INTERNET ACCOUNTING: USAGE REPORTING ARCHITECTURE

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. This Internet Draft is a product of the Internet Accounting Working Group of the IETF.

Internet Drafts are draft documents valid for a maximum of six months. Internet Drafts may be updated, replaced, or obsoleted by other documents at any time. It is not appropriate to use Internet Drafts as reference material or to cite them other than as a "working draft" or "work in progress."

Please check the I-D abstract listing contained in each Internet Draft directory to learn the current status of this or any other Internet Draft.

1. Statement of Purpose and Scope

This INTERNET DRAFT describes an architecture for Internet usage reporting so that:

- network usage information is presented to collection and processing applications (e.g. billing) in a standarized format.

- the usage reporting protocol structure can be consistently applied to any protocol/application at any network layer (e.g. network, transport, application layers).

- usage reporting units are defined in such a way that the units are valid for multiple networking protocol stacks and that usage reporting protocol implementations are useful in multi-protocol environments.

- a near-term framework for usage reporting is established to encourage experimentation with internet accounting; results and effectiveness can be compared across multiple implementations now. Long-term and more complete protocols are currently limited to research efforts; stable standards are not expected to emerge for several years.

The usage reporting architecture specifies common metrics for measuring usage in an Internet environment. By using the same metrics, usage data can be exchanged and compared across multiple platforms. Usage data can be used for:

- o attribution of network usage to subscribers,
- o quantification of network performance,
- o usage-based policy enforcement, and
- o usage-based cost recovery (billing)

This document addresses the first of these, attribution of network usage to subscribers. The architecture outlined here targets connectionless IP-level services as its primary responsibility.

The usage reporting architecture is deliberately structured so that specific protocol implementations may extend coverage to multi-protocol environments and to other protocol layers, such as usage reporting for application-level services. Use of the same model for both network- and application-level billing may simplify the development of generic billing/statistics applications which process and/or correlate any or all levels of usage information.

The usage reporting architecture is NOT A PROTOCOL SPECIFICATION. It specifies and structures the information that a usage reporting protocol needs to collect, describes requirements that such a protocol must meet, and outlines tradeoffs.

For performance reasons, it may be desirable to use traffic information gathered through usage reporting in lieu of similar network statistics. Although the quantification of network performance is not the purpose of this architecture, the usage data may serve a dual purpose. This architecture favors accounting requirements over statistical convenience.

Policy-based routing and access control policies require mechanisms to enforce answers to the question: "who may use the network for what purpose". In the future, tighter coordination between usage reporting and access control should enable the use of real-time controls such as quotas. This architecture does not cover enforcement at this time.

The cost recovery structure decides "who pays for what". The major issue here is how to construct a tariff (who gets billed, how much, for which things, based on what info, etc). Tariff issues include fairness, predictability (how well can subscribers forecast their network charges), practicality (of gathering the data and administering the tariff), incentives (e.g. encouraging off-peak use), and cost recovery goals (100% recovery, subsidization, profit making). These issues are not covered here, although usage data reporting is one possible component of a comprehensive billing system.

Background information explaining why this approach was selected is provided by:

Individual collection protocol documents will address precise formats, e.g. MIB (management information base) specifications for SNMP or other management protocols.

2. Internet Accounting Framework

The accounting framework and terminology used by OSI Accounting Management is applicable here. The OSI reference model (ISO 7498-4 OSI Reference Model Part 4: Management Framework) defines the scope of OSI accounting as follows:

"Accounting management is the set of facilities which enables charges to be established for the use of managed objects and costs to be identified for the use of those managed objects. Accounting management is the set of facilities to

(a) inform users of costs incurred or resources consumed,

(b) enable accounting limits to be set for the use of managed objects, and

(c) enable costs to be combined where multiple managed objects are invoked to achieve a given communication objective."

Usage reporting mechanisms satisfy the measurement of "resources consumed" in (a). Pricing, i.e. establishing the cost of using these resources, is left to billing applications which are not covered here. Quotas are the mechanism for enforcing (b). Combining costs (c) is achieved through the post-processing of usage data by accounting applications not covered here.

The near-term architecture describes usage reporting only. Other aspects of an overall architecture are left for future extension or replacement by a long-term Internet accounting architecture. The following sections outline a model of internet accounting, specifically the usage reporting function, which is further refined throughout this document.

2.1 Internet Accounting Model

The Internet accounting model draws from working drafts of the OSI accounting model. It separates accounting functions into the parts shown below.

```
               NETWORK MANAGER ------------------
                   /     \                  \
                  /       \                  \
                 /         \                  \
                /           \                  \
           METER   <----->   COLLECTOR   <----->   APPLICATION
```

- o NETWORK MANAGER (or simply, MANAGER): The network manager is responsible for the control of the meter and collector, and determines and identifies backup collectors and managers as required.

- o METER: The meter performs the measurement and aggregate the results. Some characteristics of the meter are implementation-specific.

- o COLLECTOR: The collector is responsible for the integrity and security of data during transport from the meter to the application. This responsibility includes accurate and preferably unforgeable recording of accountable (billable) party identity.

- o APPLICATION: The application manipulates the usage data in accordance with policy, and determines the need for information from the metering devices.

QUOTAS are a means for information to be transferred from the usage reporting system to network management's access control function for the purpose of enforcement, i.e. limits placed on usage. A complete implementation of quotas may involve real-time distributed interactions between meters, the quota system, and access control. Enforcement of quotas is beyond the scope of the near-term architecture.

Standard information required for performing the collection of usage information of meters can be viewed as the product of protocol exchanges between the following parties:

- o the METER itself, where traffic is measured and usage data "generated".

- o the MANAGER, who manages the topology of the networks and relationships between entities in the network.

- o the COLLECTOR, or recipient of the usage data.

The exchanges can be categorized as follows:

- o between METER and COLLECTOR

  The data which travels this path is the usage record itself. The purpose of all the other exchanges is to manage the proper execution of this exchange. Usage record format is described in this section. Usage records which travel from meter to collector consist of meter id, address list, subscriber id, attribute list (not yet defined, since it is only applicable to local-area reporting), and values (packet counts, byte counts, and timestamps). In general, the collector generates no traffic to the meter, with the exception of polls where a polling protocol is used. The collector may know about other characteristics of the interfaces which are being metered through other means. Most notably, if an interface is accounting on a statistical the collector should at least know the average sampling rate and preferably be able to set the sampling rate to control the accounting process. (Sampling algorithms are not prescribed by the architecture, however it should be noted that any sampling techniques must be accompanied by documentation documenting adequate security and statistical validity which should be approved by the Internet Engineering Task Force before adoption.)

- o between MANAGER and METER

  The manager is responsible for controlling the meter. Meter management consists of commands which start/stop usage reporting, manage the exchange between meter and collector(s) (to whom do meters report the data they collect), set reporting intervals and timers, and set reporting granularities.

  Although most of the control information consists of commands to the meter, the meter may need to inform the manager of unanticipated conditions and meter responses to time-critical situations, such as buffer overflows.

- o between NETWORK MANAGER and COLLECTOR

  These parallel the manager to meter exchange, permitting feedback on collection performance and controlling access to the collected traffic statistics. Frequently the manager and the collector will be the same entity.

o between COLLECTORs (COLLECTOR - COLLECTOR)

A CASCADE of collectors is formed when one collector aggregates information from other intermediate collectors.

```
                                  COLLECTOR A
                           /                       \
                          /                         \
               COLLECTOR B                         COLLECTOR C
          /         |        \                   /      |       \
         /          |         \                 /       |        \
COLLECTOR D  COLLECTOR E  COLLECTOR F      METER C1  METER C2  METER C3
    |            |            |
 METER D1     METER E1     METER F1
```

Collectors exchange data with other collectors only when cascading is in effect (hierarchical reporting) or collection systems are voluntarily exchanging data for cross-verification or merging incomplete data sets (both examples of peer reporting). One method of cascading reporting is for the collector closer to the actual meter to behave as a meter with regard to the aggregating (closer to the root) collector, using the METER to COLLECTOR exchange to relay data towards the root. The preferred method is file transfer. A generic usage reporting file format for data exchange between collection systems has yet to be specified, e.g. a version or offshoot of AMA based on the modifications made for SMDS accounting.

Since redundant reporting may be used in order to increase the reliability of usage data, exchanges among multiple entities must be considered as well.

- o multiple METERs to a COLLECTOR

  Several uniquely identified meters report to one or more collectors. Meters are identified by the collection protocol or by a header within each usage message from the meter to the collector. A collector must be able to accept data in varying granularities. Collectors may receive reports on the progress of packets at various metering points along the path which the packet travels. When the collected data is processed or analyzed, parallel information from the network management system may be required in order to determine which meter recorded the entry or exit point of the packet from the network.

- o one METER to multiple COLLECTORs

Meters may also report the same information to multiple collectors for the purposes of redundancy. In that case, the collectors should agree on a single set of collection rules.

- o between MANAGERs (MANAGER - MANAGER)

  Synchronization between multiple management systems is the province of network management protocols. This usage reporting architecture specifies only the network management controls necessary to perform the usage reporting function and does not address the more global issues of simultaneous or interleaved conflicting commands from multiple network management stations or the process of transferring control from one network management station to another.

3. Usage Reporting Components

The usage reporting architecture specifies a means for collecting information about network usage in connectionless Internet environments. Usage is reported on connectionless protocol packets sent at the internet layer. For example, in the OSI protocol suite, the datagrams being counted are OSI CLNP datagrams. In the DoD Protocol Suite, the datagrams are IP datagrams. More precisely, the packets being counted are datagram fragments - the individual units in which the connectionless network protocol carries data, known as Protocol Data Units or PDUs. Routing protocol traffic may also be counted. Connection-oriented protocols can be reported in the same format.

The following sections address:

- o meters
- o flows and reporting granularity
- o usage records

3.1 Meters

Meters count the quantities specified by VALUES and attribute them to ACCOUNTABLE ENTITIES. The accountable entity is the network subscriber.

The approach to usage reporting at the IP level outlined here assumes that routers or equivalent traffic monitors throughout the Internet are instrumented with meters to measure traffic. Issues surrounding the choice of meter placement are discussed in the Internet Accounting Background RFC.

The purpose of defining meters at the internet level is to devise a way of succinctly aggregating subscriber usage information. Since IP service is connectionless, there is by definition no way to tell whether a datagram with a particular source/destination combination is part of a stream of packets or not.

Each packet is completely independent. In order to provide fully detailed reporting about the actions of subscribers on the network, a separate usage record would have to be maintained for each packet detailing the usage information. This would result in a very high level of overhead, possibly as high as one packet of usage information for each packet of data.

Therefore, usage aggregation provides an economical and practical way to measure internetwork traffic and ascribe it to a network subscriber.

3.2 Flows and Reporting Granularity

For the purpose of usage reporting we define the concept of a FLOW, which is an artificial logical equivalent to a call or connection. A flow is a portion of traffic, delimited by a start and stop time, that is attributable to a particular accountable entity. Values (packet counts, byte counts, etc.) associated with a flow are aggregate quantities reflecting events which take place in the DURATION between the start and stop times. The start time of a flow is fixed for a given flow; the end time may increase with the age of the flow.

Sample Entity	[Attributes]	Values
10.1.0.1	IP/UDP	Packets, Bytes, Start/Stop Time

GRANULARITY is the "control knob" by which an application and/or the meter can trade off the overhead associated with performing usage reporting for the level of detail supplied. A coarser granularity means a greater level of aggregation; finer granularity means a greater level of detail. Thus, the size and number of flows measured at a meter can be regulated by changing the granularity of the accountable entity, the attributes, or time intervals. Flows are like an adjustable pipe - many fine granularity streams can carry the data with each stream accounted for individually, or data can be bundled in one coarse granularity pipe.

Flow granularity is controlled by adjusting the level of detail at which the following are reported:

o the accountable entity

- o the categorization of packets (attributes)

- o the lifetime/duration of a flow (the reporting interval).

Settings for these granularity factors may vary from meter to meter. Also, they may be static (established by definition or agreement) or dynamic (set by a control protocol).

The granularity of ACCOUNTABLE ENTITIES is primarily specified by the ADDRESS LIST associated with a flow. That is, a flow's address list determines a subset of the traffic visible to the meter by specifying restrictions on the set of subscribers associated with that traffic. Beyond the local-area (i.e. for Internet traffic which crosses administrative boundaries) the following three types of address specifiers will be used to identify flows:

- o source address of the packets

- o destination address of the packets

- o source/destination address pair of the packets

For example, if a flow's address list is specified as "source address = IP address 10.1.0.1", then all IP packets from that address are counted in that flow. If a flow's address list is specified as "source address = IP address 10.1.0.1, destination address = IP address 26.1.0.1" then only IP packets from 10.1.0.1 to 26.1.0.1 are counted in that flow. When source/destination address pairs are used to designate flows, the set of flow data is referred to as a TRAFFIC MATRIX.

The addresses appearing in a flow's address list may include one or more of the following three types:

- o the INTERFACE NUMBER of the meter, i.e. the port on which the meter measured the traffic. Together with a unique address for the meter this uniquely identifies a particular physical level port or port matrix.

- o the ADJACENT (intermediate-system) NETWORK ADDRESS, which identifies the adjacent internet hop on the path of the packet. Since the network layer address within the network-layer protocol packet refers to end-systems only, the adjacent system (upstream or downstream neighbor) address must be derived from the sub-network address or translated into the appropriate network layer address (or unique name) for that neighbor.

o the END-SYSTEM NETWORK ADDRESS, which identifies the source or destination of the NETWORK-LEVEL packet.

Reporting by adjacent intermediate sources and destinations or simply by meter interface (most useful when the meter is embedded in a router) supports hierarchical internet reporting schemes as described in the background RFC. That is, it allows backbone and regional networks to measure usage to just the next lower level of granularity (i.e. to the regional and stub/enterprise levels, respectively), with the final breakdown according to end user performed by the stub/enterprise networks.

In cases where network addresses are dynamically allocated (e.g. mobile subscribers), further subscriber identification will be necessary for accurate accounting. Therefore, provision is made to further specify the accountable entity through the use of an optional SUBSCRIBER ID as part of the flow id. A subscriber ID may be associated with a particular flow either through a static rule table or through proprietary means within the meter.

Granularity of accountable entities is further specified by additional ATTRIBUTES. These attributes include characteristics such as traffic priority or other type of service characteristics.

User-level reporting is not addressed at this time, since it requires the addition of an IP option to identify the user, although the addition of a user-id as an entity at a later date is not precluded by this architecture.

This model can be continued at levels above the network level, such as transport and application for TCP/IP networks or transport/session/presentation /application for OSI networks. However, since the charter of the Internet Accounting Working Group ends at the internet-address (network layer), extensions to the usage record for application reporting will be left for future work.

For local-area reporting (within an administration), flows between subscriber entities can be subdivided into finer granularity by specifying ATTRIBUTES associated with the measured traffic. A sample IP attribute is:

- o QUALITY OF SERVICE: An internet header contains type of service bits, which indicate that the router should give the packet precedence for throughput, reliability, or delay.

Local-area reporting may later specify additional protocol layers in the address list, such as:

- o TRANSPORT PROTOCOL TYPE: this usually means TCP or UDP.

- o APPLICATION PROTOCOL TYPE: Many users want to peek up one

more layer for TCP connections (probably a violation of
protocol layering for an IP-level router, though not for a

host-based meter) to know whether the data is FTP (File Transfer), SMTP (E-Mail), Telnet (Virtual Terminal) and for UDP, if it is Domain Name Service (DNS).

For example, for a flow with a flow id including only TCP in its attributes, only TCP datagrams would be counted. This level of granularity is considered too detailed to perform well at the backbone level.

The set of rules controlling the reporting granularity are known as the COLLECTION RULES. As will be shown, the collection rules form an integral part of the reported information - i.e. the recorded usage information cannot be properly interpreted without a definition of the rules used to collect that information. It is expected that the collection rules will change rather infrequently; nonetheless, the rules in effect at any time must be identifiable via a RULE ID.

The usage data contained in the meter is further distinguished by the GROUP MASK. There are 8 arbitrary groups which may be allocated for administrative and policy purposes. For example, one group of usage records (specifiable under the rule set) may have priority over another group. A mask may identify groups which the meter may discard in case of buffer overflow. Different groups may even be collected from different collection stations, depending on the flexibility of the collection protocol.

Each group is represented by a bit in a an 8-bit mask. A particular usage record may be a member of multiple groups if multiple bits are set. The masks and polling algorithms should be set up in such a way as to avoid unintentional multiple reporting of individual records.

Since on-going counts in a particular bucket may be reported repeatedly during the lifetime of a flow in a fashion analogous to call-progress messages in X.96, the collection system may discard earlier progress messages as more complete messages are received.

3.3 Usage Records

A USAGE RECORD contains the descriptions of and values for one or more flows. Quantities are counted in terms of number of packets and number of bytes per flow. Each usage record contains the entity identifier of the meter (a network address) and a list of reported flows. The number of flows which can be reported in a single usage record may be limited by the maximum packet size of the collection protocol. If the collection protocol's maximum packet size is smaller than the largest usage record, the granularity of the usage record may be reduced until the usage record fits into the available space.

Therefore a usage record contains the following information in some form:

```
+-------------------------------------------------------------------+
|    RECORD IDENTIFIERS:                                            |
|      Meter Id (& digital signature if required)                   |
|      Timestamp                                                    |
|      Collection Rule ID                                           |
|-------------------------------------------------------------------|
|    FLOW IDENTIFIERS:            |     COUNTERS                    |
|      Address List               |       Packet Count              |
|      Subscriber ID (Optional)   |         Byte Count              |
|      Attributes (Optional)      |         Flow Start/Stop Time    |
|      Group ID flags             |                                 |
+-------------------------------------------------------------------+
```

The flow data is collected by the meter (e.g. in a router) as memory permits and forwarded at the reporting intervals to collectors where the data is stored more permanently in some aggregate form. The processing of data after delivery to the accounting application is beyond the scope of this document.

4. Meter Services

This section describes the operation and control of meters. The collection and control protocol document must specify the exact format in which information is reported; this section describes the information that can be derived from the data reported by the collection system and characterizes the demands placed on the collection and control protocols. Similarly, meter placement is discussed in the Internet Accounting Background document.

4.1 Between Meter and Collector - Usage Data Transmission

The usage record contents are the raison d'etre of the system. The accuracy, reliability, and security of transmission are the primary concerns of the meter/collector exchange. Since errors may occur on networks, and Internet packets may be dropped, some mechanism for ensuring that the usage information is transmitted intact is needed. The reliability of the collection protocol under light, normal, and extreme loads should be understood before selecting among the collection methods.

4.2 Collection Protocol Requirements: Polling, Interval Reporting,
and Traps

o POLLING

The collector sends a poll to the meter to indicate that the meter should respond with the requested record. Even where polling is used, meters under duress must be able to send data as spontaneous traps. The poll should contain a "piggyback ack", indicating that the collector has received the last message. The acknowledgement will allow the meter to discard completed flow records.

o INTERVAL REPORTING

The meter spontaneously generates usage information at intervals pre-specified by the manager. Even though the meter sends the data, some form of acknowledgement from the collection host with retransmission, or transmission via fully redundant paths to fully redundant collection hosts, must be used to provide reliability. Since the meter may wish to wait for an acknowledgement before flushing buffers, traps are still a necessary emergency mechanism.

o TRAPS

This may be threshold reporting or exception mechanism only. The meter senses a threshold condition and spontaneously fires a trap with the usage records to the collector (and, if an exception, sends a trap to the network manager as well indicating that an exception condition has occurred.)

In any case, the following scenarios must be considered:

(a) a poll or acknowledgement from the collector to the meter is lost,

(b) a message containing usage data from the meter to the collector is lost, or

(c) the meter fills its buffers faster than the poller empties it.

POLLING and INTERVAL reporting differ in that POLLING gives control of the precise timing to the COLLECTION host and INTERVAL reporting gives this control to the METER. Either end may want to have this control for load-balancing purposes, but it can't be had by both.

SNMP favors POLLING over INTERVAL reporting as a mechanism. The SNMP trap mechanism is available for the meter as a load-balancing emergency mechanism. The collection host should send acknowledgements to the meter anyway, and polls are messages on which acknowledgements can piggyback. The following discussion assumes that a POLLING algorithm is used with TRAPS as an emergency mechanism.

The network manager controls the scheduled interval. Therefore the collector and the meter request changes in reporting interval or granularity through their exchanges with the network management entity, and the network management entity arbitrates the default interval and granularity. (Minor or short-term deviations and load spikes are handled through the regular polling and trap mechanisms.)

Under normal polling conditions, the collection host specifies which set of usage records it is prepared to receive and the meter provides them. The poll contains an acknowledgement, so the meter may now flush reported and acknowledged records from its buffers. By using rolling counters in the meters, if a usage report is lost, the next report should contain information on the open flows. (For reliability, closed flows should not be flushed until an acknowledgement is received, or the flow has been reported twice, or an equally suitable reliability mechanism is employed.)

4.3 Rolling Counters, Timestamps, and Report-in-One-Bucket-Only

Once an usage record is sent the decision needs to be made whether to clear any existing flow records or whether to maintain them and add to the counts when recording subsequent traffic on the same flow. The second method, called rolling counters, is recommended and has several advantages. Its primary advantage is that it provides greater reliability - the system can now often survive the loss of some usage records. The next usage record will very often contain yet another reading of many the same flow buckets which were in the lost usage record. The "continuity" of data provided by rolling counters can also supply information used for "sanity" checks on the data itself, to guard against errors in calculations.

The use of rolling counters does introduce a new problem: how to distinguish a follow-on flow record from a new flow record. Consider the following example.

```
                      CONTINUING FLOW          OLD FLOW, then NEW FLOW.

                      start time = 1            start time = 1
Usage record N:       flow count=2000           flow count=2000 (done)

                      start time = 1            start time = 5
Usage record N+1:     flow count=3000           new flow count = 3000

Total count:                 3000                        5000
```

In the continuing flow case, the same flow was reported when its count was 2000, and again at 3000: the total count to date is 3000. In the OLD/NEW case, the old flow had a count of 2000. Its record was then stopped (perhaps because of temporary idleness, or MAX LIFETIME rules), but then more traffic on with the same characteristics came so a new flow record was started and it quickly reached a count of 3000. The total flow count from both the old and new records is 5000.

The flow START TIMESTAMP field is sufficient to resolve this. In the example above, the CONTINUING FLOW flow record in the second usage record has an old FLOW START timestamp, while the NEW FLOW contains a recent FLOW START timestamp.

Each packet counted may show up in only one usage record, so as to avoid multiple counting of a single packet (prevent double billing). The record of a single usage flow is informally called a "bucket". If multiple, sometimes overlapping, records of usage information are required (aggregate, individual, etc), the network manager should collect the counts in sufficiently detailed granularity so that aggregate and combination counts can be reconstructed in post-processing on the raw usage data.

For example, consider a meter from which it is required to record both "total packets coming in interface #1" and "total packets arriving from any interface sourced by IP address = a.b.c.d". Although a bucket can be declared for each case, it is not clear how to handle a packet which satisfies both criteria. It must only be counted once. By default, it will be counted in the first bucket for which it qualifies, and not in the other bucket. Further, it is not possible to reconstruct this information by post-processing. The solution in this case is to define not two, but THREE buckets, each one collecting a unique combination of the two criteria:

```
        Bucket 1:  Packets which came in interface 1,
                   And sourced by IP address a.b.c.d

        Bucket 2:  Packets which came in interface 1,
                   And NOT sourced by IP address a.b.c.d

        Bucket 3:  Packets which did NOT come in interface 1,
                   And sourced by IP address a.b.c.d

       (Bucket 4:  Packets which did NOT come in interface 1,
                   And NOT sourced by IP address a.b.c.d )
```

The desired information can now be reconstructed by post-processing. "Total packets coming in interface 1" can be found by adding buckets 1 & 2, and "Total packets sourced by IP address a.b.c.d" can be found by adding buckets 1 & 3. Note that in this case bucket 4 is not explicitly required since its information is not of interest, but is supplied here in parentheses for completeness.

4.4 Exception Conditions

Exception conditions are more difficult, particularly when the meter runs out of buffer space. Since, to prevent accounting twice for a single packet, packets can only be counted in a single flow at any given time, discarding records will result in the loss of information. The mechanisms to deal with this are as follows:

Meter Outages:

In case of impending meter outages (controlled crashes, slow power outages, etc.), the meter should simply trap the high-priority data to the collection system followed by the low-priority data, optionally followed by duplicate traps to the network management system or backup collection system.

Collector Outages:

If the collection system is down or isolated, the meter should inform the network management system of its failure to communicate with the collection system. Usage data is trapped to the backup collection system and/or directly to the network management system.

Management Outages:

If the network management system does not appear to be responding, the meter should continue reporting.

Buffer problems:

First, the network manager is informed by trap that there is too much usage data. This can usually be attributed to the interaction between the following controls:

(a) the reporting interval is too infrequent,

(b) the reporting granularity is too fine, or

(c) the throughput/bandwidth of circuits carrying the usage data is too low.

The network manager may change any of these parameters in response to the meter (or collector's) plea for help, or simply permit low-priority usage data to be discarded.

If it's a buffer problem and flushing the low-priority data will be sufficient, then the low-priority data is sent by trap to the collection system (optionally to the network management system as well as emergency backup collector), and the low-priority data is flushed from the system. Hopefully this will give enough time for the high-priority data to be reported at the regular interval.

If buffer problems are anticipated, the high-priority data is sent by trap to the collection system and optionally to the backup network management system, but not flushed until the need is immediate and the low priority data has already been trapped and flushed.

If the buffer requirements are so urgent or persistent that data cannot be sent as a trap, the meter may have permission from the network manager (configurable) to discard low-priority data and/or drop the reporting granularity as an exception-handling capability, in which case it should make attempts to inform the network manager and collection system of its actions. (The alternative is to refuse to pass traffic on new flows, an option which is not acceptable in most networks.)

4.5 Usage Record Content Description

The usage record is described below.

In the ADDRESS_LIST field, the "ADJACENT" address refers to the adjacent router, i.e., either the "previous hop" router or the "next hop" router. The address of the ADJACENT router may be collected in a local format (e.g. X.25, Ethernet,etc.) but it is preferred if the IP address form is used. (This may require an address translation, such as RARP tables.)

In the FLOW_RECORD field, the "Source" field is somewhat misnamed in that it handles both addresses of the true originating IP source as well as addresses of the ADJACENT (previous hop) router (see above). It might better be thought of as a "FROM" field. Similarly, the "Destination" field contains both the true IP destination address as well as the address of the ADJACENT (next hop) router, and might be thought of as the "TO" field.

> The Usage Record has a header containing default values for the flow records within it. Although collection protocols may have varying restrictions on format which make this structure impractical, the data delivered by the collection protocol should be complete enough that the following information can be reconstructed. This organization of data is selected to illustrate how this architecture can be expanded.

```
UsageRecord ::= SEQUENCE {
    RuleTab     [0] RuleTableID,-- Unique ID of RuleTable in effect
    StartTime   [1] TimeStamp,  -- Default Start Time for this rec.
    EndTime     [2] TimeStamp,  -- Default Stop Time for this record
```

```
    GroupMask   [3] OCTET STRING (SIZE (1)) OPTIONAL, Masks Not Required
    FragmentScale [4] INTEGER (1..127),-- counts are divided by 2 to the n
```

```
    OctetScale  [5] INTEGER,    -- counts are divided by 2 to the n

    SEQUENCE OF FlowRecord.  -- counts for individual flows
    }

FlowRecord ::=
    GroupMask [0] OCTET STRING (SIZE (1)) OPTIONAL,
    Flow    [1] FlowID,
    Values  [2] FlowData.

FlowID ::=
    Source-From [0] Address-list OPTIONAL, -- Must have source or dest
    Destination-To [1] Address-list OPTIONAL,  --     or both
    SubscriberID [2] Address-list OPTIONAL.
    -- attributes such as TOS to be added here later for local area work

-- The address list construct
-- in future, might have any address for any layer in the protocol
-- stack (session, presentation, application)

Address-list ::= SEQUENCE {
    interface        [0] INTEGER OPTIONAL,
    adjacent_address [1] NetWork_Address OPTIONAL,
    internet_address [2] NetWork_Address OPTIONAL,
    subscriberId     [3] OCTET STRING OPTIONAL
    }

NetWork_Address ::= CHOICE {
    n-1LayerAddress [0] IMPLICIT OCTET STRING ,
    ipAddress     [1] IMPLICIT IpAddress,
    nsapAddress   [2] IMPLICIT OCTET STRING,
    idprAddress   [3] IMPLICIT OCTET STRING<
    decnetAddress [4] IMPLICIT OCTET STRING
    }

FlowData ::= BEGIN
        acctFlowToOctets        Counter,        -- To Counters
        acctFlowToPDUs          Counter,
        acctFlowFromOctets      Counter,        -- From Counters
        acctFlowFromPDUs        Counter,
        acctFlowFirstTime       TimeTicks,
        acctFlowLastTime        TimeTicks
        }
TimeStamp :: = CHOICE {
     [0] TimeTicks  -- 1/100s of a second since base time
     }       -- base time since boot time or other base time
             -- established between meter, manager, and collector
```

5.0 Between Management and Meter - Control Functions and Exceptions

Because there are a number of parameters that must be set for internet usage reporting to function properly, and viable settings may change as a result of network traffic characteristics, it is desirable to have dynamic network management, as opposed to static meter configurations. Many of these operations have to do with space tradeoffs - if memory at the meter is exhausted, either the reporting interval must be decreased or a coarser granularity of aggregation must be used so that more data fits into less space.

Increasing the reporting interval effectively stores data in the meter; usage data in transit is limited by the effective bandwidth of the virtual link between the meter and the collector, and since these limited network resources are usually also used to carry user data (the purpose of the network), the level of usage reporting traffic should be kept to an affordable fraction of the bandwidth. ("Affordable" is a policy decision made by the network administration.) At any rate, it must be understood that the operations below do not represent the setting of independent variables; on the contrary, each of the values set has a direct and measurable effect on the behavior of the other variables.

Network management operations follow:

- o NETWORK MANAGEMENT AND COLLECTOR IDENTIFICATION

  The network management station should ensure that meters report to the correct set of collection stations, and take steps to prevent unauthorized access to usage information. The collection stations so identified should be prepared to poll if necessary and accept data from the appropriate meters. Alternate collection stations may be identified in case both the primary network management station and the primary collection station are unavailable. Similarly, alternate network management stations may be identified.

- o REPORTING INTERVAL CONTROL

  The usual reporting interval should be selected to cope with normal traffic patterns. However, it may not be unusual for a meter to exhaust its memory during traffic spikes even with a correctly set reporting interval. Some mechanism must be available for the meter to tell the network management station that it is in danger of exhausting its memory (by declaring a "high water" condition), and for the network management station to arbitrate (by decreasing the polling interval, letting nature take its course, or by telling the meter to ask for help sooner next time.)

o DUMP CONTROL

At some level of buffer usage it may be agreed that usage data is endangered, i.e. may be lost due to lack of memory. In this case, the meter needs to know at what level of buffer usage it should start to dump usage data (without waiting for a poll). Since this is a complex calculation which includes bandwidth and delay characteristics of the network, as well as the processing rate of the collector, it is assumed that the network management station is best able to determine the correct algorithm with the help of the meter and collector. A second panic level may result, when the meter actually does run out of buffer space for usage data. In this case, the meter and manager should agree on which usage data is of lower priority - i.e. which usage data should be deliberately flushed (even if without being reported) in order to make room for higher priority information.

o GRANULARITY CONTROL AND GROUPING OF DATA BY MASKS

Granularity control is a catch-all for all the parameters that can be tuned and traded to optimize the system's ability to reliably account for and store information on all the traffic (or as close to all the traffic as an administration requires). Granularity

(a) controls flow-id granularities for each interface,

(b) determines the number of buckets into which user traffic will be lumped together,

(c) prioritizes or groups of these buckets into different reporting categories.

Granularity rules are organized into a tree with decision points at each addressable protocol layer, starting with the physical interface. Each leaf on the decision tree also carries a "category" with it.

o FLOW LIFETIME CONTROL

Flow termination parameters include timeout parameters for obsoleting inactive flows and removing them from tables and maximum flow lifetimes. This is intertwined with reporting interval and granularity, and must be set in accordance with the other parameters.

4.2.2 Management to Meter: (polls and control)

SET HIGH WATER MARK

A % value interpreted by the meter which tells the meter when to send a trap indicating that the management station should increase the polling interval.

SET FLOOD MARK

A % value interpreted by the meter to indicate how full the table SHOULD be before the meter considers panicking and dumping the contents of the meter to the management station in raw (e.g., SNMP OPAQUE) form. 0% indicates that that a trap should be sent each time a counter is incremented. 100% indicates that a trap should never be sent.

SET FLOW TERMINATION PARAMETERS

The meter should have the good sense in situations where lack of resources may cause data loss to purge flow records from its tables:

(a) flows that have already been reported and show no activity since the last report

(b) oldest flows, or

(c) flows with the smallest number of unreported packets

- INACTIVITY TIMEOUT The time in seconds since last packet seen (and last report) after which the flow may be terminated.

- MAX LIFETIME Guidelines for the maximum lifetime of a flow. (Not mandatory, but the meter should make an effort at reporting time to purge flows that have had a lifetime greater than this value, even if it results in the instantaneous creation of a new flow with identical parameters.

SET FLOW PRIORITY [GROUP MASK] (mask is an 8-bit quantity)

Tell meter which flows are considered "critical" - i.e. in a crisis which flows can least afford to lose data. Reporting masks set by the COLLECTION RULES TABLE. This is used to indicate precedence among other things.

REPORT [GROUP MASK (0 or default indicates report ALL)]

Poll to meter indicating that a normal report of indicated flows should be made (i.e. any flow whose rule has indicated that it has a bit set which is set in the mask.)

SET GRANULARITY [RULE TABLE] see RULE TABLE, next section.

5.1 Rule Tables: Granularity Control

A rule table is a sequence of numbered rules which describe the granularity at which a meter should count. It is structured to support a "decision tree" hierarchy. For example, some rules can be used at a high-level to identify a large subclass of packets, and other rules can be at a mid-level to further break down the subclass into finer subclasses, and still other rules can be "leaf" rules which actually identify individual flows (buckets). Note that some rule tables will consist of only a few rules (possibly just one) resulting in the definition of only a few flows (buckets). In general, there will be a hierarchy of rules, such that the outcome of matching a particular rule might be to go to yet another rule for further qualification.

5.2 Classification Criteria

The information upon which such classifications are made come from two sources: the data fragment (or packet) itself, and the path that the fragment traveled. The fragment itself specifies:

- o address of the packet's source
- o network address of the packet's ultimate network destination
- o Other attributes, such as protocol used or type-of-service fields. (These attributes are not supported below but could be added later).

The path the packet traveled specifies:

- o the interface that the packet arrived on
- o the interface that the packet will leave on
- o the previous hop router/source address (address from layer n-1)
- o the next hop router/sink address (address from layer n-1)

The rule table, then, provides a way to classify packets according to the above parameters.

The rules use a form of "wild card" matching to allow entire "regions of address space", such as an entire source network, to be matched using a single rule. The wild card matching symbol notation is an asterisk (*).

Leaf rules support a feature which allows a single leaf to be expanded into several buckets via an "individuate" mask. For example, if a leaf rule identifies all packets which arrived from a particular source IP address, rather than count all of those packets into a single bucket, it may be desirable to further subdivide those packets according to which "next hop" they used. In that case, the individuate mask would identify the "destination adjacent interface" as the field to differentiate on, causing packets with different values in those fields to be counted in separate buckets.

Both the wild card matching mask and the individuate mask are simply short cuts. The same effect could be achieved without them but the rule table would become extremely large and the number of comparisons required might severely impact performance.

5.3 Representation of Flow Identification in the Flow Record

Once a packet has been classified and is ready to be counted, an appropriate flow record must either already exist or must be created. The flow record has a flexible format where unnecessary identification fields may be omitted. The determination of which fields of the flow record to use, and of what values to put in them, is specified by the leaf node of the rule table tree.

The leaf rules may contain additional information, such as a subscriber ID, which is to be placed in the attribute section of the usage record. That is, if a particular flow matches the qualifications for the leaf rule, then the corresponding flow record should be marked not only with the qualifying identification fields, but also with the additional information. Using this feature, several leaf nodes may each carry the same subscriber ID value, such that the resulting usage flow records will each contain the same subscriber ID value which can then be used in post-processing or between collector and meter as a collection criterion.

5.4 Standard Rule Tables

Although the rule table is a flexible tool, it can also become very complex. The following standard rule tables should be sufficient for most applications:

- o ADJACENT SYSTEMS: tell the meter to records packets by the IP address of the Adjacent Systems (neighboring originator or next-hop). (Variants on this table are "report source" or "report sink" only.) This strategy might be used by a regional or backbone network which wants to know how much aggregate traffic flows to or from its subscriber networks.

- o END SYSTEMS: tell the meter to record packets by the IP address pair contained in the packet. (Variants on this table are "report source" or "report sink" only.) This strategy might be used by an End System network to get detailed host traffic matrix usage data.

- HYBRID SYSTEMS: For one interface, report End Systems, for another interface report Adjacent Systems. This strategy might be used by an enterprise network to learn detail about local usage and use an aggregate count for the shared regional network.

5.5 Rule Table Components

The rule table is structured to allow decision-tree operations. Each rule begins with the specification of which field should be used for this rule's classification test. For example, the selected field might be "previous hop IP address". Each field may be further qualified by a corresponding field_mask. In this example, the intention might be to restrict the qualification to only look at the top two bytes of the previous hop IP address. The field_mask, then, would contain logical 1's corresponding to the subfields of interest and 0's otherwise. In this case, the field_mask 255.255.0.0 would be used.

Having extracted the appropriate portion of the field, the next section of the rule attempts to match the selected field against specified values. Each value is represented as part of an "action set". There can be many action sets in a rule. Each action set specifies a value to match and further instructions should there be a match. If there is no match, then the next sequential rule is evaluated.

5.6 Rule Table Definition

The following is the rule table definition.

```
--
-- The Rule Table
--

-- FieldIdentifier ::= CHOICE {
--       address        [0] IMPLICIT Network-Address,
--       mibVariable [1] IMPLICIT OBJECT IDENTIFIER
-- }

-- FieldValue ::= Opaque
```

```
-- PatternMask ::= OCTET STRING
```

```
-- Pattern ::= SEQUENCE {
--     mask1 PatternMask,
--     mask2 PatternMask
-- }

-- RuleAction ::= CHOICE {
--      direct [0] IMPLICIT ENUMERATED { ignore(1), count(2) },
--      goto [1] IMPLICIT INTEGER rule number to jump to
--      }

RuleTable ::= SEQUENCE OF AcctRuleEntry.

AcctRuleEntry ::= SEQUENCE {
        acctRuleIndex   INTEGER,                -- index
        acctRuleSelector        INTEGER,  -- what to select on
        acctRuleMask    Opaque,  -- the mask value
        acctRuleMatchedValue   Opaque,   -- the matched value
        acctRuleAction INTEGER,          -- action to take
        acctRuleJumpIndex       INTEGER -- where to go
        }

acctRuleSelector
        INTEGER { source-interface(1), destination-interface(2),
                 source-adjacent(3), destination-adjacent(4),
                 source-network(5),  destination-network(6)}
        DESCRIPTION "Defines the source of the value to match."


acctRuleMask
        DESCRIPTION "The initial mask used to compute the desired value.
        Depending on the data type being prepared, this could either
        be an OCTET STRING, or an INTEGER."


acctRuleMatchedValue
        DESCRIPTION "The resulting value to be matched for equality.
        Specifically, if the attribute chosen by the acctRuleSelector
        logically ANDed with the mask specified by the acctRuleMask
        equals the value specified in the acctRuleMatchedValue, then
        continue processing the table entry based on the action
        specified by the acctRuleAction entry. Otherwise, proceed to
        the next entry in the rule table."

acctRuleAction INTEGER { ignore(1), leaf(2), goto(3) }
        DESCRIPTION "The action to be taken. If ignore(1), stop the search.
          If leaf(2), then count the flow based on the values set
        aside during the walk thru the rule table.
        Otherwise, if goto(3), then record the value of the
```

attribute indicated by the acctRuleSelector, and
use the value of the acctRuleJumpIndex to start the

matching process at at new entry in the rule table."

acctRuleJumpIndex INTEGER
DESCRIPTION "index into the Rule table. Where to re-start the search. Must take on one of the values for acctRuleIndex."

Notes:

Caution must be taken to ensure that rule tables map into non-looping trees.

When address tests are used (field = address type), perform tests on the interface number first, the link level address second, the network address third, and the attributes (if any are defined later) last. Within an address type, test the source address first and the destination address last.

5.7 Meter to Management: (traps and responses)

CONTROL PARAMETERS:

DECLARE DATA LOSS	Trap to let manager know that usage data is being lost.
DECLARE HIGH WATER	Trap to request that manager increase polling interval. (Used when number of flows increases.)
DECLARE FLOOD / FLUSH	Trap dumping the flow records currently being monitored by the meter.

6.0 Between Management and Collector - Control Functions

Interactions between the manager and the collector are left in the province of the collection protocol definition.

7.0. Anticipated Collection Protocols

SNMP An Internet Accounting Meter Services MIB is needed. The working group recommends that SNMP security services be used in conjunction with the MIB and suggests that a reliable datagram service or transport service be used if and when available. Also, the introduction of a table retrieval service would greatly ease implementation and improve efficiency.

APPENDIX

A.1 Network Characterization

Internet users have extraordinarily diverse requirements. Networks differ in size, speed, throughput, and processing power, among other factors. There is a range of usage reporting capabilities and requirements. For usage reporting purposes, the Internet may be viewed as a continuum which changes in character as traffic passes through the following representative levels:

```
        International
        Backbones/National  ---------------------------
                                 /               \
        Regional/MidLevel    -----------   -------------
                             /    \     \ /    /     \
        Stub/Enterprise     ---   ---   ---   ----   ----
                            |||   |||   |||   ||||   ||||
        End-Systems/Hosts   xxx   xxx   xxx   xxxx   xxxx
```

Note that mesh architectures can also be built out of these components, and that these are merely descriptive terms. The nature of a single network may encompass any or all of the descriptions below, although some networks can be clearly identified as a single type.

BACKBONE networks are typically bulk carriers that connect other networks. Individual hosts (with the exception of network management devices and backbone service hosts) typically are not directly connected to backbones.

REGIONAL networks are closely related to backbones, and differ only in size, the number of networks connected via each port, and geographical coverage. Regionals may have directly connected hosts, acting as hybrid backbone/stub networks. A regional network is a SUBSCRIBER to the backbone.

STUB/ENTERPRISE networks connect hosts and local area networks. STUB/ENTERPRISE networks are SUBSCRIBERS to regional and backbone networks.

END SYSTEMS, colloquially HOSTS, are SUBSCRIBERS to any of the above networks.

Providing a uniform identification of the SUBSCRIBER in finer granularity than that of end-system, (e.g. user/account), is beyond the scope of the current architecture, although an optional field in the usage reporting record may carry system-specific "accountable

(billable) party" labels so that meters can implement proprietary or
non-standard schemes for the attribution of network traffic to

responsible parties.

A.2 Recommended Usage Reporting Capabilities

Initial recommended usage reporting conventions are outlined here according to the following internet building blocks. It is important to understand what complexity reporting introduces at each network level. Whereas the hierarchy is described top-down in the previous section, reporting requirements are more easily addressed bottom-up.

End-Systems
Stub Networks
Enterprise Networks
Regional Networks
Backbone Networks

END-SYSTEMS are currently responsible for allocating network usage to end-users, if this capability is desired. From the internet protocol perspective, end-systems are the finest granularity that can be identified without protocol modifications. Even if a meter violated protocol boundaries and tracked higher-level protocols, not all packets could be correctly allocated by user, and the definition of user itself varies too widely from operating system to operating system (e.g. how to trace network usage back to users from shared processes).

STUB and ENTERPRISE networks will usually collect traffic data either by end-system network address or network address pair if detailed reporting is required in the local area network. If no local reporting is required, they may record usage information in the exit router to track external traffic only. (These are the only networks which routinely use attributes to perform reporting at granularities finer than end-system or intermediate-system network address.)

REGIONAL networks are intermediate networks. In some cases, subscribers will be enterprise networks, in which case the intermediate system network address is sufficient to identify the regional's immediate subscriber. In other cases, individual hosts or a disjoint group of hosts may constitute a subscriber. Then end-system network address pairs need to be tracked for those subscribers. When the source may be an aggregate entity (such as a network, or adjacent router representing traffic from a world of hosts beyond) and the destination is a singular entity (or vice versa), the meter is said to be operating as a HYBRID system.

At the regional level, if the overhead is tolerable it may be advantageous to report usage both by intermediate system network address (e.g. adjacent router address) and by end-system network address or end-system network address pair.

BACKBONE networks are the highest level networks operating at higher link speeds and traffic levels. The high volume of traffic will in most cases preclude detailed usage reporting. Backbone networks will usually account for traffic by adjacent routers' network addresses.