ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 3, Issue 2, pp: (62-66), Month: April - June 2015, Available at: www.researchpublish.com

An Efficient Fast Re-Route Method

[1]Rushabh Mehta, [2]Kartik Dhumale, [3]Sakshi Shah, [4]Deepa Gaikwad, [5]Prof S.B.Ware

Sinhgad Institute of Technology (Information Technology), India

Abstract: **Today the world is rapidly changing and technology is zooming exponentially, the number of users of internet is also increasing rapidly which cause increase in traffic and causes more no of failures in transmission of packets. The backbone process of the global internet or any internal network is Routing. Existing routing techniques includes static and dynamic routing which are implemented using routing algorithms like RIP, EIGRP, OSPF, IS-IS. These routing algorithms intelligently route the packets on the network. There are many scenarios where a packet is sent, the routing algorithm finds an appropriate path for the packet, routes it and it is delivered. There are cases where packets do not get delivered because of many reasons like link failures. A method to find an alternate path, after a link failure, from a source node to a destination node, before the Interior Gateway Protocol (e.g., OSPF or IS-IS) reroutes the packet by notifying the sender which takes about 100ms. A smarter way to deliver the packet is to reroute the packets from the last successful node. In this method an alternate route is found out instantly after a node failure, and updating the routing table of other nodes according to the new route. The target application (up to tens of nodes) accesses the sub-network of a service provider's network, which is a typical scale, encountered in practice; a service provider typically has many such small regional access networks. The fast reroute method would establish a new path from the source to destination in a much lesser time than the existing system i.e. IGP (OSPF).**

Keywords: **OSPF, Re-Routing, Routing, IS-IS, Reconvergence.**

I. INTRODUCTION

In this paper we find alternate method for finding an alternate path during link failure. The current FRR's do not provide an adequate mechanism for technique to re-route packets of link failure. To overcome this a FRR is needed that calculates shortest paths instantaneously.

II. BACKGROUND

Fast Re-Route provides redundancy or an LSP path. When you enable Fast Re-Route, detours are precomputed and pre-established along the LSP. In case of a network failure on the current LSP path, traffic is quickly routed to one of the detours. Fig 1 illustrates an LSP from router A to router E, showing the established detours. Each detour is established by an upstream node to avoid the link toward the immediate downstream node and the immediate downstream node itself.

Each detour might traverse through one or more label-switched routers that are not shown in the figure

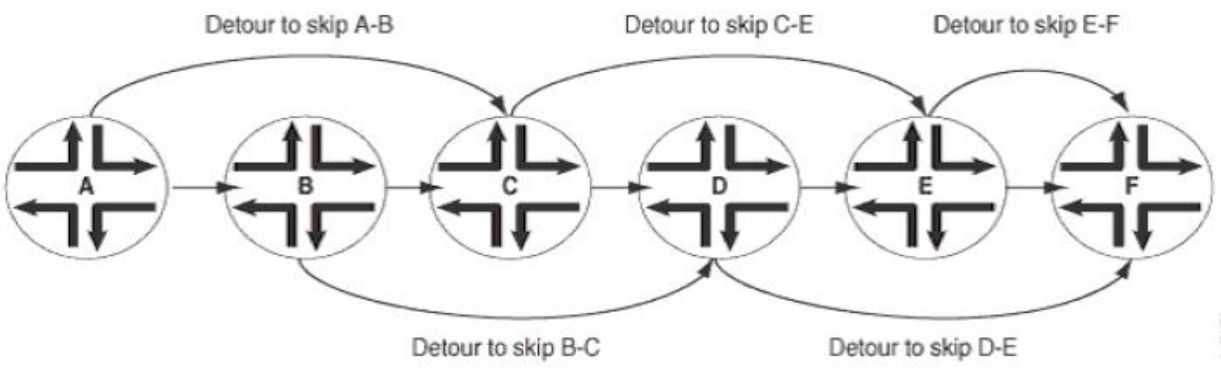


Figure 1: Detours Established for an LSP Using Fast

III. NODE DESCRIPTION

A. Server: This type of node is has the capability of sending packets into the network.
B. Sink: This type of node generally is the destination in the network,
C. Node: This is the normal node in the network which maintains all information of a node like routing table, routing data, etc.
D. Attacker: This is the type of node which can cause link failure in the network enabling the user to fast re-route the packets in the metwork.

IV. TOPOLOGY USED

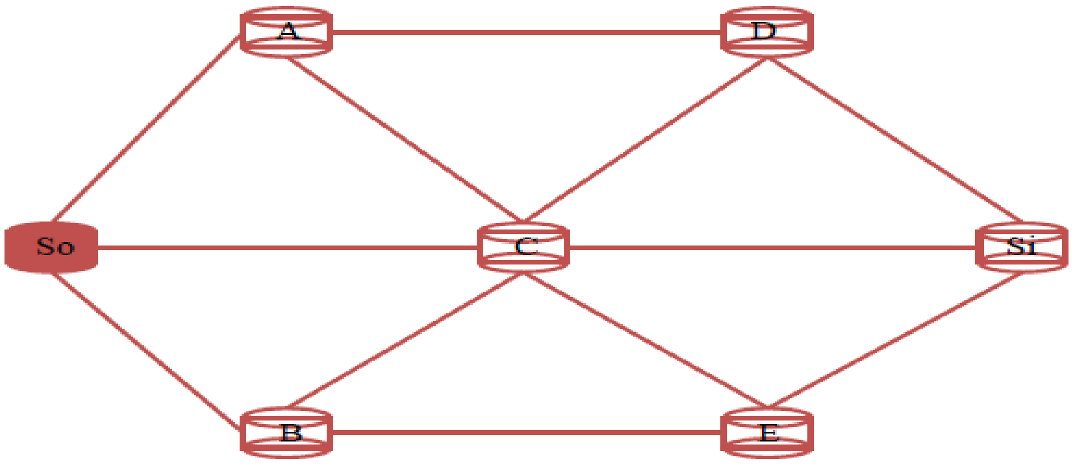


Fig. 2 Topology

- **So:** Source Node
- **A:** Node A
- **B:** Node B
- **C:** Node C
- **D:** Node D
- **E:** Node E also attacker node
- **Si:** Sink Node

V. THE METHOD

Consider *G=(N,A)* be an undirected connected graph with *N* as the node set and *A* as the set of arcs. For *x*∈*N, let N(x)* be the set of neighbours of *x*, where a neighbour of x is a node one arc away from *x*. We associate with each undirected arc(i,j) ∈*A*a cost *C(i,j),* and require each *C(i,j)* to be positive integer. For *i,j N*, let *C'(i, j)* be the cost of the shortest path G between *i* and *j*. We use the method *Route(s,d)* ,where is *s* is *the* source node and *d* is the destination node, for the fast re-route in the case of link failure to avoid the reconvergence . In procedure *Route(s,d)*, *P* is the ordered list of nodes that have been visited, and *P* <*{P, x}* means that *x* is inserted after the rightmost element in *P*. Also, *Δ(n)* is the multiplicity of node n, indicating how many times n has been visited by the current packet.

Procedure *Route(s,d)*

```
1. Initialize: $P=Ø$, $Δ(n)=0$ for $n∈ N$, and $x=s$;

2. While $(x=d)${

    2.1.1. Let $P=\{y∈N(x) \mid Δ(y)=min n∈N(x) Δ(n) \}$
    2.1.2. Pick any $y∈Y$ for which he sum $c(x,y) + c'(y,d)$ is smallest;
    2.1.3. Set $Δ(x) <Δ(x)+1$, $P <\{P,x\}$, and send the packet $P$ from $x$ to $y$;
    2.1.4. Set $x<y$;
    2.1.5  }
```

In words, if *x* is the latest node to receive the packet. We find the set of neighbours of *x* with the lowest multiplicity. From this set, we pick the neighbour *y* for which C(x, y) + C'(x, y) is smallest. We add *x* to*P*, augment the multiplicity of *x* by 1, and send the packet and *P* and *y*.

Note: *P* can be used to compute the multiplicities.

Shortest Path:

Given a vertex, say vertex (that is, a source), this section describes the shortest path algorithm.
The general algorithm is:

1. Initialize the array smallest Weight so that smallest Weight[u] = weights[vertex, u].
2. Set smallestWeight[vertex] = 0.
3. Find the vertex, v, that is closest to vertex for which the shortest path has not been determined.
4. Mark v as the (next) vertex for which the smallest weight is found.
5. For each vertex w in G, such that the shortest path from vertex to w has not been determined and an edge (v, w) exists, if the weight of the path to w via v is smaller than its current weight, update the weight of w to the weight of v + the weight of the edge (v, w).

Because there are n vertices, repeat Steps 3 through 5, n – 1 times.
Consider the following example to find out the shortest path from source (Node A) to destination (Node D)

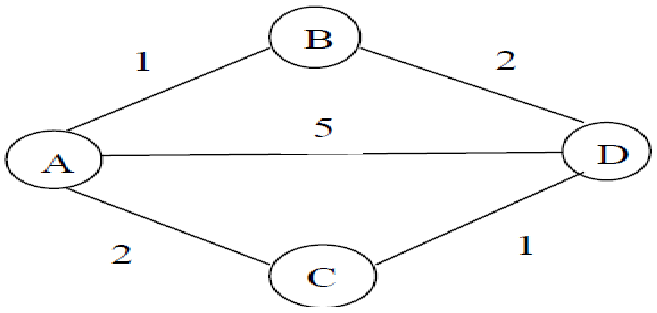


Table. 1

Edge	Cost	Path
B	1	A-B
C	2	A-C
D	5	A-D

Source: A

Direct Cost :Select A-B

Table. 2

Edge	Cost	Path
B	1	A-B
C	2	A-C
D	3	A-B-D

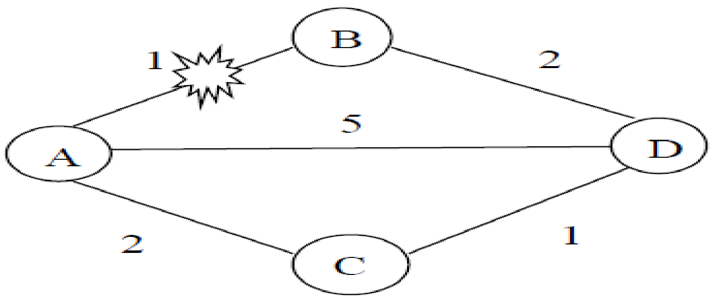
Therefore ,
A-B-D (3)<A-D(5)
Adjusted from B
The final path is A-B-D
Select A-C

Table. 3

Edge	Cost	Path
C	2	A-C
D	1	C-D
D	3	A-C-D

The optional path is A-C-D.

Hence, we have two paths to reach the destination node D from source node A.
But, suppose that the path select is A-B-D and the link between A-B drops or fails. In that case, in Fast re-route system the packet will be sent back to the previous node (here, node A) and will be routed from there to the destination from the other path (here, A-C).



So, here we will choose the path A-C-D to send.

Table. 4

Edge	Cost	Path
B	2	A-C
C	1	C-D
D	3	A-C-D

VI. FUTURE SCOPE

Importantly, we plan to use more than one Routing Protocol’s like IS-IS to split the rerouted traffic for load balancing. More importantly, we plan to enhance our algorithm to detect multiple link and node failures. In such cases we should avoid fast rerouting and rely on the original IGP convergence mechanism.

VII. CONCLUSION

We presented various network designs and backup pat reconfiguration schemes for restoring multiple failures using FRR. These are the first designs with formal (worst case) guarantees of no congestion and no loss of connectivity. Although our proofs of correctness are complex, the descriptions of the designs and protocols are simple. Using a lower bound argument, we showed that one of our constructions is nearly optimal in the number of edges. A numerical evaluation showed that, e.g., a network graph with average degree 7 with 57% overbuild can support failures of up to 4 edges.

REFERENCES

[1] Rosenberg, E.; Uttaro, J., "A Fast Re-Route Method," *Communications Letters, IEEE* , vol.17, no.8, pp.1656,1659, August 2013

[2] www.juniper.net%2Ftechpubs%2Fen_US%2Fjunos14.2%2Ftopics%2Fconcept%2Fmpls-fast-reroute-overview.html&ei=pdoPVdnCCsvJuASlu4HIAw&usg=AFQjCNEZplwOVO6WOWLC9XnK8bCicfJnSg&bvm=bv.88528373,d.c2E

[3] Yong Liu, Dept. of ELEN, TAMU, College Station, TX, USA, Reddy, A.L.N., A fast rerouting scheme for OSPF/IS-IS networks, Computer Communications and Networks, 2004. ICCCN 2004. Proceedings. 13th International Conference on 11-13 Oct. 2004