

HOW TO BUILD A GATEWAY

Virginia Strazisar

IEN #109

August 31, 1979

□

TABLE OF CONTENTS

1. Introduction.....	1
2. Information Maintained by Gateways.....	1
3. Initialization.....	3
4. Determining Connectivity to Networks.....	3
5. Determining Connectivity to Neighbors.....	4
6. Exchanging Routing Information.....	4
7. Computing Routes.....	5
8. Forwarding Traffic.....	7
9. Non-Routing Gateways.....	7
10. Adding New Neighbors and Networks.....	8
11. Communications with Hosts.....	9
12. Future Modifications.....	10
13. Packet Formats.....	11

14. Examples.....	19
15. Tables and Variables.....	23
16. Events and Responses.....	25
□	

## 1. Introduction

This is a description of how to implement a gateway. The gateway forwards internet traffic formatted as described in IEN #111, "Internet Protocol". The gateway polls its attached networks and neighbor gateways to determine its connectivity to them, then exchanges this information with its neighbor gateways in order to compute routes to each network in the catenet. When gateways or network interfaces fail, the gateways compute alternate routes to the networks.

The design for this gateway routing strategy was originally presented in IEN #30, "Gateway Routing, An Implementation Specification". That document gives an overview of the design and explains some of the decisions made in designing this routing strategy. Since IEN #30 was released, this routing strategy has been implemented in several gateways. During implementation, several modifications were made to the original design, thus, those sections of IEN #30 covering the detailed specification of the design are obsolete and are replaced by this document.

## 2. Information Maintained by Gateways

Gateways must maintain information about their connectivity to networks and other gateways. The functional description below explains how this information is obtained and modified by the gateways. For the purpose of explaining the gateway functions, this information is organized into the following tables and variables.

### Number of Networks

The number of networks for which the gateway maintains routing information and to which it can forward packets. The gateways maintain and exchange routing information indexed by network numbers. Thus, the Number of Networks also corresponds to the highest numbered network to which the gateway can route traffic. (Network numbers are listed in IEN #117, "Assigned Numbers".)

### Number of Neighbors

The number of neighbor gateways with which the gateway exchanges routing information. A neighbor gateway of gateway X is any gateway that has an interface on the same network as gateway X. If gateway X and its neighbor have more than one network in common, then each neighbor gateway interface on a network to which gateway X is attached is considered to be a separate neighbor.

### Gateway Addresses

The address of the gateway on each network to which it is attached.

### Neighbor Addresses

The addresses of the neighbor gateways.

### Connectivity to Neighbors

A vector of the status of connectivity between the gateway and each of its neighbor gateways. The connectivity is determined by polling the neighbors.

### Routing Updates

The routing updates that are sent to the neighbor gateways. Each routing update contains the distance from the gateway to each network. There is one routing update for each neighbor.

### Distance Matrix

A matrix of the routing updates received from each of the neighbor gateways. Routing updates contain the distance from a gateway to each network.

### Minimum Distance Vector

A vector of the minimum distance to each network. Distance is measured in networks traversed; a gateway which is physically attached to a network is zero hops from that network; if the gateway must send through one other gateway to get to a network, then it is one hop from that network.

### Routing updates from non-routing neighbor gateways

A routing update for each neighbor gateway that does not participate in this routing scheme. This is the update that the gateway would receive from this neighbor if the neighbor did participate in routing.

### Routing Table

A table containing, for each network, a list of the neighbor gateways on a minimum length route to that network.

### Send Sequence Number

The sequence number that the gateway will use to transmit the next routing update to its neighbors.

### Receive Sequence Numbers

The sequence number that the gateway received in the last routing update from each of its neighbors. There is a sequence number for each neighbor.

### 3. Initialization

The gateway is initialized with the following information:

The gateway's address on each network to which it is attached.

The addresses of its neighbors.

A routing update from each of its neighbor gateways that do not participate in routing.

Initially, the gateway should assume that all its neighbor gateways are down, that it is disconnected from networks to which it is attached, and that the distance reported in routing updates from each neighbor to each network is infinity (see "Computing Routes").

### 4. Determining Connectivity to Networks

The gateway must determine its connectivity to networks to which it is physically attached. The gateway is connected to a network if it can send and receive internet packets on its interface to that network. The method that the gateway uses to determine its connectivity to a network is network dependent. In some networks, the host to network protocol determines whether or not data packets can be sent and received on the host interface. In these networks, the gateway can simply check status information provided by the protocol in order to determine if it can communicate with the network. In other networks, where the host to network protocols are less complex, it may be necessary for the gateway to send traffic to itself to determine if it can communicate with the network. In these networks, the gateways can periodically poll the network to determine if the network interface is operational.

For purposes of computing distances and routes to networks, if the gateway can send and receive traffic on its network interface, then its distance to the network is zero; if it cannot send and receive traffic on the interface, then its distance to

the network is infinity. Note that if a gateway's network interface is not working, it may still be able to send traffic to the network on an alternate route via one of its neighbor gateways.

### 5. Determining Connectivity to Neighbors

A neighbor gateway is in one of three states: up, down, or recovering. In the latter case, the gateway has communicated with its neighbor, and is in a waiting period to determine whether it will be able to continue to communicate with the neighbor. This state exists to prevent a gateway from using a neighbor gateway to forward traffic when communications between it and the neighbor are extremely lossy. The gateway determines its connectivity to each neighbor gateway as follows. Initially,

the neighbor is assumed to be "down". The gateway sends an echo packet (see "Packet Formats") to the neighbor gateway every N seconds and increments a count of echo packets outstanding to that neighbor. When the gateway receives an echo reply packet from the neighbor, it clears the count of outstanding echo packets. If that neighbor was "down", the gateway starts an X second timer and sets the neighbor's status to "recovering". If after X seconds, the neighbor's status is still "recovering", then the neighbor's status is changed to "up". If the count of echo packets outstanding to a neighbor is equal to Z, the neighbor's status is set to "down". As probes are sent every N seconds, a neighbor's status is changed to "down" if it does not respond within (N \* Z) seconds. Note that the waiting period, X, during which the neighbor's status is "recovering" should be at least several times the interval (N \* Z seconds) needed to declare the neighbor "down". This prevents the neighbor gateway from cycling rapidly between the "up" and "down" states.

The gateway maintains a vector of the connectivity between it and its neighbors. This vector is used in computing distances and routes to networks. The vector contains K entries where K is the number of neighbors. If the state of the Jth neighbor is either "down" or "recovering", then the Jth entry of the vector is set to infinity; if the state of the Jth neighbor is "up" then the Jth entry of the vector is set to one.

## 6. Exchanging Routing Information

The gateway receives and transmits routing information reliably using sequence numbered packets and a retransmission and acknowledgement scheme. This scheme works as follows. For each neighbor, the gateway remembers the Receive Sequence Number, R, that it received in the most recent routing packet from that neighbor. This value is initialized by setting it to the sequence number received in the first routing packet received

- 4 -

□

from a neighbor after that neighbor's status is set to "up". On receipt of a routing packet from a neighbor, the gateway subtracts the Receive Sequence Number, R, from the sequence number in the routing packet, S. If this value (S-R) is greater than or equal to zero, then the gateway accepts the routing packet, sends an acknowledgement to the neighbor containing the sequence number S, and replaces the Receive Sequence Number, R, with S. If this value (S-R) is less than zero, the gateway rejects the routing packet and sends a negative acknowledgement to the neighbor with sequence number R.

The gateway has a Send Sequence Number, N, for sending routing packets to all of its neighbors. This sequence number can be initialized to any value. The Send Sequence Number is incremented each time a new routing update is created. On receiving an acknowledgement for a routing update, the gateway subtracts the sequence number acknowledged, A, from the Send Sequence Number, N. If the value (N-A) is non-zero, then an old routing update is being acknowledged. The gateway continues to retransmit the most recent routing update to the neighbor that sent the acknowledgement. If (N-A) is zero, the routing update has been acknowledged. Note that only the most recent routing update need be acknowledged; if a second routing update is generated before the first routing update is acknowledged, only

the second routing update need be acknowledged.

If a negative acknowledgement is received, the gateway subtracts the sequence number negatively acknowledged,  $A$ , from its Send Sequence Number,  $N$ . If this value ( $N-A$ ) is less than zero, then the gateway replaces its Send Sequence Number,  $N$ , with the sequence number negatively acknowledged plus one,  $A+1$ , and retransmits the update to all its neighbors. If ( $N-A$ ) is greater than or equal to zero, then the gateway continues to retransmit the routing update using sequence number  $N$ . In order to maintain the correct sequence numbers at all gateways, routing updates must be retransmitted to all neighbors if the Send Sequence Number changes, even if the routing information does not change.

The gateway retransmits routing updates periodically until they are acknowledged and whenever its Send Sequence Number changes. The gateway sends routing updates only to neighbors that are in the "up" state, not to neighbors that are "down" or "recovering". Examples of the sequence number strategy are given below.

## 7. Computing Routes

A routing update contains the number of networks that a gateway is reporting about followed by a list of the distances to these networks, indexed by network number. Assuming that the gateway has checked the sequence number of a routing update that it has

- 5 -

□

received and has decided to accept the update, the information in the update is processed as follows.

The gateway contains a  $N \times K$  distance matrix, where  $N$  is the number of networks and  $K$  is the number of neighbor gateways. An entry in this matrix, represented as  $d(I,J)$ , is the distance to network  $I$  from neighbor  $J$  as reported in the most recent routing update from neighbor  $J$ . The gateway also contains a vector indicating the connectivity between itself and its neighbor gateways. The values in this vector are computed as discussed above (see "Determining Connectivity to Neighbors"). The value of the  $J$ th entry of this vector, which is the connectivity between the gateway and the  $J$ th neighbor, is represented as  $d(J)$ .

The gateway copies the routing update received from the  $J$ th neighbor into the appropriate row of the distance matrix, then updates its routes as follows. The gateway calculates a minimum distance vector, containing the minimum distance to each network from the gateway. The  $I$ th entry of this vector, represented as  $\text{MinD}(I)$  is:

$$\text{MinD}(I) = \text{minimum over all neighbors of } d(J) + d(I,J)$$

where  $d(J)$  is the distance between the gateway and the  $J$ th neighbor and  $d(I,J)$  is the distance from the  $J$ th neighbor to the  $I$ th network. If the  $I$ th network is attached to the gateway and the gateway can send and receive traffic on its network interface (see "Determining Connectivity to Networks"), then the gateway sets the  $I$ th entry of the minimum distance vector to zero.

Using the minimum distance vector, the gateway computes a list of neighbor gateways through which to send traffic to each network. The entry for network  $I$  contains all neighbors such that:

$\text{MinD}(I) = d(J) + d(I,J)$

In other words, the entry for network I contains all neighbors such that the distance from the gateway to the neighbor plus the distance from the neighbor to the network is equal to the minimum distance from the gateway to the network.

After updating its routes to the networks, the gateway computes the new routing updates to be sent to its neighbors as follows. For each neighbor, J, the gateway constructs a routing update which is a N entry vector where N is the number of networks. The Jth entry of this vector is:

$\text{MinD}(I) \text{ if } \text{MinD}(I) \text{ less than or equal to } d(I,J)$   
 $\text{infinity if } \text{MinD}(I) \text{ greater than } d(I,J)$

- 6 -

□

where  $d(I,J)$  is the distance from the Jth neighbor to the Ith network and  $\text{MinD}(I)$  is the minimum distance from the gateway to the Ith network. In other words, the gateway reports its distance to a network to a neighbor only if it is as close to or closer to a network than its neighbor.

Finally, the gateway must determine whether it should send routing updates to its neighbors. 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. If no routing information has changed since the last routing update was sent, then the gateway does not need to send new routing updates. The gateway sends routing updates only to neighbors that are currently in the "up" state.

The gateway requests routing updates from neighbors that are in the "up" state, but have not yet sent a routing update to it. Routing updates are requested by setting the appropriate bit in the routing update being sent (see "Packet Formats"). Similarly, if a gateway receives a routing update from a neighbor in which the bit requesting a routing update is set, the gateway sends the neighbor the most recent routing update.

## 8. Forwarding Traffic

On receipt of a packet to be forwarded, the gateway extracts the internet destination network field from the packet. If the gateway is attached to the network, and its network interface is operational, then the gateway simply composes the appropriate local network header for the destination network and sends the packet.

If the gateway is not connected to the network, then the gateway checks the list of the neighbors on the route to the destination network. If there are no neighbors on the list, then the gateway drops the packet and sends the internet source a "destination unreachable" message (see "Communications with Hosts").

If there are one or more neighbors on a route to the destination

network, then the gateway sends the packet to one of these neighbors. If there is more than one neighbor, then the neighbors are used in a round robin fashion.

## 9. Non-Routing Gateways

Non-routing gateways are gateways that forward internet traffic, but that do not participate in this routing scheme. Whenever

- 7 -

□

possible, traffic is forwarded only through gateways that participate in this routing scheme. Non-routing gateways are used to forward traffic only if they provide the only route to a network. These gateways are used as follows. Consider a gateway,  $G_1$ . The only non-routing gateways that it needs to know about are its neighbor gateways, as these are the gateways from which it would normally receive routing updates. For each non-routing neighbor gateway of gateway  $G_1$ , compute the routing update that would be sent to  $G_1$  assuming that all gateways and network connections are operational. These routing updates are assembled in  $G_1$ . The gateway,  $G_1$ , first computes its minimum distance vector as explained above, 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. For purposes of computing the minimum distance to a network, the gateway,  $G_1$ , assumes that the distance between itself and the neighbor gateway is zero. After computing the minimum distance vector, the gateway compiles the list of neighbor gateways through which to send traffic to each network and sends routing updates to its neighbors as explained above (see "Computing Routes"). The gateway does not send routing updates to the non-routing neighbor gateways.

## 10. Adding New Neighbors and Networks

Gateways dynamically add routing information about new neighbors and new networks to their tables. The gateway maintains a list of neighbor gateway addresses. When a routing update is received, the gateway searches this list of addresses for the internet source address of the routing update packet. If the internet source address of the routing update is not contained in the list of neighbor addresses, the gateway adds this address to the list of neighbor addresses. The gateway accepts the routing update, sets the neighbor's connectivity status to "up", and computes new routes as explained above. The gateway also begins polling this new neighbor to monitor its connectivity. Note that this strategy requires that one gateway in each pair of neighbor gateways must have the neighbor's address assembled in its tables. The newest gateway can be given a complete list of neighbors, thus avoiding the need to re-assemble older gateways when new gateways are installed.

Gateways obtain routing information about new networks as follows. The gateway maintains a count of the number of networks for which it currently contains routing information,  $N$ . When a routing update is received, the gateway compares this number to the number of networks reported in the routing update,  $M$ . If  $M$

□

is greater than  $N$ , then the gateway updates the count of networks for which it maintains routing information to include the new networks. Next, the gateway expands its distance matrix to include the number of networks reported. The distance matrix is the matrix of distances to networks as reported in routing updates from the neighbor gateways. In expanding the matrix, the gateway assumes the distance to all new networks is infinity. After expanding the matrix, the gateway computes new routes and new routing updates as outlined above.

Note that expanding neighbor address tables and distance matrices may present an implementation problem. In practice, these tables can be assembled to contain some maximum number of neighbors and networks. If a new neighbor cannot be added to a gateway's tables, then routing updates or echo packets from that neighbor should be ignored. The new neighbor will assume that the gateway is down and will not use it to forward traffic. If a gateway cannot add a new network to its routing tables, it can still accept routing updates containing information about the new network, but it will be unable to route traffic to the new network. The routing updates that it sends to its neighbors will not contain information about the new network. Whenever a gateway receives a routing update that does not report the distance to a network, the gateway should assume that the distance is infinity.

#### 11. Communications with Hosts

The gateway sends messages to internet hosts in several situations: when the gateway cannot reach the internet destination, when the gateway does not have the buffering capacity to forward a packet, and when the gateway can direct the host to send traffic on a shorter route. These situations are explained in more detail below. The formats for messages sent from a gateway to a host are given in the section "Packet Formats".

If, according to the information in the gateway's routing tables, the network specified in the internet destination field of a packet is unreachable, i.e. the distance to the network is infinity, the gateway sends a destination unreachable message to the internet source host of the packet. In addition, in some networks, the gateway may be able to determine if the internet destination host is unreachable. Gateways in these networks may send destination unreachable messages to the source host when the destination host is unreachable.

The gateway may discard internet packets if it does not have the buffer space needed to queue the packets for output to the next network on the route to the destination network. If the gateway

□

discards a packet, it sends a source quench message to the

internet source host of the packet. The source quench message is a request to the host to cut back the rate at which it is sending traffic to the internet destination. The gateway sends a source quench message for every message that it discards. On receipt of a source quench message, hosts should cut back the rate at which they are sending traffic to the specified destination until they no longer receive source quench messages from the gateway. The hosts can then gradually increase the rate at which they are sending traffic to the destination until they again receive source quench messages from the gateway.

The gateway sends a redirect message to a host in the following situation. A gateway, G1, receives an internet packet from a host on a network to which the gateway is attached. The gateway, G1, checks its routing table and obtains the address of the next gateway, G2, on the route to the packet's internet destination network, X. If G2 and the host identified by the internet source address of the packet are on the same network, a redirect message is sent to the host. The redirect message advises the host to send its traffic for network X directly to gateway G2 as this is a shorter path to the destination. The gateway forwards the original data packet to its internet destination.

## 12. Future Modifications

As there are now many networks planned or implemented, and as these networks may be assigned a wide range of network numbers, it is no longer feasible to maintain routing information in tables indexed by network numbers. Gateways will be modified to use a hash table scheme to convert network numbers to internal indices to reference routing tables. The information in routing updates exchanged by the gateways will no longer be indexed by network number. The format of routing updates will be modified to include both the network number and the distance to the network. These modifications will be documented in a separate note.

## 13. Packet Formats

Gateway to gateway and gateway to host communications use the version 4 internet protocol. The first octet of the data portion of the packet is a gateway type field; the value of this field determines the format of the remaining data. The format of internet packets is explained in IEN #111, "Internet Protocol". Unless otherwise noted under the individual format descriptions, the values of the internet header fields are as follows:

Version	4	
IHL	4	Internet header length in 32-bit words; this is 5.
Type of Service	0	
Total Length		Length of internet header and data in octets.
Identification, Flags, Fragment Offset		Used in fragmentation, see IEN #111.
Time to Live		Time to live in seconds; as this field is decremented at each machine in which the packet is processed, the value in this field should be at least as great as the number of gateways which this packet will traverse. (This must be considered in the messages sent to hosts.)
Protocol	3	
Header Checksum		The 16 bit one's complement of the one's complement sum of all 16 bit words in the header. For computing the checksum, the checksum field should be zero. This checksum may be replaced in the future, see updates of the Internet Protocol Specification.
Source Address		The address of the gateway that composes the packet. Unless otherwise noted, this can be any of the gateway's addresses.
Destination Address		The address of the gateway or host to which the packet should be sent.

- 11 -

1

## Routing Update

0	1	2	3
0 1 2 3 4 5 6 7 8 9 0	1 2 3 4 5 6 7 8 9 0	1 2 3 4 5 6 7 8 9 0	1
! Gateway	Type	unused	Sequence Number
! NU !	N	Distance 1	Distance 2
Distance 3			

Gateway Type	1
Sequence Number	The 16-bit sequence number used to transmit

routing updates.

NU A 1-bit field. This bit is set if the source gateway requests a routing update from the destination gateway.

**N** The number of networks for which distances are reported in this update.

Distance 1..N      A set of octets that are the values in the routing update for distances to network numbers 1 through N. The network numbers are listed in IEN #117, "Assigned Numbers". If the Ith network is unreachable, the distance to the network is infinity, which is represented as 177 (octal).

- 12 -

1

### Acknowledgement or Negative Acknowledgement

0	1	2	3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1	+	+	+
! Gateway	Type	unused	!
+	+	+	+

Gateway Type 2 for acknowledgement; 10 (decimal) for negative acknowledgement.

**Sequence Number** The 16-bit sequence number that the gateway is acknowledging or negatively acknowledging.



Destination Unreachable Packet

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1  
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+  
! Gateway Type ! Code ! unused !  
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+  
! Internet Header + 64 bits of Original Data Packet !  
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

Destination Address      The source network and address from the  
original data packet.

Gateway Type      3

Code      0 = net unreachable; 1 = host unreachable.

Internet Header + 64 bits  
of Data Packet      The internet header plus the first 64 bits  
of the original data packet. This data is  
used by the host to match the message from  
the gateway to the appropriate process.  
If a higher level protocol uses port numbers,  
they are assumed to be in the first 64 data  
bits of the original data packet.

1

## Source Quench Packet

**Destination Address** The source network and address of the original data packet.

## Gateway Type 4

□

### Redirect Packet

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1  
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+  
! Gateway Type ! unused !  
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+  
! Gateway Network! Gateway Address !  
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+  
! Internet Header + 64 bits of Original Data Packet !  
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

Destination Address The source network and address of the original data packet.

Gateway Type 5

Gateway Network Address of the gateway to which traffic for the network specified in the internet destination network field of the data packet should be sent.

Internet Header + 64 bits of Data Packet The internet header plus the first 64 bits of the original data packet. This data is used by the host to match the message from the gateway to the appropriate process. If a higher level protocol uses port numbers, they are assumed to be in the first 64 data bits of the original data packet.

□

## Echo or Echo Reply Packet

0	1	2	3
0 1 2 3 4 5 6 7 8 9 0	1 2 3 4 5 6 7 8 9 0	1 2 3 4 5 6 7 8 9 0	1 2 3 4 5 6 7 8 9 0 1
-----	-----	-----	-----
! Gateway	Type	unused	!
-----	-----	-----	-----

**Source Address** In an echo packet, the address of the gateway on the same network as the neighbor to which it is sending the echo packet. In an echo reply packet, the source and destination addresses are simply reversed.

Gateway Type 8 for echo packet; 0 for echo reply.

## Net Interface Status Packet

! Gateway Type ! unused !  
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+  
Source Address  
Destination Address The address of the gateway's network  
interface. The gateway can send Net  
Interface Status Packets to itself to  
determine if it is able to send and  
receive traffic on its network interface.

Gateway Type 9

- 18 -

□

#### 14. Examples

The following examples illustrate the sequence number scheme used for sending and receiving routing updates.

Example 1:

Gateway G has three neighbors: A, B and C.  
Gateway G's Send Sequence Number is 5.  
Connectivity between gateway G and gateway A was broken and is now restored; thus, A has not received the last update from G.  
Gateway A's Receive Sequence Number for G is 3.  
Gateway B's Receive Sequence Number for G is 4.  
Gateway C's Receive Sequence Number for G is 4.

Initially, gateway G has the following values in its tables:

Send Sequence    Received Acknowledgement from Neighbors

	A	B	C
5	N	N	N

(N for no, Y for yes)

G (5) ---> A            G sends routing update 5 to its neighbors

G (5) ---> B

G (5) ---> C

A (ACK 5) ---> G    A computes (sequence number received in packet - Receive Sequence Number) = (5-3); as this is greater than or equal to zero, A acknowledges routing update 5 and replaces Receive Sequence Number for G with 5

B (ACK 5) ---> G    B computes (sequence number received in packet - Receive Sequence Number) = (5-4); as this is greater than or equal to zero, B acknowledges routing update 5 and replaces its Receive Sequence Number for G with 5

C (ACK 5) ---> G

On receipt of these acknowledgements, G computes (Send Sequence Number - sequence number acknowledged) = (5-5). As this value is zero, G notes that each of its neighbors has acknowledged its routing update.

- 19 -

□

Gateway G now has the following values in its tables:

Send Sequence    Received Acknowledgement from Neighbors

	A	B	C
5	Y	Y	Y

Gateway A's Receive Sequence Number for G is 5.

Gateway B's Receive Sequence Number for G is 5.

Gateway C's Receive Sequence Number for G is 5.

□

**Example 2:**

Gateway G has three neighbors: A, B, and C.

Gateway G's Send Sequence Number is 24.

Gateway A has been disconnected from gateway G for some long time, thus, it has missed many updates from G and G's Send Sequence Number has now wrapped around.

Gateway A's Receive Sequence Number for G is 500.

Gateway B's Receive Sequence Number for G is 23.

Gateway C's Receive Sequence Number for G is 23.

Initially, gateway G has the following values in its tables:

Send Sequence    Received Acknowledgement from Neighbors

      A    B    C

24            N    N    N (N for no, Y for yes)

G (24) ----> A            G sends routing update 24 to all its neighbors

G (24) ----> B

G (24) ----> C

A (NAK 500) ----> G    A computes (sequence number received in packet - Receive Sequence Number) = (24-500); as this is less than zero, A sends a negative acknowledgement with sequence number 500

On receipt of the negative acknowledgement, gateway G computes (Send Sequence Number - sequence number negatively acknowledged) = (24-500). As this value is less than zero, G replaces its Send Sequence Number with 501 and retransmits the routing update with this sequence number.

G (501) ----> A

G (501) ---> B	
G (501) ---> C	
A (ACK 501) ---> G	A computes (sequence number received in packet - Receive Sequence Number) = (501-500); as this is greater than or equal to zero, A acknowledges sequence number 501 and replaces its Receive Sequence Number for G with 501
B (ACK 24) ---> G	B receives the update from G with sequence number 24; B computes (sequence number received in packet - Receive Sequence Number) = (24-23); as this is greater than or equal to zero, B acknowledges sequence number 24 and replaces its Receive Sequence Number for G with 24
C (ACK 24) ---> G	

- 21 -

□

On receipt of these acknowledgements, G computes (Send Sequence Number - sequence number acknowledged) = (501-24). As this value is non-zero, G retransmits the routing update to these neighbors.

G (501) ---> B	
G (501) ---> C	
B (ACK 501) ---> G	B computes (sequence number received in packet - Receive Sequence Number) = (501-24); as this is greater than or equal to zero, B acknowledges sequence number 501 and replaces its Receive Sequence Number for G with 501

C (ACK 501) ---> G

On receipt of these acknowledgements, G computes (Send Sequence Number - sequence number acknowledged) = (501-501). As this value is zero, G notes that each of its neighbors has acknowledged its routing update.

Gateway G now has the following values in its tables:

Send Sequence	Received Acknowledgement from Neighbors
	A      B      C
501	Y      Y      Y

Gateway A's Receive Sequence Number for G is 501.  
 Gateway B's Receive Sequence Number for G is 501.  
 Gateway C's Receive Sequence Number for G is 501.

□

## 15. Tables and Variables

The following is a list of variables and tables in a typical gateway implementation.

1. Number of Networks

2. Number of Neighbors

3. Gateway Addresses

The addresses of the gateway's network interfaces.

4. Neighbor Gateway Addresses

The address of each gateway network interface that is on the same network as this gateway.

5. Neighbor Connectivity Vector

A vector of the connectivity between this gateway and each of its neighbors. It contains  $K$  entries where  $K$  is the Number of Neighbors. Values in this vector are computed as explained in "Determining Connectivity to Neighbors".

6. Distance Matrix

A matrix of the routing updates received from the neighbor gateways.

7. Minimum Distance Vector

A vector containing the minimum distance to each network. It is computed as described in "Computing Routes".

8. Routing Updates

A set of vectors giving the distance from this gateway to each network. There is a separate vector to be sent to each neighbor. These are computed as described in "Computing Routes".

9. Routing Updates from Non-Routing Gateways

The routing updates that would have been received from each neighbor gateway that does not participate in this routing strategy. These are computed as described in "Non-Routing Gateways".

□

## 10. Routing Table

A table containing, for each network, a list of the neighbor gateways on a minimum distance route to the network. This is computed as described in "Computing Routes".

## 11. Send Sequence Number

The sequence number that will be used to send the next routing update.

## 12. Receive Sequence Numbers

The sequence numbers that the gateway received in the last routing update from each of its neighbors.

## 13. Received Acknowledgement Vector

A vector indicating whether or not each neighbor has acknowledged the sequence number in the most recent routing update sent.

□

## 16. Events and Responses

The following is a list of the events that occur at a gateway and the gateway's responses. The variables and tables referred to are listed above. This is a general guide for an implementation; see the sections above for the details of the various computations.

1. Connectivity to a network to which the gateway is attached changes.

1. Update the Minimum Distance Vector.
2. Recompute the Routing Updates.
3. Recompute the Routing Table.
4. If any routing update has changed, send the new routing updates to the neighbors.

2. Connectivity to a neighbor gateway changes.

1. Update the Neighbor Connectivity Vector.
2. Recompute the Minimum Distance Vector.
3. Recompute the Routing Updates.
4. Recompute the Routing Table.
5. If any routing update has changed, send the new routing updates to the neighbors.

3. A Routing Update Packet is received.

1. Compare the internet source address of the Routing Update Packet to the Neighbor Addresses. If the address is not on the list, add it to the list of Neighbor Addresses, increment the Number of Neighbors, and set the Receive Sequence Number for this neighbor to the sequence number in the Routing Update Packet.
2. Compare the Receive Sequence Number for this neighbor to the sequence number in the Routing Update Packet to determine whether or not to accept this packet. If the packet is rejected, send a Negative Acknowledgement Packet. If the packet is accepted, send an Acknowledgement Packet and proceed with the following steps.

- 25 -

□

3. Compare the number of networks reported on in the Routing Update Packet to the Number of Networks. If the number of networks in the packet is greater than the Number of Networks, then replace Number of Networks with the number of networks in the packet. Expand the Distance Matrix to account for the new networks.

4. Copy the routing update received into the appropriate row of the Distance Matrix.

5. Recompute the Minimum Distance Vector.
6. Recompute the Routing Updates.
7. Recompute the Routing Table.
8. If any routing update has changed, send the new routing updates to the neighbors.

4. An Acknowledgement or Negative Acknowledgement Packet is received.

1. Compare the sequence number in the packet to the Send Sequence Number. If necessary, replace the Send Sequence Number, and retransmit the routing updates. If the Send Sequence Number is acknowledged, update the entry in the Received Acknowledgment Vector for the neighbor that sent the acknowledgement.

5. A data packet is received.

1. Forward the data packet using the information in the Routing Table entry for the packet's destination network.
2. If the destination is unreachable, send a Destination Unreachable Packet to the internet source of the data packet.
3. If the data packet is discarded because there are no buffers available in the gateway, then send a Source Quench Packet to the internet source of the data packet.
4. If the gateway to which the data packet is being forwarded and the data packet's internet source are on the same network, send a Redirect Packet to the internet source of the data packet.