OSPF with Deterministic Routing

Sachin Bojewar
Associate Professor, Vidyalankar Institute of Technology, Wadala, Mumbai, India

Dnyaneshwar Dhangar
Vidyalankar Institute of Technology, Wadala, Mumbai, India

Abstract

Link-state routing protocols, that are commonly deployed in the Internet, tend to converge globally after any of real-time applications. topological change in the network i.e. news of the change is propagated to all the routers in the network. Such network-wide protocol convergence introduces transient inconsistency in the routing tables of the routers in the network. This inconsistency of routing tables may lead to the formation loops in the network. Study shows that such loops cause a significant packet forwarding discontinuity in the network and severely affects performance

In this paper we propose a technique to handle transient Convergence is initiated only for long-lasting failures in the network. single link failures in OSPF networks, without initiating global IP convergence process. Our technique is based on adding information in the packets traversing the failure by the routers attached to the failed link, and then locally rerouting them.

1. Introduction

The current Internet has evolved from a small network ARPANET, primarily built for research purposes, to an enormous size network consisting of thousands of Autonomous Systems (AS) operated by different institutions, such as the Internet Service Providers (ISP), companies, universities etc. Its use has also changed from being just a research-purpose network to a general-purpose network opened for commercial purposes. Such evolution of the Internet has seen a large number of applications being deployed on it for commercial purposes. Many of these applications, such as VoIP, gaming etc, have stringent delay and loss requirements. Such large-scale deployments of delay and loss-sensitive applications have led to stringent demands on stability of routing in the Internet. Stable routing demands routing stability in the event of failure or upgradation of any network component. In the event of any failure, the router adjacent to the failure has the responsibility informing every other router in the network about the failure. Other routers, in response to the failure, update their routing tables computing new routes avoiding the failure. This process, in which every router involves itself in computing the new view of the network is called *routing convergence*. Convergence has serious effects on the performance of the delay-sensitive applications mentioned above. Until every router has the same global view of the network, loops could be formed during routing. Such loops can lead to delay in routing packets or even loss of packets, resulting in serious performance degradation of the applications

2. Problem Statement

Current distributed routing paradigms (such as link-state, distance-vector, and path-vector) involve a convergence process consisting of an iterative exploration of intermediate routes triggered by certain events such as link failures. The convergence process increases router load, introduces outages and transient loops, and slows reaction to failures. We propose a new routing paradigm where the goal is not to reduce the convergence times but rather to eliminate the convergence process completely. To this end, we propose a technique called OSPF With Deterministic Routing. OSPF that allows data packets to be routed to their destination as long as a path to the destination exists in the network. Our simulation will show that: OSPF with deterministic routing using FCP can provide both low loss rate as well as low control overhead, also the overhead is very low, compared to prior work in backup path pre-computations,

3. Proposed System Architecture

Consider the topology for the formation of loops during convergence period as shown in figure 1.

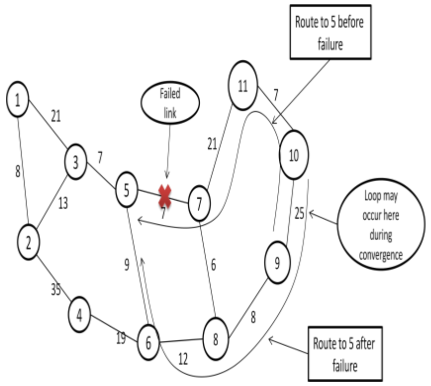


Figure 1: Routing Loops with Convergence

Global Convergence Suppression, Local Rerouting Forwarding Loops

IP convergence process, being *global*, can result in routing instability in the form of *loops*. The probability and duration of such loops depends upon the convergence period, which in turn depends upon the diameter of the network. The longer the network takes to converge, the greater the probability and duration of the loops. This in turn can lead to packet drop behaviour in the network, if Time-to-live(TTL) field of the packets in loop gets exhausted. Forwarding loops could be avoided by putting the global convergence process on hold, and instead initiating *local rerouting* at the *detecting router(s)*(routers attached to the failed link, detecting failure) after the failure. But this also does not solve the problem as loops could still ensue in the network. A straightforward local recomputation of new shortest paths by the *detecting router(s)* could result in a loop since other nodes are not aware of the failure and their routing tables do not reflect the failure. Figure 1 reproduced in figure 2 with link failure of link 8 ↔ 11 instead of 6 ↔ 9. In this Figure node 1 is the source and node 13 is the destination and path is 1 → 3 → 6 → 8 → 11 → 13. So when node 6 receives the packets it forwards them to 8. When node 8 receives the packets for 13, it tries to forward them to 11 but finds that the link 8 ↔ 11 is down. It is assumed that *detecting node* (i.e node 8) suppresses the failure advertisements and instead initiates local rerouting around the failure by computing new shortest paths.

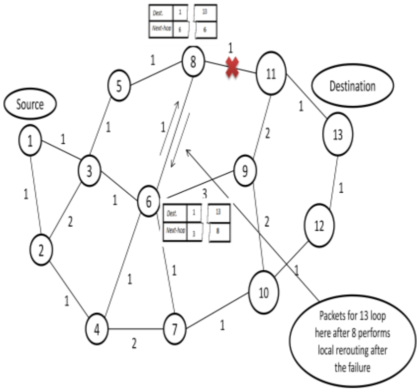
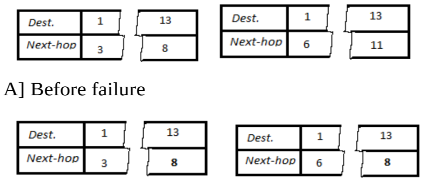


Figure 2: Local Rerouting – formation of routing loops.

Figure 3 depicts the forwarding tables of node 6 and node 8 before and after the failure. It is easily seen that the next-hop for destination 13 at node 6 before the failure was node 8 and at node 8 it was 11. After the failure, node 8 recomputes its forwarding table and changes its next-hop for 13 via 6 while 6 - still unaware of the failure - routes through 8. This leads to a loop and packets destined for 13 may eventually be dropped.



Dest.	1	13
Next-hop	3	8

Dest.	1	13
Next-hop	6	11

A] Before failure

Dest.	1	13
Next-hop	3	8

Dest.	1	13
Next-hop	6	8

B] After failure

Figure 3: Forwarding tables of node 6 and

Local Rerouting with explicit path inthe packets

If local rerouting is to work, additional information is required to be put in the packets routed around the failure by the detecting routers, so that routers further on the packets' path can safely route the packets to their destinations using this information.: The detecting router(s) not only reroute the packets traversing the failure onto the new paths, but also add the entire path information in the packets' headers. Other routers seeing this path information in packets' headers route the packets according to paths. In figure 4 Packets from node 1 destined for 13 follow the path 1 → 3 → 6 → 8 →11 → 13 before the failure. As node 6 is not aware of the failure it forwards the packet to node 8 according to its (outdated) routing table. Node 8 after receiving the packets consults its forwarding table and tries to forward them to node 11.

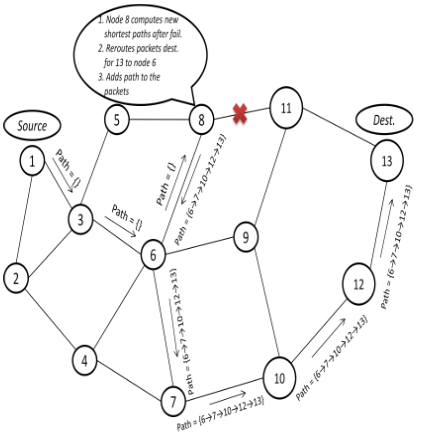


Figure 4: Local Rerouting with Explicit Path Information in the Packets.

As it finds that the link to 11 is down, it (1) computes new shortest path to the packets' destination avoiding this link, (2) adds entire route to 13 in the packets and (3) locally reroutes the packets onto the new next-hop (node 6 in this case). Node 6 receives the packets and sees the *path* information and forwards them to node 7 instead of routing them back to 8. Subsequent routers on the path also forward the packets in the same way. Finally the packets reach their destination. Only if the failure lasts longer than the transient period, that node 8 (and also 11) initiates a global convergence process.

Carrying Failed Link in the packets

Although local rerouting of packets by the detecting routers after the failure with explicit route inside packet headers does save us convergence - and the associated loops - for transient failures, it can leads to sub-optimal routing behavior. consider the situation depicted in Figure 5. Again we consider the failure of link 8 ↔ 11. Assume that node 7 has packets for node 13. The shortest path between 7 and 13 in the absence of failure is 7 ↔ 6 ↔ 8 ↔ 11 ↔ 13 i.e. the path traverses the failed link 8 ↔ 11. Therefore, as before, when packets reach 8, it reroutes them with the new path which is 6 ↔ 7 ↔ 10 ↔ 12 ↔ 13 in their headers, to their destination. As 7 is never aware of the failure, packets to 13 are always first routed to 8, which then reroutes them back to 7 onto the new path. Thus every packet loops once on the 7 ↔ 6 ↔ 8 path, before being delivered to its destination (i.e. node 13). This situation is depicted in Figure 5. This unnecessarily increases end-to-end delay of the packets to their destination. This situation occurs certainly because routers on the new path are not aware of the failure because of suppression of failure notification by the detecting routers.

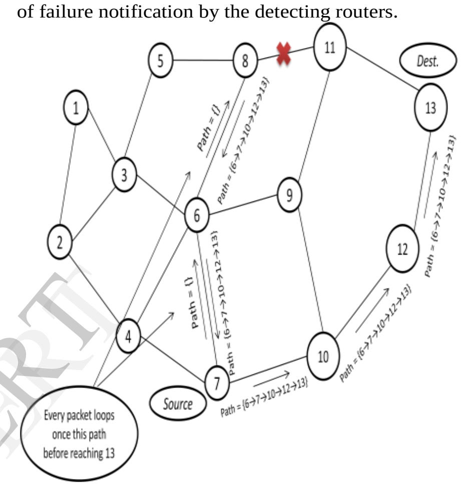


Figure 5: Packet Looping Due To Failure Suppression

To remedy this situation we need to inform *some* routers on the new path about the failure, so that they *log* the failure information and also compute new routes avoiding the failed link. But then we need to answer two fundamental questions:

1. How to inform routers on the new path about the failure? and
2. Which routers should log the failure and update their routing tables?

For the first problem we certainly cannot use the routing protocol mechanism as it will lead to convergence. So as before we carry the failure information in the packet headers. This solves our first problem. For the second problem we need to identify those routers on the packets' new route which have the failed link included in their shortest path to the packets' destination. This mechanism involves the use of SPT by a router for every packet that carries a failed link and a route its header, to see if it uses the failed link to reach the packet's destination. This certainly requires complex computations by the router and severely degrades its throughput. If a packet for destination D carrying a failed link and route in its header arrives at a router R, the router logs the failure information (and computes new routes) if previous hop of the packet is the same as the next-hop according to its forwarding table.

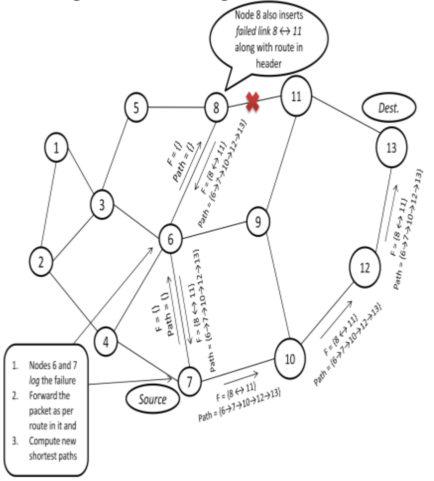


Figure 6: Carrying failed link in packet header

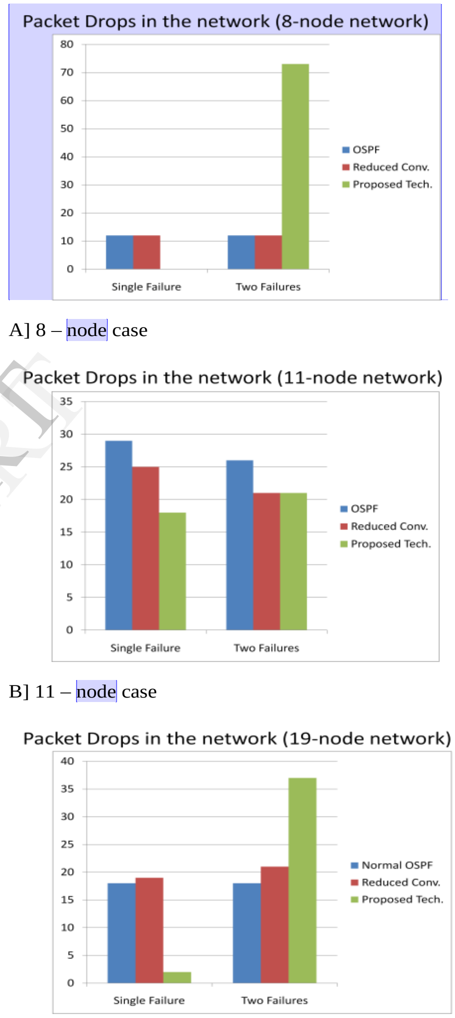
Consider again Figure 4. As before when a packet destined for 13 arrives at node 8, it reroutes the packet onto new path 7 ↔ 6 ↔ 8 ↔ 11 ↔ 13. Node 8 not only adds the route to the packet but also failed link 8 ↔ 11.Node 6 on the new path, when it receives the packet and *sees* the failed link in the header, checks from its forwarding table if next-hop router is the same as its previous hop. As it is true, it logs the failure and forwards the packet according the route in its header. Node 6 then recomputes its forwarding table to accommodate the failure. Similar action is taken by node 7 as shown in Figure 6.

4. Results

Figures 7[A]-[C] show the results As stated in the beginning the goal of our work is to reduce the packet drops occurring in the network. We explain our results in the following sections.

Single link failure

The technique performs better than OSPF terms of number of packet drops. However as compared to Optimized Convergence technique the results are almost same. Similarity of the results is due to similar link failure detection mechanism in the two. But Optimized Convergence requires convergence after the failure and our technique does not. As stated convergence leads to packet losses in the network due to loops but we did not attempt to simulate formation of routing loops during convergence and consequent packet drops. Thus results do not capture those drops and hence the two techniques show similar results.



A] 8 – node case

B] 11 – node case

C] 19 – node case

Figure 4: Results

Two single link failures

Results for these cases are shown in figure7 [A]-[C]. As seen the results are negative for both Optimized Convergence and Proposed technique. Further investigation is required for such unexpected performance degradation. This also may be due to implementation limitation as both techniques require fast failure detection which we currently did not attempt in our implementation.

5. Conclusion and Future Work

We proposed our protocol as an extension to OSPF to reduce packet drops in the network due to single link failures. The proposed protocol differs from OSPF in that unlike OSPF, it does attempt convergence after a link failure in the network. The idea that we propose is simple: instead of relying on convergence to handle single link failures, the routers attached to the failed link compute new routes avoiding failure and add complete route in the packets received by them. Thus any router receiving those packets can safely route the packets according to the path inserted. The routers attached to the failure also add the failed link the packet's header so that any router on the new path that uses the failed link in its SPT to reach packet's destination also updates its routing table to reflect the failure. As seen from the results our technique achieves lower packet drops as compared to OSPF in single link failure case but this comes at the cost of extra overhead due to route information added to the packets. Also more than one single link failures in the network show degradation in performance with more packet drops. As said earlier further investigations are required to detect the reason for such a behavior. Also enhancements are required to reduce the overhead due path information in the packets. This forms the future work of our proposal.

6. References

[1] G. Iannaccone, C. Chuah, R. Mortier, S. Bhattacharya, and C. Diot, "Analysis of link failures in an IP backbone," in *IMW*, Nov. 2002.

[2] A. Markopulu, G. Iannaccone, S. Bhattacharya, C. Chuah and C. Diot, "Characterization of failures in an IP backbone," in *Proc. IEEE INFOCOM*, Mar. 2004.

[3] P Francois and O. Bonaventure,"Avoiding transient loops during IGP convergence in IP networks," in *Proc. INFOCOM*, 2005.

[4] C. Alaettinoglu, V. Jacobson, and H. Yu, "Towards Millisecond IGP Convergence", IETF Internet draft 2000.

[5] A. Kvalbein, A. F. Hansen, T. Cicic, S. Gjessing, and O. Lysne, "Fast IP Network Recovery using Mumtiple Routing Configurations",

[6] K. K. Lakshminarayanan, M. C. Caesar, M. Rangan, T. Anderson,S. Shenker, and I. Stoica, "Achieving Convergence-Free Routing using Failure-Carrying Packets," In SIGCOMM, 2007.

[7] S. Rai, B. Mukherji, 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.

[8] D. Medhi, and K. Ramasamy, *Network Routing: Algorithms, Protocols, and Architectures*, Morgan Kaufman, 2007.

[9] J. T. Moy, *OSPF Anatomy of an Internet Routing Protocol*, Pearson Education, 1998.