WAN Event-Driven Diagnosis Based on SNMP Delegates
– IEICE Technical Report FTS 97-xx –

Elias P. Duarte Jr. **Takashi Nanya** **Glenn Mansfield** **Shoichi Noguchi**
{elias, nanya}@cs.titech.ac.jp **{glenn,noguchi}@tia.ad.jp**

[1]Graduate School of Information Science, Tokyo Institute of Technology
2–12–1 Ookayama Meguro-ku Tokyo 152, Japan

Abstract. *In the manager-agent network management paradigm, if a fault occurs on the network route from the manager to an agent, it causes false alarms and a collapse of the management entity. This paper proposes an approach to solve this problem. With network topology information, a set of proxy nodes are invoked whenever the corresponding network routes are breached. For a given node that becomes unreachable, the NMS uses that node's neighbors as proxies, through which it tries to reach the original node. In case a neighbor is found to be unreachable, the algorithm is applied to that neighbor, and so on. Eventually the NMS diagnoses which nodes are unreachable in the network. An effective MIB to implement the proxies is presented, which allows their deployment at virtually no cost.*

Keywords: Network Management, SNMP, Fault Management, Diagnosis

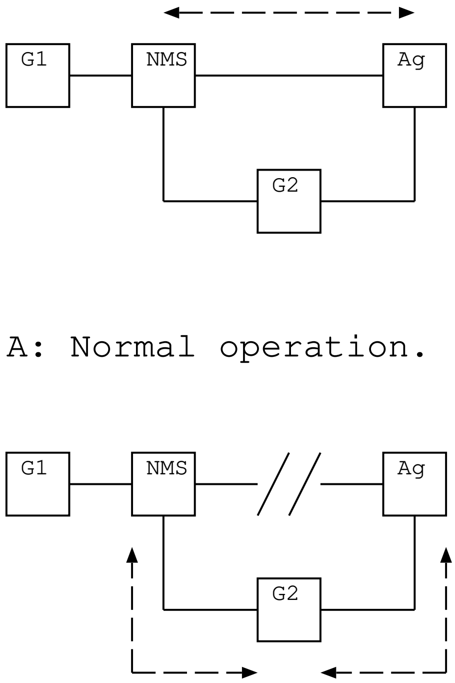
1. Introduction

As computer networks expand, there is a pressing need for integrated network management systems capable of handling errors. In the manager-agent network management paradigm, a network management station (NMS) monitors the network by querying agents about the state of communication nodes, links and protocol entities. Agents may also issue alarms to inform about abnormal conditions. If a fault occurs on the network route from the manager to an agent, the network management system will be unable to determine the state of part of the network. Instead of helping to solve the fault, the management system becomes partially non-operational because of it. In current systems this situation causes false alarms and a collapse of the management entity, as it is an application layer entity and has no control over network routing. This paper proposes an approach to solve this problem. With network topology information, a set of proxy nodes are invoked whenever the corresponding network routes are breached. This approach constitutes an event-driven diagnosis algorithm. For a given node that becomes unreachable, the NMS uses that node's neighbors as proxies, through which it tries to reach the original node. In case a neighbor is found to be unreachable, the algorithm is applied to that neighbor, and so on. Eventually the NMS has a picture of which nodes are unreachable in the network. An effective MIB to implement the proxies is presented, which allows their deployment with virtually no cost.

2. SNMP Delegates

Consider the simple network topology in figure 1, where the NMS is connected to an agent (Ag) and also to two gateways, G1 and G2. Considering communications involving the

NMS and the Ag, suppose that routing is such that the direct link is used to communicate the queries and replies, as shown in part A of the figure. If the link between the NMS and agent fails, network management queries will be delayed until the network layer recovers from the error. The delay may be significant as a new route for the agent should be discovered. During this delay, the NMS won't be fully operational, and it coincides exactly with the moment that there is an error to be solved. This weakness could be solved if a management entity could relay the queries from NMS to AG and the corresponding replies from AG to NMS, as shown in part B of the figure. The condition to obtain this solution is that the routes used by the new management node be available when a failure occurs in the network route between manager and agent. The NMS should verify which nodes can act as proxy. For the example in figure 1, G2 is selected to be a proxy.



A: Normal operation.

B: Fault recovery.

Figure 1. Management communication routes.

Although management communication happens between the NMS and the agents, the paths used also employ ordinary communication nodes, that are not necessarily agents nor managers. A node may be an NMS, agent or an ordinary communication node. Even if between a pair of vertices there are multiple different paths, the communication occurs along a precisely defined route. The route selection is carried out by the network layer protocol entities. This path is called the *network route* between two points. If there is a failure along any component of a *network route*, communication between the end points is breached until the network adapts itself to the failure and sets up a new network route.

An *application route* is a concatenation of one or more *network routes*. It serves the purpose of bridging or relaying messages. In the case of a failure along a network route used for manager-agent communications, the agent becomes unreachable through that route. But a set of other network routes can be employed that provides an alternative route to reach the agent, that route is an application route.

For a simple network topology like that of figure 1 the position of the proxy is quite obvious, but for a more complex network, it is not a simple decision. A previous paper has shown an algorithm for placing an optimal set of proxies in [1]. This algorithm is based on network layer routes. In [2] it was shown that network routes may not persist for long periods or not. Thus an algorithm for placing proxies must be independent of network routes. The algorithm presented in the next section depends only on the physical topology of the network.

3. Diagnosis Using Delegates

In this section we present the algorithm for diagnosing a fault that has made a node to become unreachable. At the same time, the algorithm allows the node to be reached through an alternative fault-free path, if one exists.

The idea on which the algorithm is based is intuitive, as can be seen from the following analogy. If you lose contact with a friend, and want to know about him, what would you do? A good idea would be to ask information to somebody that you know have a relationship with that friend.

The algorithm does the same: when an agent becomes unreachable, the NMS tries to reach that agent through the agent's neighbors. The NMS has the topology of the network and can use any neighbor of a node that has become unreachable as a proxy. In turn, a neighbor may also be unreachable, and the algorithm can be applied recursively, i.e. using the neighbors of the neighbor as proxies.

This algorithm is possible because network routes are not transitive, i.e., the route from node A to node C is usually different from the route from node A to node B to node C. Thus when an agent is unreachable through the direct network route, it may be possible to reach that agent through an alternative application route.

Application routes allow a network application to have a routing engine that gives tolerance of faults on the network layer routes. Whenever the network route is not working, the application has an alternative.

It should be clear that the algorithm terminates, for at some point either reachable nodes are found, or even the neighbors of the NMS are unreachable. The result is that the NMS ends up diagnosing the cause why the node has become unreachable, i.e. which links on the path to that node are timing out. The algorithm is shown below:

```
Algorithm Rescue (node);

Begin

 mark node as visited;
 for every neighbor of node do
   if neighbor is NOT unreachable
   then use neighbor as proxy for node;
 end_for;
 if some proxies report that node is alive
 then mark node as fault-free;
 else if all proxies report time out
      then mark node as unreachable
      end_if;
 end_if;

 if some proxies report that node is not alive
 then mark respective links as timing-out;
 end_if;

 for every neighbor that didn't reply &
     has not been visited
  rescue(neighbor);
 end_for;

End.
```

To allow this algorithm to be implemented effectively, any agent must be able to act as a proxy, which is allowed by the implementation shown in the next section.

3.1. Example

Consider the simple network in figure 2. The dashed lines show the network route used by the NMS to communicate with the agent.

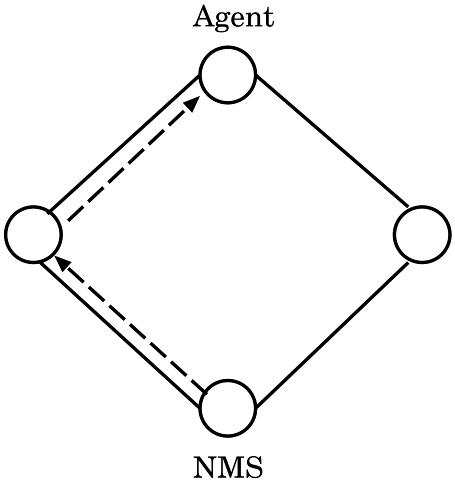


Figure 2. Direct route between NMS and agent.

Now consider that the agent is faulty. Using the proposed algorithm, the NMS activates both agent's neighbors as proxies, as shown in figure 3. As both proxies reply that the agent is timing out their queries, the NMS concludes that the agent is unreachable.

In case of the link fault illustrated in figure 4, one of the proxies delivers the query/reply between NMS and agent, while the other reports that the agent is timing out. The NMS can thus conclude that the specific link is faulty.

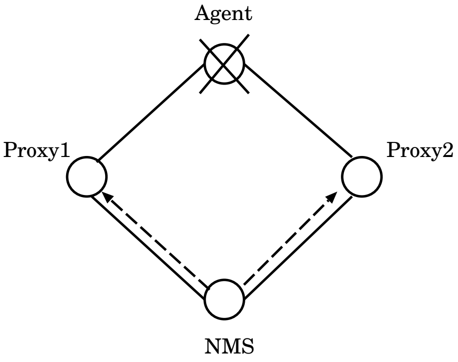


Figure 3. The agent is faulty.

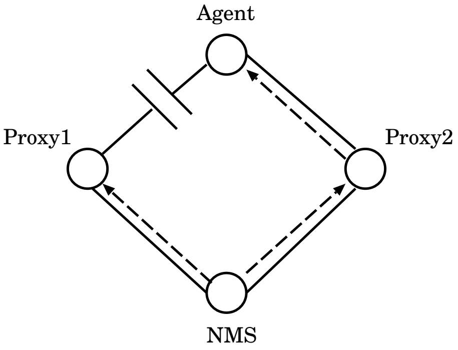


Figure 4. One link is faulty.

Another link fault, shown in figure 5, requires that NMS uses the agent as a proxy. When trying to reach the agent through its neighbors, one of the neighbors times out. The other neighbor successfully delivers the query/reply between NMS and agent. The NMS proceeds thus to try to reach the node that timed out though that node's neighbors, which is the agent. It may be required to use source routing to use the agent as proxy. As that communication succeeds, the NMS can conclude that the specific link is faulty.

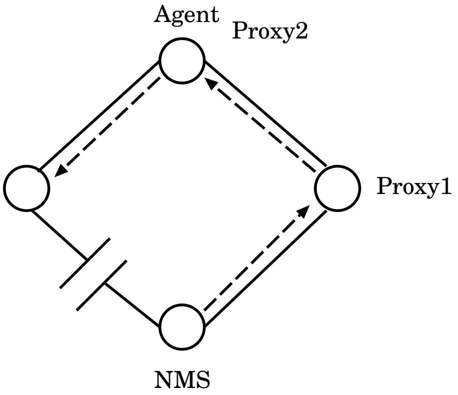


Figure 5. Another link is faulty.

4. Implementation

The proxy was implemented as a conventional SNMP MIB: a simple and flexible approach that allows any agent to become a proxy with virtually no cost. For a reference on the structure of management information we suggest [3].

The MIB contains a table of which a row is made up of the following objects:

```
RproxyEntry ::= SEQUENCE {
  agentAD IpAddress,
  mgmtOBJ OBJECT IDENTIFIER,
  commPXY DisplayString,
  resultPXY DisplayString }
```

The NMS sets the address of the agent to be queried in variable *agentAD*, the object identifier to be queried in *mgmtOBJ*, and the community that should be used in *commPXY*. After that, by querying the *resultPXY* object, the proxy will issue an snmpget on the agent whose address is *agentAD*, for the object whose identifier is *mgmtOBJ* and using *commPXY* as the community. The result of the query is sent back to the NMS.

5. Conclusion

In this paper we presented a practical approach to implement event-driven diagnosis for centralized network management systems based on SNMP. An algorithm based on the concept of neighbor proxies was introduced, that allows the NMS to determine the source of problems on the path to an unreachable agent.

Although this is an effective approach to handle emergencies in current systems it cannot be used to diagnose the whole network like the algorithm presented in [4].

References

1 E.P. Duarte Jr., G. Mansfield, and S. Noguchi, *Improving the Dependability of Network Management Systems*. Proceedings of the 10^{th} IEEE International Conference on Information Networking, Osaka, December 1994.

2 V. Paxson, *End-to-End Routing Behavior in the Internet*. Proceedings of the ACM SIGCOMM, Stanford, 1996.

3 J. Case, K. McCloghrie, M. Rose, and S. Waldbusser, *Structure of Management Information for version 2 of the Simple Network Management Protocol (SNMPv2)*. RFC1442. April 1993.

4 E.P. Duarte Jr., G. Mansfield, T. Nanya, and S. Noguchi, *Non-Broadcast Network Fault Management Based on System-Level Diagnosis*. Proceedings of the IFIP/IEEE International Symposium on Integrated Network Management, San Diego, May 1997.