A Bulk-retrieval Technique for Effective Remote Monitoring in a Mobile Environment

Glenn Mansfield Keeni  
Cyber Solutions Inc.

Kazuhide Koide  
Tohoku University

Takeo Saitoh  
Cyber Solutions Inc.

Norio Shiratori  
Tohoku University

Abstract

For effective management it is necessary to have methods by which management applications can periodically retrieve large volumes of information from network devices. In this work we examine the issues related to standards-based bulk-information retrieval and then propose a technique that extends the limits of periodic bulk-retrieval. The impact of the technique is demonstrated using a prototype implementation with special emphasis on network monitoring, traffic management and security management by comparing with the widely used RMON technology. We then discuss the problems of monitoring in a mobile environment in the context of the newly defined IETF standards track mobile IPv6 management information base (MIPv6MIB) and show how the bulk retrieval technique can be used to provide a mechanism for reliable monitoring in a mobile environment.

1 Introduction

Information about the network is essential for network operation, management, research and development. The standard network management framework of SNMP[1] provides a flexible, secure and powerful protocol for retrieving information from network devices. In SNMP a manager retrieves information from agents. Agents generally reside on network devices- routers, servers, etc. The Remote Monitoring MIB (RMON)[2] agent is a passive and dedicated agent that has a versatile and configurable design. RTFM[3], focuses on traffic metering and provides useful information for flow monitoring.

Network monitoring involves retrieving values of Managed Object (MO) instances from agents. If an agent is polled by a manager at intervals of \( N \) seconds, the manager effectively sees the value of the MO instance averaged over \( N \) seconds. Given that the information from an observation is \( \log_2 \frac{1}{p(x)} \) where, \( x \) is the observed value and \( p(x) \) is the probability of observing the value, there is loss of information when, instead of individual values only the average is available. Intuitively, we can say that the loss of information due to averaging is less when the average is closer to the maximum or the minimum value. On the other hand the loss of information is most when the average is exactly in the middle of the range. Assuming that the variable \( x \) is random i.e. \( p(x) = \text{constant} \), we can show that in the worst case the loss of information is given by \( (N-1)\log_2(X + 1) \) where, \( N \) is the number of instances that have been averaged and \( X \) is the maximum value of the observable.

To improve the information content, it is necessary to retrieve the values of \( n \) samples of an MO instance sampled at smaller intervals, say \( t \), over the period \( T = n \times t \) instead of the singleton value of the MO instance at the end of \( T \).

Statistical aggregates and parameterization using analytical models may suffice in profiling the macro network characteristics. For better estimation of characteristics like burstiness, latency and jitter and, for detecting and analyzing traffic patterns, particularly when the network bandwidth is large, measurements at smaller (sub-second) intervals are necessary. As the measurement interval gets smaller the number of values of an MO instance that need to be retrieved, within a given time, becomes larger.

SNMP uses Object Identifiers (OIDs) to represent the names of MO instances in the payload. The overhead due to the OIDs becomes critical as the number of retrievals gets large. This work describes a technique for reducing the overheads in SNMP data retrievals, and demonstrates the increased detail which such an approach can bring when observing network phenomena.

Work is ongoing to support mission critical applications in a mobile environment. An IETF standards track Mobile IPv6 Management Information Base (MIPv6 MIB)[4] has been defined. Yet, monitoring mobile devices is a little explored area. The traditional concepts and models of monitoring do not seem to hold well in a mobile environment as network connectivity cannot be taken for granted and the characteristics of the communication channels may fluctuate widely. Further, the 'monitorability' of a mobile device, which will generally depend on whether it is switched on, may be unpredictable. These call for careful examination of
the issues and for innovative solutions.

In section 2 we examine the issues involved in routine retrieval of bulk data. In section 3 we propose the concept of Aggregate Managed Objects and show how its use improves the efficiency of monitoring. In section 4 we describe some simple experiments and evaluate the impact of the technology. In section 5 we discuss the problems of monitoring in a mobile environment and then show how the bulk retrieval technique can be used to provide a mechanism for reliable monitoring in a mobile environment. Concluding remarks are presented in section 6.

2 Bulk and Precision Measurement - The Issues

There is a cost associated with polling agents for MOs. The cost is the network bandwidth due the network management traffic and the packet header processing overhead at the manager and the agent. This cost constrains the number of MO instances that can be polled and the interval at which polling can be carried out. In cases where the management traffic takes the same path as normal traffic, the observed value itself may be affected by the measurements.

SNMP allows bulk data retrieval e.g. by packing several MO instances in the same packet. These may be instances of different MOs or instances of the same MO sampled at different instants of time. Essentially, it envisages collecting information about several observations in a single query. This translates to lesser number of packets, reduced packet processing overheads and lesser bandwidth consumption due to lesser protocol headers.

For example, say sampling is carried out at intervals of \( t \), and each query retrieves information on \( n \) samples. Then by polling the agent at intervals of \( n + t \) the manager can obtain detailed information with a time-granularity of \( t \). The reduction in packet processing overhead and the protocol header overhead is directly proportional to \( n \). On the downside this bulk retrieval process introduces latency which increases with \( n \). Moreover, \( n \) cannot be indefinitely increased as large packets may cause fragmentation.

The SNMP payload essentially comprises of one or more [OID, value] pairs. In the example described later in this work (section-4), the MO instance is sampled at 10ms intervals and polled at 5 second intervals. A total of 500 MO instances values need to be retrieved at each poll. Given that the payload corresponding to a single OID is 12 bytes and that for the value is 6 bytes, the total payload would be of the order of 9 Kbytes. The traffic overhead problem becomes accentuated as the number of polled MO instances increases. In this example, a maximum 6 different MO instances may handled in the same packet as the UDP maximum datagram size is 64 Kbytes.

2.1 Related work

One way of bypassing the overhead of SNMP data retrieval is to use out of band mechanisms to fetch the bulk data. SNMP-based mechanisms may be used by the manager to specify the bulk data requirements to the agent and/or to trigger the retrieval. Another approach is to process the bulk data at the source and communicate the results to the manager.

In the payload compression approach, the PDU compression[5] technique proposes to use compression algorithms to compress the entire payload - but this technique is difficult to implement without defining a new PDU-type. The OID Delta Compression (ODC)[6] technique proposes to encode an OID as a delta of the previous OID. This achieves substantial reduction in the part of the payload that is due to OIDs. But it is dependent on the ordering of the OIDs and requires a minimal overhead for each OID. The SNMP Bulk-get reduces the size of the query but the response size remains unchanged. Other bulk-retrieval techniques[7] have focused on improving the efficiency of the SNMP Bulk-get along with variations of OID compression. RTFM[3] has introduced the convention of a DataPackage wherein the value of a single flowPackageData MO instance is actually a BER-encoded[8] sequence of the values of several RTFM-MIB MO instances. This reduces the number of OIDs in the payload.

3 MO Aggregation: The Concept

Unlike the previous approaches which have focused on payload reduction by compressing OIDs, we propose to de-
fine new Aggregate MOs which represent the set of target MOs. We introduce the concept of an Aggregate MO (AgMO). An AgMO is just another MO as far as the SNMP protocol is concerned. No new protocol operations will be required to handle these MOs. As in the case of any other MO, it requires additional instrumentation at the agent and at the manager. In this mechanism the user defines an Aggregate MO (AgMO) corresponding to one or more (predefined) MO instances (Fig. 1-(a)). Semantically, the value of an AgMO instance will be equivalent to the BER-encoded sequence of the values of the corresponding MO instances. The order of the concatenation will be determined by the order in which the MO instances are specified. With the definitions done, the user can do an SNMP ‘get’ on an instance of the AgMO to fetch the value of the constituent MO instances. Since, in the conventional case, OIDs for each of the constituent MO instances would be carried in the requests and the response, assuming that the average size of an OID is $S$ bytes and that an AgMO has $N$ constituent MO instances, the reduction in payload size is roughly $2 \times (N - 1) \times S$ bytes for each retrieval.

As a further refinement of the aggregation technique, we introduce the Time-based Aggregated Managed Object (TAgMO) shown in Fig. 1-(b). The TAgMO is an MO that represents the values of a (user) specified MO instance sampled at (user) specified intervals for a (user) specified number of times. In this case the user defines a TAgMO by specifying the MO instance that needs to be sampled, the sampling interval and the desired number of samples that will be included in one TAgMO. The value of a TAgMO instance will include the timestamp at which the first sample was taken.

To save on bandwidth and to avoid fragmentation, compression techniques may be used. Data compression is ineffective and sometimes counter-productive when applied to the individual values of conventional MO instances which may be relatively short. The value of an Aggregate MO instance is typically long and yields good compression ratio with standard lossless compression algorithms. Data compression introduces extra processing overhead at both the manager and the agent end. The trade-off between reduced bandwidth consumption, and reduced processing overheads at the manager and agents needs careful consideration. With powerful and/or dedicated managers and agents the processing overhead due to compression may become an insignificant issue.

4 Implementation and Evaluation

The Aggregate MO concept is realized as a MIB[9] which is implemented using the widely used net-snmp-5.0.6 package as base. The user defines the appropriate TAgMO using SNMP ‘set’ commands and then uses SNMP ‘get’ on an instance of the TAgMO to fetch the values of the constituent MO instance sampled at the specified interval.

The graph in Fig. 2-(a) shows traffic at 1 second intervals. A small part of the same graph is shown at 10ms intervals in Fig. 2-(b). The graph shows a clear pattern. The traffic is due to a series of consecutive SNMP-get-next queries and corresponding responses that traverses the MIB-II variables in a lexicographical order. The initial set of MIB-II variables are from the ‘system’ group and are replied to quickly by the responding host. This results in quicker query-responses cycles and consequently higher traffic. The subsequent set of variables in the MIB tree belongs to the ‘interfaces’ table. These require kernel reads causing the responses to be relatively slow. The SNMP query-response traffic reduces. For the routing table the traffic increases again probably because the routing table is built and cached in the agent. Subsequently the traffic falls when TCP and
UDP variables are serviced, followed by a rise in the traffic when the agent services the SNMP variables which are stored in local. Using aggregated MOs we can monitor traffic at finer time resolutions and get a deeper insight into the dynamics of the network and the applications running on the connected devices.

To evaluate the effect of MO aggregation we compared the three different cases. (a) traditional SNMP bulk-get (b) Aggregate MO retrieval without compression and (c) Aggregate MO retrieval with compression. For each case we measured the rate at which MO instance values were retrieved and the corresponding traffic, on an Intel Xeon 2.4GHz PC with a 1Gbps network interface. The rate of data-retrieval for constant packet size (Ethernet frame size 1500 bytes) is shown Fig. 3-(a). The network traffic due to a constant amount of data retrieval (30 MO instance values) is shown in Fig. 3-(b).

Using simple SNMP Bulk-get only 46 MO instance values could be retrieved whereas, using Aggregate MOs, this figure increased to 106. When compression (Lempel-Ziv coding) was used - the number of retrieved MO instance values increased to 538. The figure depended very much on the redundancy in the string representing the value of the Aggregate MO instance. In some cases the number of MO instance values that could be retrieved went well beyond 20,000.

5 Mobile Monitoring

Though considerable work has been done on mobility itself [10][11][4], there has been little investigation on monitoring mobile devices. There is a basic difference in monitoring devices in a static environment and monitoring devices in a mobile environment. In usual non-mobile environments network connectivity is taken for granted - loss of connectivity is a ‘fault’ condition. On the other hand in a mobile environment loss of connectivity is not a network fault. It is more likely due to the changed physical environment into which the mobile device has moved. Also, the characteristics of connectivity, when it is there, may show wide variations e.g. wide variations of RTT are routine. Fig. 4 shows a sample of RTT values experienced by a manager monitoring a vehicle in a MIPv6 network environment.

To explain some of the issues involved in monitoring a mobile device, say we are monitoring the longitude ($X_i$) and latitude ($Y_i$) of a vehicle at times $T_i$. By plotting the $(X_i,Y_i)$ values on a map the route of the vehicle may be traced.

In Fig. 5-(a) the allow shows the actual route of the vehicle. The communication between the monitor and the vehicle was disrupted on the route from A to C and from C to B probably because the vehicle entered a tunnel. The route reconstructed from the monitored data is in Fig. 5-(b).
5.1 The problems of data collection from mobile entities

For network monitoring generally the unreliable User Datagram Protocol (UDP) is used. It is considered a better candidate than the reliable Transmission Control Protocol (TCP) due to its statelessness and light implementation. Monitoring applications using the standard Simple Network Management Protocol (SNMP) generally use a timeout and maxretries parameter. A query is sent, if there is no response within timeout duration a retry is carried out unless maxretries retries have already been made. It is clear that the timeout parameter will depend on the RTT between the monitoring application and the monitored device.

Another problem that appears in, but is not limited to, the context of a mobile environment is that of intermittent agent activity. For example, if the onboard computer on an automobile is being monitored, the computer is likely to be powered up at ignition and powered off when ignition is turned off. In a mobile environment the manager faces an additional problem trying to figure out whether the lack of response is due to lack of network connectivity or the agent is not up.

5.2 The solutions

It would be advantageous for a monitoring application to have larger timeout and maxretries to cope with the contingency of a large RTT and/or occasional packet loss. But, to ensure that sampling is carried out within an interval, we have the constraint 
\[
\text{timeout} \times \text{maxretries} < \text{interval}
\]

It is clear that timeout and maxretries cannot be arbitrarily increased without losing data. On the other hand if interval is increased then the information loss increases.

The Data Aggregation technique described in section 3 can be advantageously used to increase the polling interval without losing information. That will allow larger timeout, larger maxretries without any degradation of information content.

5.2.1 Tagged and Persistent Polling

In this section, we describe the Tagged and Persistent Polling method which takes care of intermittent network connectivity as well as intermittent agent activity.

At the agent, the data is tagged and stored in a time sequenced manner at regular intervals ‘p’ e.g. at some time instance ‘t’ the value of the observable is \( data_t \). The timestamp of the data itself may be used as a convenient tag.

The data cached at the agent is shown in Fig. 6. Note that there will be ‘holes’ in the data. This will happen when the agent is not active (e.g. when the mobile device is powered off). If there is a request for non-existent data then the agent will reply “no data”.

The manager polls the agent at some polling interval ‘P’ for the tagged data. \( P = n \times p; \text{ integers } n, p > 0 \) The manager asks for the value of the observable at some time ‘t’. In other words it asks for \( data_t \).

There will be 3 possible responses:

- \( data_t \) will be returned
- response “no data” will be returned
- There will be no response : Timeout

The manager will act based on these responses as shown in the algorithm below.

Note that we can conveniently define a Time-based Aggregated Managed Object such that \( data_t \) is actually a concatenation of \( n \) values of the observable starting at time \( t \) and sampled at some interval \( p \). That will allow us to have results of sampling at \( p \) second intervals by polling at \( P \) intervals \( (n \times p = P) \).

5.3 Evaluation

We used the Tagged and Persistent Polling method for collecting data from a moving vehicle using MobileIPv6.
Manager’s clock is $T$.
Manager wants to collect data from time $t$ onwards ($t < T$).
(Data is probably collected up to time $t - P$)

while ( )
try to get $data_t$
if the response is ‘no data’ ||
response is $data_t$
if ($t + P > T$) # polling mode
    $t = t + P$
    sleep ($T - t + P$);
else # quick retrieval mode;
    # try the next data point
    $t = t + P$
if the response is Timeout # persistent mode
    # Continue trying for same data point
    $t = t$;

Figure 7. Algorithm

The underlying link layer was provided by the PHS[12] system used in Japan. We compared the result with normal polling. Due to poor link layer connectivity, not more than 70% of the data could be collected using traditional polling. Using the proposed method remarkable improvement was observed. All the data could be collected. The data collection delay was reduced. Further, as a side effect of the timestamp-tagging, the accuracy of the data timestamps improved.

5.4 New Applications

The Tagged and Persistent Polling method provides a powerful mechanism for reliable and accurate monitoring of mobile devices. This opens up a new genre of applications starting from simple information collection from probes attached to moving vehicles to mission critical applications monitoring patient information in ambulances.

6 Conclusion

In this work we have proposed a bulk retrieval technique that fits well within the standard network management framework of SNMP. It drastically reduces the payload burden of the OIDs. The payload is further reduced by applying compression techniques. Experiments with a prototype implementation of the proposed technique have been carried out. The results, compared with results from RMON based mechanisms, show the effectiveness of the technique in network operations and management. We have examined the issues of monitoring in a mobile environment in the context of the newly defined IETF standards track Mobile IPv6 MIB. We have presented a new Tagged and Persistent Polling algorithm that uses the bulk retrieval technique for reliable and effective monitoring. Prototype implementations are used in the Internet-Car (ICAR)[13] project to retrieve bulk information from moving vehicles over narrow bandwidth wireless links using SNMP. We believe the technique expands the scope of network monitoring using SNMP into mobile and mission critical applications.

References