Fast Recovery Approaches from Failures in IP Networks

T.Balachander[1,] P.Rajasekar[2] and M.Arulprakash[3]

***Abstract*—** The Internet has evolved into a platform with applications having strict demands on robustness and availability, like trading systems, online games, telephony, and video conferencing. For these applications, even short service disruptions caused by routing convergence can lead to intolerable performance degradations. This paper develops novel mechanisms for recovering from failures in IP networks with proactive backup path calculations and IP-tunneling. The primary scheme provides resilience for up to two link failures along a path. The highlight of the developed routing approach is that a node re-routes a packet around the failed link without the knowledge of the second link failure. The proposed technique requires three protection addresses for every node, in addition to the normal address. Associated with every protection address of a node is a protection graph. Each link connected to the node is removed in at least one of the protection graphs and every protection graph is guaranteed to be two-edge connected. The network recovers from the first failure by tunneling the packet to the next-hop node using one of the protection addresses of the next-hop node; and the packet is routed over the protection graph corresponding to that protection address. We prove that it is sufficient to provide up to three protection addresses per node to tolerate any arbitrary two link failures in a three-edge connected graph. Our proposed model provides recovery from dual link or single node failures over several network topology. An extension to the basic scheme provides recovery from single node failures in the network. It involves identification of the failed node in the packet path and then routing the packet to the destination along an alternate path not containing the failed node.

***Keywords*--** IP fast reroute, failure recovery, multiple link failures, node failure, network protection, key, independent trees**.**

I. INTRODUCTION

Traditional routing in IP networks involves computing a forwarding link for each destination, referred to as the primary (preferred) forwarding link. When a packet is received at a node, it is forwarded along the primary forwarding link corresponding to the destination address in the packet. To recover from the failure of the forwarding link, a node must re-route the packet over a different link, referred to as the backup forwarding link. The backup forwarding link at different nodes in the network must be chosen in a consistent manner to avoid looping.

T.Balachander is Assistant Professor, Department of Computer Science and Engineering, SRM University, Chennai - 603 203, India. Email: balachander.t@ktr.srmuniv.ac.in

P.Rajasekar is Assistant Professor, Department of Information Technology SRM University, Chennai - 603 203, India. Email: rajasekar.p@ktr.srmuniv.ac.in

M.Arulprakash is Assistant Professor, Department of Computer Science and Engineering, SRM University, Chennai - 603 203, India. Email: arulprakash.m@ktr.srmuniv.ac.in

Measurement studies indicate that about 30% of unplanned failures affect more than one link. Half of these affect links that are not connected to the same node. It is sometimes possible to identify Shared Risk Link Groups (SRLG) of links that are likely to fail simultaneously, by a careful mapping of components that share the same underlying fiber infrastructure. This might, however, be a complex and difficult task, since the dependencies in the underlying transport network might not be fully known, and can change over time.

A recovery method that can recover from two independent and simultaneous link failures will greatly reduce the need for such a mapping.

The goal of this paper is to enhance the robustness of the network to - a) dual link failures; and b) single node failures. To this end, we develop techniques that combine the positive aspects of the various single-link and node failure recovery techniques. In the developed approach, every node is assigned up to four addresses – one normal address and up to three protection addresses. The network recovers from the first failure using IP-in-IP tunneling (RFC2003 [10]) with one of the "protection addresses" of the next node in the path. Packets destined to the protection address of a node are routed over a protection graph where the failed link is not present.

Every protection graph is guaranteed to be two-edge connected by construction, hence is guaranteed to tolerate another link failure. We develop an elegant technique to compute the protection graphs at a node such that each link connected to the node is removed in at least one of the protection graphs, and every protection graph is two-edge connected.

The highlight of our approach is that we prove that every node requires at most three protection graphs, hence three protection addresses. When a tunneled packet encounters multiple link failures connected to the same next-hop node, we conclude that the next-hop node has failed. The packet is then forwarded to the original destination from the last good node in the protection graph along a path which does not contain the failed node.

II. RELATED WORK - FAST RECOVERY FROM SINGLE LINK FAILURES

2.1 Equal cost multi-path (ECMP)

(ECMP) [11] is a technique employed in IP networks today that allows multiple forwarding links for a specific destination as long as the cost of the paths through each forwarding link is the same as the shortest path cost to the destination. A more general approach is to allow the use of any downstream path [12] as a forwarding link. The presence of several downstream

paths can be exploited to give fast recovery from failures, as specified in [13]. With this approach, every packet, whether forwarded along the primary or backup forwarding link, will be forwarded to a node with a lower cost to the destination than the current node. This monotonicity property of the multiple paths keeps the routing algorithm simple, where a packet need not be identified whether it was a re-routed packet or not. In addition, the failure of a link need not be advertised in the network. However, the obvious drawback of such a method is that it cannot offer recovery from all single link or node failures, since it is not always possible to find alternate downstream paths for all destinations.

2.2 Multi-Protocol Label Switching (MPLS)

In [14], Iselt et al. establish virtual links in the network using Multi-Protocol Label Switching (MPLS) with a specific cost that would enable every node in the network to have equal-cost multi-paths to a destination node.Narvaez et al.[15] develop a method that relies on multi-hop repair paths to route around a failed link. This approach requires message exchanges among nodes within a local neighborhood around the failed link, in order to avoid looping and achieve local re- convergence of routing table.

Reichert et al. [17] propose a routing scheme named O2, where all routers have two or more valid loop-free next hops to any destination. However, the technique does not guarantee single link failure recovery in any two-edge connected network.

The IETF community is also showing interest in a solution for fast rerouting in IP networks. Shand and Bryant[18] present a framework for IP fast reroute, where they mention three candidate solutions for IP fast reroute that all have gained considerable attention. These are multiple routing configurations (MRC) [3], failure insensitive routing (FIR) [4], [19], and tunneling using Not-via addresses (Not-via) [2]. The common feature of all these approaches is that they employ multiple routing tables. However, they differ in the mechanisms employed to identify which routing table to use for an incoming packet.

2.3 Multiple Routing Configurations (MRC)

The MRC approach divides the network into multiple auxiliary graphs, such that each link is removed in at least one of the auxiliary graphs and each auxiliary graph is connected. Every node maintains one routing table entry corresponding to each auxiliary graph for every destination. If the primary forwarding link fails, a packet is routed over the auxiliary graph where the primary link was removed. The routing table to use (or equivalently the auxiliary graph over which the packet is forwarded) is carried in the header of every packet. The drawback of this approach is that it does not bound the number of auxiliary graphs employed. For example, a ring network with n nodes would require n auxiliary graphs, thus requiring dlog ne bits to specify the routing table to use. The MRC approach has been extended to handle multiple failures [20]. The auxiliary graphs are constructed such that for any combination of two component failures, there exists an auxiliary graph that does not use the two failed components. With this approach, the number of auxiliary graphs needed increases. In [20], medium-sized networks require as much as 12 auxiliary graphs to guarantee recovery from two link failures.

III. Network Model

Consider a network represented as a graph G(N , L), where N denotes the set of nodes and L denotes the set of links in the network. The links are assumed to be bidirectional. An edge x → y represents a directed link from node x to node y. A link failure is assumed to affect the edges on both directions. The link failures are known only to nodes connected to the failed link and the information is not propagated to the rest of the network. We assume that the network employs a link- state protocol (such as OSPF or IS-IS) by which every node is aware of the network topology. We make no assumptions about symmetric link weights in the networks.

A network must be three-edge connected in order to be resilient to two arbitrary link failures, irrespective of the recovery strategy employed. We assume that the given network is three-edge-connected. Verification of three-edge connectivity and determination of three-vertex connected components have been extensively studied [27], [28], [29], and the complexities of verification and decomposition algorithms are O(|L|). A network must be two-vertex connected in order to be resilient to any single node failure.

IV. Proposed Methodology To Fast Recovery From Failures

4.1 Recovery From Dual Link Failures Using Tunneling

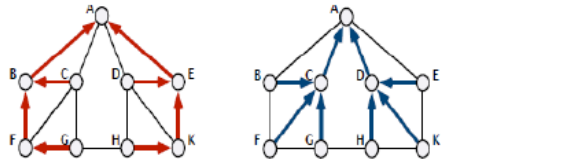
In order to recover from arbitrary dual link failures, we assign up to four addresses per node – one normal address and up to three protection addresses. These addresses are used to identify the endpoints of the tunnels carrying recovery traffic around the protected link. The default (normal) address of a node $u \in N$ is denoted by u_0 . This acts as the primary address for the routing protocol. In addition, there are three backup addresses denoted by u_1 , u_2 , and u_3 which are employed whenever a link failure is encountered.

The links connected to node u are divided into three protection groups, denoted by L_{u1} , L_{u2} , and L_{u3} .

Node u is associated with three protection (auxiliary) graphs – G_{ui} (N , L\$L_{ui}$), where i = 1, 2, 3. The protection graph G_{ui} is obtained by removing the links in L_{ui} from the original graph G. The highlight of our approach is that each of the three protection graphs is two-edge connected by construction. We prove in Section IV-A that such a construction is guaranteed in any three-edge connected graph. Let $S_{ug} = \{v \mid u–v \in L_{ug}\}$ denote those nodes that are connected to u through a link that belongs to L_{ug} . Nodes in S_{ug} are the only nodes that will initiate tunneling of packets (to protection address u_g) upon failure of the link connecting node u.

A. Colored trees

An efficient approach to route packets along link- or node-disjoint paths in packet-switched networks with minimum routing table overhead and lookup time is to employ colored trees (CTs) [25], [26]. In this approach, two trees, namely red and blue, are constructed rooted at a destination such that the paths from any node to the destination on the two trees are link- or node-disjoint. Figure 1 shows an example network with red and blue trees rooted at node A. It is necessary and sufficient for a network to be two-edge (vertex) connected to compute colored trees such that the paths from a node to the root on the two trees are link-disjoint (node-disjoint).



(a) Red Tree rooted at A (b) Blue tree rooted at A

Fig. 1. Example network with colored trees rooted at node A.

The colored trees approach provides two forwarding links (red and blue) at every node for a destination, thus falls into the class of techniques that employ multiple routing tables. While it resembles MRC, the colored tree approach employs only two routing tables, thus requiring one overhead bit to be carried in the packet header. This overhead bit may be eliminated by computing the forwarding link based on input link. The packets received on a red (blue) link may be forwarded to the red (blue) neighbors. The packets received over links that are not on either tree may be forwarded on any of the outgoing links. The colored trees may also be employed for tunneling, where if the preferred forwarding link fails, the packet is tunneled to the next node. If the failed forwarding link is present on the red (blue) tree, then the packet is tunneled using blue (red) tree. If the failed forwarding link is not present on any of the trees, the packet may be tunneled to the next node on either tree.

However, with colored trees, the packet may be redirected directly to the destination, while still employing any desired routing algorithm when there are no failures. Under this approach, every packet carries a one-bit overhead that specifies if the packet has seen a failure or not. If this bit is set to 0, the packet is forwarded based on the destination address only. If this bit is set to 1, the packet is routed based on the destination address and incoming link.

4.1.1 Computing Protection Graphs

The decomposition of the graph into three protection graphs for every node u ∈ G is achieved by temporarily removing node u and obtaining the connected components in the re- sultant network. If the network is two-vertex connected, then removal of any one node will keep the remaining network connected. However, if the network is only one-vertex-connected, removal of node u may split the network into multiple connected components. In such a scenario, we consider every connected component individually. We assign the links from a connected component to node u into different groups based on further decomposition and compute the protection groups. We then combine the corresponding protection groups obtained from multiple connected components.

4.1.2 Key Generation

The unique key is generated to each node. Each node communicates using the unique key. At the time of data transmission,these keys are used to protect the data. Authorized user only can communicate using this key.The MD5 algorithm is used for integrity and data protection.

4.1.3 Packet Forwarding

By default, all packets are forwarded towards the destination prefix decided by the destination address in the packet header. Traffic is routed on graph G towards the selected egress node. A packet destined to d is transmitted with address d_0 , and is routed on graph G. The network is assumed to employ any desired routing algorithm under no failure scenario.

Every node is assumed to route the packet based on the destination address and the interface (incoming link) over which the packet was received. For every destination-interface pair, the routing table at a node specifies the interface (outgoing link) over which the packet has to be forwarded. Note that if the network employs shortest path routing, the outgoing link for default destination address for a node would be the same, irrespective of the incoming interface.

Consider a packet destined to egress node d that has forwarding link as x–y at node x. Let link x–y belong to group g (∈ {1, 2, 3}) at node y. In the event that link x–y is not available, node x stacks a new header to the packet with destination address as y_g . The packet is now routed on the protection graph G_{yg} , where it may encounter at most one additional link failure. Given that the protection graph is two-edge connected, we employ the colored tree technique to route the packet. Under the colored tree approach, in every protection graph G_{yg} , we construct two trees, namely red and blue, rooted at y_g such that the path from every node to y_g are link-disjoint. Observe that an incoming link in the protection graph may either be red or blue. Therefore, the tree on which a packet is routed is identified based on the incoming link. Thus, it is not necessary to explicitly specify the tree in the packet header. Without loss of generality assume that the packet is routed on the red tree. Given that the packet experiences a failure in the protection graph, it is simply forwarded along the blue tree. Once the packet reaches the desired node y_g , the top header is removed, and the packet continues on its original path in G. It is worth noting that the neighbors of y whose link to y are removed in G_{yg} are the only nodes that will transmit packets to the protection address y_g.

4.1.4 Forwarding Tree Selection in a Protection Graph

Consider a packet, destined to egress node d, that encounters a failure at node x, where the default forwarding link is x–y.

Node x stacks a new header to the packet with the destination address as y_g . The packet may now be transferred either along

the red or blue tree. There are two approaches to select the default tree over which the packet is routed.

The first approach is referred to as the red tree first (RTF), where every packet is forwarded along the red tree. Upon failure of a red forwarding link in the protection graph, the packet will be forwarded along the blue tree. When a blue forwarding link fails, the packet is simply dropped as it indicates that the packet has already experienced two link failures. Note that if the RTF approach is employed.

The second approach is referred to as the shortest tree first (STF), where a packet is forwarded along that tree which provides the shortest path to the root of the tree. As the packets are first forwarded on the shortest tree,the packets experience lower delays under single link failure scenarios.

While the red tree may offer the shortest path for node x in the protection graph G_{yg} , the blue tree may offer the shortest path for another node x_0 in the same protection graph, where x, $x_0 \in S_{yg}$. A packet that is forwarded on the red (blue) tree will be re-routed to the blue (red) tree upon a red (blue) forwarding link failure. The limitation of this approach is that it may result in perennial looping if more than two links fail in the network. Unlike the RTF approach, where a packet to be forwarded on the blue link implies that it has already experienced two link failures, the STF approach does not provide any implicit indication on the number of failures experienced by the packet.

4. 2 Fast Recovery From Single Node Failures

In a network, the failure of a node causes the failure of all the links connected to it. For a neighbor u of a failed node v, the node failure will appear as a failure of link $u - v$. Thus, further information is required at node u to correctly identify the node failure. As node failures are rare compared to link failures, we develop a mechanism to recover from single node failures by enhancing the dual-link failure recovery mechanism discussed thus far. We consider the first or second link failures encountered around a particular node are just link failures and the node itself is operational. We assume the failure of at least three links connected to a node is sufficient to conclude the failure of the node.

In order to identify a possible node failure, we introduce the PNF bit. Upon encountering the first link failure, a packet would have this bit set to 0. When the packet encounters the second link failure (or first link failure in the protection graph), this bit is set to 1 if the failed forwarding link is connected to the root of the tree, thus indicating that two links connected to the same node have failed. The packet will be rerouted to the other tree in the protection graph. When the packet encounters the third failure, the packet will be dropped if the third failed link is not connected to the root of the tree. If the third failed link is connected to the root of the tree (and the PNF bit is set), then we infer a node failure.

we construct up to three protection graphs and compute two colored trees in each graph to recover from dual link failures. In addition to these protection graphs, for every node d, we construct two colored trees rooted at d, referred to as R_d and B_d , such that the path from any node to the root of the tree are node-disjoint. We will employ these two trees in order to route the packet directly towards the destination prefix when a node failure is inferred. Thus, a total of four pairs of colored trees with each node as root are employed when two-link or single-node failure recovery is required.

V. Conclusion and Future Work

The paper develops two novel schemes to provide failure resilience in IP networks using IP-in-IP encapsulation based tunneling. The first scheme handles up to two link failures. The first failure is handled by routing the packet in a protection graph, where each protection graph is designed to handle another link failure. The paper develops the necessary theory to prove that the links connected to a node may be grouped such that at most three protection graphs are needed per node. All backup routes are constructed a priori using three protection addresses per node, in addition to the normal address, making the scheme scalable with the size of the network with minimal overhead. The paper uses aspects from established schemes as intermediate steps and does not impose restrictions on the routing protocol handling the normal failure-free scenario. The paper discusses two approaches, namely RTF and STF, to forward the tunneled packet in the protection graph, describing the benefit of shorter paths in STF at the cost of an extra overhead bit. The second scheme extends the first scheme so that it provides recovery from dual link failures or a single node failure. A node failure is assumed when three separate links connected to the same node are unavailable. The packet is then forwarded along a path to the destination avoiding the failed node.Through simulations, we show that the average recovery path lengths are significantly reduced with the STF approach as compared to the RTF approach.

Our solution provides fast recovery from both dual-link or single node failures along a network path, and also achieves data-protection using keys for secure communication and gives a prevention measure to data-leakage or data loss. Application to less than three-edge-connected Networks can be still employed by slightly modifying this solution could be another work as future study.

References

[1] S. Kini, S. Ramasubramanian, A. Kvalbein, and A. Hansen, “Fast recovery from dual link failures in ip networks,” in Proceedings of IEEE INFOCOM, 2009, pp. 1368–1376.

[2] M. Shand, S. Bryant, and S. Previdi, “IP fast reroute using not-via addresses,” Internet Draft, March 2010, draft-ietf-rtgwg-ipfrr-notvia-addresses-05.

[3] A. Kvalbein, A. F. Hansen, T. ˇCiˇci´c, S. Gjessing, and O. Lysne, “Fast IP network recovery using multiple routing configurations,” in IEEE INFOCOM, Apr. 2006.

[4] S. Lee, Y. Yu, S. Nelakuditi, Z.-L. Zhang, and C.-N. Chuah, “Proactive vs. reactive approaches to failure resilient routing,” in IEEE INFOCOM, Mar. 2004.

[5] S. Ramasubramanian, M. Harkara, and M. Krunz, “Linear time dis- tributed construction of colored trees for disjoint multipath routing,” Elsevier Computer Networks Journal, vol. 51, no. 10, pp. 2854–2866, July 2007.

[6] G. Schollmeier, J. Charzinski, A. Kirst¨adter, C. Reichert, K. J. Schrodi, Y. Glickman, and C. Winkler, “Improving the resilience in IP networks,” in Proceedings of HPSR, Torino, Italy, Jun. 2003, pp. 91–96.

[7] D. Ward and D. Katz, “Bidirectional forwarding detection,” draft-ietf- bfd-base-11.txt, January 2010, internet Draft, work in progress.

[8] P. Francois, C. Filsfils, J. Evans, and O. Bonaventure, "Achieving sub-second IGP convergence in large IP networks," ACM SIGCOMM Computer Communication Review, vol. 35, no. 2, pp. 35 – 44, Jul. 2005.
[9] A. Markopoulou, G. Iannaccone, S. Bhattacharyya, C.-N. Chuah, and C. Diot, "Characterization of failures in an IP backbone network," in Proceedings INFOCOM, Mar. 2004.
[10] C. Perkins, "IP encapsulation within IP," RFC 2003, Oct. 1996.
[11] Intermediate system to Intermediate system routing information exchange protocol for use in conjunction with the Protocol for provid- ing the Connectionless-mode Network Service (ISO 8473), ISO/IEC 10589:2002, ISO, November 2002.
[12] "International standard iso/iec 10589," 2002.
[13] A. Atlas and A. Zinin, "Basic specification for ip fast reroute: Loop-free alternates," RFC5286, September 2008.
[14] A. Iselt, A. Kirstadter, A. Pardigon, and T. Schwabe, "Resilient routing using ECMP and MPLS," in Proceedings of HPSR, Phoenix, Arizona, USA, Apr 2004.
[15] P. Narvaez and K. Y. Siu, "Efficient algorithms for multi-path link state routing," in Proceedings of ISCOM, 1999. [Online]. Available: http://citeseer.ist.psu.edu/narvaez99efficient.html
[16] R. Rabbat and K.-Y. Siu, "Restoration methods for traffic engineered networks for loop-free routing guarantees," in Proceedings of ICC, Helsinki, Finland, Jun. 2001.
[17] C. Reichert, Y. Glickmann, and T. Magedanz, "Two routing algorithms for failure protection in IP networks," in Proceedings of the 10th IEEE Symposium on Computers and Communications (ISCC), Jun. 2005, pp. 97–102.
[18] M. Shand and S. Bryant, "IP fast reroute framework," RFC 5714, January 2010.
[19] S. Nelakuditi et al., "Failure insensitive routing for ensuring service availability," in IWQoS'03 Lecture Notes in Computer Science 2707, Jun. 2003.
[20] A. F. Hansen, O. Lysne, T. ˇCiˇci´c, and S. Gjessing, "Fast Proactive Recovery from Concurrent Failures," in ICC 2007, June 2007.
[21] S. Hanks, T. Li, D. Farinacci, and P. Traina, "Generic router encapsulation (GRE)," RFC 1701, Oct. 1994.
[22] J. Lau, M. Townsley, and I. Goyret, "Layer 2 tunnelling protocol - version 3 (L2TPv3)," RFC 3931, Mar. 2005.
[23] A. Li, P. Francois, and X. Yang, "On improving the efficiency and manageability of notvia," in Proceedings of ACM CoNEXT, 2007.
[24] S. Bryant and M. Shand, "IPFRR in the presence of multiple fail- ures," http://tools.ietf.org/html/draft-bryant-shand-ipfrr-multi-01, Octo- ber 2008.
[25] S. Ramasubramanian, M. Harkara, and M. Krunz, "Distributed linear time construction of colored trees for disjoint multipath routing," in Proceedings of IFIP Networking, Coimbra, Portugal, May 2006, pp. 1026–1038.
[26] G. Jayavelu, S. Ramasubramanian, and O. Younis, "Maintaining colored trees for disjoint multipath routing under node failures," IEEE/ACM Transactions on Networking, vol. 17, no. 1, pp. 346–359, February 2009.
[27] J. Hopcroft and R. E. Tarjan, "Dividing a graph into triconnected components," in SIAM Journal of Computing, vol. 2, no. 3, 1973, pp. 135–158.
[28] S. M. Lane, "A structural characterization of planar combinatorial graphs," Duke Math Journal, vol. 3, no. 3, pp. 460–472, 1937.
[29] Z. Galil and G. F. Italiano, "Maintaining the 3-edge-connected components of a graph on-line," SIAM Journal of Computing, vol. 22, no. 1, pp. 11–28, 1993.