Volume 5, Issue 4, 2015 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Multiple Routing Configuration for Fast IP Network Recovery

[1]M. Jhansi*, M. Swathi, [3]G. Usha Rani
[1] Asst. Professor, [1, 2] Student
CSE Department, MLR Institute of Technology
India

Abstract— As the Internet takes an increasingly central role in our communications infrastructure; the slow convergence of routing protocols after a network failure becomes a growing problem. 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 strictly connectionless, and assumes only destination based hop by hop 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, 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

Keywords— MRC(Mobile Routing Configurations, P and Network traffic, communication system routing

I. INTRODUCTION

In recent years the Internet has been transformed from a special purpose network to a 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. 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 reconvergence 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 reconvergence 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 and BGP context, and has an adverse effect on real time applications. Events leading to a reconvergence have been shown to occur frequently. 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. Multiple Routing Configurations is a proactive and local protection mechanism that allows recovery in the range of milliseconds. MRC allows packet forwarding to continue over preconfigured alternative next hops immediately after the detection of the failure. Using MRC as a first line of defence 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 MRC without compromising network stability. 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.

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. The backup configurations differ from the normal routing configuration in that link weights are set so as to avoid routing traffic in certain parts of the network. We observe that if all links attached to a node are given sufficiently high link weights, traffic will never be routed through that node. The failure of that node will then only affect traffic that is sourced at or destined for the node itself. Similarly, to exclude a link (or a group of links) from taking part in the routing, we give it infinite weight. The link can then fail without any consequences for the traffic.

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. Finally, we design a for-warding process that takes advantage of the backup configurations to provide fast recovery from a component failure.

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 con-figuration that will route the traffic to its destination on a path that avoids the failed element. Also, the backup configurations must be constructed so that all nodes are reachable in all con-figurations, i.e., there is a valid path with a finite cost between each node pair.

III. LOCAL FORWARDING PROCESS

Given a sufficiently high, the algorithm presented in Section IV will create a complete set of valid backup configu-rations. 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 schema shown in Fig:1 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 a backup configuration where the failed component is isolated. The detecting node marks the packet as belonging to this configuration, and forwards the packet. From the packet marking, all transit routers identify the packet with the selected backup con-figuration, and forward it to the egress node avoiding the failed component.

Consider a situation where a packet arrives at node, and cannot be forwarded to its normal next-hop because of a component failure. The detecting node must find the correct backup configuration without knowing the root cause of failure, i.e.,whether the next-hop node or link has failed, since this information is generally unavailable.

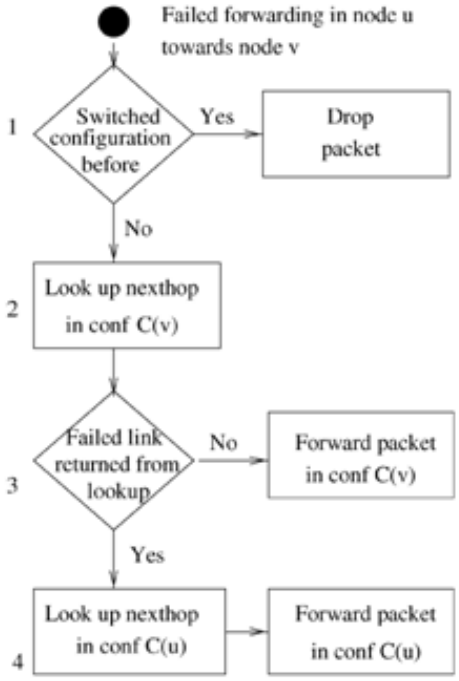

Fig1:Packet forwarding state diagram

Evaluation: MRC requires the routers to store additional routing configurations. The amount of state required in the routers is related to the number of such backup configurations. Since routing in a backup configuration is restricted, MRC will potentially give backup paths that are longer than the optimal paths. Longer backup paths will affect the total network load and also the end-to-end delay.

Full, global IGP re-convergence determines shortest paths in the network without the failed component. We use its perfor-mance as a reference point and evaluate how closely MRC can approach it. It must be noted that MRC yields the shown perfor-mance immediately after a failure, while IP re-convergence can take seconds to complete.

IV. BACKUP SETUP

Minimum number of backup configurations that Algorithm 1 could produce in a wide range of synthetic topologies. Each bar in the figure represents 100 different topologies given by the type of generation model used, the links-to-node ratio, and the number of nodes in the topology. Table I shows the minimum number of configurations Al-gorithm 1 could produce for selected real-world topologies of varying size. For the Sprint US network, we show results for both the POP-level and router level topologies.

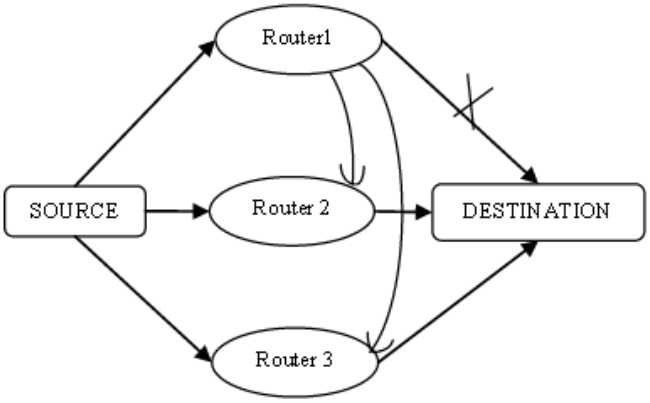

Fig 2: Backup configuration

The table also shows how many nodes that are covered by LFAs, and the number of configurations needed when MRC is used in combination with LFAs. Since some nodes and links are completely covered by LFAs, MRC needs to isolate fewer components, and hence the number of configurations decreases.

TABLE II
NUMBER OF BACKUP CONFIGURATIONS FOR SELECTED REAL-WORLD NETWORKS

Network	Nodes	Links	Confs	LFA	Confs
Sprint US (POP)	32	64	4	17	4
Sprint US (R)	284	1882	5	186	5
Geant	19	30	5	10	4
COST239	11	26	3	10	2
German Telecom	10	17	3	10	-
DFN	13	37	2	13	-

When a router detects that a neighbour can no longer bereached 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. This process is called backup configuration shown Fig 2.

Algorithm 1:

Algorithm 1: Creating backup configurations.

```
1  for i ∈ {1...n} do
2      C_i ← (G, w_0)
3      S_i ← ∅
4      B_i ← C_i
5  end
6  Q_n ← N
7  Q_a ← ∅
8  i ← 1
9  while Q_n ≠ ∅ do
10     u ← first (Q_n)
11     j ← i
12     repeat
13         if connected(B_i \ ({u}, Λ(u))) then
14             C_tmp ← isolate(C_i, u)
15             if C_tmp ≠ null then
16                 C_i ← C_tmp
17                 S_i ← S_i ∪ {u}
18                 B_i ← B_i \ ({u}, Λ(u))
19         i ← (i mod n) + 1
20     until u ∈ S_i or i=j
21     if u ∉ S_i then
22         Give up and abort
```

The number and internal structure of backup configurations in a complete set for a given topology may vary depending on the construction model. If more configurations are created, fewer links and nodes need to be isolated per configuration, giving a richer (more connected) backbone in each configuration. On the other hand, if fewer configurations are

constructed, the state requirement for the backup routing information storage is reduced. However, calculating the minimum number of con-figurations for a given topology graph is computationally demanding. One solution would be to find all valid configurations for the input consisting of the topology graph and its associated normal link weights , and then find the complete set of configurations with lowest cardinality. Finding this set would involve solving the Set Cover problem, which is known to be NP-complete [8].

IV. CONCLUSIONS

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 for-warding 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. We have evaluated the effect MRC has on the load distribution in the network while traffic is routed in the backup configurations, and we have proposed a method that minimizes the risk of congestion after a link failure if we have an estimate of the demand matrix. In the COST239 network, this approach gave a maximum link load after the worst case link failure that was even lower than after a full IGP re-convergence on the altered topology. MRC thus achieves fast recovery with a very limited performance penalty.

REFERENCES

[1] A. Basu and J. G. Riecke, "Stability issues in OSPF routing," in Proc. ACM SIGCOMM, San Diego, CA, Aug. 2001, pp. 225–236.

[2] 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.

[3] 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. 63–7

[4] M. J. O'Mahony, "Results from the COST 239 project. Ultra-high ca-pacity optical transmission networks," in *Proc. 22nd European Conf. Optical Communication (ECOC'96)*, Sep. 1996, pp. 11–14.

[5] B. Fortz and M. Thorup, "Internet traffic engineering by optimizing OSPF weights," in *Proc. IEEE INFOCOM*, 2000, pp. 519–528.

[6] D. S. Johnson, "Approximation algorithms for combinatorial prob-lems," in *Proc. 5th Annu. ACM Symp. Theory of Computing*, 1973, pp. 38–49.ˇ

[7] A. Kvalbein, T. Cicicˇ´, and S. Gjessing, "Post-failure routing per-formance with multiple routing configurations," in *Proc. IEEE INFOCOM*, May 2007, pp. 98–106.

[8] P. Pan, G. Swallow, and A. Atlas, "Fast reroute extensions to RSVP-TE for LSP tunnels," RFC 4090, May 2005.

[9] A. Raja and O. C. Ibe, "A survey of IP and multiprotocol label switching fast reroute schemes," *Comput. Netw.*, vol. 51, no. 8, pp. 1882–1907, Jun. 2007.

[10] P. Narvaez, K.-Y. Siu, and H.-Y. Tzeng, "Local restoration algorithms for link-state routing protocols," in *Proc. IEEE Int. Conf. Computer Communications and Networks (ICCCN'99)*, Oct. 1999, pp. 352–357.

[11] Z. Zhong, S. Nelakuditi, Y. Yu, S. Lee, J. Wang, and C.-N. Chuah, "Failure inferencing based fast rerouting for handling transient link and node failures," in *Proc. IEEE INFOCOM*, Mar. 2005, vol. 4, pp. 2859–2863.

