

HIERARCHICAL END-TO-END SERVICE RECOVERY

Mohamed El-Darieby¹, Dorina Petriu¹ and Jerry Rolia²

¹*Systems and Computer Engineering Dept., Carleton University, Ottawa, ON, Canada*
K1S5B6 {mdarieby, [petriu](mailto:petriu@sce.carleton.ca)}@sce.carleton.ca

²*Internet Systems and Storage Lab., Hewlett-Packard Labs, Palo Alto, CA, 94304, USA*
jar@hpl.hp.com

Abstract: Failed networks, for example MPLS, can cause signaling storms the size of which can grow dramatically with network size. This paper presents a new scalable fault notification protocol that reduces the size of this storm. The protocol causes failure notification signals to travel vertically up and down a network hierarchy instead of horizontally along the service routes. This reduces the signaling required for longer paths and provides signaling scalability for larger network sizes. The protocol also alleviates the number of fault notification signals by hierarchically aggregating them. Aggregation is based on path management information maintained in the network hierarchy. We show how the protocol reduces the size of the signaling storm analytically and with simulation. However, this comes at the price of increased storage overheads when compared with existing restoration techniques.

Keywords: Service recovery, Hierarchical networks, Fault notification, MPLS

1. INTRODUCTION

The efficient provisioning of network services is a competitive differentiator among network Service Providers (SP). A SP has to ensure service survivability by recovering network failures in a responsive manner. In general, end-to-end service recovery requires the notification of ingress network nodes about such failures. Fault notification constitutes a major part of recovery time and is a fundamental criterion for evaluating the performance of the recovery process [3]. It results in flooding the network with recovery signals (a.k.a. signaling storm [2]). The magnitude of the

signaling storm grows dramatically with network size because 1) services are expected to span larger numbers of network nodes as the network size grows; and 2) the number of services carried over an individual link grows proportionally as the transport technology matures (e.g. OC 92 versus OC 3 links).

In this paper, we propose a new Scalable end-to-end Path Recovery Protocol (SPRP) that addresses scalability aspects of the fault notification problem. SPRP works in conjunction with a service creation protocol called the Hierarchical Distributed Routing Protocol (HDRP) [9]. Both HDRP and SPRP use a standard hierarchical network structure (e.g. the ATM PNNI). However, in SPRP and HDRP, hierarchical nodes/ (Bandwidth Brokers) participate in service provisioning (routing, signaling & fault notification), whereas Peer Group Leaders in PNNI only reflect static network connectivity information and do not participate in fault notification.

To provide signaling scalability for longer paths, failure notification signaling messages travel vertically up and down the network hierarchy instead of horizontally along the routes traversed by the paths. This is enabled by the path state information maintained at nodes at higher levels of the hierarchy by HDRP.

SPRP alleviates the number of fault notification signals by aggregating signaling messages based on the physical proximity of ingress routers. The protocol uses path management information maintained in the network hierarchy to aggregate the signals. A single notification message will be raised per path aggregate (instead of one per individual path).

We evaluate analytically the performance of SPRP and compare it to existing fault notification approaches. Analytical analysis gives insights into why and how SPRP reduces the size of the signaling storm compared to these approaches. The simulation experiments we conducted confirm these analytical results.

To the best of our knowledge, this work is the first to propose hierarchical end-to-end service recovery. We present SPRP in the context of Multi-Protocol Label Switching (MPLS) [5, 13] because MPLS is the lowest layer of the network architecture that has knowledge of end-to-end paths. This implies that the fastest end-to-end path recovery is achievable at this layer. In addition, MPLS is becoming the *de facto* standard for controlling many IP-based network transport infrastructures [2]. An MPLS service is a path/ route that traverses many networks.

This paper is organized as follows: In section 2, we describe the proposed recovery protocol, and in the following section, we relate our protocol with other research and standards. Section 4 provides analytical analysis and simulation results regarding the overhead of SPRP and compares it to other schemes. Conclusions are offered in the last section.

2. HIERARCHICAL SERVICE RECOVERY

This section describes our approach to hierarchical networks, the Scalable Path Recovery Protocol and introduces signaling aggregation to enhance its performance.

A hierarchical network is a traditional solution to the network-scaling problem [12]. Nodes are organized into different interconnected sub-networks called Autonomous Systems (AS)/ domains. An AS consists of a number of interconnected network nodes (i.e. IP routers or ATM switches). We assume Service Level Agreements are in place among ASs. A Bandwidth Broker (BB) manages and maintains topological and state information about the nodes and links of an AS. BBs are connected to each other and to physical nodes with fault tolerant connections.

The process of grouping BBs (at one hierarchy level) into logical ASs and abstracting such ASs via a BB (at the next higher level) is done at all levels of the hierarchy (see Figure 1). The physical nodes of the network belong to level 1, and their managing BBs to level 2. Level-2 BBs are interconnected in a manner that corresponds to the interconnectivity of level-1 ASs. Level-1 BBs are grouped into logical level-2 ASs. Each level-2 AS is represented and managed in turn by a level-3 BB, which maintain topology and state information about level-2 ASs, and so on. This hierarchy is assumed to be static. Fault-tolerant signaling channels between different BBs (same-level and child-parent) are also assumed. Internet standards (e.g. OSPF and BGP) propose a two-level hierarchy. The ATM PNNI standard supports a hierarchy of up to 105 levels [12].

BBs are software processes that can run on network nodes or on separate server nodes. We assume the latter approach in accordance with recent directions on the surveillance and control of the MPLS/TE-based networks that suggest the use of a centralized resource manager (i.e. BB) within each Autonomous system [15, 16]. Server nodes are cluster-based server farms that can grow in capacity based on the intensity of route calculation requests. Server nodes are implemented in branch offices or Internet Data Centers. As a result, storage and communications overhead are not a system concern.

2.1 The Scalable Path Recovery Protocol (SPRP)

SPRP solves the problem of very long paths by signaling vertically, instead of horizontally, up and down the network hierarchy. We assume that SPRP messages have highest priorities. Figure 1 gives an example of SPRP signaling. The SPRP algorithm is shown in Figure 2. We use the following notation:

- v_{ik} : a network node or a BB with address i , at level k . Physical nodes have $k = 1$ and BBs have $k > 1$
- pid : (ID, v_{sk}) : a pair that uniquely identifies a path
- $Parent_BB_k(v_{il})$: a level k BB that has a view of node v_{il}
- P_{ik} : $(i, v_{sk}, v_{ak}, v_{bk}, \dots, v_{tk})$: a level k path with $pid = i$ that has v_{sk} as ingress node, v_{tk} as egress node
- $Plist_k$: list of $pids$ of failed level k paths
- $AggP_{jk}$: $(j, v_{jk}, v_{(j-1)k}, \dots, v_{sk}, Plist_k)$: a level k aggregate route that aggregates a set of $AP_{i(k-1)}$ with $pids$ in $Plist_k$ and that traverses the route listed

We use the “.” notation to refer to members of a data structure. For example, to refer to the ingress router for path P_{ik} , we use $P_{ik}.Ingress$.

SPRP is executed in response to a failure detected by a node, v_{fl} . A *Failure* message is sent vertically up the hierarchy that contains a list of *pid*'s and ingress routes of paths affected by the failure. As a $Parent_BB_k(v_{fl})$ receives a *Failure* message, it scans the list of affected paths and if it views P_{jk} . Ingress where $P_{jk} \in Plist_k$, it notifies its child BB that views this ingress router. This is accomplished by sending a *FailureNotify* message down the hierarchy. The entry for P_{jk} is then removed from $Plist_k$. Then, $Parent_BB_k(v_{fl})$ sends a *Failure* message up the hierarchy to its parent BB (i.e. $Parent_BB_{k+1}(v_{fl})$).

This is repeated up the hierarchy until $Parent_BB_{L+1}(v_{fl})$, which we call *RootBB*, is reached. *RootBB* is defined to have a view of *all* ingress routers of the paths that belong to $Plist_L$. *RootBB* sends a *FailureNotify* message down the hierarchy to each $Parent_BB_L(P_{iL}.ingress)$ where $P_{iL} \in Plist_L$.

A $Parent_BB_k(P_{jk}.ingress)$ that receives a *FailureNotify* message forwards it down the hierarchy to notify $Parent_BB_{k-1}(P_{jk}.ingress)$ about the failure in P_{jk} . This is repeated down the hierarchy till the *FailureNotify* message reaches $P_{jk}.ingress$. This completes the notification process for this path.

As an example, Figure 1 shows three uni-directional paths that are affected by a failure in the link connecting nodes 2.1.2 and 2.3.2. The paths follow the thick lines at level 1. The ingress and egress nodes of the three paths are as follows: 2.3.1 and 1.4.1 for path #1, 2.1.5 and 2.2.3 for path #2 and 4.3.4 and 2.2.1 for path #3. The master node for the failed link (node 2.1.2) sends *Failure* message (#1 in Figure) to BB 2.1 that contains the *pids* of the affected paths (#1, 2 and 3). BB 2.1 scans the list of ingress nodes of failed paths looking for the nodes that it has a view of. It sends a *FailureNotify* message (#2) to node 2.1.5 about the failure in path #2 and removes the *pid* for path #2 from the list of affected paths. As the list is not empty yet, BB 2.1 sends a *Failure* message (#3) to BB 2. BB 2 finds the ingress node of path #3 to be within its view, so it sends *FailureNotify* message (#4) to BB 2.3 and removes path #3 from the list. BB 2.3, in turn, sends a *FailureNotify* message (#5) to physical node 2.3.1. Now, BB 2 realizes that some paths (i.e. #1) are out of its view, so it sends a *Failure* message (#6) to BB *. BB * has a view of node 4.2.1 so it sends a *FailureNotify* message (#7) to BB 4. BB 4 notifies BB 4.3 that in turn notifies node 4.3.4 of the failure (message #8 and #9). Now, BB * realizes there are no path left to be processed and it stops forwarding messages up the hierarchy.

The number of SPRP vertical notification messages required for each affected path will be $2*(K-1)$ where the BB that has a view of the failure master node and the ingress router of the affected path is at level K . One extreme case (e.g. path #2) is if the two nodes are in the same network domain then *RootBB* is at level 2 and the number of notification messages is 2. The other extreme (e.g. path #1) is where *RootBB* coincides with the root of the network hierarchy (level 4) where the number of notification messages will be 6 messages.

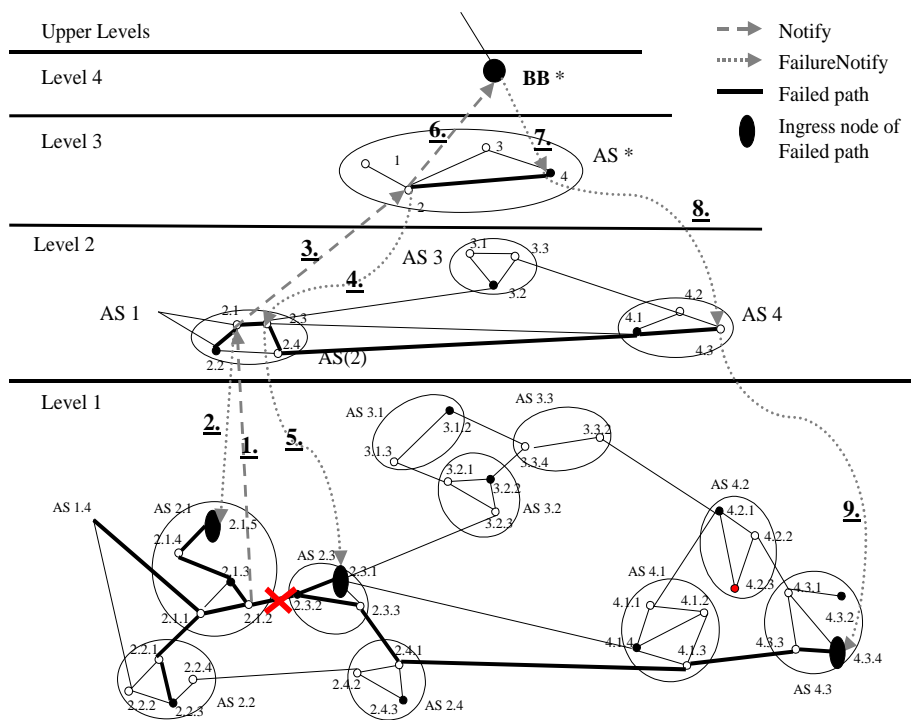


Figure 1. The Hierarchical Restoration Protocol

2.2 SPRP with Signaling Aggregation

To enhance the performance of SPRP, we propose to take advantage of the physical proximity of the ingress routers of some of the affected paths, such that one notification message is sent for them instead of a message for each path. This exploits path management information maintained in the network hierarchy by HDRP.

Information about individual paths that traverse the same route and that have ingress routers within a network domain can be aggregated. Aggregated information describes an *aggregate path* such that network nodes along this route identify the set of individual paths (the aggregate path) by a single *pid*. The concept of an aggregate path resembles that of Aggregation Areas [11]. An individual path is a (non-aggregated) physical or abstract path. HDRP implements path aggregation at all levels of the network hierarchy to reduce information storage overheads.

Aggregation is recursive. An aggregate path may aggregate a number of aggregate paths, each of whom aggregates in turn a number of lower level (aggregate or individual) paths. In other words an aggregated path is a rooted hierarchy (tree) of aggregate and individual paths.

```

Switch (msg →type){
  Case Failure (FailedPlist(l-1)) {
    Initialize FailedPlistl = NULL;
    For (each path  $P_{kl} \in \text{FailedPlist}_{(l-1)}$ ){
      If ( $v_{il}$  is Parent_BB ( $P_{kl}, v_{s1}$ ))
        Send (FailureNotify,  $P_{kl}, v_{s1}$ , FailedPi(l-1));
      Else{
        If ( $P_{kl} \in \text{AggP}_{jl}$ ) {
          If ( $\text{AggP}_{jl} \notin \text{FailedPlist}_l$ )
            FailedPlistl += {AggPkl};
            FailedPlistl → AggPkl → Plistl +=  $P_{kl}$ 
          }
        Else FailedPlistl += {AP( $P_{kl}$ );}
      }
    }
    Send (Failure, Parent_BB ( $v_{il}$ ), FailedPlistl);
  } /*end case*/

  Case FailureNotify (FailedPlist(l+1)) {
    If ( $v_{il}$  is a physical node) Switchover & stop();
    For (each path  $P_{kl} \in \text{FailedPlist}_{(l+1)}$ ){
      If ( $P_{kl}$  is an Aggregate path){
        For (each  $P_{jl} \in P_{kl}$ )
          Send (FailureNotify, Parent_BB( $P_{jl}, v_{s1}$ ),  $P_{jl}, \text{list}$ )
        }
      Else
        Send (FailureNotify, Parent_BB( $P_{kl}, v_{s1}$ ),  $P_{jl}, \text{list}$ );
    }
  } /*end case*/
}

```

Figure 2. The SPRP algorithm

SPRP uses aggregate paths to reduce the size of the signaling storm. SPRP raises *one* notification message for each aggregate path instead of a notification message per each individual path. The message contains the *pid* of the aggregate path rather than *pids* of individual paths.

A scenario for SPRP with signaling aggregation is as follows: as node v_{jl} detects a network failure, it sends a *Failure* message up the hierarchy to its parent BBs. $\text{Parent_BB}_k(v_{jl})$ applies the SPRP algorithm with an extra step: $\text{parent_BB}_k(v_{jl})$ composes a new list of affected paths replacing individual component paths with their aggregate path. If an individual path is not to be aggregated with other paths, it is added to the list as an individual path. Then, $\text{Parent_BB}_k(v_{jl})$ forwards the *Failure* message to $\text{Parent_BB}_{[k+1]}(v_{il})$. This is repeated up the hierarchy until *RootBB* has been notified of the failure. *RootBB* sends *FailureNotify* messages to its child BBs that have these ingress routers within their views.

A BB_{ik} that receives a *FailureNotify* message has a view of the ingress routers of the failed path (aggregate or individual) in the message. In case of an aggregate path, BB_{ik} deaggregates it into its children paths. A *FailureNotify* message is sent per each child path down the hierarchy. In the case of an individual route, only one *FailureNotify* message is sent down the hierarchy. This is repeated down the hierarchy until all the ingress routers have been notified of the failure.

For example, Figure 3 shows 4 paths following the thick lines at level 1. The ingress and egress nodes of the three paths are as follows: 4.3.4 and 2.2.1 for path #1, 4.3.2 and 2.2.1 for path #2, 4.2.1 and 2.2.1 for path #3, and 4 2.1.5 and 1.4.1 for path #4. As the physical link connecting node 2.1.2 and node 2.3.2 fails, node 2.1.2 sends a *Failure* message (#1, in Fig. 3) to BB 2.1. BB 2.1 realizes that the ingress for path #4 is within its view. So, it sends a *FailureNotify* message (# 2) to the ingress of that path (node 2.1.5) and removes its entry from the list. BB 2.1 also realizes that paths #1 and #2 have the same ingress router at level 1. So, it replaces the entries for paths #1 and #2 with a single entry for their aggregate path (which we call $AggP_{12}$). Then, it sends a *Failure* message (#3) to BB 2 with the new list of affected paths. BB 2 finds it has no view of ingress routers of the paths in the list. BB 2 realizes that $AggP_{12}$ and path #3 have the same ingress router at level 2, so it removes replaces their *pid*'s with the *pid* of their aggregate path ($AggP_{13}$). Then, it sends a *Failure* message (#4) to BB *. BB * has a view of the ingress routers of the aggregate path. It sends a *FailureNotify* message (#5) to BB 4 that views the ingress nodes of $AggP_{13}$. BB 4 decomposes the aggregate path into its component paths (which are $AggP_{12}$ and path #3). It sends *FailureNotify* messages to BB 4.2 about the failure in path #3 and to BB 4.3 about the failure in $AggP_{12}$. BB 4.2 sends a *FailureNotify* message to node 4.2.1 about the failure in path #3. BB 4.3 notifies the ingress routers of the different component paths of $AggP_{12}$. That is, it sends a *FailureNotify* messages to nodes 4.3.4 and 4.3.2. This completes the failure notification process.

3. RELATED WORK

Network recovery has two basic models [11]: 1) *restoration* where a recovery path is established on demand as the network nodes are notified with the fault; and 2) *protection switching* where the recovery path is established prior to the occurrence of the fault. The scope of the recovery process may span a single network link, a segment (a set of links) of a path or an end-to-end path.

MPLS path recovery involves the following processes [13]: working and recovery path setup, fault detection and notification, switchover from the working path to the recovery path, fault repair [8] and a switchback from the recovery path to the working path.

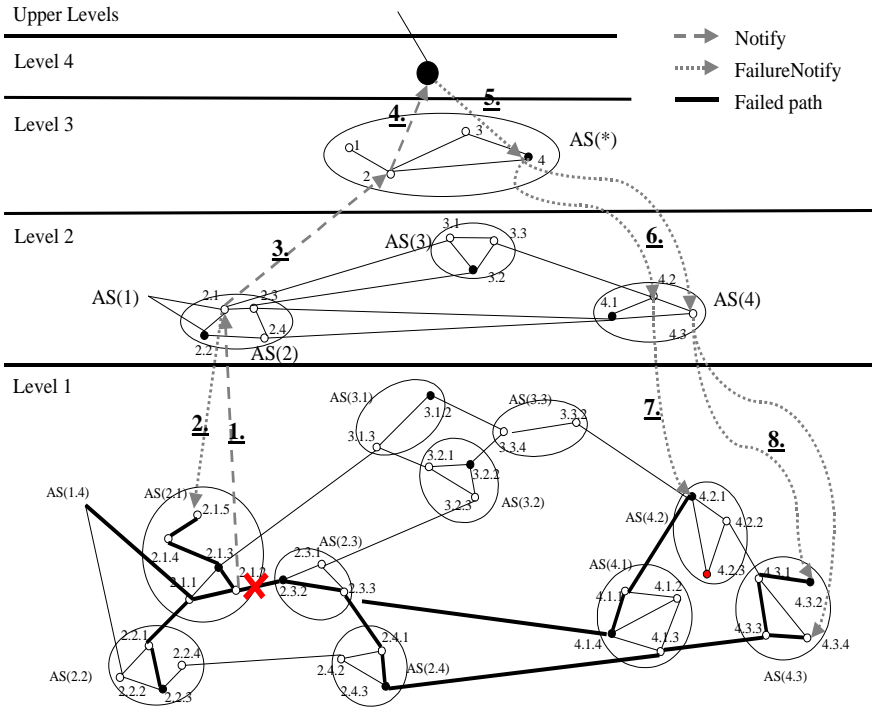


Figure 3. Hierarchical Service Recovery Signaling

In [1], a signaling protocol for fast restoration in optical networks is proposed. The protocol may be used for span and end-to-end restoration. The end-to-end restoration mechanism is as follows: A *failure indication* message is sent by the master node of the failed link along the working path towards the ingress node of the path. This message is transmitted periodically every 90 ms until a *Failure acknowledgment* message is received from the ingress node. Then, the ingress node sends a *switchover request message* towards the egress router of the path along the recovery path. This message is transmitted periodically every 90 ms until a *switchover response message* is received from the egress node. The authors define the data structures maintained at different nodes, and the general format for the used messages.

Existing MPLS path restoration schemes [1, 11] require the master node of the failed link to re-transmit failure notification signals periodically if they do not receive an acknowledgement from the ingress node of the MPLS path. Consequently, as an ingress node for a path receives a failure notification message, it acknowledges that through sending a *FailureAck* message. In SPRP, the *FailureAck* message travels exactly the same route as the failure notification message but in the reverse direction.

In [5], a “scalable and bandwidth efficient” path recovery mechanism for MPLS networks is proposed. The mechanism does not require acknowledgements or handshaking because it is based on the periodical transmittal of failure notification

messages until the “switching over” nodes learn of the failure. Specifically, the authors focus on fault notification for aggregated/ (merged) MPLS paths. The mechanism is scalable because it builds a point-to-multipoint *Reverse Notification Tree (RNT)* that is the exact mirror of the aggregated working paths. Due to the sending of only one signaling message along the shared segments of the RNT, the mechanism enables a reduction in the signaling overhead. We evaluate this method and compare it to SPRP in the following section.

4. EVALUATION

In this section we evaluate analytically the performance of SPRP and compare it to existing fault notification approaches. Using multicast is a price to pay for faster fault notification. This is not exclusive to SPRP (see RNT described in section 3). Then we present the results of our simulation experiments that evaluate how SPRP reduces the size of the fault notification tree compared to standard approaches [13].

The analytical comparison is based on the power-law relationship [4] that relates the total number of multicast links and the average unicast path length in terms of the number of receivers:

$$L(m) \approx \mu \cdot m^k \quad (1)$$

where $L(m)$ is the total number of multicast links in the multicast tree, μ is the average path length between any two nodes, m is the number of receivers in the tree (which represents the number of ingress routers to be notified) and $k \approx 0.8$ for a number of real life and generated topologies.

Equation (1) characterizes the reduction in the size of the signaling storm achieved by the RNT [5] approach that builds a multicast tree as compared to unicast approach [1].

SPRP hierarchically unicast fault notification signals up and down the hierarchy. SPRP with aggregation builds a number of multicast trees to perform fault notification. Each multicast tree is rooted at a BB that has a view of the master node of the failed link and an ingress router. The multi cast tree has the ingress routers of the affected paths as its leaves. Equation (1) characterizes the effect of aggregation on SPRP. Aggregating singling messages takes advantage of the savings multicast achieves as compared to the unicast case. This comes at the non-trivial cost of implementing multicast. In SPRP, μ_{SPRP} is on average equal to L , which is the number of levels in the network hierarchy.

SPRP with aggregation further reduces the size of the signaling storm with respect to RNT. This is because SPRP has much smaller μ than RNT. We can estimate μ_{RNT} as the effective diameter of the network, which gives the average number of hops between any two nodes in the network with high probability (about 80%). According to [10], the effective diameter of an N -nodes network (i.e. μ_{RNT}) is:

$$\mu_{RNT} = \left(\frac{N^2}{N + 2E} \right)^{1/H} \quad (2)$$

where H is the hop-plot exponent which is a constant value for the graph. For inter-domain data sets, the average value for H is 4.7 [10]. E is the number of edges in the graph and can be estimated by:

$$E(N) = \frac{N}{2(R+1)} \left(1 - \frac{1}{N^{R+1}} \right) \quad (3)$$

μ_{RNT} depends on network size. We choose $R = -0.80$ which is the value for many Internet topologies discussed in [10]). Substituting from (3) into (2) with the values given above, then μ_{RNT} can be roughly approximated by:

$$\mu_{RNT} \approx \left(\frac{0.2 * N}{1.2 - N^{-0.2}} \right)^{1/4.7} \quad (4)$$

For our simulation experiments, we used the OMNet++ simulator [14]. We developed simulation models for different topologies of the network and for different hierarchy structures. We considered networks of 256 and 1296 nodes and hierarchies of 3 and 5 levels. The topologies are described in Table I.

TABLE I. DIMENSIONS FOR NETWORK HIERARCHY STUDIED

Hierarchy	H ₁	H ₂	H ₃	H ₄
Network Size	256	256	1296	1296
Number of links E given by (1)	53464	53464	199517	199517
Number of levels $K+1$	3	5	3	5
Number of nodes per domain m	16	4	36	6
Number of physical domains	16	64	36	216

Path creation requests are assumed to arrive following a Poisson distribution. The source and destination of each connection is randomly chosen. 2000 requests are generated during each simulation run. At the end of each simulation run, network failures are generated on each node of the physical network that carries a specific number of paths. The number of SPRP messages caused by each of these failures is counted. Multiple runs are carried out and their results are averaged. We used the fully connected topology aggregation scheme when building the hierarchy.

Hierarchical message aggregation reduces the size of the fault notification-signaling storm. This is shown in Figures 4 and 5 that compare the average size of the storm for standard unicast approaches to its average size with SPRP. The size of the storm is characterized in terms of the number of affected paths. The Figures show the effect of the aggregation of signaling messages on the size of the signaling

storm by showing the average size of the storm with no signaling aggregation to its size with signaling aggregation.

SPRP proves to be a scalable technique for failure notification. Hierarchical signaling aggregation results in small increases in the size of the signaling storm as the network size increase. For example, the average size of the signaling storm for 300 failed paths is 1310 and 1520 messages for hierarchies H_2 and H_4 (which have the same number of levels), respectively. This is relatively a small increase in the size of the signaling storm compared with the increase from 4300 to 6970 messages with the unicast approach (for the same network sizes and number of paths).

The deeper the hierarchy is, the larger the size of the signaling storm for the same network size and for the same service workload. Analytically, a deeper hierarchy has a larger μ . A deeper hierarchy also results in a smaller domain size and consequently the ingress routers of the affected paths will be more sparingly distributed among network domains. This results in larger number of smaller multicast trees built by SPRP. This alleviates the overhead of building and maintaining them as compared with RNT.

Figures 6 and 7 show the control overhead required for SPRP operation. The average number of BBs managing a path is shown in terms of the path length. The effect of different hierarchy structures on the control overhead is also shown. The required number of BBs to manage a path is a good indication for the overhead of SPRP. It reflects the amount of management data that has to be maintained in the hierarchy as a function of the length of the corresponding path.

For the same network size at the physical level, as the hierarchy grows deeper, the domain size becomes smaller, the number of physical domains traversed by a path of a given length is likely to become larger, and consequently a larger number of managing BBs is required.

For example, the domain sizes are 16 and 4 for hierarchies H_1 and H_2 , respectively. In Fig. 6, a path of 20 nodes is managed by an average of 9.4 BBs in H_1 , and of 15.5 BBs in H_2 . The same argument applies for different networks considering hierarchies with the same depth (e.g. H_1 and H_3). To manage path of 20 nodes, H_3 requires an average of 5.6 BBs (in Fig. 7) as opposed to the 9.4 BBs required by H_1 . H_1 and H_3 correspond to the two-tier architecture of the Internet.

5. CONCLUSIONS

We propose a new Scalable end-to-end Path Recovery Protocol (SPRP) that results in faster recovery times. In SPRP, signaling messages travel vertically up and down the network hierarchy instead of horizontally along the paths. This reduces the number of signaling messages to notify the ingress router of the path. The protocol uses path management information maintained in the network hierarchy to aggregate signals so that a single notification message would be raised per the path aggregate.

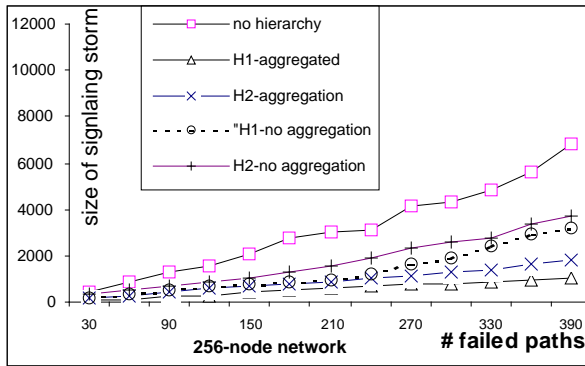


Figure 4. Average size of signaling storm in terms of path length

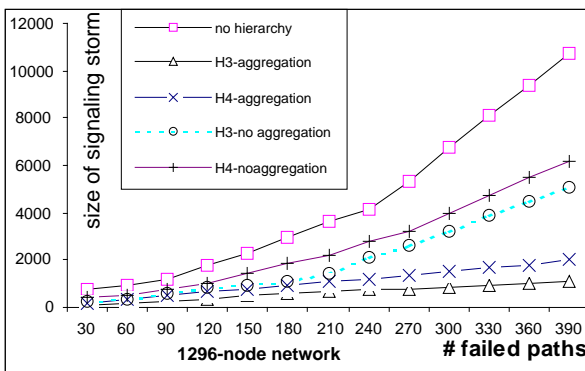


Figure 5. Average size of signaling storm in terms of path length

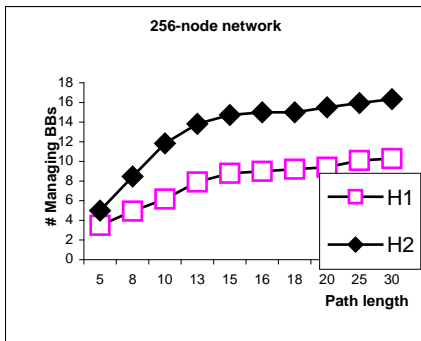


Figure 6. Average number of BBs managing a path in terms of path length

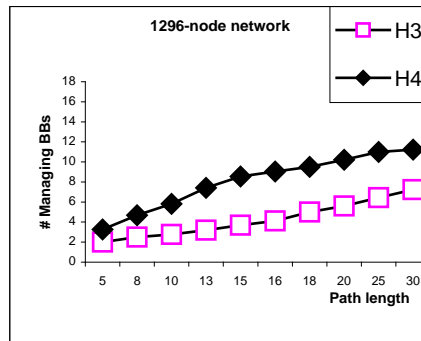


Figure 7. Average number of BBs managing a path in terms of path length

SPRP is a scalable technique for failure notification as the size of the network grows as shown by analytical analysis and simulation results. The performance of SPRP is a function of the number of levels in the network hierarchy while the performance of RNT depends on network size. The protocol provides improved performance when compared with the RNT and standard unicast approaches. Hierarchical structure affects the size of the signaling storm. The deeper the hierarchy, the larger the size of the signaling storm for the same network size and for the same service workload. These advantages come at the cost of increased control overhead required in the network hierarchy.

REFERENCES

- [1] D. Brungard et al, "Generalized MPLS Recovery Functional Specification," IETF Draft, "Work In progress," August 2002.
- [2] G. Bernstein, et al, "Optical Inter Domain Routing Considerations," IETF draft, "Work In progress," Nov. 2001.
- [3] G. Li, J. Yates, R. Doverspike, and D. Wang; "Experiments in Fast Restoration using GMPLS in Optical/Electronic Networks," Optical Fiber Comm. Conf., March 2001.
- [4] J. Chuang and M. Sirbu, "Pricing Multicast Communications: A Cost-Based Approach," Telecommunication Systems 17: (3), July 2001.
- [5] K Owens et al, " A Path Protection/Restoration Mechanism for MPLS Networks," IETF draft, "Work In progress," July 2001.
- [6] M. El-Darieby and J. Rolia, "Performance Modeling of a Service Provisioning Design," in Lecture Notes in Computer Science, Vol.1890, Springer Verlag, 2000, pp. 81-92.
- [7] M. El-Darieby, D. Petriu, and J. Rolia, "Scalability Analysis of Virtual Network-Based Service Provisioning," In IFIP/IEEE Integrated network Management, Seattle, USA 2001.
- [8] M. El-Darieby, and A. Biezschaad, "Intelligent Mobile Agents: Towards Network Fault Automation" In IFIP/IEEE Integrated network Management, Boston, USA 1999.
- [9] M. El-Darieby, D. Petriu, and J. Rolia, "HDRP: A Hierarchical Distributed Routing Protocol," Technical Report-07-03, Systems and Computer Engineering Dept., Carleton University, Submitted for publication.
- [10] M. Faloutsos, P. Faloutsos and C. Faloutsos, "On Power-Law Relationships of the Internet Topology," In ACM SIGCOMM'99, Boston, MA, USA, 1999.
- [11] F. Baker, C. Iturralde, F. Faucheur and B. Davie, "Aggregation of RSVP for IPv4 and IPv6 reservations," Internet Draft "Work In progress," <draft-ietf-issll-rsvp-aggr-02.txt>.
- [12] R. Cohen, R. Emek and E. Felstine, "Framework for Multicast in Hierarchical Networks," In IEEE *INFOCOM*, Tel-Aviv, March 2000.
- [13] V. Sharma et al, "Framework for MPLS-based Recovery," Internet Draft, "Work In progress," July 2001.
- [14] The OMNET++ Simulator: <http://www.hit.bme.hu/phd/vargaa/omnetpp.htm>
- [15] G. Ash, "Traffic Engineering & QoS Methods for IP-, ATM- & TDM based Multiservice Networks," Internet Draft, "Work In progress," <draft-ash-te-qos-routing-01.txt>.
- [16] S. Salsano et al, "Inter-domain QoS Signaling: the BGRP Plus Architecture," Internet Draft, "Work In progress," <draft-salsano-bgrpp-arch-00.txt>.