Organizing Multipath Routing in Cloud Computing Environments

Amr Tolba[1,2]
[1]Computer Science Department, Community College, King Saud University, Riyadh 11437, Saudi Arabia
[2]Mathematics and Computer Science Department, Faculty of Science, Menoufia University, Shebin-El-kom 32511, Egypt

***Abstract*—One of the objectives of organizing cloud systems is to ensure effective access to remote resources by optimizing traffic engineering (TE) procedures. This paper considers the traffic engineering problem in a cloud environment by using a multipath routing technique. The multipath routing algorithm is used to identify the maximum number of disjoint paths in the graph which overcomes the problem in the junction area estimation process. So, the algorithm forms a plurality of non-overlapping and partially intersecting paths between any two nodes is proposed. Finally, the conditions for the formation of multipath virtual channels to ensure minimum build-time posts for the parallel transmission of its parts are also discussed.**

Keywords—Cloud Computing; Remote resource; Optimizing traffic engineering; Multipath routing; Disjoint paths; Parallel transmission

I. INTRODUCTION

Traffic engineering (TE) is an essential tool for the provision of reliable, differentiated, and fast network services. According to the Internet Engineering Task Force (IETF), TE is roughly signified as dealing with the aspect of network engineering pertaining to problems of performance optimization and evaluation of Internet Protocol (IP) networks. Moreover, TE often deals with traffic demand by mapping different types of a given network topology to reflect changing network conditions by adaptively reconfiguring its processes. It is better than Quality-of-Service (QoS) routing in the sense that TE normally aims at highly efficient operational networks while meeting particular constraints, whereas the major aim of QoS routing is to meet particular constraints of QoS for traffic flow from a given source to a destination.

An essential part of a multipath routing framework is optimal routing. In optimal routing, source-to-destination traffic is split at tactical points to allow the gradual altering of traffic along alternative paths. The main aim of optimal routing is to avoid traffic, particularly along paths that are the shortest in terms of packet transmission time. For increasing input traffic, alternative paths are used to avoid an overload along the shortest path. So, the multipath routing algorithm is used to identify the maximum number of disjoint paths in the graph which overcomes the problem in the junction area estimation process. Thus, this paper proposes an algorithm to form a plurality of non-overlapping and partially intersecting paths between any two nodes. The conditions for the formation of multipath virtual channels to ensure minimum build-time posts for the parallel transmission of divided data parts like transmission, routing path are also discussed.

This work considers the traffic engineering problem in a cloud-based environment by using the multipath routing technique. It is expected that multipath routing will improve the flow quality of streaming in cloud environments, without particularly considering the short flows with dynamic routing. In terms of resource control, multipath routing can direct strong traffic oscillations, route flapping and excessive signaling message overhead and so on, taking an account of topology changes due to the dynamic routing, despite its potential benefits and use in static routing. Outdated information routed by packets can direct to load oscillations; thus, the objective of TE can be attained by routing traffic demands along different types of multiple paths.

The rest of this paper is organized as follows: section II discusses the studies related to the research work, section III presents the proposed algorithm, finding the maximum number of disjoint paths, the protocol for finding the minimum of the junction area of the graph and the conditions for the formation of multipath virtual channels to ensure the minimum build-time posts for parallel transmission of its parts, section IV discusses and analyzes the simulation results and finally, section V concludes this work.

II. RELATED WORK

Cloud computing allows users to worry less about understanding the details of infrastructures, and focus on optimizing the appropriate services and resources in the computational complexity. At the same time, the application of parallel transportation management systems has become a popular topic of research in the field [1, 2].

Cloud computing platforms that provide Infrastructure as a Service are a form of virtual machines (VMs) for users, and are based on shared infrastructure, hardware, and software. At present, modern network technologies of clusters, grids, and cloud computing [3, 4] are widely used in virtual private networks (VPNs) [5-7], which are built, as a rule, on global computer networks. Virtualization is carried out at different levels: server, storage, and network. Virtualization on local networks forms a private cloud as a VPN with a star or tree topology [5-10]. At the same time, fat-tree topologies or switch-centric networks are becoming critical components of data center networks (DCN). This topology is known as a non-blocking multi-path network that utilizes several equal-cost paths between adjacent layers to help eliminate bandwidth bottlenecks in the core layers, in addition to supporting large-scale networks consisting of several thousand physical servers [11-13].

In a report, He et al. [14] emphasized the challenging task of network management as they grow in size and complexity. They reviewed several optimization techniques that have been applied to network management problems. By realizing that optimization problems in network management are induced by assumptions adopted in the protocol design, they argued that protocols should first be designed with optimization in mind, rather than optimizing existing protocols and principles by changing architectures. Maguluri et al. [15] used a stochastic model for load balancing and scheduling in cloud computing clusters. They assumed that jobs arrive at a cluster according to a stochastic process, and utilized virtual machines (VM) with a focus on resource allocation problems and scheduling VM configurations. They primarily contributed to the development of frame-based non-preemptive VM configuration policies, and claimed that these policies are nearly throughput optimal, in contrast to the widely used best-fit policy that is known to be throughput suboptimal. Their simulations indicated that long frame durations are throughput perspective by providing satisfactory delay performance. Recently, Manjur et al. [16] have proposed a unified storage allocation scheme (USAS) for VM. The proposed algorithm is able to allocate space dynamically according to the requests of users (e.g., OS images) and employs storage partitioning theory.

Similar to tele-traffic engineering methodology in heterogeneous networks (HetNets) proposed by Saied et al. [17], Chiesa et al. [18] considered the standard model of traffic engineering (TE) with equal-cost multipath (ECMP) and proved that "ECMP can provably achieve optimal traffic flow for the important category of CLOS datacenter networks" in contrast to the known approximation. They also addressed a shortcoming in ECMP in the suboptimal routing of large flows by presenting a suitable algorithm for scheduling with provable approximation, thereby shedding new light on the performance of TE with ECMP.

Similarly, Liu et al. [19] considered multipath routing specific to communication networks from a traffic engineering perspective in a multi-commodity setting through linear programming. They showed that a multipath measure (MPM) is zero or close to zero under certain traffic conditions and topological structures, hence implying that there is limited multipath gain compared to that in single-path routing. For the all-pair traffic case, multipath routing was observed to be advantageous for small networks. They claimed that the effective distribution of traffic in multipath routing is significantly better over network resources, which is believed to be somewhat in opposition "load sharing."

In another report by Wang et al. [20], AMPLE, based on offline link weight optimization, was introduced. Using this, they were able to monitor network dynamics at short timescales, thereby coping almost optimally with unpredictable traffic dynamics. They also formulated a new proposal for achieving superior service quality and overall network performance in IP networks with reference to real network topologies and traffic traces.

Gojmerac et al. [21, 22] proposed another algorithm called Adaptive Multipath Routing (AMP) for dynamic traffic engineering on the Internet, with continuous load distribution within a network domain, hence offloading congested links in real time. They reviewed several methods and algorithms in this context, and presented important areas of application of AMP for emerging networking architectures. This finding was based on their earlier work, where AMP was used within autonomous systems.

III. Proposed Algorithm

A survey of the literature reveals that optimization theories need to be designed in order to develop a better organizing process suitable for multipath routing in cloud computing environments, for the analysis and design of various components of traffic management to realize an optimal and versatile traffic engineering protocol. Therefore, one of the main objectives of organization in cloud systems is to create effective access to remote resources by optimizing the procedures of TE for transmitting data via cloud computing. The construction of multipath traffic consists of the following tasks:

1) Formation of a plurality of paths with predetermined QoS parameters.

2) Organization of multipath virtual channels focusing on data transfer involving different types of traffic.

3) Management of the transfer of information.

Therefore, this paper addresses the traffic engineering problem in cloud environments using the multipath routing technique for data transformation via a cloud structure. An algorithm is proposed to solve the problem of finding the maximum number of disjoint paths, and a protocol for finding the minimum of the junction area of the graph is presented. Finally, the conditions for the formation of multipath virtual channels to ensure minimum build-time posts for parallel transmission of its parts are also discussed.

A. Terms of paths adjacency

Data transformation in a cloud structure is carried out by using multipath routing, such as non-intersecting paths, and paths that have common nodes or links. Paths with common non-adjacent nodes are called intersecting paths, and those with common communication channels are called adjacent tracks. The choice of a set of paths depends on the required QoS information to be transmitted and the efficiency of the information transmission network. This involves considering ways of formulating adjacency conditions for an arbitrary graph $G = (V, E)$.

Lemma 1. Path $P_i =(V_i,E_i)$ and $P_j =(V_j,E_j)$ do not intersect under the following condition:

$$(V_i / (V_{0i} \cup V_{ei})) \cap (V_j / (V_{0j} \cup V_{ej})) =\varnothing, \quad (1)$$

where V_i and V_j are sets of vertices for paths $P_i =(V_i,E_i)$ and $P_j =(V_j,E_j)$, v_{0i} and v_{ei} are the initial and final points of path $P_i =(V_i,E_i)$, and v_{0j} and v_{ej} are the initial and final points of path $P_j =(V_j,E_j)$.

Lemma 2. Paths $P_i =(V_i,E_i)$ and $P_j =(V_j,E_j)$ intersect when

$$(V_i / (V_{0i} \cup V_{ei})) \cap (V_j / (V_{0j} \cup V_{ej})) \neq\varnothing, \text{ and } E_i \cap E_j=\varnothing. \quad (2)$$

Lemma 3. Paths $P_i =(V_i,E_i)$ and $P_j =(V_j,E_j)$ are adjacent when

$(V_i / (V_{0i} \cup V_{ei})) \cap (V_j / (V_{0j} \cup V_{ej})) \neq \varnothing$, and $E_i \cap E_j \neq \varnothing$. (3)

The coefficient of intersection k_{ri}, and path $P_i=(V_i,E_i)$ can be determine from the ratio of N_r vertices in common with other ways to a variety of N_{Pi} own vertices path P_i (V_i,E_i), i.e., $k_{ri}=N_r/N_{Pi}$. At $k_{ri}=0$ leading to a condition that path does not intersect with other paths, while at $N_r=1$, the path partially overlaps.

Accordingly, under the adjacency factor k_{ci}, paths $P_i=(V_i,E_i)$ will be the ratio of N_c common ribs to a plurality N_{Ei} all the edges of the path, i.e., $k_{ci} = N_c/N_{Ei}$. At $k_{ci} = 0$, leading to a condition that the path is not adjacent, while at $N_c = 1$, the path considered as a weak bound path.

B. Determining the minimal set of junctions

The maximum number of paths depends on network topology, the degree of the vertices of the network, and the set of values k_{ri} and k_{ci}. The maximum number of disjoint paths between two vertices v_i and v_j is determined by a graph of the minimum set of joint vertices $V_S = V/(V_1 \cup V_2)$, i.e., the minimum set of vertices whose removal divides the graph $G=(V,E)$ into two subgraphs: $G_1=(V_1,E_1)$ and $G_2=(V_2,E_2)$. In this case, $v_i \in V_1$, and $v_j \in V_2$,. Sets $V_1= V/(V_2 \cup V_S)$ and $V_2= V/(V_1 \cup V_S)$.

Determining the minimum set of junctions can significantly reduce the complexity involved in finding the set of disjoint paths for known combinatorial algorithms, such as Dijkstra's algorithm. In the formation of k paths, the complexity incurred is $O(kN^2)$, where N is the number of nodes in the network. In this case, the paths between nodes $v_i \in V_1$ and $v_j \in V_2$ are formed in the subgraphs $G_1=(V_1,E_1)$ and $G_2=(V_2,E_2)$, at first from the top v_i to the vertices of set V_S, and then from these vertices to vertex v_j. The time complexity of the search for k disjoint paths in subgraph $G_1=(V_1,E_1)$, by using Dijkstra's algorithm, is $O(kN_1^2)$, where N_1 is the set of subgraph vertices $G_1=(V_1,E_1)$. Accordingly, the time complexity of the search for k disjoint paths in subgraph $G_2=(V_2,E_2)$ is $O(kN_2^2)$ or $O(k (N-(N_1+ k))^2)$. For example, when $N= 90$ and $k=10$, the complexity of the formation of 10 direct routes between two vertices is O(81000). As the graph divides $G=(V,E)$ using a minimal set of junctions with $N_1= N_2$, the subgraph with 40 vertices and $k=10$ will have a complexity of $O(2kN_1^2)$ = O(32000). In the latter case, the complexity is less by about 2.5 times than $N_1= N_2$ condition.

Thus, the problem of finding the maximum number of non-overlapping or partially overlapping paths can be reduced to the problem of finding a minimum set of junction graphs with the subsequent formation of disjoint paths to the heights of the minimum set of junctions. This reduces the complexity of the algorithm to form a plurality of disjoint paths.

The proposed algorithm determines the minimal set junction based on the procedure of forming a junction between two subgraphs $G_1=(V_1,E_1)$ and $G_2=(V_2,E_2)$ of graph $G=(V,E)$, where the number of N_1 vertices in subgraph $G_1=(V_1,E_1)$ varies from 1 to $(N-1)$. This is a result of sequentially generating several sets V_S, including the selected V_{Smin} with minimum power h_{Smin}. Forming a plurality of junctions in subgraph $G_1=(V_1,E_1)$ will help distinguish between internal and boundary vertices.

Vertices of set $V^i_1= \{v_i| i=1,2,\ldots,n\}$ in subgraph $G_1(V_1,E_1)$ not adjacent to the vertices separating sets V_s will be referred to as internal vertices. For a set of internal vertices $V^i_1 \subset V_1$, $E^i =\{ v^i_{k,j}| v_k \in V_1, v_j \in V_1\}$. Accordingly, edge $e^i_{k,j}$ is an internal edge.

Vertices of set $V_b= \{v^b_i| i=1,2,\ldots,n\}$, adjacent to vertices of set V_s, are called the boundary vertices of subgraph $G_1(V_1,E_1)$. Accordingly, edge $e^b_{k,j}$ is the boundary edge. In the set of boundary vertices $V_b \subset V_1$, and the edges are belongs to the $E^b =\{ e^b_{k,j}| v_k \in V_1, v_j \in V_s\}$ in which the internal S^i_i and external S^b_i vertex has the degree value is v_i.

The number of tops of internal edges v_k determines the internal degree S^i_k. In turn, an edge $e^b_{k,j}=\{ v_k \in V_1, v_j \notin V_1\}$ is external to subgraph $G_1(V_1,E_1)$, here the vertex $v_k \in V_c$. The number of external edge tops v_k defines the outer degree S^b_k.

The process of determining the minimum set of junctions V_{Smin} involves the successive formation of set of vertices V_S, and determining V_{Smin}:

Begin

1. *From the vertices adjacent to the initial vertex v_i, a set of vertices V_S is formed, which in this case is V_{Smin}.*
2. *A plurality of adjacent vertices is included in subgraph $G_1 = (V_1,E_1)$.*
3. *A new set of vertices $V_1= V_1+ V_S$ of subgraph $G_1 = (V_1,E_1)$ is generated.*
4. *A new set of boundary vertices V^o_1 is formed.*
5. *On the basis of vertices $v_i \notin V_1$ adjacent to the vertices of set V^o_1, vertex set V_S is formed.*
6. *The power h_S of the vertex set V_S is calculated.*
7. *The power h_S of set V_S is compared with power h_{Smin} of set V_{Smin}. If $h_{Smin} > h_S$, the set of junctions V_S becomes V_{Smin}.*
8. *The graph $G_2 = (V_2,E_2)$ is formed with a new set of vertices $V_2= V_2/V_S$.*
9. *If $V_2 \neq \{v_i\}$ return to Step 2.*

End

The process of determining the minimum set of junctions in the formation of a path between vertices v_0 and v_{18} is shown in Fig. 1, and consists of the following steps:

1. *For subgraph $G_1(V_1,E_1)$ consisting of a single vertex v_0, the set of vertices $V_1=\{ v_0 \}$.*
2. *The set of internal ribs $E^i =\varnothing$, as subgraph $G_1=(V_1,E_1)$, contains only one vertex v_0.*
3. *The set of external ribs $E^0= \{e^o_{0,1},e^o_{0,2},e^o_{0,3}\}$, representing the external degrees of each vertex v_0, is $S^o_0 =3$.*
4. *The vertices $\{v_1,v_2,v_3\}$ are a plurality of separated vertices $V_s=\{v_1,v_2,v_3\}$ between subgraphs $G_1=(V_1,E_1)$ and $G_2=(V_2,E_2)$, where $V_2= V_0/(V_1 \cup V_s)$, and the original graph $G=(V,E)$.*
5. *$V_{Smin} = V_S$; $h_{Smin}= h_S=3$.*
6. *Subgraph $G_1=(V_1,E_1)$ with set of vertices $V_1=\{ v_0, v_1, v_2, v_3\}$ is formed.*
7. *The set of boundary vertices $V^o_1= \{v_1, v_2, v_3 \}$.*
8. *Of the vertices $\{v_4,v_5,v_6,v_7,v_8,v_9\}$, adjacent to the set of boundary vertices V^o_1, $\{v_4,v_5,v_6,v_9\}$ are internal, since they are not associated with the vertices of subgraph $G_2=(V_2,E_2)$.*
9. *In this case, $V_S=\{ v_7, v_8\}$; $h_S=2$, and $V_{Smin}=\{ v_7, v_8\}$.*

10. *V^o_1={ v_7, v_8}.*
11. *The external vertices adjacent to vertex set V^o_1 are the vertices {v_{10},v_{11},v_{12},v_{13}}, which form V_S c h_S = 4.*
12. *Vertex $v_{14} \in V_2$ is directly connected to vertex $v_{10} \in V_S$, forming V_S= {v_{14},v_{11},v_{12},v_{13}}; h_S=4.*
13. *Vertex $v_{16} \in V_2$ is directly connected to vertex $v_{13} \in V_S$, forming V_S= {v_{14},v_{11},v_{12},v_{16}}; h_S=4.*
14. *Vertex $v_{15} \in V_2$ is directly connected to vertex $v_{12} \in V_S$, forming V_S= {v_{14},v_{11},v_{15},v_{16}}; h_S=4.*
15. *Vertex $v_{17} \in V_2$ is directly connected to vertex $v_{16} \in V_S$, forming V_S= {v_{14},v_{11},v_{15},v_{17}}; h_S=4.*
16. *V_2={v_{18}} The process of forming V_S finishes here. V_{Smin}={ v_7, v_8}. In this case, the maximum number of disjoint paths between vertices v_0 and v_{18} is 2.*

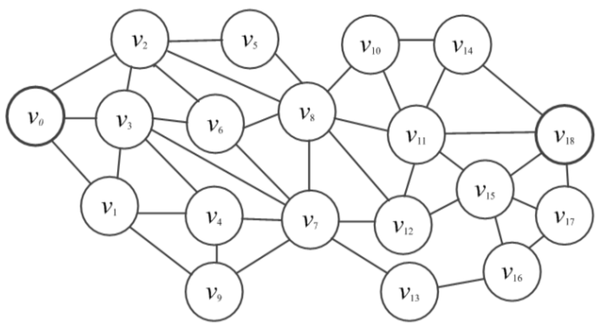

Fig. 1. Information transmission network graph

C. Determining plurality of disjoint paths

It should be noted that the initial vertex between v_i and the vertices of the set junction V_{Smin} may contain several disjoint paths, the number of which is greater than or equal to the cardinality of V_{Smin}. Between the vertices of junction set V_{Smin} and final vertex v_j, there may also be several disjoint paths.

In order to avoid operation directed enumeration characteristic of combinatorial algorithms of ways, a streaming algorithm to form paths from one node to multiple nodes on the basis of the "branch and bound" method is proposed. At the initial stage, the decision tree consists of primary vertices, e.g., vertices v_0 (see Fig. 1) and related vertices V_b = {v_1, v_2, v_3}, which in this case are boundary vertices of subgraph G_1= (V_1,E_1). Vertex v_0 refers to a set V_0 of internal vertices of subgraph G_1= (V_1,E_1). In forming paths in a set V_0, every time a vertex is added $v_i \in V_b$, , having fewer external branches as compared to other boundary vertices.

Accordingly, to a set V_b, vertex v_j is added adjacent to vertex v_i, with minimal external degree S^b_j. Thus, a decision tree is constructed from the root in vertex v_0 until it has all disjoint paths to a given node.

Given this notation, the algorithm to form a plurality of paths from vertex v_i to the vertex of a given set V_z of vertices is as follows:

Begin

1. *Form the initial set V_0 ={v_i} of internal vertices of subgraph G_1=(V_1,E_1).*
2. *Form a set of boundary of vertices V_b= {v_j| j=1,2,…,k}, which in this case is a set of vertices adjacent to vertex v_i.*
3. *For j=1 to k, specify path P_j= { v_i, $v_j \in V_b$ }.*
4. *For subgraph G_1(V_1,E_1), form the set of paths W_1={ P_j }.*
5. *Of the vertices $v_j \in V_b$, define vertex v_m with the minimal external degree S^b_m .*
6. *Move vertex v_m to the set of internal vertices, $v_m \in V_0$*
7. *Form a subgraph G_1(V_1,E_1) where V_0= $V_0 \cup v_m$.*
8. *If, among vertices v_i, there is no vertex $v_k \in V_z$ adjacent to vertex v_m, go to Step 9. If, among vertices v_i, there is vertex $v_k \in V_z$ adjacent to vertex v_m, the formation of path P_i to vertex v_k concludes.*
9. *Path P_i is added to the set of paths.*
10. *If a set of external vertices $V_b \neq \varnothing$, go to Step 4.*

End

As an example, consider forming a plurality of paths between vertex v_0 and vertices v_7 and v_8 (Fig. 1), as follows:

Begin

Step 1: Initial border set: {V_1 V_2 V_3} / form a plurality of boundary nodes for vertex v_0 */*

Step 2: Paths: {V_0 V_1} {V_0 V_2} {V_0 V_3} / form paths from vertex v_0 to vertices v_3,v_2 ,v_1 */*

Step 3: Paths: {V_0 V_1} {V_0 V_2 V_8} {V_0 V_3} / the formation of final path **P_1= {v_0,v_2,v_8}** */*

Step 4: Forming new border set: {V_1 V_3} / a new set of boundary nodes */*

Step 5: Paths: {V_0 V_1} {V_0 V_2 V_8} {V_0 V_3 V_7} / formation of final path **P_2= {v_0,v_3,v_7}**. */*

Step 6: Forming new border set: {V_1} / a new set of boundary nodes */*

Step 7: Selecting node V_1 (counter=1) / selection boundary vertex with the minimum value of external degree */*

Step 8: Selecting node V_9 (counter = 2) / selection of external vertex with the minimum value of external degree */*

Step 9: Paths: {V_0 V_1 V_9} {V_0 V_2 V_8} {V_0 V_3 V_7} / forming a path from vertex v_0 to vertex v_9 */*

Step 10: Forming new border set: {V_9 V_2 V_3} / a new set of boundary nodes */*

Step 11: Paths: {V_0 V_1 V_9 V_7} {V_0 V_2 V_8} {V_0 V_3 V_7} / formation of final path **P_3= {v_0,v_1,v_9,v_7}**. */*

Step 12: Border set: = $\varnothing$ / a new set of boundary nodes =$\varnothing$ */*

End

The process of forming disjoint paths is shown in Fig. 2. The second step of the algorithm generates a path (Fig. 2a) between the initial vertex and adjacent vertices. The number of such paths is the degree of the initial vertex. In Step 9 of the algorithm, paths are formed (Fig. 2b) from the initial vertex v_7 to vertices v_7, v_8, and v_9. The algorithm generates a plurality of disjoint paths (Fig. 2c) from the initial vertex to the ends of vertices. Thus, between vertex v_0 and vertices v_7 and v_8 are formed the following paths: P_1= {v_0,v_2,v_8}; P_2= {v_0,v_3,v_7}; P_3= {v_0,v_1,v_9,v_7}.

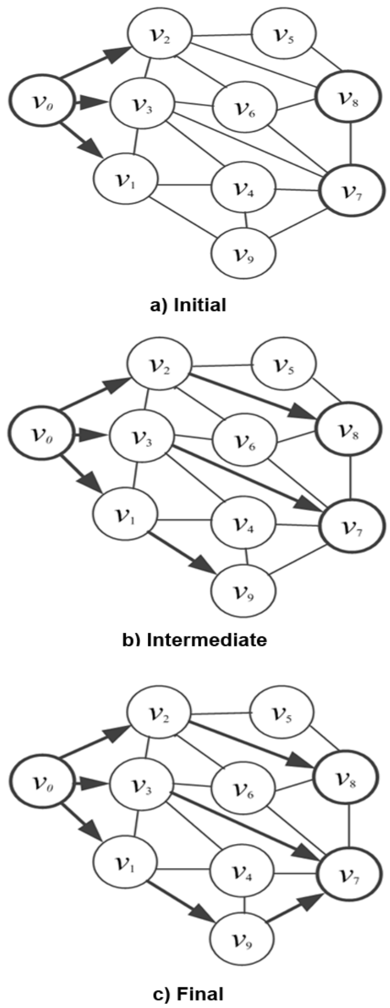

Fig. 2. Steps to form a plurality of disjoint paths

A characteristic feature of this algorithm is that it forms a set of paths according to predetermined criteria for optimal QoS. In this case, the length L_i of path P_i—namely P_1= P_2=2; P_3=3.

The formation of the set of paths between the boundary nodes (v_7,v_8) and final vertex v_{18} is carried out in a similar manner, starting with the final vertex:

Begin

Step 1: Initial border set: {V_{11} V_{14} V_{15} V_{17}} / forms a plurality of boundary nodes for vertex v_{18} */*

Step 2: Paths: {V_{18} V_{11}} {V_{18} V_{14}} {V_{18} V_{15}} {V_{18} V_{17}} / forms paths from vertex v_{18} to vertices v_{11},v_{14},v_{15},v_{17} */*

Step 3: Selecting node V_{11} (counter=2) / selection of external vertex adjacent to final vertex v_8 */*

Step 4: Paths: {V_{18} V_{11} V_8} {V_{18} V_{14}} {V_{18} V_{15}} {V_{18} V_{17}}/ formation of the final path* ***P_4= {v_{18},v_{11},v_8}****. */*

*Step 5: Selecting node V_{14} (counter=1) /*selection of the next vertex with the minimum value of external degree */*

Step 6: Paths: {V_{18} V_{11} V_8} {V_{18} V_{14} V_{10}} {V_{18} V_{15}} {V_{18} V_{17}} / forming path from vertex v_{18} to vertex v_{14} */*

Step 7: Forming new border set: {V_{10} V_{15} V_{17}}/ a new set of boundary nodes */*

Step 8: Paths: {V_{18} V_{11} V_8} {V_{18} V_{14} $V_{10}$$V_8$} {$V_{18}$ V_{15}} {V_{18} V_{17}} / formation of the final path* ***P_5= {v_{18},v_{14},v_{10},v_8}****. */*

Step 9: Forming a new border set: {V_{15} V_{17}}/ a new set of boundary nodes */*

Step 10: Selecting node V_{17} (counter=1) / selection of external vertex with the minimum value of external degree */*

Step 11: Paths: {V_{18} V_{11} V_8} {V_{18} V_{14} $V_{10}$$V_8$} {$V_{18}$ V_{15}} {V_{18} V_{17} V_{16}} / forming a path from vertex v_{18} to vertex v_{16} */*

Step 12: Forming new border set: {V_{15} V_{16}}/ a new set of boundary nodes */*

Step 13: Selecting node V_{13} (counter = 1) / selection of an external vertex with the minimum value of external degree */*

*Step 14: Paths: {V_{18} V_{11} V_8} {V_{18} V_{14} $V_{10}$$V_8$} {$V_{18}$ V_{15}} {V_{18} V_{17} $V_{16}$$V_{13}$} forming path from vertex v_{18} to vertex v_{13} */*

Step 15: Forming new border set: {V_{13} V_{15}}/ a new set of boundary nodes */*

Step 16: Selecting node V_{13} (counter = 1) / selection of external vertex adjacent to final vertex v_8 */*

Step 17: Paths: {V_{18} V_{11} V_8} {V_{18} V_{14} $V_{10}$$V_8$} {$V_{18}$ V_{15}} {V_{18} V_{17} $V_{16}$$V_{13}$$V_7$} / formation of the final path* ***P_6= {v_{18},v_{17},v_{16},v_{13},v_7}****. */*

*Step 18: Forming new border set: {V_{13}} /*a new set of boundary nodes*/*

Step 19: Selecting node V_{13} (counter = 1) / selection of external vertex with the minimum value of external degree */*

*Step 20: Paths: {V_{18} V_{11} V_8} {V_{18} V_{14} $V_{10}$$V_8$} {$V_{18}$ $V_{15}$$V_{13}$} {$V_{18}$ V_{17} $V_{16}$$V_{13}$$V_7$} /*forming path from vertex v_{18} to vertex v_{13} */*

Step 21: Forming new border set: {V_{13}}/ a new set of boundary nodes */*

*Step 22: Paths: {V_{18} V_{11} V_8} {V_{18} V_{14} $V_{10}$$V_8$} {$V_{18}$ $V_{15}$$V_{13}$$V_7$} {$V_{18}$ V_{17} $V_{16}$$V_{13}$$V_7$} /*formation of the final path P_7 = {v_{18},v_{15},v_{13},v_7}. */*

Step 23: Border set: = ∅ / a new set of boundary nodes =∅ */*

End

As a result, between the boundary vertices (v_7, v_8) and final vertex v_{16} are formed the following disjoint paths: P_4 ={v_{16},v_{11},v_8}, P_5 = {v_{16},v_{12},v_{10},v_8}, P_6 = {v_{16},v_{15},v_{14},v_{12},v_8}, and P_7 = {v_{16},v_{13},v_{10},v_8}. These, together with paths P_1= {v_0,v_2,v_8}, P_2= {v_0,v_3,v_7}, and P_3= {v_0,v_1,v_9,v_7}, can form two disjoint paths between vertices v_0 and v_{16}, and 12 partially overlapping paths (Fig. 3). The shortest paths are disjoint paths (P_1 + P_4) ={v_0,v_2,v_8, v_{11},v_{16}} for length $L_{1,4}$ = 4, and path (P_2 + P_7) ={v_0,v_3,v_7, v_{10}, v_{13},v_{16}} for length $L_{2,7}$ = 5.

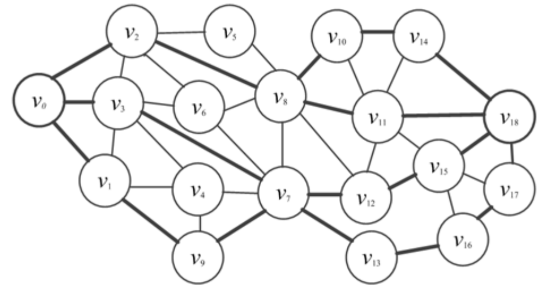

Fig. 3. The set of partially overlapping paths

The advantage of this algorithm is that it eliminates the possibility of crossing paths, which arises in the case of the sequential formation of paths between nodes. For example, between vertex v_0 and v_7 (see Fig. 3), there are the following disjoint paths: $P_1 = \{ v_0, v_1, v_9, v_7 \}$ by length $L_1 = 3$, $P_2 = \{ v_0, v_3, v_7 \}$ by length $L_2 = 2$, and $P_3 = \{ v_0, v_2, v_6, v_7 \}$ by length $L_1 = 3$. Between vertex v_0 and v_8, there are the following disjoint paths: $P_4 = \{v_0, v_2, v_8, v_7\}$, $L_4 = 3$, $P_5 = \{ v_0, v_2, v_8 \}$, $L_5 = 2$, and $P_6 = \{v_0, v_3, v_7, v_8\}$, $L_6 = 3$. In this set, path P_4 is excluded from of the set of disjoint paths between vertex v_0 and set of vertices $V_{Smin} = \{v_7, v_8\}$ because it includes vertices v_7 and v_8. The sets of non-intersecting paths are $M_1 = \{ P_1, P_2, P_5\}$ and $M_2 = \{ P_1, P_3, P_6\}$. The sets of path *M1* comprises P_1 by length $L_1=3$ and two paths P_2 and P_5 by length $L_5=2$ and $L_2=2$. The sets of path *M2* contains all the same path length equal 3.

Between vertex $v_7 \in V_{Smin}$ and v_{18}, there are paths $P_7 = \{v_7, v_{13}, v_{16}, v_{17}, v_{18}\}$ and $P_8 = \{v_7, v_{12}, v_{15}, v_{18}\}$. Between vertex $v_8 \in V_{Smin}$ and vertex v_{18}, there are paths $P_9 = \{v_8, v_{12}, v_{15}, v_{18}\}$, $P_{11} = \{v_8, v_{11}, v_{18}\}$, and $P_{10} = \{v_8, v_{10}, v_{14}, v_{18}\}$. Path P_9 maximally intersects with path P_8, and is excluded from the set disjoint paths between vertices $V_{Smin}=\{v_7, v_8\}$ and vertex v_{18}. Thus, it may be formed by the sets of the following disjoint paths: $M_3 = \{ P_7, P_8, P_{10}, P_{11}\}$ and $M_4 = \{ P_7, P_9, P_{10}, P_{11}\}$. Both sets contain paths of different lengths $L_7 = 4$, $L_8 = 3$, $L_9 = 3$, $L_{10} = 3$, and $L_{11} = 2$.

Thus, depending on the desired transmission quality, QoS parameters between vertices v_0 and v_{18} may form the shortest path: for example, the path $\{P_5, P_{11}\}$ of length $L_{5,11} = 4$. The longest path is $\{P_1, P_7\}$ with a length of $L_{1,7} = 7$. In organizing, a parallel transmission path may be formed $\{P_5, P_{10}\}$, $\{P_2, P_8\}$, and $\{P_6, P_{11}\}$, of length 5. In this case, parallel to the transmitted part, the data will be collected without additional delay in the receiving node.

D. Determining the parallel transmission of paths

In general, between vertices v_0 and v_{18} may be formed the following set of paths: $M_{13} = \{ M_1, M_3\}$, $M_{14} = \{ M_1, M_4\}$, $M_{23} = \{ M_2, M_3\}$, and $M_{24} = \{ M_2, M_4\}$. Each of the paths sets M_1 and M_2 is connected to one of a plurality of paths M_1 or M_2.

The presence of a sufficiently large set of all possible paths makes easier the process of multipath transmission traffic. During the multipath transmission the QoS parameter has been maintained using the nature of the traffic requirements in the multipath virtual channel.

For example, if the data is divided into different pieces and those data has been transferred in to the parallel route for managing the data transfer delay like minutes IGRP and EIGRP, then the number of transmission is managed by RIP protocol. Thus, the difference in the path metric value for parallel transmission should be minimal. Figure 4 shows the assembly of three parts of the data transmitted by the same route metric as a case of delay of information transmission, where : T_i – represents that the transmitted time of *i* –th data part, C*i*- denotes that the treatment time (recording) *i* –th data part.

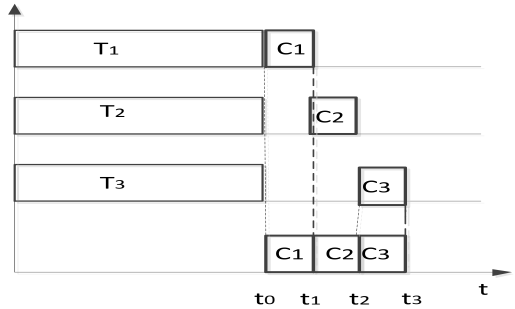

Fig. 4. Assembling parts of data transmitted along routes with the same delay

In this case, the time required for data assembly (t_3- t_1) is minimum and equal to 3(t_1- t_0). At transmission delay of each data part on $_\nabla t = (t_1 - t_0)$ relative to the previous part of the data (shown in Fig. 5), the time of whole data assembly remains minimal.

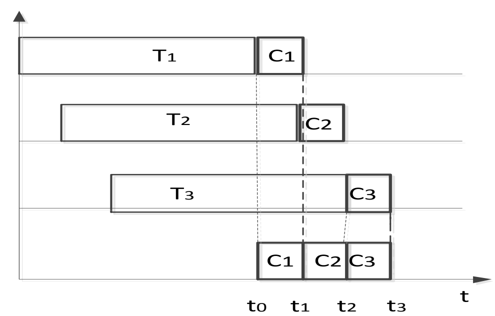

Fig. 5. Assembling parts of data sent along routes with almost identical delay

In case of a delay $\tau_i > (t_1 - t_0)$ in transfer, the *i*-th part of the time required to assemble data parts is increased by $t_z = \tau_i - (t_1 - t_0)$, as shown in Fig. 6.

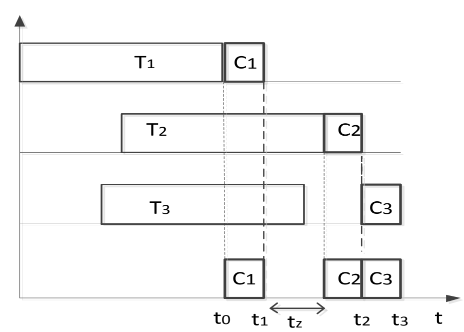

Fig. 6. Assembling the parts of data sent along routes with long delays

The use of partially overlapping paths allows the formation of paths with similar metrics. For example, consider a set of paths P_i with metrics M_i:

$P_{11} = \{v_0, v_2, v_8, v_{11}, v_{18}\}$, $M_{11} = 4$;
$P_{12} = \{v_0, v_3, v_6, v_8, v_{10}, v_{14}, v_{18}\}$, $M_{12} = 6$;
$P_{13} = \{v_0, v_1, v_9, v_7, v_{12}, v_{15}, v_{18}\}$, $M_{13} = 6$.

The difference between the metrics is $M_{13} - M_{12} = 0$, $M_3 - M_1 = 2$, and $M_2 - M_1 = 2$, respectively. In this case, the time required to assemble the entire data (t_3- t_1) (Fig. 7) is maximal, and is equal to $4_\nabla t$.

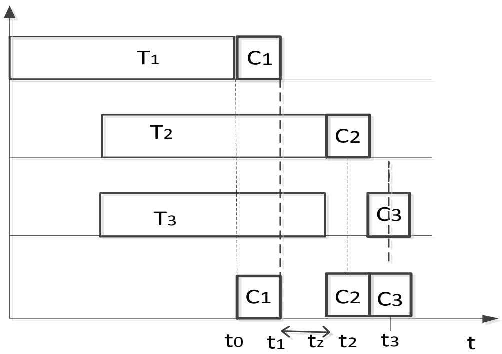

Fig. 7. Assembling parts of data sent along routes with varying metrics

In forming the next set of paths,

$P_{14}= \{v_0, v_2, v_8, v_{10}, v_{12}, v_{16}\}$, $M_4 = 5$,
$P_{15}= \{v_0, v_3, v_6, v_8, v_{11}, v_{16}\}$, $M_5 = 5$, and
$P_{16}= \{v_0, v_1, v_4, v_9, v_7, v_{13}, v_{16}\}$, $M_6 = 6$,

Therefore, the difference between the metrics is less than or equal to one i.e. $M_6 - M_5$=1; $M_6 - M_4$=1; $M_5 - M_4$=0. In this case, data generation time is $3_\nabla t$ (Fig. 8).

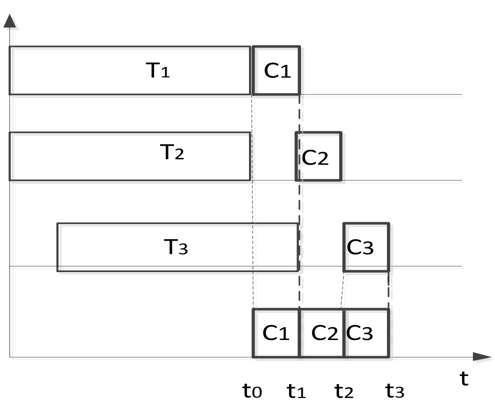

Fig. 8. Assembling parts of data sent along routes with almost identical delays

Thus, the possibility of the formation of various multipath virtual channels allows the optimization of the transfer of information in cloud computing.

IV. Results and Discussion

A. End-to-end delay

The end to end delay is a measure which is used to calculate the average time taken for transmitting the packet in the network .It was calculated using different numbers of nodes, such as 50, 75, 125, 100, and 150. Each node setup incurred different simulation times, such as 100, 150, 200, 250, 300, and 350 (ms). The average end-to-end delay was as shown in Fig. 9. The proposed optimization of TE procedures showed promising results in terms of end-to-end delay due to a minimum delay for different kinds of nodes.

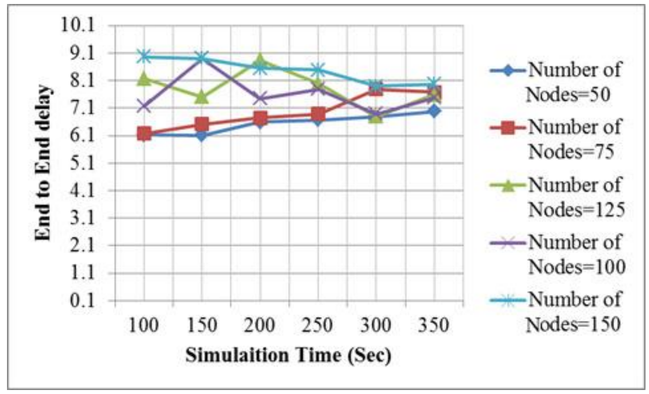

Fig. 9. Average end-to-end delay

B. Packet delivery ratio

Fig. 10 shows the percentage of packet delivery ratio with respect to increasing simulation time. It is clear from the results that packet delivery ratio increases as the time of packets produced by source increases. The average packet delivery ratio was calculated using different numbers of nodes, such as 50, 75, 125, 100, and 150. Each node setup required different simulation times, such as 100, 150, 200, 250, 300, and 350 (ms). The proposed optimization of TE procedures showed promising results in terms of packet delivery ratio.

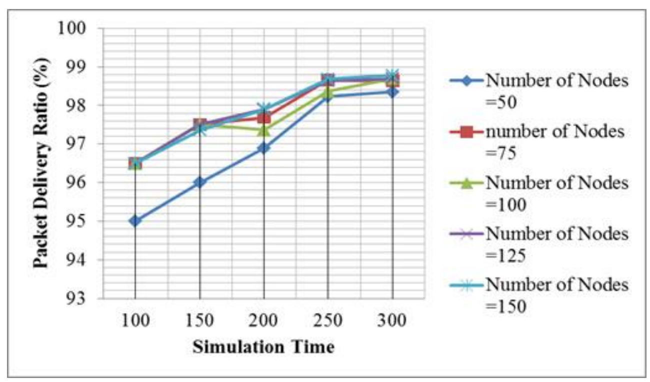

Fig. 10. Packet delivery ratio

Fig. 11 shows that the total packet delivery ratio of the system, here the expected results are more or less same as the proposed system generated results which means that the proposed multipath virtual channels has ensured minimum build time posts for parallel transmission of its individual parts in cloud environment.

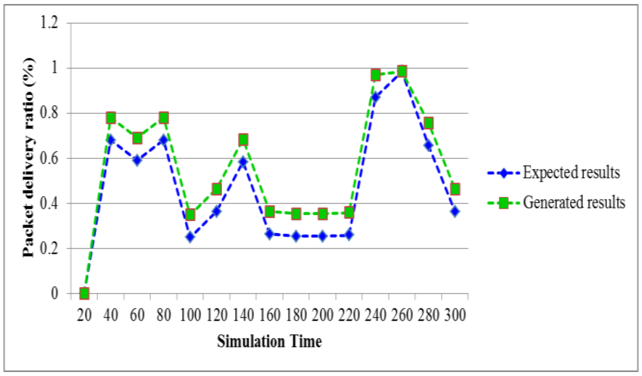

Fig. 11. Total delivery rate

V. Conclusions

This paper proposed an algorithm to calculate the minimum junction area to determine the maximum number of disjoint paths and partially overlapping paths for transforming data via cloud computing. The proposed method of forming partially overlapping paths by creating disjoint paths to the heights of the junction allowed a significant reduction in complexity. The formation of multipath virtual channels based on QoS requirements made it easy to design and improve cloud computing traffic. The possibility of multipath virtual channel formation with the same transmission delay for each path ensures minimal assembly time of data due to the parallel transmission of its parts. A simulation yielded promising results in terms of end-to-end delay and packet delivery ratio.

Competing Interests

The author declares that he has no competing interests.

References

[1] W. Fei-Yue, Parallel system methods for management and control of complex systems, Control and Decision, 19(5) (2004) 485–489.

[2] W. Fei-Yue, Toward a revolution in transportation operations: AI for complex systems, IEEE Intelligent Systems, 23(6) (2008) 8–13.

[3] N. Sadashiv, S.M.D. Kumar, Cluster, grid, and cloud computing: A detailed comparison, 6th International Conference on Computer Science & Education (ICCSE) (2011) 477–482.

[4] Y. Jadeja, K. Modi, Cloud computing: Concepts, architecture and challenges, International Conference on Computing, Electronics and Electrical Technologies (ICCEET), (2012) 877–880.

[5] Z. Xiao, Y. Xiao, Security and privacy in cloud computing, Communications Surveys & Tutorials, 15(2) (2013) 843–859.

[6] S. Pearson, A. Benameur, Privacy, security, and trust issues arising from cloud computing, IEEE Second International Conference on Cloud Computing Technology and Science (CloudCom) (2010) 693–702.

[7] A. Di Costanzo, M.D. De Assuncao, R. Buyya, Harnessing cloud technologies for a virtualized distributed computing infrastructure, IEEE Internet Computing 13(5) (2009) 24–33.

[8] M.A. Vouk, Cloud computing: Issues, research, and implementation, Journal of Computing and Information Technology, CIT, 16(4) (2008) 235–246.

[9] W. Jansen, T. Grance, Guidelines on security and privacy in public cloud computing, NIST special publication 800 (2011) 144.

[10] A. Tolba, A. Ghoneim, A stepwise self-adaptive model for improving cloud efficiency based on multi-agent features, Journal of Software 10(8) (2015) 1037–1044.

[11] Y. Sun, J. Chen, Q. Liu, W. Fang, An improved fat-tree architecture for large-scale data centers, Journal of Communications, 9(1) (2014), 91–98.

[12] H. Takabi, J.B.D Joshi, G.-J. Ahn, Security and privacy challenges in cloud computing environments, IEEE Security & Privacy 8(6) (2010), 24–31.

[13] N.M. Chowdhury, R. Boutaba, Network virtualization: State of the art and research challenges, IEEE Communications Magazine 47(7) (2009) 20–26.

[14] J. He, J. Rexford, M. Chiang, Don't optimize existing protocols, design optimizable protocols, ACM SIGCOMM Computer Communication Review, 37(3) (2007) 53–58

[15] S.T. Maguluri, R. Srikant, L. Ying, Stochastic models of load balancing and scheduling in cloud computing clusters, IEEE INFOCOM (2012) 702–710.

[16] M Kolhar, Saied M. El-atty, M Rahmath "Storage allocation scheme for virtual instances of cloud computing" Neural Comput & Applic (2016). doi:10.1007/s00521-015-2173-8

[17] Saied M. Abd El-atty and Z. M. Gharsseldien, "Performance analysis of an advanced heterogeneous mobile network architecture with multiple small cell layers" Wireless Netw (2016)
doi:10.1007/s11276-016-1218-y.

[18] M. Chiesa, G. Kindler, M. Schapira, Traffic engineering with equal-cost multipath: An algorithmic perspective, IEEE INFOCOM (2014) 1590–1598.

[19] X. Liu, S. Mohanraj, M. Pióro, D. Medhi, Multipath routing from a traffic engineering perspective: How beneficial is it? 22nd International Conference on Network Protocols (ICNP) (2014) 143–154.

[20] N. Wang, K.H. Ho, G. Pavlou, AMPLE: An adaptive traffic engineering system based on virtual routing topologies, IEEE Communications Magazine 50(3) (2012) 185–191.

[21] I. Gojmerac, P. Reichl, L. Jansen, Towards low-complexity Internet traffic engineering: The adaptive multi-path algorithm, Computer Networks, 52(15) (2008) 2894–2907.

[22] I. Gojmerac, T. Ziegler, F. Ricciato, P. Reichl, Adaptive multipath routing for dynamic traffic engineering, Global Telecommunications Conference, IEEE GLOBECOM'03 6 (2003) 3058–3062.