

Internet Engineering Task Force
INTERNET-DRAFT
Expires July 1998

Dave Thaler
Merit
24 January 1997

Globally-Distributed Troubleshooting (GDT): Protocol Specification
<[draft-thaler-gdt-00.txt](#)>

Status of this Memo

This document is an Internet Draft. Internet Drafts are working documents of the Internet Engineering Task Force (IETF), its Areas, and its Working Groups. Note that other groups may also distribute working documents as Internet Drafts.

Internet Drafts are valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet Drafts as reference material or to cite them other than as a "work in progress".

To learn the current status of any Internet-Draft, please check the "1id-abstracts.txt" listing contained in the Internet-Drafts Shadow Directories on ftp.is.co.za (Africa), nic.nordu.net (Europe), munnari.oz.au (Pacific Rim), ds.internic.net (US East Coast), or ftp.isi.edu (US West Coast).

Abstract

This document describes a protocol for "globally-distributed troubleshooting" (GDT). GDT automates, where possible, the process of problem reporting and referral between customers and Internet Service Providers (ISPs), as well as between ISPs. GDT also provides an automatic mechanism for periodic status reports, and allows an ISP to make information (such as expected time to repair) on a current problem readily accessible to those directly affected by it, without requiring human intervention.

Draft

GDT Protocol Specification

January 1997

1. Introduction

1.1. Purpose

The GDT protocol automates, where possible, the process of problem reporting and referral between customers and ISPs, as well as between multiple ISPs. GDT provides an automatic mechanism for periodic status reports, and allows an ISP to make information (such as expected time to repair) on a current problem readily accessible to those directly affected by it, without requiring human intervention.

1.2. Terminology

This document uses the same words as [RFC 2119](#) for defining the significance of each particular requirement. These words are:

MUST:

This word or the adjective "required" means that the item is an absolute requirement of the specification. An implementation is not compliant if it fails to satisfy one or more of the MUST requirements for the protocols it implements.

SHOULD:

This word or the adjective "recommended" means 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.

MAY:

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

This document also uses the following technical terms:

agent:

An entity supporting the GDT protocol. There are three levels of sophistication of agents: simple agents, experts, and expert location servers (ELSSs).

network element, or element:

A logical network object (e.g., an Ethernet, a process, or a TCP

Expires July 1998

[Page 2]

Draft

GDT Protocol Specification

January 1997

connection).

area of expertise:

An identifier representing a specific set of network elements.

capability:

An (access mode, area of expertise) pair representing the knowledge and permissions necessary to troubleshoot problems with a set of network elements. Access mode may be either diagnosis-only or diagnosis-and-repair.

expert:

An agent with one or more capabilities.

expert location server (ELS):

An agent which has the responsibility of locating experts with capabilities covering a given problem.

hypothesis:

A guess that a specific problem, identified by a (problem type, network element) pair, exists. Each hypothesis is submitted to an expert with an area of expertise covering the given network element.

problem type:

A general classification of problems. Problem types are classified as follows.

Superficial problem types:

lowH

The element is experiencing degraded performance (low health). This is the most general problem type, of which all others are specific instances.

highU

The element is experiencing degraded performance due to unusually

high utilization (congestion), as opposed to the element being down or malfunctioning.

Intermediate problem types:

lowerH

Degraded performance is due to degraded performance of an element upon which the current element depends.

Expires July 1998

[Page 3]

Draft

GDT Protocol Specification

January 1997

downstreamH

Degraded performance is due to degraded performance at one or more downstream elements.

higherU

Unusually high utilization is due to unusually high resource demands from a higher layer.

upstreamU

Unusually high utilization is due to unusually high throughput demands from one or more upstream elements.

We assume that every problem observed is ultimately caused by the presence of one or more of the following primary problem types at one or more network elements:

highD

The element is demanding too many resources from other elements.

lowC

The element's capacity is insufficient to efficiently support normal operation.

badHW

The hardware or software implementation is malfunctioning.

status:

Problem status values are classified as follows:

Unconfirmed

A test is in progress to confirm that the problem exists.

Diagnosis-Deferred

The test for the existence of the problem has been deferred until a later time.

Rejected

The test failed, indicating that the problem does not exist, and the problem state will shortly go away.

Indeterminate

Either no test is known, or the test was inconclusive, and the problem state will shortly go away.

Expires July 1998

[Page 4]

Draft

GDT Protocol Specification

January 1997

Confirmed

Existence of the problem is acknowledged, and causes are currently being investigated.

Covered

Another problem is known to be causing the current problem, and hence the expert is waiting for the cause to be repaired.

CantRepair

No repair is possible for the problem.

Isolated

A repair and retest are in progress.

Repair-Deferred

The repair has been deferred until a later time.

Repaired

The problem was successfully repaired, and the problem state will shortly go away.

WentAway

The problem disappeared, and hence the problem state will shortly go away.

Retesting

All previously-confirmed causes are gone, and a retest is in progress.

Deleted

No state for the problem exists.

[2.](#) Protocol Overview

Problem reporting and resolution is a multi-phase procedure. In the first phase, a Hypothesis message is submitted to an expert whose area of expertise includes the network element which is experiencing problems. Since there is no restriction on where such hypotheses may originate, the expert next attempts to verify the existence of the reported problem. If the problem is confirmed, the expert then generates additional hypotheses about potential causes, which are in turn submitted to appropriate experts. This process continues until one or more problems are confirmed which have no causes. Repairs are then requested for these problems. If repairs can not be immediately

Expires July 1998

[Page 5]

Draft

GDT Protocol Specification

January 1997

initiated, repairs are then attempted at their immediate effects, and so on back down the cause tree.

[2.1.](#) Agent Requirements

There are three (logical) tables which must be present in some form in all GDT agents. While there may be more efficient ways of representing the same information, example representations are given below.

Capability Table

The entry for each known capability has, associated with the capability itself, the IP address of an expert, and a Capability-Timer. This table is initialized upon startup to hold the agent's own capabilities (if any), plus at least one all-encompassing (default) capability for a "parent" ELS. This table SHOULD be saved to stable storage.

Hypothesis Table

This table holds information on problems which the agent believes may exist, and is waiting for them to be repaired or otherwise resolved. The entry for each (proposed problem type, network element) pair has, associated with it, the last known status and sequence number received from a remote expert, an expert list, and a Expert-Timer. The expert list is a set of experts with capabilities covering the given hypothesis.

Reliable Message Table

This table keeps state for those messages which must be transmitted reliably. The entry for each message to be sent reliably has, associated with a given combination of message, destination IP address, and port number, a transmission count and an Ack-Timer.

Any GDT agent may also be an expert. In addition to fulfilling all of the requirements for simple agents, GDT experts have the following additional table:

Problem Table

This table holds information on current problems, within the agent's areas of expertise, on which the agent is currently working to diagnose or repair. The entry for each active (problem type, network element) pair has, associated with it, a problem status value, an origin list, a cause list, and a Deletion-Timer. Each entry in the origin list contains an IP address and port number from which a Hypothesis about this problem was received, and an Origin-Timer. The

Expires July 1998

[Page 6]

Draft

GDT Protocol Specification

January 1997

cause list is a set of hypotheses in the hypothesis table which are potential causes of the given problem, each with an associated Retracted-bit.

Finally, some experts may be Expert Location Servers (ELS's). In addition to fulfilling all of the requirements for experts, ELS's have the following additional table:

Child Table

This table holds information on the ELS's current children in the hierarchy. The entry for each child has, associated with the child's address, a maximal prefix length, an active prefix length, and a Child-Timer.

2.1.1. Advertising Capabilities

An expert may advertise capabilities with an access mode of either diagnosis-only or diagnosis-and-repair. To advertise a capability with an access mode of diagnosis-only for some area of expertise, the expert MUST fulfill the following requirements for any element within the area of expertise:

- (1) Be able to perform a test, or request that an operator perform a test, to Confirm or Reject (or report that the test was Indeterminate) a Hypotheses that the given network element is experiencing each of the following problem types: lowH, highU, highD, lowC. The expert MAY also have one or more tests for badHW.
- (2) Be able to obtain the list of elements (if any) above the element (i.e., those elements which depend upon the given element), or report that the list resolution was Indeterminate.
- (3) Be able to obtain the list of elements (if any) below the element (i.e., those elements upon which the given element depends), or report that the list resolution was Indeterminate.
- (4) Be able to obtain the list of elements (if any) upstream of the element (i.e., those adjacent elements which send data to the given element), or report that the list resolution was Indeterminate.
- (5) Be able to obtain the list of elements (if any) downstream from the element (i.e., those adjacent elements to which the given element sends data), or report that the list resolution was Indeterminate.

Expires July 1998

[Page 7]

Draft

GDT Protocol Specification

January 1997

To advertise a capability with an access mode of diagnosis-and-repair, the expert MUST fulfill the following requirement in addition to those listed above:

- (1) Be able to request that an operator perform, report on automatic performing (if the element is self-correcting) of, or itself perform each of the following types of repairs:

- o When one element is imposing too many resource demands on another element (higherU, upstreamU, highD), the amount of resources available to it should be decreased.
- o When capacity is insufficient to meet normal demand (lowC), the capacity should be increased.
- o When an element at a lower layer is faulty (lowerH), the higher layer element may be reconfigured so that it no longer depends on the faulty element (e.g., using a backup link instead).
- o When a hardware fault or software bug exists (badHW), the problem should be corrected, or the element replaced.

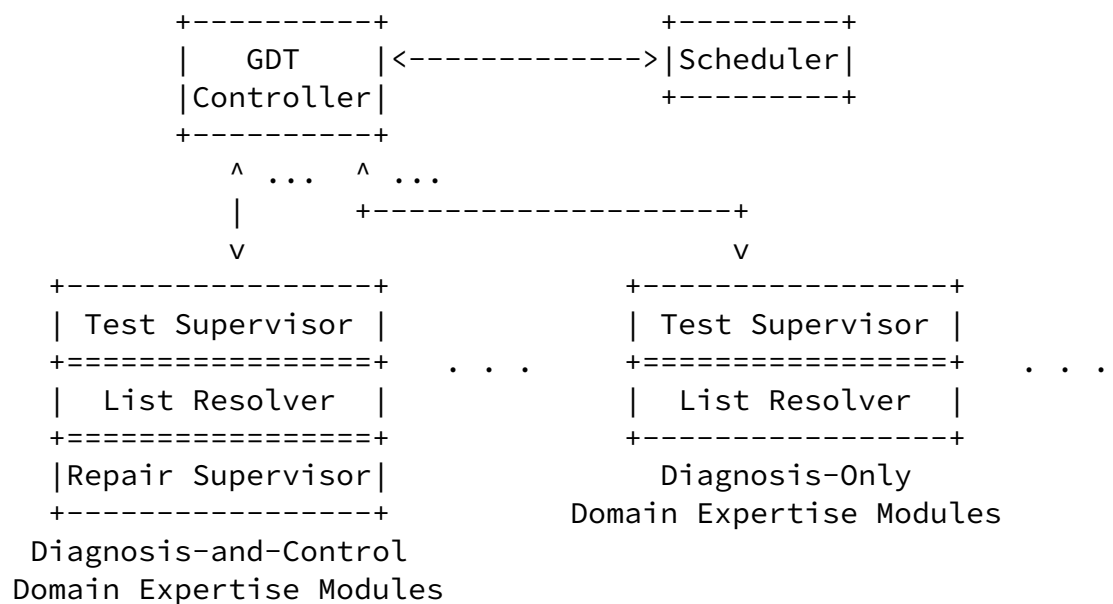


Figure 1: Logical Architecture of a GDT Expert

Figure 1 shows the logical architecture of a single GDT expert. This document specifies the behavior of the (logical) GDT Controller module. A (logical) scheduler is used to schedule tests and repairs. For each

of its capabilities, if any, the agent has a (logical) domain-expertise

module which is able to conduct tests, resolve lists of related elements, and potentially supervise repairs.

[2.2.](#) Expert Location

Expert location servers are organized into a hierarchy by location, where the leaves are actual experts. At startup time, experts and location servers locate an appropriate parent, and add themselves to the hierarchy. Agents starting up also locate one or more appropriate servers and install default entries for these servers in their local capability cache.

To do this, each expert location server is configured with a "maximal policy prefix", denoting the range of addresses of experts for which it is willing to be a parent. (This allows some policy control over the server hierarchy.) Root servers should have a maximal policy prefix of 0/0. The length of the associated mask will be called the "maximal mask length".

A hierarchy of experts is then constructed dynamically subject to the policy constraints. The hierarchy thus constructed has the following properties which prevent loops in the hierarchy:

- o Every expert has an "active mask length" which is the longer of: its own maximal mask length, and its parent's active mask length + 1.
- o Every child has an address within its parent's active policy prefix.

An analysis of this scheme can be found in [\[1\]](#).

To locate an initial server, an agent SHOULD first attempt to locate a nearby server using an expanding-ring multicast search over a well-known group address to which all experts listen. If this search fails, the agent then uses a manually-configured list of servers, such as a list of local servers.

[2.3.](#) Reliable Message Transport

All messages are sent using UDP. Experts listen on a well-known port number, while clients may use any local port number. Since GDT is a soft-state, connectionless protocol, some messages must be retransmitted to achieve reliability. Hypothesis and Retract messages are always

acknowledged by receiving Status-Report messages. Status-Report messages are sometimes sent reliably, and are acknowledged by Status-Ack messages. Each message also contains a 16-bit sequence number to detect out-of-order packets.

When a message is to be sent reliably, an entry for it is created in the (logical) reliable message table. The message will then be transmitted up to three times at intervals of [[Ack-Timeout](#)] seconds before giving up.

[3.](#) Basic Behavior

In this section, we describe the detailed protocol which all GDT agents must perform. Packet formats are described in [Section 6](#).

[3.1.](#) Startup

When an agent first starts up, it must determine one or more expert location servers and create default entries in its capability cache for them. To do this, it SHOULD first join the RootAdv group and then perform an expanding-ring multicast to locate a nearby server; if this fails, it may wait until an Am-Root message is received and use the root.

To perform an expanding-ring search, the agent joins the GDT-Server-Location group, and multicasts periodic Whois-Server messages to the All-GDT-Servers group with increasing TTLs until either a maximum TTL is reached, or an Am-Server message is received from a legal parent.

When an Am-Server message is received from a legal parent (and the agent is not an ELS), the agent may leave the GDT-Server-Location group. The source of the message is then chosen as the agent's parent, and the rules in [Section 3.2](#) are followed.

[3.2.](#) Setting One's Parent

Whenever an agent changes its parent, any existing "default" entries are first removed from the capability table. If the new parent is not null, then a default entry for it is added to the capability table.

Draft

GDT Protocol Specification

January 1997

[3.3.](#) Detecting a problem and sending a Hypothesis

Problems are detected in an implementation-dependent manner (e.g., by other protocols). When a problem is observed, the agent identifies the network element experiencing a problem (element naming is discussed in a separate document [\[2\]](#)), and checks its hypothesis table for an existing hypothesis entry.

If an existing entry is found:

- (1) If the agent is an expert, and the hypothesis was proposed by an expert as the cause of a problem in the problem table, processing continues (for that problem only) as if a Status-Report had been received with the last known status of the hypothesis entry, using the rules in [Section 4.7](#).
- (2) If the agent is not an expert, then the problem has already been reported, and nothing further need be done at this point.

If no entry is found in the hypothesis table:

- (1) If an entry exists in the reliable message table for a Retract of the same Hypothesis, the entry is deleted.
- (2) A new hypothesis entry is created with a last known status of Unconfirmed.
- (3) The agent then checks its capability table for a list of one or more experts whose capabilities cover that element, and places them in the expert list of the hypothesis entry. A capability covers an element if all required attributes match, and no optional attributes conflict.
- (4) If no experts were found, processing continues as if a Status-Report had been received with a status value of Indeterminate (which we will refer to as an SR:Indeterminate message) had been received (Sections [3.6.2](#) and [4.7.2](#)).
- (5) If any experts were found with diagnosis-and-repair capability, a Hypothesis message is reliably sent to one of those experts. If all experts found had diagnosis-only capability, a Hypothesis

message is reliably sent to one of them. To choose among multiple equivalent experts, the following algorithm is employed (an analysis of which can be found in [3]):

- (a) Compute the CRC-32 checksum (X) of the network element identifier (as discussed in [2]).
- (b) For each possible expert address E, compute a value:
$$\text{Value}(X, E_i) = (1103515245 * ((1103515245 * E + 12345) \text{ XOR } X) + 12345) \bmod 2^{31}$$
- (c) The expert with the highest resulting value is then chosen as the destination expert. This algorithm ensures that all agents reporting the same problem use the same destination expert as long as they see the same set of expert capabilities, while dividing up problems among equivalent experts.

[3.4.](#) Sending a Retract

Any agent terminating nicely SHOULD retract all outstanding hypotheses. An agent MAY also retract any outstanding hypothesis at any time. (For example, as discussed in [Section 4.7.3](#), experts retract hypotheses as a result of another hypothesis being confirmed.)

When an agent wishes to retract a hypothesis, it sends a Retract message to the same expert to which it sent the Hypothesis message. Unless the agent is terminating, this Retract is sent reliably. The corresponding entry is then deleted from the Hypothesis Table. If an entry exists in the reliable message table for the original Hypothesis message, that entry is deleted as well.

[3.5.](#) Receiving a Redirect

When an agent receives a Redirect containing a list of capabilities, it MAY add the included capabilities to its capability table.

The agent then searches for a hypothesis entry which matches the problem

type and network element in the Redirect. If none is found, the Redirect is dropped. Otherwise, the redirect origin is removed from the hypothesis' list of experts, the experts listed in the Redirect are added to the list, the Hypothesis is reliably sent to one of the experts, and the Expert-Timer is cancelled if it was running.

[3.6.](#) Receiving a Status-Report (SR)

When an agent receives a Status-Report, it does the following:

- (1) If the Ack-Request bit is set, a Status-Ack is sent to the origin of the message.
- (2) The agent then searches for a matching hypothesis entry. If none is found, or if the source of the SR was not the expert to which the matched hypothesis was sent, the SR is silently dropped and no further processing of the Status Report is done. Otherwise,
- (3) The hypothesis entry's Expert-Timer is restarted.
- (4) If an entry exists in the (logical) reliable message table for the Hypothesis, the reliable message entry is deleted.
- (5) If the status in the SR is neither Unconfirmed nor Diagnosis-Deferred, and an entry exists in the reliable message table for a Retract of the hypothesis, then the reliable message entry is deleted. If the status in the SR is Deleted, the hypothesis state is deleted (see [Section 4.9](#)).

When a problem whose status is being reported is the same as the problem in the hypothesis and the signed 16-bit difference between the included sequence number and the stored sequence number is positive, the new status and sequence number are stored in the hypothesis entry, and additional processing for some SR's is done as follows:

[3.6.1.](#) Receiving SR:Rejected

When a SR:Rejected message is received, the matched hypothesis state is deleted (see [Section 4.9](#)).

[3.6.2.](#) Receiving SR:Indeterminate

When a SR:Indeterminate message is received, the expert is removed from the matched hypothesis entry's expert list. If any experts remain in the expert list, a Hypothesis message is sent to one of them, the Expert-Timer is cancelled if it was running, and the matched hypothesis is marked as Unconfirmed. If the expert list is empty, the hypothesis entry's Expert-Timer is stopped; in addition, if no problem table entries have the hypothesis in the cause list, the hypothesis state is

Expires July 1998

[Page 13]

Draft

GDT Protocol Specification

January 1997

deleted (see [Section 4.9](#)).

[3.6.3.](#) Receiving SR:Repaired, SR:CantRepair, or SR:WentAway

When a SR:Repaired, or SR:CantRepair, or SR:WentAway message is received, the hypothesis entry's Expert-Timer is stopped. If the agent is not an expert (or the agent is an expert and no problem entries have the hypothesis in the cause list), the hypothesis state is deleted (see [Section 4.9](#)).

[3.7.](#) Timers

Timers are implemented in an implementation-specific manner. For example, a timer may count up or down, or may simply expire at a specific time. Setting a timer to a value T means that it will expire after T seconds.

Ack-Timer:

An Ack-Timer is kept for each reliable message entry. It is initialized to [[Ack-Timeout](#)] when the entry is created (i.e., when a message is to be sent reliably). It is cancelled if the entry is

deleted (i.e., when an acknowledgement is received). The first and second times it expires, the timer is reset to [[Ack-Timeout](#)], and the message is resent. If it expires a third time, and the message sent was a Status-Report, the destination is removed from the origin list of the matching problem state. If it expires a third time, and the message sent was a Hypothesis, processing continues as if an SR:Indeterminate message had been received from the destination. If it expires a third time, and the message sent was a Retract, the hypothesis entry is deleted (see [Section 4.9](#)). If it expires a third time, and the destination was the current parent, the agent expires all state for its old parent, resets its parent to be null (if it is not an expert) or equal to the root (if it is an expert), and repeats the parent selection process in [Section 3.1](#).

Expert-Timer:

An Expert-Timer is associated with each hypothesis entry. It is set to [[Expert-Timeout](#)] whenever a Status-Report message is received with a matching problem type and network element. If it expires, the hypothesis is processed as if an SR:Indeterminate message had been received (Sections [3.6.2](#) and [4.7.2](#)), and a new hypothesis is reported of lowH with unicast connectivity between the local agent and the remote expert ([Section 3.3](#)) unless this (new) problem is identical to

the hypothesis whose Expert-Timer expired.

Hypothesis-Timer:

At startup time, the Hypothesis-Timer is initialized to a random value between 0 and [[Hypothesis-Period](#)] seconds. When it expires, the timer is immediately reset to [[Hypothesis-Period](#)] seconds, and a Hypothesis is sent to the first expert in the expert list of each hypothesis state entry.

Capability-Timer:

For each non-local capability in the capability table, its Capability-Timer is reset to the associated Holdtime in any Redirects or (if the agent is an ELS) Am-Child messages received which contain it. When it expires, the capability entry is deleted.

[3.7.1](#). Default Values

[Ack-Timeout]

The time after which a message will be resent unless an acknowledgement was received. Default: 5 seconds.

[Expert-Timeout]

The time after which a hypothesis will be marked Indeterminate unless a Status-Report for it is received. Default: 190 (= default [SR-Period]*3 + 10) seconds.

[Hypothesis-Period]

The time between sending periodic Hypothesis messages for all hypotheses. Default: 300 seconds.

[4.](#) Expert Behavior

In addition to following all of the rules for simple agents, GDT experts must do additional processing as follows.

[4.1.](#) Startup

The root and parent are both initialized to be null, and the expert joins the RootAdv group. The root will be set as soon as an Am-Root message is received from a legal parent.

The expert SHOULD also begin an expanding-ring search, as described in [Section 3.1](#). The parent will be set as soon as an Am-Root or Am-Server message is received from a legal parent.

[4.2.](#) Receiving an Am-Root message

When an Am-Root message is received, the following actions are taken.

The source's address and active mask length are extracted from the message. If the agent's own address does not fall within this prefix,

the message is silently dropped and no further processing is done.

If the agent has no root state, or if the source's maximal mask length is less than the stored root's maximal mask length, or if the mask are equal but the source has a lower address than the stored root, then:

- (1) The source's information is stored as the new root.
- (2) If the agent has no parent, the source's information is also stored as the new parent, and the actions in [Section 3.2](#) are performed.

If the source is the current root, the Root-Timer is restarted.

[4.3.](#) Receiving a Transfer

When an expert receives a Transfer, it searches its reliable message for an Am-Child message with the same sequence number as in the acknowledgement. If none is found, the Am-Parent message is silently dropped, and no further processing is done.

Otherwise, the reliable message entry is deleted.

If the expert then checks to see whether the message was sent by its current parent. If not, the message is silently dropped, and no further processing is done.

The new parent's address, maximal prefix length, and active prefix lengths are then extracted from the message and stored, and the actions in [Section 3.2](#) performed.

Finally, it reliably sends an Am-Child message to its new parent.

[4.4.](#) Receiving an Am-Parent message

When an Am-Parent message is received, the reliable message table is searched for an Am-Child message with the same sequence number as in the acknowledgement. If none is found, the Am-Parent message is silently

dropped, and no further processing is done.

Otherwise, the reliable message entry is deleted.

The expert then checks to see whether the message was sent by its current parent. If not, the message is silently dropped, and no further processing is done.

The Parent-Timer is then refreshed.

[4.5.](#) Receiving a Hypothesis message

When an expert receives a Hypothesis message, it first checks to see if one of its own capabilities covers the given hypothesis. If a local capability was found, the expert searches for any matching problem entry, and proceeds as follows:

- (1) If a matching problem entry was found, the origin of the Hypothesis is added to the problem entry's origin list, and a Status-Report with the problem's current status is then returned to the origin. Else,
- (2) If no matching entry was found, a new problem entry is created with the problem type and element copied from the Hypothesis received. The problem status is set to Unconfirmed, and the origin of the Hypothesis included in the entry's origin list. The expert then schedules a test of the Hypothesis (see [Section 4.10.1](#)). Finally, if the problem status is still Unconfirmed after the test is scheduled, an SR:Unconfirmed message is (unreliably) sent to the origin.

If no local capability was found, it next checks its capability table for any other experts with capabilities covering the given hypothesis. If none are found, an SR:Indeterminate message is (unreliably) sent to the origin of the Hypothesis.

If no local capability was matched, but one or more remote experts were found, the expert must either act as a proxy and forward the Hypothesis, or reply to the origin with a Redirect message containing the list of

remote experts. In either case, the list of experts is first ordered in the same manner described in [Section 3.3](#).

[4.6](#). Receiving a Retract message

When an expert receives a Retract message, it first searches for a matching problem entry. If a problem is found whose status is Confirmed, Covered, Retesting, Isolated, or Repair-Deferred, the Retract is silently dropped.

Otherwise, if a problem entry was found, the origin is removed from its origin list. If the origin list becomes null for an entry whose status is Diagnosis-Deferred, the test SHOULD be unscheduled, a Retract reliably sent for any hypothesis in the cause list which is marked as Unconfirmed or Diagnosis-Deferred, and the problem entry deleted. If the list origin becomes null for a problem entry whose status is Unconfirmed, the test in progress MAY be aborted, a Retract reliably sent for any hypothesis in the cause list which is marked as Unconfirmed or Diagnosis-Deferred, and the entry deleted.

Finally, a SR:Deleted message is sent to the origin of the Retract.

[4.7](#). Receiving a Status-Report (SR)

When a Status-Report is received, in addition to following the rules given in [Section 3.6](#), an expert also follows the rules outlined in this section for each problem entry, P, which previously (before any deletions described in Section {s:agentSR}) contained the matched hypothesis, H, in its cause list.

- (1) If the status in the Status-Report is not Deleted, Unconfirmed, or Diagnosis-Deferred, P's Retracted-bit for H is cleared.
- (2) If P is the same as the problem (R) whose status was reported in the SR, then a circular dependency must exist. To break the loop in the cause tree, the hypothesis H SHOULD be deleted (see [Section 4.9](#)).

If the Relay-bit was set in the received message, the Status-Report is relayed to all origins of problems with the hypothesis in the cause list. Furthermore, if the Ack-Request bit was set in the message received, the message is relayed reliably.

Draft

GDT Protocol Specification

January 1997

When a problem whose status is being reported (R) is the same as the problem in the hypothesis H, and the signed 16-bit difference between the included sequence number and the stored sequence number is positive, additional processing is done as described below.

[4.7.1.](#) Receiving SR:Diagnosis-Deferred

If there are any other hypotheses in the cause list which were not previously submitted, the expert may submit one or more of them.

[4.7.2.](#) Receiving SR:Indeterminate

If, after normal processing, no alternative experts exist, and P's status is Confirmed, Covered, or CantRepair, and all hypotheses left in P's cause list have been marked as Indeterminate:

- (1) If there is only one hypothesis in the cause list, then this hypothesis (H) is used below. Otherwise, if a hypothesis (H) of badHW of the same element is not already in the cause list, one is created and added to the cause list, and an entry is created for it in the problem table as well.
- (2) If the expert (if any) for H is the local agent itself, then the associated problem table entry for H is treated as having been confirmed (see [Section 4.10.5](#)). Otherwise, H is marked as Repair-Deferred, and a repair is scheduled for P (see [Section 4.12.1](#)).

If, after normal processing, no alternative experts exist, and P's status is Unconfirmed, Diagnosis-Deferred, or Intermediate, and all hypotheses left in P's cause list have been marked as Indeterminate, the problem status is set to Indeterminate, and a Status-Report with the Ack-Request bit set is reliably sent to all origins. The problem entry is then scheduled for deletion by setting its Deletion-Timer to [\[Deletion-Timeout\]](#).

[4.7.3.](#) Receiving SR:Confirmed, SR:Coveted, SR:Retesting, or SR:Isolated

- (1) If the current problem status is either Unconfirmed or Diagnosis-

Deferred, the problem status is set to Confirmed, and a SR:Confirmed message with the Relay and Ack-Request bits set is

Expires July 1998

[Page 19]

Draft

GDT Protocol Specification

January 1997

reliably sent to all origins.

- (2) For each Unconfirmed or Diagnosis-Deferred cause of the current problem, if the Retract-bit was not set, the Retract-bit is set. If, as a result, all problem entries with the hypothesis in the cause list have the Retract-bit set, then a Retract message is reliably sent to the same expert to which the Hypothesis was submitted.
- (3) If the current problem status is then Confirmed, it is changed to Covered.
- (4) If the current problem status is Repair-Deferred, the scheduled repair SHOULD be unscheduled and the status changed to Covered.

[4.7.4.](#) Receiving SR:Repair-Deferred

- (1) If the current problem status is either Unconfirmed or Diagnosis-Deferred, the problem status is set to Confirmed, and a SR:Confirmed message with the Relay and Ack-Request bits set is reliably sent to all origins.
- (2) If the current problem status is Confirmed, it is changed to Covered.
- (3) If P's status is Covered, and all hypotheses in the cause list have been marked as Indeterminate or Repair-Deferred, then a repair is scheduled for P (see [Section 4.12.1](#)).

[4.7.5.](#) Receiving SR:Repaired

- (1) If the current problem status is either Unconfirmed or Diagnosis-Deferred, the problem status is set to Confirmed, and a SR:Confirmed message with the Relay and Ack-Request bits set is reliably sent to all origins.

- (2) If the current problem status is Confirmed, it is changed to Covered.
- (3) If the current problem status is Isolated, the repair in progress MAY be aborted and the state changed to Covered.

Expires July 1998

[Page 20]

Draft

GDT Protocol Specification

January 1997

- (4) If the current problem status is Repair-Deferred, the scheduled repair SHOULD be unscheduled and the state changed to Covered.
- (5) If the current problem status is Covered, it is changed to Retesting, and a retest is scheduled ([Section 4.10.1](#)).
- (6) If the current problem status is Retesting, the problem entry is of intermediate type, and there are no other hypotheses in the cause list which are marked as Confirmed, Covered, Retesting, Repair-Deferred, or Isolated, then the problem entry's status is set to Repaired, a Status-Report with the Ack-Request bit set is reliably sent to all agents in the entry's origin list, and the problem entry is scheduled for deletion by setting its Deletion-Timer to [\[Deletion-Timeout\]](#). In addition, for any hypotheses in the cause list whose last known status is Unconfirmed or Diagnosis-Deferred, the associated Retract-bit is set. If, as a result, all problem entries with the hypothesis in the cause list have the Retract-bit set, then a Retract message is reliably sent to the same expert to which the Hypothesis was submitted.

[4.7.6](#). Receiving SR:WentAway

- (1) If the current problem status is either Unconfirmed or Diagnosis-Deferred, the problem status is set to Confirmed, and a SR:Confirmed message with the Relay and Ack-Request bits set is reliably sent to all origins.
- (2) If the current problem status is Confirmed, it is changed to Covered.

- (3) If the current problem status is Isolated, the repair in progress MAY be aborted and the state changed to Covered.
- (4) If the current problem status is Repair-Deferred, the scheduled repair SHOULD be unscheduled and the state changed to Covered.
- (5) If the current problem status is Covered, it is changed to Retesting, and a retest is scheduled ([Section 4.10.1](#)).
- (6) If the current problem status is Retesting, the problem entry is of intermediate type, and there are no other hypotheses in the cause list which are marked as Confirmed, Covered, Retesting, Repair-Deferred, or Isolated, then the problem entry's status is set to WentAway, a Status-Report with the Ack-Request bit set is reliably

Expires July 1998

[Page 21]

Draft

GDT Protocol Specification

January 1997

sent to all agents in the entry's origin list, and the problem entry is scheduled for deletion by setting its Deletion-Timer to [[Deletion-Timeout](#)]. In addition, for any hypotheses in the cause list whose last known status is Unconfirmed or Diagnosis-Deferred, the associated Retract-bit is set. If, as a result, all problem entries with the hypothesis in the cause list have the Retract-bit set, then a Retract message is reliably sent to the same expert to which the Hypothesis was submitted.

[4.8.](#) Receiving a Status-Ack

When a Status-Ack is received, the reliable message table is searched for a Status-Report with the same sequence number as in the acknowledgement. If none is found, the Status-Ack is silently dropped. Otherwise, the reliable message entry is deleted.

[4.9.](#) Deleting hypothesis state

Whenever hypothesis state is deleted at an expert, the following actions are performed for each problem entry, P, with the hypothesis in its cause list:

- (1) Remove the hypothesis from the cause list.

- (2) If P's status is Unconfirmed or Diagnosis-Deferred (which may happen with intermediate problem types), and the cause list is empty, the problem status is set to Rejected, and a Status-Report with the Ack-Request bit set is reliably sent to all origins. The problem entry is then scheduled for deletion by setting its Deletion-Timer to [[Deletion-Timeout](#)].
- (3) If P's status is Unconfirmed or Diagnosis-Deferred (which may happen with intermediate problem types), and its cause list contains one or more hypotheses, all of which have been marked as Indeterminate, the problem status is set to Indeterminate, and a Status-Report with the Ack-Request bit set is reliably sent to all origins. The problem entry is then scheduled for deletion by setting its Deletion-Timer to [[Deletion-Timeout](#)].
- (4) If P's status is Confirmed and the cause list is empty, a repair is scheduled ([Section 4.12.1](#)).

Expires July 1998

[Page 22]

Draft

GDT Protocol Specification

January 1997

- (5) If P's status is Confirmed and the cause list is non-empty, P is treated as if a SR:Indeterminate message had been received for a cause, by following the rules given in [Section 4.7.2](#).

[4.10](#). Supervising Tests

[4.10.1](#). Scheduling a test

A test is scheduled for a problem whenever a new Hypothesis is received or a SR:Repaired or SR:WentAway message is received for a cause.

To schedule a test, the following steps are performed:

- (1) If the current problem status is Covered, it is changed to Retesting.
- (2) If the problem type is higherU, upstreamU, downstreamH, or lowerH, a resolution is begun for the higher list, upstream list,

downstream list, or lower list, respectively. If the current problem status is Unconfirmed, and resolution is done in a non-blocking fashion, then the problem status is changed to Diagnosis-Deferred.

- (3) If the problem is a primary or superficial type, an expert may elect to begin a test immediately, or to defer it until a later time. (This decision is made by the Scheduler in an implementation-specific manner). If the current problem status is Unconfirmed, and the test was deferred, the problem status is changed to Diagnosis-Deferred.

4.10.2. When the time for a deferred test arrives

If the current problem status is Diagnosis-Deferred, the problem status is set to Unconfirmed. The test is then begun.

4.10.3. When a test completes, with negative results

When a test completes, rejecting the hypothesis tested:

- (1) If the problem status was Unconfirmed, the status is set to Rejected.

Expires July 1998

[Page 23]

Draft

GDT Protocol Specification

January 1997

- (2) If the problem status was Retesting, then the status is set to Repaired if any cause was marked as Repaired, and to WentAway otherwise.
- (3) If the problem status was Repair-Deferred, the status is set to WentAway.
- (4) If the problem status was Isolated, the status is set to Repaired.

If the problem status changed, a Status-Report with the Ack-Request bit set is reliably sent to all agents in the entry's origin list.

The problem entry is then scheduled for deletion by setting its

Deletion-Timer to [[Deletion-Timeout](#)].

For any hypotheses in the cause list whose last known status is not Indeterminate, Rejected, WentAway, or Repaired, the Retract-bit is set. If, as a result, all problem entries with the hypothesis in the cause list have the Retract-bit set, then a Retract message is reliably sent to the same expert to which the Hypothesis was submitted.

[4.10.4.](#) When a test is indeterminate

When a test completes, and the result is indeterminate, or when no test can be done:

- (1) The problem status is set to Indeterminate, and a SR:Indeterminate message with the Ack-Request bit set is reliably sent to all agents in the problem entry's origin list.
- (2) The problem entry is then scheduled for deletion by setting its the Deletion-Timer to [[Deletion-Timeout](#)].

[4.10.5.](#) When a test completes, with positive results

When a test completes, confirming a hypothesis:

- (1) If the previous problem status was Unconfirmed, the problem status is set to Confirmed, a triggered SR:Confirmed with the Relay-bit set is reliably sent to all origins of the problem entry, and causal hypotheses are generated (see [Section 4.10.6](#)). Else,

- (2) If the previous problem status was Retesting, the problem status is set to Confirmed, and causal hypotheses are generated (see [Section 4.10.6](#)). Else,
- (3) If the previous problem status was Repair-Deferred, the problem status is set to Isolated and a repair is immediately initiated. Else,

(4) If the previous problem status was Isolated:

If other possible repairs exist, the next repair MAY be scheduled ([Section 4.12.1](#)).

Otherwise, the problem status is set to Confirmed and causal hypotheses generated (see [Section 4.10.6](#)). (This covers the case where a new cause arose during the repair, but can result in redoing the same repair when multiple repairs are possible, unless the expert remembers that it has just tried the repair and failed.)

[4.10.6](#). Generating causal hypotheses

If the problem type is lowH, hypotheses of highU, lowerH, and downstreamH (and optionally badHW) are generated about the current element. These hypotheses are added to the problem entry's cause list, and one or more (recommend all) of them are submitted to itself ([Section 3.3](#)).

If the problem type is highU, hypotheses of upstreamU, higherU, lowC, and highD (and optionally badHW) are generated about the current element. These hypotheses are added to the problem entry's cause list, and one or more (recommend all) of them are submitted to itself ([Section 3.3](#)).

If the problem type is lowC, highD, or badHW, no hypotheses are submitted since they represent primary-type problems. Instead, a repair is scheduled ([Section 4.12.1](#)).

[4.11](#). Resolving lists of elements

Intermediate problem types are those whose immediate causes are problems with other elements. For intermediate problems, a list of other, potentially problematic, elements must be resolved.

[4.11.1](#). When a list resolution succeeds

If the list is empty, then the test is rejected ([Section 4.10.3](#)).

If the list is not empty, then hypotheses of lowH (if the problem type was lowerH or downstreamH) or highU (if the problem type was upstreamU or higherU) of each element in the list are added to the problem entry's cause list, and one or more (recommend all) of them are submitted to an expert ([Section 3.3](#)).

[4.11.2](#). When a list resolution fails

When a list resolution fails for an intermediate problem, the test for the intermediate problem is declared to be Indeterminate ([Section 4.10.4](#)).

[4.12](#). Supervising Repairs

A "repair" may entail performing an automated procedure, interacting with an operator, or simply alerting an operator and waiting until the operator notifies the agent that a manual repair has completed.

[4.12.1](#). Scheduling a repair

A repair is scheduled for a problem whenever the cause list of a Confirmed problem entry becomes empty, or when all hypotheses left in the cause list of a Covered problem entry have been marked as Repair-Deferred.

If the expert has diagnosis-only capability for the given problem, status is set to CantRepair, and a Status-Report with the Ack-Request and Relay bits set is reliably sent to all origins. (This ensures that repairs will be attempted at its immediate effects.) The problem entry is then scheduled for deletion by setting its Deletion-Timer to [\[Deletion-Timeout\]](#).

An expert may elect to begin a repair immediately, or to defer it until a later time. (This decision is made by the Scheduler in an implementation-specific manner.) If the repair is initiated immediately, the problem status is changed to Isolated. If the repair is deferred, the problem status is changed to Repair-Deferred. In either case, a Status-Report with the Relay-bit set is then sent to all origins.

[4.12.2.](#) When the time for a deferred repair arrives

The problem status is first changed to Isolated. In an implementation-specific (or domain-specific) manner, the expert then decides whether the repair has been deferred long enough that the problem must be reconfirmed (e.g., if the time elapsed since the problem was initially confirmed is greater than some threshold). If the problem must be reconfirmed, a test is immediately begun. Otherwise, the repair is immediately begun.

[4.12.3.](#) When a repair completes

Another test is immediately initiated to verify that the repair was successful.

[4.13.](#) Timers

Origin-Timer:

An Origin-Timer is associated with each origin of a problem entry. It is set to [[Origin-Timeout](#)] when the origin is first added to the entry's origin list, and is reset to that value whenever a subsequent Hypothesis message is received from that origin for the given problem. When it expires, the origin is removed from the entry's origin list. If the origin list becomes null for an entry whose status is Diagnosis-Deferred, the test SHOULD be unscheduled and the entry deleted. If the list becomes null for an entry whose status is Unconfirmed, the test in progress MAY be aborted and the entry deleted.

Deletion-Timer:

A Deletion-Timer is kept for each problem entry. It is initialized to [[Deletion-Timeout](#)] when the problem status is set to any of: Rejected, Indeterminate, Repaired, or WentAway. When it expires, the entry is deleted, and any hypotheses in the cause list which are referenced in no other problem entry cause lists are deleted (in addition, a Retract with the Ack-Request bit set is also sent for each of these hypotheses which is marked Unconfirmed or Diagnosis-Deferred).

Status-Report-Timer:

At startup time, the Status-Report-Timer is initialized to a random value between 0 and [[SR-Period](#)] seconds. When it expires, the timer is immediately reset to [[SR-Period](#)] seconds, and a Status-Report

Draft

GDT Protocol Specification

January 1997

(with the Relay-bit set if the status is Isolated or Repair-Deferred) is then sent to all origins for each problem state entry. This timer should not be reset by other events.

Advertisement-Timer:

At startup time, the Advertisement-Timer is initialized to a random value between 0 and [[Advertisement-Period](#)] seconds. When it expires, the timer is immediately reset to [[Advertisement-Period](#)] seconds, and an Am-Child message is sent to the expert's parent, containing a list of the expert's capabilities. If the expert is also an ELS, and has no parent, then an Am-Root message is multicast to the RootAdv group. This timer should not be reset by other events.

Root-Timer:

A Root-Timer is associated with the current root. It is set to the Holdtime included in the Am-Root message when the root is first set, and is reset to the included Holdtime whenever a subsequent Am-Root message is received from it. When the Root-Timer expires, if the root is also the expert's parent, and the expert is an ELS, then the root is set to the expert itself. Otherwise, the root is set to be empty.

[4.13.1](#). Default Values**[Origin-Timeout]**

The time after which state for an origin will be removed unless a periodic Hypothesis is received from it. Default: 910 (= default [[Hypothesis-Period](#)]*3 + 10) seconds.

[Deletion-Timeout]

The time between scheduling deletion of an entry, and the actual deletion. Default: 5 seconds.

[SR-Period]

The time between sending periodic Status-Reports for all entries. Default: 60 seconds.

[Advertisement-Period]

The time between sending periodic Am-Child messages to the parent ELS. Default: 1 day.

[Capability-Holdtime]

The holdtime for one's capabilities include in Am-Child messages sent. This should be set to $2.5 * [\text{Advertisement-Period}]$. Default:

Expires July 1998

[Page 28]

Draft

GDT Protocol Specification

January 1997

2.5 days.

[5.](#) Expert Location Server (ELS) Behavior

In addition to following all of the rules for experts, ELS's must do additional processing as follows.

[5.1.](#) Startup

The ELS initializes its active mask length to be equal to its maximal mask length, and initializes its root to be itself and its parent to be null.

The ELS then joins the RootAdv group.

The ELS SHOULD also begin an expanding-ring search, as described in [Section 3.1](#). The parent will be set as soon as an Am-Root or Am-Server message is received from a legal parent. When the expanding ring search completes (or [\[Capability-Holdtime\]](#) seconds after startup, if no such search is done), the ELS should join the All-GDT-Servers group so it may receive Whois-Server messages.

[5.2.](#) Receiving a Whois-Server message

When a Whois-Server message is received, the ELS checks to see if the included address is within the ELS's active policy prefix. If not, the message is silently dropped.

If the included mask length is shorter than the ELS's own maximal mask length, or if they are equal but the origin's address is lower than the ELS's own address, then the message is silently dropped.

If the Whois-Timer is already running, the message is silently dropped.

Otherwise, the ELS starts its Whois-Timer to a random value between 0 and [[Whois-Delay](#)] seconds, using the smallest clock granularity available.

Expires July 1998

[Page 29]

Draft

GDT Protocol Specification

January 1997

[5.3.](#) Receiving an Am-Server message

The message is first processed as with an Am-Root message, following the rules in [Section 4.2](#); afterwards, if the origin is the root, then the ELS leaves the GDT-Server-Location group and any expanding-ring search in progress is ended.

Otherwise, if the Whois-Timer is running, and the Am-Server message specifies that the sender's active policy prefix is equal to or less specific than the ELS's own active policy prefix, then the Whois-Timer is cancelled.

[5.4.](#) Receiving an Am-Parent message

If the message was not dropped according to the rules of [Section 4.4](#), then the following additional steps are taken:

- (1) The expert's own active prefix length is reset to the maximum of: its own maximal prefix length, and (new parent's active prefix length + 1).
- (2) For each child in the child table whose address no longer falls with the expert's own active prefix, the child is removed from the table and a Transfer is sent to it, redirecting it to the expert's parent.
- (3) For each child in the child table whose active mask length is not greater than the expert's own active mask length, an Am-Parent

message is sent to the child.

[5.5.](#) Receiving an Am-Child message

When an ELS receives an Am-Child message, it compares the senders's address and maximal prefix length included in the message with the active policy prefixes of its own children.

- (1) If any child is a legal parent of the sender, the ELS replies to the sender with a Transfer, redirecting it to that child, and, if the sender was also in the child table, it is removed. Else,
- (2) If no child is a legal parent of the sender, but the ELS itself is a legal parent, then:

Expires July 1998

[Page 30]

Draft

GDT Protocol Specification

January 1997

- (a) It adds the included capabilities to its capability table and initializes Capability-Timers to the associated holdtimes.
 - (b) It then sends an Am-Parent message back to the sender with the same sequence number as in the advertisement.
 - (c) If the sender was not in the child table, the sender is added. For each other child for which the sender is a legal parent, a Transfer is then sent to the other child, redirecting it to the sender, and the other child is removed from the child table.
 - (d) The Child-Timer for the new child is then restarted.
Else,
- (3) If the ELS is not a legal parent of the sender, then the ELS replies to the sender with a Transfer, redirecting it to the ELS's parent (or an address of 0 if it has no parent), and if the sender was also in the child table, it is removed.

[5.5.1.](#) Sending Am-Child messages with Aggregate Capabilities

If the ELS has a parent ELS, then every time the Advertisement-Timer

expires, the ELS will send an Am-Child to its parent (as all experts do).

The included capabilities MUST cover all capabilities which have been learned through Am-Child messages. The ELS should aggregate capabilities where possible, and must not include duplicate capabilities in the advertisement.

[5.6.](#) Timers

Child-Timer:

A Child-Timer is associated with each entry in the child table. It is set to the Holdtime included in the Am-Child message when the child is first added to the child table, and is reset to the included Holdtime whenever a subsequent Am-Child message is received from that child. When it expires, the entry is removed from the child table.

Whois-Timer:

The Whois-Timer is started when a Whois-Server message is received. It is not reset when another Whois-Server message is received while the timer is already running. The timer is cancelled if an Am-Root

message is received. If the timer expires, the ELS multicasts an Am-Root message (whether or not it is the root) to the RootAdv group.

[5.6.1.](#) Default Values

[Origin-Timeout]

The time after which an origin will be removed unless a periodic Hypothesis is received from it. Default: 910 (= default [\[Hypothesis-Period\]](#)*3 + 10) seconds.

[Deletion-Timeout]

The time between scheduling deletion of an entry, and the actual deletion. Default: 5 seconds.

[SR-Period]

The time between sending periodic Status-Reports for all entries.

[Advertisement-Period]

[Capability-Holdtime]

[Whois-Delay]

[Page 32]

January 1997

The header of each GDT message has the format illustrated below. The source IP address, port number, and message length are all contained in the encapsulating IP and UDP headers.

[illegible]

GDT Ver

Identifies the protocol version. The version number of the protocol defined in this memo is zero (0).

Rsvd

Some messages use these fields for special purposes. Unless otherwise specified, these bits are transmitted as zero and ignored upon receipt.

MType

Types for specific GDT messages. GDT message types are:

- | | |
|----|---------------|
| 0 | Hypothesis |
| 1 | Retract |
| 2 | Redirect |
| 3 | Status-Report |
| 4 | Status-Ack |
| 5 | Am-Child |
| 6 | Am-Parent |
| 7 | Transfer |
| 8 | Am-Root |
| 9 | Whois-Server |
| 10 | Am-Server |

Sequence Number

The sequence number is used by the receiver to detect out-of-order packets. The sequence number MUST increment by at least one for each GDT message sent to the same destination concerning the same problem. (It MAY, for example, increment by one for each message sent regardless of the destination or problem.) The sequence number is only used by the receiver to detect out-of-order packets.

Expires July 1998

[Page 33]

Draft

GDT Protocol Specification

January 1997

An Encoded-Unicast-Address has the following format:

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
Addr Family										Encoding Type										Unicast Address										...									

In addition, an Encoded-Network-Element and an Encoded-Capability both have the following format:

```

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
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|           Length           |           Value           ...
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

Options, each composed of a type, length (of the value only), and value, can be included in some messages. An option has the following format:

```

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
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|   Type   |           Length           |   Value   ...
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

Legal option types include:

0--8 Cause

The option identifies a cause of the reported problem, where the type is a problem type (see [Section 6.1](#)), and the value is a network element identifier. This option is typically included in relayed Status-Report messages.

[16](#) Expected time to confirm (ETC)

The length is set to 4, and the value is the expected number of seconds until a test for a problem's existence is completed. This option is typically included in SR:Unconfirmed and SR:Diagnosis-Deferred messages.

[17](#) Expected time to repair (ETR)

The length is set to 4, and the value is the expected number of seconds until the problem is repaired. This option is typically included in SR:Isolated and SR:Repair-Deferred messages.

[18](#) Attributes

The value is a set of additional (optional) attributes known for a network element. This option is typically included in SR:Hypothesis and SR:Redirects.

Other option types are reserved. Unknown options should be ignored, but should be propagated in relayed Status-Reports.

Draft

GDT Protocol Specification

January 1997

[6.1.](#) Hypothesis Message

```

      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
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|GDT Ver| Rsvd |MType=0| Rsvd |           Sequence Number           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| ProblemType |           Encoded-Network-Element           ...
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

GDT Ver, Rsvd, Sequence Number, Encoded-Network-Element

See above.

ProblemType

Legal values are:

```

    0    lowH
    1    highU
    2    lowerH
    3    higherU
    4    upstreamU
    5    highD
    6    lowC
    7    badHW
    8    downstreamH

```

[6.2.](#) Retract Message

```

      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
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|GDT Ver| Rsvd |MType=1| Rsvd |           Sequence Number           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| ProblemType |           Encoded-Network-Element           ...
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|           (optional) Attribute-Option           ...
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

GDT Ver, Rsvd, Sequence Number, ProblemType, Encoded-Network-Element

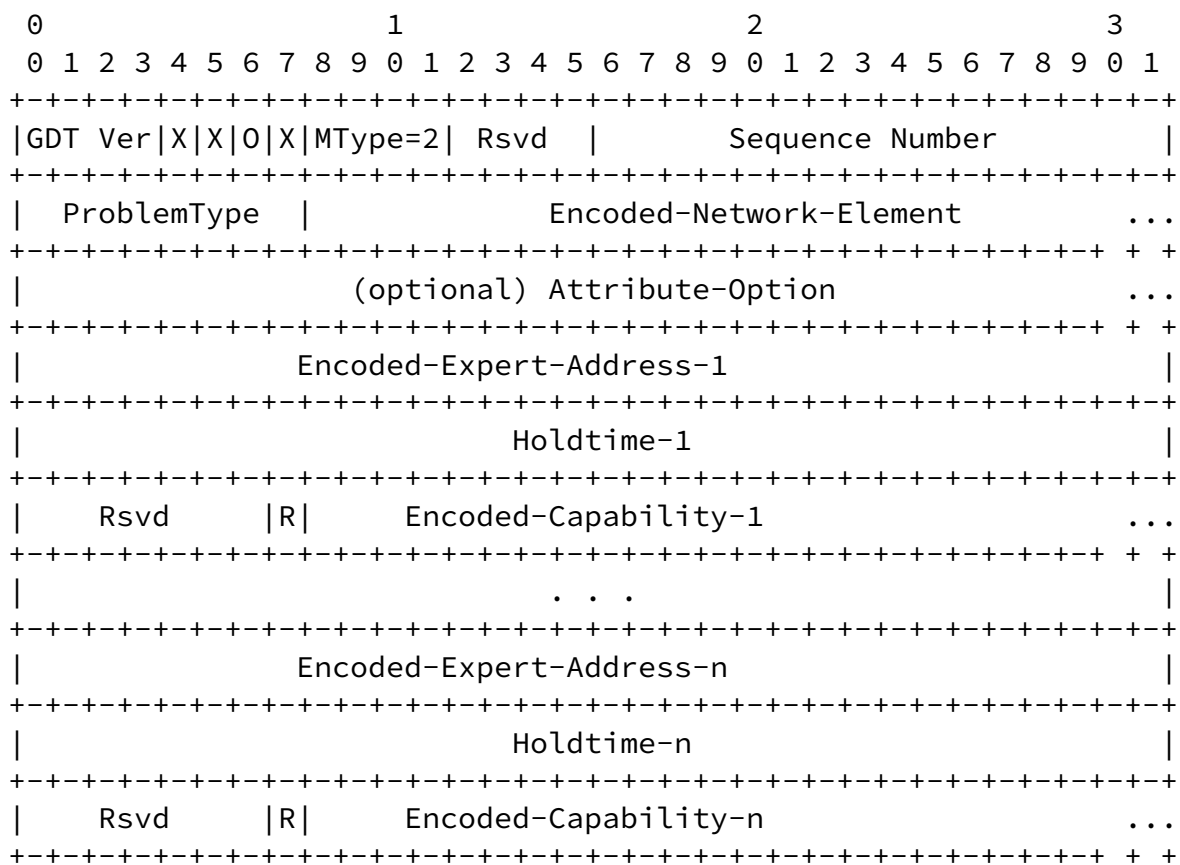
See above.

Draft

GDT Protocol Specification

January 1997

6.3. Redirect Message



GDT Ver, Rsvd, Sequence Number, ProblemType, Encoded-Network-Element
See above.

Option present-bit (0)

If set, an Attribute option is included; if cleared, no options are included.

Encoded-Expert-Address

The address of an expert whose capability follows. The format of this field is an Encoded-Unicast-Address as shown above.

Holdtime

The time-to-live (in seconds) of the following capability.

Encoded-Capability

The encoded capability of the indicated expert.

Expires July 1998

[Page 37]

Draft

GDT Protocol Specification

January 1997

6.4. Status-Report Message

[illegible]

GDT Ver, Sequence Number, ProblemType, Encoded-Network-Element

See above.

Ack-Request bit (A)

This bit indicates that a Status-Ack is requested. Currently, this bit is only set in triggered SR:Confirmed messages.

Relay-bit (R)

This bit indicates that the Status-Report is to be relayed to origins of the problem's effects. When relayed, the Ack-Request and Relay bits, Status field, and any Options are preserved in the relayed Status-Report. If no Cause option is present in a Status-Report received with the Relay-bit set, a Cause option is added to the Status-Report sent, using the problem type and network element from the original Status-Report. Currently, the Relay-bit bit is set in

SR:Isolated, SR:Repair-Deferred, and triggered SR:Confirmed messages.

Reserved-bits (X)

Transmitted as zero. Ignored upon receipt.

Status

Status values are:

- | | |
|---|--------------------|
| 0 | Unconfirmed |
| 1 | Diagnosis-Deferred |
| 2 | Rejected |
| 3 | Indeterminate |
| 4 | Confirmed |
| 5 | Covered |
| 6 | CantRepair |

Expires July 1998

[Page 38]

Draft

GDT Protocol Specification

January 1997

- | | |
|----|-----------------|
| 7 | Isolated |
| 8 | Repair-Deferred |
| 9 | Repaired |
| 10 | WentAway |
| 11 | Retesting |
| 12 | Deleted |

Options

A Cause option MUST be included if (and only if) the Status-Report is a relayed version of another Status-Report. The Cause option includes the problem type and network element of the original Status-Report. The Status field thus indicates the status of the cause when this option is present, rather than the status of the reported problem.

An ETC option SHOULD be included if the Status is Unconfirmed or Diagnosis-Deferred.

An ETR option SHOULD be included if the Status is Isolated or Repair-Deferred.

[6.5.](#) Status-Ack Message

```

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
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|GDT Ver|  Rsvd |Mtype=4| Status|           Sequence Number           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|  ProblemType  |           Encoded-Network-Element           ...
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

Sequence Number

The sequence number of the Status-Report which is being acknowledged.

All other fields are described above.

Expires July 1998

[Page 39]

Draft

GDT Protocol Specification

January 1997

[6.6.](#) Am-Child Message

```

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
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|GDT Ver|  Rsvd  |MType=5|  Rsvd  |           Sequence Number           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                               Holdtime                               |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|    Rsvd    |R|    Encoded-Capability-1    ...
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|            | |            . . .
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|    Rsvd    |R|    Encoded-Capability-n    ...
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

Holdtime

The amount of time (in seconds) the capabilities are valid. This field allows capabilities to be aged out, and should be set to

[Capability-Holdtime].

Repair-bit (R)

If set, this bit indicates that the following capability is valid for both diagnosis and repair; otherwise, the capability is valid for diagnosis only.

6.7. Am-Parent Message

[illegible]

Sequence Number

The sequence number of the Am-Child message which is being acknowledged.

Expires July 1998

[Page 40]

Draft

GDT Protocol Specification

January 1997

6.8. Transfer Message

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								
GDT Ver Rsvd MType=7 Rsvd										Sequence Number																													
										Encoded-Expert-Address										...																			
MaxMaskLen										CurrMaskLen																													

Encoded-Expert-Address

The address of the expert which the receiver should try as a parent.

MaxMaskLen

The length of the indicated expert's maximal policy prefix.

CurrMaskLen

The length of the indicated expert's active policy prefix.

6.9. Am-Root Message

```

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
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|GDT Ver| Rsvd  |MType=8| Rsvd  |           Sequence Number           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                               Holdtime                               |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|  MaxMaskLen  |  CurrMaskLen  |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

Holdtime

The amount of time (in seconds) this announcement is valid. This field allows the Root to be aged out, and should be set to the sender's [[Capability-Holdtime](#)].

MaxMaskLen

The length of the sender's maximal policy prefix.

CurrMaskLen

The length of the sender's active policy prefix.

Expires July 1998

[Page 41]

Draft

GDT Protocol Specification

January 1997

6.10. Whois-Server Message

```

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
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|GDT Ver| Rsvd  |MType=9| Rsvd  |           Sequence Number           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

```

|  MaxMaskLen  |      TTL      |
+---+---+---+---+---+---+---+---+---+---+---+---+

```

MaxMaskLen

For an ELS, this is the length of the sender's maximal policy prefix.
All other agents should set this field to 255.

TTL

The TTL at which the Whois-Server message is being sent, and at which
the server should respond with an Am-Server message.
All other fields are described above.

6.11. Am-Server Message

```

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
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|GDT Ver| Rsvd  |MTyp=10| Rsvd  |      Sequence Number      |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                               Holdtime                       |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|  MaxMaskLen  |  CurrMaskLen  |
+---+---+---+---+---+---+---+---+---+---+---+---+

```

All fields are as those for the Am-Root message.

7. References

- [1] Thaler, D., and C.V. Ravishankar, "Distributed Top-Down Hierarchy Construction", INFOCOM'98.
- [2] Thaler, D., "GDT Element Naming", Work in Progress, Jan. 1998.
- [3] Thaler, D., and C.V. Ravishankar, "Using Name-Based Mappings to Increase Hit Rates", IEEE/ACM Transactions on Networking, Feb. 1998.

8. Security Considerations

In general, Hypothesis messages need not be authenticated, since problem reports may be accepted from all sources. Before taking any further action, however, an expert will verify the existence of a reported problem.

One potential denial-of-service attack is sending a large number of Hypotheses for non-existent problems. Experts may combat such attacks by caching results of previous tests, and by deferring tests when a denial-of-service attack is suspected. In extreme cases where not enough resources exist to keep state for such origins, the expert may reply by sending an SR:Indeterminate message, implying that the test was indeterminate. This option does not require any state to be kept.

If Status-Report messages are unauthenticated, an attacker could either cause a non-existent problem to be falsely confirmed, in which case the origin will continue to wait for more feedback until the expert times out, or cause a true problem to be falsely rejected, in which case the origin must simply deal with the symptom (just as if the remote expert were unreachable).

[9.](#) Address of Author

Dave Thaler
Merit Network, Inc
4251 Plymouth Rd., Suite C
Ann Arbor, MI 48105-2785
Phone: +1 313 647 4813
EMail: thalerd@merit.net

Table of Contents

1	Introduction	2
1.1	Purpose	2
1.2	Terminology	2
2	Protocol Overview	5
2.1	Agent Requirements	6
2.1.1	Advertising Capabilities	7
2.2	Expert Location	9
2.3	Reliable Message Transport	9
3	Basic Behavior	10
3.1	Startup	10
3.2	Setting One's Parent	10
3.3	Detecting a problem and sending a Hypothesis	11
3.4	Sending a Retract	12
3.5	Receiving a Redirect	12
3.6	Receiving a Status-Report (SR)	13
3.6.1	Receiving SR:Rejected	13
3.6.2	Receiving SR:Indeterminate	13
3.6.3	Receiving SR:Repaired, SR:CantRepair, or SR:WentAway	14
3.7	Timers	14
3.7.1	Default Values	15
4	Expert Behavior	15
4.1	Startup	15
4.2	Receiving an Am-Root message	16
4.3	Receiving a Transfer	16
4.4	Receiving an Am-Parent message	17
4.5	Receiving a Hypothesis message	17
4.6	Receiving a Retract message	18
4.7	Receiving a Status-Report (SR)	18
4.7.1	Receiving SR:Diagnosis-Deferred	19
4.7.2	Receiving SR:Indeterminate	19
4.7.3	Receiving SR:Confirmed, SR:Coveted, SR:Retesting, or SR:Isolated	19
4.7.4	Receiving SR:Repair-Deferred	20
4.7.5	Receiving SR:Repaired	20
4.7.6	Receiving SR:WentAway	21
4.8	Receiving a Status-Ack	22
4.9	Deleting hypothesis state	22
4.10	Supervising Tests	23
4.10.1	Scheduling a test	23
4.10.2	When the time for a deferred test arrives	23
4.10.3	When a test completes, with negative results	23
4.10.4	When a test is indeterminate	24

Draft

GDT Protocol Specification

January 1997

4.10.5	When a test completes, with positive results	24
4.10.6	Generating causal hypotheses	25
4.11	Resolving lists of elements	25
4.11.1	When a list resolution succeeds	26
4.11.2	When a list resolution fails	26
4.12	Supervising Repairs	26
4.12.1	Scheduling a repair	26
4.12.2	When the time for a deferred repair arrives	27
4.12.3	When a repair completes	27
4.13	Timers	27
4.13.1	Default Values	28
5	Expert Location Server (ELS) Behavior	29
5.1	Startup	29
5.2	Receiving a Whois-Server message	29
5.3	Receiving an Am-Server message	30
5.4	Receiving an Am-Parent message	30
5.5	Receiving an Am-Child message	30
5.5.1	Sending Am-Child messages with Aggregate Capabilities	31
5.6	Timers	31
5.6.1	Default Values	32
6	Packet Formats	33
6.1	Hypothesis Message	36
6.2	Retract Message	36
6.3	Redirect Message	37
6.4	Status-Report Message	38
6.5	Status-Ack Message	39
6.6	Am-Child Message	40
6.7	Am-Parent Message	40
6.8	Transfer Message	41
6.9	Am-Root Message	41
6.10	Whois-Server Message	42
6.11	Am-Server Message	42
7	References	42
8	Security Considerations	43
9	Address of Author	43

Expires July 1998

[Page 45]