International Journal of Innovative
Computing, Information and Control
Volume **14**, Number **3**, June **2018**

ICIC International ©2018 ISSN 1349-4198
pp. **899–913**

FEASIBILITY OF IP BY ADAPTIVE VIRTUAL ROUTING IN IGP NETWORKS TO ENHANCE SERVICES IN CLOUD COMPUTING

Radwan Saoud Abujassar

School of Technology
Arab Open University
Kuwait-Branch, Al-Ardia 92400, Kuwait
r.abujassar@aou.edu.kw

Received October 2017; revised February 2018

Abstract. *In this paper, a new technique is proposed to improve the data services in cloud computing by finding an available path by making an interoperation between the lower networks layer. The proposed technique enables layer 2&3 co-operation together to make path slowing detection and then the traffic will be re-routing via the mentioned path in less time between request and server receiver. The load balance and rerouting in less time lead to improve the services in the cloud computing network. The technique involves detecting many alternative path via each adjacencies node in the network topology. The rerouting for the requested service to the receiver server is required to create an algorithm to compute the utilization for each path after the routing protocol in IGP network constructs the routing table for network topology. In case of failure, the data packets do not need to wait for the routing protocol to update the routing table for the network topology. This is because the proposed technique will nominate other existing paths to the destination to pass the packet to the final destination. The proposed algorithm will convert the traffic via its available path with ensuring that there is no loop in the network until the routing protocol considers the updating information for all nodes in the topology. In the network, layer 2 has demonstrated its ability to detect the path failure in case loss of signal. It is extremely quickly through the immediate detection of the loss of light signals for the link or node. The mechanism switches the data packets through an adjacent node to its requested server in cloud network via the life node. The aim of this mechanism is to avoid loss of packets especially for real-time traffic and improve QoS.*

Keywords: Quality of service (QoS), Service level agreement (SLA), Open shortest path first (OSPF), Internet protocol (IP)

1. **Introduction.** The architecture of the IP routing is categorized into two main types: intra-domain and inter-domain routing. The intra-domain routing protocol, also called interior gateway protocol (IGP), includes OSPF, IS-IS and RIP protocols. In RIP protocol, the distance vector algorithm (Bellman-Ford algorithm [1]) is used to compute the shortest path between source and destination [2]. On the other hand, the Dijkstra algorithm is used in link state protocol (OSPF and IS-IS) [3, 4]. In inter-domain protocol, the data packets are forwarded between a set of autonomous systems (AS) networks. Additionally, the inter-domain network is applied on large-scale networks, which are connected to each other via the only routing protocol in the network, called border getaway protocol (BGP) [5]. In IGP network, inter and intra domains are different from each other in their objectives. This is because the intra-domain is always trying to compute the best and shortest path routing, whereas the inter-domain routing seeks commercial routing. The intra-domain deals with tens of thousands of destination prefixes, whereas in inter-domain, the BGP protocol deals with 250,000 prefix destinations [6]. In a network, the data will be forwarded from the host location to another one via router by checking the IP address for

the destination [7]. The control plan is an important component in the IP router that can drive the route to make a decision whereby it can forward the packets based on the link state advertisement (LSA) information of exchange routing information between routers to construct forwarding states. In this paper, QoS is one of many factors that consider network services more efficient by enabling high performance in the transmission data. QoS informs how new architectures have been developed to support numerous applications, such as, video and VOIP [8]. Raising QoS demands that constraints are overcome in order to find a path that is tolerant of all traffic without degrading the network [9]. Traffic engineering aims to support load balancing in the network. It could be useful when there are two paths that share the same cost to the destination. Hence, we will propose a new idea to reduce congestion on one path even in case of failure by checking the utilization for each link between source and destination [10]. The new approach will use layer 2 to detect failure and layer 3 to find a provision path and pass all packets to their destination via different paths until the main path recovers. Significantly, the design of network plays an important role in the recovery mechanism through the number of ingresses and egresses for each node [8]. The rest of this paper is organized as follows. In Section 2, we describe related works about incentive schemes including previous work for the Ethernet and network layers. In addition, we describe the proposed techniques and their mechanisms as well as using some mathematical modelling with theoretical analysis for the proposed technique in Section 3. In Section 4, we have showed our methodology for testing the new technique, and showing the configuration environment on real networks. The results are shown and the performance of the proposed technique is evaluated in Section 5. Finally, conclusion and future work are discussed in Section 6.

2. **Related Works.** The cloud computing is considered a distributed system because it has the abilities to provide multiple external customers by using internal technologies. Service providers offered many services to end customers. They have a service level agreement (SLA), according to which one of the service providers must deliver services with a better performance to the end users [11]. Service may suffer disruptions for short durations due to the many factors that affect the network, causing economic damage, and degrading network performance. Therefore, researchers proposed a number of different solutions for the cloud computing services in the network. This is because flooding and load over flow are common and varied in the everyday operation of the cloud computing network. When load occurs, it causes undesired behaviour, for example, the delay for the completed task is increased because the pre-processors on service nodes cannot execute many high complexity jobs during a short time. Therefore, the aim of this paper is to develop and propose a potential solution for the cloud computing network through alleviating and allocating job task based on the service node availability. The proposed algorithm should reduce the time to complete each task required by the users. IP recovery mechanism performance could be enhanced by improving the current interaction time between layers 2 and 3. Layer 2 can detect failure faster than layer 3 within a short time due to loss of light or signal. Hereafter, when failure occurs, layer 2 should inform layer 3 by a trigger about the failure to start to re-update the routing table and compute the new shortest path coupled with support from the proposed technique. The packet will pass from an alternative path to its destination until the routing protocol in layer 3 is able to complete all these processes. The time for interactions between layer 2 and layer 3 should become shorter, therefore, achieving a reduction in recovery time and making the network more reliable. Ten percent of the failures related to the hardware, which took longer than forty-five minutes to repair. Forty percent of the failures related to the software, which took fifteen minutes to repair and reboot the systems. Forty-six percent of the failures

related to the link or nodes being down, which in some cases took less than one minute to repair. Four percent of the failures required human intervention, which took from 15 min to 45 min to repair. There are three processes to reduce recovery time in case of failure: failure detection; failure notification; re-compute the new shortest path. Table 1 shows the components restoration time when failure occurs. OSPF is used widely in the networks [12]. The OSPF protocol uses Dijkstra algorithm [13] as a routing algorithm in the network. Hence, all routers use the Dijkstra algorithm to compute the shortest path tree (SPT). The IP packets are routed by OSPF protocol based on the IP destination address, which is in the IP packet header. When OSPF is used in the network, each router discovers and maintains a full picture view of the network topology by flooding link state advertisements (LSAs) as shown in Figure 1.

Table 1. Component of the failure convergence time [18]

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		
LSP processing	10ms/hop	
SPF computation	100-400ms	
Forwarding information update	20 entries/ms	

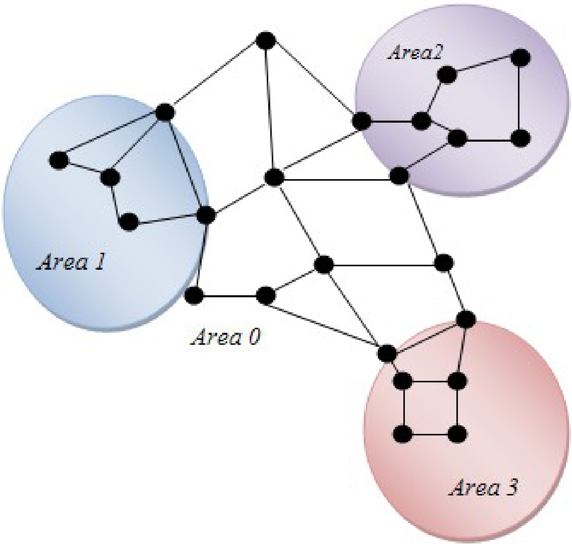


Figure 1. Clustering network

The highlight points are, if the router has multiple outgoing links then one of them will be a primary path. However, the outgoing links may configure as backup ones, which are used in case a failure occurs. In this case, the router will use a Hash function with some information in the header to assign packets for which outgoing links will transit to the destination. Hash functions are well defined procedures or functions that are used to speed table lookup to find an item in a database. In [14] the author discusses the cost links in the network and traffic engineering. The cost of links is considered as an important parameter in determining the best path for the routing protocol algorithm. One of many problems here is when the backup path passes other traffic then the link load will become high, which will lead to congestion in the network. Hence, when it occurs, loss of packets will increase. The traffic engineer (TE) mechanism has solved this problem by allocating the traffic through more than one path of equal cost (if it exists) with less utilization. If we assumed there are other paths that have a similar cost compared to the primary shortest one, we can then shift and divide the traffic from all ECMPs to decrease utilization of the primary path and achieve load balance on the network. In addition, the ECMP will avoid loop in the network. In our mechanism, the link load has been measured by [15, 16],

$$\mathrm{LoadLinkMetric} = (\mathrm{DataSize})/(\mathrm{capacity} * \mathrm{Time}) \tag{1}$$

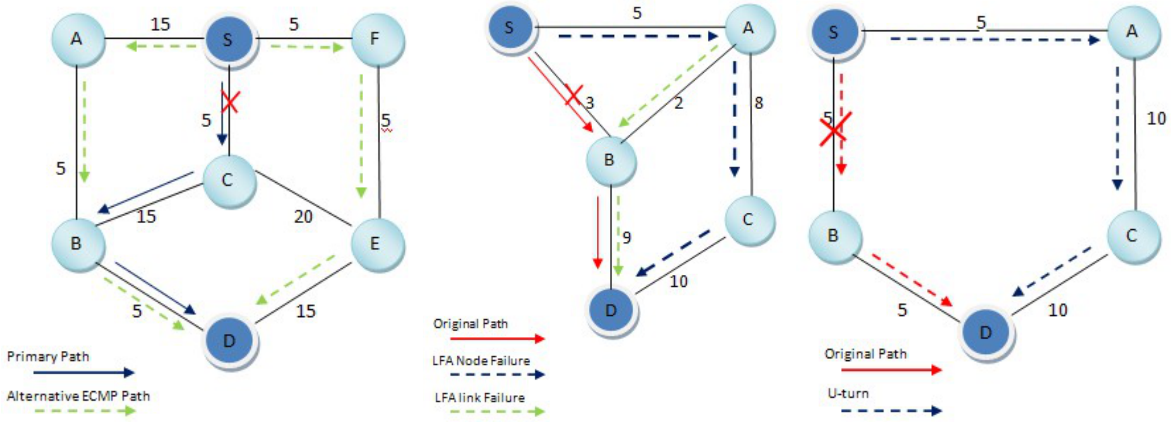
$$\mathrm{LinkCost}\,(\mathrm{utilization}) = \mathrm{linkcost} * w * \mathrm{utilization} \tag{2}$$

$$\mathrm{Utilization} = (\mathrm{Databits} * 100)/(\mathrm{BW} * \mathrm{interval}) \tag{3}$$

Link/Node Failure Detection in IP Environment. Node or link failure can occur suddenly and without any notifications in the IGP network; therefore, this failure will impact the network performance by reducing available capacity and disturbing IP packet forwarding. In [17], the authors presented the convergence time in SPRINT backbone network. Once the routing protocol converges, the network topology and all routers receive the new information; hence all link-state databases will be more coherent, which will return the network to the stable status. In case of failure, seven processes are required before the routing protocol starts to re-converge the network, each of which takes a different amount of time as described in Table 1.

The table shows that the link layers take up to 6.6 seconds in order to re-converge the network after the failure occurred. On the other hand, the authors in [17] found that the minimum or default values for the convergence time overall took up to 2-3 seconds. Regarding the high time of default values, in [19] the authors recommended that the minimum values can be configured in the network. This is because the current routers have high processor performance, which leads to fast restoration. According to the previous studies, the current routers are able to update FIB with 20 entries within a few milliseconds. The total of restoration time can be performed within less than a second without using any recovery schema when failure occurs. However, the restoration schema is considered to offer more flexibility with regard to the location of failures. The convergence time can be of the order of a hundredth of a milliseconds or even tens of seconds in the BGP networks [18, 20]. The authors showed that the inter-domain routers may need tens of minutes to reach the full view of the network topology when failure occurred. 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 applications such as video or VoIP traffic, because of unsynchronized receiving of updated information; therefore, these routers will keep sending packets until they receive notification that a failure has occurred. In [21], the authors provided a study of the effect of failure on network stability. This study analyzed monitoring logs provided by a regional provider which connected 130 cities. The mechanism was used to monitor the routers' interface by sending ping messages every ten minutes between them. The results showed that 80% of links are repaired within two hours. The main disadvantage of this mechanism is that the ping messages do not provide any information about the location of failures, which consequently cannot lead to reducing recovery time to less than ten minutes. On the other hand, link/node failure still exists in the network, and both links and nodes can go down suddenly without any notification. Recent research analyzing link or node failure has sought to identify the reasons for failure, and which component is more unreliable, underlining the causes of failure in the network and highlighting the impact of incident failure on network performance. Additionally, node or link failures have the potential to cause a local loop in the network. A local loop in the network leads not only to increased loss of packets, but also to producing congestion in the network. Some studies have discussed the methodology for detecting local loop when it occurs in the network. Loop detection is based upon monitoring the packet sequence number when the trace-route measurement records the same sequence packets multiple times, which indicates the incidence of a loop in the network. In [17], the authors investigated the incidence of failures in Sprint's IP backbone, and how they impact real applications in the network such as VoIP. The service level agreements are based on three measurements: packet loss, packet delay and routers port availability. Regarding these measurements, the SLAs may guarantee that average loss of packet rate is 0.3%, and average delay is 55 milliseconds within the continental USA, with routers ports availability of up to 99%. The authors in [22] issued their study about network failures based upon three insights: first, the most frequent failures that cause networks to go down are connected to the hardware, software and human errors; second, estimate the impact of failures in the network (i.e., how much a failure will affect the network once it occurs), and make limited resources ready for troubleshooting in case failure occurs; third, the effect of network redundancy allows the routing protocol to find an alternative path to pass packets through it when the node/link goes down. However, the network redundancy has shown that the packet loss in the network reduced by up to 40% when failure occurred.

IPFRR has many schemas such as equal cost multi-path (ECMP), loop free alternate (LFA), U-turns, Tunnels (as described above) and Not Via Address [23, 24, 25]. Routers can deliver data packets to the destination through multiple paths when the ECMP exists. This is because all paths have the same cost, with different transit links to the destination. In Figure 2(a), Node S can forward the traffic via three ECMP paths because they have the same total cost to the destination node D. In LFA, each router will compute an alternative next hop and it will be used when failure occurs and is detected in the network. Therefore, the traffic in the network will not be disrupted during the network convergence, as shown in Figure 2(b). In this figure, the scenario shows that Node S is sending traffic to the destination D via S-B-D. If link S-B goes down, S will reroute the traffic via pre-computed path LFA, which is S-A-B-D. In case node B fails, the traffic will be rerouted via S-A-C-D. Once the new routing table is updated and the new primary path has been computed by the routing protocol, the traffic will be passed along the newly computed next backup hop. The traffic will continue passing along backup next hop for a period of time, which



(a) ECMP (b) LFA (c) U-turn

Figure 2. IPFRR schemas (ECMP, LFA, U-turn) [23]

should be enough to ensure that the new primary path has been computed, before the traffic switches back and passes through new primary path. Figure 2(c) shows the U-turn schema that can exist; U-turn schema selects an adjacent node whose primarily next hop is node S. Hence, node A, when it receives traffic from node S and needs to forward it to the destination D, returns the traffic to node S according to its routing table, because node S is the next primary hop for node A and vice-versa, which leads to creating loops in the network, therefore, Node S has to identify the traffic as U-turn traffic, because when node A receives this traffic, it can recognize that this traffic should pass via its loop free alternate path which is via S-A-C-D; the loop can thereby be prevented. The drawback in this schema is that it requires more computational complicity time compared to LFA. Not-via needs a special IP address given to each protected interface. Not-via indicates that packets are given this address should they be delivered to the router who announces for that address. Thence, not-via address requires two address: the normal IP address and not-via address. When routers are alerted that a failure has occurred, the repairing router will tunnel data packets towards the not-via address component. This router will receive the data packets and then de-encapsulate them and check the normal IP destination address to forward them without any further problems to the final destination. Figure 3 depicts a small scenario to show how not-via address mechanism works.

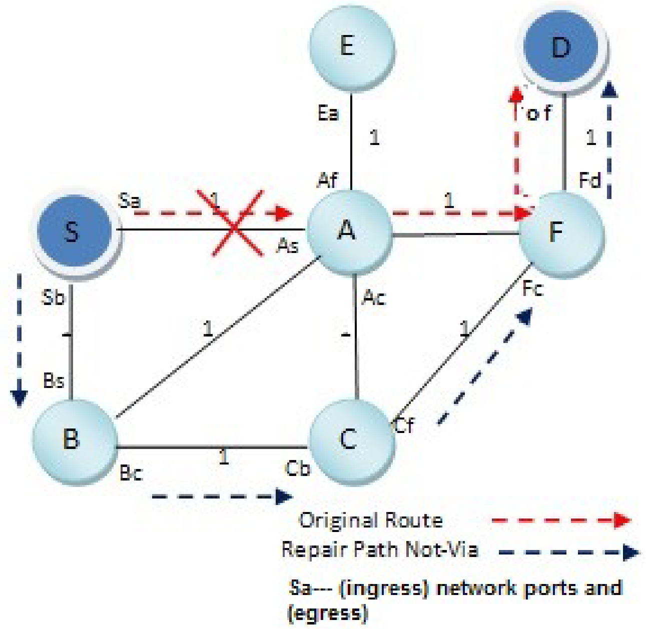


Figure 3. Set of not-via addresses for each interface [23]

Node S sends traffic to destination D via F. When S apprehends that A has failed, it tunnels the data packets to Fa, which is considered the shortest path to destination D from source S to node F, without the need to pass via node A; thereafter, when the data packets reach node F, they will forward to final destination D based on the routing table for node F, which is the shortest path. In contrast, the authors of [26, 27] have proposed a novel technique called failure insensitive routing (FIR). They described the main objectives of this technique as being to provide a fast local rerouting to deal with transient link failures with minimal changes to the current networking infrastructure. FIR approach is considered for the realization of high service and reliability without changing the conventional destination-based forwarding paradigm. Thence, there are two main ideas that support the FIR approach based on local re-routing respectively: interface-specific forwarding and failure inference [28].

3. **The Proposed Technique Aggregate Link/Node Performance.**

The novelty in this paper we have configured a virtualization path via the proposed algorithm in case of overhead or failure in the network. This technique leads to make the requested task from the user that start to be executing with less delay. As we know, for the end users, they are expecting to get a full service with less failure and distortion. The reduce affection from the mentioned factors lead to increase the reliability and robust between the user and service provider. We have assumed that all networks will be a mesh topology. This is because it contains many numerous of links and nodes in one network and this topology acts the intermediate between request and server. Each node can connect with more than one node in different egress ports. Each source node connects with an adjacent node (its neighbour). Mesh topology is a good example to use and begin to implement my ideas and conduct experiments to test them. Mesh topology has an efficient design, which can make more than one alternative route with showing the availability. Mesh topology comprises a large network (as mentioned above), which is relevant to my research. Moving forward we used a hybrid technique on a large number of connected nodes on different networks [29].

3.1. **Algorithm.**

We assume that all nodes on the main path can determine an adjacent node as a backup when the topology starts working. Described below are some of the conditions that must be fulfilled to achieve what is demanded. We addressed the following conditions in all our experiments. The adjacent router should not be affected by the failure or congestion (it is not on the main path between request and server). We have illustrated the proposed algorithm in Algorithm 1. The adjacent node must have a disjointed path with the main path. Each node will know about the failure through layer 2. The new path should have enough capacity to tolerate additional packets from the other node in case of failure. The delay for each link in the topology must be less than or equal to the delay on the main path. The algorithm has proved its efficiency through the theoretical analysis and its implementation on the NS2 simulator. A theoretical analysis has shown how the alternative route is computed and merged with the new routing table to be activated when any affection problem occurred on the service.

3.2. **Theoretical analysis for the proposed algorithm.**

In network, the packets are forwarded to the destination via less cost of quality metrics. Many routing protocols can be displayed within that premise. However, the pro-active protocol maintains up the routing table up to date and periodically this will tack more time than pre-active protocol. We assume there is a graph, $G = (V, E)$, where V is the set of nodes and E is the set of edges connected with different nodes. A virtual edge exists between the nodes according to the routing table which is already maintained by the routing protocol. We assume in the proposed algorithm that the routing table is already built and the packet starts to be forwarded in RT_{r0}. Then the nodes can know their adjacencies through the routing table as $\Gamma(V_n)$. Hence, the path from source to destination will be as following:

$$Path_{RT_{r(i+1)}}(s, d) \leftarrow \emptyset$$

Theorem 3.1. *For pro-active protocols such as OSPF, the Dijkstra algorithm finds the shortest paths from a single source to all other nodes in the topology. With adjacency matrix representation, the running time is $O(n^2)$ By using an adjacency list representation, and a partially ordered tree data structure for organizing the set $\{V - S\}$.*

Algorithm 1 *AlternativePath* returns a set of alternative paths with low utilization

```
procedure $AlternativePath(T_r, s, d, edges\_to\_avoid)$
$T_r$: The routing table
$V$: The vertex set in graph $G(V, E)$
$\Gamma(v)$: The set of adjacent vertices to a vertex checking the U $v$
$s$: The source vertex
$d$: The destination vertex
$p_a(s, d) \leftarrow \emptyset$
if $s \neq d$ then
    $q_{sub} \leftarrow \emptyset$
    $Q \leftarrow \emptyset$
    $Enqueue(Q, (q_{sub}, s))$
    while $Q \neq \emptyset$ and $p_a(s, d) = \emptyset$ do
        $(q_{sub}, x) \leftarrow Front(Q)$
        for all $k \in \Gamma(x)$ do
            $e \leftarrow (x, k)$
            if $(q_{sub} \cup e) \cap edges\_to\_avoid = \emptyset$ then
                if $P_r(T_r, k, d) \cap edges\_to\_avoid = \emptyset$ then
                    $p_a(s, d) \leftarrow q_{sub} \cup e \cup P_r(T_r, k, d)$
                    break
                else
                    $Enqueue(Q, (q_{sub} \cup e, k))$
                end if
            end if
        end for
        $Dequeue(Q)$
    end while
    $Q \leftarrow \emptyset$
end if
return $p_a(s, d)$
end procedure
```

We assumed there is a $G(V_0 \dots V_i, E)$, where V_i is the number of nodes from node 0 to node $i-1$; E is number of edges between nodes. The design of network should contain the number of nodes $\{N\}$. The number of edges for the proposed network can be computed as follows: # edge = 2^n according to the graph tree computation. Let $V = \{V_0, V_1, V_2, \dots, n - V_i\}$, and assume the source node is $\{V_0\}$; cost of arc(i, j) initially $= \infty$.

Initially: The set $V = \{V_0, V_1, V_2, \dots, n - V_i\}$ which contain all nodes; $S = V_0$ to push all nodes have the shortest path. Source node is $\{V_0\}$ and $G\_K = \emptyset$; which is a graph set of alternative K edges to the destination. Choose a vertex $W \in V_1 - S$, such that $D[W]$ is a minimum distance as we assumed that S is the V_0; $S = S \cup W$; for each vertex $V \in V_1 - S$

$$D[V] = \min(D[v], D[w] + C[w, v]) \quad S = \{V_0, \dots, V_d\}$$

If the primary path goes down, then the computations will be as following: $S = S \cup \{w\} \div \{K\}$; for each vertex $v \in V - \{S\&K\}$. The all graphs now are covered so the next step is to check the paths for all servers.

$$D[v] = \min(D[v], D[w] + C[w, v])$$

According to the above aggregation equations, we can visit the graph and determine which path can be used after the primary path or even works both together for making load balance if they are given the same level of services to the end user. The $D[v]$ is the main path for one of the main destination services; hence, we now checking if there is any other path that can guide to the same destination and add this path to our path list. Therefore, we defined a random variable $X_j \in \{0, 1\}$ indicates to the link status between any two nodes $-A$ and B of a sub region. The index of j indicates to the time when A sends the update message to the node head. The sequence of $\{X_0, X_1, \ldots, X_j\} = \{(X_j^{\infty}) \div n = 1\}$. The random process $\{X_j\}_j^{\infty}$ is modelled as a chains. Here, the proposed algorithm has shown the aim functions through computing a backup path which can be the optimum path between source and destination. This backup can be considered the optimum and nominate path in some cases for creating path lists.

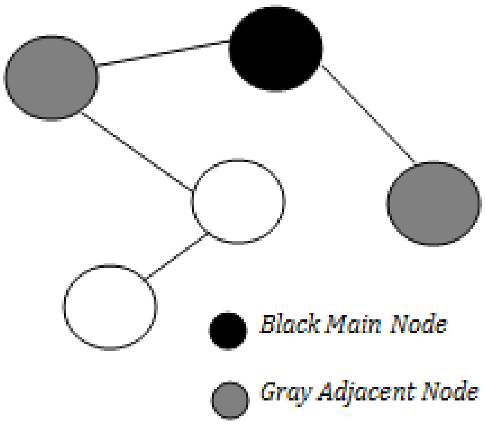


Figure 4. Node adjacent as a nominated

4. **Simulation Environment.** A network simulation (NS2) was performed to evaluate the performance of the proposed enhanced recovery routing between nodes in the network topology. A comparison of the simulation results of the OSPF protocol with and without our extension code was made. The evidence gathered by the NS2 simulation offered good support for the transmission data in the networks. At the physical and data-link layers, the researcher used the IEEE 802 Ethernet. During the simulation, each node checks its adjacent one to start creating its backup route. In addition, we configured layer 2 to detect the failure when the signal was lost to each node's port. The simulation was repeated ten times and an average calculated. The packet size was 512KB and the bit rate was set to 2MB/s. A traffic rate of 200KB/s was generated from the source node to the destination during the simulation. Based on the parameters in Table 1, we have shown the simulation results in the form of line graphs in the following section.

Each graph illustrates a comparison between the OSPF protocol operating with and without computing a backup path while varying the number of nodes in each topology. Before evaluating the performance issues of network topologies with respect to computing a backup path over different networks, it is important to determine what network parameters could affect the QoS of the streamed video traffic. Here the research focuses on three parameters, which may better reveal the effect of video traffic techniques.

- Packet loss ratio: the packet ratio between dropped and sending data packets.
- Average end-to-end delay: the average time between transmission and arrival data packets.

In order to evaluate the effect of the density of mobile nodes, it is necessary to know the number of nodes in each topology. This is because the number of node densities enables to discover and maintain new paths according to the node numbers. When the number of nodes increases then more alternative paths will become available. When the number of nodes is reduced the number of alternative paths will diminish; however, the computing of alternative paths will be completed in less time. In addition, a larger number of nodes lead to an increase in the probability of paths breaking compared to when the number of nodes is less. This means that nodes are more stable when they divide in a small area and thus the time for data transmission will become extended. We also measured the packet delay for different numbers of nodes in different network topologies.

5. **Performance and Analysis Evaluation.** For the transmission data between the nodes in on-peak and off-peak times, we have been monitoring the network throughout the day and as one can observe the traffic load is very high during on-peak times and then become reduced when the working day over for users. That means, the service during on-peak times will become degraded because of congestion in the network and the number of failures for the link or nodes will be high and QoS will be prejudiced.

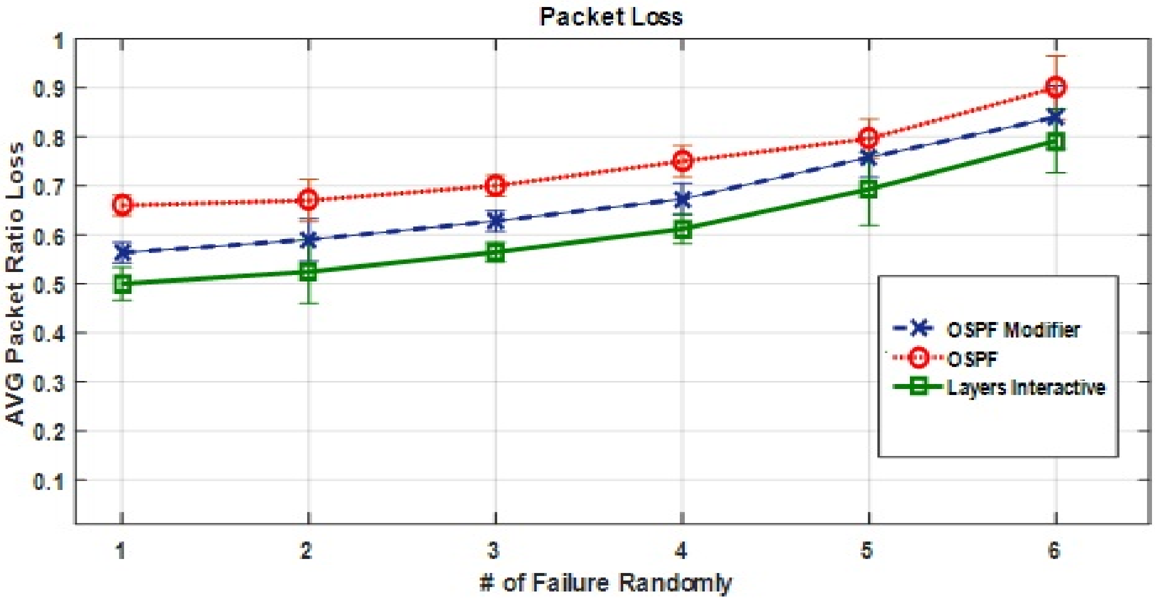


Figure 5. Avg. loss packet

When the destination node is further from the source node. Each one will retain its data packets in its buffer until it can find an available path to forward them for their destination. If a node receives packets that exceed the size of the buffer then it will start to drop them. In lower node density, the loss of packets will be less. This is because the detection failure will be faster and the number of hops between source and destination will be less. In addition, searching for and computing an alternative path will be faster than in high node density in different topologies. Hence, the incidence of rerouting will become reduced and improved when the detection of failure is by layer 2 and rerouting layer 3 and it will enhance our OSPF protocol.

In Figure 6, however, the hybrid mechanism makes the destination node receive the packet from source node in less time compared with alternative path and OSPF protocol with respect to the two ways that have been used for the rerouting mechanism. In the figure, we can see that the hybrid mechanism (virtual path) produces a stable line graph

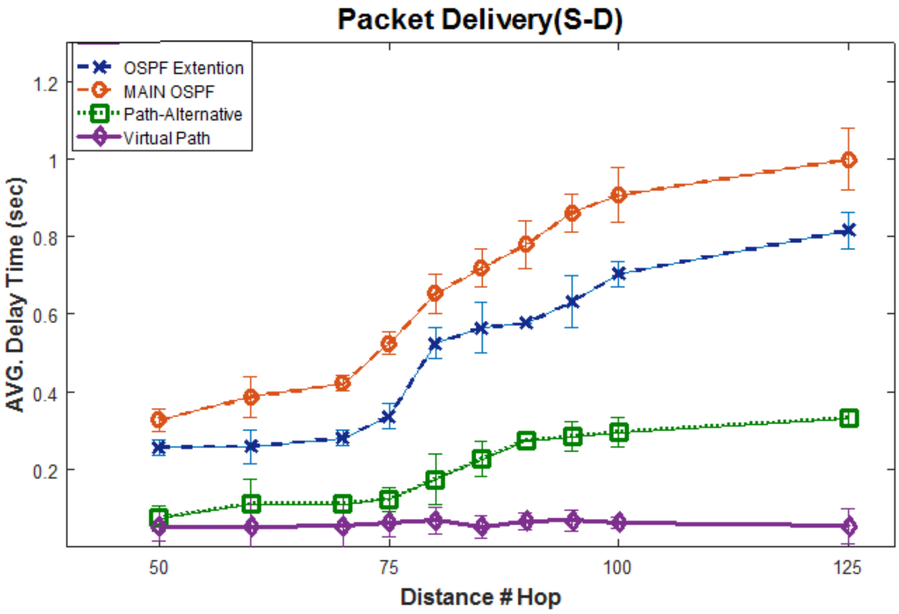


FIGURE 6. Delivering packet from source to destination

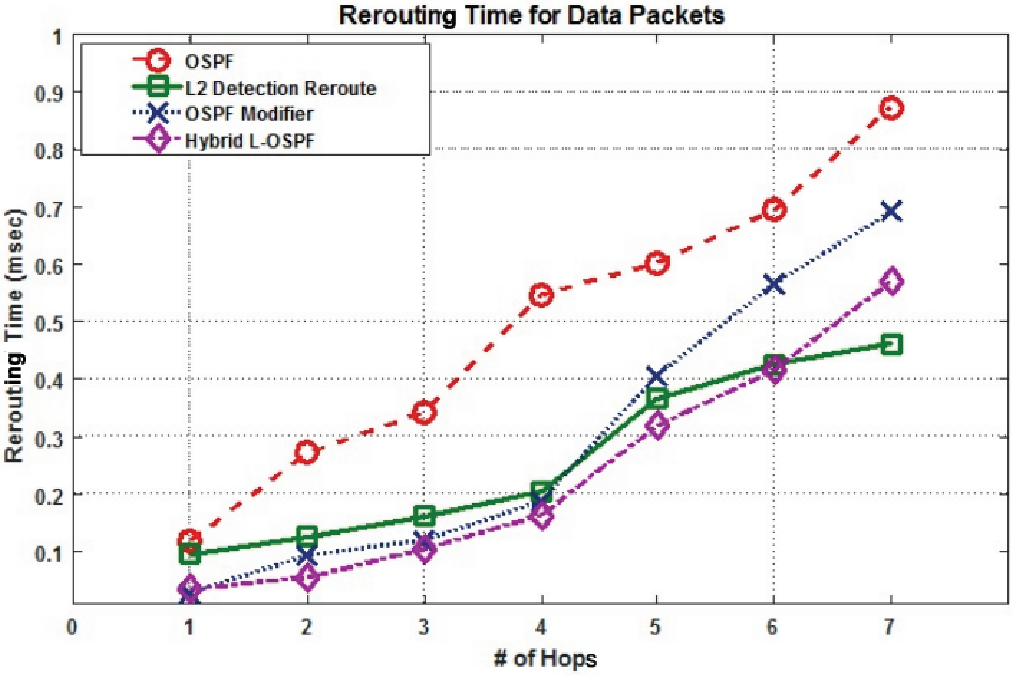


FIGURE 7. Rerouting traffic

because detection and the computation times are almost the same for all node cases. In addition, the detection and computing times are very short for the alternative path. Figure 7 shows that the rerouting time for the hybrid mechanism has improved in all cases. In some instances, however, layer 2 also produced good rerouting times because some links or node failures can come down for a period and then suddenly come up. In addition, rerouting time is dependent on the detection of failure and in itself proves that it has occurred. Failures can be confirmed by Hello packets that have already been configured from Table 1. The time for receiving data packets at their destinations needs to be in less time with respect to the type of traffic, such as, VOIP or video traffics.

According to Figure 6 we have shown when the main path between nodes is starting to be in a limited utilized with high congestion, then the proposed algorithm is showing how the data packets can switch from main path to an alternative virtual path between source and the demand node to keep the service up with high performance servicing. In the proposed technique we have tried to make the lower layers such as 2 and 3 be working together for increasing the time detection and used the alternative in the networks. The continuity index per second for packets arriving in different topologies is shown in Figure 8. The packets arriving after one second (including buffering time) will be eliminated. This figure indicates that the low number of nodes performs better compared to high ones for different topologies. The number of connections in low nodes makes the data packet service better even when the nodes or links will be down. The success of local recovery is improved, because it helps salvage the data packets by using the backup route. This means that the backup route has improved the continuity index for live streaming packets in all cases as expected with better performing for less number of nodes. This is because the congestion will be less according to the updated packets sent during the failure. In high density nodes, comfortable viewing is experienced for video traffic. The continuity index is reduced when the number of nodes increased. Bearing in mind the time the packets spend in the buffer and in updating the routing table. Figure 8 shows that broadcasting messages will increase overheads, which will affect the continuity index in the case of live streaming. Recovery, however, shows a high continuity index compared to the OSPF protocol in relation to congestion in the network and the ability of existing backup path to pass the packets when a loss of connection has occurred.

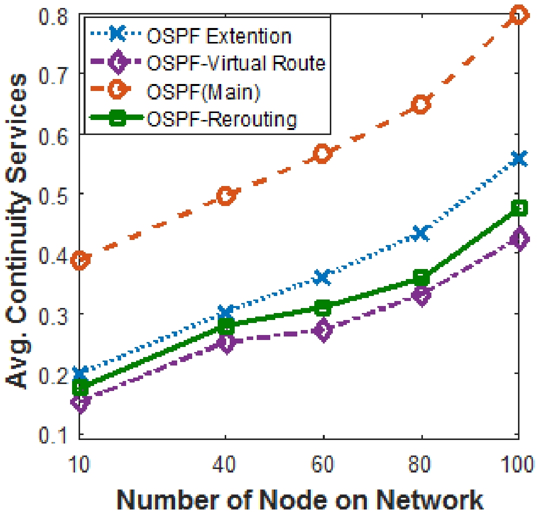


Figure 8. Continuity index

The load of a path is measured by the maximum load of its component links as shown in Figure 9. The figure shows that under high load condition the average loading of virtual rerouting paths is not the lowest compared with other techniques and routing protocols. However, the virtual rerouting shows that the load can be accepted regarding to the other results. The OSPF returns lower-loaded paths than OSPF-modifier and virtual rerouting. This is because both of them are sending some extra packets for computing the backup path. For computing the average path, we define stretch factor of a path.

$$P(x,y) \text{ as } \delta(x,y) = T(x,y)/C(x,y) \tag{4}$$

where $T(x,y)$: length of $P(x,y)$, $\delta C(x,y)$.

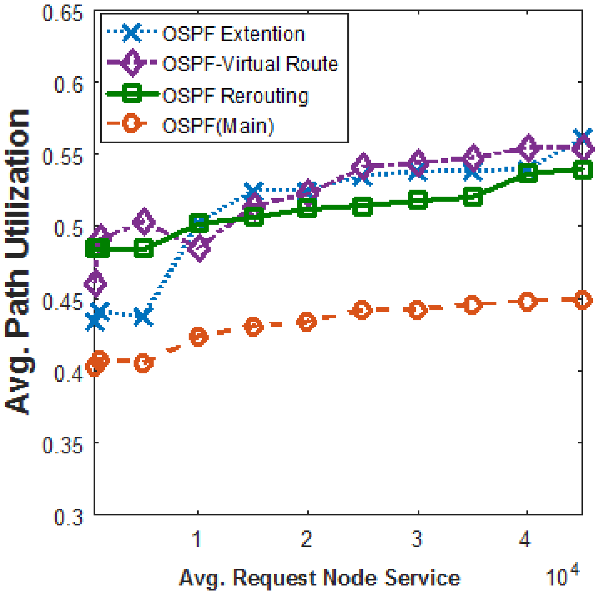


Figure 9. Avg. path load

The length of the shortest path returned by OSPF. As shown, we have the least average even at the highest network load conditions. Virtual rerouting paths have the highest. The least loaded shorter length OSPF paths returned under high load conditions demonstrates the stability of the algorithm.

6. **Conclusion.** Various provision schemas for IGP networks are proposed in this paper. IGP was the main focus of the research and the objective was to enable robust sensitive applications traffic such as video and VOIP over this infrastructure network. Different schemas and approaches were investigated and experimented with for that purpose, and the results show that if a provision mechanism is in place in the network, then it is possible to produce high QoS aimed to meet the first target of this thesis by investigating the issue of any problem raised during the transmission data in IGP network. Through link failure, high utilization or congestion shows and proves that any of these problem can lead to degraded network performance, and trigger such problems as local loops in the network. The study in this paper demonstrated an inter-cooperation between the provision schema and routing protocol in IGP network. Avoiding loops in the network was considered in the computed backup path when it is computed by source node. This is because a loop in the network can trigger two main problems: congestion in the network with increasing link utilization during the loop, and buffer size for the access node (in regard to the long time that the packets will be held in node buffers before beginning to be dropped). Hence, the occurrence of loops when failure occurs in the network leads to increased end-to-end delay of any sensitive traffic passing through the network, and increased packet loss. Motivated by the findings in this paper, a new algorithm was proposed and implemented in order to work with OSPF protocol to achieve fast serving for packet switching in the network. The algorithm finds an alternative path over source node to switch the traffic once any of the mentioned problem occurred along used path without the need to wait to reconnect or return back to the service in different time. This mechanism has a clear benefit in backbone networks where resources are available, and they can offer many alternative paths between source and destination. As demonstrated in this paper, losing connections

between nodes is a frequent phenomenon, which necessitates continuously updating the routing table for every change that occurs in the network.

Acknowledgment. We would like to thank AOU university and their supports and provide experiment tools and allow to exchange some ideas with the expert people.

REFERENCES

[1] A. Goldberg and T. Radzik, A heuristic improvement of the Bellman-Ford algorithm, *Applied Mathematics Letters*, vol.6, no.3, pp.3-6, 1993.

[2] R. Gargees, B. Morago, R. Pelapur, D. Chemodanov, P. Calyam, Z. Oraibi, Y. Duan, G. Seetharaman and K. Palaniappan, Incident-supporting visual cloud computing utilizing software-defined networking, *IEEE Trans. Circuits and Systems for Video Technology*, vol.27, no.1, pp.182-197, 2017.

[3] J. Hawkinson and T. Bates, *Guidelines for Creation, Selection, and Registration of an Autonomous System (AS)*, http://tools.ietf.org/html/rfc1930, 1996.

[4] T. H. Cormen, *Introduction to Algorithms*, The Massachusetts Institute of Technology, 2001.

[5] A. Tolba, Organizing multipath routing in cloud computing environments, *International Journal of Advanced Computer Science and Applications*, vol.8, no.1, pp.455-462, 2017.

[6] U. Black, *IP Routing Protocols: RIP, OSPF, BGP, PNNI and Cisco Routing Protocols*, Prentice Hall, 2000.

[7] A. Mishra, R. Jain and A. Durresi, Cloud computing: Networking and communication challenges, *IEEE Communications Magazine*, vol.50, no.9, 2012.

[8] D. Kliazovich, J. E. Pecero, A. Tchernykh, P. Bouvry, S. U. Khan and A. Y. Zomaya, CA-DAG: Modeling communication-aware applications for scheduling in cloud computing, *Journal of Grid Computing*, vol.14, no.1, pp.23-39, 2016.

[9] K. Gai, M. Qiu, H. Zhao, L. Tao and Z. Zong, Dynamic energy-aware cloudlet-based mobile cloud computing model for green computing, *Journal of Network and Computer Applications*, vol.59, pp.46-54, 2016.

[10] K.-T. Foerster, M. Parham and S. Schmid, A walk in the clouds: Routing through VNFs on bidirected networks, *Proc. of Algocloud*, 2017.

[11] E. Akin and T. Korkmaz, Routing algorithm for multiple unsplittable flows between two cloud sites with QoS guarantees, *International Conference on Computing, Networking and Communications (ICNC)*, pp.917-923, 2017.

[12] J. Moy, *OSPF: Anatomy of an Internet Routing Protocol*, Addison-Wesley Professional, 1998.

[13] U. Brandes, A faster algorithm for betweenness centrality, *Journal of Mathematical Sociology*, vol.25, no.2, pp.163-177, 2001.

[14] J. Moy, *Link-State Routing in Routing in Communications Networks*, Prentice Halls, http://www.faqs.org/rfcs/rfc2328.html, 1995.

[15] G. Malkin, *Rip Version 2 – Carrying Additional Information*, http://wiki.tools.ietf.org/html/rfc2453, 1994.

[16] A. Atlas and A. Zinin, Basic specification for IP fast reroute: Loop-free alternates, *Internet Engineering Task Force, Work in Progress*, draft-ietf-rtgwg-ipfrr-spec-base-03.txt, 2008.

[17] G. Iannaccone, C. Chuah, R. Mortier, S. Bhattacharyya and C. Diot, Analysis of link failures in an IP backbone, *Proc. of the 2nd ACM SIGCOMM Workshop on Internet Measurment*, pp.237-242, 2002.

[18] G. Iannaccone, C. N. Chuah, S. Bhattacharyya and C. Diot, Feasibility of IP restoration in a tier 1 backbone, *Network*, vol.18, no.2, pp.13-19, 2004.

[19] A. Markopoulou, G. Iannaccone, S. Bhattacharyya, C. N. Chuah and C. Diot, Characterization of failures in an IP backbone, *The 23rd Annual Joint Conference of the IEEE Computer and Communications Societies*, vol.4, pp.2307-2317, 2004.

[20] C. Labovitz, A. Ahuja, A. Bose and F. Jahanian, Delayed Internet routing convergence, *ACM SIGCOMM Computer Communication Review*, vol.30, no.4, pp.175-187, 2000.

[21] C. Labovitz, A. Ahuja and F. Jahanian, Experimental study of Internet stability and backbone failures, *The 29th Annual International Symposium on Fault-Tolerant Computing, Digest of Papers*, pp.278-285, 1999.

[22] P. Gill, N. Jain and N. Nagappan, Understanding network failures in data centers: Measurement, analysis, and implications, *Proc. of SIGCOMM*, 2011.

[23] M. Gjoka, V. Ram and X. Yang, Evaluation of IP fast reroute proposals, *The 2nd International Conference on Communication Systems Software and Middleware*, pp.1-8, 2007.

[24] S. Bryant, M. Shand and S. Previdi, *IP Fast Reroute Using Not-via Addresses*, draft-bryant-shand-ipfrr-notvia-addresses-03.txt, 2006.

[25] A. F. Hansen, T. Cicic and S. Gjessing, Alternative schemes for proactive IP recovery, *The 2nd Conference on Next Generation Internet Design and Engineering*, 2006.

[26] S. Nelakuditi, S. Lee, Y. Yu, Z. L. Zhang and C. N. Chuah, Fast local rerouting for handling transient link failures, *IEEE/ACM Trans. Networking*, vol.15, no.2, pp.359-372, 2007.

[27] Z. Zhong, Z. Nelakuditi, Y. Yu, S. Lee, J. Wang and C. N. Chuah, Failure inferencing based fast rerouting for handling transient link and node failures, *Proc. of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies*, vol.4, pp.2859-2863, 2005.

[28] F. du P. Calmon, J. M. Cloud, M. Medard and W. Zeng, *Multi-Path Data Transfer Using Network Coding*, US Patent 9,537,759, 2017.

[29] J. Hawkings, J. Wadham, M. Tranter, J. Telling, E. Bagshaw, A. Beaton, S.-L. Simmons, D. Chandler, A. Tedstone and P. Nienow, The greenland ice sheet as a hotspot of phosphorus weathering and export in the arctic, *Global Biogeochemical Cycles*, 2016.