Network Working Group   B. Stockman
Request for Comments: 1404   NORDUnet/SUNET
                                                              January 1993

                A Model for Common Operational Statistics

Status of the Memo

Abstract

   This memo describes a model for operational statistics in the
   Internet. It gives recommendations for metrics, measurements,
   polling periods, storage formats and presentation formats.

Acknowledgements

Table of Contents

1. Introduction

   Today it is not uncommon for many network administrations to collect
   and archive network management metrics that indicate network
   utilization, growth, and outages. The primary goal is to facilitate
   near-term problem isolation and longer-term network planning within
   the organization. There is also the larger goal of cooperative
   problem isolation and network planning between network
   administrations. This larger goal is likely to become increasingly
   important as the Internet continues to grow.

   There exist a variety of network management tools for the collection
   and presentation of network management metrics. However, different
   kinds of measurement and presentation techniques makes it difficult
   to compare data between networks. Plus, there is not common
   agreement on what metrics should be regularly collected or how they
   should be displayed.

There needs to be an agreed-upon model for

   1) A minimal set of common network management metrics to satisfy the
      goals stated above.

   2) Tools for collecting these metrics.

   3) A common storage format to facilitate the usage of these data by
      common presentation tools.

   4) Common presentation formats.

Under this Operational Statistics model, collection tools will
collect and store data in a given format to be retrieved later by
presentation tools displaying the data in a predefined way. (See
figure below.)

                      The Operational Statistics Model

     (Collection of common metrics, by commonly available tools, stored in
     a common format, displayed in common formats by commonly available
     presentation tools.)

```
                    !-----------------------!
                    !        Network        !
                    !---+-------------+---!
                       /               \
                      /                 \
                     /                   \
           --------+------         ----+---------
           !   New     !           !   Old     !
           ! Collection !          ! Collection !
           !   Tool    !           !   Tool    !
           !--------+---!          !------+-----!
                     \                    !
                      \          !-------+--------!
                       \         ! Post-Processor !
                        \        !--+-------------!
                         \          /
                          \        /
                           \      /
                      !--+-------+---!
                      !    Common    !
                      !  Statistics  !
                      !   Database   !
                      !-+--------+---!
                       /          \
                      /            \
                     /              \
                    /         !-+-------------!
                   /          ! Pre-Processor !
                  /           !-------+-------!
           !-----------+--!           !
           !   New     !    !-------+-------!
           ! Presentation !  !    Old     !
           !    Tool    !   ! Presentation !
           !--------+----!  !    Tool    !
                    \       !--+-----------!
                     \          /
                      \        /
                  !-+--------------+-!
                  ! Graphical Output !
                  ! (e.g., to paper  !
                  ! or X-window)     !
                  !------------------!
```
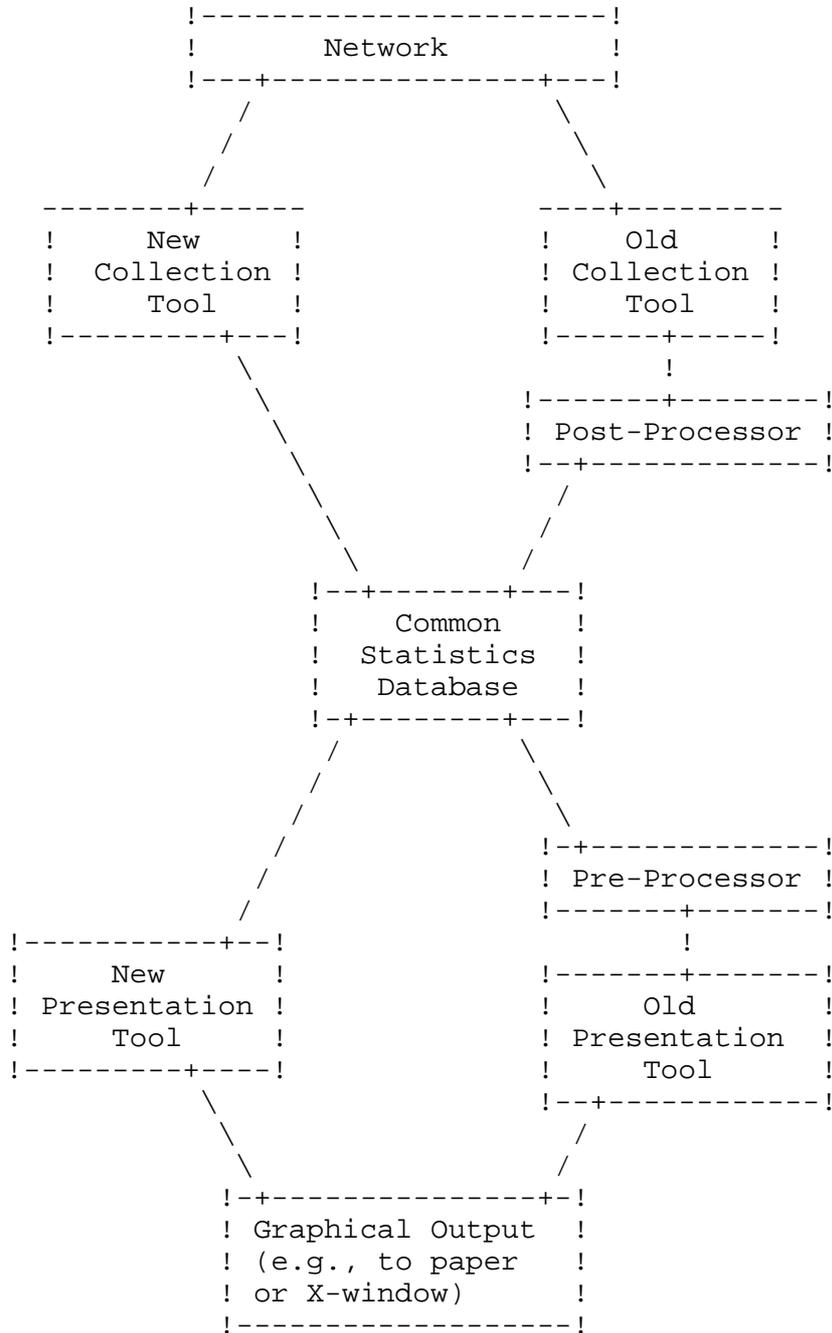
This memo gives an overview of this model for common operational
statistics. The model defines the gathering, storing and presentation
of network operational statistics and classifies the types of
information that should be available at each network operation center
conforming to this model.

The model defines a minimal set of metrics, how these metrics should
gathered and stored. Finally the model gives recommendations on the
content and the layout of statistical reports making it possible to
easily compare networks statistics between NOCs.

The primary purpose of this model is to define ways and methods on
how NOCs could most effectively share their operational statistics.
One intention with this model is to specify a baseline capability
that NOCs conforming to the this model may support with a minimal
development effort and a minimal ongoing effort.

2. The Model

The model defines three areas of interest on which all underlying
concepts are based.

        1. The definition of a minimal set of metrics to be gathered

        2. The definition of a format for storing collected statistical
           data.

        3. The definition of methods and formats for generating
           reports.

The model indicates that old tools used today could be retrofitted
into the new paradigm. This could be done by providing conversion-
filters between the old and the new environment tools. In this sense
this model intends to advocate the development of public domain
software for use by participating NOCs.

One basic idea with the model is that statistical data stored at one
place could be retrieved and displayed at some other place.

2.1 Metrics and Polling Periods

The intention here is to define a minimal set of metrics that easily
could be gathered using standard SNMP based network management tools.
These metrics should hence be available as variables in the Internet
Standard MIB.

If the Internet Standard MIB is changed also this minimal set of
metrics could be reconsidered as there are many metrics viewed as

important but currently not being defined in the standard MIB. For
some metrics being highly desirable to collect there are currently no
way to get them into the Internet Standard MIB as these metrics
probably are not possible to retrieve using SNMP. Tools and methods
in gathering such metrics should be explicitly defined if such
metrics are to be considered. This is, however, outside of the scope
of this memo.

2.2 Format for Storing Collected Data

A format for storing data is defined. The intention is to minimize
redundant information by using a single header structure where all
information relevant to a certain set of statistical data is stored.
This header section will give information on when and where the
corresponding statistical data where collected.

2.3 Reports

Some basic classes of reports are suggested with regards to different
views of network behavior. For this reason reports on totals of
octets and packets over some period in time are regarded as essential
to give an overall view of the traffic flows in a network.
Differentiation between application and protocols to give ideas on
which type of traffic is dominant is regarded as needed. Finally
reports on resource utilization are recommended..

Depending on the intention with a report the timeperiod over which it
spans may vary. For capacity planning there may be a need for longer
term reports while in engineering and operation there may be
sufficient with reports on weekly or daily basis.

2.4 Security Issues

There are legal, ethical and political concerns of data sharing.
People are concerned about showing data that may make one of the
networks look bad.

For this reason there is a need to insure integrity, conformity and
confidentiality of the shared data. To be useful, the same data must
be collected from all of the involved sites and it must be collected
at the same interval. To prevent vendors from getting an unfair
performance information, certain data must not be made available.

3. Categorization of Metrics

3.1 Overview

   This section gives a classification of metrics with regard to scope
   and easiness of retrieve. A recommendation of a minimal set of
   metrics is given. The section also gives some hints on metrics to be
   considered for future inclusion when available in the network
   management environment. Finally some thoughts on storage requirements
   are presented.

3.2 Categorization of Metrics Based on Measurement Areas

   The metrics used in evaluating network traffic could be classified
   into (at least) four major categories:

     - Utilization metrics
     - Performance metrics
     - Availability metrics
     - Stability metrics

3.2.1. Utilization Metrics

   These category describes different aspects of the total traffic being
   forwarded through the network. Possible metrics are:

     - Total input and output packets and octets.
     - Various peak metrics.
     - Per protocol and per application metrics.

3.2.2 Performance Metrics

   These metrics describes the quality of service such as delays and
   congestion situations. Possible metrics are:

     - RTT metrics on different protocol layers.
     - Number of collisions on a bus network
     - Number of ICMP Source Quench messages.
     - Number of packets dropped.
     - etc.

3.2.3 Availability Metrics

   This could be considered as the long term accessibility metrics on
   different protocol layers. Possible metrics are:

        - Line availability as percentage uptime.
        - Route availability
        - Application availability

3.2.4 Stability Metrics

    These metrics describes short term fluctuations in the network which
    degrades the service level. Also changes in traffic patterns could be
    recognized using these metrics. Possible metrics are:

      - Number of fast line status transitions
      - Number of fast route changes (also known as route flapping)
      - Number of routes per interface in the tables
      - Next hop count stability.
      - Short term ICMP behaviors.

3.3 Categorization Based on Availability of Metrics

    To be able to retrieve metrics the corresponding variables must be
    possible to access at every network object being part of the
    management domain for which statistics are being collected.

    Some metrics are easily retrievable as being defined as variables in
    the Internet Standard MIB while other metrics may be retrievable as
    being part of some vendor's private enterprise MIB subtree. Finally
    some metrics are considered as impossible to retrieve due to not
    being possible to include in the SNMP concept or that the actual
    measurement of these metrics would require extensive polling and
    hence download the network with management traffic.

    The metrics being categorized below could each be judged as an
    important metric in evaluating network behaviors. This list may
    serve for reconsider the decisions on which metric to be regarded as
    reasonable and desirable to collect. If the availability of below
    metrics changes these decisions may change.

3.3.1 Per Interface Variables Already in Internet Standard MIB
      (thus easy to retrieve)

          ifInUcastPkts   (unicast packet in)
          ifOutUcastPkts   (unicast packet out)
          ifInNUcastPkts   (non-unicasts packet in
          ifOutNUcastPkts (non-unicast packet out)
          ifInOctets   (octets in)
          ifOutOctets   (octets out)
          ifOperStatus   (line status)

3.3.2 Per Interface Variables in Internet Private Enterprise MIB
      (thus could sometimes be possible to retrieve)

          discarded packets in
          discarded packets out
          congestion events in
          congestion events out
          aggregate errors
          interface resets

3.3.3 Per Interface Variables Needing High Resolution Polling
      (which is hard due to resulting network load)

          interface queue length
          seconds missing stats
          interface unavailable
          route changes
          interface next hop count

3.3.4 Per Interface Variables not in any MIB
      (thus impossible to retrieve using SNMP but possible to include
       in a MIB).

          link layer packets in
          link layer packets out
          link layer octets in
          link layer octets out
          packet interarrival times
          packet size distribution

3.3.5 Per Node Variables
      (not categorized here)

          per protocol packets in
          per protocol packets out
          per protocol octets in
          per protocol octets out
          packets discarded in
          packets discarded out
          packet size distribution
          sys uptime
          poll delta time
          reboot count

3.3.6 Metrics not being Retrievable with SNMP

        delays (RTTs) on different protocol layers
        application layer availabilities
        peak behavior metrics

3.4 Recommended Metrics

   A large amount of metrics could be regarded for gathering in the
   process of doing network statistics. To facilitate for this model to
   reach general consensus there is a need to define a minimal set of
   metrics that are both essential and also possible to retrieve in a
   majority of today network objects. As an indication of being
   generally retrievable the presence in the Internet Standard MIB is
   regarded as a mandatory requirement.

3.4.1 Chosen Metrics

   The following metrics were chosen as desirable and reasonable being
   part of the Internet Standard MIB:

   For each interface:

        ifInOctets   (octets in)
        ifOutOctets   (octets out)
        ifInUcastPkts   (unicast packets in)
        ifOutUcastPkts (unicast packets out)
        ifInNUcastPkts (non-unicast packets in)
        ifOutNUcastPkts (non-unicast packets out)
        ifInDiscards   (in discards)
        ifOutDiscards   (out discards)
        ifOperStatus   (line status)

   For each node:

        ipForwDatagrams (IP forwards)
        ipInDiscards   (IP in discards)
        sysUpTime   (system uptime)

   All of the above metrics are available in the Internet Standard MIB.
   However, there also other metrics which could be recommended such as
   the RTT metric which probably never will be in any MIB. For such
   metrics other collection tools than SNMP have to be explicitly
   defined. The specification of such tools are outside scope of this
   memo.

4. Polling Frequencies

   The reason for the polling is to achieve statistics to serve as base
   for trend and capacity planning. From the operational data it shall
   be possible to derive engineering and management data. It shall be
   noted that all polling and saving values below are recommendation and
   not mandatory.

4.1 Variables Needing High Resolution Polling

   To be able to detect peak behaviors it is recommended that a period
   of maximum 1 minute (60 seconds) is used in the gathering of traffic
   data. The metrics to be gathered at this frequency is:

   for each interface

        ifInOctets   (octets in)
        ifOutOctets   (octets out)
        ifInUcastPkts   (unicast packets in)
        ifOutUcastPkts   (unicast packets out)

   If not possible to gather data at this high polling frequency, it is
   recommended that an even multiple of 60 seconds is used. The initial
   polling frequency value will be part of the stored statistical data
   as described in section 4 below.

4.2 Variables not Needing High Resolution Polling

   The other part of the recommended variables to be gathered, i.e.,

   For each interface:

        ifInNUcastPkts (non-unicast packets in)
        ifOutNUcastPkts (non-unicast packets out)
        ifInDiscards   (in discards)
        ifOutDiscards   (out discards)
        ifOperStatus   (line status)

   and for each node:

        ipForwDatagrams (IP forwards)
        ipInDiscards   (IP in discards)
        sysUpTime   (system uptime)

   These variables could be gathered at a lower polling rate. No
   specific polling rate is mentioned but it is recommended that the
   period chosen is an even multiple of 60 seconds.

5. Pre-Processing of Raw Statistical Data

5.1 Optimizing and Concentrating Data to Resources

   To avoid redundant data being stored in commonly available storage
   there is a need for processing the raw data. For example if a link is
   down there is no need to continuous store a counter that is not
   changing. Using variables such as sysUpTime and Line Status there is
   the possibility of not continuously storing data collected from links
   and nodes where no traffic have been transmitted over some period of
   time.

   Another aspect of processing is to decouple the data from the raw
   interface being polled. The intention should be to convert such data
   into the resource being of interest as for example the traffic on a
   given link. Changes of interface in a gateway for a given link should
   not be visible in the provided data.

5.2 Aggregation of Data

   A polling period of 1 minute will create the need of aggregating
   stored data. Aggregation here means that over a period with logged
   entries, a new aggregated entry is created by taking the first and
   last of the previously logged entries over some aggregation period
   and compute a new entry.

   Not to loose information on the peak values the aggregation also
   means that the peak value of the previous aggregation period is
   calculated and stored.

   This gives below layout of aggregated entries

   It is foreseen that over a relatively short period, polled data will
   be logged at the tightest polling period (1 minute). Regularly these
   data will be pre-processed into the actual files being provided.

   Suggestions for aggregation periods:

   Over a

        24 hour period   aggregate to 15 minutes,
        1 month period   aggregate to 1 hour,
        1 year period   aggregate to 1 day

   Aggregation is the computation of new average and maximum values for
   the aggregation period based on the previous aggregation period data.
   For each aggregation period the maximum, and average values are
   computed and stored. Also other aggregation period could be chosen

when needed. The chosen aggregation period value will be stored
together with the aggregated data as described below.

6. Storing of Statistical Data

   This section describes a format for storing of statistical data. The
   goal is to facilitate for a common set of tools for the gathering,
   storing and analysis of statistical data. The format is defined with
   the intention to minimize redundant information and by this minimize
   required storage. If a client server based model for retrieving
   remote statistical data is later being developed, the specified
   storage format should be possible to used as the transmission
   protocol.

   The format is built up by three different sections within the
   statistical storage, a label section, a device section and a data
   section. The label section gives the start and end times for a given
   data section as well as the file where the actual data is stored.
   The device section specifies what is being logged in the
   corresponding data section.

   To facilitate for multiple data sections within one log-file, label
   sections, device sections and data sections may occur more than once.
   Each section type is delimited by a BEGIN-END pair. Label and device
   sections could either be stored directly in the data-file or as
   separate files where the corresponding data-file is pointed out by
   the data-file entry in the label section.

   A data section must correspond to exactly one label section and one
   device section. If more label sections and device sections each data
   section will belong to the label section and device section
   immediately prepending the data section if these sections are stored
   within the data-file. How files are physically arranged is outside
   the scope of the document.

6.1 The Storage Format

     stat-data ::=
     <label-section><FS><device-section><FS><data-section><FS>
     [<device-section><FS><data-section><FS>]

     FS ::= "," | <LF> | <LF> # any text here <LF>

   The file must start with a label specification followed by a device
   specification followed by a data section. If the storing of logged
   data is for some reason interrupted a new label specification should
   be inserted when the storing is restarted. If the device being logged
   is changed this should be indicated as a new label and a new device

specification.

It shall here be noted that the actual physical storage of data is a
local decision and can vary a lot. There can be one data-file per
interface or multiple interfaces logged within the same data-file.
Label and device sections may be stored in a separate file as well as
within the data-file.

6.1.1 The Label Section

```
   label-section ::=    "BEGIN_LABEL"  <FS>
                        <start_time>    <FS>
                        <stop_time>     <FS>
                        <data_file>     <FS>
                        "END_LABEL"


   start-time  ::= <time-string>
   end-time    ::= <time-string>
   file-name   ::= <ascii-string>
   time-string ::= <year><month><day><hour><minute><second>
   year        ::= <digit><digit><digit><digit>
   month       ::= 01 | ... | 12
   hour        ::= 00 | ... | 23
   minute      ::= 00 | ... | 59
   second      ::= 00 | ... | 59
   digit       ::=  0 | ... | 9

   ascii-string ::= same as MIB II definition of <ascii-string>
```

The times defines start and stop times for the related set of logged
data. The time is in UTC.

6.1.2 The Device Section

```
   device-section ::= "BEGIN_DEVICE" <FS>
                      <device-field> <FS>
                      "END_DEVICE"

   device-field    ::= <networkname><FS><routername><FS><linkname><FS>
                       <bw-value><FS><bw-sort><FS><proto-type><FS>
                       <proto-addr><FS><time-zone><FS><tag-table>
                       [<tag-table>]


   networkname     ::= <ascii-string>
   routername      ::= <fully qualified domain name>
   linkname        ::= <ascii-string>
```

```
   bw-value          ::= <actual bandwidth value>
   bw-sort           ::= "bps" | "Kbps" | "Mbps" | "Gbps" | "Tbps"
   proto-type        ::= "IP" | "DECNET" | "X.25" | "CLNS"
   proto-addr        ::= <network-address depending on proto-type>
   timezone          ::= <"+" | "-"><00 | ... | 12><00 | 30>
   tag-table         ::= <tag><FS><tag-class><FS><variable-field>
                         [<FS><variable-field>]
   tag-class         ::= "total" | "peak"
   variable-field ::= <variable-name> <FS> <initial-polling-period><FS>
                         <aggregation-period>
   tag               ::= <ascii-string>
   variable-name  ::= <ascii-string>

   initial-polling-period ::= <digit>[<digit>]
   aggregation-period     ::= <digit>[<digit>]
```

The network name is a human readable string indicating to which
network the logged data belong.

The routername is the fully qualified name relevant for the network
architecture where the router is installed.

The linkname is a human readable string indicating the the
connectivity of the link where from the logged data is gathered.

The bandwidth should be the numerical value followed by the sort
being used. Valid sorts are bps, Kbps, Mbps, Tbps.

The prototype filed describes to which network architecture the
interface being logged is connected. Valid types are IP, DECNET, X.25
and CLNP.

The network address is the unique numeric address of the interface
being logged. The actual form of this address is dependent of the
protocol type as indicated in the proto-type field. For Internet
connected interfaces the "three-dot" notation should be used.

The time-zone indicates the timedifference that should be added to
the timestamp in the datasection to give the local time for the
logged interface.

The tag-table lists all the variables being polled. Variable names
are the fully qualified Internet MIB names. The table may contain
multiple tags. Each tag must be associated with only one polling and
aggregation period. If variables are being polled or aggregated at
different periods one separate tag in the table has to be used for
each period.

As variables may be polled with different polling periods within the
same set of logged data, there is a need to explicitly associate a
polling period with each variable. After being processed the actual
period covered may have changed as compared to the initial polling
period and this should be noted in the aggregation period field. The
initial polling period and aggregation period should be given in
seconds.

As aggregation also means the computation of the max value for the
previously polled data, the aggregation process have to extend the
tag table to include these maximum values. This could be done in
different ways. The variable field for the aggregated variables is
extended to also include the peak values from the previous period.
Another possibility is to create new tags for the peak values. To be
able to differentiate between polled raw data, aggregated total and
aggregated peak values some kind of unique naming of such entities
has to be implemented.

6.1.3 The Data Section

```
    data-section     ::= "BEGIN_DATA"<FS>
                         <data-field><LF>
                         "END_DATA"

    data-field       ::= <timestamp><FS><tag><FS>
                         <poll-delta><FS><delta-val>
                         [<FS><delta-val>]

    poll-delta  ::= <digit> [<digit>]
    tag         ::= <ascii-string>
    delta-value ::= <digit> [<digit>]
    timestamp   ::= <year><month><day><hour><minute><second>
    year        ::= <digit><digit><digit><digit>
    month       ::= 01 | ... | 12
    hour        ::= 00 | ... | 23
    minute      ::= 00 | ... | 59
    second      ::= 00 | ... | 59
    digit       ::=  0 | ... | 9
```

The datafield contains the polled data from a set of variables as
defined by the corresponding tag field. Each data field begins with
the timestamp for this poll followed by the tag defining the polled
variables followed by a polling delta value giving the period of time
in seconds since the previous poll. The variable values are stored as
delta values for counters and as absolute values for non-counter
values such as OperStatus. The timestamp is in UTC and the time-zone
field in the device section is used to compute the local time for the
device being logged.

6.2 Storage Requirement Estimations

   The header sections are not counted in this example. Assuming the
   the maximum polling intensity is used for all the 12 recommended
   variables and assuming the size in ascii of each variable is 8 bytes
   will give the below calculations based on one year of storing and
   aggregating statistical data.

   Assuming that data is saved according to the below scheme

        1 minute non-aggregated   saved 1 day.
        15 minute aggregation period   saved 1 week.
        1 hour aggregation period   saved 1 month.
        1 day aggregation period   saved 1 year.

   this will give:

   Size of one entry for each aggregation period:

                          Aggregation periods

                  1 min   15 min   1 hour   1 day

    Timestamp   14   14   14   14
    Tag   5   5   5   5
    Poll-Delta   2   3   4   5
    Total values   96   96   96   96
    Peak values   0   96   192   288
    Field separators 14   28   42   56

    Total entry size 131   242   353   464

   For each day 60*24 = 1440 entries with a total size of 1440*131 = 187
   Kbytes.

   For each weak 4*24*7 = 672 entries are stored with a total size of
   672*242 = 163 Kbytes

   For each month 24*30 = 720 entries are stored with a total size of
   720*353 = 254 Kbytes

   For each year 365 entries are stored with a total size of 365*464 =
   169 Kbytes.

   Grand total estimated storage for during one year = 773 Kbytes.

7. Report Formats

   This section suggest some report formats and defines the metrics to
   be used in such reports.

7.1 Report Types and Contents

   There is the longer term needs for monthly and yearly reports showing
   the long term tendencies in the network. There are the short term
   weekly reports giving indications on the medium term changes in the
   network behavior which could serve as input in the medium term
   engineering approach. Finally there is the daily reports giving
   instantaneous overviews needed in the daily operations of a network.

   These reports should give information on:

        Offered Load   Total traffic at external interfaces.
        Offered Load   Segmented by "Customer".
        Offered Load   Segmented protocol/application.

        Resource Utilization   Link/Router.

7.2 Contents of the Reports

7.2.1 Offered Load by Link

     Metric categories: input octets per external interface
                        output octets per external interface
                        input packets per external interface
                        output packets per external interface

   The intention is to visualize the overall trend of network traffic on
   each connected external interface. This could be done as a bar-chart
   giving the totals for each of the four metric categories. Based on
   the time period selected this could be done on a hourly, daily,
   monthly or yearly basis.

7.2.2 Offered Load by Customer

     Metric categories: input octets per customer
                        output octets per customer
                        input packets per customer
                        output packets per customer

   The recommendation is here to sort the offered load (in decreasing
   order) by customer. Plot the function F(n), where F(n) is percentage
   of total traffic offered to the top n customers or the function f(n)
   where f is the percentage of traffic offered by the n'th ranked

customers.

The definition of what should be meant by a customer has to be done
locally at the site where the statistics are being gathered.

The cumulative could be useful as an overview of how the traffic is
distributed among users since it enables to quickly pick off what
fraction of of the traffic comes from what number of "users."

A method of displaying both average and peak-behaviors in the same
bar-diagram is to compute both the average value over some period and
the peak value during the same period. The average and peak values
are then displayed in the same bar.

7.2.3 Resource Utilization Reporting

7.2.3.1 Utilization as Maximum Peak Behavior

The link utilization is used to capture information on network
loading. The polling interval must be small enough to be significant
with respect to variations in human activity since this is the
activity that drives loading in network variation. On the other hand,
there is no need to make it smaller than an interval over which
excessive delay would notably impact productivity. For this reason 30
minutes is a good estimate the time at which people remain in one
activity and over which prolonged high delay will affect their
productivity. To track 30 minute variations, there is a need to
sample twice as frequently, i.e., every 15 minutes. Using above
recommended polling period of 10 minutes this will hence be
sufficient to capture variations in utilizations.

A possible format for reporting utilizations seen as peak behaviors
is to use a method of combining averages and peak measurements onto
the same diagram. Compare for example peak-meters on audio-equipment.
If for example a diagram contains the daily totals for some period,
then the peaks would be the most busy hour during each day. If the
diagram was totals on hourly basis then the peak would be the maximum
10 minutes period for each hour.

By combining the average and the maximum values for a certain
timeperiod it will be possible to detect line utilization and
bottlenecks due to temporary high loads.

7.2.3.2 Utilization Visualized as a Frequency Distribution of Peaks

Another way of visualizing line utilization is to put the 10 minutes
samples in a histogram showing the relative frequency among the
samples vs. the load.

8. Considerations for Future Development

   This memo is the first effort in formalizing a common basis for
   operational statistics. One major guideline in this work has been to
   keep the model simple to facilitate for vendors and NOCs to easily
   integrate this model in their operational tools.

   There are, however, some ideas that could be progressed further to
   expand the scope and usability of the model.

8.1 A Client/Server Based Statistical Exchange System

   A possible way of development could be the definition of a
   client/server based architecture for providing Internet access to
   operational statistics. Such an architecture envisions that each NOC
   should install a server who provides locally collected information in
   a variety of forms for clients.

   Using a query language the client should be able to define the
   network object, the interface, the metrics and the time period to be
   provided. Using a TCP based protocol the server will transmit the
   requested data. Once these data is received by the client they could
   be processed and presented by a variety of tools needed. One
   possibility is to have an X-Window based tool that displays defined
   diagrams from data, supporting such types of diagrams being feed into
   the X-window tool directly from the statistical server. Another
   complementary method would be to generate PostScript output to be
   able to print the diagrams. In all cases there should be the
   possibility to store the retrieved data locally for later processing.

8.2 Inclusion of Variables not in the Internet Standard MIB

   As has been pointed out above in the categorization of metrics there
   are metrics which certainly could have been recommended if being
   available in the Internet Standard MIB. To facilitate for such
   metrics to be part of the set of recommended metrics it will be
   necessary to specify a subtree in the Internet Standard MIB
   containing variables judged necessary in the scope of performing
   operational statistics.

8.3 Detailed Resource Utilization Statistics

   One area of interest not covered in the above description of metrics
   and presentation formats is to present statistics on detailed views
   of the traffic flows. Such views could include statistics on a per
   application basis and on a per protocol basis. Today such metrics are
   not part of the Internet Standard MIB. Tools like the NSF NNStat are
   being used to gather information of this kind. A possible way to

achieve such data could be to define a NNStat MIB or to include such
variables in the above suggested operational statistics MIB subtree.

APPENDIX A

   Some formulas for statistical aggregation

   The following naming conventions are being used:

        For poll values poll(n)_j

        n = Polling or aggregation period
        j = Entry number

   poll(900)_j is thus the 15 minute total value.

        For peak values peak(n,m)_j

        n = Period over which the peak is calculated
        m = The peak period length
        j = Entry number

   peak(3600,900)_j is thus the maximum 15 minute period calculated
                    over 1 hour.

   Assume a polling over 24 hour period giving 1440 logged entries.

   =========================

   Without any aggregation we have

        poll(60)_1
        ......
        poll(60)_1439

   ========================

   15 minute aggregation will give 96 entries of total values

        poll(900)_1
        ....
        poll(900)_96

                    j=(n+14)

```
     poll(900)_k = SUM  poll(60)_j  n=1,16,31,...1425
                   j=n                k=1,2,....,96
```

There will also be 96 1 minute peak values.

```
                      j=(n+14)
     peak(900,60)_k = MAX poll(60)_000j  n=1,16,31,....,1425
                      j=n                k=1,2,....,96
```

=======================

Next aggregation step is from 15 minute to 1 hour.

This gives 24 totals

```
                      j=(n+3)
     poll(3600)_k = SUM  poll(900)_j  n=1,5,9,.....,93
                      j=n             k=1,2,....,24
```

and 24 1 minute peaks calculated over each hour.

```
                      j=(n+3)
     peak (3600,60)_k = MAX  peak(900,60)_j  n=1,5,9,.....,93
                      j=n                    k=1,2,....24
```

and finally 24 15 minute peaks calculated over each hour.

```
                      j=(n+3)
     peak (3600,900) = MAX poll(900)_j  n=1,5,9,.....,93
                      j=n
```

==================

Next aggregation step is from 1 hour to 24 hour

For each day with 1440 entries as above this will give

```
                      j=(n+23)
```

```
      poll(86400)_k = SUM  poll(3600)_j  n=1,25,51,.......
                      j=n                k=1,2............


                      j=(n+23)
      peak(86400,60)_k   = MAX peak(3600,60)_j  n=1,25,51,....
                      j=n                       k=1,2........


          which gives the busiest 1 minute period over 24 hours.


                      j=(n+23)
      peak(86400,900)_k  = MAX peak(3600,900)_j  n=1,25,51,....
                      j=n                        k=1,2,.......


          which gives the busiest 15 minute period over 24 hours.


                      j=(n+23)
      peak(86400,3600)_k = MAX poll(3600)_j  n=1,25,51,....
                      j=n                    k=1,2,.......


          which gives the busiest 1 hour period over 24 hours.

    ===================
```

There will probably be a difference between the three peak values in
the final 24 hour aggregation. Smaller peak period will give higher
values than longer, i.e., if adjusted to be numerically comparable.

```
 poll(86400)/3600 < peak(86400,3600) < peak(86400,900)*4
        < peak(86400,60)*60
```

APPENDIX B

    An example

    Assuming below data storage:

    BEGIN_DEVICE
        ....
        UNI-1,total,ifInOctet,   60, 60,ifOutOctet,   60, 60
        BRD-1,total,ifInNUcastPkts,300,300,ifOutNUcastPkts,300,300
        ....

    which gives

    BEGIN_DATA
        19920730000000,UNI-1,60, val1-1,val2-1
        19920730000060,UNI-1,60, val1-2,val2-2
        19920730000120,UNI-1,60, val1-3,val2-3
        19920730000180,UNI-1,60, val1-4,val2-4
        19920730000240,UNI-1,60, val1-5,val2-5
        19920730000300,UNI-1,60, val1-6,val2-6
        19920730000300,BRD-1,300, val1-7,val2-7
        19920730000360,UNI-1,60, val1-8,val2-8
        ...

    Aggregation to 15 minutes gives

    BEGIN_DEVICE
        ....
        UNI-1,total,ifInOctet,   60,900,ifOutOctet,   60,900
        BRD-1,total,ifInNUcastPkts,300,900,ifOutNUcastPkts,300,900
        UNI-2,peak, ifInOctet,   60,900,ifOutOctet,   60,900
        BRD-2,peak, ifInNUcastPkts,300,900,ifOutNUcastPkts,300,900
        ....

    where UNI-1 is the 15 minute total
          BRD-1 is the 15 minute total
          UNI-2 is the 1 minute peak over 15 minute (peak = peak(1))
          BRD-2 is the 5 minute peak over 15 minute (peak = peak(1))

    which gives

    BEGIN_DATA
        19920730000900,UNI-1,900, tot-val1,tot-val2
        19920730000900,BRD-1,900, tot-val1,tot-val2
        19920730000900,UNI-2,900, peak(1)-val1,peak(1)-val2

```
      19920730000900,BRD-2,900, peak(1)-val1,peak(1)-val2
      19920730001800,UNI-1,900, tot-val1,tot-val2
      19920730001800,BRD-1,900, tot-val1,tot-val2
      19920730001800,UNI-2,900, peak(1)-val1,peak(1)-val2
      19920730001800,BRD-2,900, peak(1)-val1,peak(1)-val2
      ......
```

   Next aggregation step to 1 hour generates:

   BEGIN_DEVICE
```
      ....
      UNI-1,total,ifInOctet,         60,3600,ifOutOctet,        60,3600
      BRD-1,total,ifInNUcastPkts,300,3600,ifOutNUcastPkts,300,3600
      UNI-2,peak,ifInOctet,          60,3600,ifOutOctet,        60,3600
      BRD-2,peak,ifInNUcastPkts, 300, 900,ifOutNUcastPkts,300, 900
      UNI-3,peak,ifInOctet,         900,3600,ifOutOctet,       900,3600
      BRD-3,peak,ifInNUcastPkts, 900,3600,ifOutNUcastPkts,900,3600
```

   where
   UNI-1 is the one hour total
   BRD-1 is the one hour total
   UNI-2 is the 1 minute peak over 1 hour (peak of peak = peak(2))
   BRD-2 is the 5 minute peak over 1 hour (peak of peak = peak(2))
   UNI-3 is the 15 minute peak over 1 hour (peak = peak(1))
   BRD-3 is the 15 minute peak over 1 hour (peak = peak(1))

   which gives

   BEGIN_DATA
```
      19920730003600,UNI-1,3600, tot-val1,tot-val2
      19920730003600,BRD-1,3600, tot-val1,tot-val2
      19920730003600,UNI-2,3600, peak(2)-val1,peak(2)-val2
      19920730003600,BRD-2,3600, peak(2)-val1,peak(2)-val2
      19920730003600,UNI-3,3600, peak(1)-val1,peak(1)-val2
      19920730003600,BRD-3,3600, peak(1)-val1,peak(1)-val2
      19920730007200,UNI-1,3600, tot-val1,tot-val2
      19920730007200,BRD-1,3600, tot-val1,tot-val2
      19920730007200,UNI-2,3600, peak(2)-val1,peak(2)-val2
      19920730007200,BRD-2,3600, peak(2)-val1,peak(2)-val2
      19920730007200,UNI-3,3600, peak(1)-val1,peak(1)-val2
      19920730007200,BRD-3,3600, peak(1)-val1,peak(1)-val2
      ......
```

   Finally aggregation step to 1 day generates:

```
   UNI-1,total,ifInOctet,60,86400,ifOutOctet,60,86400
```

```
    BRD-1,total,ifInNUcastPkts,300,86400,ifOutNUcastPkts,300,86400
    UNI-2,peak,ifInOctet,60,86400,ifOutOctet,60,86400
    BRD-2,peak,ifInNUcastPkts,300,900,ifOutNUcastPkts,300,900
    UNI-3,peak,ifInOctet,900,86400,ifOutOctet,900,86400
    BRD-3,peak,ifInNUcastPkts,900,86400,ifOutNUcastPkts,900,86400
    UNI-4,peak,ifInOctet,3600,86400,ifOutOctet,3600,86400
    BRD-4,peak,ifInNUcastPkts,3600,86400,ifOutNUcastPkts,3600,86400


    where
    UNI-1 is the 24 hour total
    BRD-1 is the 24 hour total
    UNI-2 is the  1 minute peak over 24 hour
        (peak of peak of peak = peak(3))
    UNI-3 is the 15 minute peak over 24 hour (peak of peak = peak(2))
    UNI-4 is the  1 hour   peak over 24 hour (peak = peak(1))
    BRD-2 is the  5 minute peak over 24 hour
        (peak of peak of peak = peak(3))
    BRD-3 is the 15 minute peak over 24 hour (peak of peak = peak(2))
    BRD-4 is the  1 hour   peak over 24 hour (peak = peak(1))

    which gives

    BEGIN_DATA
       19920730086400,UNI-1,86400, tot-val1,tot-val2
       19920730086400,BRD-1,86400, tot-val1,tot-val2
       19920730086400,UNI-2,86400, peak(3)-val1,peak(3)-val2
       19920730086400,BRD-2,86400, peak(3)-val1,peak(3)-val2
       19920730086400,UNI-3,86400, peak(2)-val1,peak(2)-val2
       19920730086400,BRD-3,86400, peak(2)-val1,peak(2)-val2
       19920730086400,UNI-4,86400, peak(1)-val1,peak(1)-val2
       19920730086400,BRD-4,86400, peak(1)-val1,peak(1)-val2
       19920730172800,UNI-1,86400, tot-val1,tot-val2
       19920730172800,BRD-1,86400, tot-val1,tot-val2
       19920730172800,UNI-2,86400, peak(3)-val1,peak(3)-val2
       19920730172800,BRD-2,86400, peak(3)-val1,peak(3)-val2
       19920730172800,UNI-3,86400, peak(2)-val1,peak(2)-val2
       19920730172800,UNI-3,86400, peak(2)-val1,peak(2)-val2
       19920730172800,UNI-4,86400, peak(1)-val1,peak(1)-val2
       19920730172800,BRD-4,86400, peak(1)-val1,peak(1)-val2
       ......
```

Security Considerations

    Security issues are discussed in Section 2.4.

Author's Address

    Bernhard Stockman
    NORDUnet/SUNET NOC
    Royal Institute of Technology
    Drottning Kristinas Vag 37B
    S-100 44 Stockholm, Sweden

    Phone: +46 8 790-6519
    Fax : +46 8 241-179
    Email: boss@sunet.se