BYPASS THE LOOPS IN ENORMOUS NETWORKS

D Kishore Kumar
Assistant Professor, Department of Information Technology, GITAM University

ABSTRACT: ***In general comprehensive networks have chance of frequent failures; this can be failures of links between the routers when they communicate for the routing table which results in loop occurrence. Routers have the terrible feature of updating their routing tables where its convergence time depends on its routing protocols. Here in this paper, a new advanced attempt is made that saves the convergence time as well as the packet losses. The mechanism is Refreshing with Fast Merging (RFM) which regularly amends the forwarding routing tables with very less time. Here we have taken a complicated network of routers for stimulation later there occurs failures in links. These loops can be evaded and shown by constructing a shortest path tree.***

KEYWORDS: ***Refreshing with Fast Merging (RFM), Shortest Path Tree, Link failures***.

INTRODUCITON: Earlier Research networks are Inter network Protocol networks which carried best effort packets where the link state Intra domain routing protocols were used in enormous networks [1]. All these protocols are used in LSP's. Internet service providers mainly obtain links failures as key problem.

Temporary loops when link fails can be occurred due to change in topology. Points of presence used network Point - to -point link and Local Area Network point to point link Internet gateway Protocol will gather as soon as possible. When the link is not protected locally.

Internet Gateway Provider metrics are the source of changes in Internet Protocol networks. When there is increase in sudden traffic then this Internet gateway provides metric come in to play. A router sometimes facesinternal failures which are the means of software update. The routers keep on updating their forwarding information base the resources are kept up until this job is done and links to forwarded packet will not come in to influence here. When a router is failed then the packets reaching to this router will be deviated to the adjacent router through fast Reroute Technique to a node. So that finally packets reach safely to destination.

OUR APPROACH: University studies have proved that in a large backbone networks routers frequently fail their links. The Internet has been very important personal assistance to every person and also the users are increasing day by day as well as the services are more depended on the Internet. In provide 24*7 reliability to the customers Internet Service Providers should take care of the failures. The routers need to be quickly updated about their adjacent routers in case of failures. So that forwarding packets will not affect the destination. OSPF which is widely used routing protocol for the link advertisements in response to charge in topology which again results in new routing table. This will happen nationwide sometimes which also results in delay in traffic sometimes packets may also be dropped [1].
RFM [3] will mainly achieve two goals
1) evade loops andforward 2) optimal time delay.
RFM prevents when packets losses due to lack of valid routes.
Optimal time delay can be achieved when forwarded packets move along with shortest path as well as network is ready for any change. The disadvantages in this technique results that each packet should also maintain cost of the remaining path to move to the destination. This results in larger byte to the header.

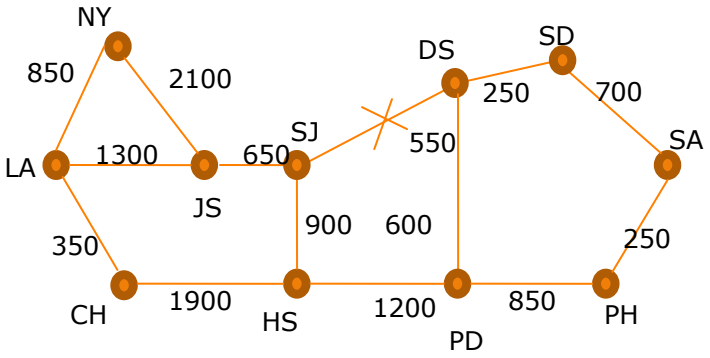
HANDLING LOOPS: During Reverse Shortest path Tree calculations, all the routers independently maintain its waiting list with respect to down link. The refreshing routing table of the Router updated its Forwarding Information Base for a destination.Selecting the outgoing interfaces for destination when updating its Forwarding Information Base according to its new topology is considered for the removal or the metric increase of all the affected links. Completion messages means of a link sent by the router in the form of FIB for all the final destinations before the event [4].If has not updated its FIB for destination, it cannot have sent a completion message for any of the failing links that it uses to reach. The failing links that a router on uses to reach are used by to reach, so that cannot have received all the necessary completion messages for any of those links. In other words, did not send a completion

message for the links that it uses to reach. Thus, locks the FIB update for those links long its paths towards them. We provide the pseudo code that implements the ordering with completion messages. A router will compute the reverse shortest path to process the metric increase of a set of link tooted on each link belonging to, that it uses in its current, outdated shortest path tree. The ranking is associated with it [5]. This depends on the next hop or the router to record the path in the list. Neighbors receive completion messages with the link. When the rank is zero with the help of the given link then the forward information base is updated directly for the final destinations via link, and lately sends the completion messages to the corresponding next hop or the router.

When there occur waiting list for the process then it reaches neighbors and starts considerations for the rank via the link. When the waiting list is empty then the times collapses for the update of FIB

```
for each Link X→ Y Є S do
        if X→Y ЄSPTold(R) then
        Link RSPT= rSPT (X → Y);
        LinkRank = depth(R, Link RSPT );
        I(X → Y) = Nexthops (R, X → Y);
        ifLinkRank == 0 then
        foreach d: X → Y ЄPathold(R , d) do
        UpdateFIB(d);
        end
        foreach N Є I(X → Y) do
                send(N, CM(X → Y));
        end
        end
        else
        WatingList(X → Y)=Childs(R,LinkRSPT);
        StartTimer(X → Y, LinkRank * MAXFIBTIME);
        end
end
end
        Upon reception of CM(X → Y) from Neighbor N:
        WatingList(X → Y).remove(N);
        Upon (WaitingList(X → Y).becomesEmpty()
        Timer(X → Y).hasExpride());
        foreach d: X → Y Є Path(R,d) do
        UpdateFIB(d);
        end
        foreach N Є I(X → Y) do
        send(N,CM(X → Y));
        end
        Pseudo code for Avoiding Link Failures\
```

We consider a network to explain how to avoid the transient loops occur in the network by converging link state routing protocol. The American cities are connected in this network like San Jose (SJ),San Antonio (SA), Los Angeles (LA), New York (NY), Jack Sonville (JS), Chicago (CH), Houston (HS), Dallas (DS), San Diego (SD), Phoenix (PH), Philadelphia (PD)



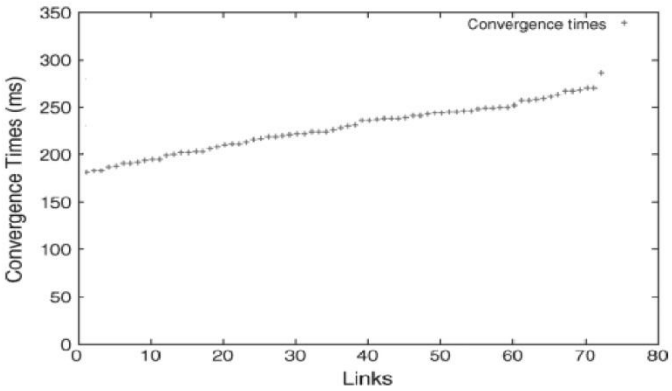
Example: Internet topology with Internet Gateway Protocol Costs

To understand this problem, let us consider the Internet2/Abilene backbone.1 Fig. 1 shows the IGP topology of this network. Assume that the link between SJandDS fails but was protected by an MPLS tunnel between DSandSJ via PDandHS. When PD receives a packet with destination JS, it forwards it to DS, which forwards it back to PD, but inside the protection tunnel, so that SJ will DE capsulate the packet, and forwards it to its destination, JS.

This suboptimal routing should not last long, and thus after a while the routers must converge, i.e., adapt to the new shortest paths inside the network, and remove the tunnel. As the link is protected, the reach ability of the destinations is still ensured and thus the adaptation to the topological change should be done by avoiding transient loops rather than by urging the updates on each router. The new LSP generated by DS indicates that DS is now only connected to SDandPD. Before the failure, the shortest path from PH to SJ, JS, NYandLA was via SA, SDandDS. After the failure, SA will send its packets to SJ, JS, NYandLA via PH, PDandHS. During the IGP convergence following the failure of link SJ−DS, transient loops may occur between SAandPH depending on the order of the forwarding table updates performed by the routers. If SA updates its FIB before PH, the packets sent by SA to SJ via PH will loop on the PH-SA link. To avoid causing a transient loop between PHandSA, PH should update its FIB before SA for this particular failure. A detailed analysis of the Internet2 topology shows that transient routing loops may occur during the failure of most links, except NY−JSandNY−LA. The duration of each loop will depend on how and when the FIB of each router is updated. Measurements on commercial routers have shown that updating the FIB may require several hundred of milliseconds. Transient routing loops of hundred milliseconds or more are thus possible and have been measured in real networks. As shown with the simple example above, the transient routing loops depend on the ordering of the updates of the FIBs. In the remainder of this paper,. This proof is constructive as we give an algorithm that routers can apply to compute the ranks that let them respect the proposed ordering.

CONVERGENCE TIMES IN ISP NETWORKS: In this section, we analyze by simulations the convergence time of the proposed technique, in the case of a link down event. The results obtained for link up events are very similar. Indeed, the updates that are performed in the FIB of each router for the shutdown of a link impact the same prefixes for the linkup of the link. The only difference in the case of a link up is that the routers do not need to compute a reverse Shortest Path Tree. As no packet are lost during the convergence process.

Lsp_process_delay	[2,4]ms
Update_hold_down	180ms
rspt_computation_tome	[3,5]ms
Completion_message_process_delay	[2,4]ms
Completion_message_sending_delay	[2,4]ms

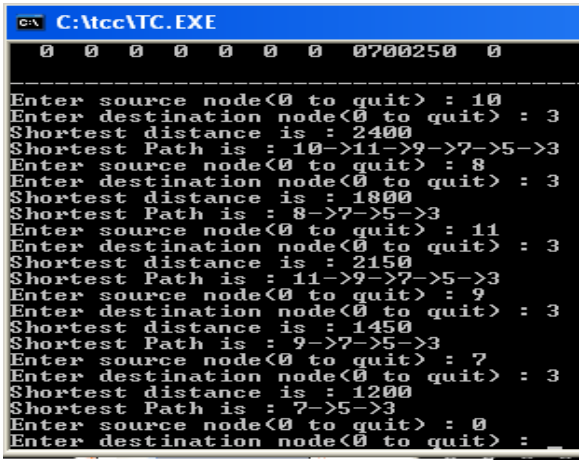


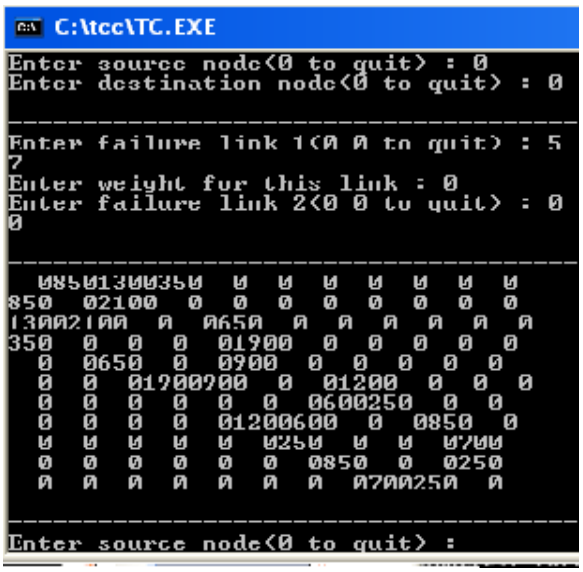
We cannot define the convergence time as the time required bringing the network back to a consistent forwarding state, as it would always be equal to zero. What is interesting to evaluate here is the time required by the mechanism to update the FIB of all the routers by respecting the ordering.

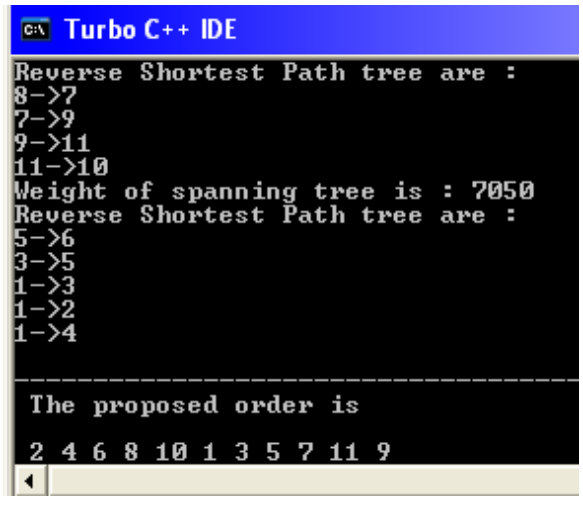
EXPERIMENTAL RESULTS:

```
C:\tcc\TC.EXE
Enter number of routers : 11
-----------------------------------
Enter link 1(0 0 to quit) : 1
2
Enter weight for this link : 850
Enter link 2(0 0 to quit) : 1
3
Enter weight for this link : 1300
Enter link 3(0 0 to quit) : 1
4
Enter weight for this link : 350
Enter link 4(0 0 to quit) : 2
3
Enter weight for this link : 2100
Enter link 5(0 0 to quit) : 3
5
Enter weight for this link : 650
Enter link 6(0 0 to quit) : 4
6
Enter weight for this link : 1900
```

```
C:\tcc\TC.EXE
Enter link 13(0 0 to quit) : 9
11
Enter weight for this link : 700
Enter link 14(0 0 to quit) : 10
11
Enter weight for this link : 250
Enter link 15(0 0 to quit) : 0
0
-----------------------------------
The adjacency matrix is :
  08501300350   0   0   0   0   0   0   0
850   02100   0   0   0   0   0   0   0
13002100   0   0650   0   0   0   0   0   0
350   0   0   0   01900   0   0   0   0   0
  0   0650   0   0900550   0   0   0   0
  0   0   01900900   0   01200   0   0   0
  0   0   0   0550   0   0600250   0   0
  0   0   0   0   01200600   0   0850   0
  0   0   0   0   0   0250   0   0   0700
  0   0   0   0   0   0   0850   0   0250
  0   0   0   0   0   0   0   0700250   0
-----------------------------------
Enter source node(0 to quit) : _
```


C:\tcc\TC.EXE
0 0 0 0 0 0 0 0700250 0
Enter source node(0 to quit) : 10
Enter destination node(0 to quit) : 3
Shortest distance is : 2400
Shortest Path is : 10->11->9->7->5->3
Enter source node(0 to quit) : 8
Enter destination node(0 to quit) : 3
Shortest distance is : 1800
Shortest Path is : 8->7->5->3
Enter source node(0 to quit) : 11
Enter destination node(0 to quit) : 3
Shortest distance is : 2150
Shortest Path is : 11->9->7->5->3
Enter source node(0 to quit) : 9
Enter destination node(0 to quit) : 3
Shortest distance is : 1450
Shortest Path is : 9->7->5->3
Enter source node(0 to quit) : 7
Enter destination node(0 to quit) : 3
Shortest distance is : 1200
Shortest Path is : 7->5->3
Enter source node(0 to quit) : 0
Enter destination node(0 to quit) :


C:\tcc\TC.EXE
Enter source node(0 to quit) : 0
Enter destination node(0 to quit) : 0
Enter failure link 1(0 0 to quit) : 5
7
Enter weight for this link : 0
Enter failure link 2(0 0 to quit) : 0
0
0 0850 1300350 0 0 0 0 0 0 0
850 02100 0 0 0 0 0 0 0 0
1300210 0 0650 0 0 0 0 0 0
350 0 0 0 01900 0 0 0 0 0
0 0650 0 0900 0 0 0 0 0
0 0 01200700 0 01200 0 0 0
0 0 0 0 0 0 0600250 0 0
0 0 0 0 01200600 0 0850 0
0 0 0 0 0 0250 0 0 0700
0 0 0 0 0 0 0850 0 0250
0 0 0 0 0 0 0 0700250 0
Enter source node(0 to quit) :


Turbo C++ IDE
Reverse Shortest Path tree are :
8->7
7->9
9->11
11->10
Weight of spanning tree is : 7050
Reverse Shortest Path tree are :
5->6
3->5
1->3
1->2
1->4
The proposed order is
2 4 6 8 10 1 3 5 7 11 9

CONCLUSION:

Enormous networks consist of many topologies described above. Router keeps updating their routing table with a Forward Information Base Technique. Updates sometimes causes looped in order to evade them we came up with a technique called RFM (Refreshing and Fast Merging) where packet losses can also be evaded. This is most common in enormous networks. We have proposed aRefresh applicable for the failures of protected links and the increase of a link metric and another ordering for the establishment of a new link or the decrease of a link metric. We also came up with Experimental Results which shows our approach for optimizing this problem.

REFERENCES

[1] P. Francois and O. Bonaventure, "Avoiding transient loops during IGP Convergence in IP Networks," in *Proc.IEEE INFOCOM*, March 2005.

[2] ISO, "Intermediate system to intermediate system routing information exchange protocol for use in conjunction with the protocol for providing the connectionless-mode network service (iso 8473)," ISO/IEC, Tech. Rep. 10589:2002, April 2002.

[3] P. Pan, G. Swallow, and A. Atlas, "Fast Reroute Extensions to RSVP-TE for LSP Tunnels," May 2005, Internet RFC 4090.

[4] M. Shandand S. Bryant, "IP Fast Reroute Framework," October 2006,

[5] A. Shaikh, R. Dube, and A. Varma, "Avoiding Instability during Graceful Shutdown of OSPF," in *Proc. IEEE*

[6] C. Alaettinoglu, V. Jacobson, and H. Yu, "Towards millisecond IGP congergence," November 2000, internet draft, draft-alaettinoglu, ISISconvergence-00.

[7] M. Shand and S. Bryant, "IP Fast Reroute Framework," October 2006.

[8] A. Shaikh, R. Dube, and A. Varma, "Avoiding Instability during Graceful Shutdown of OSPF," in Proc. IEEE.