Multiple Routing Configurations For Fast IP Network Recovery

[1]M-Tech II Year student, [2] Asst.Prof ,Nimra Institute Of Technology, [3]HOD,.Professor .Nimra Institute Of Science & Technology

V.Lakshman narayana[1] , Sk.Mastan,M-Tech[2] , Dr.M.Kishore kumar M-tech, Ph.D.[3]

***Abstract*—Now A Days Internet plays a major role in day to day communication,if a network gets failed the recovery is becoming a major problem. It takes a much time to re-establish the Link To assure fast recovery from link and node failures in IP networks, we present a new recovery scheme called Multiple Routing Configurations (MRC). Our proposed scheme guarantees recovery in all single failure scenarios, using a single mechanism to handle both link and node failures, and without knowing the root cause of the failure. MRC is pure connectionless, and assumes only destination based peer-to-peer forwarding. MRC is based on keeping additional routing information in the routers, and allows packet forwarding to continue on an alternative output link immediately after the detection of a failure. It can be implemented with only minor changes to existing solutions. In this paper we present MRC, and analyze its performance with respect to scalability, backup path lengths,shortest path discovery, and load distribution after a failure. We also show how an estimate of the traffic demands in the network can be used to improve the distribution of the recovered traffic, and thus reduce the chances of congestion when MRC is used.**

***Index Terms*—Availability, computer network reliability, Communication system fault tolerance, communication system routing, protection.**

I. INTRODUCTION

Inrecent years the Internet has been transformed from a special purpose network to an ubiquitous platform for a wide range of everyday communication services. The demands on Internet reliability and availability have increased accordingly. A disruption of a link in central parts of a network has the potential to affect hundreds of thousands of phone conversations or TCP connections, with obvious adverse effects.

Manuscript received December 21, 2006; revised July 20, 2007 and February 06, 2008; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor J. Yates. First published July 25, 2008; current version published April 15, 2009.

A. Kvalbein and A. F. Hansen are with Simula Research Laboratory, 1367 Lysaker, Norway (e-mail: amundk@simula.no).

T. ˇ Ciˇcic´ was with Simula Research Laboratory, 1367 Lysaker, Norway. He is now with Media Network Services, Norway, and also with the Department of Informatics, University of Oslo, 1371 Oslo, Norway.

The main idea of MRC is to use the network graph and the associated link weights to produce a small set of backup network configurations. The link weights in these backup configurations are manipulated so that for each link and node failure, and regardless of whether it is a link or node failure, the node that detects the failure can safely forward the incoming packets towards the destination on an alternate link. MRC assumes that the network uses shortest path routing and destination based hop-by-hop forwarding.

The shifting of traffic to links bypassing the failure can lead to congestion and packet loss in parts of the network [9]. This limits the time that the proactive recovery scheme can be used to forward traffic before the global routing protocol is informed about the failure, and hence reduces the chance that a transient failure can be handled without a full global routing re-convergence

The rest of this paper is organized as follows. In Section II we describe the basic concepts and functionality of MRC. We then define MRC formally and present an algorithm used to create the needed backup configurations in Section III. In Section IV, we explain how the generated configurations can be used to forward the traffic safely to its destination in case of a failure.We present performance evaluations of the proposed method in Section V. In Section VI, we discuss how we can improve the recovery traffic distribution if we have an estimate of the demands in the network. In Section VII, we discuss related work, and finally we conclude in Section VIII.

II. MRC OVERVIEW

MRC is based on building a small set of backup routing configurations, that are used to route recovered traffic on alternate paths after a failure. Our MRC approach is threefold. First, we create a set of backup configurations, so that every network component is excluded from packet forwarding in one configuration. Second, for each configuration, a standard routing algorithm like OSPF is used to calculate configuration specific shortest paths and create forwarding tables in each router, based on the configurations The use of a standard routing algorithm guarantees loop-free forwarding within one configuration.

In our approach, we construct the backup configurations so that for all links and nodes in the network, there is a configuration where that link or node is not used to forward traffic. Thus, for any single link or node failure, there will exist a configuration that will route the traffic to its destination on a path that avoids the failed element. In Section III, we formally describe MRC and how to generate configurations that protect every link and node in a network.

Using a standard shortest path calculation, each router creates a set of configuration-specific forwarding tables. For simplicity, we say that a packet is forwarded according to a

configuration, meaning that it is forwarded using the forwarding table calculated based on that configuration. In this paper we talk about building a separate forwarding table for each configuration, but we believe that more efficient solutions can be found in a practical implementation.

When a router detects that a neighbor can no longer be reached through one of its interfaces, it does not immediately inform the rest of the network about the connectivity failure. Instead, packets that would normally be forwarded over the failed interface are marked as belonging to a backup configuration,and forwarded on an alternative interface towards its destination. The selection of the correct backup configuration and thus also the backup next-hop, is detailed in Section IV.

If a failure lasts for more than a specified time interval, a normal re-convergence will be triggered. MRC does not interfere with this convergence process, or make it longer than normal. However, MRC gives continuous packet forwarding during the convergence, and hence makes it easier to use mechanisms that prevents *micro-loops* during convergence, at the cost of longer convergence times [12]. If a failure is deemed permanent, new configurations must be generated based on the altered topology.

III. GENERATING BACKUP CONFIGURATIONS

In this section, we will first detail the requirements that must be put on the backup configurations used in MRC. Then, we propose an algorithm that can be used to automatically create such configurations. The algorithm will typically be run once atthe initial start-up of the network, and each time a node or link is permanently added or removed.

Isolated links do not carry any traffic. Restricted links are used to isolate nodes from traffic forwarding. The restricted link weight ω_r must be set to a sufficiently high, finite value to achieve that. Nodes are isolated by assigning at least the restricted link weight to all their attached links. For a node to be reachable, we cannot isolate all links attached to the node in the same configuration. More than one node may be isolated in a configuration. The set of isolated nodes in C_i is denoted S_i and the set of normal (non-isolated) nodes.

Definition: A node $\mu \in N$ is *isolated* in C_i if

$$\begin{aligned} &\forall (u,v) \in A, w_i(u,v) \ge w_r \\ \wedge \quad &\exists (u,v) \in A, w_i(u,v) = w_r. \end{aligned} \tag{1}$$

With MRC, restricted and isolated links are always attached to isolated nodes as given by the following rules. For all links

(u,v) € A,

$$w_i(u,v)=w_r \Rightarrow (u\in S_i \wedge v\in \overline{S_i}) \vee (v\in S_i \wedge u\in \overline{S_i}) \tag{2}$$
$$w_i(u,v)=\infty \Rightarrow u\in S_i \vee v\in S_i. \tag{3}$$

This means that a restricted link always connects an isolated node to a non-isolated node. An isolated link either connects an isolated node to a non-isolated node, or it connects two isolated nodes. Importantly, this means that a link is always isolated in the same configuration as at least one of its attached nodes. These two rules are required by the MRC forwarding process described in Section IV in order to give correct forwarding without knowing the root cause of failure. When we talk of a backup configuration, we refer to a configuration that adheres to (2) and (3).

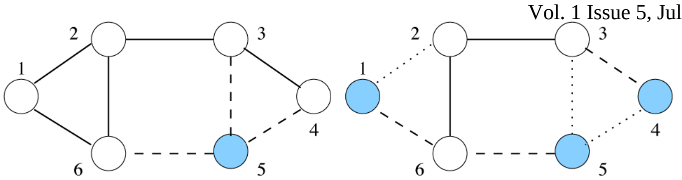


Fig. 1. Left: node 5 is isolated (shaded color) by setting a high weight on all its connected links (stapled). Only traffic to and from the isolated node will use these restricted links. Right: a configuration where nodes 1, 4 and 5, and the links 1–2, 3–5 and 4–5 are isolated (dotted).

Definition: A configuration C_i is *valid* if and only if

$$\begin{aligned} \forall u,v \in N : \mathcal{N}(p_i(u,v)) \setminus \left(\overline{S_i} \cup \{u,v\}\right) = \emptyset \\ \wedge w_i(p_i(u,v)) < \infty. \end{aligned} \tag{4}$$

We observe that all backup configurations retain a characteristic internal structure, in that all isolated nodes are directly connected to a core of nodes connected by links with normal weights:

$$w_i(u,v)=w_r \Rightarrow (u\in S_i \wedge v\in \overline{S_i}) \vee (v\in S_i \wedge u\in \overline{S_i}) \tag{5}$$

A backbone is connected if all nodes in $\overline{s}_{\mathrm{I}}$ are connected by paths containing links with normal weights only:

Definition: A backbone B_i is *connected* if and only if

$$\forall u,v \in B_i : a \in \mathcal{A}(p_i(u,v)) \Rightarrow w_i(a) \le w_{max} \tag{6}$$

An important invariant in our algorithm for creating backup configurations is that the backbone remains connected. Since all backup configurations must adhere to (2) and (3), we can show that a backup configuration with a connected backbone is equivalent to a valid backup configuration:

$$w_i(p_i(u,v)) \le 2w_r + w_i(p_i(u',v')) < \infty \tag{7}$$
$$\mathcal{N}(p_i(u,v)) \setminus \left(\overline{S_i} \cup \{u,v\}\right) = \emptyset \tag{8}$$

In backup configurations, transit traffic is constrained to the configuration backbone. A restricted link weight μ_r In backup configurations, transit traffic is constrained to the configuration backbone. A restricted link weight:

Proposition 3.2: Let be a node isolated in the valid backup Configuration C_i. Then, restricted link weight value

$$\mu_r = | A. \omega_{max} \qquad (9)$$

To guarantee recovery after any component failure, every node and every link must be isolated in one backup configuration. Let C={ C_1,…..C_n} be a set of backup configurations. We say that

Definition: A set,of backup configurations is *complete* if

$$\begin{aligned} & \forall a \in A, \exists C_i \in \mathcal{C} : w_i(a) = \infty \\ \wedge \quad & \forall u \in N, \exists C_i \in \mathcal{C} : u \in S_i. \end{aligned} \qquad (10)$$

The algorithm can be implemented either in a network management system, or in the routers. As long as all routers have the same view of the network topology, they will compute the same set of backup configurations.

Algorithm 1: Creating backup configurations.

```
for $i \in \{1 \ldots n\}$ do
    $C_i \leftarrow (G, w_0)$
    $S_i \leftarrow \emptyset$
    $B_i \leftarrow C_i$
end
$Q_n \leftarrow N$
$Q_a \leftarrow \emptyset$
$i \leftarrow 1$
while $Q_n \neq \emptyset$ do
    $u \leftarrow$ first $(Q_n)$
    $j \leftarrow i$
    repeat
        if connected$(B_i \setminus (\{u\}, A(u)))$ then
            $C_{tmp} \leftarrow$ isolate$(C_i, u)$
            if $C_{tmp} \neq$ null then
                $C_i \leftarrow C_{tmp}$
                $S_i \leftarrow S_i \cup \{u\}$
                $B_i \leftarrow B_i \setminus (\{u\}, A(u))$
        $i \leftarrow (i \bmod n) + 1$
    until $u \in S_i$ or $i$=$j$
    if $u \notin S_i$ then
        Give up and abort
end
```

Main loop: Initially, n backup configurations are created as copies of the normal configuration. A queue of nodes (Q_n) and a queue of links (Q_a) are initiated. The node queue contains all nodes in an arbitrary sequence. The link queue is initially empty, but all links in the network will have to pass through it. Method First retrns the first item in the queue, removing it from the queue.

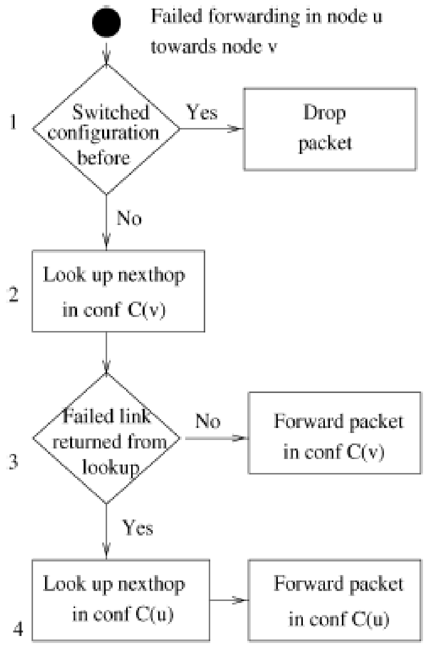


Fig. 2. Packet forwarding state diagram.

Proposition 4.1: Node selects configuration C_i so that v ∉ N(P_i(ų , d)) , if v≠d.

Proof: If v≠d. node ų selects C(v) in step 2, and neither node v nor link (ų , v) will be in the shortest path P_i(ų , d). If C(ų) = C_i and C(v)= C_i as in Fig. 3(a), then C(ų , v) = C_i according to the definition of an isolated node and (2). Forwarding step 2 will select C(v)= C_i and A(P_i(ų , v)) does not contain (ų , v) .

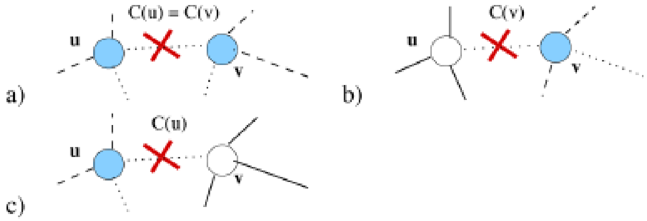


Fig. 3.

A. Implementation Issues

The forwarding process can be implemented in the routing equipment as presented above, requiring the detecting node to know the backup configuration C(v) for each of its neighbors. Node ɥ would then perform at most two additional next-hop look-ups in the case of a failure. However, all nodes in the network have full knowledge of the structure of all backup configurations.

The shifting of traffic from the normal path to a recovery path changes the load distribution in the network, and can in some cases lead to congestion and packet loss. We therefore test the effect our scheme has on the load distribution after a failure. To do this, we have performed simulations of the European COST239 network [22] shown in Fig. 4, designed to connect
major cities across Europe. All links in the network have

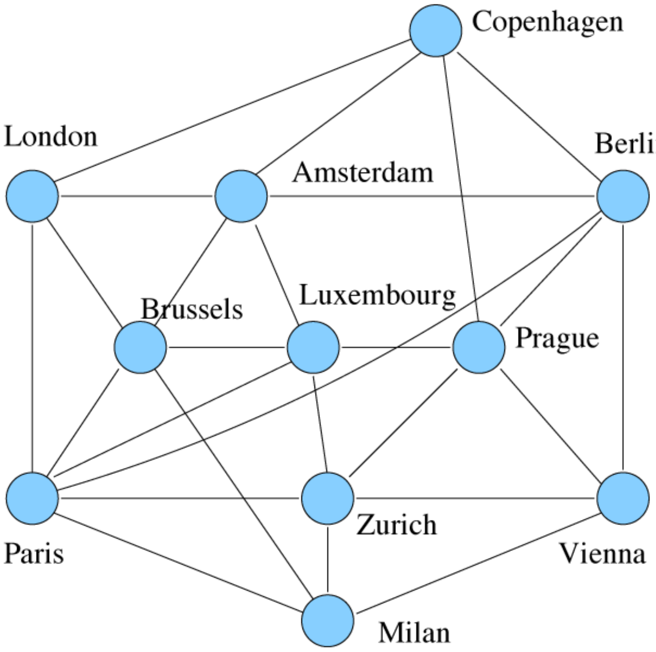


Fig. 4. The COST239 network.

equal capacity. To achieve a good load distribution and inimize
the chances of congestion in the failure-free case, we adopt the link weight optimization heuristic introduced in [23]. They define a piecewise linear cost function Φ that is dependent on the load l (a) on each of the links in the network. Φ is convex and resembles an exponentially growing function. They then introduce a local search heuristic that tries to minimize the value of Φ by randomly perturbing the link weights. This local search heuristic has been shown to give performance that is close to the optimal solution that can be achieved by a connection oriented technology like MPLS.

Backup Path Lengths

Fig. 6 shows path length distribution of the recovery paths after a node failure. The numbers are based on 100 different synthetic Waxman topologies with 32 nodes and 64 links. All the topologies have unit weight links, in order to focus more on the topological characteristics than on a specific link weight
configuration. Results for link failures show the same tendency
and are not presented.

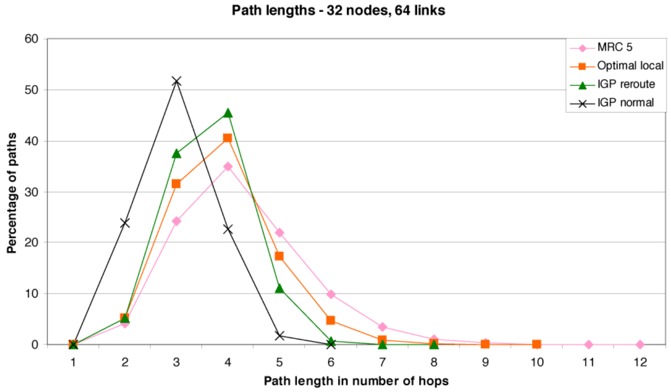


Fig. 6. Backup path lengths in the case of a node failure.

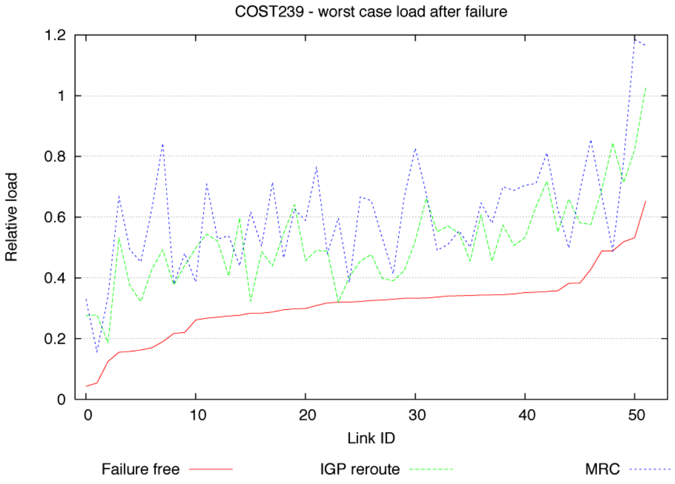


Fig. 8. Load on all unidirectional links in the failure free case, after IGP re-convergence, and when MRC is used to recover traffic. Shows each individual links worst case scenario.

VI. Recovery Load Distribution

MRC recovery is local, and the recovered traffic is routed in a backup configuration from the point of failure to the egress node. This shifting of traffic from the original path to a backup

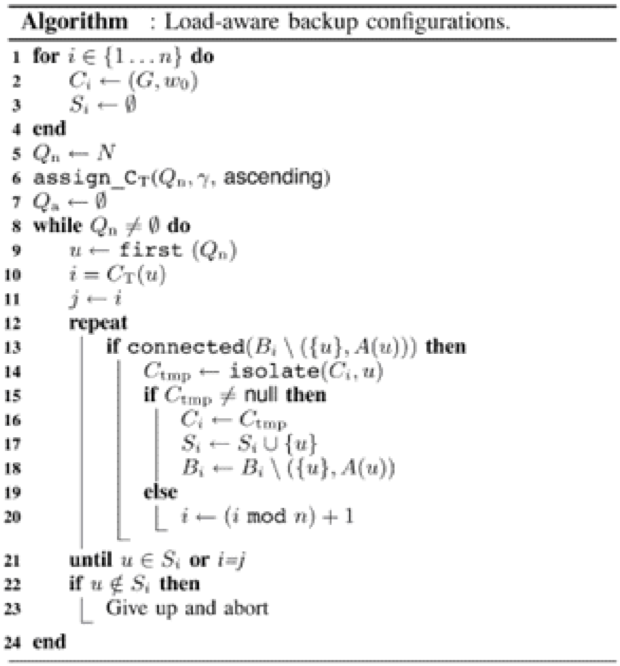
path affects the load distribution in the network, and might lead to congestion. If MRC is used for fast recovery, the load distribution in the network during the failure depends on three factors:

(a) The link weight assignment used in the normal Configuration C_0,

(b) The structure of the backup configurations, i.e., which links and nodes are isolated in each $C_i \in \{C_1, \ldots\ldots C_n\}$,

(c) The link weight assignments used in the backbones $B_1 \ldots\ldots B_n$ of the backup configurations.

The link weights in the normal configuration (a) are important since MRC uses backup configurations only for the traffic affected by the failure, and all non-affected traffic is distributed according to them. The backup configuration structure (b) dictates which links can be used used in the recovery paths for each failure. The backup configuration link weight assignments (c) determine which among the available backup paths are actually used.



```
Algorithm   : Load-aware backup configurations.
1  for $i \in \{1 \ldots n\}$ do
2      $C_i \leftarrow (G, w_0)$
3      $S_i \leftarrow \emptyset$
4  end
5  $Q_n \leftarrow N$
6  assign_C_T($Q_n, \gamma$, ascending)
7  $Q_a \leftarrow \emptyset$
8  while $Q_n \neq \emptyset$ do
9      $u \leftarrow$ first ($Q_n$)
10     $i = C_T(u)$
11     $j \leftarrow i$
12     repeat
13         if connected($B_i \setminus (\{u\}, A(u))$) then
14             $C_{tmp} \leftarrow$ isolate($C_i, u$)
15             if $C_{tmp} \neq$ null then
16                 $C_i \leftarrow C_{tmp}$
17                 $S_i \leftarrow S_i \cup \{u\}$
18                 $B_i \leftarrow B_i \setminus (\{u\}, A(u))$
19             else
20                 $i \leftarrow (i$ mod $n) + 1$
21     until $u \in S_i$ or $i{=}j$
22     if $u \notin S_i$ then
23         Give up and abort
24 end
```

Definition: The potential γ (ų) of a node ų is the sum of the

load on all its incoming and outgoing links:

$$\gamma(u) = \sum_{v \in N} \left(l(u,v) + l(v,u) \right). \qquad (11)$$

Definition: The potential γ_i of a backup configuration C_i is the sum of the potential of all nodes that are isolated in C_i:

$$\gamma_i = \sum_{u \in S_i} \gamma(u). \qquad (12)$$

Our modified backup configuration construction method is defined in Algorithm 3. As in Algorithm 1, the input to our algorithm for generating backup configurations is the normal configuration C_0 and the number n of backup configurations we want to create. We start our configuration generation algorithm by ordering all nodes with respect to their potential and assigning each node to a tentative backup configuration C_T(ų) (line 6 in Algorithm 3), so that the potential γ_i of each backup configuration is approximately equal:

$$\gamma_i \approx \gamma_j, \quad i, j \in \{1, \ldots, n\}. \qquad (13)$$

Fig. 9. Load on all unidirectional links in the COST239 network after the worst case link failure. Top: Optimized MRC versus complete IGP rerouting. Bottom: Standard versus optimized MRC.

VII. Related Work

MRC operates without knowing the root cause of failure, i.e., whether the forwarding disruption is caused by a node or link failure. This is achieved by using careful link weight assignment according to the rules we have described. The link weight assignment rules also provide basis for the specification of a forwarding procedure that successfully solves the last hop problem. The performance of the algorithm and the forwarding mechanism has been evaluated using simulations. We have shown that MRC scales well: 3 or 4 backup configurations is typically enough to isolate all links and nodes in our test topologies. MRC backup path lengths are comparable to the optimal backup path lengths—MRC backup paths are typically zero to two hops longer. MRC thus achieves fast recovery with a very limited performance penalty.

References

[1] D. D. Clark, "The design philosophy of theDARPAinternet protocols," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 18, no. 4, pp. 106–114, Aug. 1988.

[2] A. Basu and J. G. Riecke, "Stability issues in OSPF routing," in *Proc. ACM SIGCOMM*, San Diego, CA, Aug. 2001, pp. 225–236. [3] C. Labovitz, A. Ahuja, A. Bose, and F. Jahanian, "Delayed internet routing convergence," *IEEE/ACM Trans. Networking*, vol. 9, no. 3, pp. 293–306, Jun. 2001.

[4] C. Boutremans, G. Iannaccone, and C. Diot, "Impact of link failures on
VoIP performance," in *Proc. Int. Workshop on Network and Operating System Support for Digital Audio and Video*, 2002, pp. 63–71.

[5] D.Watson, F. Jahanian, and C. Labovitz, "Experiences with monitoring OSPF on a regional service provider network," in *Proc. 23rd Int. Conf.*
Distributed Computing Systems (ICDCS'03), Washington, DC, 2003,
pp. 204–213, IEEE Computer Society.

[6] P. Francois, C. Filsfils, J. Evans, and O. Bonaventure, "Achieving sub-second IGP convergence in large IP networks," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 35, no. 2, pp. 35–44, Jul. 2005.

[7] A. Markopoulou, G. Iannaccone, S. Bhattacharyya, C.-N. Chuah, and
C. Diot, "Characterization of failures in an IP backbone network," in *Proc. IEEE INFOCOM*, Mar. 2004, vol. 4, pp. 2307–2317.

[8] S. Nelakuditi, S. Lee, Y. Yu, Z.-L. Zhang, and C.-N. Chuah, "Fast local
rerouting for handling transient link failures," *IEEE/ACM Trans. Networking*,
vol. 15, no. 2, pp. 359–372, Apr. 2007.

[9] S. Iyer, S. Bhattacharyya, N. Taft, and C. Diot, "An approach to alleviate
link overload as observed on an IP backbone," in *Proc. IEEE INFOCOM*, Mar. 2003, pp. 406–416.

[10] S. Rai, B. Mukherjee, and O. Deshpande, "IP resilience within an autonomous system: Current approaches, challenges, and future directions,"
IEEE Commun. Mag., vol. 43, no. 10, pp. 142–149, Oct. 2005.

[11] S. Bryant, M. Shand, and S. Previdi, "IP fast reroute using not-via addresses," Internet Draft (work in progress), draft-ietf-rtgwg-ipfrrnotvia-
addresses-01, Jun. 2007.

[12] P. Francois, M. Shand, and O. Bonaventure, "Disruption free topology
reconfiguration in OSPF networks," in *Proc. IEEE INFOCOM*, Anchorage,
AK, May 2007, pp. 89–97.

[13] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. San Francisco, CA: W. H. Freeman & Co., 1979.

[14] P. Psenak, S. Mirtorabi, A. Roy, L. Nguen, and P. Pillay-Esnault, "MTOSPF: Multi Topology (MT) routing in OSPF," IETF Internet Draft (work in progress), draft-ietf-ospf-mt-07.txt, Nov. 2006.

[15] T. Przygienda, N. Shen, and N. Sheth, "M-ISIS: Multi Topology (MT)
routing in IS-IS," Internet Draft (work in progress), draft-ietf-isis-wgmulti-
topology-11.txt, Oct. 2005.

[16] M. Menth and R. Martin, "Network resilience through multi-topology
routing," in *Proc. 5th Int. Workshop on Design of Reliable Communication Networks (DRCN)*, Oct. 2005, pp. 271–277.

[17] A. Medina, A. Lakhina, I. Matta, and J. Byers, "BRITE: An approach to
universal topology generation," in *Proc. IEEE MASCOTS*, Aug. 2001, pp. 346–353.

[18] B. M. Waxman, "Routing of multipoint connections," *IEEE J. Sel.*

Areas Commun., vol. 6, no. 9, pp. 1617–1622, Dec. 1988.

[19] T. Bu and D. Towsley, "On distinguishing between internet power law
topology generators," in *Proc. IEEE INFOCOM*, New York, Jun. 2002, pp. 638–647.

[20] Rocketfuel Topology Mapping. [Online]. Available: http://www.cs. washington.edu