Improving the Resilience in IP Networks

Gero Schollmeier[§], Joachim Charzinski[*], Andreas Kirstädter[*], Christoph Reichert[‡],
Karl J. Schrodi[*], Yuri Glickman[‡], and Chris Winkler[*]
[§]Email: gero.schollmeier@genion.de
[*]Siemens AG, Munich, Germany, Email: {joachim.charzinski|andreas.kirstaedter|karl.schrodi|chris.winkler}@siemens.com
[‡]FhG Fokus, Berlin, Germany, Email: {reichert|glickman}@fokus.fhg.de

***Abstract*— Quality of Service issues of IP networks are mostly related to guaranteeing bandwidth for flows. However, many interactive real-time applications also require this bandwidth in an uninterrupted fashion. This paper describes how multi-path routing and local failure reaction can be employed to provide uninterrupted QoS to applications. We show how multi-path route sets can be found in reasonably meshed networks and how multi-path routing can be used to save on the spare capacity required in case of link failures.**

I. Introduction

IP routing has been designed to re-establish connectivity after almost any failure of network elements. However, current implementations usually fail to do so in a time frame acceptable for interactive human communication as reconfiguration may often take longer than the few hundred milliseconds which are typically deemed to be acceptable. The delays result from infrequent link supervision messages and from the fact that always a number of nodes have to be informed about the failure and need time to evaluate and initiate appropriate countermeasures. While an interruption of connectivity of several seconds may be tolerable for most machine-to-machine communication, it severely limits the use of current IP networks for real-time human communication.

Link failures are by far the most frequent failures in a network [1], [2], [3]. Therefore, protection against link failures will be a significant step towards improved network performance. The basic idea we propose to protect a network against link failures is the use of multiple active paths at any node towards any destination. When a node locally detects a failed link or port, it can autonomously remove the defective element from the forwarding table and continue using the remaining next hops for forwarding packets. We explicitly refer to this as fast local reaction [4], [5] in contrast to a reaction involving other components of the network.

In Sec. II of this paper we review briefly current IP routing, including existing proposals for improved failure reaction. In Sec. III we outline a method to drastically improve network availability, which we call O2 routing and point out the basic topology requirement a network must meet in order to support the proposed routing. Using an example network, we also point out the difference of our proposed routing compared to conventional shortest path routing. A simple algorithm to actually compute the proposed loop-free multi-path routes is outlined in Sec. IV, including an example which shows that even in a sparse network O2 routing is often possible. Finally we show in Sec. V that the proposed routing offers not only significant advantages in reliability but also, beyond its original design goal of accelerated failure reaction, an improved traffic performance.

[1]Acknowledgement. This work was partially funded by the Bundesministerium für Bildung und Forschung (ministry of education and research) of the Federal Republic of Germany under contract 01AK045. The authors alone are responsible for the content of the paper

II. State-of-the-Art IP Routing

Today the primary and conceptually equivalent intra domain routing protocols in IP networks are OSPF [6] and IS-IS [7]. Both provide all routers with a complete view of the topology of a network domain. Each router can then determine the shortest path (in terms of cost metrics assigned to links and interfaces) towards each destination and store the corresponding next hop in its forwarding table. The shortest path approach automatically guarantees loop-free routing.

Basically, single path routing suffers from two shortcomings: (1) A single link failure will cause an often time consuming rerouting of traffic, which is not acceptable for traffic with stringent QoS requirements. (2) The single path routing tends to be very susceptible to congestion in case of dynamic load changes.

Typical values for failure reaction in today's IP networks fall in the range of tens of seconds. There are proposals to speed up the failure detection to sub-second times e.g. [8], [9] and for very homogeneous and moderately sized networks a convergence time in the range of seconds is reported. Since a distributed reaction requires message exchange and involves multiple network elements the failure behavior depends on the size and structure of the network and can therefore in general not be accelerated further (this dilemma can be avoided by a completely local failure handling scheme as proposed in this paper). As an answer, one might be tempted to establish two or more disjoint paths. However, since the paths usually take multiple hops, a failure will again cause a message exchange until the source node is informed to switch over to the alternate path. Furthermore, appropriate measures must be taken to avoid loops when using disjoint paths. It has also been proposed in [5] to use back-up LSP paths (labeled switched paths) such as those provided by MPLS [10]. However this requires additional network capacity and management for the (normally unused) LSP paths.

To appear in Proc. IEEE HPSR 2003, Torino, Italy, June 2003

With respect to the second issue, sophisticated algorithms have been proposed and implemented to assign link costs in such a way that all available links carry roughly equal load relative to their bandwidth, e.g. [11], [12]. As a simple "rule of thumb", Cisco proposes to set the cost of a link equal to the inverse of the bandwidth. However, this as well as more immediate cost functions such as the length or propagation delay of links often do not result in a routing providing a balanced traffic distribution [13], [12]. The load balancing can to some degree be addressed by the Equal-Cost Multi Path (ECMP) feature of OSPF. Yet in practice there will typically not exist a link cost assignment which renders multiple next hops for all destinations at all routers which would enable a local failure handling. For instance, [2] provides rules to assign link costs as to alleviate link overload but proves that a solution which also covers the inherent problems (e.g. last-hop problem) exists only for an extremely restricted set of topologies and is not widely applicable.

As a consequence, while current IP networks are very good in recovering from a loss of connectivity, they do so too slowly for many interactive applications.

III. An Improved Routing Method

Fast recovery from link failures and efficient usage of resources are often viewed as conflicting requirements. Apart from cost issues like keeping the required number of links and the spare capacity reasonably low, the most critical, but at the same time mandatory requirements are to have

- multiple alternate paths at *every* node to facilitate local failure reaction and
- loop-free destination based routing.

Whereas the first requirement reflects the fundamental idea to increase availability, the second requirement covers the practical aspect that destination based forwarding is the method implemented in today's routers. Using destination based forwarding their sophisticated wire-speed packet engines can remain unchanged.

A. Resilient unequal cost multi-path routing

To address these issues together with load sharing, we propose a new routing algorithm providing each node with at least two disjoint *next hops* (connecting to different neighboring nodes and using different cable ducts) towards any given destination. The challenge in such a routing algorithm is to avoid loops in a destination based forwarding environment.

By using at least two next hops, a node can locally and thus very fast re-distribute the traffic to the remaining next hop(s) if a route fails. This local reaction will always be faster than any distributed reaction involving multiple elements and requiring message exchange.

An admission control [14] at the borders of the network can be used to limit admitted priority traffic to a certain admission threshold. Routing and allocated traffic distribution weights will ensure that up to that threshold priority traffic will always reach any destination even in case of a link failure.

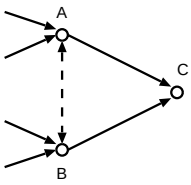

Fig. 1. Basic last-hop cell of an O2 network.

Local failure detection can be accelerated drastically by introducing a new fast failure detection mechanism [15]. Independent of the slow HELLO message exchange of the routing protocols, additional IP probing messages are exchanged at a much faster rate, say every 25ms, so that a failure can be detected e.g. after 100ms (4 intervals)[1]. This scheme covers all failures up to and including the IP layer and hence protects links, line cards and parts of the router's forwarding path. It enhances a potentially present layer 2 failure handling but also completely guards unprotected environments like Gigabit Ethernet over DWDM. After a failure both nodes terminating a failing link will locally remove the corresponding next hop from their routing tables and use the remaining next hop entries for the same destination to continue forwarding packets. As the nodes can perform this action locally and without informing other nodes, the fault reaction will be very fast and meet the QoS requirements even of critical services such as interactive real-time voice or video services.

B. O2 Routing

Evidently the minimum number of next hops per node required to improve resilience is two. To keep the discussion comprehensible we will in the following focus on basic issues in providing exactly two next hops. In Sec. V-A we will then confirm that two next hops already provide a significant advance over the single path approach.

We call our proposed algorithm an "O2" algorithm (for "outdegree 2", using graph theory terminology). First consider Fig. 1. It shows a basic routing in an O2 network at the last hop towards a destination C. Nodes A and B are both neighbours of C and are linked with each other. To make A and B O2 nodes, the latter link, shown as a broken line, will have to be used in either direction for packets towards node C if one of the direct links towards C fails. In order to prevent routing loops, the link A-B is not used for traffic towards node C unless one of the links A-C or B-C fails. We will therefore call such a link a "joker link" (or simply a "joker"), as it can be locally used when needed by any of the nodes A or B without first informing the node at the other end. Note that the "joker" attribute is specific for the considered destination, i.e. different links in the network will serve as joker links for different destinations, while they are used as normal links for forwarding packets towards other destinations.

The resulting routing from a given source towards a destination will be called a "hammock" and the set of hammocks

[1] A 50 octets packet sent every 25ms generates a load of 16kbit/s which is negligible for backbone network links.

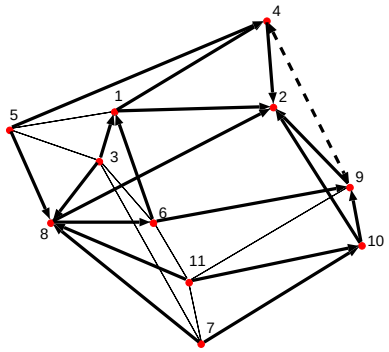

Fig. 2. Hammock set towards node 2 in the COST 239 network.

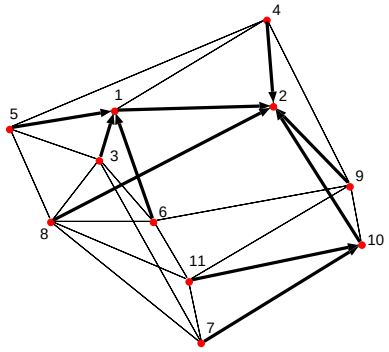

Fig. 3. Shortest paths towards node 2 in the COST 239 network.

from all nodes towards a given destination will be called a "hammock set". It can be shown that unless there are parallel links available on the last hop, which for reliability reasons need to be guided in different cable ducts, at least one joker link will be required in every hammock set, i.e. for every destination [16]. In sparsely meshed networks, more jokers may be needed.

Fig. 2 shows such a hammock set in the COST 239 reference network [17] with destination node 2 (Berlin). The thin lines indicate links which are not used for routing towards node 2, but they are used in hammock sets towards other destinations. In Fig. 2, the joker is the link between nodes 9 and 4, indicated by the dashed double-headed arrow. As can easily be seen, no other joker link is required in this hammock set and each node has a choice of two next hops towards the destination node. Assuming equal distribution of traffic between the next hops at each node, the average hop count for all sources to destination node 2 is 1.625 in this case.

For comparison, Fig. 3 shows a shortest path routing towards the same destination. In comparison to the O2 routing in Fig. 2, fewer links are used, which is reflected in the slightly lower average hop count of 1.5, but at the same time there is no load sharing between links and no chance for fast local failure reaction. After any link failure, all nodes need to be informed before they can compute new routing tables.

IV. The O2 Algorithm

In this section we outline an algorithm that, when given a suitable network topology, produces an O2 routing as introduced above. The network must meet some necessary conditions in order to be suitable for O2 routing: Each node must form at least one triangle as shown in Fig. 1 with its neighbours. A simplified algorithm to derive the routing towards a given destination D is as follows:

For each destination D_i do:

1) *Initialize the set $S_1(D_i)$ of all nodes which have a direct link towards the destination D_i. Initialize hammock set $R(D_i)$ by those direct links.*
2) *Check whether the nodes in $S_1(D_i)$ are directly interconnected and select one of these interconnections as the joker link for destination D_i.*
3) *Store the target node and the two nodes terminating the joker link (which are now O2 nodes) in a list L_{O2} of O2 nodes and remove them from $S_1(D_i)$.*
4) *Check the remaining nodes in $S_1(D_i)$ whether they have a connection to one of the nodes already contained in L_{O2}. If yes, add the corresponding directed link to $R(D_i)$ and move the node from $S_1(D_i)$ to L_{O2}.*
5) *Repeat step 4 until no more nodes are removed from $S_1(D_i)$.*
6) *Check the remaining nodes in the network which are not yet part of L_{O2} whether they have connections to two nodes in L_{O2}. If yes, add the corresponding directed links to $R(D_i)$ and add the node to L_{O2}.*
7) *Repeat step 6 until all nodes of the network (except the destination) are contained in L_{O2} or no new O2 node was found in this step.*

No more jokers are allowed in any of the additional "rounds" of the algorithm after step 6 has been executed for the first time. As the steps have to be repeated for every destination node in the network, the complete algorithm is of order $O(n_N^3)$ for a network with n_N nodes.

If in a given network topology a node is not O2 connected to a destination, the above algorithm will immediately detect that and can provide a warning so that e.g. a link can be added or modified. However, examining some practical core networks, we found that in most cases a relatively simple modification of links will be sufficient to make a network O2 capable, even if it was originally designed only for shortest-path routing. Of course, a node with only a single link can hardly ever be an O2 capable node, but such a node should not be a core node carrying transit traffic anyway.

We also point out that the above simple algorithm will always compute a loop free routing, thus enabling destination based routing. The required router resources are similar to those needed for shortest-path routing. This is an immediate result of the fact that after the initial joker has been placed, "downstream" connections towards the destination will only be allowed to nodes which are already O2 nodes. Simultaneously, a node will not be allowed to make any additional outgoing connection once it has become an O2 node.

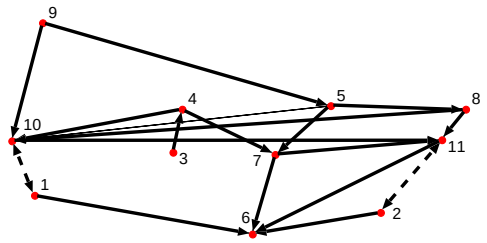

Fig. 4. A hammock set in a sparsely connected network.

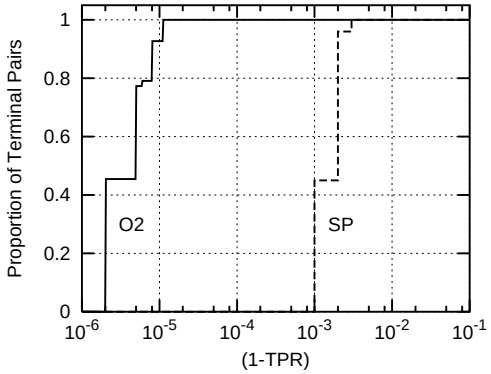

Fig. 5. Cumulative distribution of (1-TPR) for the COST 239 network, Comparison of O2 vs. shortest path.

The algorithm even covers node failures: In case of a node failure, which is handled like a failure of all of its links, the neighboring nodes will locally forward the traffic around the failure on their remaining next hops. Of course the traffic from the failing node will disappear.

With additional simple tie brake rules (e.g. precedence of higher node number) the algorithm operates deterministically. All nodes can independently compute the same set of routes based on a common view of the network topology just like the shortest path calculation in OSPF and IS-IS. Generally speaking, O2 routing does not define a new protocol but offers a replacement for the shortest path routing algorithm embedded into the routing protocols.

Both the hammock set in Fig. 2 and the hammock in Fig. 6 as well as the evaluation of O2 properties in Sec. V were calculated using the above algorithm. But even in less connected networks, O2 routing is possible using more complex algorithms (which go beyond the scope of this paper). Fig. 4 shows an example of a sparse network, a former Sprint network topology taken from [18]. The hammock set towards node 6 (Fort Worth) is highlighted. It can be seen that of course node 3 (Boulder) with only one link cannot provide O2. Furthermore, two jokers instead of one, namely 1-10 and 2-11, are required to provide O2 for all remaining nodes. Only a single link (5-10) is not used in that hammock set whereas in the more densely connected COST 239 network in Fig. 2, which also has 11 nodes, typically 6–8 links will not be part of a given hammock set.

V. Evaluation of O2 Routing

A. Availability Issues

In this section we will confirm that two next hops already provide a significant advantage over the single path approach. Typically links in today's networks (including the respective line cards at both ends of a link) show an unavailability ratio somewhere in the range of 10^{-2} to 10^{-3} [19], [3], corresponding to an availability of 0.99 to 0.999.

Assuming for sake of simplicity that the unavailability ratio for all links is equal and that failure events are independent, the probability to have k simultaneously unavailable links in a network with n_L links is given by the Binomial distribution. Assuming an unavailability ratio of 10^{-3} (corresponding to a good quality line) and a network of n_L=25 links, the probability of one failure will be 0.025, i.e. an availability of 0.975, compared to a probability of two failures of 0.00029, i.e. an availability of 0.99971. Thus the improvement in availability for priority traffic in case two paths are available instead of only one will be roughly two orders of magnitude[2].

A common end-to-end resilience measure is the probability to find a working path between a pair of terminals at a given time, the so called terminal pair reliability (TPR) [20]. Generally the calculation of the TPR is not trivial. However, for simple networks it can be derived applying straightforward combinatorics. Fig. 5 charts the cumulative distribution of (1-TPR) for all node pairs in the COST 239 network for O2 routing and shortest path routing assuming a link unavailability ratio of 10^{-3}. It can be seen from the figure that the O2 TPR is improved by two orders of magnitude compared to the shortest path TPR. It has to be noted that this comparison assumes uninterrupted QoS ("QoS TPR"). Of course also the shortest path network will after some time converge to a new routing and hence the plain "Connectivity TPR" of both routing methods is much higher.

Naturally, the improved availability with multi-path routing does not come for free. Even in the case of a single failure, the traffic on some of the remaining links will increase while for other links the traffic may decrease because they are "behind" the failing link and thus shielded from some of the traffic. This will be discussed in the following section.

B. Traffic Performance of O2 Routing

Apart from the increased availability discussed in Sec. III, the distribution of traffic[3] over two links instead of only one at each node will result in a significant improvement of load sharing throughout the network. This in turn will cause the network to be less sensitive to re-directed traffic after a failure as well as to sudden overload on one or some few edge-to-edge hammocks, thus making the network more robust.

With O2 routing, traffic towards one destination is carried by more links than with shortest path routing. This is especially

[2]If the various links have different availability or if correlated link failures have to be taken into account, the Binomial distribution is not applicable.

[3]Load sharing can be realized per-packet (e.g. round robin) or based on a hash value computed from source and destination IP addresses [21]. The latter maintains the packet sequence integrity. Both schemes are implemented in current routers (ECMP) [6].

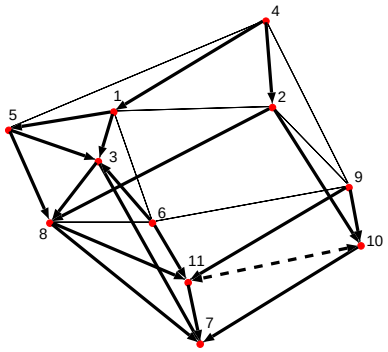

Fig. 6. Hammock from node 4 to node 7 in the COST 239 network.

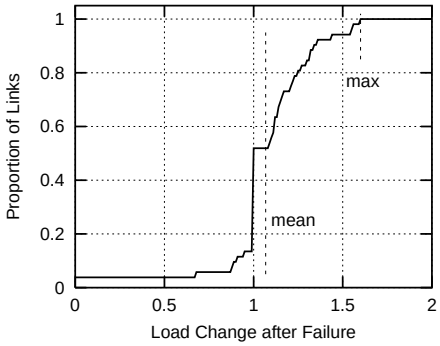

Fig. 7. Cumulative distribution of relative increase of traffic on all other links after a failure of link 8-2 in the COST 239 network.

obvious in case of a failure, where in (single) shortest path routing all traffic between a source and a destination is redirected to another single path whereas with O2 routing the redistributed load after a failure is shared by more paths. With O2 routing, the number of links carrying traffic towards a given destination increases roughly as 2^{n_H} with the number of hops n_H. With a finite number of hops, the practical increase of the number of used links is more limited, as all paths must end at one of the ports of the destination.

An example of traffic distribution in an O2 routing network is given in Fig. 6 showing the single hammock (not the hammock set) from node 4 (Copenhagen) to node 7 (Milan) in the COST 239 network. Instead of 3 links in shortest path routing, our O2 approach uses 14 links and all links connected to Milan are actually used by the hammock. Four links are used in parallel for the second hop. Thus we can in fact expect a significantly improved network performance in case of heavy local traffic bursts or link failures. As "traffic invariably arises where you least expected it" [22], this is a significant advantage. Alternatively, this load sharing advantage can of course be exploited to reduce the safety margin in capacity planning and thus reduce network cost.

For the traffic matrix given in [17], Fig. 7 shows the cumulative distribution of the traffic change on all links in the network after a worst case failure of link 8-2, which is the link between the two nodes with highest mutual traffic. The horizontal axis is the ratio of link load after the failure to link load before the failure. Thus a ratio of 1 indicates constant load. The vertical axis is the proportion of links in the network that experience at most the charted load change. The vertical step of 39% in Fig. 7 at a ratio of 1 indicates that the load on more than one third of the links remains unaffected by the failure of link 8-2. On the most heavily affected link, the traffic increases by a factor of roughly 1.6. Around 14% of the links are even less loaded than before because they are "behind" the failed link and are thus shielded from a fraction of the traffic. The failed link itself of course accounts for the ratio of 0. The average link load remains roughly constant, as indicated by the dashed vertical line which represents the mean of all load changes.

In a similar investigation for single path OSPF routing using a hop count metric, the most affected link has an increase of load by a factor of 6.5 (instead of 1.6 for O2 routing) and the mean increase of link loads was 1.52 (instead of 1.07). Better performance of the single path approach in the failure-free case can of course be achieved by applying one of the many proposed schemes to optimize OSPF weights, e.g. as mentioned in [12]. However, we also expect additional improvements for load sharing with O2 routing from optimizing traffic distribution weights, as the above example has been evaluated with default 1:1 load sharing.

The obvious drawback with O2 routing is that inevitably some paths must be used which are longer than the shortest path. Thus the total traffic in the network increases by an amount depending on the topology as well as connectivity and the traffic matrix. In the case of the COST 239 network [17], the total network traffic using the specified traffic matrix increased by roughly 25%. While it is tempting to argue that consequently the installed capacity of the network has to be increased by 25%, we have good reasons to believe that this effect will be more than compensated by the distribution effect of O2 routing. No operator can afford a network to become severely overloaded due to a single link failure or due to a sudden load burst in one or some few paths. As shown above, the capacity required to avoid such overload is significantly greater for shortest path routing than for O2 routing. For example, our observation following Fig. 7 indicates that network capacity for shortest-path routing has to be increased by a factor of 1.5 to avoid overload in the case of a failure. This is twice the amount of traffic increase due to O2 routing compared to shortest path routing. Further studies with different networks are currently being performed.

VI. Conclusion

We introduced a concept to provide multiple next hops per destination for IP routing at every network node in order to allow all nodes to locally react to link failures and thus to significantly reduce outage times in IP networks, which will be a prerequisite for offering high quality interactive real-time services. The corresponding routing can easily be established in reasonably meshed networks, as we have shown with a draft algorithm that provides routes for loop free destination

based multi-path routing. Although multi-path routing slightly increases the link loads in normal operation by sometimes using longer paths than necessary, overall network capacity can be saved because multi-path routing distributes the load change after a link failure in the network.

Further work comprises improvement of the routing algorithms, lab experiments, statistical evaluation of availability and load sharing performance as well as multi-domain resilience concepts.

REFERENCES

[1] C. Labovitz, G. R. Malan, and F. Jahanian, "Internet Routing Instability," *IEEE/ACM Transactions on Networking, vol. 6 no. 5*, pp. 515–528, 1998.

[2] S. Iyer, S. Bhattacharyya, N. Taft, and C. Diot, "An approach to alleviate link overload as observed on an IP backbone," in *Proc. Infocom*, 2003.

[3] ETSI Document EN 300416 V1.2.1, "Network Aspects (NA): Availability Performance of Path Elements of International Digital Paths," Aug. 1998.

[4] K. J. Schrodi, "High Speed Networks for Carriers," in *Proc. IFIP/IEEE PfHSN*, Apr. 2002, pp. 229–242.

[5] D. Stamatelakis and W. Grover, "IP Layer Restoration and Network Planning Based on Virtual Protection Cycles," *IEEE J. Select. Areas Commun.*, vol. 18, pp. 1938–1949, 2000.

[6] J. Moy, "OSPF Version 2," RFC 2328, Apr. 1998.

[7] ISO, "Intermediate System-to-Intermediate System (IS-IS) Routing Protocol," ISO/IEC 10589, 2002.

[8] C. Alaettinoglu, V. Jacobson, and H. Yu, "Towards Milli-Second IGP Convergence," IETF draft-alaettinoglu-ISIS-convergence-00.txt, Nov. 2000.

[9] A. Basu and J. Riecke, "Stability Issues in OSPF Routing," in *ACM SIGCOMM 2001, San Diego*, Aug. 2001.

[10] E. Rosen, A. Viswanathan, and R. Callon, "Multiprotocol Label Switching Architecture," RFC 3031, Jan. 2001.

[11] A. Riedl, "A Hybrid Genetic Algorithm for Routing Optimization in IP Networks Utilizing Bandwidth and Delay Metrics," in *IEEE Workshop on IP Operations and Management (IPOM), Dallas*, Oct. 2002.

[12] B. Fortz and M. Thorup, "Optimizing OSPF/IS-IS Weights in a Changing World," *IEEE J. Select. Areas Commun.*, vol. 20, pp. 756–767, May 2002.

[13] B. Fortz, J. Rexford, and M. Thorup, "Traffic Engineering with Traditional IP Routing Protocols," *IEEE Commun. Mag.*, pp. 118–124, Oct. 2002.

[14] J. Roberts, "Traffic theory and the Internet," *IEEE Communications Mag.*, vol. 39, pp. 94–99, 2001.

[15] K. Kompella, " Protocol Liveness Protocol ," IETF draft-kompella-ragplp-00.txt, 0ct 2002.

[16] C. Reichert, G. Schollmeier, C. Winkler, *et al.*, "Topology Considerations for Resilient IP Networks," submitted to ITC18 (International Teletraffic Congress), 2003.

[17] P. Batchelor *et al.*, *Ultra High Capacity Optical Transmission Networks – Final report of Action COST 239*. Zagreb, Croatia: CPI, 1999.

[18] Network Maps. [Online]. Available: http://www.nthelp.com/maps/

[19] M. To and P. Neusy, "Unavailability Analysis of Long-Haul Networks," *IEEE J. Select. Areas Commun.*, vol. 12, no. 1, pp. 100–109, Jan. 1994.

[20] A. Iselt, "Terminal pair availability calculation algorithms for communication networks," in *Proceedings DRCN Design of Reliable Communication Networks, Munich, Germany*, apr 2000.

[21] Z. Cao, Z. Wang, and E. Zegura, "Performance of Hashing-Based Schemes for Internet Load Balancing," in *Proceedings IEEE Infocom*, 2000.

[22] D. Johnson *et al.*, "The Evolution of a Reliable Transport Network," *IEEE Commun. Mag.*, pp. 52–57, Aug. 1999.