International Journal of Networks and Communications 2014, 4(1): 1-11
DOI: 10.5923/j.ijnc.20140401.01

An Efficient Pre-computed Backup Path on the IGP Network Communication

Radwan S. Abujassar[1,*], Ma'moun Khalid[1], Mohammed Ghanbari[2,3]

[1]School of Information Technology, MEU University, Jordan
[2]School of Electrical and Computer Engineering, University of Tehran, Iran, Tehran
[3]School of Computer Science and Electronic Engineering, University of Essex

Abstract Currently, data communication during heavy traffic transmission on the network suffers from node failures. A failure in the network is required to be restored by the routing protocols in the networks. Traditional routing protocols schemes normally compute a routing table which contains all paths between all nodes on the network. Hence, the data packets will be passed via the single shortest path which is the best path between each source and destination. In this paper, a pre-computed alternate path is introduced to assist the congested networks to continue passing the data packets from its source to the final destination once failure occurs. The proposed alternative routing table (ART) algorithm aims to re-route the traffic through a backup route when the primary path has failed. We have evaluated the performance of the proposed scheme with OSPF routing protocol through NS2 simulator. The results show that packet losses, rerouting and end to end delay times of the proposed methods are substantially improved.

Keywords Component, Link state, ART (Alternative Routing Table.)

1. Introduction

Network communication continues to increase and thus the system is required to tolerate large volumes of traffic with respect to huge capacity of links. Network communication is affected by frequent failures and this leads to find an efficient recovery mechanism. In current networks, failures occur frequently, which will affect the stability of the network. When there is a link or a node failure, the node that is connected to that failure needs to re-compute its routing table and propagate the updates to all nodes concerned with this failure. Recovery mechanism has a motivation that concerns two cases. First, the time required to detect failure.

Table 1. Components of the Failure Restoration Time [1]

Timer	Default Value	Minimum Value
Notification timer	2s	10ms
Link state Packet (LSP) generation timer	50ms	1ms
Shortest path computation timer	5.5s	1ms
Processing phase Typical values	**Processing phase Typical values**	
LSP processing	10ms/hop	
SPF computation	100 - 400 ms	
Forwarding information update	20 entries / ms	

* Corresponding author:
radwanbr@hotmail.com (Radwan S. Abujassar)
Published online at http://journal.sapub.org/ijnc

Second, the time to compute a new shortest path that takes roughly 70ms. However, the slow convergence time of the routing protocol for the network when failure occurs has inducted to find an optimal solution to carry all traffic to route from an alternative path until the routing protocol updates the routing table and re-computes a new shortest path. Alternative Routing Table (ART) algorithm aims to recover the network from failure during a short period of time. Precisely, when link or node goes down, it aims to reduce delays and improve throughput in the network. Hence, the pre-computed alternative path can be used when link or node fails on the primary path without waiting for the routing protocol to re-compute a new shortest path [12].

The Open Shortest Path First (OSPF) routing protocol is used as a dynamic link state protocol for TCP/IP or UDP traffic and is designed to update the information for topology by sending a Link State Advertisement (LSA) based on the presence of a failure. The convergence time of the recovery mechanism is still too large for the real time applications. The convergence time can be of the order of 100's of millisecond or even 10's of seconds in the Border Gateway Protocol (BGP) networks. Table 1 lists the default and minimum times for the routing protocol to re-compute a new shortest path. Hence, during the process, while the routing protocol is converging micro-loops may be created. The affection of this can lead to increase loss of packets and end to end delay in the applications such as video or VOIP traffic; because the source will keep sending packets until it receives a notification that a failure has occurred. In this paper, we

concentrate on the original routing table, which is computed by the Link State Protocol. We propose a new algorithm to create a new backup routing table with excluding all primary paths in the original routing table. The main contribution of this paper is an alternative and a fully disjoint path computed by using the original routing table, which guarantees that the backup path does not join with any node or link on the primary path between source and destination to avoid a loop in the network.

2. Related Work

An efficient routing protocol algorithm has been built for achieving robustness and fast convergence within a short time in case of failure. In [7] authers address classifications of failure. The result shows that 80% of all failures are unplanned. According to this finding, the 70% of failure is due to the effect of a single link failure at a time, and the remaining is due to the effect of a shared link risk groups. The protection schema is a proactive mechanism, which calculates backup routes in advance while the restoration schema is reactive by calculating the backup routes when failure has been detected [2, 10]. The restoration schema considers more flexibility with regard to the location of failures. The disjoint path between source and destination considers the best solution to recover the network from failure, which is guaranteed to pass the traffic through it to the destination with reduced loss of packets [15]. In [5, 4] the authors discuss the cost of the links in the network, which is considered to be an important parameter in determining the best path through the routing protocol algorithm. The minimum path cost will be determined by comparing it with other candidate paths. There are two kinds of the Dijkstra algorithms. Firstly, there is a Dijkstra algorithm to compute the best path by removing the links with bandwidths less than a threshold. Secondly, there is an on demand Dijkstra algorithm, which generates the shortest path tree to a precomputed node [3]. The node will be added to the tree depending on the requested bandwidth [16]. In [11, 14] the authors proposed a new mechanism, termed Failure Insensitive Technique (FIR). FIR uses the specific forwarding interface to provide a backup next hop with loop free. The FIR mechanism makes the node, which is connected to the failure to add a new header by re-encapsulating the packets and then re-sending them to the adjacent nodes to inform them about the fault through the interface packets that arrive. Hence, based on the interface packets when failure occurred, the adjacent node will reroute the affected packets and the other nodes which will not know about the failure by sending packets according to pre-computed routing tables. FIR has many drawbacks as follows: the encapsulation of packets is not desirable because that will reduce the throughput and make the end-to-end delay longer. In addition, FIR cannot provide protection against node failure. The Internet Protocol Fast Re-Route (IPFRR) is an applicable technique. It includes the LFA, U-turn and not-via address [9, 13]. The drawback to the IPFRR technique is that loop free is not guaranteed because the packet can be returned to the source with regard to a specific forwarding pre-computed routing table for each node on the network. In addition, not-via address needs to encapsulate/ de-capsulate packet, which affects network performance [8].

3. ART Technique

While most proposed solutions intend to reduce recovery time through many technique. The efficient routing protocol algorithms have been built for achieving robustness and fast convergence within a short time in case of link or node failure [7, 10, 2, 15]. The results show that 80% of all failures are unplanned, as shown in table 1. According to that, 70% of failures affect a single link at a time, and the remaining percentage is affected by shared link risk groups. The work in this paper provides a new technique based on ART algorithm. In ART algorithm, the computed disjoint path between source and destination considers the best solution to recover the network from node or link failure, regardless of the location of the failure, and also disjoint path is guaranteed to be local loop free in the network. A node in the network may have many interfaces, therefore when a node fails, all its interfaces will fail simultaneously, and loss of signals will occur. Hence, the disjoint path can protect data packets from dropping and deliver them safely to the destination, improving end-to-end delay. This is because disjoint path excludes all links or nodes in the primary path. The main goal of ART algorithm is to reduce recovery convergence time. This section illustrates that the core principle of ART algorithm is that once network routers receive notification of an incident of failure, the ART algorithm in turn reroutes the traffic via pre-computed backup disjoint path until the routing protocol computes the new primary path. Additionally, ART algorithm approach avoids local loops in the computed backup path. ART is invoked by the source node, which reroutes the traffic after experiencing a link failure or receiving an LSA/LSP (if the source node has already computed a backup path in advance). The node must calculate and update a set of parameters regularly, and based on these parameters the source node must decide whether to invoke backup path. All these calculations performed by the nodes are based upon the table 1.

The mechanism of ART algorithm has been illustrated in the flowchart as shown in fig. 1 to show the mechanism of computing disjoint path. The algorithm computes disjoint backup path in the five following cases:

– First: Routing protocol computing the routing table for the topology.

– Second: Source node identifies its adjacent node through the routing table and then broadcasts a small packet to all adjacent nodes, excluding the adja- cent node (first primary hop) on the primary path.

– Third: Each adjacent node starts checking if it has a disjoint path to the destination not containing any node from the primary path. Thereafter, the adjacent node will send an acknowledgement to the source of whether or not it has a disjoint path.

– Fourth: The source node receives all acknowledgements from adjacent nodes.

If there is any positive answer, then the source node will add this adjacent node as a first hop, and its neighbours as the second hop in the new routing table.

– Fifth: If all answers are negative, the nodes will keep searching until the backup routing table is completed.

Algorithm 1 *Alternative Path* returns a set of alternative paths for every possible path in the routing table.

$G(V, E)$ is an oriented graph with two sets, a set of vertices V and a set of edges E, where an edge $e = (v, u)$, $e \in E$, $v, u \in V$ is a connection from vertex v to vertex u. A path P is a set if edges $e_1, e_2, ..., e_n$, such that if $v, u, x \in V$, then $e_i = (v, x)$, $e_i+1 = (x, u)$, for all $1 \leq i \leq n - 1$.

procedure *AlternativePath*(T_r)

T_r : The routing table
V : The vertex set in graph $G(V, E)$
$\Gamma(v)$: The set of adjacent vertices to a vertex v
$P_r(T_r, s, d)$: The path connecting the vertex s to d as in T_r
$P_a(s, d)$: An alternative path such that $Pr(Tr, s, d) \cap Pa(s, d) = \emptyset$
S_P : The set of all generated alternative paths
q_{sub}: A path
Q: A queue of couple (path, vertex)
Enqueue: Insert an element in a queue
Dequeue: Removes an element from a queue
Front: The element at the front of a queue

```
S_P ← ∅
for all s ∈ V do
  for all d ∈ V do
    if s = d then
      q_sub ← ∅
      Q ← ∅
      Enqueue(Q, (q_sub, s))
      while Q = ∅ and P_a(s, d) = ∅ do
        (q_sub, x) ← Front(Q)
        for all k ∈ Γ(x) do
          e ← (x, k)
          if (q_sub ∪ e) ∩ P_r(T_r, s, d) = ∅ then
            if P_r(T_r, k, d) ∩ P_r(T_r, s, d) = ∅ then
              p_a(s, d) ← q_sub ∪ e ∪ P_r(T_r, k, d)
              S_P ← S_P ∪ p_a(s, d)
              break
            else
              Enqueue(Q, (q_sub ∪ e, k))
            end if
          end if
        end for
        Dequeue(Q)
      end while
    end if
  end for
end for
return S_P
end procedure
```

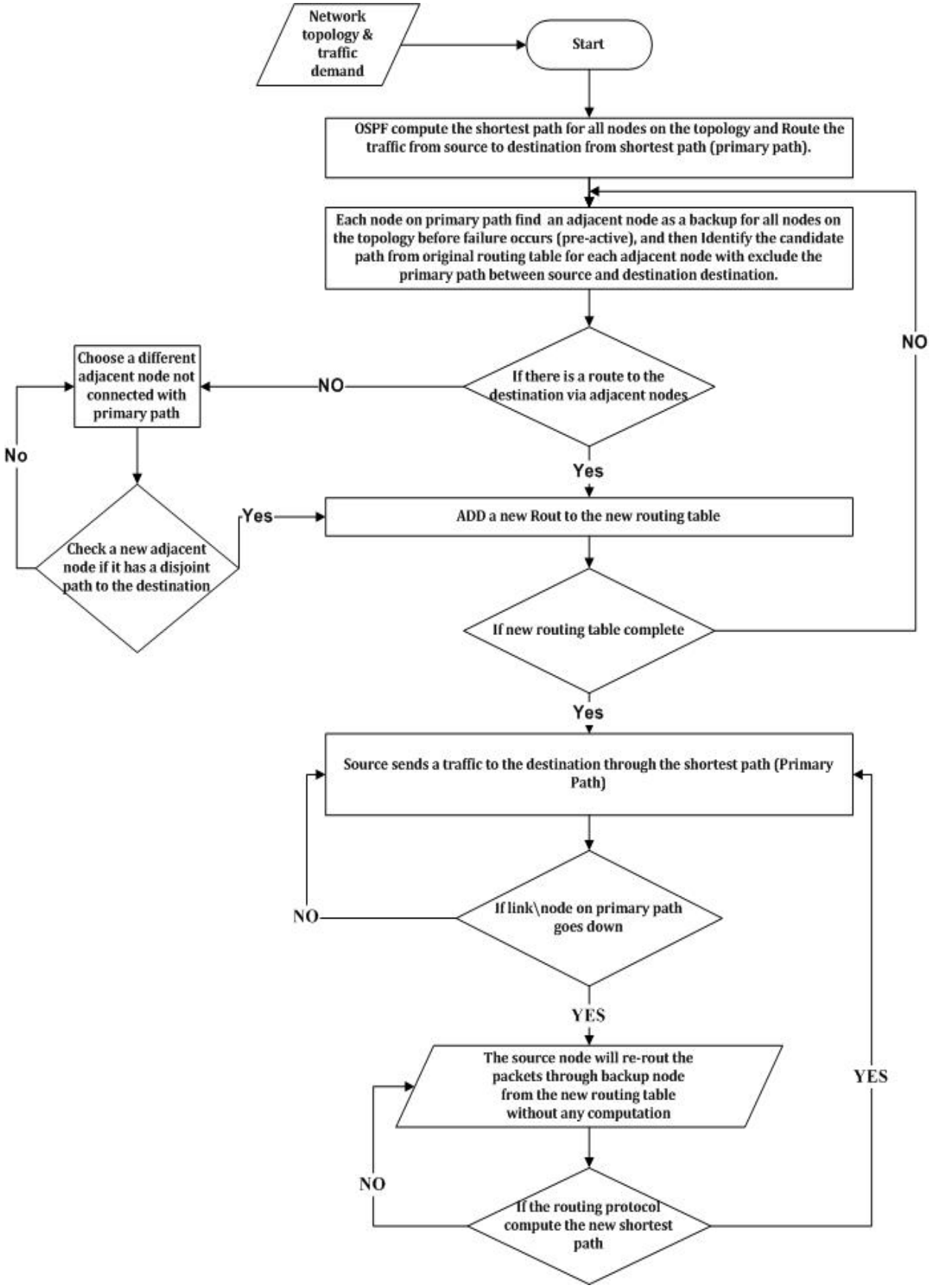
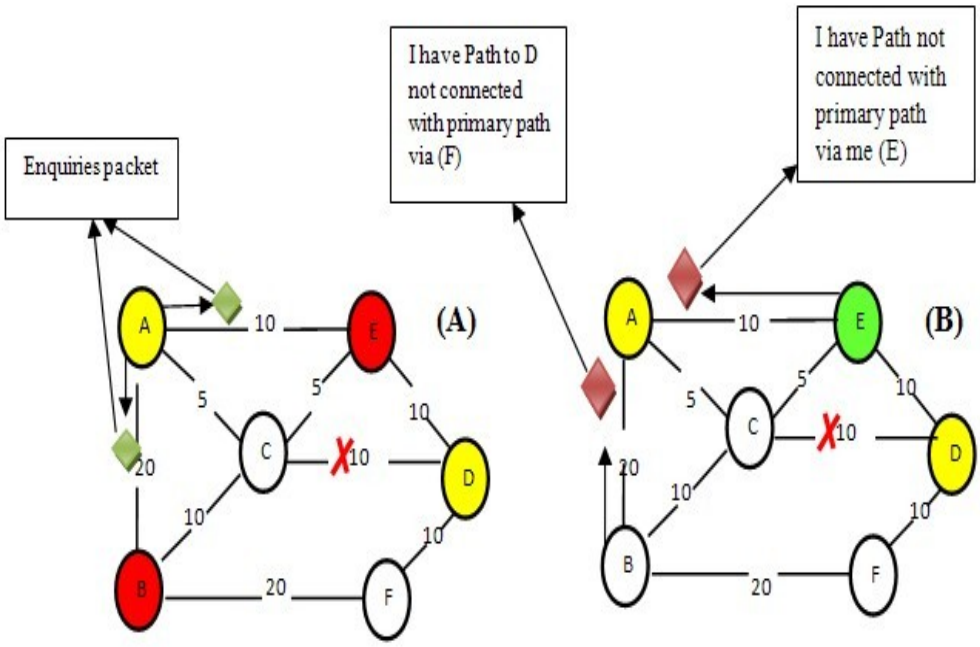


Figure 1. Algorithm Flowchart



Node	Computes Alternative Path from Primary Routing Table					
--	A	B	C	F	E	D
A	--	B	C	C	*E*	*C*
B	C	--	C	F	C	C
C	A	B	--	B	E	*D*
F	B	B	D	--	D	D
E	A	C	C	D	--	*D*
D	C	C	C	F	E	--

Adjacent Node For Node A
Node B
Node C
Node E has a path not connected with primary one to Dest. D → D

Figure 2. Example illustrating the ART mechanism

Figure 2 illustrates the mechanism of selecting the disjoint backup path. Assuming that node A is originating traffic destined to node D, node A will be the source node and will start to compute its disjoint backup path to node D. As the figure shows, the primary path between nodes {A and D} is as follows: {A->C->D}, hence, the node will broadcast small packets to the adjacent {E & B }, excluding node {C}, because it is on its primary path. Nodes E and B will check if they have a route from LSDB to destination D; if so, they will send an acknowledgement to inform the source that the route will be via them and their neighbours. If there are two paths, the source node will choose the best, based on the less cost. In fig. 2, node B cannot be a backup adjacent node in the first step because it has node {C} in its route to destination {D}. Once the source node receives an LSA informing it that failure has occurred, it will directly reroute the traffic via its backup path. On the other hand, if nodes {B and E} cannot be a backup adjacent node, and they do not have a disjoint path with the primary path to the destination, then the source node will send a packet for nodes { B and E}to check if they have an adjacent node that has a disjoint path to the destination. Hence, node {B} will send a packet to node {F} to ask if there is a disjoint path with the primary path to the destination. Node {F} will check the routing table and determine whether any path is available. As shown in figure 2, node {F} has a direct route to the destination, so node {B} will send an acknowledgement that it can be a backup adjacent node via node {F} to the destination, then the source node will add node {B} in the backup routing table as a first hop, and node {F} as a second one. The network design performs an important role in creating a backup route. For example, when two nodes are connected together by one edge and this edge between them fails, there is no possibility of rerouting the traffic between them and the graph will be a disconnected graph.

3.1. The ART Algorithm's Operation

Before describing the details of the algorithm, the

following delay timer and parameters are clarified:

– Link Failure Detection: the time between detecting the link failure and hardware level.

– SPF Delay: A default delay between receiving an LSA/LSP and starting the shortest path calculation.

– Neighbour SPT Calculation: Delay due the SPT calculations and routing table update performed at the neighbour node.

– FIB delay: The time required by any node to upload the original routing table.

– Adjacent Total Convergence Delay:

$$D_{SPT} + D_{SP\,T\,adjacent} + D_{FIB} \quad (1)$$

All delays are measured in seconds; hence, ART algorithm will operate once the source node receives a notification packet about the incidence of failure. Meanwhile, the routing protocol starts to compute the SPF and updates the routing table. When the source node receives a new copy of the routing table, it will forward packets in the new path.

The algorithm operations are based upon all timers being required to detect failure and notify all nodes about it, as shown in equation 1. Updating the routing table and computing a new shortest path in networks is presented in table 1. As shown in the algorithm in figure 2, in the negotiation phase each node sends small packets to enquire about the availability of a disjoint path to destination; at this time, each node will take on the role of both source and neighbour concurrently. The ART algorithm is a pro-active mechanism because it computes a new backup routing table, including a backup path for all nodes on the topology, in advance. However, the network topology assumes that the source node has at least one adjacent node not connected to the primary path, as shown in figure 2. When the routing protocol starts to converge along the network to construct the routing table and then forwards it for all nodes on the topology, the ART algorithm, based on the routing table, will extract all primary paths for each node on the topology to exclude the nodes connected to the primary one from the backup path. Such connectivity loss is also detected by receiving LSA/LSP from an adjacent node, or if the failure is connected directly, there will be a loss of signal. As presented in equation 1, if the source node detects a link/node failure, it calculates the time required for the routing protocol to re-converge the topology. As explained previously, the ART algorithm compels nodes to send an enquiry packet for each adjacent node to check which node has a disjoint path, but if all answers from adjacent nodes say that their paths are common with primary one, the ART algorithm will move to operate the second step of the algorithm, which is enquiring from adjacent nodes whether their neighbours have a disjoint path to the destination. If acknowledgement is positive, the source node will add it's adjacent as a first backup hop, and its neighbours as a second backup hop, to guarantee freedom from loops in the network.

The requirements for implementing the ART algorithm in nodes are classified into the categories of routing protocol modification and router modification. An extension code was implemented to work with link state protocol to compute backup path without the need to change the way of computing shortest path for the routing protocol. Additionally, this extension needs the link state protocol to keep track of different timers, and send them to the source node as an LSA/LSP. Router modification must have an extra memory size for inserting the backup path and keep it ready for use when failure occurs.

4. Related Research with Comparative Study of Convergence Time

While ART algorithm computes a disjoint backup path to reduce recovery time in the network, we can compare the recovery time of traditional link state protocol with the convergence time of a link state protocol combined with a state of the art ART algorithm. In order to compare the convergence time of a link state protocol with that of link state protocol combined with ART algorithm, the duration of each operation carried out by a regular link state routing protocol must be known; the relevant operations are link failure detection, LSP origination, flooding SPT computation and FIB updates. In a normal link state operation, all nodes (predecessor) can start forwarding traffic once they receive the link failure message, compute a new SFP and update the FIB. The Open Shortest Path First (OSPF) routing protocol is used as a dynamic link state protocol for TCP/IP or UDP traffic, and is designed to update the information for topology by sending an LSA based on the presence of a failure. The convergence time of the recovery mechanism is still too large for the real application. The convergence time can be of the order of tens of milliseconds or even tens of seconds in Border Gateway Protocol (BGP) [3]. Hence, during the process, while the routing protocol is converging, micro-loops may be created. This can lead to increased loss of packets and end-to-end delay in most applications such as video and VoIP traffics, because the source keeps sending packets to its destination until it receives a notification that a failure has occurred.

5. Results

Part of a Mesh Network was replicated by using a NS-2 simulator as a single area OSPF network. The links represented the OSPF routing metrics. UDP traffic is sent from source to destination. Every simulation involved one or two link failures on the primary path. The link failure could occur along the primary path at any time without notification and instantaneously at different hop count distances between source and destination nodes. When the simulation was run for 50 seconds, all the links can potentially fail and a loop could occur after 10 seconds [6]. This is because the routing protocol takes approximately 6.6 seconds to compute the primary path and construct the routing table for the network's topology [7]. Therefore, one needs to give the routing protocol enough time to guarantee that the routing

table for each node has been constructed, and in turn, each node has received identical copy information about the network topology. This enables data packets to be transmitted successfully to their final destination and then they can begin to compute the backup route from the original routing table. In this experiment, we ran the simulator 50 times with randomly configured failures between source and destination. In addition, the source and destination were randomly configured. This caused failure to occur arbitrarily and haphazardly. The LS protocol started to construct a routing table for the network's topology once the network started working. The UDP traffic for all the source nodes was configured to start sending from 1.0s to 50.0s. If the LS protocol is not combined with the ART algorithm, failure can lead to increased loops in the network. This phenomenon will be harmful if it causes link utilisation to reach 100%, which will increase loss of packets.

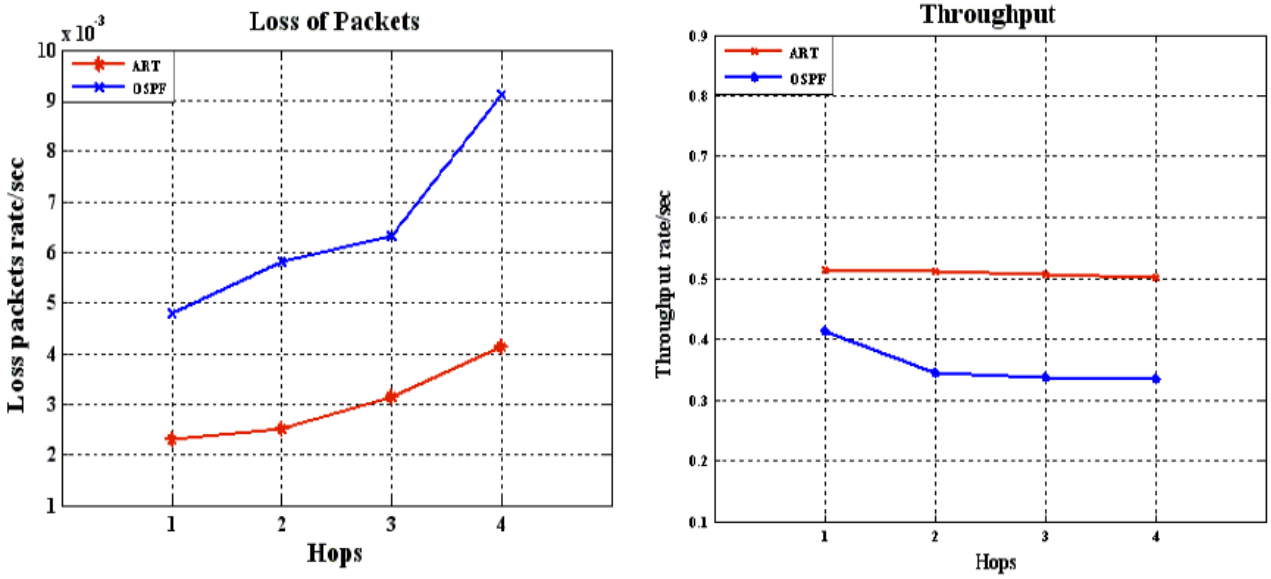
Table 2. Simulation Parameter for Backup Path

Parameter	Value
Routing Protocol	Link State, ART
Simulation Time	50sec
BW	2Mbs
Traffic	CBR
Routing protocol	LS , ART

This experiment was implemented using the parameters stated in table 2. At first, we ran the simulator with only LS protocol. After analysing the results, we again ran the simulator for the same network topology with both LS protocol with ART algorithm in order to evaluate the impacts of the computed disjoint backup path by ART algorithm.

Figure 3(a) shows the average packet loss for LS protocol only and LS protocol combined with ART algorithm. The results show that loss of packets increases linearly when the incident of failure occurs far from the source node. According to the parameters which indicate the required time to notify nodes about failure, the figure shows that loss of packets becomes higher when failure occurs after one hop from source node. In the graphs, the x-axis shows the number of hops to the destination node, while the y-axis shows the mean loss of packets during the simulation time. In case of the failure occurring far from the source (i.e. the link after three or four hops), in both cases, LS with and without ART technique, the source node will continue sending data packets until it receives a notification from routing protocol that a failure has occurred in the network. In the case of LS with ART algorithm, the traffic will be re-routed along an alternative path, which is computed by ART algorithm, until the LS protocol updates the information in the routing table and computes the new shortest path.

On the other hand, fig.3 (b) shows the average throughput per traffic flow for LS with and without ART algorithm. The graph shows that when a link failed far from source node the throughput was reduced in the network. This is because the source node forwards data packets to the destination based on its routing table, therefore when failure occurred after two or three hops from source, the time is needed to inform the source node becomes longer, which leads to increased packet loss.



(a) Loss Of Packets (b) Throughput

Figure 3. Loss Packets and Throughput

Figure 4 shows the load for the link state protocol with and without ART algorithm. This load is related to the extra packets and represents the total traffic load in the network before and after failure. The load of the link state combined with the ART algorithm has a higher load than the link state only, because ART mechanism induces all nodes on the network topology to send enquiry packets to check all its adjacent nodes, and to determine whether there is any disjoint route with the primary path to the destination. Hence, the comparison shows that the ART algorithm does not degrade from the network performance. Additionally, the load increases when a failure occurs far from the source node for two reasons: firstly, the source is still forwarding data packets until it receive a failure notification; and secondly, the packet has already sent by the source node can suffer in the network topology by transferring among nodes until they drop or the new path is computed.

Fig.5 (a)shows the end-to-end delay between source and destination. The delay by our algorithm, in some cases, is better than the link state protocol because our algorithm can select an alternative path, which is both the shortest and is at the same path for the new routing table. On the other hand, our algorithm, in a different case, will choose an alternative path but not necessarily the shortest one.

When failure occurs, the source node will wait to receive a notification from other nodes about it and then the routing protocol will start to re-compute and update the routing table. In this case, fig.5 (b) shows the re-routing time for the link state combined with the ART algorithm and without it. LS with ART algorithm required less time to reroute the traffic via another path when the primary one failed. This is because the LS with ART algorithm are waiting until the source node knows about the incident of failure in the network, without needing to wait until routing protocol updates the routing table and computes a new path. Once the source node receives this notification, it will use the alternative disjoint path to pass the data packets and deliver them to the destination. In addition, congestion can arise when failure occurs; therefore, the notification message can take longer to arrive at the source node.

Figure 6 shows the utilization of the links on the network. In link state protocols its utilization can exceed the limit size of the links because when a failure occurs then a loop might be created between the nodes whos knows about failure and the other nodes doesn't know about it. This leads to increase loss of packets and degrades the network performance. On the other hand, the link state with an ART algorithm shows a higher utilization and avoids exceeding the limit size of the link because the ART algorithm offers a backup path for re-routing packets in case of failure. This will avoid loop in the network.

Figure 7 shows when the failure might come up and down frequently during the data transmission. Therefore, the computed backup path has showed that it can reduce the delays to deliver the packets to the final destination in less time. The figure indicates if the failure can occur on the primary path, then the ART algorithm will reroute the data packets via backup path until the routing protocol will update the information for the network topology. The result shows that there is an improvement when there is an existing computed backup path.

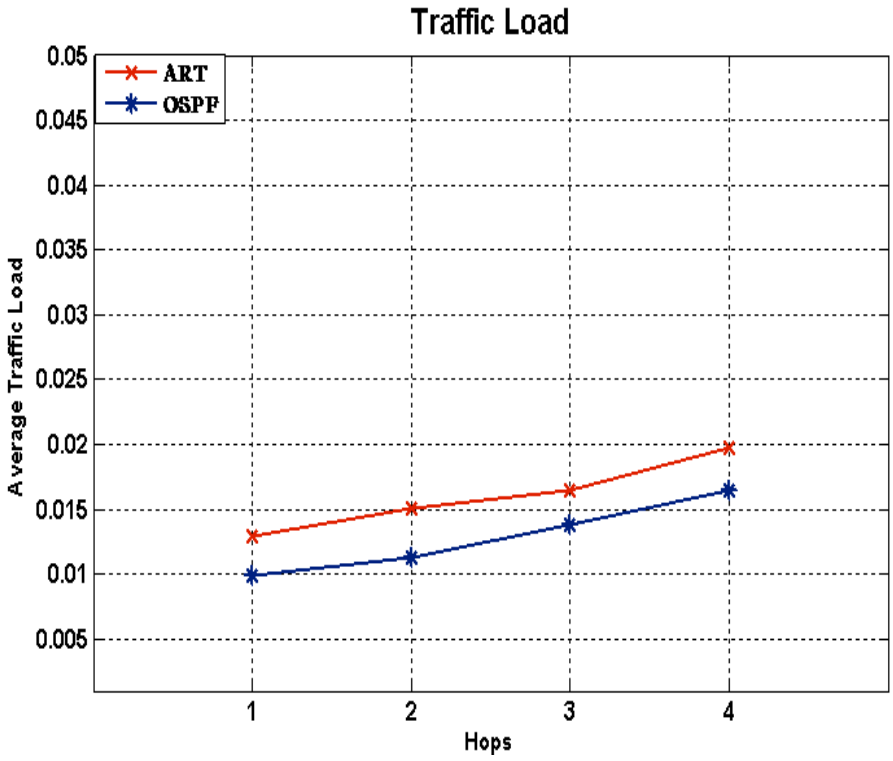
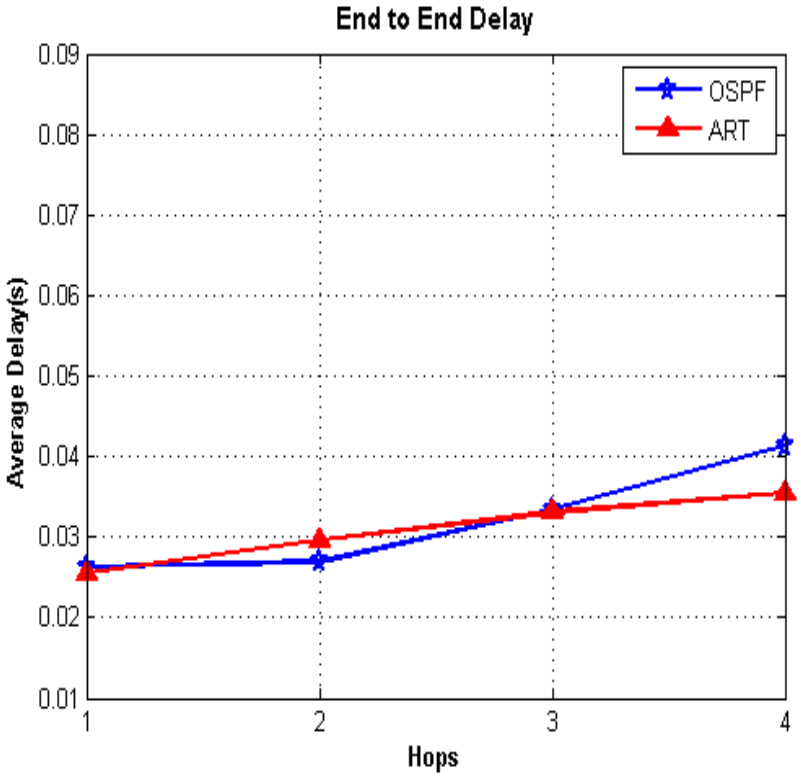
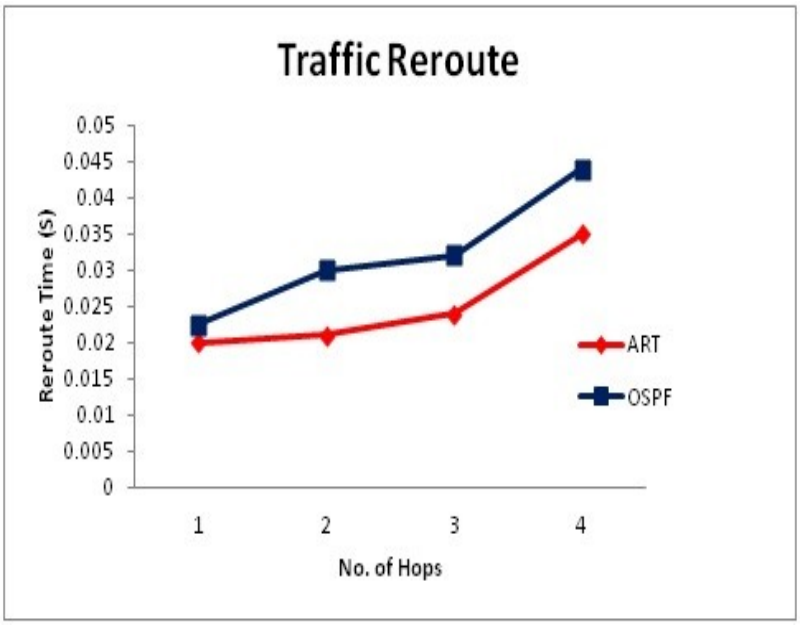


Figure 4. Load



(a) End to End Delay



(b) Reroute Time in Sec

Figure 5. End to End delay and Reroute Time in Sec

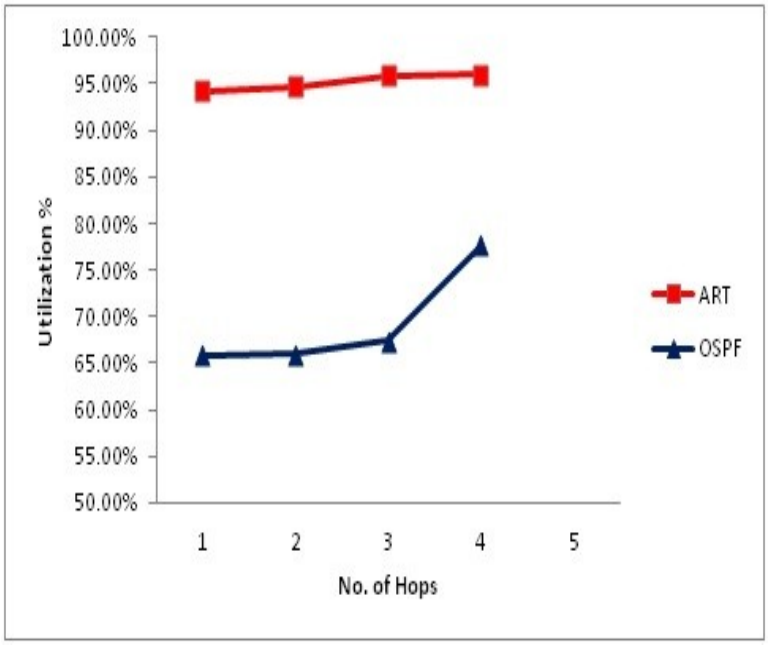


Figure 6. The effective backup path during the frequently of failure

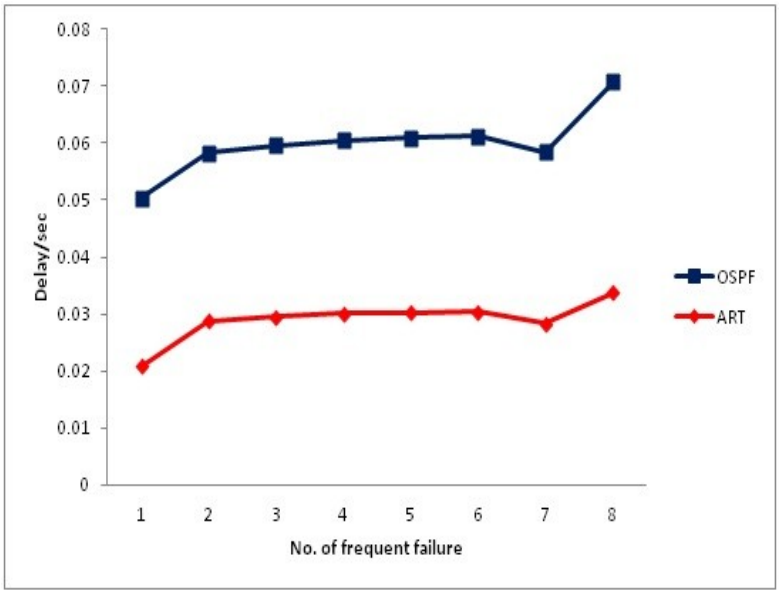


Figure 7. The effective backup path during the frequently of failure

6. Conclusions

This paper introduced a new algorithm called ART mechanism. When the failure occurred in an IGP network that will lead to degrade from network performance, and lead to producing such problems as local loops in the network. When local loops occur, many problems can arise, such as wasting network resources and delaying sensitive traffic passing through this network. The ART mechanism reduces recovery time by computing a disjoint path between the source and the destination in advance. In case of link state protocol only, the convergence delay will be higher, and links can form loops and holding time for the packets in the nodes buffer will increase and then lead to increase delay in the network. Hence, LS combined with ART algorithm not only reduces recovery time; it also improves the quality of service for sensitive traffic and reduces packet loss and delay times in the network. However, the ART algorithm creating a new backup routing table has a disjoint backup path for all nodes on the network. The new backup routing table is based on the original routing table, which is computed by a link state. For real time traffic, the results show that LS with ART algorithm reduces the loss of packets and delay between source and destination. Additionally, we have proved that the ART algorithm can avoid the occurrence of loops in the network by keeping the utilisation stable compared to link state protocol. The ART algorithm has its own messages that are sent between nodes to create the new shortest path, and

these packets do not affect the performance of the network.

REFERENCES

[1] Feasibility of IP restoration in a tier-1 backbone, volume 18, 2004.

[2] R. Banner, S. Member, and A. Orda. Multipath routing algorithms for congestion minimization. In in NETWORKING, pages 536–548, 2005.

[3] A. Basu and J. G. Riecke. Abstract stability issues in ospf routing. 35:225–236, 2001.

[4] M. Ericsson, M. Resende, and P. M. Pardalos. A genetic algorithm for the weight setting problem in ospf routing. Journal of Combinatorial Optimiza- tion, 6:299–333, 2002.

[5] P. N. et al. New dynamic spt algorithm based on a ball-and-string model. pages 706–718, 2001.

[6] B. Fortz and M. Thorup. Optimizing ospf/is-is weights in a changing world, 2002.

[7] P. Francois, C. Filsfils, J. Evans, and O. Bonaventure. Achieving subsecond igp convergence in large ip networks. 35:35–44, 2005.

[8] H. Kin-Hon, N. Wang, G. Pavlou, Botsiaris, and Christos. Optimizing post- failure network performance for ip fast reroute using tunnels. In QShine '08: Proceedings of the 5th International ICST Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness, pages 1–7, ICST, Brussels, Belgium, Belgium, 2008. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

[9] Menth, Michael, and Hartmann. Loop-free alternates and not-via addresses: A proper combination for ip fast reroute. Comput. Netw., 54:1300–1315, June 2010.

[10] P. Narvaez, K.-Y. Siu, and H.-Y. Tzeng. Efficient algorithms for multi-path link-state routing, 1999.

[11] S. Nelakuditi, S. Lee, Y. Yu, and Z. li Zhang. Failure insensitive routing for ensuring service availability. In In IWQoS03, 2003.

[12] M. Pascal, P. Jean-Jacques, and C. Stephane. Path computation for incoming interface multipath routing. European Conference on Universal Multiservice Networks, 0:75–85, 2007.

[13] P.Merindol and J.-J. Pansiot. Improving load balancing with multipath rout- ing. 2008.

[14] R. Rabbat and K.-Y. Siu. Restoration methods for traffic engineered networks for loop-free routing guarantees. 2001.

[15] G. Schollmeier, J. Charzinski, and A. Kirstdter. Improving the resilience in ip networks. pages 91–96, 2003.

[16] M. Shand and S. Brayant. A memetic algorithm for ospf routing," proc. in- forms telecom. pages 187–88, 2002.