THE HIGH-LEVEL ENTITY MANAGEMENT SYSTEM (HEMS)

STATUS OF THIS MEMO

   An overview of the RFCs which comprise the High-Level Entity
   Management System is provided. This system is experimental, and is
   currently being tested in portions of the Internet. It is hoped that
   this work will help lead to a standard for IP internetwork
   management. Distribution of this memo is unlimited.

INTRODUCTION

   Until recently, a majority of critical components in IP networks,
   such as gateways, have come from a very small set of vendors. While
   each vendor had their own set of management protocols and mechanisms,
   the collection was small, and a knowledgeable system administrator
   could be expected to learn them all.

   Now, however, the number of vendors has grown quite large, and the
   lack of an accepted standard for management of network components is
   causing severe management problems. Compounding this problem is the
   explosive growth of the connected IP networks known as the Internet.
   The combination of increased size and heterogeneity is making
   internetwork management extremely difficult. This memo discusses an
   effort to devise a standard protocol for all devices, which should
   help alleviate the management problem.

   The RFCs that currently define the High-Level Entity Management
   System are this memo along with RFC-1022, 1024, and 1023. This list
   is expected to change and grow over time, and readers are strongly
   encouraged to check the RFC Index to find the most current versions.

MONITORING AND CONTROL

   Historically, the IP community has divided network management into
   two distinct types of activities: monitoring and control. Monitoring
   is the activity of extracting or collecting data from the network or
   a part of the network to observe its behavior. Control is the
   activity of taking actions to effect changes in the behavior of the
   network or a part of the network in real-time, typically in an
   attempt to improve the network's performance.

   Note that the ability to control presupposes the ability to monitor.
   Changing the behavior of the network without being able to observe
   the effects of the changes is not useful. On the other hand,
   monitoring without control makes some sense. Simply understanding
   what is causing a network to misbehave can be useful.

   Control is also a more difficult functionality to define. Control
   operations other than the most generic, are usually device-specific.
   The problem is not just a matter of providing a mechanism for
   control, but also defining a set of control operations which are
   generally applicable across a diverse set of devices. Permitting
   remote applications to exercise control over an entity also implies
   the need for a suite of safeguards to ensure that unauthorized
   applications cannot harm the network.

   Because monitoring is the key first step, in this initial design of
   the system, the authors have concentrated more heavily on the
   problems of effective monitoring. Although the basic control
   mechanisms are defined, many components need for control, such as
   strong access control mechanisms, have not been fully defined.

OVERVIEW OF THE HEMS

   The HEMS is made up of three parts: a query processor which can
   reside on any addressable entity, an event generator which also
   resides on entities, and applications which know how to send requests
   to the query processor and interpret the replies. The query
   processor and applications communicate using a message protocol which
   runs over a standard transport protocol.

The Query Processor

   The query processor is the key to the management system. It
   interprets all monitoring and control requests. For optimal network
   management, we would like to see query processors on most network
   entities.

   To encourage the implementations of query processors, one of the
   primary goals in designing the query processor was to make it as
   small and simple as possible, consistent with management
   requirements.

   Defining the management requirements was no small task, since the
   networking community has not yet reached a consensus about what kinds
   of monitoring information should be available from network entities,
   nor what control functions are required to properly manage those
   entities. The standards for HEMS were developed through discussions
   with several interest groups, and represent the authors' best effort

to distill the varying sets of needs.

The authors settled on a system which was extensible, robust and
host-architecture independent, and as simple as possible, consistent
with the other goals. Extensibility was essential because it is
clear that management needs will continue to evolve, and a closed
system which could not be changed would be obsolete almost as soon as
it was defined. Unfortunately, extensibility is also the requirement
least consistent with simplicity since the need to make the system
extensible led the authors to use self-describing data formats and an
interpreted query language.

A robust system is required if the system is to be useful for
diagnosing network failures. If the monitoring system cannot survive
at least moderate network failures, it is not useful.

The query processor is designed to be highly extensible. An
application sends the query processor instructions about objects to
be examined or changed. The query processor locates the objects in
its host entity, and performs the requested operations. The objects
are self-describing, using the binary-encoding scheme defined in ISO
Standard ASN.1. Care has been taken to use a limited set of the
ASN.1 coding set, so that query processor's handling of data can be
optimized.

It is a key feature of HEMS that messages to the query processor
contain multiple instructions. The authors felt that this would give
much higher performance than a remote procedure system which limited
an application to one operation per message.

The set of maintained objects is standardized across all entities.
Every entity is required to manage a small set of objects. In
addition, entities of a particular type (e.g., a gateway) may be
required to manage a larger set of objects, which are optional on
other entities. Entities are also permitted to make additional,
entity-specific objects available to applications. A method for
discovering the existence of additional objects is defined.

The combination of self-describing data, the ability to add to the
standard data set, and a query language which can be easily enhanced
appeared to offer the necessary extensibility.

Event Generator

On many network entities, particularly critical network components
such as gateways, it is necessary to have a way for the devices to
send unsolicited status messages to network management centers. In
the IP community, these messages have historically been referred to

as "traps", but for compatibility with the ISO nomenclature, in the
HEMS system they are called "events".

In the HEMS system, events are handled as slightly specialized
replies to queries, and are sent to one or more management centers.
Like all other HEMS messages, events are formatted in ASN.1 format.

Each event is given a well-known code, which is standardized across
all entities. Provision is also made for entity specific event
codes.

Applications

The HEMS expects that applications will be more intelligent than the
query processor. Among other functions, the applications will have
to be able to identify and parse entity-specific values which may be
returned.

The details of applications are largely not discussed in the HEMS
specifications because there is very little that needs to be
standardized. Applications must send requests using the protocols
discussed in the next section, but the interfaces the applications
provide for displaying monitoring or control information are entirely
application dependent.

Protocols

Query processors and applications communicate using an application-
specific monitoring protocol, the High-Level Entity Management
Protocol (HEMP). This protocol provides the formatting rules for the
queries and their replies.

HEMP runs over a standard transport protocol. There was a certain
amount of debate in the community about what type of transport
protocol was best suited for monitoring. The key issue was how
reliable monitoring interactions needed to be.

The authors expect that three types of management activities will
predominate: status monitoring, firefighting, and event reporting.

Status monitoring is envisioned as occasional retrieval of monitoring
information, possibly in response to the receipt of event messages.
In these situations, the network is expected to be in good working
condition, and monitoring exchanges could probably comfortably work
with an unreliable transport protocol. The chance of data loss is
small, and probably not a serious problem since the data is unlikely
to be so important that it must be reliably delivered. (However, it
should be noted that some applications may prefer reliable delivery

because it is more convenient.)

Firefighting is a completely different situation. In this scenario,
one or more sites are using management applications to try to locate
and fix a network problem. Here we must assume that while the
network functions (i.e., data can get through), it is not very
healthy. We should assume that packets are being lost, that network
routes will be non-optimal and that it is essential that the
monitoring data (which is presumably diagnostic) get back to the
application and that control requests are reliably delivered to the
entity. In such circumstances, a reliable protocol is essential.

Events provide yet another bit of complexity. Events contain useful
status information, but experience suggests that this information
does not have to be delivered reliably. If the problem is serious
enough, it will re-occur and the event will be sent again.
Furthermore, events will often be sent to more than one management
center, which would appear to preclude the use of connection-
oriented, reliable protocols such as TCP for events.

The current decision has been to establish two possible transport
options for HEMS. More experimental systems may use the Versatile
Message Transaction Protocol (VMTP), an experimental IP transaction
protocol. Near term production systems can use a combination of the
Transmission Control Protocol (TCP) and the User Datagram Protocol
(UDP), as described in RFC-1022.

Compatibility with Common Management Information Protocol (CMIP)

Several groups have expressed interest in being able to develop
applications which can use both HEMS and the emerging ISO-defined
Common Management Information Protocol (CMIP). It turns out that
such a co-existence is feasible, and the authors have made an effort
to accomodate it.

At the highest level, both CMIP and HEMS perform operations on
objects stored in remote entities, and both systems use ASN.1
formatting to represent those objects. This makes it possible to
develop a standard set of interface routines which can be used to
access either system, even though underlying mechanics of the systems
are quite different.