Managing Modems by Periodic Polling

D. R. Seligman
Concord Communications, Inc.
33 Boston Post Road West
Marlboro, MA 01752
USA
dans@concord.com

Abstract

Network management and network monitoring technologies frequently rely on periodic SNMP polling of target devices to obtain statistical information for later reduction and analysis. Periodic polling has been applied to virtually every type of network device capable of supporting an SNMP agent, including probes, routers, switches, and servers. This paper examines the extension of periodic polling techniques to remote access devices -- modems, ISDN interfaces and remote access servers. Some unique problems are presented by the characteristics of these devices and by the way vendors have chosen to support them in SNMP MIBs. In particular, we discuss how Concord Communications, Inc., has extended its flagship network monitoring product, Network Health[tm], to support remote access devices.

1. Introduction

This paper discusses techniques that have been developed to manage modems and other remote access communication equipment by periodic polling.

Periodic polling of target devices is a technique commonly employed in managing communication networks. The Simple Network Management Protocol (SNMP), by far the most widely used network management protocol, is well-adapted to inquiry/response exchanges between a managing station and a managed device. Resident on the managed devices are software agents with well defined sets of manageable parameters which are gathered periodically by the station, stored and made available to network management applications for such functions as configuration, troubleshooting and reporting.

In particular, Concord Communications uses periodic SNMP polling in its network monitoring application, Network Health[tm]. Data is gathered from various network devices -- probes, routers, switches and servers -- stored in an Ingres database, and made available to a variety of applications for analysis and display. See Figure 1.

Network Health is primarily a reporting package. It offers summary reports on volume and utilization, daily exception reports on situations of interest, trend reports on polled variables, at-a-glance reports for quick overviews of network elements, and service level reports to determine whether quality of service standards are being met. Further information on Network Health reports can be found in [4] and [5].

In this paper we discuss how Concord has modified and extended Network Health[tm] to support remote access technologies: modems, ISDN interfaces and remote access servers, i.e., to monitor the SNMP agents associated with common vendors of this equipment and generate meaningful reports. The techniques are applicable to virtually any network management environment that relies on periodic polling.

Modems and remote access servers present several unique problems:

- Whereas data-oriented variables -- bytes and frames -- are appropriate volume and utilization metrics for most network devices, modems and ISDN interfaces are more readily characterized by time-oriented metrics, specifically call-seconds or call-minutes. These metrics need to be reduced, analyzed and displayed using analogous but nonetheless different methods.
- Modems and ISDN interfaces can occur in variety of states which need to be read, analyzed and displayed. While states are also relevant in other devices, remote access devices tend to have a larger number of interesting states, and such issues as state mappings become correspondingly more important.
- Many of the remote access agents reset counters at the end of each call (or the beginning of the next call). This feature, combined with periodic polling, results in data loss, which needs to be corrected or, at any rate, estimated to permit evaluation of the quality of the data.
- For management purposes, we are often interested in the aggregate properties of groups of modems and ISDN interfaces. We need to determine which modem groupings are of interest and how to combine data on individual modems to reflect the properties of the groups.
- Underutilized modems -- modems with large connect times compared to the bytes they transmit and receive -- represent a problem situation of considerable interest. This type of behavior needs to be detected and appropriate notifications issued.
- Modem pool saturation -- simultaneous use of all the modems in a pool, resulting in a blocking situation -- is an issue of considerable concern to network managers. This situation must be detected and reported appropriately.

This paper discusses how the basic techniques of periodic polling have been adapted to address these problems.

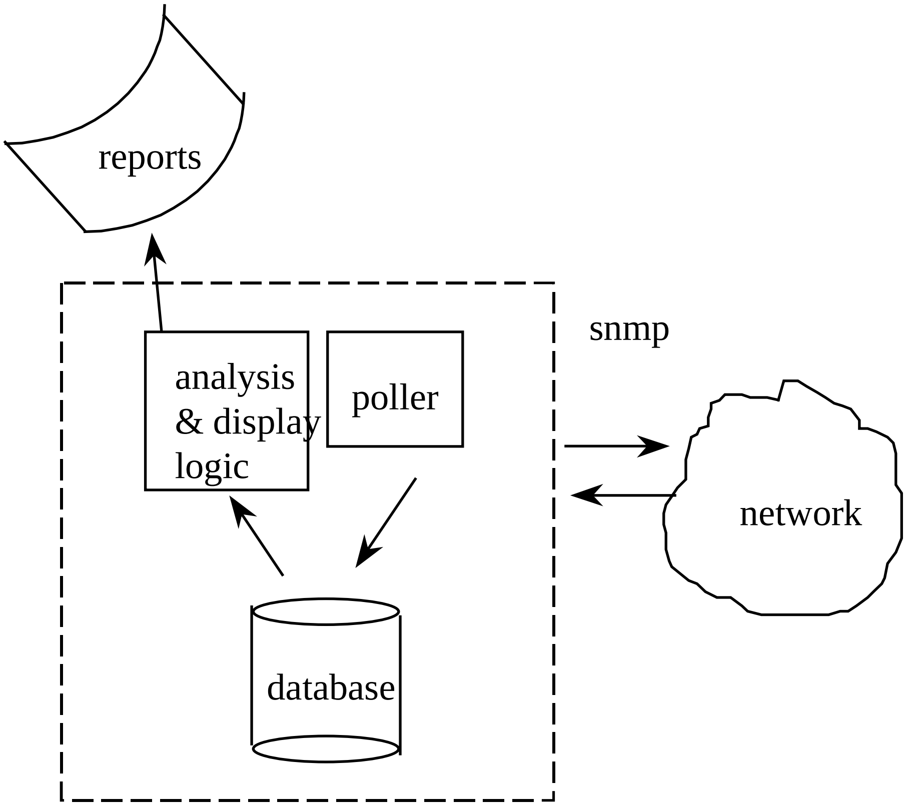

Figure 1: Network Health

2. Conceptual Issues

This section deals with several concepts necessary to an understanding of how remote access equipment is used and managed and what features are of interest in a network monitoring environment.

Modems and ISDN Interfaces

Modems and ISDN interfaces perform similar communication functions, yet are quite different in detail. Modems are designed to transmit data over analog telephone lines and perform digital-to-analog conversion to get the data onto the voice line and analog-to-digital conversion at the receiving end. ISDN interfaces, on the other hand, are entirely digital and require no conversion between analog and digital data. Thus ISDN variables tend to be a little bit different from modem variables. In particular, ISDN interfaces do not support retrains.

Retrains represent resynchronizations of two communicating modems and often result in a change in the speed of the associated connection. They typically occur when there is noise on the line and take several seconds. An inordinately large number of retrains is cause for concern. We make every effort to report on retrain counts where retrain counters are available. These counts are, of necessity, zero for ISDN interfaces.

While modems and ISDN interfaces perform pretty much the same functions, there are differences in the way they operate that translate into differences in the network management parameters supported by each. It thus becomes an open question whether to generate reports (and corresponding internal data structures) that treat them separately or view them as two slightly different manifestations of the same entity. We choose to do the latter, permitting us to reduce the number of reports supported and the consequent effort required to implement them, but pay a price in, for example, reports that might have a missing or inapplicable panel, and in suboptimal data structures.

In the discussions below, unless otherwise stated, we use the term *modem* generically, i.e., to refer to modems *or* ISDN interfaces.

Call seconds and Bytes per second

Throughput is a common parameter in Network Health reports. For a LAN or WAN link it is typically quoted in bits or bytes per second or, perhaps, as the fraction of the medium bandwidth that is actually being used to pass traffic.

Modems also pass bytes and the above metrics apply to them as well as to any medium capable of passing data, however bytes per second is of less interest than metrics associated with the time a modem is connected. For this reason, we focus our attention on connect-seconds or connect-minutes. For a single modem, the connect time is simply the time the modem is connected and utilization is connect time divided by total time.

For a remote access server or a modem pool (see below), we have multiple modems to consider and connect time is weighted by the number of modems connected. Utilization is actual connect time divided by potential connect time. For example, a remote access server with 16 modems can potentially sustain 23040 call minutes in the course of a day (number of minutes in a day multiplied by 16). We divide the actual number of call minutes by the potential call minutes to obtain the modem utilization. This is analogous to dividing bits per second of traffic by medium capacity, say, 10,000,000 bits/second to obtain line utilization for a traditional ethernet.

Many of the LAN/WAN and router reports offered by Network Health[tm] display data in bits or bytes/second. Analogous reports for remote access equipment use connect-seconds or connect minutes.

Modem Pools and Remote Access Servers

Often, a single agent serves multiple purposes. The single agent on a router provides information characteristic of the router as a whole and also of the various interfaces on the router, which may be managed separately. We would typically poll the agent for data on each of the interfaces and then sum or aggregate that data to obtain router-wide information. For example, we might poll the agent for incoming bytes for each of the individual interfaces and then aggregate the results to obtain incoming bytes for the entire router.

In a remote access environment, we typically have a remote access server (RAS) with an associated agent and a number of modems associated with that server and supported by the same agent. We poll the agent for individual modem data which we aggregate to obtain information characteristic of the entire RAS.

A number of Network Health reports deal with groupings of modems. In general, there are two types of modem groupings of interest: remote access servers and modem pools. A RAS is a group of modems within a physical chassis. A modem pool is likewise a group of modems but they are not necessarily confined to a physical chassis. There may be multiple modem pools within a single chassis or a modem pool may span multiple chassis. The concept of modem pool is intended to include the traditional idea of a "hunt group," but is a somewhat broader concept and extends to any logical collection of modems associated for any reason. See Figure 2.

Thus there are three ways modems can be managed:

(1) associating each modem with the RAS to which it is connected and managing the RAS as an entity as well as the individual modems.

(2) associating each modem with a logical modem pool and managing the modem pool as well as the modems.

(3) managing individual modems independently of any larger association.

From the standpoint of network management, there are similarities between a modem pool and a remote access server. They are both groupings of modems and their network management properties are primarily derived from the properties of their constituent modems. Of particular interest are two metrics designed to assess the degree to which a modem grouping is being used. While both metrics can be calculated for both groupings, they differ in terms of how useful they might be in one case or the other.

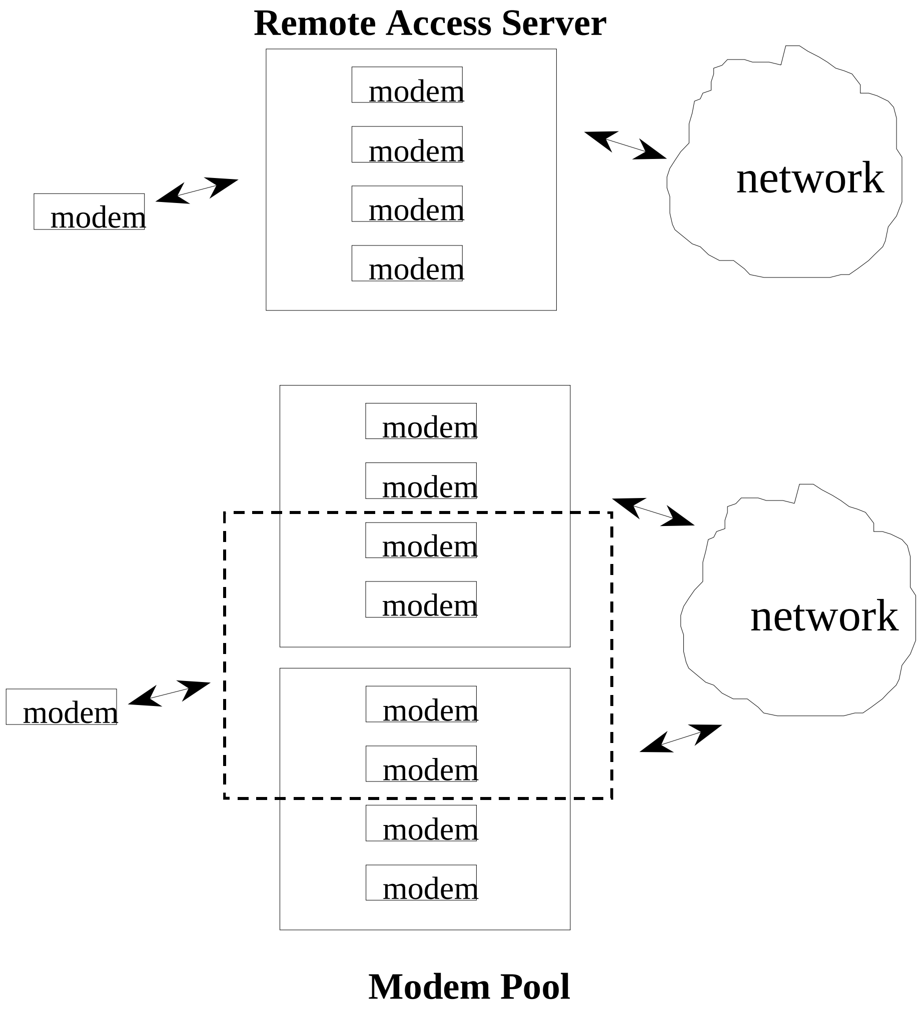

Figure 2: Modem Groupings

One, quite simply, is the percentage of modems that are actually being used at a given moment. As this variable approaches 100%, any new calls are increasingly likely to be blocked. It is particularly appropriate for modem pools, where blocking probability -- the probability that a caller cannot obtain a modem and hence receives a busy signal -- is of critical importance to network managers.

Utilization, as discussed above, represents the fraction of the modem pool or RAS capacity that is in use in terms of call-seconds divided by potential call seconds. It is more appropriate to a RAS than a modem pool, where we are less concerned with blocking probability than with how much our resource is actually being used and when utilization increases to the point that additional resources are required.

3. Management Information Bases

The variables of interest are contained in device-resident SNMP agents within a logical structure called a Management Information Base (MIB) and identified by hierarchical series of numbers called object identifiers. Most of the modem/ISDN information can be found in tables within the MIBs.

Generally, we are interested in the following information:

call duration	number of call-seconds or call-minutes the modem was connected
byte counters	bytes transmitted/received by the modem/ISDN interface
modem state	state of the modem at the end of a poll period
time of last change	time the modem state changed to the current one
connection speed	bytes/second of bandwidth allocated to the connection; this variable may vary over time within a connection and may be different for the incoming and outgoing directions
retrain counter	number of modem resynchronizations
connect counter	number of times the modem entered the connected state
error counters	various errors, including connect failures and frame errors

The various vendor MIBs [1,3,7,8] are, to say the least, less than consistent in the way they present information on modem and ISDN interfaces. Sometimes the information is in the nearly universal interface table *ifTable* [6] sometimes in proprietary tables, sometimes both; sometimes ISDN interfaces are included in the same tables as modems, sometimes they are in separate tables. It is frequently impossible to provide an unambiguous prescription for identifying ISDN interfaces of interest or distinguishing ISDN interfaces from modem interfaces. Examination of such parameters as *ifType, ifSpeed* and *ifDescr* often permit us to narrow down the field somewhat, but seldom yield unambiguous identification of the interfaces of interest all the time. When in doubt, we assume the interface to be a modem.

Another problem concerns proprietary tables. Much of the information of interest is often included in call tables, tables with variables on the current call. Unfortunately, these variables are reset at the end of the call or the beginning of the next call. In a periodic polling environment this feature presents a problem for variables which we wish to measure on a continuous basis, most particularly call duration. We attempt to sum them over all the calls within a time frame, however after each call the counters are reset to zero. If we are polling at, say, five minute intervals, we can lose the counters associated with portions of a call or, indeed, an entire short call.

4. Techniques Developed

This section describes the specific techniques developed to monitor remote access devices. Sample output for a RAS is shown in Figure 3.

Modem States

Modem state information is of interest for its insight into the behavior of an individual modem and also for its aggregate properties in remote access servers and modem pools. Unfortunately, there is little agreement among vendors on possible states. Lists of possible modem states vary widely and rarely are they the same as ISDN states, even for the same vendor. We need to manage modems and remote access equipment consistently to provide seamless support in environments where multiple vendors are being managed and also to avoid needless special casing in our software. To this end we define a set of "canonical" modem states into which we map the various collections of vendor states. Defining modem states requires us to balance two opposing tendencies. On the one hand, we want to define a sufficient number of states to differentiate among modem behaviors of interest. On the other hand, we want to limit the number of states to a reasonable number. We settled on the following seven states:

onhook	available for use
offhook	in the process of beginning or terminating a call
connect	call in progress
disabled	functionally impaired
busy	explicitly placed in the busy state, usually for management reasons
test	in the process of executing a test
other	any other state

State information is obtained through periodic polling of a modem state variable. However, in truth, this variable can only be read instantaneously at poll times and we are left to infer behavior over a continuous period from periodic samples.

In the crudest case, we have no further information and are forced to make correspondingly crude assumptions. Our approach here is to assume that the value of the modem state for the entire polling interval is the same as the polled value at the end of the interval. This convention misses very short calls that begin and end within a polling interval and imposes the granularity of a polling interval on our measurements of duration for longer calls.

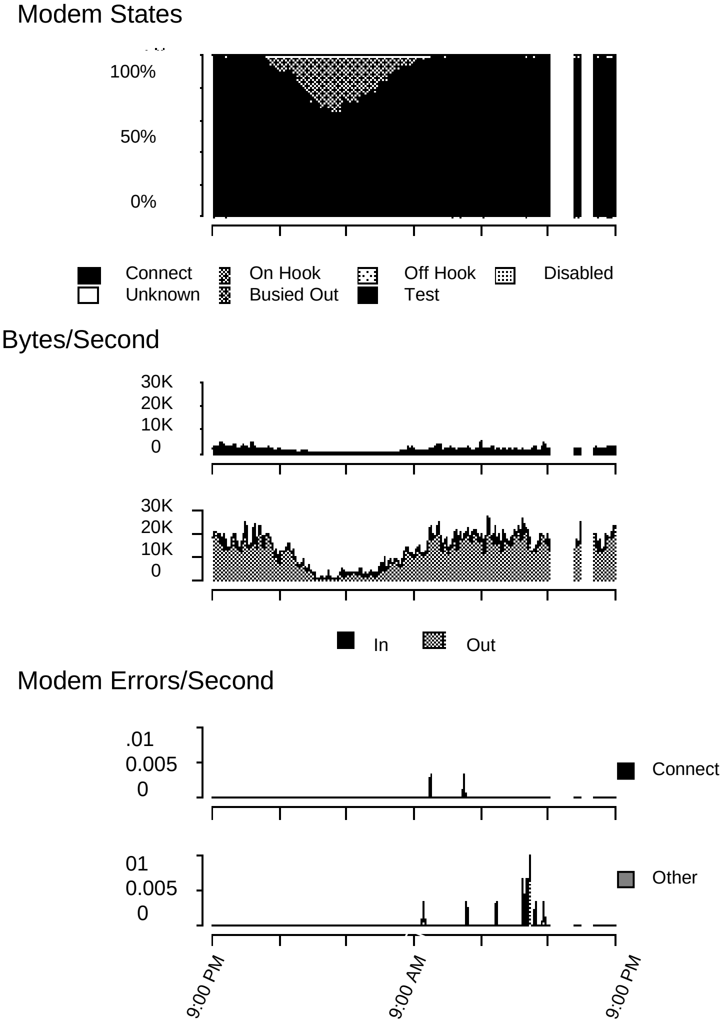

Figure 3: Sample RAS Output

If the MIB in question supports a call duration counter that increments every second or centisecond while the connection exists, we can improve our calculation of time in the connect state, since connect time can be determined by summing call durations. Yet even here there are uncertainties. Suppose, for example, the incremental value of call duration is less than the polling interval and, further, that call duration for the previous poll was zero, implying a short call that begins and ends within a single polling interval. We have no way of knowing where in the polling interval that call occurred, whether there was, in fact, a single call or multiple calls, or what the state of the modem was between calls. If we are primarily interested in the length of time the modem was connected over a period of hours or days, we generally make the assumption that all the modem connect time is embodied in the measured values of call duration, with the understanding that this is an underestimate owing to polling granularity, as discussed below.

Data Loss

Most of the information of interest to Network Health occurs in the agents in the form of counters. Counters increment monotonically until they reach the maximum value allowed by the number of bits in a word, at which point they wrap to zero and start all over again, performing a modular arithmetic. A counter is sampled at each poll. The value at the previous poll is subtracted from that of the current poll to obtain a counter difference, which is then stored in the database. The counter difference is later divided by the poll period, the difference in time between the two successive polls, to obtain an average rate.

For the most part, modem agents offer modem information in proprietary tables with indices that are keyed to the individual modems. Within these tables, relevant counters are accumulated for the duration of a call and then reset either at the end of the last call or the beginning of the next call. Calls typically last from less than a minute to several hours. Since we usually poll agents at five minute intervals, we are likely to run up against two problems:

(1) We can miss short calls entirely, if the entire call falls within a single poll period.
(2) We can lose some data from the beginning or end of a longer call.

Missed Short Calls. When a call is short compared to a polling interval, there is a significant probability that it can be missed entirely owing to reset of the call duration counter. The probability of data loss increases as the length of the call decreases relative to the polling period. We can reduce our data loss by polling faster, however polling consumes resources on the system, network and target device so there are limits to how much we can do this.

Longer Calls. Here the granularity of the polling interval and a characteristic of certain of the modem MIBs interact in an unfortunate fashion that results in a loss of a small percentage of the data. Such variables as call duration, bytes, frames, errors, discards and retrains are represented by monotonically increasing counters as long as a call is in progress. We can detect a counter reset by noting when the value of a counter for a particular poll is less than its predecessor.

We call this type of counter reset a pseudo-counter wrap, since it resembles a traditional counter wrap with one important difference.

Let t_n represent the time of poll *n*.
Let t_{n+1} represent the time of poll *n+1*.
Let c_n represent value of counter at poll *n*.
Let c_{n+1} represent value of counter at poll *n+1*.

Under normal circumstances, we expect $c_{n+1} \geq c_n$, and store $c_{n+1} - c_n$ as the incremental value of the counter (later to be divided by $t_{n+1} - t_n$ to obtain a rate). When $c_{n+1} < c_n$ for a true counter wrap, the counter is presumed to have exceeded its maximum value and started over from zero. The value of the increment can be obtained by subtracting the value $c_n - c_{n+1}$ from the known maximum value of the counter, usually 4,294,967,295 or 2^{32}-1.

In the case of pseudo-counter wrap, however, when $c_{n+1} - c_n < 0$, the counter has reached an unknown maximum, c_{max}, before starting over. There is no way of knowing c_{max}, so the best we can do is begin counting all over again from zero, i.e., use the value c_{n+1} as the increment for the polling interval, with the understanding that we are undercounting by $c_{max} - c_n$.

Estimating Lost Data. Precisely how much data we actually lose depends upon several factors:

(1) size of the polling interval
(2) whether counters are reset at the end of the call or the beginning of the next call
(3) distribution of call lengths
(4) distribution of time between successive calls
(5) distribution of counter activity within a call

To get a rough idea of the magnitude of our data loss, we assume that calls begin or end randomly within a polling interval. We expect, on average, to lose half a polling period worth of data for each call, for a fractional loss of:

$$\frac{t_{n+1} - t_n}{2 * (average\ call\ length)}$$

assuming the particular counter is distributed randomly over the call duration (this latter assumption is trivially true for call duration but may not be for other counters). The above expression neglects the information we recoup by using c_{n+1} as the increment for the problematic polling interval and so represents something of an overestimate of the amount of data lost. Figuring a poll period of 5 minutes and an average call length in the vicinity of 45 minutes (common figures), we expect a total data loss in the vicinity of 5%.

Underutilized Modems

An issue of interest to network managers is that of a modem that is used insufficiently, i.e., one that passes relatively little traffic in bytes in relation to the time it spends connected. This type of situation occurs when carelessness, management policy or limited modem availability encourage users to connect to a modem and leave the connection live for extended periods of time, even if little or no traffic is being passed. This section quantifies this notion and defines an algorithm that can be used to identify an "underutilized modem."

We select an hour as the period over which we wish to determine whether or not a modem has been underutilized. To this end we define four thresholds:

T1 average use threshold (bytes/call-minute)
T2 underutilized fraction threshold (fraction of hour)
T3 successful poll threshold (number of successful polls in the hour)
T4 connect time threshold (call-minutes during the hour)

We define a successful poll as one that returns an SNMP response with all the requested data and no errors. Thresholds *T1* and *T2* are used to determine whether an exception condition potentially exists. Thresholds *T3* and *T4* insure that the data is of sufficient quality that the exception is meaningful.

There are two criteria for flagging a modem as underutilized, one designed to deal with average use over an hour and another designed to catch cases where the average use is acceptable but the use is highly asymmetric, with much of the hour unused.

Let v_n represent volume in bytes and C_n represent connect time for successful poll n within a particular hour.

(1) We define average use as, simply, the total volume over the course of the hour divided by the total connect time.

$$\frac{\Sigma_n v_n}{\Sigma_n C_n} < T1$$

The sums are over all successful polls (the ratio is undefined if the denominator is zero). A problem situation is identified if this ratio falls below the indicated threshold.

(2) For asymmetric underutilization we first determine which of the successful polls represents an underutilized poll period:

$$\frac{v_n}{C_n} < T1$$

Let N represent the number of polls that satisfy the above inequality, and N_{tot} represent the total number of attempted polls (successful or otherwise) within the hour.

$$\frac{N}{N_{tot}} > T2$$

A problem situation is identified if the ratio exceeds the indicated threshold.

Two additional thresholds are necessary to insure that an exception is based on sufficient data.

(1) To eliminate cases of too few successful polls within the hour to get meaningful data, we need to impose the following condition:

$$N_S > T3$$

where N_S is the number of successful polls within the hour.

(2) To eliminate cases of too little connect time to provide a meaningful measurement, we place a threshold on connect time:

$$\Sigma_n C_n > T4$$

5. Conclusion

Network management of communications devices by periodic polling has been successfully applied to the vast majority of network devices capable of supporting SNMP agents. Extending this method to remote access devices presents a number of new problems owing partly to the unique characteristics of these devices, partly to conventions followed by the designers of the relevant MIBs and partly to the inherent limitations of the technique.

One of the most troubling problems, that of data loss, can be traced to the tendency of MIB designers to reset counters at the end of a call. Were these counters allowed to accumulate, or, where alternative cumulative counters provided, this problem could be avoided entirely. Other problems – inconsistencies among modem states, lack of consistency in where the modem/ISDN information is stored and ambiguity in distinguishing between modem and ISDN interfaces – can be traced to lack of consistency among the MIBs. While RFC1696 [2] does indeed represent a standard for managing modems, it seems pretty clear that it is often followed in the breach.

Nonetheless, as we have attempted to show above, it is possible to manage modems effectively with the technique, providing sufficient information to permit effective monitoring of the behavior of remote access servers and modem pools and their constituent modems and ISDN interfaces, and limiting ambiguities and data loss to generally acceptable levels.

6. References

[1] Ascend Communications, Inc., *Ascend Enterprise MIB,* copyright © 1993-1997 Ascend Communications, Inc.

[2] J. Barnes, L. Brown, R. Royston, S. Waldbusser, *Modem Management Information Base (MIB) using SMIv2, RFC1696,* 08/25/1994

[3] Cisco Systems, Inc., *CISCO-MODEM-MGMT-MIB*, Rev Date 9601110000z

[4] Concord Communications, Inc., *http://www.concord.com/products.htm,* copyright © 1997 Concord Communications and Onward Technologies

[5] Concord Communications, Inc., *Network Health User Guide, Release 4.0, Unix Platform,* copyright © 1997 Concord Communications, Inc.

[6] K. McCloghrie, F. Kastenholz, *Evolution of the Interfaces Group of MIB-II, RFC1573*, January 1994

[7] Shiva Corporation, *Shiva Enterprise MIB,* copyright © 1990, 1992, 1995, 1996 Shiva Corporation

[8] US Robotics/3Com, *Total Control SNMP MIBs*