Efficient Network Recovery using Multiple Standard Routing Configurations

Lily Rohini.D
Research Scholar, Department of MCA, School of Computing Science,
VELS University, Pallavaram, Chennai -117

Dr. Muthukumaravel. A
Associate Professor, Department of MCA, School of Computing Science,
VELS University, Pallavaram, Chennai -117

Kalpana.Y
Lecturer, Department of MCA, School of Computing Science,
VELS University, Pallavaram, Chennai -117

Abstract— In recent years, internet plays a important role in our communication infrastructure. Routing protocols after network failure become a problem due to slow convergence. To ensure fast recovery from link and node failure in IP networks, we demonstrate a new recovery scheme called Multiple Standard Routing Configuration (MSRC). Our scheme guarantees recovery from single failure scenarios, using a method to handle both link and node failures and without knowing the root cause of the failure. MSRC is strictly connectionless, and assumes only destination based on hop-by-hop forwarding. MSRC is based on keeping additional routing information in the routers, and grant packet forwarding to keep on an alternative output link immediately after the detection of a failure. It can be resolved with only minor changes to existing solutions. In this paper we present MSRC, and analyze its performance with respect to scalability, backup path lengths and load delivery after a failure.

Index Terms—Networks Protocols, Routing Protocols, Reliability, Protection

I. INTRODUCTION

IN recent 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.

The ability to recover from failures has always been a central design goal in the Internet [1]. IP networks are intrinsically robust, since IGP routing protocols like OSPF are designed to update the forwarding information based on the changed topology after a failure. This re-convergence assumes full distribution of the new link state to all routers in the network domain. When the new state information is distributed, each router individually calculates new valid routing tables.

This network-wide IP re-convergence is a time consuming process, and a link or node failure is typically followed by a period of routing instability. During this period, packets may be dropped due to invalid routes. This phenomenon has been studied in both IGP [2] and BGP context [3], and has an adverse effect on real-time applications [4]. Events leading to a re-convergence have been shown to occur frequently, and are often triggered by external routing protocols [5].

Much effort has been devoted to optimizing the different steps of the convergence of IP routing, i.e., detection, dissemination of information and shortest path calculation, but the convergence time is still too large for applications with real time demands [6]. A key problem is that since most network failures are short lived [7], too rapid triggering of the reconvergence process can cause route flapping and increased network instability [2].

The IGP convergence process is slow because it is reactive and global. It reacts to a failure after it has happened, and it involves all the routers in the domain. In this paper we present a new scheme for handling link and node failures in IP networks. Multiple Routing Configurations (MSRC) is proactive and local, which allows recovery in the range of milliseconds. MSRC allows packet forwarding to continue over preconfigured alternative next-hops immediately after the detection of the failure. Using MSRC as a first line of defense against network

failures, the normal IP convergence process can be put on hold. This process is then initiated only as a consequence of non-transient failures. Since no global rerouting is performed, fast failure detection mechanisms like fast hellos or hardware alerts can be used to trigger MSRC without compromising network stability [8]. MSRC guarantees
recovery from any single link or node failure, which constitutes a large majority of the failures experienced in a network[7].

The main idea of MSRC 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. MSRC assumes that the network uses shortest path routing and destination based hop-by-hop forwarding.

In the literature, it is sometimes claimed that the node failure recovery implicitly addresses link failures too, as the adjacent links of the failed node can be avoided. This is true for intermediate nodes, but the destination node in a network path must be reachable if operative ("The last hop problem", [9]). MSRC solves the last hop problem by strategic assignment of link weights between the backup configurations.

MSRC has a range of attractive features:

- It gives almost continuous forwarding of packets in the case of a failure. The router that detects the failure initiates a local rerouting immediately, without communicating with the surrounding neighbors.
- MSRC helps improve network availability through sup-pression of the re-convergence process. Delaying this process is useful to address transient failures, and pays off under many scenarios [8]. Suppression of the reconvergence process is further actualized by the evidence that a large proportion of network failures is short-lived, often lasting less than a minute [7].
- MSRC uses a single mechanism to handle both link and node failures. Failures are handled locally by the detecting node, and MSRC always finds a route to the destination (if operational)
- MSRC makes no assumptions with respect to the root cause of failure, e.g., whether the packet forwarding is disrupted due to a failed link or a failed router. Regardless of this, MSRC guarantees that there exists a valid, preconfigured next-hop to the destination.

The concept of multiple routing configurations and its application to network recovery is not new. Our main inspiration has been a layer-based approach used to obtain deadlock-free and fault-tolerant routing in irregular cluster networks based on a routing strategy called Up*/Down* [13]. General packet networks are not hampered by deadlock considerations necessary in interconnection networks, and hence we generalized the concept in a technology independent manner and named it Resilient Routing Layers [14][15]. In the graph-theoretical context, RRL is based on calculating spanning sub topologies of the network, called layers. Each layer contains all nodes but only a subset of the links in the network.

The work described in this paper differs substantially from RRL in that we do not alter topologies by removing links, but rather manipulate link weights to meet goals of handling both node and link failures without needing to know the root cause of the failure. In MSRC, all links remain in the topology, but in some configurations, some links will not be selected by shortest path routing mechanisms due to high weights.

The rest of this paper is organized as follows. In Sec. II we describe the basic concepts and functionality of MSRC. An algorithm used to create the needed backup configurations is presented in Sec. III. Then, in Sec. IV, we explain how the generated configurations can be used to forward the traffic safely to its destination in case of a failure. In Sec. V, we present performance evaluations of the proposed method, and in Sec. VI, we discuss related work. Finally, in Sec. VII, we conclude and give some prospects for future work.

II. MSRC OVERVIEW

MSRC is based on using a small set of backup routing configurations, where each of them is resistant to failures of certain nodes and links. Given the original network topology, a configuration is defined as a set of associated link weights. In a configuration that is resistant to the failure of a particular node n, link weights are assigned so that traffic routed according to
this configuration is never routed through node n. The failure of node n then only affects traffic that is sent from or destined
to n. Similarly, in a configuration that is resistant to failure of a link l, traffic routed in this configuration is never routed over this link, hence no traffic routed in this configuration is lost if l fails. In MSRC, node n and link l are

called isolated in a configuration, when, as described above, no traffic routed according to this configuration is routed through n or l.

Our MSRC approach is threefold. First, we create a set of backup configurations, so that every network component is isolated in one configuration. Second, for each configuration, a standard routing algorithm like OSPF is used to calculate configuration specific shortest path trees 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. Finally, we design a forwarding process that takes advantage of the backup configurations to provide fast recovery from a component failure.

Fig. 1a illustrates a configuration where node 5 is isolated. In this configuration, the weight of the stapled links is set so high that only traffic sourced by or destined for node 5 will be routed over these links, which we denote restricted links. Node failures can be handled through blocking the node from transiting traffic. This node-blocking will normally also protect the attached links. But a link failure in the last hop of a path can obviously not be recovered by blocking the downstream node (ref. “the last hop problem”). Hence, we must make sure that, in one of the backup configurations, there exists a valid path to the last hop node, without using the failed link. A link is isolated by setting the weight to infinity, so that any other path would be selected before one including that link. Fig. 1b shows the same configuration as before, except now link 3-5 has been isolated (dotted). No traffic is routed over the isolated link in this configuration; traffic to and from node 5 can only use the restricted links.

In Fig. 1c, we see how several nodes and links can be isolated in the same configuration. In a backup configuration like this, packets will never be routed over the isolated (dotted) links, and only in the first or the last hop be routed over the restricted (dashed) links.

Some important properties of a backup configuration are worth pointing out. First, all non-isolated nodes are internally connected by a sub-graph that does not contain any isolated or restricted links. We denote this sub-graph as the backbone of the configuration. In the backup configuration shown in Fig. 1c, nodes 6, 2 and 3 with their connecting links constitute this backbone. Second, all links attached to an isolated node are either isolated or restricted, but an isolated node is always directly connected to the backbone with at least one restricted link. These are important properties of all backup configurations that are further discussed in Sec. III, where we explain how backup configurations can be constructed.

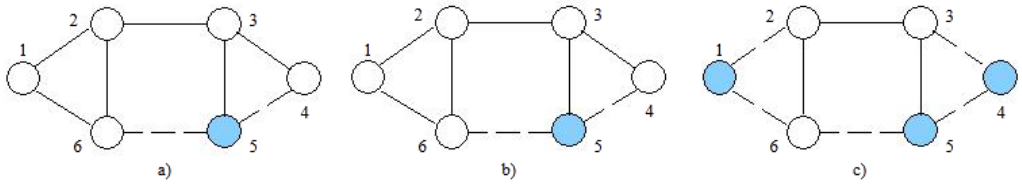

Fig. 1. a) Node 5 is isolated (shaded color) by setting a high weight on all its connected links (stapled). Only traffic to and from the isolated node will use these restricted links. b) The link from node 3 to node 5 is isolated by setting its weight to infinity, so it is never used for traffic forwarding (dotted).
c) A configuration where nodes 1, 4 and 5, and the links 1-2, 3 -5 and 4-5 are isolated.

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.

III. GENERATING BACKUP CONFIGURATIONS

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

Configuration Constraints

To guarantee single-failure tolerance and consistent routing, the backup configurations used in MSRC must adhere to the following requirements:
A node must not carry any transit traffic in the configuration where it is isolated. Still, traffic must be able to depart from and reach an isolated node.To change the default, adjust the template as follows.

A link must not carry any traffic at all in the configuration where it is isolated.

3) In each configuration, all node pairs must be connected by a path that does not pass through an isolated node or an isolated link.

4) Every node and every link must be isolated in at least one backup configuration.

5) The network topology represented by graph G must be bi-connected.

Now we propose an algorithm that can be used to automatically create such configurations. The algorithm will naturally be run once at the initial establishment of the network, and each time a node or link is permanently added or removed.

G(N,A)	Graph comprising nodes N and directed links (arcs) A .
Ci	The graph with link weights as in configuration i
Si	The set of isolated nodes in configuration Ci
Bi	The backbone in configuration Ci
A(u)	The set of links from node u
(u, v)	The directed link from node u to node v
pi(u, v)	A given shortest path between nodes u and v in Ci
N(p)	The nodes on path p
A(p)	The links on path p
Wi(u, v)	The weight of link (u, v) in configuration Ci
Wi(p)	The total weight of the links in path p in configuration Ci
Wr	The weight of a restricted link
n	The number of configurations to generate (algorithm input)

Table 1: Notation

Definition: A configuration Ci is an ordered pair (G,wi) of the graph G and a function wi : A → {1, . . . ,wmax,wr,∞} that assigns an integer weight wi(a) to each link a where a € A.

Algorithm 1: Creating backup configurations.

1) **for** i € {1 . . . n} **do**
2) Ci ← (G, w0)
3) Si ← Ø
4) Bi ← Ci
5) **end**
6) Q n ← N
7) Qa ← Ø
8) i ← 1
9) **while** Qn ≠ Ø **do**
10) u ← first (Qn)
11) j ← i
12) **repeat**
13) **if** connected (Bi \ ({u}, A (u))) **then**
14) Ctmp ← isolate (Ci, u)
15) **if** Ctmp ≠ null **then**
16) Ci ← Ctmp
17) Si ← Si Ụ{u}
18) Bi ← Bi \ ({u},A(u))
19) i ← (i mod n) + 1
20) **until** u € Si **or** i=j
21) **if** u not € Si **then**
22) Give up and abort

23) **end**

This guarantees that any other path between two nodes in the network will be chosen by a shortest path algorithm before one passing through the isolated node. Only packets sourced by or destined for the isolated node itself will traverse a restricted link with weight W, as they have no shorter path. With our current algorithm, restricted and isolated links are given the same weight in both directions in the backup configurations, i.e., we treat them as undirected links. However, this does not prevent the use of independent link weights in each direction in the default configuration.

The second requirement implies that the weight of an isolated link must be set so that traffic will never be routed over it. Such links are given infinite weight.

Given these restrictions on the link weights, we now move on to show how we can construct backup configurations that adhere to the last two requirements stated above.

IV. Recovery Load Distribution

MSRC 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. In our experience, the effect a failure has on the load distribution when MSRC is used is highly variable.

In this section, we describe an approach for minimizing the impact of the MSRC recovery process on the post failure load distribution. The algorithm presented in Sec. III creates a set of backup configurations. Based on these, a standard shortest path algorithm is used in each configuration, to calculate configuration specific forwarding tables. In this section, we describe how these forwarding tables are used to avoid a failed component.

When a packet reaches a point of failure, the node adjacent to the failure, called the detecting node, is responsible for finding the configuration where the failed component is isolated, and to forward the packet according to this configuration. With our proposal, the detecting node must find the correct configuration without knowing the root cause of failure.

A node must know in which configuration the downstream node of each of its network interfaces is isolated. Also, it must know in which configuration it is isolated itself. This information is distributed to the nodes in advance, during the configuration generation process.

V. Conclusion and Future Work

We have presented Multiple Routing Configurations as an approach to achieve fast recovery in IP networks. MSRC is based on providing the routers with additional routing configurations, allowing them to forward packets along routes that avoid a failed component. MSRC guarantees recovery from any single node or link failure in an arbitrary biconnected network. By calculating backup configurations in advance, and operating based on locally available information only, MSRC can act promptly after failure discovery.

MSRC 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 specification of a forwarding procedure that successfully solves the last hop problem.

References

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

[2] 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.

[3] Amund Kvalbein and Audun Fosselie Hansen, Multiple Routing Configurations for Fast IP Network Recovery, in IEEE, VOL. 17, NO. 2, APRIL 2009.

[4] Relaxed multiple routing configurations for IP fast reroute . Cicic, Tarik ; Hansen, A.F, Publication Year: 2008.

[5] Relaxed multiple routing configurations: IP fast reroute for single and correlated failures, Kvalbein, Amund, Publication Year: 2009.

[6] Optimality Principle Based Sink Tree Methodology for Adaptive Static Routing Using Multiple Route Configuration Scheme, Johal, Hartinder Singh, Publication Year: 2008.

[7] Post-Failure Routing Performance with Multiple Routing Configurations, Kvalbein, Amund, Publication Year: 2007

[8] On network traffic concentration and updating interval for proactive recovery method against large-scale network failures, Kamei, Satoshi.

[9] Minimum Backup Configuration-Creation Method for IP Fast Reroute ,Pelsser, Cristel, Publication Year: 2009

[10] Transport Capacity for Wireless Networks with Multi-User Links, Peel, Christian B. Wireless Communications, IEEE Transactions on Volume:11,Issue: 6, Publication Year: 2012