	3
G3	G1	1	0	1	2	INF	0	1
G1	G2	INF	1	INF	INF	0	0	2
G1	G3	1	INF	INF	2	0	0	INF
G1	G4	1	1	2	2	0	0	2
G1	G5(10)	1	1	2	2	0	0	2
G1	G5(26)	1	1	2	2	0	0	2
G4	G1	INF	INF	1	0	0	INF	INF
G1	G2	INF	1	INF	1	0	0	2
G1	G3	1	INF	INF	1	0	0	INF
G1	G4	1	1	INF	INF	0	0	2
G1	G5(10)	1	1	2	1	0	0	2
G1	G5(26)	1	1	2	1	0	0	2

```
G5(10)      G1         1     1     0     1     0     0     2
G5(26)      G1         1     1     0     1     0     0     2
G1          G2        INF    1     1     1     0     0     2
G1          G3         1    INF    1     1     0     0    INF
G1          G4         1     1     1    INF    0     0     2
G1          G5(10)     1     1    INF    1     0     0     2
G1          G5(26)     1     1    INF    1     0     0     2
          Assume  loss  of  connectivity  by  G2  to  Net.#1.
G2          G1        INF    1     1    INF   INF    0     2
G1          G2         7     1     1     1     0     0     2
G1          G3        INF   INF    1     1     0     0    INF
G1          G4        INF    1     1    INF    0     0     2
G1          G5(10)    INF    1    INF    1     0     0     2
G1          G5(26)    INF    1    INF    1     0     0     2
```

However, in this case, G3 has two paths of distance 1 through
which it can reach Net.#10.   G3 can go through G1 to reach
Net.#10 in 1 hop or it can go through G5. Since both paths are
equal and minimum distance, both are reported in the routing
update.

When routing updates are constructed to include both distance and
connectivity information as described above, gateways receive
enough information to perform path checks.   This capability
enables gateways to avoid shuttling problems.

The performance of path checks is illustrated in Figure 7.   In
Figure 8, step 8, G2 sends a routing update which shows that G2
has lost connectivity to Net.#1 (i.e., distance =inf.). G1 then
recomputes its minimum distance and routing with respect to
Net.#1. In these calculations G1 wants to find the shortest path
but also wants to ensure that it is a "true" path. Therefore, G1
looks at the first path of distance 1. This path is provided by
G3.   However, G3's "next" neighbor address on this path is G2.
G2 now shows a path of infinity to Net.#1, so G1 rejects the G3
route. In examining the other routes of distance 1 (i.e., G5(10)
and G5(26)), G1 finds that they all go through G2 which G1 knows
is disconnected. Thus, all the paths of distance 1 are rejected.

Next, G1 looks at the route of distance 2 provided by G4.   G4
shows two "next" addresses, G1 and G5, G1 rejects the path
through "next" address G1 since this path would obviously result
in shuttling.   G1 then traces the route through "next" address
G5. Looking at the routing update provided by G5, G1 sees that
G5 intends to route traffic addressed to Net.#1 through G2. G2
has already been identified as a dead-end. As G1 has examined
all known routes to Net.#1 and found no acceptable route, G1
determines that it is infinity distance to Net.#1 and sends a
routing update to all its neighbors showing infinity distance to
Net.#1 (step 9).

The rules involved in the routing scheme described in the example
plus a few additional rules of the proposed routing scheme are
defined as follows:

  A. On transmission of routing updates, the gateway reports its
      actual   distance (minimum distance vector) from each
      network. In addition, for each network, the gateway
      reports the "next" gateway address through which it will
      route any packets addressed to the network. If more than

one path of the same minimum distance exists, the gateway
will report all possible "next" gateway addresses
associated with that network. In IEN # 109 terminology,
then, each gateway exchanges its "minimum distance vector"
and its "routing table (list of neighbor gateways through
which to send traffic to each network)."

B. On transmission of routing updates, the gateway sets a
    timer The transmitting gateway does not recompute its
    minimum distance vector or send any new routing updates
    prior to timer expiration.

C. Gateways calculate their routing updates and routing tables
    on the basis of the minimum distance rule. Having selected
    the minimum distance path to a network, the gateway checks
    the "next" gateway address.

    1. If the "next" gateway address is this gateway this
        path is considered unacceptable and the next
        candidate path is checked.

    2. If the "next" gateway address is other than this
        gateway, use the information in the routing update
        matrix to determine if this path (distance of 1 or
        greater) is through a gateway which has reported
        "infinite" distance to the network in question. If
        any step of the path is through a gateway showing
        "infinite" distance to the network, the path is
        considered unacceptable and the next candidate path
        is checked.

Figure 8 shows the routing updates transmitted for the
configuration shown in Figure 7 when routing is performed IAW IEN
109. Figure 8 shows the routing updates transmitted for the
same configuration (Figure 7) when the proposed routing scheme is
used. Both examples are worked from the perspective of G1.

The examples show the differences in the routing updates
generated under each rule. In both cases, packets would be
routed correctly. In the proposed routing scheme (see Figure 8),
fewer routing updates are transmitted with the same results
obtained. This reduced traffic is due, in part, to the rule
requiring the gateway to observe a time-delay between
transmission of routing packets. This rule is particularly
useful when bringing a gateway on-line because it allows the
gateway to receive updates from all or almost all of its
neighbors before responding with its next updates. In addition,
the gateway is able to respond to the loss of connectivity to
Net.#1 correctly based on the update input from G2 (see Figure 8)
following the new method whereas, following the old rules (see
Figure 9), G1 must receive updates from all neighbors before
correctly determining that it has no path to Net.#1.

Under the proposed method, the gateway has a more complex algorithm to calculate its routing table. However, once calculated, the gateway does not need to "tailor" the updates to each neighbor but sends the same information to all neighbors. Also, the gateway does not maintain a copy of the last update transmitted to determine whether there is new information to be transmitted since it transmits updates only when its routing table changes. Under IEN # 109, the gateway is required to maintain copies of each update last sent to each neighbor and compose the new "tailored" updates on an individual basis to those last sent.

Finally, the proposed routing method can be correctly implemented without requiring any special rules regarding treatment of non-routing neighbors. The path checks built into the new routing scheme preclude the shuttling problems described above which could have occured if non-routing gateways were not treated as exceptions.

References

1) V. Strazisar, "Gateway Dynamic Routting", IEN # 25, Bolt, Beranek and Newman, January 25, 1978.

2) V. Strazisar, R. Perlman, "Gateway Routing, An Implementation Specification", IEN # 30, April 11, 1978.

3) V. Strazisar, "How to Build a Gateway", IEN # 109, August 31, 1979.