

# Relaxed Multiple Routing Configurations: IP Fast Reroute for Single and Correlated Failures

Tarik Čičić<sup>\*‡</sup>, Audun Fossellie Hansen<sup>\*§</sup>, Amund Kvalbein<sup>\*</sup>, Matthias Hartman<sup>†</sup>,  
Rüdiger Martin<sup>†</sup>, Michael Menth<sup>†</sup>, Stein Gjessing<sup>\*‡</sup> and Olav Lysne<sup>\*</sup>

<sup>\*</sup> Simula Research Laboratory, Martin Linges vei 17, N-1331 Fornebu, Norway  
Email: {tarikc, audunh, amundk, steing, olavly}@simula.no

<sup>†</sup> University of Würzburg, Institute of Computer Science, Am Hubland, D-97074 Würzburg, Germany  
Email: {hartmann,martin,menth}@informatik.uni-wuerzburg.de

<sup>‡</sup> Department of Informatics, University of Oslo, Gaustadelléen 23, N-0371 Oslo, Norway

<sup>§</sup> Telenor Research and Innovation, Snarøyveien 30, N-1331 Fornebu, Norway

**Abstract**—Multi-topology routing is an increasingly popular IP network management concept that allows transport of different traffic types over disjoint network paths. The concept is of particular interest for implementation of IP fast reroute (IP FRR). First, it can support guaranteed, instantaneous recovery from any link or node failure as well as from many combined failures. Second, different failures result in routing over different network topologies, which gives better control of the traffic distribution in the networks after a failure. Multiple Routing Configurations (MRC) [1] is the state-of-the-art IP FRR scheme based on multi-topology routing today.

In this paper we present a new, enhanced IP FRR scheme which we call “relaxed MRC” (rMRC). rMRC simplifies the topology construction and increases the routing flexibility in each topology. According to our experimental evaluation, rMRC has several benefits compared to MRC. The number of backup topologies required to provide protection against the same set of failures is reduced, hence reducing state in routers. In addition, the backup paths are shorter, and the link utilization is significantly better. The paper also presents how rMRC can provide recovery from multiple correlated failures without compromising much on the number of backup topologies required and the path lengths.

**Index Terms**—IP fast reroute, multi-topology routing, network protection, network utilization, correlated failures, shared risk groups.

## I. INTRODUCTION

When there is a connectivity failure or a topological change in a network, traditional intra-domain routing protocols like OSPF or IS-IS respond by triggering a network-wide re-convergence. Information about the failure is broadcast in the network, and all routers in the domain independently calculate a new valid routing table upon receiving the notification. This is a time-consuming process that typically involves a period of instability and invalid routing in the network [2], [3]. The time-scale of this re-convergence process has been significantly reduced with modern routers [4]. However, this is still not acceptable for emerging time-critical Internet applications with stringent demands on network availability.

A number of mechanisms for faster failure handling have been proposed for both MPLS [5] and connectionless IP networks [1], [6]–[9]. These mechanisms compute alternative

routes in advance, which are immediately ready for use by the node that detects the failure. Such mechanisms have two very attractive properties. First, they respond quickly to a failure and prevent packet loss by allowing packet forwarding to continue on alternate routes while the routing protocol converges on the new topology. Second, they allow routers to delay the sending of a failure notification for a period of time while relying on the available repair path. This way, short-lived failures can be handled without triggering a global re-convergence. A large percentage of experienced network failures are short-lived [10], and handling such failures locally can improve network stability.

Multi-topology (MT) routing is a powerful traffic engineering and network management concept based on introducing multiple logical topologies in the network. Each logical topology is used to route a special class of the network traffic, identifiable from the packet header. For example, multicast or high-priority DiffServ traffic could be routed separately from the remaining traffic. The IP community has recently shown a strong interest in this concept, and the standardization process is being completed [11], [12].

Multi-topology routing is well suited for implementation of fast local recovery in connectionless IP networks [13]. Multiple Routing Configurations (MRC, [1]) represents the state-of-the-art fast reroute scheme based on MT routing. In MRC, traffic headed to a failed network component is tagged and forwarded by the detecting node in an alternate logical topology that does not use the failed component for routing. These “backup” topologies are created using a set of rules that, in biconnected networks, guarantee that there is such a logical topology for each failure and each network destination. In general, for a node detecting a component failure (i.e., loss of signal to one of its neighbors) it is hard to know whether the neighbor node or the connecting link is broken. MRC guarantees recovery from any single link or node failure, without requiring explicit knowledge about the underlying failure.

Existing proactive recovery schemes are designed to handle single failures only. However, several studies show that mul-

multiple simultaneous failures are not uncommon in practice, and that in most cases there is a correlation between the elements that fail together [10], [14]. Such failures are often said to belong to a common Shared Risk Group (SRG). Examples of common failure correlations include IP links sharing the same conduct, fiber, network card, or router. The cause of correlated failures can be natural disasters, terror attacks, power outages or construction workers accidentally breaking a fiber conduct [15].

Multi-topology routing allows independent setting of link weights in the logical topologies. This implies that traffic can be routed according to a different set of link weights during the recovery phase than during normal operations, allowing independent traffic engineering for each topology. A careful tuning of the link weights in the logical backup topologies can improve the load distribution in the network [16]. In MRC, link-failure protection requires every link to be excluded from routing in one of the backup topologies. Such links are said to be “isolated” in this topology, and their weight is set to infinity. A typical backup topology has many isolated links, which constrains the routing of recovered traffic.

In this paper we propose an improved fast reroute scheme based on multi-topology routing. We call the scheme “relaxed MRC” (rMRC). rMRC does not require that all links are isolated, and hence it is simpler and arguably easier to deploy and manage. In addition, fewer isolated links in a topology result in less constrained routing. We analyze key performance metrics of the new scheme and show a notable improvement compared to the state-of-the-art. We also present how rMRC can be extended to provide recovery from multiple simultaneous failures within a shared risk group. We demonstrate that the shared risk group protection does not decrease the performance in terms of number of backup topologies and path lengths considerably compared to the single failure rMRC.

This paper is organized as follows. In Sec. II we provide additional background and related work in network load optimizations and IP FRR. In Sec. III we present our relaxed recovery scheme. We provide a qualitative comparison with MRC in Sec. IV. The performance evaluation of rMRC including load distribution for the single fault situation is presented in Sec. V. In Sec. VI we describe and evaluate an extension to rMRC for handling shared risk group failures. We conclude the article in Sec. VII.

## II. BACKGROUND

IP fast reroute should provide full protection against all single link and node failures in the network. The IETF IP FRR framework [6] distinguishes between different recovery schemes for use in IP networks. The simplest scheme is fast failure protection using Loop-Free Alternates (LFA, [7]). In case of failure, LFA redirects traffic to neighboring nodes which have a path to the destination that does not include the failed component. The simplest case is when there are one or more equal cost alternative paths from the detecting node to the destination (Equal-Cost Multi-Path forwarding, ECMP). ECMP can be used both for load balancing and failure recovery.

LFA is simple to implement and already available, but does not guarantee 100% failure recovery for single link and node failures [17]. Therefore, LFA is rather a short-term solution and IP-FRR schemes with 100% failure coverage are required for the future. Several such schemes are known, including Not-Via [8], FIR [9], and MRC [1]. Any of these can be used as a complement to LFA, or alone.

An important challenge when designing fast reroute schemes is to minimize the adverse consequences on the backup paths and traffic distribution [18]. Network operators often carefully configure their networks to avoid overloaded links. The shifting of traffic to alternate links after a failure can lead to congestion and packet loss in parts of the network [19]. This can be the case both while the fast-reroute is active and, in case of permanent failure, after the re-convergence process. Appropriate link weight settings can mitigate the packet loss in all phases.

To avoid congestion while traffic is being recovered by rMRC, we use load balancing techniques developed in a traffic engineering context. The first traffic engineering mechanisms for connectionless IP networks were based on finding a set of link weights that distributes the load on the available links in the network given an estimate of the traffic demands [20], [21]. Later, more robust methods have been developed that also take into account variations in the traffic demands [22] or link failures [23], [24]. In MT-based recovery schemes, load can be distributed during the recovery phase as well [16].

In [28], the authors propose to use a concept similar to MT routing to achieve increased path diversity and increased robustness. They present a method to randomly generate alternate topologies, and a way for the source node to assign traffic to each of them. Their method does not provide recovery from all failures, and the recovery time is longer than in other FRR schemes due to signaling delay.

Most work on correlated failures has focused on shared risk link group recovery in optical WDM networks and networks running GMPLS or MPLS (e.g. [25]). Another related research topic concentrates on tools for correlated failure diagnosis (e.g. [26]). A method for fast recovery from any two concurrent (not correlated) failures is described in [27]. The scalability of this scheme is probably too poor for practical applications, and it is not covering shared risk groups of size larger than two.

## III. RELAXED MRC

The core idea of MRC is to have different *backup* routing topologies in which certain nodes and links are not used for the routing of recovered traffic. If these network elements fail, traffic can still be forwarded in the corresponding backup topologies. To that end, the topology ID must be encoded in the packet header. The node detecting that the next hop for a packet is not reachable in its current topology just needs to switch the traffic to another still working routing topology.

MRC as presented in [1] and [16] creates a set of backup topologies so that each link and node in the network is isolated in one of them. Relaxed MRC (rMRC) removes the requirement that each link must be isolated in a backup topology, and uses the isolated links only when strictly necessary.

## A. Definitions

We consider a network consisting of a set of nodes  $V$  and links  $E$  defined by the network topology graph  $G = (V, E)$ . In IP networks, unidirectional links (edges)  $e = (u, v)$  are assigned administrative link weights  $w(e)$ . Traffic is carried over the paths with the least cumulative link weights to its destination. With MT routing, logical topologies are defined by assigning various link weights  $w_i(e)$  to all links  $e \in E$  such that each of them can have a different routing.

Let  $w_{\max}$  be the maximal normal link weight in the network, i.e.,  $1 \leq w(e) \leq w_{\max}, \forall e \in E$ . We define  $w_r = |E| \cdot w_{\max}$  as the *restricted link weight*. The purpose of restricted links is to influence where shortest paths are laid in backup topologies—any acyclic path consisting of edges  $e : w(e) \leq w_r$  in the network will have a cumulative weight lower than the weight of a single restricted link. Finally, we refer to a link with infinite weight as an *isolated link*.

An rMRC network topology  $T_i$  comprises the graph  $G$  and a weight function  $w_i : E \rightarrow \{1, 2, \dots, w_{\max}, w_r, \infty\}$ . rMRC distinguishes between the default topology  $T_0$  and backup topologies  $T_i, i > 0$ . In  $T_0$  no links are restricted, i.e.,  $w_0(e) \leq w_{\max}, \forall e \in E$ .

For the protection against all single node failures, each node  $v \in V$  must not be used as a transit node in at least one routing topology  $T_i$ . Then, we say that  $v$  is an *isolated node* in topology  $T_i$ . Formally, a node  $v \in V$  is isolated in topology  $T_i$  if and only if all its adjacent links have a weight of at least  $w_r$ . In rMRC, only links directly connecting two isolated nodes must be assigned an infinite weight, i.e., must be isolated themselves. This is necessary to prevent traffic (i.e., shortest paths) from ever going through an isolated node.

Isolated nodes must be placed in backup topologies so that the following invariant holds:

*Invariant 1:* All nodes must be connected by a path consisting only of non-isolated links and nodes.

This ensures that all nodes can reach each other in all backup topologies without transiting an isolated node.

Fig. 1a and 1b give an example of a typical backup topology for MRC (a) and rMRC (b) where nodes 3, 4 and 5 are isolated. The example illustrates that rMRC (b) requires less isolated links (bold-line links) than MRC (a). This leaves more available links for routing during failures.

## B. Backup Topology Construction

Backup topologies may be constructed using different methods. We present a simple heuristic algorithm that attempts to isolate approximately equally many nodes in each of a given number of backup topologies.

The algorithm initially creates backup topologies as clones of the default topology  $(G, w_0)$ , without any isolated nodes. In this algorithm, node queue  $Q_n$  is created as an arbitrary sequence (line 5).

The algorithm tries to isolate nodes as they are pulled out of the node queue (line 8). The backup topologies are selected in round-robin fashion (line 15). Function  $\text{connected}(T_i, u)$  tests if node  $u$  can be isolated in topology  $T_i$  without violating Invariant 1 (Sec. III-A). For example, if node 1 was the next

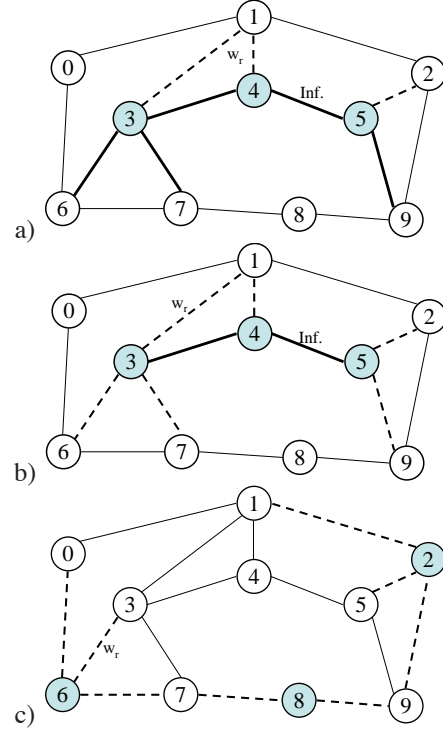


Fig. 1. Sample backup configuration in MRC (a) and relaxed MRC (b, c). In figures (a) and (b), nodes 3, 4 and 5 are isolated. In (a), links 3-4, 3-6, 3-7, 4-5 and 5-9 are isolated and do not carry any traffic in MRC. In (b), only links 3-4 and 4-5 are isolated. Fig. (c) shows another rMRC backup topology, where nodes 2, 6 and 8 are isolated.

---

### Algorithm 1: Basic rMRC backup topology generator.

---

**Input:** Desired number of backup topologies  $n$ , graph  $G$   
**Output:** Backup topologies  $T_1, \dots, T_n$ , if successful

```

1 for  $i \in \{1 \dots n\}$  do
2    $T_i \leftarrow (G, w_0)$  // Backup topology  $i$ 
3    $S_i \leftarrow \emptyset$  // Isolated nodes in  $T_i$ 
4 end
5  $Q_n \leftarrow V(G)$  // Node queue
6  $i \leftarrow 1$ 
7 while  $Q_n \neq \emptyset$  do
8    $u \leftarrow \text{first}(Q_n)$ 
9    $j \leftarrow i$ 
10  repeat
11    if  $\text{connected}(T_j, u)$  then
12      isolate( $u, T_j$ )
13       $S_j \leftarrow S_j \cup \{u\}$ 
14    else
15       $j \leftarrow (j \bmod n) + 1$ 
16  until  $u \in S_j$  or  $i = j$ 
17  // If  $i = j$ , all backup topologies tried
18  if  $u \notin S_i$  then
19    Abort execution
20   $i \leftarrow (i \bmod n) + 1$ 
21 end

```

---

---

**Procedure**  $\text{isolate}(u, T_j)$ 


---

```

1 forall  $(u, v) \in E(G)$  do
2   if  $w_j(u, v) = w_r$  then
3      $w_j(u, v) \leftarrow \infty$ 
4   else
5      $w_j(u, v) \leftarrow w_r$ 
6 end

```

---

node to be tested in the backup topology depicted in Fig. 1b, the test would return **false**. This is because node 4 then lacks a path of non-isolated nodes to all other nodes. If node 0 was the next node, the test would return **true**. In that case, procedure  $\text{isolate}(u, T_j)$  is called. This procedure alters the weights of the links adjacent to  $u$ . If a neighbor of  $u$  was already isolated, the link between them will get weight  $\infty$  (line 3 in the procedure  $\text{isolate}$ ). Else, the link will get the weight  $w_r$  (line 5). If  $\text{connected}(T_i, u)$  returns **false**, all other backup topologies are tried in sequence.

In some cases the specified number of backup topologies is too low for the input graph  $G$ , and the algorithm will have to abort and exit without success (line 18).

### C. Forwarding Information Computation

The generated topologies are input to a process that calculates backup next hops. This process is similar to the forwarding information calculation in the default (failure-free) topology. It also finds the shortest paths to all destinations, but differs in the way how it performs the the last hop calculation.

Normally, both link and node failures are protected by routing traffic around the next hop node. However, when the last link used to reach the destination (or egress router in the network) fails, only the next hop link should be avoided and not the entire node. This is known as the *last hop problem* [19] and has to be handled separately.

Contrary to MRC, rMRC does not explicitly isolate all links to solve the last hop problem. Instead, rMRC computes the shortest path *without* the failed link in the backup topology where the detecting node itself is isolated. Using the backup topology where the detecting node is isolated ensures that the traffic cannot loop back to the detecting node but still enables the rMRC forwarding to reach the destination node.

### D. Forwarding

In multi-topology routing, all packets carry a topology identifier to associate them with the topology they are routed in. The topology ID is encoded in the packet header. All nodes have to maintain routing information for all topologies to be able to forward data in any of them. This basic forwarding is shown in steps 1 and 2 in the procedure in Fig. 2.

Failure-detecting nodes have a special role. They have to change the topology the packet is routed in from the default (normal) topology to the appropriate backup topology. Topology change can occur only once; if the packet is already tagged by a backup topology, the packet is dropped to avoid looping in case of multiple failures (step 3). If the failure is

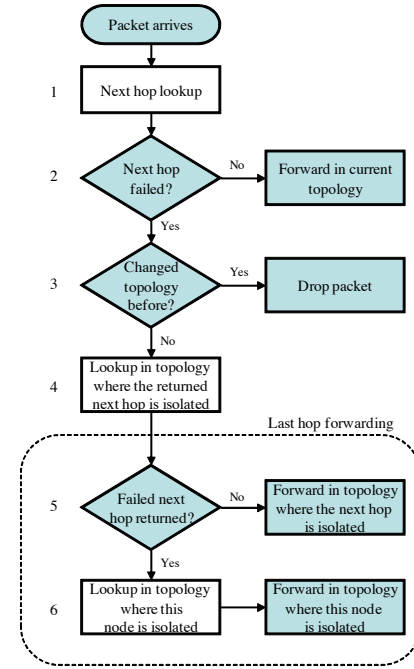


Fig. 2. rMRC forwarding procedure.

detected toward an intermediate node (not last hop) in the forwarding path, the appropriate backup topology is the one that has the failed node isolated. Then, regardless of whether there is a link or node failure that has been detected the packet is rerouted around the failure to the destination.

If the failure is detected on the last hop in the forwarding path, the same next hop can be returned in step 4, and step 5 evaluates to “Yes”. However, since the forwarding information is computed without the link between the detecting node and the destination, it is safe to forward the packet in the backup topology where the detecting node is isolated (step 6).

We illustrate the rMRC last hop handling using Fig. 1. Assume node 6 detects a failure toward the last hop node 3. The rMRC topology where node 3 is isolated is shown in Fig. 1b. Here, path 6-3 has still the lowest cost but must not be selected since link 6-3 (or node 3) has failed. Instead, rMRC uses the topology where the detecting node 6 is isolated (Fig. 1c). In this backup topology, any neighbor of node 6 may be used to reach the destination. It is however favorable to pre-calculate which neighbor is closest to the destination and store this as the next hop in this topology. In our example in Fig. 1c, node 7 is closest to the destination and selected as the backup next hop to destination 3 in this backup topology. Since node 6 itself is isolated in this topology, packets do not loop back to the failed link 6-3.

## IV. QUALITATIVE COMPARISON

The core benefit of rMRC compared to MRC as defined in [1] is that rMRC does not require every link to be isolated in one of the backup topologies. MRC requires isolated links for solving the last hop problem. rMRC solves the last hop problem by sending the recovered packets to another neighbor. In rMRC, isolated links are only used to prevent traffic from

transiting isolated nodes. As a result, more links can be used for forwarding and the flexibility on how to route traffic is improved.

### A. Calculating Next Hops

MRC explicitly isolates all links in the backup topology where one of their adjacent nodes is isolated. The last hop case is solved by using the topology where the last link is isolated. The next hop can then be determined by a shortest path calculation in this topology.

Since rMRC does not isolate all links, the last hop case is solved by an alternative next hop calculation. With rMRC, any neighbor of the detecting node except the destination can be used as next hop, and the most obvious choice would be to use the one with the shortest path to the destination. This next hop is found by calculating the shortest path from each neighbor to the destination, with the restriction of not using the last hop link. This calculation is similar to what is being standardized for Not-via in IETF [6], and it does not affect the performance noticeably.

### B. Algorithm Complexity

The complexity of the presented rMRC algorithm for topology creation is, similar to MRC, determined by the loops and the complexity of the connectivity testing. An algorithm that tests whether a network is connected is bound to worst case  $\mathcal{O}(|V| + |E|)$ . The number of runs of the inner loop in Alg. 1 is bound by the maximum node degree  $\Delta$ . In worst case, we must run through all  $n$  configurations to find a configuration where a node can be isolated. The worst case running time for the complete algorithm is then bound by  $\mathcal{O}(n\Delta|V||E|)$ .

While the worst-case running time of the rMRC algorithm is unchanged compared to MRC, the algorithm itself is more intuitive and easier to implement.

### C. Effect of Isolated Links Elimination

Our key expectation is that the reduced number of isolated links will result in rMRC yielding shorter backup path lengths than MRC. Since fewer links are isolated, rMRC gives a denser topology for routing recovered traffic, and thus shorter recovery paths.

An example can be seen in Fig. 1. If node 3 is the egress node, backup traffic can only be routed over link 1-3 when using MRC (a). For rMRC, links 1-3, 6-3 and 7-3 can be used to reach node 3 (Fig. 1b). Having more options on how to reach this isolated egress will give a higher probability of finding a shorter backup path.

A related hypothesis is that the lower backup path lengths of the relaxed method lead also to an improved load distribution in the network in failure cases. If for example a path of length  $n$  between a source and a destination is reduced to  $n - 1$ , there are one fewer router and link to carry the traffic between these nodes, decreasing the total network load. This reduction in resource consumption may be used to improve the overall network performance.

## V. EVALUATION

In this section we evaluate rMRC using several performance metrics commonly used in evaluation of IP FRR schemes, including state requirements, backup path lengths, and load distribution. We compare rMRC with MRC to pinpoint the performance differences.

Fault-tolerant multi-topology routing requires the routers to store additional information about the backup topologies. The amount of state required in the routers is related to the number of such backup topologies. An excessive amount of this state may affect router operation and therefore generating only few backup topologies is the goal. We measure how many backup topologies are needed by MRC and rMRC to guarantee fault tolerance.

When the failure occurs, IP FRR will immediately start forwarding data traffic over backup paths. As backup paths already carry their normal (non-rerouted) traffic, this increases the chance of congestion even in networks that are well provisioned for failure-free cases.

The backup path lengths are correlated with the total network load and the end-to-end delay. The backup path lengths are independent of the traffic matrix, yielding more robust results. Therefore we evaluate both the backup path lengths and how well rMRC can optimize the load distribution and avoid congestion in the case of failure.

Evaluation of, e.g., state requirements of a fast reroute scheme requires experimenting with a large number of diverse network topologies, while load distribution optimizations are computationally expensive. We therefore used two evaluation methods, one for the state requirements and backup path lengths, and one for the load distribution evaluation.

### A. State Requirements and Backup Path Lengths

1) *Method*: We used synthetic network topologies based on the Waxman model [29], created using the BRITE generator [30], as well as some publicly available real topologies. Families of 100 networks of size 16–64 nodes and two or three times as many links are tested. We use unit link weights, so that the path lengths equal the hop count.

Algorithms for MRC (as in [1]) and rMRC (Alg. 1, Sec. III) are used to create backup topologies with the minimum number of topologies. For any given topology the algorithms are run with  $n = 2, 3, \dots$ , until the first successful execution. The results of these runs are presented in the state requirements analysis.

Based on the created topologies, we measure the backup path lengths (hop count) achieved by our schemes after a node failure. The backup path lengths are calculated for each source-destination pair in the network and for each node failure on the path between them.

2) *State Requirements*: Relaxed backup topologies defined and described in Sec. III do not isolate all links. Therefore, there is more flexibility in rMRC than in MRC to decrease the number of backup topologies. Figure 1 illustrates this difference. Assume that the process of isolating nodes (and links for MRC) should continue from the topologies presented for MRC (Fig. 1a) and rMRC (Fig. 1b). For MRC, nodes 1,

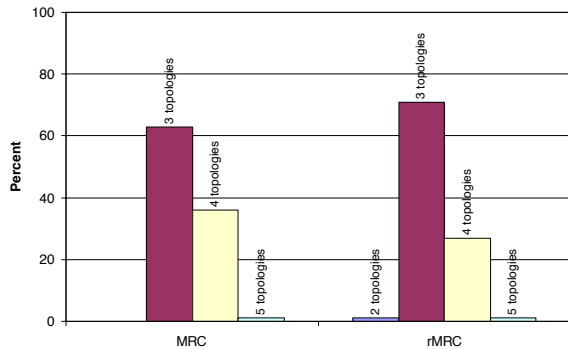


Fig. 3. Number of backup topologies for MRC and rMRC calculated for 100 random Waxman topologies with 32 nodes and 64 links.

TABLE I  
NUMBER OF BACKUP TOPOLOGIES FOR SOME REAL TOPOLOGIES

Network	Nodes	Links	MRC	rMRC
Geant	19	30	5	4
Cost239	11	26	3	2
DFN	13	38	2	2

2 and 7 are not candidates to be isolated, because isolating any of them would disconnect one or more of nodes 4, 5 and 3 from the rest of the topology. For rMRC, it is only node 1 that must be excluded from the list of candidates, since its isolation would lead to disconnection of node 4.

Figure 3 and Tab. I show the number of backup topologies generated with the MRC and rMRC. We observe that the increased flexibility with rMRC can decrease the number of topologies needed.

3) *Path Lengths*: Since routing in a backup topology is restricted, rMRC results in backup paths that are equally long or longer than the optimal paths in the re-converged network.

Figure 4 shows the distribution of path lengths for normal failure-free routing, routing re-converged to the failure topology, and MRC and rMRC during a node failure in networks with 32 nodes and 64 links (other network sizes show the same tendency). We see that the performance of less constrained rMRC is slightly better than the performance of MRC and closer to the optimal full IP re-convergence. It is important to remember that IP FRR gives that performance immediately after the failure is detected, while the optimal scheme does not yield this until the re-convergence is completed.

Mean path lengths for different network sizes are shown in Fig. 5. As the size of the networks increases the path lengths also increase. Still, rMRC shows a better performance compared to MRC. In Fig. 6, we show how the number of backup topologies influences the backup path lengths for MRC and rMRC in topologies with 32 nodes and 64 links. Increasing the number of backup topologies to a few more than the minimum achievable improves the performance. However, the improvement diminishes if the number of backup topologies reaches a certain level.

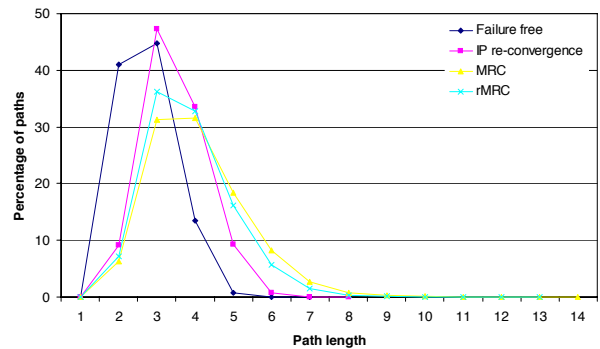


Fig. 4. Path length distribution.

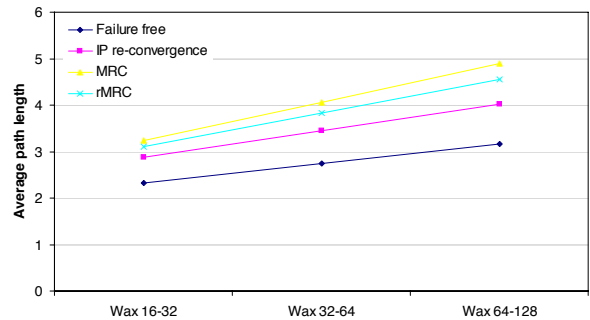


Fig. 5. Mean path length as function of the network size.

## B. Network Load Distribution

1) *Method*: When the rMRC fast reroute is active in the network, the load distribution during recovery depends on three factors:

- 1) The link weight assignment used in the default (normal) topology,
- 2) The structure of the backup topologies, (i.e., which links and nodes are isolated in each of them),
- 3) The link weight assignments used in the normal links ( $w(e) \leq w_{\max}$ ) of the backup topologies.

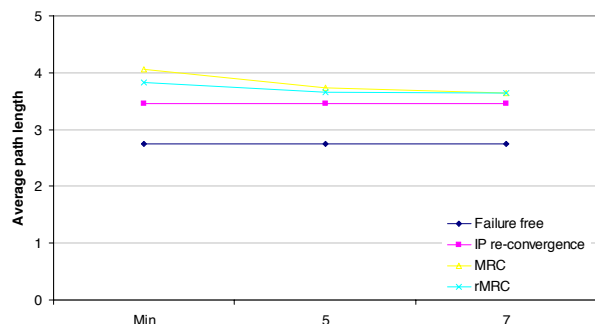


Fig. 6. Mean path length as function of the number of backup topologies. All networks have 32 nodes and 64 links. “Min” means the minimal number of backup topologies achieved by our algorithm for the given input topology; typically 3 or 4.

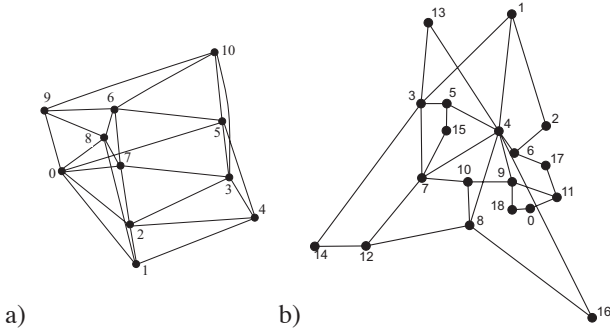


Fig. 7. Cost239 topology (a) and Geant topology (b) used in the evaluation.

The link weights in the default topology (1) are important since all non-affected traffic is distributed according to them, while backup topologies are used only for the traffic affected by the failure. The backup topology structure (2) dictates which links are used in the recovery paths for each failure. The backup topology link weight assignments (3) determine which among the available backup paths are actually used.

The load distribution in the network (1) and (3) can be improved using IP link weight optimization techniques. The optimization process modifies the link weights trying to reduce the utilization of the link with the highest traffic load subject to the given source-destination traffic matrix.

There are different approaches regarding the question whether IP link weights should be optimized primarily for the load distribution in the failure-free case or for the fast reroute phase (in which case some of the failure-free performance may be lost). This mainly depends on the network operators' management policies. Fault-tolerant multi-topology routing allows link weight settings in the backup topologies independent from the default topology. This allows us to optimize the failure free phase *and* improve the fast reroute load balancing.

We use ECMP forwarding to further improve the load distribution. Since this implies the existence of this mechanism in the routers, we also use ECMP for fast reroute in cases an alternate equal-cost path is available after failure. (r)MRC is then only used when there is no such equal cost alternative.

*a) Considered Network Topologies:* For the computationally demanding load distribution optimizations, we used several realistic network topologies, and present results for Geant and Cost239 networks. Geant is an illustrative existing network, while Cost239 is a good representative of a projected topology that shows how the future networks should look like to properly support resilience mechanisms and fault management. This is reflected among other things in the network connectivity, Geant being relatively sparse compared to Cost239 (Fig. 7).

*b) Optimization Framework:* Network operators often plan and configure their network based on an estimate of the traffic demands from each ingress node to each egress node. Clearly, the knowledge of such a demand matrix provides the opportunity to construct the backup topologies in a way that gives better load balancing and avoids congestion after a failure.

In this paper we optimize the load distribution for MRC and

rMRC using the same three-step procedure:

- 1) The link weights in the normal topology are optimized for the given demand matrix while only taking the failure free situation into account.
- 2) For “intelligent” backup topology construction, the load distribution in the failure free case is used to weight the impact of each node failure on the link load in the network. The aim is to isolate nodes that carry a large amount of transit traffic in the backup topologies with good connectivity. Thus, if such a node fails, there are many possible recovery paths, leading to a larger optimization potential. To that purpose, [16] describes a heuristic that sums up the total transit traffic through each node and isolates fewer heavy-traffic nodes, or more light-traffic nodes, per backup topology.
- 3) When the backup topologies are constructed, the link weights (for links where  $w(e) \leq w_{\max}$ ) of the backup topologies are optimized to get a good load distribution after any link or node failure.

For a clear comparison, we compare backup topologies with identical isolated node sets for MRC and rMRC. The backup topologies are constructed using the algorithm described in [1]. In rMRC, the isolated link weights are relaxed to  $w_r$  except between the isolated nodes, as described in Sec. III-A.

*c) Traffic Matrix:* To evaluate the load distribution in the network, we require the knowledge of the traffic matrix. The structure of the matrix directly influences the link weight setting given by the optimization procedure. Thus, it is necessary to know the traffic demands between all origin and destination pairs in the network. Even for real networks, this data is generally unavailable due to its confidentiality and difficulties in collecting it. We chose to synthesize the origin-destination (OD) flow data by drawing OD demand values from an exponential distribution and matching these values with the OD pairs according to the heuristic described in [31]. In short, we sort the OD pairs according to their node degree and the likelihood of one of them being used as the backup node in the case of a single link failure. Then, we match the sorted OD pair list with the sorted list of demand intensities generated using the gravity model, which is suited for this purpose [32]. The generated OD matrix is scaled so that the most loaded link in the failure-free case has 100 % utilization.

*d) Optimization Method:* The traffic distribution in a network can be measured in terms of maximum link utilization and minimized by appropriate link weight settings. We use an optimization method based on a simulated annealing-like principle [33]. In this paragraph, we formalize our optimization objectives.

We represent the link weights for topology  $T_i$  by a vector  $w_i$  with one entry for each link (edge)  $e \in E$ . Given the link weight vector  $w_0$  for the default topology  $T_0$ , we evaluate the link utilization  $\rho(e, w_0)$  on all links  $e \in E$  in the network during the failure-free case. This yields our objective function for optimization step (1) from above:

$$\text{minimize } \rho_{\max}^E(w_0) = \max_{e \in E} (\rho(e, w_0)) \quad (1)$$

The algorithm implemented by our software heuristically

searches the vector space of possible link weight vectors  $w_0$  as described in [33].

Given the backup topologies  $T_i (i = 1, \dots, n)$  with their link weights  $w_i$  and the link weight vector  $w_0$  for default topology  $T_0$ , we now can evaluate the link utilization  $\rho^{w_0}(e, s, w)$  for link  $e \in E$  in failure scenario  $s \in S$ , where  $w = (w_1, \dots, w_n)$  are the link weights vectors for the backup topologies. The set  $S$  hereby denotes the set of protected network element failure scenarios, e.g., all single link and node failures, and does not contain the failure-free case. Note that during failure scenario  $s$  the nodes adjacent to the failure send traffic over appropriate backup topologies according to MRC or rMRC. Thus,  $\rho^{w_0}(e, s, w)$  is composed of the link utilization in the individual topologies  $T_i$  where the routing follows  $w_i$ . This yields our objective function for optimization step (3) from above:

$$\text{minimize } \rho_{\max}^{w_0, E, S}(w) = \max_{e \in E, s \in S} (\rho^{w_0}(e, s, w)) \quad (2)$$

subject to the condition that the weights of restricted and isolated links may not be changed. The heuristic again searches the space of possible link weight vectors for backup topologies  $T_i$  where  $w_0$  for the default topology remains fixed.

2) *Results:* We present the load distribution for the tested networks in form of the complementary cumulative distribution function (CCDF). We show results for the failure-free load balancing, the reconverged network after a failure (but without a new link-optimization process), then for rMRC and finally for the MRC fast reroute. If, for example, a CCDF line matches values  $x = 0.5$  and  $y = 0.68$ , this means that 68% of the links have load utilization of 50% or more. The results are scaled so that the link with the highest load in the failure-free case has unit utilization 1.00. For all distributions except the failure-free case, the depicted values represent the maximal load, i.e., the load a particular link has experienced after the worst-case failure. The effect of both link and node failures is shown. Note that in these simulations, we never drop traffic due to congestion. Instead, we let the utilization of some links exceed 100%. Hence, all load values should be considered relative.

For Cost239 (Fig. 8), with the link weights optimized for the failure-free case, the maximum link utilization for re-converged routing is 1.73. Optimized rMRC has the maximum link utilization of 1.50, while MRC has 1.87 for link failures and 2.03 for node failures. Again, it is important to remember that IP FRR outperforms the re-converged routing immediately after the failure is detected—it does not need to wait for the routing process to converge.

The results indicate a significantly lower fast reroute load if rMRC is deployed rather than MRC. If we divide all links by traffic load into two equally large groups, the difference is particularly big (up to 35 %) for the high-load half, while MRC and rMRC behave similarly for the low-load half. It is interesting that this significant difference is observed despite that in some 60 % of the cases nodes select an ECMP alternate for the affected traffic, in which case rMRC or MRC recovery is not used at all.

The difference is dramatic for the Geant network for node failures (CCDF in Fig. 9b). The relative maximum link utilization for re-converged routing and the optimized rMRC is almost the same and lies around 3.42, while the optimized MRC performs poorly with a ratio of 7.76.

Analysis of these results shows that the sparse connectivity of Geant effectively hinders the optimization process. For example, if the Geant point of presence in Austria fails (represented by node 9 in Fig. 7b), a string of East-European countries (nodes 18, 0, 11, 17) is left without an important point of attachment. All traffic from these nodes passes link 6-17 that quickly becomes fully utilized and stops the optimization process. Furthermore, the congestion on 6-17 happens much sooner in MRC than rMRC. This is because rMRC only isolates link 9-10 among links adjacent to node 9, while in MRC, also 4-9 and 9-18 are isolated. When a neighbor that has node 9 as the last hop discovers the failure, it assumes a link failure and reroutes the traffic. In MRC all this traffic is routed toward the only non-isolated link 11-9, while in rMRC it can be rerouted also over links 4-9 and 18-9.

## VI. MULTIPLE CORRELATED FAILURES

High-quality network equipment is manufactured so that the expected mean time before the given component fails is very long. When failures do happen, the operator quickly replaces the failed component to restore the service. Thus, while any combination of network links and nodes may fail simultaneously, the probability of two independent, simultaneous failures is relatively low.

Many components do however share some physical or system relation, and the likelihood of their simultaneous failure is much higher. A single duct of optical fiber can carry many logical IP connections. A power supply failure may cut out a large set of colocated network nodes. We call such failures *correlated*, and they occur frequently in practice [10], [14]. Components that share some kind of failure correlation are said to comprise a *Shared Risk Group* (SRG). The cause of correlated failures can be natural disasters like flooding, terror attacks, power outages, or construction workers accidentally breaking a fiber conduct [15].

Relaxed MRC provides a greater flexibility of backup topology creation and opens the door to handle multiple correlated failures with IP FRR. The good news about correlated failures is that they are often possible to anticipate. It is, for example, often known which links share the same duct, or the same interface card on a router. The single failure recovery schemes presented in Sec. III will in some cases be able to recover the traffic from more than one failure, however, they provide no guarantees. A modified rMRC algorithm that takes into account SRGs may yield much better results under multiple simultaneous failures. In this section, we propose and evaluate such an algorithm that we denote rMRC-SRG.

### A. Types of Correlation—Shared Risk Groups

In a large network there is a vast number of combinations of potential failures. It is not scalable or required to design a recovery scheme to protect against all the combinations. We



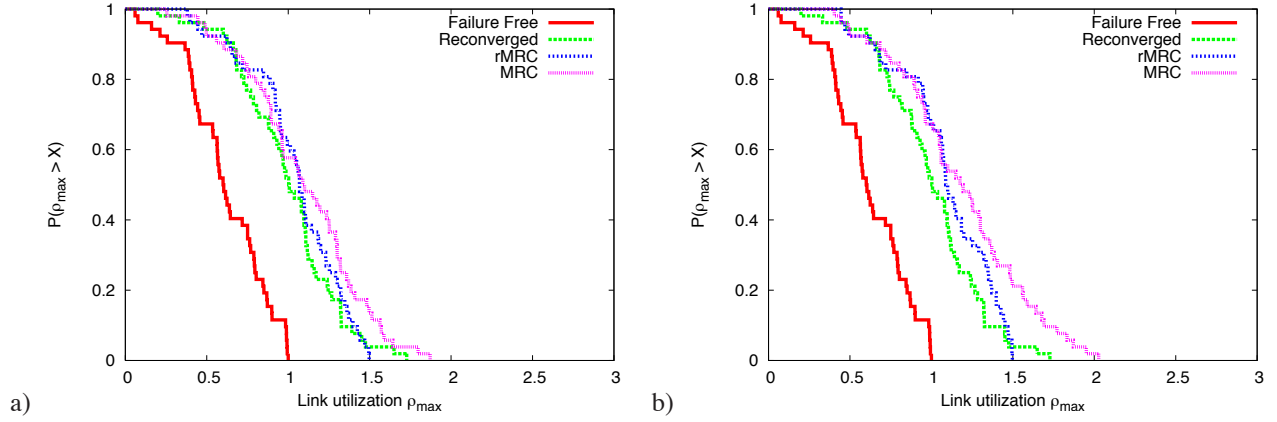


Fig. 8. Load distribution on Cost239 links, a) after link failures and b) after node failures.

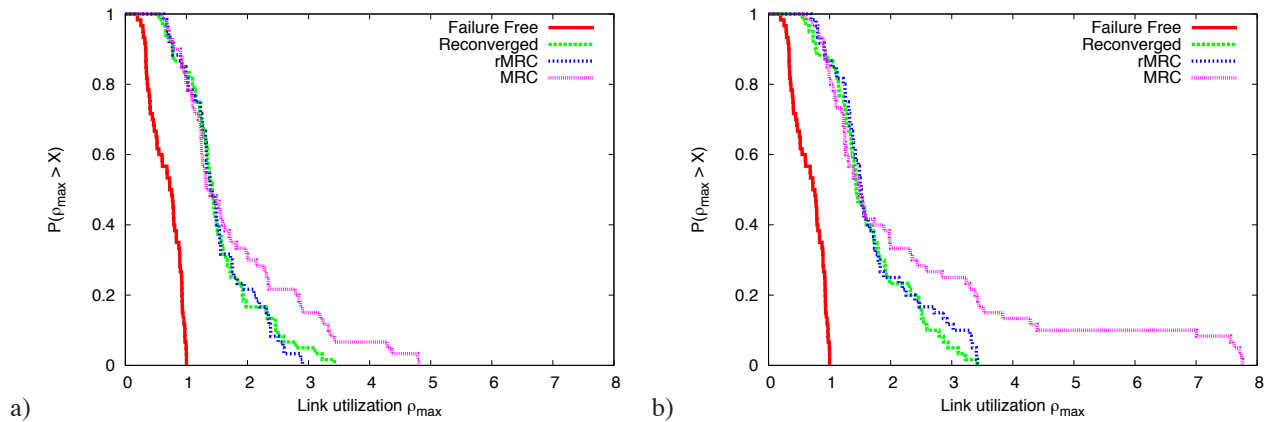


Fig. 9. Load distribution on Geant links, a) after link failures and b) after node failures.

focus on the three main causes of correlations observed in fixed IP networks: simultaneous failure of neighboring nodes, links sharing the same network interface card on a router and links sharing the same fiber or conduct [34]. In addition, all links sharing the same router can often be regarded as correlated, however this will be implicitly covered by node recovery in a method for one fault tolerance. We only address components whose failure is possible to protect. If for instance a node is an articulation point in the topology, we do not attempt to protect that node.

Figure 10a shows the basic principle of a shared risk group of *neighbor* nodes. Such nodes can fail simultaneously due to power outage or other catastrophes like floods and terror attacks. These nodes are assumed to be located in nearby physical locations and also sharing some physical resources. A point of presence (PoP) could be a typical example. Also regular maintenance like software updates and router restarts might be interpreted as neighboring nodes failing simultaneously.

Figure 10b shows the basic principle of a shared risk group of links sharing the same network interface *card* on a router. This group definition will also represent links that share the same fiber or conduct attached to the same node. Failures on the interface card, failures on a fiber component or a fiber cut

will cause these links to fail simultaneously.

Figure 10c shows the basic principle of a shared risk group of links sharing the same *conduct*. This type of correlation covers links that do not share an end point (node). A correlation where the links in a conduct also share a node is covered by the shared risk group in Fig. 10b (card).

We make the following assumptions regarding the types of correlation. For the neighbor groups (Fig. 10a), we assume that every node in the group has a connection to a node that is not in the group. Else, it is not possible to guarantee a communication path to a non-failed node in the case where not the entire group has failed. For card groups (Fig. 10b), we assume point-to-point links that can only be in one card group at each end. For conduct groups (Fig. 10c) we assume that a link can be part of more than one group as a link can share conducts with other links in different parts of a conduct stretch.

### B. Basic principles of rMRC-SRG

rMRC-SRG is designed to guarantee recovery from any single component failure or any single shared risk group failure that has been planned for in advance. To accomplish this, we build a set of logical backup topologies that make sure that each single node and each SRG has been isolated from

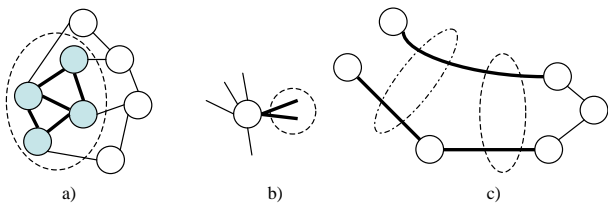


Fig. 10. a) Neighbor nodes. b) Links sharing interface card or fiber. c) Links sharing conduct.

carrying traffic in at least one of the topologies. We use restricted and isolated links and isolated nodes (described in Sec. III-A) as tools to isolate all the components. Since the SRG scheme is based on rMRC, single links do not have to be isolated explicitly. However, we use isolated links to isolate the conduct groups and the card groups.

*Neighbor group:* For each neighbor group, there must exist a backup topology where all the nodes constituting the group are isolated, i.e., the links attached to the nodes are assigned the link weight  $w_r$  as described in Sec. III-A. Some of those links might be assigned weight of infinity if they belong to, for instance, a card group (described below) that is isolated in the same backup topology. Links between isolated nodes in a neighbor group will have the weight of infinity too.

*Interface card group:* For each card group, there must exist a backup topology where the links constituting a card group will not carry any traffic. In this backup topology, these links will have the weight of infinity, i.e., they are isolated links.

*Conduct group:* For each conduct group, there must exist a backup topology where the links constituting a conduct group will not carry any traffic. In this backup topology, these links will have the weight of infinity, i.e., they are isolated links.

One backup topology can potentially isolate several nodes, links, and SRGs as long as the Invariant 1 from Sec. III-A holds. Since the shared risk SRGs are isolated using isolated nodes and links, the invariant also implies that the path will avoid all components in a group.

### C. rMRC-SRG Algorithm

We have developed an updated version of the rMRC algorithm to create backup topologies that take into account the existence of SRGs. Similarly to Alg. 1, the new algorithm operates on an arbitrary biconnected network graph. It takes three sets of correlated network components as input: the set of neighbor groups  $C_N$ , the set of interface card groups  $C_I$ , and the set of conduct groups  $C_C$ . For simplicity, the new algorithm (Alg. 3) does not take the desired number of backup topologies as input. Instead, it creates backup topologies as long as there are any non-isolated elements.

Intuitively, there should be a difference in the number of backup topologies required to isolate all SRGs and single nodes in the network graph, depending on which of the sets  $C_N$ ,  $C_I$  and  $C_C$  are attempted to isolate first. Therefore, Alg. 3 uses an ordered list of SRG sets as its queue structure  $Q$ .

(line 1). Since a node can be isolated using any combination of isolated and restricted links, Alg. 3 first isolates the more restrictive cases of interface cards (line 2) and conducts (line 3). Then, neighbor node SRGs are added, and, finally, all single nodes (line 5). The single nodes are converted in single-element sets to preserve the set-queue semantics of  $Q$ .

---

### Algorithm 3: rMRC-SRG backup topology generator.

---

**Input:** Graph  $G$ , correlated failure sets  $C_N, C_I, C_C$   
**Output:** Backup topologies  $T_1, \dots, T_n$

```

1  $Q \leftarrow \emptyset$  // Ordered list of correlated failures
2  $Q.addAll(C_I)$ 
3  $Q.addAll(C_C)$ 
4  $Q.addAll(C_N)$ 
5  $Q.addAll(\text{singletonSets}(V(G)))$ 
6  $n \leftarrow 0$ 
7 while  $Q \neq \emptyset$  do
8    $n \leftarrow n + 1$ 
9    $T_n \leftarrow (G, w_0)$  // New backup topology
10   $S \leftarrow Q.first()$ 
11  while  $S \neq \emptyset$  do
12    if  $\text{connected}(T_n, S)$  then
13      foreach  $e \in S$  do
14        case ( $e$  typeof Link)
15           $w_n(e) \leftarrow \infty$ 
16        case ( $e$  typeof Node)
17           $\text{isolate}(e, T_j)$ 
18       $Q.remove(S)$ 
19     $S \leftarrow Q.next()$ 

```

---

Algorithm 3 processes the entire queue from the first to the last element attempting to isolate all correlated sets in the current backup topology  $T_n$  (line 9). In this algorithm the semantics of function  $\text{connected}()$  are overloaded to accept a set of components instead of a single node as in Alg. 1. Function  $\text{connected}()$  returns **false** if removal of all network components (link or nodes) specified in set  $S$  disconnects the graph  $G$ , and **true** if the graph remains connected.

If the graph remains connected, the elements from the component set  $S$  are processed one-by-one and isolated depending on their type (line 14). The processed SRG sets are removed from the queue.

### D. Forwarding

When a failure is detected on the next hop, the rMRC forwarding described in Sec. III-D assumes that there is a mapping from the failed link to a backup topology that avoids that link. No such mapping exists in the SRG case, since SRGs may overlap. This means that several backup topologies might need to be checked before finding the one that protects the failed SRG.

The basic idea for rMRC-SRG forwarding is therefore to successively try the topologies from zero to  $id_{MAX}$  (i.e., the default topology ID to the highest topology ID, Fig. 11). Care must be taken to avoid looping in presence of concurrent failures that cannot be recovered. Since a node can never change the topology ID to a lower topology ID than the current ID, we can safely drop the packets when the topology ID has

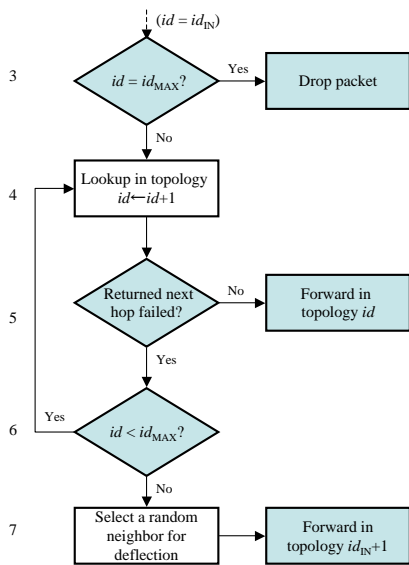


Fig. 11. Modified forwarding procedure to accommodate overlapping SRGs.

reached the maximal ID (step 3). If the incoming topology ID  $id_{IN}$  is less than  $id_{MAX}$ , steps 4 to 6 try successively backup topologies until the  $id_{MAX}$  has been reached or a valid next hop has been found. If a valid next hop has been found, the packet is forwarded to that next hop, marked with the corresponding topology ID (step 5).

If, however,  $id_{MAX}$  has been reached without having a valid next hop, there is a possibility that the failed interface is toward an egress node. In that case, all backup topologies will return the failed interface as next hop. The packets are then forwarded in the backup topology with ID one higher than the topology ID the packet had when it entered the node. As long as there are not more failures than a single link, node or SRG failure, there will exist a backup topology that brings the packets to the correct egress node without looping back to the failed component. If there are more failures than planned for, in the worst case the packets will be dropped when  $id_{MAX}$  is reached.

### E. Evaluation

When we build backup topologies that protect against correlated failures, it is expected that the number of backup topologies increases due to the number and combinations of components that must be isolated. In addition, we expect the path lengths to increase due to more isolated components in each topology. In this section we will evaluate the scalability and backup path lengths for the rMRC-SRG scheme, comparing it with the single failure scheme rMRC.

1) *Method*: We reuse the synthetic topologies from the evaluation in Sec. V and perform additional experiments on the DFN network.

We specify SRGs in four classes called “Neighbors”, “Card”, “Conduct” and “Combination” and two sizes, called A and B (Tab. II). When we evaluate the combination of SRGs, one of the other SRG classes is selected randomly with equal probability. In the evaluations, we vary the number of SRGs

TABLE II  
SHARED RISK GROUP SIZES, UNIFORMLY RANDOMLY SELECTED UNLESS STATED OTHERWISE.

	Size A	Size B
Neighbors	2 or 3	2-5
Card	2 or 4	2 (25%) or 4 (75%)
Conduct	2-5	2-8

(A or B) in the topologies so that they comply with what would be a reasonable number in the networks of interest.

2) *Number of Backup Topologies*: We have generated 100 random networks using the Waxman model with the number of nodes and links denoted as (*W-nodes-links*) in Tables III and IV. The size of the groups is either in category A or B. The number of backup topologies is denoted as (minimum, **average**, maximum) from the 100 topologies.

Table III shows how the different group types influence the number of backup topologies for rMRC-SRG. We observe that the number is the same for the three different types. For the combination, the number is a bit lower as it is easier to isolate groups of different types in a common backup topology. We also observe that the number of backup topologies increases with the number and size (category B) of groups. The last row in the table shows that the number of backup topologies increases with lower average node degree (i.e., W-32-64 with 4 SRGs of size B as opposed to W-32-96 with the same number and size of SRGs).

Table IV shows the number of backup topologies for rMRC-SRG compared to rMRC that is designed for single failures only. We have used combinations of the three group types since that will be the most relevant scenario in a real network. We observe that rMRC-SRG, which protects against correlated and single failures, requires more backup topologies compared to rMRC. In addition, more shared risk groups require more backup topologies. On the other hand, it is clear that the size of the groups has little influence on the number of backup topologies needed. This is due to the structure of the different shared risk groups. The size of a card group does not have an influence since the isolated links are attached to the same node and can easily be isolated in the same backup topology no matter whether the size is 2 or 4. Neighbor groups consist of nodes located together, and isolating 5 neighbors instead of 3 neighbors should not influence much the connectivity of the rest of the network. Isolating links does not influence so much the number of backup topologies, since isolated links are less likely to disconnect the network than isolated nodes. This is particularly true when the average node degree is high. Hence, the size of a conduct group has no significant influence on the number of backup topologies. We also see the same tendencies for the DFN network.

3) *Recovery Success Rates*: rMRC-SRG guarantees recovery from planned SRG failures. To motivate the need of this scheme, we give here an evaluation of the success rates of the single failure scheme rMRC during correlated failures. For the evaluation of correlated failures, we used combination of the three group types with the group sizes B. We measured the percentage of source-destination pairs that still can reach each other after having experienced at least one of the failures.

TABLE III  
NUMBER OF BACKUP TOPOLOGIES WITH rMRC-SRG AND DIFFERENT GROUP TYPES

Network	#groups	Size	Neighbors	Card	Conduct	Combination
DFN (13-38)	4	A	3	3	3	3
DFN (13-38)	4	B	3	3	3	3
W-32-96	4	A	3, 3, 3	3, 3, 3	3, 3, 3	2, 2.27, 3
W-32-96	4	B	3, 3, 3	3, 3, 3	3, 3, 3	2, 2.33, 3
W-32-96	8	A	4, 4, 4	4, 4, 4	4, 4, 4	2, 2.73, 3
W-32-96	8	B	4, 4, 4	4, 4, 4	4, 4, 4	2, 2.97, 3
W-32-64	4	B	3, 3.18, 4	3, 3.17, 4	3, 3.17, 4	3, 3.02, 4

TABLE IV  
NUMBER OF BACKUP TOPOLOGIES FOR DIFFERENT RECOVERY APPROACHES

Network	#groups	Size	rMRC-SRG	rMRC
DFN (13-38)	8	A	3	2
DFN (13-38)	8	B	3	
W-32-96	24	A	4, 4.04, 5	2, 2.3, 3
W-32-96	24	B	4, 4.07, 5	
W-128-384	24	A	4, 4, 4	2, 2.77, 3
W-128-384	24	B	4, 4, 4	
W-128-384	96	A	6, 6, 6	
W-128-384	96	B	6, 6.02, 7	

TABLE V  
RECOVERY SUCCESS RATES OF GROUP FAILURES WITH rMRC

	DFN (13-38) (g=8)	W-32-64 (g=24)	W-32-96 (g=24)	W-64-128 (g=48)
rMRC	88%	75.7%	77.3%	84.3%

For each synthetic network type, we have used five randomly generated topologies, and the figures given are the average of those five.

Table V shows that rMRC does not provide sufficient recovery guarantees during correlated failures. The number of groups is denoted as "g=x". We have observed the same tendencies for networks of different sizes and node degrees. rMRC-SRG provides 100 % protection in all networks.

How do the presented schemes perform in face of uncorrelated failures? We have generated random simultaneous failures of 2 and 3 nodes and counted the cases where rMRC and rMRC-SRG successfully recover the connectivity (only failure combinations where the network remains connected are counted). Table VI shows the results from a random Waxman topology with 32 nodes and 96 links.

We observe that rMRC-SRG gives higher success rate than the single failure scheme. The advantage is particularly large for 3 failures. The good performance of rMRC-SRG can be explained by three functional properties. First, rMRC-SRG can try all topologies before dropping packets. Second, it requires more topologies than rMRC, and hence have more topologies to choose from. Third, rMRC-SRG isolates more components in each topology to handle the correlations and their combinations.

4) *Backup Path Lengths*: We have calculated the difference in path lengths between rMRC and rMRC-SRG in a scenario with one node failure in the network. There are two reasons for studying the one-failure case. First, the proposed recovery scheme (rMRC-SRG) isolates several components in each backup topology, and hence the routing flexibility is restricted

TABLE VI  
RECOVERY SUCCESS RATES OF UNCORRELATED FAILURES

#Node failures	rMRC	rMRC-SRG
2	93.7%	99.7%
3	87.7%	99.1%

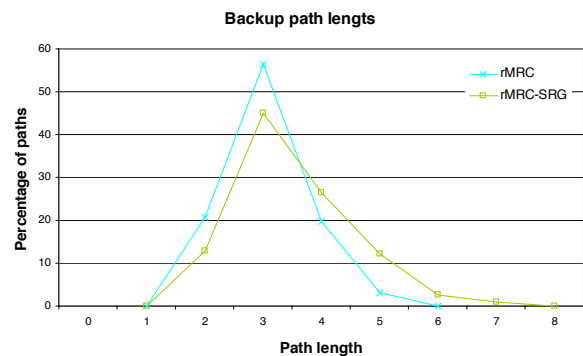


Fig. 12. Path length distribution.

even if there is only one failure in the network. Second, the one-failure case is the most dominant case in practice [10].

Figure 12 gives the results from a random Waxman topology with 32 nodes and 96 links. It shows the percentage of paths that have a particular length. rMRC-SRG gives longer backup paths than the single failure scheme rMRC. The main reason for this is the fact that more components and also localized components are isolated in the same topology. This influences the routing flexibility and the detour that the traffic must take to reach the destination.

## VII. CONCLUSION

In this paper we have proposed relaxed Multiple Routing Configurations (rMRC) for IP fast reroute. It is a simplification and enhancement of conventional MRC in the sense that the requirements for the backup topologies are relaxed. We have explained the basic operation, the backup topology creation, and the link weight optimization that are applicable to MRC and rMRC. Using these algorithms, we compared the performance of the new rMRC to the one of MRC, normal IP re-convergence, and failure-free IP routing.

The results showed that the relaxed requirements of the rMRC have several benefits. The presented algorithm can guarantee link and node fault tolerance with fewer backup topologies than MRC. Furthermore, rMRC increases the connectivity of the backup topologies, so that the length of

the backup paths is shortened and the link utilization in failure cases is lower due to improved load distribution. Our evaluation clearly indicates that rMRC is the superior multi-topology routing based approach for IP fast reroute today.

We have presented and evaluated an rMRC variant that protects shared risk groups in addition to single nodes and links. We have described a modified topology generation algorithm and a new forwarding scheme. The evaluation has shown that this multi-failure support does still keep the number of backup topologies and path lengths within acceptable bounds while guaranteeing recovery from the specified SRG failures.

Our tests indicate that sparser networks may not always be able to improve the load distribution using link weight optimizations and current backup topology algorithms. As future work, we will explore more advanced mechanisms for backup topology creation. We believe this in conjunction with link weight optimizations will further improve the load distribution.

## REFERENCES

- [1] A. Kvalbein, A. F. Hansen, T. Čičić, S. Gjessing, and O. Lysne, "Fast IP network recovery using multiple routing configurations," in *Proceedings of IEEE INFOCOM*, Apr. 2006.
- [2] A. Basu and J. G. Riecke, "Stability Issues in OSPF Routing," in *Proceedings of ACM SIGCOMM*, August 2001, pp. 225–236.
- [3] D. Watson, F. Jahanian, and C. Labovitz, "Experiences with monitoring OSPF on a regional service provider network," in *Proceedings of IEEE ICDCS*, 2003, pp. 204–213.
- [4] 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, July 2005.
- [5] V. Sharma and F. Hellstrand, "Framework for multi-protocol label switching (MPLS)-based recovery," in *IETF*, RFC 3469, Feb. 2003.
- [6] M. Shand and S. Bryant, "IP Fast Reroute Framework," IETF Internet Draft (work in progress), Feb. 2008, draft-ietf-rtgwg-ipfrr-framework-08.txt.
- [7] A. Atlas and A. Zinin, "Basic specification for IP fast reroute: Loop-free alternates," Internet Draft, Feb. 2008, draft-ietf-rtgwg-ipfrr-spec-base-10.txt.
- [8] S. Bryant, M. Shand, and S. Previdi, "IP fast reroute using not-via addresses," Internet Draft, Feb. 2008, draft-bryant-shand-ipfrr-notvia-addresses-02.txt.
- [9] S. Nelakuditi, S. Lee, Y. Yu, Z.-L. Zhang, , and C.-N. Chuah, "Fast local rerouting for handling transient link failures," *IEEE/ACM Transactions on Networking*, vol. 15, no. 2, pp. 359–372, Apr. 2007.
- [10] A. Markopoulou, G. Iannaccone, S. Bhattacharyya, C.-N. Chuah, and C. Diot, "Characterization of failures in an IP backbone network," in *Proceedings IEEE INFOCOM*, Mar. 2004.
- [11] P. Psenak, S. Mirtorabi, A. Roy, L. Nguen, and P. Pillay-Esnault, "MT-OSPF: Multi topology (MT) routing in OSPF," IETF, RFC4915, June 2007.
- [12] T. Przygienda, N. Shen, and N. Sheth, "M-ISIS: Multi topology (MT) routing in IS-IS," IETF RFC 5120, Feb. 2008.
- [13] M. Menth and R. Martin, "Network resilience through multi-topology routing," in *Proceedings of the 5th International Workshop on Design of Reliable Communication Networks (DRCN)*, Oct. 2005.
- [14] G. Iannaccone, C.-N. Chuah, R. Mortier, S. Bhattacharyya, and C. Diot, "Analysis of link failures in an ip backbone," in *2nd ACM SIGCOMM Workshop on Internet Measurement*, Nov. 2002, pp. 237–242.
- [15] C. Partridge and P. Barford, *The Internet Under Crisis Conditions: Learning from September 11*. Washington, D.C.: The National Academic press, 2003.
- [16] A. Kvalbein, T. Čičić, and S. Gjessing, "Post-failure routing performance with multiple routing configurations," in *Proceedings of IEEE INFOCOM*, Apr. 2007.
- [17] M. Gjoka, V. Ram, and X. Yang, "Evaluation of IP fast reroute proposals," in *Proceedings COMSWARE*, Jan. 2007, pp. 1–8.
- [18] S. Rai, B. Mukherjee, and O. Deshpande, "IP resilience within an autonomous system: Current approaches, challenges, and future directions," *IEEE Communications Magazine*, vol. 43, no. 10, pp. 142–149, Oct. 2005.
- [19] S. Iyer, S. Bhattacharyya, N. Taft, and C. Diot, "An approach to alleviate link overload as observed on an IP backbone," in *Proceedings of IEEE INFOCOM*, Mar. 2003.
- [20] B. Fortz and M. Thorup, "Internet traffic engineering by optimizing OSPF weights," in *Proceedings of IEEE INFOCOM*, 2000, pp. 519–528.
- [21] A. Sridharan, R. Guirin, and C. Diot, "Achieving near-optimal traffic engineering solutions for current OSPF/IS-IS networks," *IEEE/ACM Transactions on Networking*, vol. 13, no. 2, pp. 234–247, April 2005.
- [22] B. Fortz and M. Thorup, "Optimizing OSPF/IS-IS weights in a changing world," *IEEE Journal on Selected Areas in Communications*, vol. 20, no. 4, pp. 756 – 767, May 2002.
- [23] —, "Robust optimization of OSPF/IS-IS weights," in *Proceedings of INOC, European Network Optimization Group (ENOG) conference*, oct 2003, pp. 225–230.
- [24] A. Sridharan and R. Guerin, "Making IGP routing robust to link failures," in *Networking*, Waterloo, Canada, 2005.
- [25] M. T. Frederick, P. Datta, and A. K. Somani, "Sub-graph routing: a generalized fault-tolerant strategy for link failures in WDM optical networks," *Comput. Networks*, vol. 50, no. 2, pp. 181–199, 2006.
- [26] S. Kandula, D. Katabi, and J.-P. Vasseur, "Shrink: a tool for failure diagnosis in ip networks," in *MineNet '05: Proceeding of the 2005 ACM SIGCOMM workshop on Mining network data*. New York, NY, USA: ACM, 2005, pp. 173–178.
- [27] A. F. Hansen, O. Lysne, T. Čičić, and S. Gjessing, "Fast proactive recovery from concurrent failures," in *Proceedings IEEE ICC*. IEEE, 2007.
- [28] M. Motiwala, N. Feamster, and S. Vempala, "Path splicing: Reliable connectivity with rapid recovery," in *Proceedings of HotNets VI*, Nov. 2007.
- [29] B. M. Waxman, "Routing of multipoint connections," *IEEE Journal on Selected Areas in Communications*, vol. 6, no. 9, pp. 1617–1622, Dec. 1988.
- [30] A. Medina, A. Lakhina, I. Matta, and J. Byers, "BRITE: An approach to universal topology generation," in *Proceedings of IEEE MASCOTS*, Aug. 2001, pp. 346–353.
- [31] A. Nucci, A. Sridharan, and N. Taft, "The problem of synthetically generating IP traffic matrices: Initial recommendations," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 35, no. 3, pp. 19–32, July 2005.
- [32] M. Roughan, "Simplifying the synthesis of internet traffic matrices," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 35, no. 5, pp. 93–96, Oct. 2005.
- [33] M. Menth, M. Hartmann, and R. Martin, "Robust IP link costs for multilayer resilience," in *In Proceedings 6th IFIP-TC6 Networking Conference*, May 2007.
- [34] R. R. Kompella, J. Yates, A. Greenberg, and A. C. Snoeren, "Ip fault localization via risk modeling," in *NSDI'05: Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation*. Berkeley, CA, USA: USENIX Association, 2005, pp. 57–70.



**Tarik Čičić** received the M.S. degree and the Ph.D. degree from the University of Oslo, Norway, in 1997 and 2002 respectively. He has been a postdoctoral fellow at Simula Research Laboratory, Norway, 2003–2007. Today he is the CTO of Media Network Services, Norway, and an associated professor at the Department of Informatics, University of Oslo. His research interests include network architectures, communication protocols modeling, network measurements, and network resilience.



**Audun Fosselie Hansen** received his M.Sc. degree from the University of Oslo in 2001. He is currently working towards his Ph.D. degree at Simula Research Laboratory, Oslo, Norway. His thesis deals with fast reroute in IP networks. He is also holding a part time position at Telenor R&I, Norway, and he has participated in several European projects. His main research interests are resilience in fixed and wireless IP networks, routing in heterogeneous wireless networks and media streaming to heterogeneous networks and mobile terminals. Hansen is member

of the IEEE.



**Stein Gjessing** received his Dr. Philos. degree in 1985, and is currently a professor of Computer Science in Department of Informatics, University of Oslo, and an adjunct researcher at Simula Research Laboratory. His original work was in the field object oriented concurrent programming. He has worked with computer interconnects (Scalable Coherent Interface (IEEE Std. 1596) and LAN/MANs (Resilient Packet Ring (IEEE Std. 802.17). His main research interests are currently within network resilience, including sensor networks, Internet-like networks and

optical networks.



**Amund Kvalbein** Amund Kvalbein received his Cand. Scient degree from the University of Oslo in 2003, and his PhD degree from the same institution in 2007. The focus of his PhD was mechanisms for fast recovery from failures in IP networks. In particular, he was central in the development of "Multiple Routing Configurations", which is a method for local, proactive recovery with traditional intra-domain routing protocols. He is now working as a Post Doc at Simula Research Laboratory in Oslo, where he leads the REPAIR research project

focusing on resilience in IP networks. From September 2007 to August 2008, he is a visiting researcher at the College of Computing at Georgia Institute of Technology, where he works with reliability and scalability issues in both intra-domain and inter-domain routing.



**Matthias Hartmann** studied computer science and mathematics at the Universities of Austin/Texas and Wuerzburg/Germany. Currently, he is pursuing his Master degree and working at Simula research center in Oslo/Norway on the analysis of IP fast reroute mechanisms.



**Olav Lysne** is Research Director at Simula Research Laboratory, and professor in computer science at Simula Research Laboratory and the University of Oslo. He received the Masters degree in 1988 and Dr. Scient. degree in 1992, both at the University of Oslo. The early research contributions of Lysne were in the field of algebraic specification and term rewriting. However, his most important scientific contributions are in interconnection networks, focusing on problems like effective routing, fault tolerance and Quality of Service. In this area he has participated

in numerous program committees, and he served as general chair of ICPP 2005. Lysne is a member of the IEEE.



**Rüdiger Martin** studied computer science and mathematics at SUNY Albany/New York and Wuerzburg/Germany from where he received his diploma degree in computer science in 2003. Since then he is a researcher at the Department of Distributed Systems at the University of Wuerzburg and pursuing his PhD. His research focus is on load balancing mechanisms, load models for capacity overprovisioning, and analysis of resilience mechanisms in communication networks. He received several IEEE best paper awards.



**Michael Menth** studied computer science and mathematics at the University of Wuerzburg/Germany and Austin/Texas. He worked at the University of Ulm/Germany and Wuerzburg and obtained his PhD in 2004. Currently, he is assistant professor and heading the research group "Next Generation Networks" at the Institute of Computer Science in Wuerzburg. His special interests are performance analysis, optimization of communication networks, resource management, resilience issues, and Future Internet. Dr. Menth holds numerous patent applica-

tions and received various scientific awards for innovative work.