Network Working Group K. Kinnear
INTERNET DRAFT American Internet Corporation

March 1998

Expires September 1998

DHCP Safe Failover Protocol
<draft-ietf-dhc-safe-failover-proto-00.txt>

Status of this Memo

Abstract

The DHCP protocol [RFC 2131] [1] allows multiple DHCP servers. A recent draft, the DHCP Failover Protocol [3], has been distributed which is designed to provide a redundant DHCP solution. While clearly a work in progress, it is equally clear that it can be made to work and meet the goals that the authors have set for it. One of the goals of the Failover protocol is that it should avoid duplicate IP address allocations, except under some rare circumstances. While this approach may be quite reasonable in a wide variety of environments, some environments require a DHCP redundancy solution which can (optionally) ensure that no possibility of duplicate IP address allocation exists.

The Safe Failover protocol is designed to build on top of the Failover protocol, and add to it certain protocol exchanges, practices and algorithms which will create a DHCP redundancy solution which avoids duplicate IP address allocation.

1. Introduction

The DHCP Safe Failover protocol is designed to layer on top of the DHCP Failover protocol, and in like manner support automatic failover from a primary to a secondary server.

In certain unusual failure cases servers implementing the DHCP Failover protocol can end up allocating the same IP address to two clients. The DHCP Safe Failover protocol is designed to prevent this situation from occurring.

1.1. The Language of Requirements

Throughout this document, the words that are used to define the significance of particular requirements are capitalized. These words are:

- "MUST"

  This word or the adjective "REQUIRED" means that the item is an absolute requirement of this specification.

- "MUST NOT"

  This phrase means that the item is an absolute prohibition of this specification.

- "SHOULD"

  This word or the adjective "RECOMMENDED" means that there may exist valid reasons in particular circumstances to ignore this item, but the full implications should be understood and the case carefully weighed before choosing a different course.

- "SHOULD NOT"

  This phrase means that there may exist valid reasons in particular circumstances when the listed behavior is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing any behavior described with this label.

- "MAY"

  This word or the adjective "OPTIONAL" means that this item is truly optional. One vendor may choose to include the item because a particular marketplace requires it or because it enhances the product, for example; another vendor may omit the

same item.

1.2. Terminology

This document uses the following terms:

- "DHCP client" or "client"

  A DHCP client is an Internet host using DHCP to obtain configuration parameters such as a network address.

- "client"

  Whenever the term client is used in this draft, it refers to a DHCP client (and not a server communicating with another server using this protocol).

- "DHCP server" or "server"

  A DHCP server is an Internet host that returns configuration parameters to DHCP clients.

- "binding"

  A binding is a collection of configuration parameters, including at least an IP address, associated with or "bound to" a DHCP client. Bindings are managed by DHCP servers.

- "subnet address pool"

  A subnet address pool is the set of IP address which is associated with a particular network number and subnet mask. In the simple case, there is a single network number and subnet mask and a set of IP addresses. In the more complex case (sometimes called "secondary subnets"), several (apparently unrelated) network number and subnet mask combinations with their associated IP addresses may all be configured together into one subnet address pool.

- "primary server" or "primary"

  A DHCP server configured to provide primary service to a set of DHCP clients for a particular set of subnet address pools.

- "secondary server" or "secondary"

A DHCP server configured to act as backup to a primary server for a particular set of subnet address pools.

- "stable storage"

  Every DHCP server is assumed to have some form of what is called "stable storage". Stable storage is used to hold information concerning IP address bindings (among other things) so that this information is not lost in the event of a server failure which requires restart of the server.

2. Goals and Limitations of the Protocol

2.1. Goals and Requirements

1. Implementations of this protocol must work with existing DHCP client implementations based on the DHCP protocol [1].

2. Implementations of the protocol must work with existing BOOTP relay implementations.

3. Avoid binding an IP address to a client while that binding is currently valid for another client. In other words, don't allocate the same IP address to two clients.

4. Ensure that an existing client can keep its existing IP address binding if it can communicate with either the primary or secondary DHCP server implementing this protocol -- not just whichever server that originally offered it the binding.

5. Do not add any requirement for communication with another server to the processing between a DHCPDISCOVER and a DHCPOFFER or between a DHCPREQUEST and a DHCPACK.

   DISCUSSION:

   The implications of this goal are that "lazy" update of IP address binding information is required. In other words, because of this goal, the protocol cannot require one server to update another server with information concerning a new IP address binding prior to sending the DHCPACK to the DHCP client.

6. Ensure that a new client can get an IP address from some server.

7. Ensure that in the face of partition, where servers continue to run but cannot communicate with each other, the above goals and requirements may be met. In addition, when the partition condition is removed, allow graceful automatic re-integration without requiring human intervention.

8. If either primary or secondary server loses all of the information that is has stored in stable storage, it should be able to refresh its stable storage from the other server.

2.2. Limitations of this Protocol

1. Determination of permanent server failure.

   The protocol provides a way to detect when the primary and secondary servers cannot communicate, but once this condition has been detected, does not provide any way to further distinguish between network failure and direct failure of one of the servers.

2. Some additional IP addresses required.

   In order to handle the failure case where both servers are able to communicate with DHCP clients, but unable to communicate with each others, each server has to have a small number of IP addresses that it can use to allocate to newly arrived DHCP clients during such a period. The number of additional IP addresses required is based only on the arrival rate of new DHCP clients, and is not influenced in any way by the total number of DHCP clients supported by the server pair.

3. Not a load-sharing solution.

   This protocol does not provide any solutions to load-sharing, since only one DHCP server in the pair is supposed to be communicating with DHCP clients.

3. Protocol Overview

The DHCP Failover Protocol [3] presents a protocol designed to provide a redundant DHCP capability. It makes a number of simplifying assumptions in order to allow specification of a straightforward, easy to understand and relatively easy to implement protocol.

Some environments need a protocol which provides a redundant DHCP

solution in which the possibility of duplicate IP address allocation is significantly reduced from that provided by the Failover protocol.

These are typically environments where the cost of correcting a possible duplication IP address assignment is sufficiently great that it is worth considerable effort to ensure that such a condition does not occur.

There are two cases where the Failover protocol [3] can end up with duplicate IP address allocations:

- Primary server crash before "lazy" update:

  In the case where the primary server sends an ACK to a client for a newly allocated IP address and then crashes prior to sending the corresponding update to the secondary server, the secondary server will have no record of the IP address allocation. When the secondary server takes over, it may well try to allocate that IP address to a different client. In the case where the first client to receive the IP address is not on the net at the time (yet while there was still time to run on its lease), an ICMP echo (i.e., ping) will not prevent the secondary server from allocating that IP address to different client.

  A more likely and subtle version of this problem is where the primary server crashes after extending a client's lease time, and before updating the secondary with new time using a lazy update. After the secondary takes over, if the client is not connected to the network the secondary will believe the client's lease has expired when, in fact, it has not. In this case as well, the IP address can be reallocated to a different client.

- Network partition where servers can't communicate but each can talk to clients:

  Several conditions are required for this situation to occur. First, due to a network failure, the primary and secondary servers cannot communicate. As well, some of the clients of the primary server must be able to communicate with the primary server, and some of the clients of the primary server must now only be able to communicate with the secondary server. When this condition occurs, both primary and secondary servers are going to allocate IP addresses for new clients from the same pool of available addresses. At some point, then, two clients will end

up being allocated the same IP address. This will cause potentially serious problems when the network failure that created this situation is corrected.

While the Failover protocol has protocol messages to detect this condition when communications are restored, in some environments preventing it is preferable to detecting it.

The remainder of section 3 discusses the concepts employed to solve the problem presented by each of the above situations. In order to define and explain these concepts, this protocol defines states into which both the primary and secondary server may transition. Section 3.1 briefly discusses these states, and then section 3.2 goes on to discuss how the problems presented above are solved.

3.1. Primary and Secondary States

In order to allow conceptual discussion and rigorous specification of the safe failover protocol, several states are defined into which either the primary or secondary server may transition. These states are briefly discussed in this section. Full specification of the each server's actions in each state is deferred until section 4 for the primary server and section 5 for the secondary server.

The possible server states are:

- NORMAL State:

  NORMAL state is the state used by a server when it can communicate with the other server in the primary-secondary server pair.

  When in this state, the primary responds to DHCP clients requests, while the secondary does not.

- COMMUNICATION-INTERRUPTED state:

  COMMUNICATION-INTERRUPTED state is the state into which a server goes automatically whenever it cannot communicate with the other server. Both the primary and secondary servers can go into this state, although the behavior changes that result are different. Primary and secondary servers cycle automatically (i.e., without administrative intervention) between NORMAL and COMMUNICATION-INTERRUPTED state as the network connection between them undergoes failures and then becomes operational or as the other server in the pair cycles between operational and non-operational.

  No duplicate IP address allocation can occur while the servers cycle between these states.

  In this state both servers respond to DHCP client requests. When allocating new IP addresses, each server allocates from a different pool. When responding to renewal requests, each server will

allow continued renewal of a DHCP client's current lease on an IP address.

- PARTNER-DOWN state:

  PARTNER-DOWN state is a state either server can enter. Once a server has entered NORMAL state, the PARTNER-DOWN state is entered only on command of an external agency (typically an administrator of some sort) or after the expiration of an externally configured minimum safe-time after the beginning of COMMUNICATION-INTERRUPTED state.

  When in this state, the server ceases to operate in such a way such that it assumes that the other server could still be operational and communicating to a different set of clients, but instead assumes that it is the only server operating.

  Only one server should be operating in this state at a time.

  The server in this state will respond to DHCP client requests. It will allow renewal of all outstanding leases on IP addresses, and will allocate IP addresses from its own pool, and after a fixed period of time, it will allocate IP addresses from the set of all available IP addresses.

  The server will transition out of PARTNER-DOWN state after automatic re-integration the companion server is complete. This automatic re-integration will typically be initiated by the restart of the server which was down.

- RECOVER state:

  This state indicates that the server has no information in its stable storage. A server in this state will attempt to refresh its stable storage from the other server.

- POTENTIAL-CONFLICT state:

  This state indicates that the two servers are attempting to reintegrate with each other, but at least one of them was running in a state that did not guarantee automatic reintegration would be possible. In POTENTIAL-CONFLICT state the servers may determine that the same IP address has been offered and accepted by two different DHCP clients.

- SYNC state:

In this state, the secondary server attempts to synchronize its stable storage with the primary server. Both the primary and secondary may have information that the other lacks.

3.2. Conceptual Overview

At the beginning of this section, two specific scenarios were discussed that need to be handled by the safe failover protocol. Now that an introduction to the possible server states has been given, the techniques used to solve these problems may be explained.

The following techniques are used:

- Control of lease time.

  This protocol requires a DHCP server to deal with several different lease intervals and places specific restrictions on their relationships. The purpose of these restrictions is to allow the other server in the pair to be able to make certain assumptions.

  The different lease times are:

  - desired client lease interval

    The desired client lease interval is the lease interval that the DHCP server would like to give to the DHCP client in the absence of any restrictions imposed by the safe failover protocol. Its determination is outside of the scope of this protocol. Typically this is the result of external configuration of a DHCP server.

  - actual client lease interval

    The actual client lease internal is the lease interval that that DHCP server gives out to the DHCP client. It may be shorted than the desired client lease interval (as explained below).

  - primary server lease interval

    The primary server lease interval is the interval after which the primary server believes that DHCP client's lease will expire.

  - desired secondary server lease interval

    The desired secondary server lease interval is the interval the primary server tells to the secondary server after which the

lease will expire.

o acknowledged secondary server lease interval

The acknowledged secondary server lease interval is the interval the secondary server has most recently acknowledged.

The key restriction (and guarantee) that the primary server makes with respect to lease intervals is that the actual client lease interval never exceeds the acknowledged secondary server lease interval (if any) by more than a fixed amount. This fixed amount is called the "maximum delta lease interval" (MDLI).

The MDLI MAY be configurable, but for correct server operation it MUST be known to both the primary and secondary servers.

The primary server MUST record in its state both the primary server lease interval and the most recently acknowledged secondary server lease interval. It is assumed that the desired client lease interval can be determined through techniques outside of the scope of this protocol.

DISCUSSION:

This protocol mandates no particular detailed algorithms concerning these lease intervals, as long as the key relationship of the actual client lease interval < (acknowledged secondary server lease interval + MDLI).

In the interests of clarity, however, let's examine a specific example. The MDLI in this case is 1 hour. The desired client lease interval is 3 days. In operation this might work as follows:

When a primary server makes an offer for a new lease on an IP address to a DHCP client, it determines the desired client lease interval (in this case, 3 days). It then examines the acknowledged secondary lease interval (which in this case is zero). Since the actual client lease interval can not be allowed to exceed the current secondary lease interval by more than the MDLI, the offer made to the DHCP client (the actual client lease interval) is for (essentially) the MDLI, 1 hour.

Once the primary server has performed the ACK to the DHCP client, it will update the secondary server with the lease

information. However, the secondary server lease interval will be composed of the current actual client lease interval

+ (1.5 * desired client lease interval). Thus, the secondary server is updated with a lease interval of 4.5 days + 1 hour.

When the primary server receives an ACK to its update of the secondary server's lease interval, it records that as the acknowledged secondary server lease interval. The primary server MUST ensure that the secondary server has received and recorded in its stable storage the secondary server lease interval.

When the DHCP client attempts to renew at T2 (approximately one half an hour from the start of the lease), the primary server again determines the desired client lease time, which is still 3 days. It then compares this with the remaining acknowledged secondary server lease interval (adjusting for the time passed since the secondary server was last updated), which is 4.5 days + .5 hours. The actual client lease interval can now be equal to the desired client lease interval as it is less than the acknowledged secondary lease interval.

When the primary DHCP server updates the secondary DHCP server after the DHCP client's renewal ACK is complete, it will calculate the secondary server lease interval as the actual client lease interval (3 days this time) + .5 the desired client lease interval (1.5 days). In this way, the primary attempts to have the secondary always "lead" the client in its understanding of the client's lease interval.

Once the initial actual client lease interval of the MDLI is past, the protocol operates effectively like the DHCP protocol does today in its behavior concerning lease intervals. However, the guarantee that the actual client lease interval will never exceed the acknowledged secondary server lease interval by more than the MDLI allows full recovery from failures in lazy update.

The key problem with lazy update (in the absence of the control of time described above) is that if, after updating a client with a particular lease interval, the primary server fails in the small window before updating the secondary server, the secondary server will believe that a lease has expired even though the client still retains a valid lease on that IP address.

Even with the regime described above, the actual client lease interval can exceed the secondary server lease interval -- but by an amount less than or equal to the MDLI.

In the case where the secondary needs to take over from the primary, in COMMUNICATIONS-INTERRUPTED state the secondary will not reallocate any IP addresses from one client to a different clients. When transitioning to the PARTNER-DOWN state (where the secondary is allowed to reallocate IP addresses), the secondary need merely wait the MDLI before reallocating any IP addresses from one client to another.

In this way, any clients which have a lease on an IP address with a lease time greater that than known by the secondary will either have contacted the secondary during that time or the their lease will have expired.

Two guarantees are required for this to be effective. The first is the just described guarantee concerning actual client lease time and its difference from the secondary server lease time. The second is that the secondary server must be able to depend on the fact that, after transition to PARTNER-DOWN state, the primary server is guaranteed to not respond to DHCP clients (as described in section 3.1).

- Secondary creates its own address pool

  When in COMMUNICATION-INTERRUPTED state, the secondary needs to be able to allocate IP address to new clients appearing on the network to which it can communicate. Were the secondary to simply allocate apparently available IP addresses, it could conflict with the primary, which might be operating and simply unable to communicate with the secondary.

  In order to allow the secondary to respond to new DHCP clients appearing on the network during periods when it is in the COMMUNICATIONS-INTERRUPTED state, the secondary must acquire from the primary a set of IP addresses which it can use for this purpose. In order to create this address pool, the secondary uses the "Reply to" DHCP option to create and maintain the address pool. (See Section 6.)

- Controlled re-allocation of IP addresses

  Careful control of re-allocation of IP addresses is central to the correct operation of this protocol.

  When in NORMAL state the primary server must update the secondary server whenever a lease expires or an IP address is released, and

it must ensure the secondary has been successfully updated before offering the IP address of the expired or released IP address to a different client.

When in COMMUNICATION-INTERRUPTED state, neither server will allow an IP address previously used by one client to be offered to a different client.

When in PARTNER-DOWN state, either server MUST wait for the MDLI beyond the scheduled expiration of the lease for every IP address before re-allocating that IP address to different DHCP client. For IP addresses that are available (or have already expired), the server must wait for at least the MDLI before allocating available IP addresses from what was previously the "other" server's pool of available IP addresses.

o Safe Period

Due to the restrictions imposed on each server while in COMMUNICATIONS-INTERRUPTED state, long-term operation in this state is not feasible for either server. One reason that these states exist at all is to allow the servers to easily survive transient network communications failures of a few minutes to a few days (although the actual time periods will depend a great deal on the DHCP activity of the network in terms of arrival and departure of DHCP clients on the network).

Eventually, when the servers are unable to communicate, they will have to move into a state where they no longer can re-integrate without the some possibility of a duplicate IP address allocation. There are two ways that they can move into this state (known as PARTNER-DOWN).

They can either be informed by external command that, indeed, the partner server is down. In this case, there is no difficulty in moving into the PARTNER-DOWN state since it is an accurate reflection of reality and the protocol has been designed to operate correctly (even during reintegration) if, when in PARTNER-DOWN state the partner is, indeed, down.

The other difficulty is when the servers are running unattended for extended periods, and in this case the option is provided to configure something called a "safe-period" into each server. This OPTIONAL safe-period is the period after which either the primary or secondary server will automatically transition to PARTNER-DOWN from COMMUNICATIONS-INTERRUPTED state. If this transition is completed and the partner is not down, then the possibility of duplicate IP address allocations will exist.

The goal of the "safe-period" is to allow network operations

staff some time to react to a server moving into COMMUNICATIONS-INTERRUPTED state. During the safe-period the only requirement

is that the network operations staff determine if both servers are still running -- and if they are, to either fix the network communications failure between them, or to take one of the servers down before the expiration of the safe-period.

The length of the safe-period is installation dependent, and depends in large part on the number of unallocated IP addresses within the subnet address pool and the expected frequency of arrival of previously unknown DHCP clients requiring IP addresses. Many environments should be able to support safe-periods of several days.

During this safe period, either server will allow renewals from any existing client. The only limitation concerns the need for IP addresses for the DHCP server to hand out to new DHCP clients and the need to re-allocate IP addresses to different DHCP clients.

The number of "extra" IP addresses required is equal to the expected total number of new DHCP clients encountered during the safe period. This is dependent only on the arrival rate of new DHCP clients, not the total number of outstanding leases on IP addresses.

In the unlikely event that a relatively short safe period of an hour is all that can be used (given a dearth of IP addresses or a very high arrival rate of new DHCP clients), even that can provide substantial benefits in allowing the DHCP subsystem to ride through a minor problems that could occur and be fixed within that hour. In these cases, no possibility of duplicate IP address allocation exists, and re-integration after the failure is solved will be automatic and require no operator intervention.

4. Primary Server Operation

The operation of the primary server is as specified in the Failover protocol [3], with the exception of the following situations and cases.

4.1. Primary Server Initialization

When the primary server starts, there are three possibilities: it has never started before and therefore has no record of any previous state nor of any client binding information; it has started before and has a record of a previous state and possibly of some client binding information; it has started before, but failed catastrophically, and now has no record of any previous state (nor of any client

binding information).

When the primary server starts, if it has any record of a previous state, then if that state was NORMAL or COMMUNICATIONS-INTERRUPTED it moves to COMMUNICATIONS-INTERRUPTED state. If that state was PARTNER-DOWN or POTENTIAL-CONFLICT, then it moves to PARTNER-DOWN state. If that state was RECOVER, then the primary server moves into the RECOVER state.

If it has no record of any previous state, then either this is an initial startup, or a recovery from a catastrophic failure where stable storage and all client binding information was lost. These are distinguished by recovery from a catastrophic failure being indicated by some external configuration indication to the primary server. If this external indication is present, the primary server moves into RECOVER mode and attempts to recover its client binding database from the secondary server. If that indication is not present, the primary server moves directly into PARTNER-DOWN state.

4.2. Primary Server State Transitions

Figure 4.2-1 is the diagram of the primary server's state transitions. The remainder of this section contains information important to the understanding of that diagram.

The server stays in the current state until all of the actions specified on the state transition are complete. If communications fails during one of the actions, the server simply stays in the current state and attempts a transition whenever the conditions for a transition are later fulfilled.

In the state transition diagram below, the "+" or "-" in the upper right corner of each state is a notation about whether communication is ongoing with the secondary server. The legend "responsive" and "unresponsive" in each state indicates whether the primary server is responsive to DHCP client requests in the respective state.

In the diagram state transition diagram below, when communication is reestablished between the primary and secondary server, the primary server must record the state of the secondary server when the communication was reestablished. If the state of the secondary server changes while communicating, then the primary server moves through the communications-failed transition, and into whatever state results. It then immediately moves through whatever state

transition
   is appropriate given the current state of the secondary server.

   DISCUSSION:

The point of this technique is simplicity, both in explanation of the protocol and in its implementation. The alternative to this technique of memory of partner state and automatic state transition on change of partner state is to have every state in the following diagram have a state transition for every possible state of the partner. With the approach adopted, only the states in which communications are reestablished require a state transition for each possible partner state.

All state transitions of the primary server must be recorded in its stable storage, and thus be available to the server after a server restart.

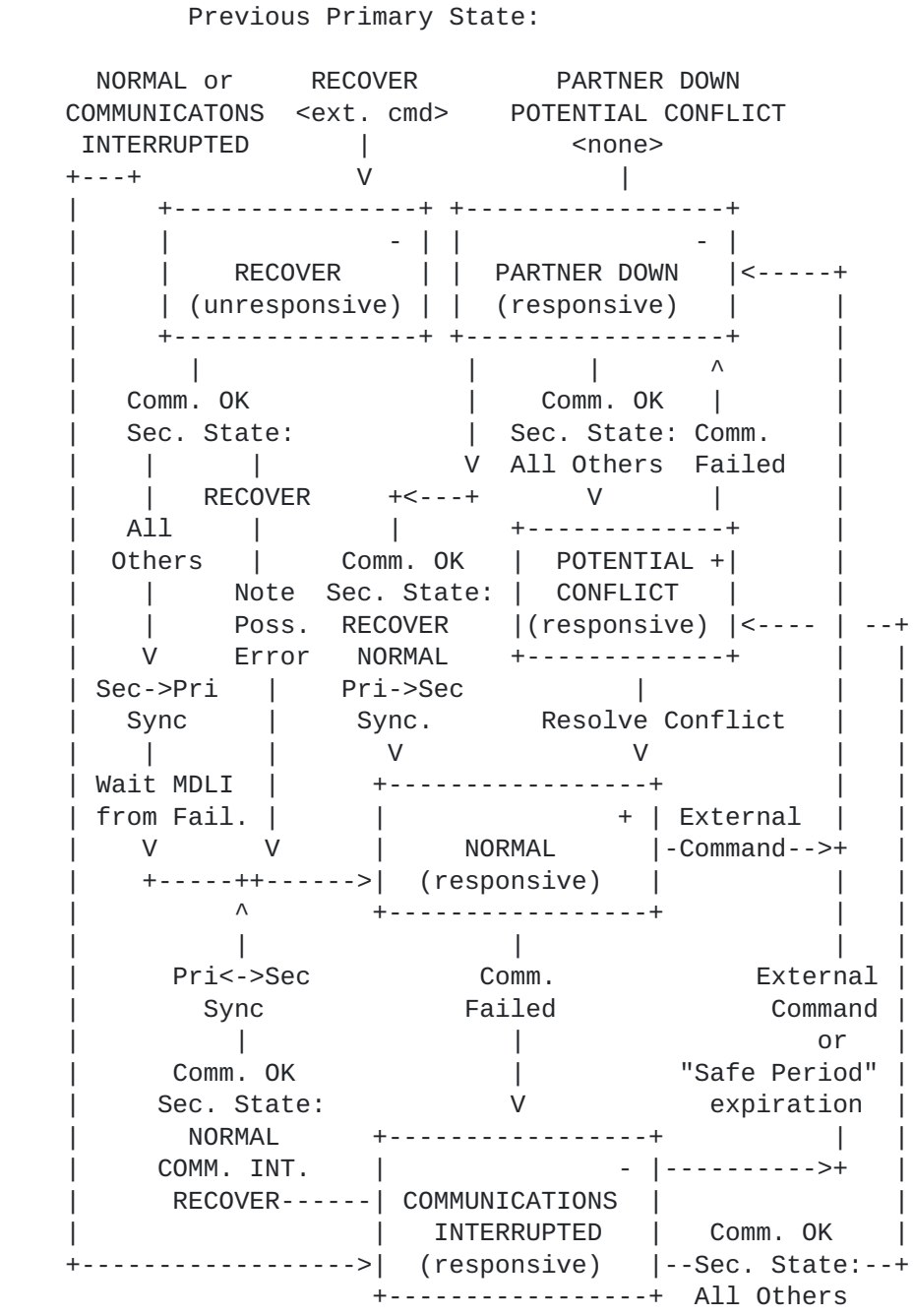

Figure 4.2-1: Primary server state diagram.

4.3. Primary Server in PARTNER-DOWN state

When it is in PARTNER-DOWN state, the primary server operates largely as does a normal DHCP server, with none of the special algorithms described below. In PARTNER-DOWN state the primary server MUST respond to DHCP client requests.

Any available IP address tagged as belonging to the secondary server (at entry to PARTNER-DOWN state) MUST NOT be used until the MDLI beyond the entry into PARTNER-DOWN state has elapsed.

The primary server MUST NOT allocate an IP address to a DHCP client different from that to which it was allocated at the entrance to PARTNER-DOWN state until the MDLI beyond the its expiration time has elapsed. If this time would be earlier than the current time plus the MDLI, then the current time plus the MDLI is used.

Two options exist for lease times, with different ramifications flowing from each.

If the primary server wishes the safe failover protocol to protect it from loss of stable storage in any state, then it should adhere to the restrictions on lease time discussed in section 3.2 and used in COMMUNICATION-INTERRUPTED state, section 4.6, below.

If the primary server wishes to forego the protection of the safe failover protocol in the event of loss of stable storage, then it need recognize no restrictions on actual client lease times while in PARTNER-DOWN state.

The primary server MUST poll the secondary server and attempt to establish communications and synchronization with it. It does this by attempting to update the secondary server with its existing binding information.

Once the primary succeeds in contacting the secondary server, the primary checks the state of the secondary server. If the state of the secondary server is RECOVER or NORMAL, then both servers have been running in such a way that duplicate IP address allocations are inhibited. In this case, the primary server updates the secondary server with its client binding information, and moves into the NORMAL state.

If, once contact has been established, the state of the secondary server is anything other than RECOVER or NORMAL then the primary server moves into the POTENTIAL-CONFLICT state.

4.4. Primary Server in RECOVER state

When primary server is initialized in the RECOVER state it expects to refresh its stable storage from an existing secondary server. In this state the primary server MUST NOT respond to DHCP client requests.

When the primary server succeeds in contacting the secondary server, if it determines that the secondary server is itself in the RECOVER state (which indicates that the secondary server has no existing client binding information), the primary server will move directly into NORMAL state after signaling some kind of an error (since some person had to explicitly start the primary server in RECOVER state to refresh its lost client binding information from the secondary, and the secondary had no state).

If the primary server determines that the secondary server is in any state other than RECOVER, then the secondary server has some client binding information that the primary server needs before it moves into the NORMAL state. The primary server will attempt to refresh its state from the secondary server, and it will remain in the RECOVER state until it is successful in doing so.

The primary server MUST remain in RECOVER state until a period of at least the MDLI has passed since the primary server was known to have failed. This is to allow any IP addresses that were allocated by the primary server prior to loss of primary server client binding information in stable storage to contact the secondary server or to time out.

DISCUSSION:

The actual requirement on this wait period in RECOVER is that it start when the primary server went down, not necessarily when it came back up. If the time when the primary server failed is known, then it could be communicated to the recovering server, and the wait period could be reduced to the MDLI less the difference between the current time and the time the server failed. In this way, the waiting period could be minimized.

4.5. Primary Server in NORMAL state

When in NORMAL state, the primary server takes the following actions to implement the Safe Failover protocol:

- Lease Time Calculations

As discussed in section 3.2, "control of lease time", the lease interval given to a DHCP client can never be more than the maximum delta lease interval greater than the acknowledged secondary server lease interval.

As long as the primary server adheres to this constraint, the specifics of the lease intervals that it gives to either the DHCP client or the secondary DHCP server are implementation dependent.
One possible approach is shown in section 3.2, but that particular approach is in no way required by this protocol.

- Lazy Update of Secondary Server

  After an ACK of a IP address binding, the primary server attempts to update the secondary with the binding information. The lease time used in the update of the secondary MUST be at least that given to the DHCP client in the DHCPACK. It MAY, however, be longer.

- Reallocation of IP Addresses Between Clients

  As specified in the Failover protocol, whenever a client binding is released or expires, an DHCPBNDDEL message must be sent to the secondary server.

  However, until a DHCPBNDACK is received for this message, the IP address cannot be allocated to another client.

4.6. Primary Server in COMMUNICATION-INTERRUPTED Mode

When in COMMUNICATIONS-INTERRUPTED state the primary server operates in such a way that correct operation is ensured even if the secondary server is still up and operational, but unable to communicate to the secondary server. When communications are reestablished between the primary and secondary servers, if both are still in COMMUNICATIONS-INTERRUPTED state, then the re-integration of their operation will proceed automatically and without human intervention. The protocol is designed to ensure that reintegration will proceed in an error free manner and that no actions taken by either server while in COMMUNICATIONS-INTERRUPTED state will cause problems during re-integration.

The primary server operates in COMMUNICATION-INTERRUPTED state as it does in NORMAL state.

However, since it cannot communicate with the secondary in this state, the acknowledged-secondary-lease-time will not be updated in

any new bindings. This is likely to eventually cause the actual-

client-lease-times to be the current-time plus the MDLI (unless this is greater than the desired-client-lease-time).

The primary server can simply queue updates to the secondary on communication interruption and stay in the NORMAL state. If, when communications with the secondary is reestablished, the secondary remains in the NORMAL state as well, then the queued updates for the secondary will simply be processed.

COMMUNICATION-INTERRUPTED state for the primary server is a signal that it has stopped queuing updates to the secondary, and is able to respond to a variety of possible secondary states.

It is anticipated that some alarm condition would be raised upon the transition from NORMAL state to COMMUNICATION-INTERRUPTED state.

Once the primary server has been in COMMUNICATION-INTERRUPTED state for a period equal to the safe-period, then it can (if configured to do so) transition into the PARTNER-DOWN state. An external command may also force a transition to PARTNER-DOWN state.

5. Secondary Server Operation

The operation of the secondary server is as specified in the Failover protocol [3], with the exception of the following situations and cases. Note that the secondary server responds to DHCP client requests only in the PARTNER-DOWN and COMMUNICATIONS-INTERRUPTED states.

5.1. Secondary Server Initialization

When the secondary server starts, there are three possibilities: it has never started before and therefore has no record of any previous state nor of any client binding information; it has started before and has a record of a previous state and possibly of some client binding information; it has started before, but failed catastrophically, and now has no record of any previous state (nor of any client binding information).

When the secondary server starts, if it has any record of a previous state, then if that state was NORMAL, COMMUNICATIONS-INTERRUPTED, or SYNC, it moves to COMMUNICATIONS-INTERRUPTED state. If that state was PARTNER-DOWN or POTENTIAL-CONFLICT, then it moves to PARTNER-DOWN state. In all other cases (both other previous states and the cases where there is no record of a previous state), the secondary server moves into the RECOVER state.

5.2. Secondary Server State Transitions

The server stays in the current state until all of the actions specified on the state transition are complete. If communications fails during one of the actions, the server simply stays in the current state and attempts a transition whenever the conditions for a transition are later fulfilled.

In the state transition diagram below, the "+" or "-" in the upper right corner of each state is a notation about whether communication is ongoing with the primary server. The legend "responsive" and "unresponsive" in each state indicates whether the secondary server is responsive to DHCP client requests in the respective state.

In the state transition diagram below, when communication is reestablished between the secondary and primary server, the secondary server must record the state of the primary server when the communications was reestablished. If the state of the primary server changes while communicating, then the secondary server moves through the communications-failed transition, and into whatever state results. At that time, it then immediately moves through whatever state transition is appropriate for the current state of the primary server.

All state transitions of the secondary server must be recorded in its stable storage, and thus be available to the server after a server restart.

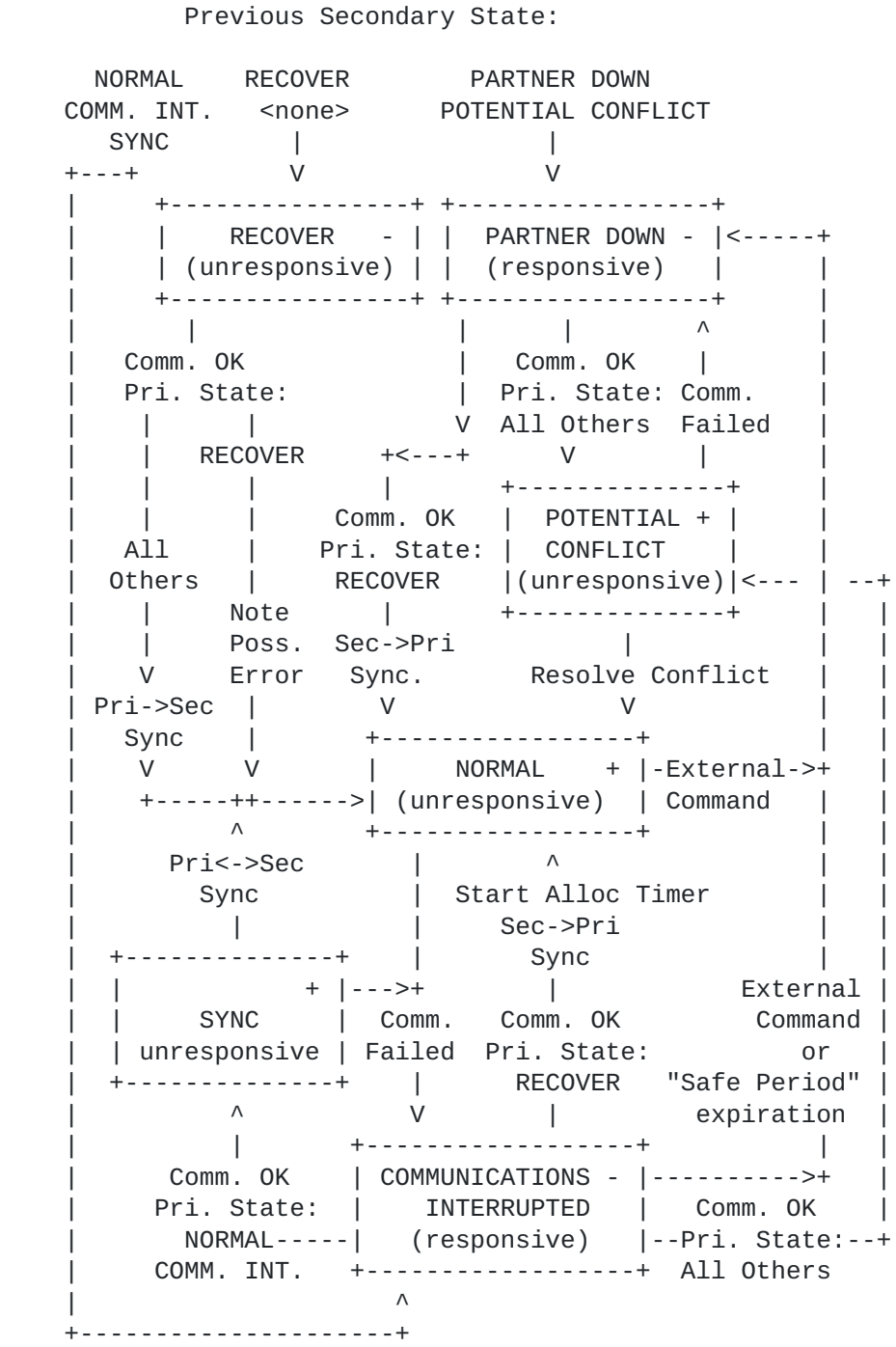

Figure 5.2-1: Secondary Server State Diagram.

5.3. Secondary Server in RECOVER state

The secondary DHCP server comes up in the RECOVER state when it has no record of any previous state (or that previous state was RECOVER).

It stays in this state until it establishes communication with the primary server, and is unresponsive to DHCP client requests in this state. Essentially it is idle until it can contact the primary server.

When it establishes communication with the primary server, it attempts to load its client binding database from that of the primary server using the techniques specified by the Failover protocol (note that at present, in [3], the Failover protocol does not specify this technique).

Once the secondary server's client binding database is refreshed from that of the primary, the secondary server moves into NORMAL state.

5.4. Secondary Server in NORMAL state

In normal state, the secondary server receives state updates from the primary server in DHCPBNDxxx messages. It records these in its client binding database in stable storage and then sends the corresponding DHCPBNDACK message to the primary server.

While in NOMRAL state, the secondary server MUST also acquire a series of IP addresses from the primary server to be used to satisfy DHCPDISCOVER requests from DHCP clients when in COMMUNICATIONS-INTERRUPTED state. See Section 6 for details of the acquisition process.

The secondary server periodically polls the primary server with the DHCPPOLL message. If it fails to receive a DHCPPRPL message in reply after a configured number of retries or some administratively determined time, the secondary server transitions into COMMUNICATION-INTERRUPTED state.

If an external command is received by the secondary server, it can move from NORMAL to PARTNER-DOWN state directly. Such a command might be sent when the primary server was removed from server, and an operator wanted the secondary server to take over immediately and completely from the primary server. (Note that the secondary server takes over from the primary server when in COMMUNICATIONS-INTERRUPTED state, but less completely than in PARTNER-DOWN state).

5.5. Secondary Server in COMMUNICATION-INTERRUPTED state

When in COMMUNICATIONS-INTERRUPTED state the secondary server operates in such a way that correct operation is ensured even if the primary server is still up and operational, but unable to communicate to the secondary server. When communications are reestablished between the primary and secondary servers, if both are still in COMMUNICATIONS-INTERRUPTED state, then the re-integration of their operation will proceed automatically and without human intervention. The protocol is designed to ensure that reintegration will proceed in an error free manner and that no actions taken by either server while in COMMUNICATIONS-INTERRUPTED state will cause any conflicts to occur during re-integration.

In COMMUNICATIONS-INTERRUPTED state, the secondary server responds to DHCP client requests.

When processing a DHCPREQUEST from a DHCP client, the secondary server MUST ensure that the client-lease-time is never more than the maximum-delta-lease-interval from the current-time, independent of the desired-client-lease-time.

When processing a DHCPRELEASE request from a DHCP client or the expiration of a lease, the secondary server must not reallocate the IP address to a different client. If the same client subsequently performs a DHCPDISCOVER request, the secondary server SHOULD offer it the previously used IP address.

When processing a DHCPDISCOVER request from a DHCP client, the secondary MUST allocate IP addresses from the list of IP addresses that it acquired from the primary server in RECOVER state. When it exhausts this list, it MUST stop responding to DHCPDISCOVER requests (except those it can satisfy by offering expired or released IP addresses to their previously bound clients).

The secondary server MUST continue to send DHCPPOLL messages to the primary server when in COMMUNICATION-INTERRUPTED state. If it receives a DHCPPRPL message in reply, the secondary server determines the state of the primary server. If the primary server is in NORMAL or COMMUNICATIONS-INTERRUPTED state, then the secondary server moves into the SYNC state.

If, however, the primary server is in RECOVER state, then the secondary server updates the primary server with its known client binding information, and moves into NORMAL state upon completion of that update.

If instructed to by an outside agency (e.g., an administrator), the

secondary server SHOULD move into PARTNER-DOWN state. Once the secondary server has been in COMMUNICATION-INTERRUPTED state for a period equal to the safe-period, then it may (if configured to do so) transition into the PARTNER-DOWN state in the absence of an external command.

5.6. Secondary Server in SYNCH state

The secondary server does not respond to DHCP client requests when in SYNCHRONIZING state.

DISCUSSION:

This is the entire reason for this states existence, otherwise the activities specified for this state could happen as part of a state transition from the COMMUNICATIONS-INTERRUPTED state to the NORMAL state. However, in the COMMUNICATIONS-INTERRUPTED state the secondary server responds to DHCP client requests. Having the secondary server respond to DHCP client requests during the synchronization process (and thus taking actions requiring further synchronization) seemed like a bad idea.

The secondary server synchronizes its information with the primary server while in SYNCHRONIZING state. Both primary and secondary servers may have information the other lacks because of operations performed while communications were interrupted.

During the synchronization process, the secondary server continues to poll the primary server with DHCPPOLL messages. If it fails to receive a reply, it moves back into COMMUNICATION-INTERRUPTED state.

When synchronization is complete, the secondary server moves into NORMAL state.

5.7. Secondary Server in PARTNER-DOWN state

The secondary server responds to DHCP client requests when in PARTNER-DOWN state.

Any available IP address which does not belong to the private pool established by the secondary server (at entry to PARTNER-DOWN state) MUST NOT be used until the MDLI beyond the entry into PARTNER-DOWN state has elapsed.

The secondary server MUST NOT allocate an IP address to a DHCP client

different from that to which it was allocated at the entrance to PARTNER-DOWN state until the MDLI beyond the its expiration time has

elapsed. If this time would be earlier than the current time plus the MDLI, then the current time plus the MDLI is used.

Two options exist for lease times, with different ramifications flowing from each.

If the secondary server wishes the safe failover protocol to protect it from loss of stable storage in any state, then it should adhere to the restrictions on lease time discussed in section 3.2 and used in COMMUNICATION-INTERRUPTED state, section 4.6, below.

If the secondary server wishes to forego the protection of the safe failover protocol in the event of loss of stable storage, then it need recognize no restrictions on actual client lease times while in PARTNER-DOWN state.

The secondary server continues to poll the primary server with DHCPPOLL messages. If the secondary server receives a reply, and the primary server is in the RECOVER state, the secondary server updates the primary server with all of the secondary's client binding information, and then moves into the NORMAL state.

If communications with the primary server are reestablished, and the primary server is in any other state but RECOVER, the secondary server moves into the POTENTIAL-CONFLICT state (as does the primary server).

5.8. Secondary Server in POTENTIAL-CONFLICT state

The secondary server enters POTENTIAL-CONFLICT state when the combination of its state and that of the primary indicate that a potential conflict of IP address allocation has occurred. There is no guarantee that such a conflict has occurred -- just the possibility. In this state each server compares its client binding information with that of the other server and any conflicts are resolved in an implementation dependent manner.

When (and if) the resolution process completes, each server moves into the NORMAL state.

6. Open Issues

A number of details remain to be worked out. They are as follows:

- Communicating Server State

A DHCP option which contains the DHCP server state must be defined for each server to use in messages sent between servers using the protocol.

- Subnet Level Granularity

  This protocol talks about a "server" being in one state or another implying that the entire server is in just one state (with respect to the Safe Failover protocol) largely because that is the easiest way to think about the basic approach of this protocol. The Failover protocol uses the same approach.

  However the intent is for this protocol to operate independently in each subnet address pool for which a primary and secondary server are defined. In this way, the "server" state really refers to the "subnet address pool" state. Once the general concepts behind this protocol are validated, the editing work to make it clear that it operates at subnet granularity will be performed.

- Secondary Acquisition of an IP Address Pool

  A number of potential ways exist to allow one DHCP server to acquire an IP address pool from another DHCP server. Perhaps the easiest way is to define a new DHCP option called "Return to specified IP address". This option only affects the calculation of the IP address to which DHCPOFFERs, DHCPACKs, DHCPNAKs are sent.

  This option will cause any reply packets from a DHCP server to be sent to the IP address in the option.

  Using this option, the secondary DHCP server can acquire IP addresses from the primary server and renew those leases as appropriate.

  The primary server recognizes that the secondary server IP address is contained in the "Return to" DHCP option, and tags these leases in its internal database.

  Once tagged in this way, the primary server will take two special actions with regard to these leases.

  1. The primary server will allow clients other than the secondary

server to renew or rebind the leases on these IP addresses when in any state.

2. When in PARTNER-DOWN state, and after the period equal to the MDLI has elapsed, the primary server will add any of these IP addresses that remain available to its pool of available IP addresses.

o Detailed Protocol Message Definition

Detailed protocol messages must be defined for this protocol.

7. Acknowledgments

Some of the ideas in this proposal came from the an initial inter-server draft authored by Ralph Droms, and he acknowledged contributions by Jeff Mogul, Greg Minshall, Rob Stevens, Walt Wimer, Ted Lemon, and the DHC working group. Additional ideas were generated while collaborating on a later version of the interserver draft with Bob Cole. Ideas have also been contributed by Mark Stapp, Brad Parker, and Glen Waters, and Ellen Garvey.

8. References

[1] Droms, R., "Dynamic Host Configuration Protocol", RFC 2131, March 1997.

[2] Alexander, S., Droms, R., "DHCP Options and BOOTP Vendor Extensions", Internet RFC 2132, March 1997.

[3] Rabil, G., Dooley, M., Kapur, A., Droms, R., "DHCP Failover Protocol", draft-ietf-dhc-failover-00.txt.

[4] Gudmundsson, Olafur, "Security Architecture for DHCP", draft-ietf-dhc-security-arch-00.txt.

9. Author's information

Kim Kinnear
American Internet Corporation
4 Preston Ct.
Bedford, MA 01730-2334

Phone: (781) 276-4587
EMail: kinnear@american.com