Scalable, High Speed, Internet Time Synchronization

Advanced Research Projects Agency
Contract DABT 63-95-C-0046

Quarterly Progress Report
1 June 1995 - 31 August 1995

David L. Mills
Electrical Engineering Department
University of Delaware

1. Introduction

The work reported during this interval consists primarily of preliminary revisions to the Network Time Protocol (NTP) formal specification to support new protocol modes as described in the contract proposal. The primary function of these revisions is to change the way that client protocol associations are handled to support multicast servers. The changes provide a mechanism for a local peer to calibrate its local clock using client/server mode in order to determine the local clock offset and round-trip network delay. This step requires a bidirectional protocol exchange, in which the local peer must send and receive packets. Upon calibration, subsequent multicast packets received (in listen-only mode) can provide an accurate estimate of the local time.

2. NTP Version 4 Protocol Changes (Preliminary)

The following text is designed to be merged at some later time in the NTP Version 3 specification document RFC1305 as part of the evolution to Version 4 of the protocol. Additional changes are expected in order to support the distributed modes and packet formats described in the contract proposal.

2.1 Peer Configuration

The NTP protocol is designed to support various associations which maintain the protocol and determine whether the local clock or the remote clock is the source of system timing. An association can be configured in advance or mobilized on arrival of a packet. A configured association is mobilized upon startup of the protocol and runs continuously. An ephemeral association is mobilized upon arrival of the request packet and demobilized when the reply packet is sent. This is normally the case with client/server modes, where a client mode request triggers a server mode reply and the association is demobilized at the server.

A persistent association is mobilized upon arrival of a packet, after which it continues operation with a timer running. This is normally the case with symmetric modes, where a symmetric-active mode packet mobilizes a symmetric-passive association. In symmetric passive mode, the local peer continues to probe the remote symmetric active peer until it fails to respond, at which time the passive association is demobilized. The resources claimed by a mobilized association are reclaimed upon demobilization of that association.

In symmetric active and passive modes, either the local or remote clock can set the local clock, depending on prevailing stratum assignments. The only difference between symmetric active and symmetric passive modes is that the active peer association is configured and runs continuously, whereas the passive peer association is dynamic and runs only as long as it receives packets from the active peer. If no packets are received within the timeout interval, the dynamic peer (the one in symmetric passive mode) demobilizes the association.

In client mode, the local peer association is configured and runs continuously. The remote peer association is ephemeral and is mobilized upon arrival of a client mode packet. A client mode peer can set its local clock as the result of the protocol exchange, while a server mode peer can never set its clock by this exchange. Another way of looking at it is to say that the local clock can be set only as the result of a persistent association; symmetric mode and client mode peers maintain persistent state, while server mode peers do not.

Associations are normally configured only for active modes: symmetric active, client and multicast. These correspond to modes of the same name. An association in one of these modes is always mobilized and does not change modes during operation. On the other hand, an ephemeral or persistent association can only be mobilized in a passive mode: symmetric passive or server, except in the case of multicast, as described later.

Note that a configured association is specific to a particular source and destination address and mode. Packets will be sent continuously (possibly rate attenuated) with given source and destination addresses determined by interchanging the source and destination addresses of the association. For configured associations, the mode of transmitted packets follows the mode of the association. Certain combinations of local peer configuration and remote peer configuration are normal, others are abnormal but still work, and others are errors. The text below describes each of these in detail.

In the interests of liberal behavior, it is useful to consider all combinations of configuration, normal and otherwise. Ordinarily, only one of two communicating peers is configured and the other is not. It is an error (results in ambiguous association matching) to configure a peer with more than one symmetric active association for the same remote peer. It is an error to configure a peer with more than one client association for the same remote peer. However, it is not an error (although of dubious utility) to configure a peer for a single symmetric active configuration along with a single client association for the same remote peer.

A symmetric-active association mobilizes a persistent symmetric-passive association in its peer and a client mode association mobilizes an ephemeral server association in its peer. However, for reliability reasons the local and remote peers may be configured in a symmetric active mode. Such configurations allow either peer to set its local clock from the other peer. It could also happen that one peer is configured in symmetric active and the other in client mode, or both could be configured in client mode. By the rules below, if both are configured in symmetric active mode, then only one association in each peer will be mobilized. However, if both are configured in client mode, or if one is configured in symmetric active mode and the other in client mode, distinct server associations are mobilized. In the extreme case where each peer is configured in a symmetric active and a client association, three associations are mobilized, a persistent one for the symmetric modes plus an ephemeral association which is mobilized in either peer upon arrival of a client mode packet from the other. These configurations, while not generally useful, are not considered errors by the state machine.

2.1.1 Configured Associations

A configured association is specified by the following data:

Mode: Symmetric active, client or multicast. These names may be equivalent under other names, like server, peer, etc.

Source address: The IP address of a remote peer or a designated multicast (group) address. This is used as the destination address for transmitted packet. A configured association with a multicast source address can only be used to send packets; arriving packets are never matched to these associations.

Destination address: The IP address of a local interface. This is used as the source address for transmitted packets.

Authentication key identifier (optional): If absent, authentication is not used; if present, all packets received on this association must carry the given key identifier and crypto-checksum matching this key.

Polling controls: Maximum and minimum poll intervals, maybe other strategic information as well.

Selection controls: Prefer keyword, maybe other strategic information as well.

2.1.2 Global Configuration

Some configuration information applies to the peer population at large, such as access controls, key identifiers and other things. These include:

Encryption keys: Each peer contains a table of cached key media indexed by key identifier. Some means is necessary to incorporate these data in the peer data base. The design might include also a hot-list, where some cached keys can be marked valid/invalid without physically removing the key media. If so, provisions should be made to enable/disable individual key identifiers by system management means. If the peer is to be synchronized to a non-configured association from a remote configured association, a list of trusted key identifiers must be provided.

Multicast addresses/groups: It is the intent that multicast capability be turned off by default in the base software distribution and must be specifically enabled in the configuration file for the daemon. The particular multicast address(es) the daemon should listen on are specified in the configuration file entries for multicast associations. For the purposes of the protocol and state machine, all multicast addresses (groups) are processed in the same manner and there is no distinction between them anywhere in the protocol machine.

Promiscuous bit: It is the intent that the capability of the local peer to update the system clock be enabled only under specific circumstances and associations. These circumstances are:

1. A configured association where peer packets are access controlled, authenticated and pass all sanity checks. Access control and authentication are optional in this case and required only if specified in the configuration file entry for the association.

2. A non-configured association where peer packets are access controlled, authenticated and pass all sanity checks. Access control is optional and authentication is required in this case. For this purpose, a set of default filters, key identifiers and key media are specified in the global configuration data.

3. If specifically enabled by the promiscuous bit, then a non-configured association where packets are access controlled, authenticated and pass all sanity checks. Access control and authentication are optional in this case and required only if specified in the configuration file. For this purpose, a set of default filters, key identifiers and key media are specified in the global configuration data.

In all cases, the NTP rules must be followed, i.e., the local clock can be set only on receipt of a symmetric mode (active or passive) or server mode packet in response to a symmetric mode (active or passive) or client mode transmitted packet, respectively. If a multicast capability has been configured, a received multicast packet can also set the local clock, but only if the promiscuous bit is set.

2.2 Preliminary Packet Processing

Before an arriving packet is matched to an association, certain sanity checks must be performed. These are designed to discard the packet before any processing is done, in order to avoid pollution of state variables which are vulnerable to accidental or purposeful protocol attacks. The current suite of sanity checks include:

Test1: The UDP checksum and port fields must be correct and the packet length must be sufficient to cover the NTP header and authentication fields, if present.

Test2: The packet must pass access-control checks, if implemented and enabled in the configuration file.

There is some debate whether authentication checks should be done here. The intent of the design is that authentication should be separately enabled by configured association; that is, it may happen that some associations are to be authenticated, others not. It may happen that future authentication schemes, such as that proposed for IPv6, provide authentication services based on IP source and/or destination addresses. However this is considered unlikely, since it implies in some sense a multi-level security architecture. In what is believed the more common case where security/authentication would be an all-or-none service, an appropriate processing check could be:

Test2a: If the authentication feature has been enabled, the packet must have the correct authentication field format, key identifier, key and be properly authenticated.

If authentication is implemented on a per-association basis, as in the current model, the actual authentication process must be delayed until the connection is correctly matched and the key identifier is determined.

2.3 Association Matching

Configured associations are mobilized at protocol startup; dynamic associations can be mobilized upon arrival of a packet in symmetric (active) mode, client mode or multicast mode. Normally, packets of different modes match different associations, even if other associations with the same source and destination addresses are present, but with different modes. In some cases, packets of different modes, but the same source and destination addresses, match the same association. These cases are detailed below.

The association table contains the mode and source and destination addresses of configured associations, as well as those persistent associations that happen to be mobilized. The source and destination fields correspond to those in an arriving packet. These are always unicast addresses, even if the arriving packet has a multicast destination address. In the case of multi-homed servers or clients which results in each peer being assigned multiple addresses, the association table may contain distinct associations for each combination of these modes and addresses, even if for the same peer.

An arriving unicast packet is matched to the mode, source address and destination address in the association table. In some cases it is possible for different associations to have the same source and destination addresses, but different modes. However, in all cases, either none or exactly one match can occur for any incoming packet. Note that a packet can never match a configured multicast association (explained later). If a match is not found, a new association is mobilized before the packet is processed further. The packet is delivered to the protocol machine instantiated for that address combination. Each distinct match is an instantiation of a separate protocol machine with separate state variables; normally, state variables from one association are transparent with respect to all other instantiations.

An arriving multicast packet is treated as a special case. In this case the interface on which the packet arrived and its address is assumed known. For the purposes of association matching, this address replaces the destination multicast address, so that only a single association may be matched. Thus, a packet with a multicast destination address results in a new association only with respect to the interface address upon which it arrived. As mentioned earlier, the packet cannot match a configured multicast association since configured multicast associations are valid only in a transmit-only mode and hence do not receive any packets.

The reason for this behavior is that it will probably be the case that multicast packets arriving on different local interfaces will have travelled different routes in the network with different propagation times. Subsequently, when newly mobilized associations attempt to determine the one-way propagation time, the time must be determined separately for each distinct route in the network. In case a multicast route flaps back and forth between two local interfaces, this behavior may be suboptimal. This problem may require experimental resolution.

In order to understand the somewhat idiosyncratic scheme involved in matching associations, it may be useful to summarize the ground rules for each mode.

Symmetric (Active) Mode

Symmetric modes are designed so that either peer can become the source of timing for both peers. In the current version of NTP (version 3), the symmetric active mode is used if the configuration file specifies “peer”. The only packets that will be delivered to a symmetric active configured association are symmetric active and symmetric passive mode packets matching the same source and destination addresses.

Either one or both the local and remote peers can operate in symmetric active mode, or one can operate in symmetric active mode and the other in symmetric passive mode. It is possible to have a situation where both the peers operate in symmetric passive mode as in the case of a protocol error or when peers are suddenly reconfigured. Section 2.3.3 describes this in greater detail.

Client Mode

Client mode is designed so that the local clock can be set from the remote peer, but the remote peer cannot set its clock from the local peer. In version 3 of the NTP software, this mode is used if the configuration file specifies “server”. The only packets that will be delivered to a client mode association are server mode packets matching the same source and destination addresses.

Note that in principle it is possible to configure a symmetric active association as well as a client association with the same source and destination addresses. In such cases, the remote peer will mobilize both a persistent symmetric passive association and an ephemeral server association, each of which operates independently of the other. The utility of this might seem questionable, but the state machine would not consider it an error.

Symmetric Passive Mode

If a configured symmetric active association is mobilized in the peer, both symmetric active and passive packets are delivered to it. If not, a persistent symmetric passive mode association is mobilized upon receipt of symmetric active packet. Only symmetric active packets matching the source and destination addresses are delivered to a symmetric passive association. Symmetric passive packets are ignored, unless there is a symmetric active association to deliver the packets to.

Server Mode

A server mode association is dynamically mobilized upon arrival of a client mode packet. This association is always ephemeral; that is, it exists only for the duration to receive the client mode packet and generate the reply, after which the association is demobilized. Since the client/server interaction is in principle atomic, no other packets will be delivered to the association during its lifetime.

Multicast Client Mode

A multicast client association is mobilized upon the arrival of a multicast mode packet when (a) there is no configured client association and (b) there is no existing multicast client association matching the packet source and destination addresses. This mode is always persistent; that is, it exists from the arrival of the first multicast packet until the association times out for some reason (see below).

An arriving multicast mode packet can be matched only to a multicast client association. If there is already a client mode association matching the same source and destination addresses, no association is mobilized and the packet is dropped. This behavior is necessary, since there could be an ambiguity when a server mode packet arrives, which could be matched to either a client association or a multicast client association operating in calibrate mode (see below).

A multicast client association can be in two states, calibrate or listen. In calibrate state, the association operates as in client/server mode; that is, it listens for both server mode and broadcast mode packets in order to calibrate the one-way delays. When calibration is complete, the association switches to listen state and listens only for multicast mode packets while discarding server mode packets.

2.4 Summary of Packet Processing

In summary, the following operations are performed for each packet arrival (by mode):

Symmetric Active: If a configured symmetric active association exists and matches the packet source and destination addresses, route the packet to it. Otherwise, mobilize a symmetric passive association with packet source and destination addresses and route the packet to it.

Symmetric Passive: If a configured symmetric active association exists and matches the packet source and destination addresses, route the packet to it. Otherwise, drop the packet.

Client: Mobilize an ephemeral server association with packet source and destination addresses and route the packet to it. Demobilize the association after sending the server mode reply.

Server: If a configured client association exists and matches the packet source and destination addresses, route the packet to it. Otherwise, if a persistent multicast client association exists and matches the packet source and destination addresses, route the packet to it. Otherwise, drop the packet.

Multicast: If a configured client association exists and matches the packet source and destination addresses, drop the packet. Otherwise, if a persistent multicast client association exists and matches the packet source and destination addresses, route the packet to it. Otherwise, mobilize a persistent multicast client association with packet source and destination addresses, set the state to calibrate, and route the packet to it.

2.5 Packet Validation

Once the association has been matched, the option bits, authentication key identifiers and whatnot are available. At this point a number of sanity checks can be made on the header. If any of these checks fail, the packet is dropped without prejudice.

Test3: If authentication is enabled for this association, the packet must be properly authenticated.

Test4: The peer clock must be synchronized (leap bits other than 11) and the interval since the peer clock was last updated (peer.org - peer.reftime) must be less than an architecture constant.

Test5: The peer stratum must be less than or equal to the local stratum. Note that, in the case where a low-stratum remote peer has just failed and all other remote peers have greater stratum, the local peer will eventually time out and re-synchronize to the remaining remote peers.

Test6: The peer root delay and peer root dispersion must be within respective architecture constants.

If the above sanity checks are satisfied, an association is mobilized. However, when a persistent association is first mobilized, it does not have sufficient information to synchronize until a subsequent exchange of packets with the remote peer. The association can be synchronized and the local clock updated only if the following sanity checks are satisfied.

Test7: The transmit timestamp must not match the last transmit timestamp received from the same peer; otherwise the packet could be an old duplicate.

Test8: The originate timestamp must match the last originate timestamp sent to the same peer; otherwise the packet might be out of order, bogus or worse.

Test9: Both the originate and receive timestamps must be nonzero; otherwise, the association might not be synchronized.

If Tests 1-9 succeed, the local clock offset, round-trip delay and dispersion are calculated.

Test10: The calculated delay and dispersion must be within respective architecture bounds.

2.6 State Machine

All association operations are managed by a state machine with defined input events, output actions and state transitions. There are two input events: a packet arrival and a timeout. A packet arrival is classified by packet type (symmetric active, symmetric passive, client, server or multicast). Output actions and state transitions are defined informally by the following descriptions. Formal definitions will be defined in the specification to follow.

State machine operations depend on whether the association is configured or not and, if configured, in what mode: symmetric, client or multicast. State transitions depend on the mode of the arriving packet and the mode of the configured association. Associations configured in an active mode are always mobilized, in that a timer is running and packets are sent (in the given mode) at each timeout. The timeouts, which are of varying durations, are managed by the protocol. If upon arrival, a packet is not matched to an association, a new one is mobilized as described previously.

When a multicast client association is mobilized, it is initialized in calibrate state. In this state, both multicast and server mode packets matching the source and destination addresses are routed to the association. The calibrate mode persists until the following requirements are met: (a) sufficient data is accumulated in client/server mode to justify setting the system clock (this normally requires at least four clock offset/round-trip delay measurements), and (b) at least one multicast message (other than the initial one that mobilizes the association) is received.

In the calibrate state, reachability is updated using the server mode reply packets in the same way as client/server mode. A reachability timeout in the calibrate state causes an immediate transition to listen state, with the reachability register initialized to one; that is, in a state such that an additional seven poll intervals may occur until a reachability timeout in listen state.

When both the above conditions are met, the multicast client association switches to listen state. In this state it ceases to send client mode messages or process server mode replies (they are dropped). Each arriving multicast message updates the peer state variables (reachability and offset). A reachability timeout in the listen state causes the association to be demobilized.

Summary and Conclusions

It is the intent that the above processing rules replace the existing rules specified in RFC1305. These rules will in principle result in no changes in the operation of existing NTP servers and clients and should in general be transparent to all prior versions of the protocol. The new rules do specify detailed behavior in configurations that formerly were considered erroneous, but generally resulted in correct clock synchronization. Of primary utility is the detailed specification of the multicast client mode, which previously was implemented in an ad-hoc manner. The above rules specify precisely how such implementations should be built and how they should respond.

3. Plans for Next Quarter

During the next quarter, we expect to revise the NTP protocol daemon for Unix to conform to the above rules, in effect cleaning out some old deadwood and ad-hoc protocol code, and replacing with new code that behaves in an orderly, deterministic way. We also expect to begin work on packet format revisions to support the NTP distributed mode, in which each of possibly many servers can multicast a vector of received timestamps to other participants in the subnet in order to determine clock offsets and round-trip times relative to all clocks in the subnet. Finally, we expect to begin work on algorithms to automatically configure a subnet based on defined metrics and constraints as described in the contract proposal.