Routing via MST K-Algorithm in Access Mesh Networks for IPTV Reliability

Muhammad Akram[1*], C. A.Rahim[*], Amjad Hussain Zahid[2]**

[*]The Institute of Management Sciences (PAK-AIMS), 54660 Lahore, Pakistan
[**]University of Engineering and Technology, Lahore, Pakistan

Summary
Computing the reliability and routing is very important to identify and quantify the impact of network failure. For reliability of IPTV networks including fixed setup and for mobile setup, different models and algorithms were proposed. These models are probabilistic link model, deterministic models and proactive models with different algorithms for example Dijkstra algorithm, Legacy STP algorithm and Bellman–Ford algorithm. But these models have limitations such as predicted probabilities and computationally expensive matrix inversion and also the run time worst-case. To avoid the multipath routing for initial route, the Kruskal algorithm with filter is suggested in this work. The implementation of suggested algorithm shows the efficient solution which is suitable to topological for multipath routing failures.

Key words:
IPTV, MST, Routing reliability, Access mesh networks

1. Introduction

Reliability is the hottest issues in any network infrastructure design to ensure QoS (quality of services) and QoE (quality of experience). The assessment of end user experience interacting with proposed technology and business entities is known as QoE. A reliable network performs its intended function for a specific period of time .The reliability failures are survivable multicast routing, node to node link blocking and delay due to complicated cyclic routing. To ensure the reliability different algorithms and methods are proposed. This research undertakes an investigation to determine if end-to-end routing algorithms could be suitable for IPTV network and to implement the MST Kruskal’s algorithms for IPTV reliability. A logical extension of this study is to determine if the suggested algorithms can provide benefits and improvements for reliable routing algorithm. Algorithms are put forward that use minimum spanning tree (MST) route estimation. Internet Protocol Television (IPTV) provides digital information and audio video contents, as an operator service over the broadband networks [1]. IPTV transmission setup is basically a content delivery network (CDN) in which the IP broadcasting service stream is transmitted via a server operated by the service provider that provides IP broadcasting service. The CND also offers a two-way communication for viewers, which is called IP multicast network. The service assurance aspects specific to IPTV network ensures reliability of each network element, network segment, and subnetworks with respect to routing across different network domains. It is also essential to evaluate the routing metrics and reliability failures with the path rejection probability and adaptive MST algorithms in case of multicast routing for broadband IPTV services.

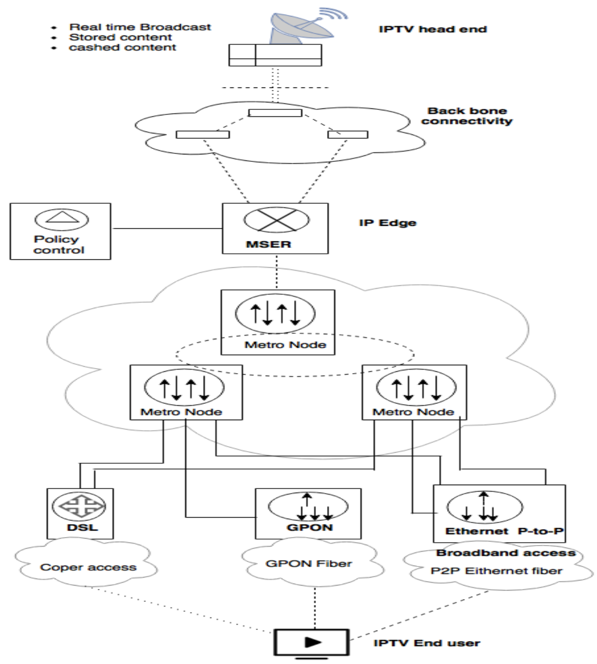


Figure 1-Architecture of broadband access network for IPTV [2]

The service providers for IPTV with different access technologies are Fibre, xDSL, HFC (hybrid fibre-coaxial) and wireless (e.g WiMax, 4G). The broadband access network is based on different elements such as Access nodes, Metro aggregation nodes and IP edge nodes [2]. With respect to broadband portfolios for access nodes, the network must support different types of DSL technologies, fibre based Ethernet and gigabit passive optical network

Manuscript received November 5, 2017
Manuscript revised November 20, 2017

(GPON). In the metro aggregation nodes for transport-based connectivity services the metro networks transport traffic falls between access and IP edge nodes. The edge nodes are the versatile IP edge platforms to enhance the efficiency and reduce the operations in a single multi service edge routers (MSER) platform. To manage the per-user policy profiles and settings a policy-control function is used. Figure 1 explains the architecture of the broadband access network between the IPTV headend and end-users. The GPON and MSER enable to support the requirements of broadband access and IPTV network. In figure 1 three different broadband access are shown such as the coper access DSL, GPON and point to point Ethernet fibre, it is end user choice that which one broadband access network is batter with respect to QoE for IPTV [2]. The major services such as switched digital broadcast channels (SDB), video-on-demand (VOD), digital video recorder, and network based Personal video recorder (nPVR), interactive TV applications (iTV), electronic program guide (EPG), targeted or advanced advertising are offered by IPTV for end users. In order to meet IPTV's services without any delay and higher cost, a network must address the Quality of Service (QoS). The end users of IPTV can also receive the TV Channels over mobile device which is managed by access network of UMTS and LTE.

The figure 2 refers the overall structure of static and mobile IPTV network. The IP multimedia subsystem (IMS), IPTV and web services have same service and control applications. Each application and service is linked with a common Resource Management (RM). RM has a key role for both networks and the heterogeneous wireless networks to ensure service continuity of active sessions among the operator core network in terms of end user perception to maintain the QoS and QoE.

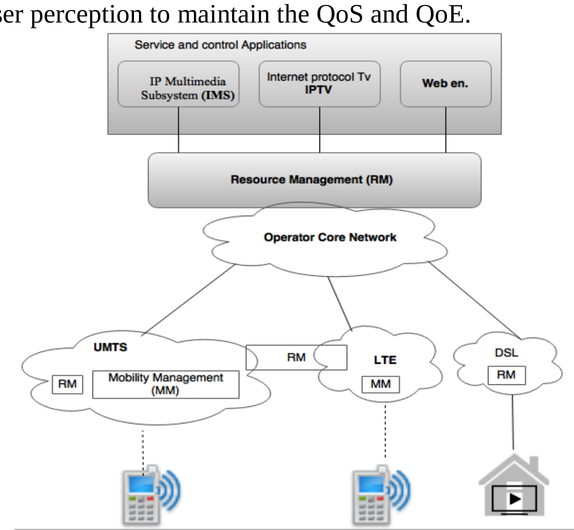


Figure 2.Access network with multiple nodes [3]

In case of multiple communication nodes, the network convergence is used, which is an efficient communication of AV contents and data within a single network. The use of multiple modes in a single network has to convenience and requires the reliable routing infrastructures. Figure 3 shows service convenience and network convergence in which the multiple communication nodes UMTS, LTE and DSL are converged with one access domain (AD) among the different other domains. In this convergence, the access domain and logical variants of nodes have to perform traffic forwarding in access domain. According to figure 4, the forwarding router (FR) is used for traffic forwarding in access domain and the mobile routing is required for mobile IPTV to handle mobility efficiently at the edge of the access network.

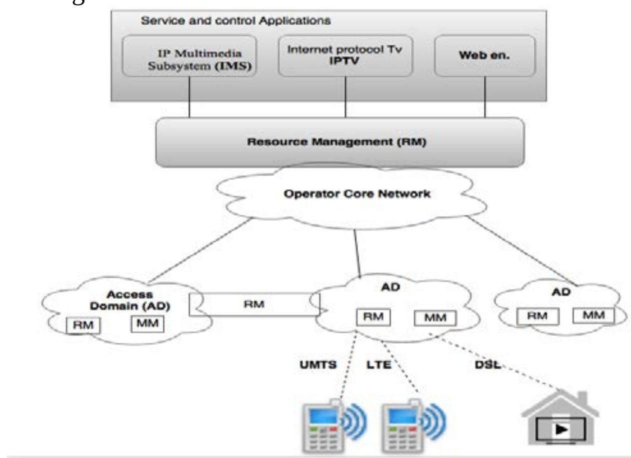


Figure 3.Service convergence and network convergence [3]

The access domain (AD) is a combination of forward and unified routers which is linked with different communication domains like DSL, LTE and UMTS as mentioned in figure 4. The access networks have system specific routes with respect to nodes, protocols and interfaces. Every node has a logical variant that a unified router (UR) connects different access technologies with the unified access domain and the gateway node connects the access domain via aggregation router (AR) [3].

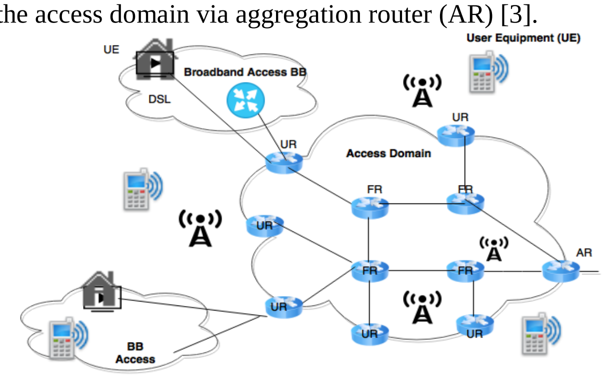


Figure 4. Access domain logical variants of nodes

With reference to figure 4 the access domain has a mesh network which offers a number of access points to link between AR, FR and UR with different users. Each node in the mesh network of access domain is associated to each other in longer zone. The Multi-path routing for mesh network has principally critical issues such as delays due to routing nodes, transmission media contention, hardware failure and underprivileged routing choices. To address these issues different models and algorithms were proposed with different algorithms for example Dijkstra algorithm, Legacy STP algorithm and Bellman–Ford algorithm. But these models have limitations such as predicted probabilities and computationally expensive matrix inversion and also the longest running time. To avoid the multipath routing for first route setup, the minimum spanning tree (MST) Kruskal algorithm with mesh is implemented in this work. This study is based on following; related work as literature survey will be discussed in section II, execution and analysis of Kruskal's algorithm will be described in section III, implementation of Kruskal's Algorithm for IPTV reliable service delivery will be described in section IV and Conclusion will be discussed in section V.

2. Related work as Literature Review

Previously several models and algorithms have been carried out to address the routing problems in access mesh network. To understand this study the following algorithms are discussed for a comparison between suggested implementation of MST Kruskals algorithm in access mesh networks for IPTV Reliability.

2.1 Dijkstra Algorithm

The shortest path calculation of the unvisited vertices in the graph by using a specific method is known as Dijkstra Algorithm [4]. The method of Dijkstra algorithm is based on following such as weighted directed graph, the shortest path node which is the starting point s, and all adjacent nodes [5].

In access network or distribution network scenario, let the vertices of the graph represents nodes and distances between pairs of nodes. These nodes are connected by a direct path, this direct path is presented as edge path costs. In this scenario the shortest route can be find out via the Dijkstra's algorithm. The steps for algorithm representing in graph as weighted, directed graph with source vertex s. S represents as visited vertices. All set of vertices in graph are represented by priority queue Q. The working of algorithm can be explained as that the graph G= (N, L), where G with list of vertices and edges (N and L) are initialized for every node and set 0 value to current node and set ∞ to another unvisited node. The distance from current node to its entire neighbour node also calculated and marked as visited node. The distance is stored which is final and minimal without calculating the visited node again. Using the Big-O notation, the taken time of this algorithm on a graph with edges L and vertices N are actually considered as a function of |L| and |N|. The new closest point belongs to its group S is basically a shortest distance. Dijkstra algorithm has to repeat above mention process until it cannot find a new closest point. So algorithm have to repeat until all points belong to group S. There are some pitfalls of this algorithm such as it can only find the neighbours of the immediate node. While implementing this algorithm by choosing only a short node then the other nodes have to fork. Therefore this algorithm is not backtracking, because it will finally run out of appropriate neighbours to check already visited nodes [6].

2.2 Legacy STP Algorithm

Legacy spanning tree protocol algorithm is a typical spanning tree algorithm (STA) to calculate the loop-free subset of backup network topology. This algorithm is designed to manage direct link and its failure. Figure 5 shows three fully meshed topology based switches labelled as root bridge R, backup root bridge B and a designated root bridge S for link 3 [7]. Switch B notices the failure if link L1 goes down. It claims to be the new root while starting to send BPDUs to S. This new BPDU is received by S from B, it ignores to the one it had stored for port P. After some seconds by default, for port P ages out and the BPDU stored on S. Then S starts to send its BPDU to B and it stops sending its BPDU after acknowledgment. Port P takes twice extra time. The recovery from indirect link failure, the full connectivity is restored with the default parameters. To save 20 seconds it ages out after the port receive lower BPDUs. Every bridge accepts and retains only current information via one root port toward the root bridge. The alternate paths toward the root bridge are blocked by bridges. The previous information is removed and the new information immediately accepted only in case of a superior root bridge. With every port that receives even blocked ports, the switches store the most recent STP BPDUs are stored by switched with such port that collects even congested ports.

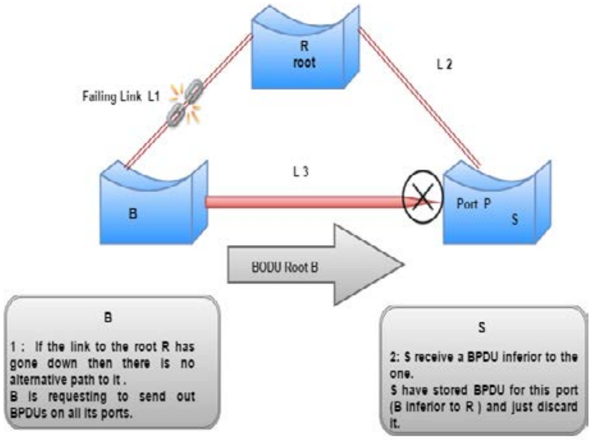


Figure.5 STP recovers an Indirect Link Failure

There is a drawback of Legacy STP Algorithm that the unnecessary unicast flooding is used. Noticeably, legacy feature did not reject the requirement for topology change (TC) event propagation, but Flex Links are allowed for maximum stability if the root port has been changed.

2.3 Bellman–Ford algorithm

In a weighted digraph to calculate the shortest path between a single source vertex and the other vertices, this algorithm is used. On the base of principle of relaxation, this algorithm is a class for weighted directed graphs [8].

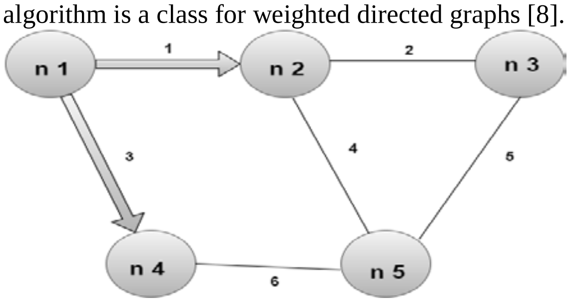


Figure 6.Bellman–Ford algorithm step 1

O (| E | · | V |) is the run time of Bellman-Ford algorithm on graph G= (V, E)

In the above equation vertices are mentioned by V and edges by E. This algorithm takes care of negative weight cycles with its 6 lengthy steps. In step 1, let node n1 conducts its distance vector to node n2 and n3 as shown in figure 6. The link and cost of step 1 from n_2 via link n_1 is 1 and the cost from n_2 without any link is 0.

Step 2:

Node n_2 conducts its distance vectors to its neighbour's n_1, n_3 and n_5 as shown in figure 7.

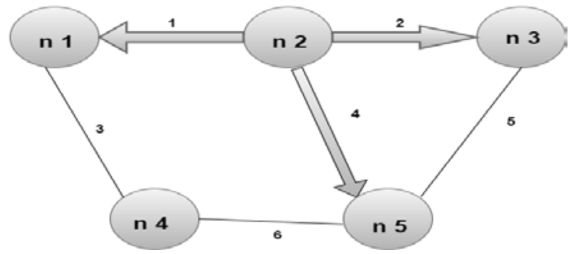


Figure 7.Bellman–Ford algorithm step 2

The link and cost of step 2 can be explained as from source n_1 the cost is 0, and for n_2 the cost is 1 via link n_2. And if the source is n_3 with link n_2 then the cost is 3 for n_1 and for n_2 the cost is 2. If the source is n_5 with link n_2 then the cost for n_1 is 3 and for n_2 is 2.

Step 3:

Node n_4 neighbour's n_1 and n_5 receive information as shown in figure 8.

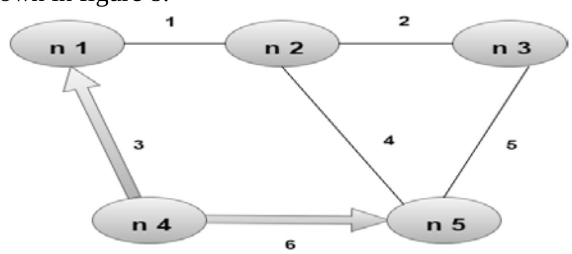


Figure 8.Bellman–Ford algorithm step 3

The link and cost of step 3 according to graph can be explained as, if source node is n_1 with link n_2 then the cost for n_2 is 1 and cost for n_4 is 3 with the link of n_4. If the source node is n_5 with link n_2 then the cost for n_1 is 5 and cost for n_2 is 4. From source node n_5 with the link n_4 then the cost for n_4 is 6.

Step 4:

Node n_3 neighbour's n_2 and n_5 information sharing in the given graph as shown in figure 9.

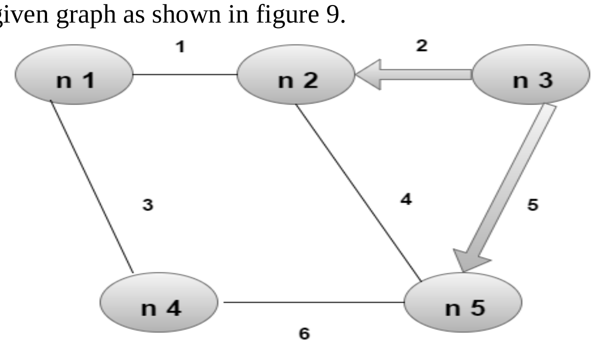


Figure 9.Bellman–Ford algorithm step 4

The link and the cost of step 4 can be explained as, if source node is n_2 with link of n_1 then the cost for n_1 is 1, and for n_2 is 0, for n_3 with link n_3 the cost is 2. With the source of n_2 the cost for n_4 with link n_1 is 4 and cost for n_5 with link n_5 is also 4. With the source node n_5, the cost of n_1 for link n_2 is 5 and cost of n_2 is 4 for link n_2. The maximum coat is 6 for n_4 with link n_4 if the source node is n_5.

Node n_3 has two neighbours' n_2 and n_5 where n2 is associative with n_1 and its cost is 1. Link form n_2 to n_2 has a 0 cost, while n_2 to n_3 has cost 2. Link from n_2 to n_4 is via node n_1 and its cost is 4.

Step 5:

The neighbours of Node n_5 are n_2, n_3 and n_4 which receive the information according to the given graph as shown in figure 10.

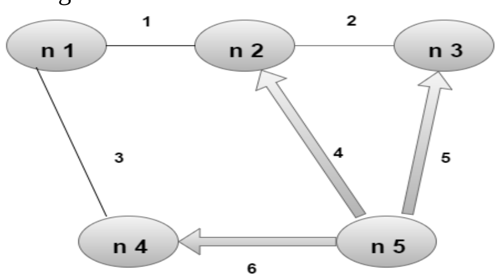


Figure 10.Bellman–Ford algorithm step 5

This step has total five nodes, which are linked from node n_2, n_3 and n_4 and n_5 itself. The link and cost of each node of step 5 is different.

If the source node is n_2 then the cost is given as n_1 link n_1 then cost is 1, n_2 link n_2 then cost is 0, n_3 link n_3 then cost is 2, n_4 link n_1 then cost is 4 and for n_5 link n_5 then cost is4.

If Source node is n_3 then for n_1 link n_2 the cost is 3, n_2 link n_2 then cost is 2, n_3 link n_3 then cost is 0, n_5 link n_4 then the cost is 11, n_5 link n_5 then cost is 5.

If the Source node is n_4 for n_1 link n_1 then cost is 3, n_1 link n_2 then cost is 4, n_5 link n_3 then cost is 11 and for n_5 link n_5 then the cost is 6.

Step 6:

To become a stable graph, the neighbour nodes receive the distance vector from all those nodes whose distance vector has been changed. The final distance vector to each node in the graph can be explained as

If Source is n_1 for n_2 link n_2 then the cost is 1, n_3 link n_2 the cost is 3, n_4 link n_4 then the cost is 3.

If Source is n_2 for n_1 link n_1 then the cost is 1, n_3 link n_3 then the cost is 2, n_1 link n_4 then cost is 4, n_5 link n_5 then the cost is 4 and n_2 link n_5 then the cost is 5.

According to the step 6 of algorithm the final distance vectors to each node and cost there are some drawbacks such as bouncing effect, count to infinity and looping [9]. The bouncing effect remains between the two nodes till the cost of sending the data packet increases. This effect causes a lot of time wastage and the cost. Another factor is the count infinity, in this case the data packets cannot be sent to node n1 but, because the network is not aware of it the data packets sent over to required node which keep bouncing therefore the count would be done till infinity which may cases a lot of delay and cost for transferring of packets.

As this algorithm is called a distributed algorithm and it can be used for AV content delivery but at each node it has limited resources for both bandwidth and power. The AV contents usually have a high data rate therefore media stream is required to be distributed over multiple paths by avoiding the delay over any individual link.

3. Execution and Analysis of Suggested Kruskal's algorithm

3.1 Initial Discussion

The process of selecting paths in a network is called routing metric which is concerned with packet switching and to ensure QoS. IPTV with real time nature of the content delivery is particularly touchy to variations in network conditions and multipath in case of wireless mesh network. Delay is metrics for QoS , affecting any network service [10]. The Multi-path routing is principally critical issue happened throughout data transmission procedure taking into account their link capability for multiple paths. Therefore, the total delay experienced by end-users is caused by multi path routing [11]: These delays are given as:

- Time taken for a signal propagation over physical medium
- Delays due to routing nodes in case of the network's traffic flows
- Transmission media divergence, failure of paraphernalia

Also the poor routing choices can cause large increases in delay.

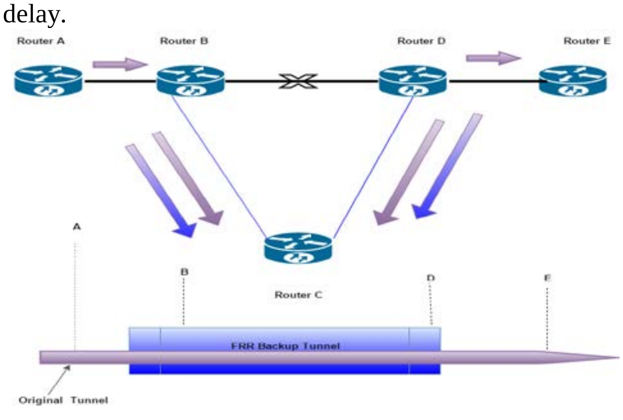


Figure.11 FRR Link protection

To understand the problem statement of this study it is important to explain the reliability failures. In case of failure, over an IP backbone the distribution of real-time multimedia affects uncompromising scatterings for safekeeping and restoration.

The QoE refers limited to less than a few seconds delay experienced by a user viewing the AV content at the receiving end via IPTV set-top box. In case of IP over Dense Wavelength Division Multiplexing (DWDM) network, another failure is link failure. The router failure should be recovered within a fast restoration time, most of the carriers route the multicast traffic by the Interior Gateway Protocol (IGP). Restoration using IGP reconfiguration at the IP layer requires IGP to be aware of the failure. Link-based Fast Re-route (FRR) protects traffic carried in a tunnel between two routers [12]. FRR is a common approach to provide tunnel link availability in case of link failure. There are two types of FRR one is local protection (for link and node) and other one is path protection for individual tunnels. The link protection scenario is shown in figure 11. If the link B to D is failed, then A to E tunnel is converted in B to D tunnel. A have to re-compute the tunnel path as A to B to C to D to E with in the specific time while using the backup tunnel. In case of Node protection operation let suppose the Router D failed then link protection will not help as the backup tunnel terminates on Router D which is also the N-Hop of the protected link. However if the network capacity in not up to the mask, it can result in congestion during network failures. To overcome these limitations the routing algorithms are used.

The routing algorithms in a network consist in finding a path or multiple paths without exceeding the capacity of the links. These algorithms need extra computation time per one node but it can be reduced through the decrease in the number of node [13]. The routing algorithms are basically two types one is centralized and other one is decentralized with static and adaptive characteristics. The centralized algorithm computes and distributes routes while in case of decentralized algorithms, each router sees only local information. The static algorithms are based on the routing table which is maintained and updated manually therefore it is comparatively slow. On the base of topology protocol the dynamic routing table is managed [14]. For a reliable network suppose there is a network with several branches with multi paths, its goal is to connect all branches with the minimum total cost, then the resulting connection should be a spanning tree. The routing algorithms must be aware of the failure and it can compute the new shortest paths with restoration. The path would be a subset of non-cycles but still connects to every node and end user. The objective is to gain minimum total cost by connecting all nodes, for this purpose the minimum spanning tree (MST) algorithms are used and the suggested algorithm among them is Kruskal's algorithm.

3.2 Kruskal's Algorithm

Kruskal's algorithm is a method to compute MST for a connected weighted graph (G) with set of vertices (V) and edges (E) as G = (V, E). Each edge is a pair of Vertex and has weight or a cost. A weighted graph associates with sum of selected edges. Network uses MST of a weighted graph G; the edges are added or sum up to minimize the weight with several nodes and different paths to connect them with the minimum total cost. Kruskal creates a forest (set of trees) in which each vertex is basically a separate tree. The tree is used to sort all the edges in the graph. An edge is added at iteration that reduces the number of components and this algorithm takes ordered list of edges as input and sorts the edges in non-decreasing order of weight [15]. In sorted order, for each edge (u; v) if a condition takes such place that vertices u and v belong to two different trees, then two trees into a single tree by adding (u; v) have to combine. Kruskal's algorithm is basically a greedy algorithm and its description is given as

- Create a set of trees (Ţ) which is not yet spanning
- Create a set of edges (ᴇ) which is nonempty
- A priority queue for the edges until addition of n-1
- For n-1 edges, extract the minimum weight edge from the queue
- If a cycle is formed then it is rejected
- Otherwise it is added to the set of trees to join two trees together.

Each step of Kruskal's algorithm is join two trees in the set ţ together, so that at the end, there will only be one tree in ţ. The Pseudocode of Kruskal's algorithm can be more precisely described as:

Algorithm: KRUSKAL(G)
Input: undirected and weighted graph
G = (V, E)
Output: MST (T).

i. T = ∅
ii. For each vertex , MAKE-SET(v)
iii. v ∈ G.V, for each edge sorted order do
x ← Find (υ)
y ← Find(ν)
iv. weight(u, v), increasing:
v. if (u) ≠ (v): $x \neq y$then
where u and v are FIND-SET
vi. T ← T ∪ {(υ,ν)}
vii. Union(x, y)
viii. return T

3.3 The execution of Kruskal's algorithm

Let suppose a network want to connect its five end-users, to find out the minimum distance the Kruskal's algorithm can be implemented as

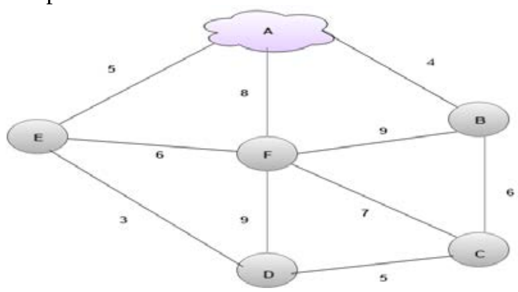


Figure. 12 Network model for Kruskal's algorithm

The edges list of figure.12 in order of size is shown in given table as

Table 1.Order of edges and size of input Network model

Edges	Size
ED	3
AB	4
AE	5
CD	5
BC	6
EF	6
CF	7
AF	8
BF	9
CF	9

According to the given table.1 of edges and their size, the Pseudocode first step of Kruskal's algorithm is to select the shortest edge in the network which is ED with size 3.

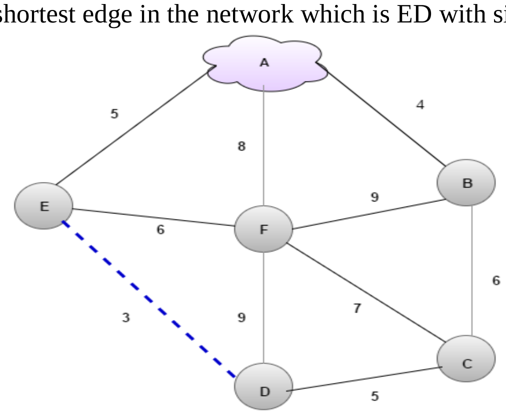


Figure 13.Selection of shortest edge

In figure 13, the shortest edge size 3 is selected for ED edges. The next step is to select another shortest edge AB with the size of 4 which does not create a cycle as shown in figure 14.

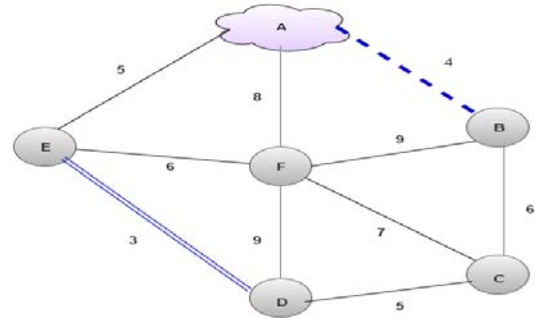


Figure 14.Selection of another shortest edge

According to figure 14 the next shortest edge is AE with the size of 5, every newly selected path is indicated with dashed line and the previous selected path is indicated by double line. The next shortest edge which does not create a cycle is CD with the size of 5 as shown in figure 15.

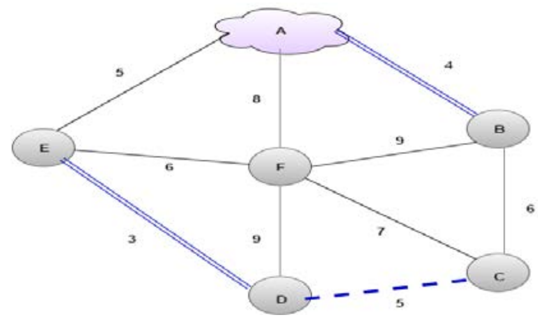


Figure 15.Selection of another shortest edge

As this network has three edges with the same size, therefore these edges can be selected if they do not create cycle. In this scenario, AE also be selected as shown in figure 16.

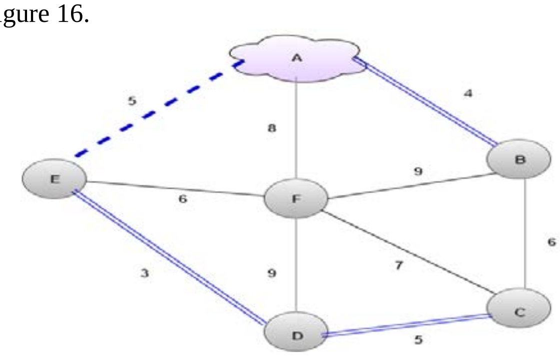


Figure 16.Selection of another shortest edge with same size

The next shortest path is EF than remaining AF, FD, CF and BF, so EF is selected because it does not create the cycle as shown in figure 17.

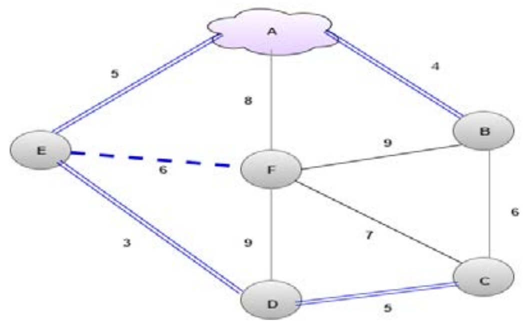


Figure 17.Selection of another shortest edge

According to Kruskal’s algorithm, the paths which are creating the cycle or loop are not selected therefore the connection of all vertices is shown in figure 18.

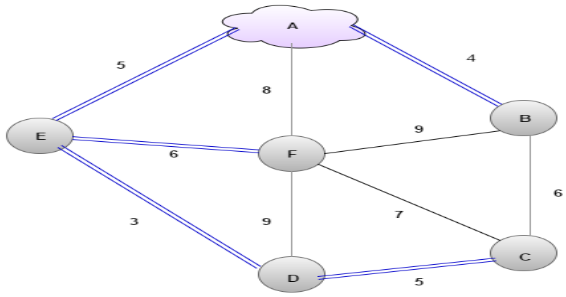


Figure 18.Connection of all vertices

As shown in figure 17 that EF and BC has same size 6, but the only edge EF is selected because if BC is selected then it creates a cycle which is not requirement of this algorithm. In figure 18 all vertices have been connected without any cyclic and the total weight of tree can be calculated as

Table 2.Total weight of tree

ED	3
AB	4
CD	5
AE	5
EF	6
Total	23

The final tree of the graph is shown in figure 19

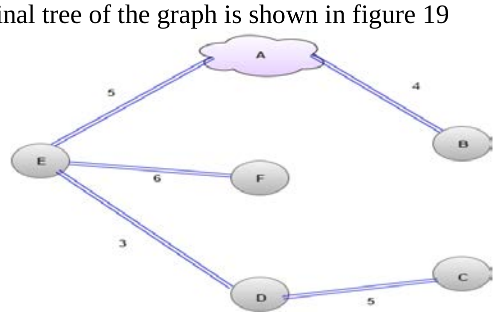


Figure 19.Connection of all vertices makes a tree with length of 23

The Kruskal's algorithm is very easy to implement and to understand and it gives good result for large number of vertices and edges. The resultant tree in figure 19 verifies the step 4 to 8 of Pseudocode of Kruskal’s algorithm as given

4. foreach weighted ordered (u, v), increasing:
5. if (u) $\neq$ (v): $x \neq y$then
6. T ← T ∪ {(υ,ν)}
7. UNION(x, y)
8. return T

3.4 Analysis of Kruskal’s algorithm

According to above mentioned pseudocode line 2 and 3 explains a number of |V| calls of MAKE-SET (v) for each vertex. In line 3 Sorting can be done in |E| * log (|E|). FIND-SET and UNION is called by O(|E|) in line 5. The performance of Kruskal’s algorithm is given by the performance of the Union-Find operations to keep track of vertices with respect to components. For O(E) two 'find' operations and one union for each edge is used. A disjoint-set is connected with union O(E) operations which takes O(E log V) time. This set is a representative with some member of the set. The disjoint set Union-Find has three operations for x (υ) and y (ν), as

i. Create-Set for (x) containing a single item x
ii. Find-Set(x). This set is used to find the set that contains x
iii. Union(x, y) to merge the set containing x
iv. Containing y to a single set.

Union-Find is based on linked lists and weighted union therefore each set is a linked list of attributes head and tail based objects. Head is the first element in the list and tail is the last one while objects have not specific order in the list.
For the set member, the attributes pointer is used to the set object. In Weighted-Union on the list, the single UNION operation takes O (n) time because both sets have same n members. To prove that this algorithm acts as MST, let T be the spanning tree for G generated by algorithm and let Ŧ spanning tree for G with a minimum cost. To prove that both T and Ŧ have same cost or with different cost, let consider the resultant tree T which is shown in figure 20 and other one which creates the cycle is labelled with Ŧ.

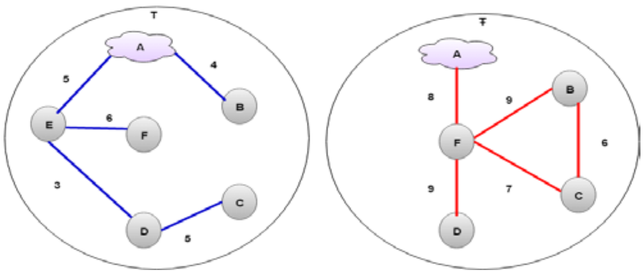


Figure 20.T is resultant tree and Ŧ with cycle

E(T) with x has minimum cost edge while the E(Ŧ) has maximum cost edge with a cycle as shown in the figure 21.

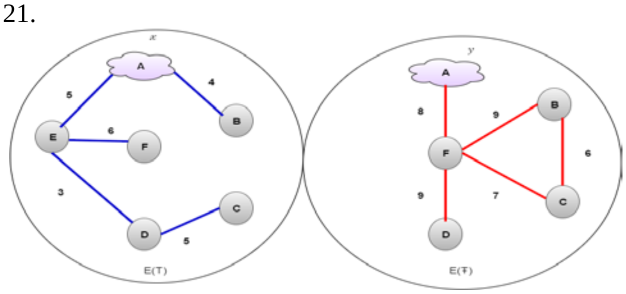


Figure 21.Inclusion of E(T) in to e(Ŧ)=T*

If edges of T and Ŧ are same then let

E (T) ≠ E (Ŧ)
Let x be a minimum cost edge $x \in$ E (T)
where
$x \notin$ E (Ŧ)
i.e x is minimum cost edge in E (T) - E (Ŧ)

Inclusion of x in Ŧ creates a cycle with the edges BCF and by merging all the edges of x and E (Ŧ) creates cycles. Therefore the edges BCF is not in T else x will form cycle in T as well.

Let that edge be $x \notin$ E (Ŧ)

If ω (x) < ω (y) then x would be selected by Kruskal's algorithm first and included into T. Because of choice of x would have been considered for inclusion in a subset T* of T ∩ Ŧ hence, x would not form a cycle in T* because T* ⊆ Ŧ. Here the cycle is created by E (Ŧ) ∪ { x } and it is broken by y. In summary [16], this algorithm requires O (΄E log΄E) this means that

i. O (v) time to initialize
ii. Time O (΄E log΄E) for sort weighted edges
iii. Time O (΄E ∝ (΄E, V)) = O (΄E log΄E) for process edges.

4. Implementation of Kruskal's Algorithm for IPTV reliable service delivery

In case of network topology and wireless mesh network the multi-path routing is one of significant problem and it is based on their ability of the link for multiple paths.

With the example of IPTV access mesh network, it offers more number of access points to each communication between services and end users. The every node is linked to each other in larger area with the set of static mesh routers to extend the reach of defined internet gateway. The multi-hop routes are dynamically formed by using static mesh routers among the end users and the access gateway.

As shown in figure 22 the nodes have different combinations of connections such as DSL, LTE and UMT (wired and wireless connections).

The wired connections directly connect to the core network and provide as gateways to the mesh clients, however due to self-regulation process ability the wireless mesh networks takes minimum time to installation and protection cost.

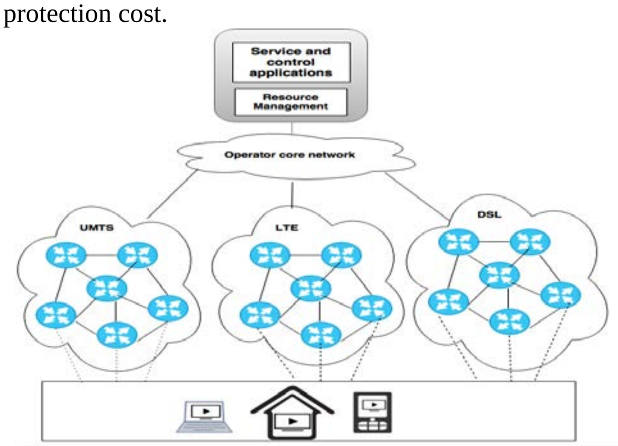


Figure 22.Access mesh network with multiple nodes

However, an efficient algorithm is required for multiple path route selection, end-to-end delay, and content delivery ratio. For this purpose the Kruskal's Algorithm can be implemented to achieve reliable communication without elevated flooding and to decrease the communication delay. Nodes in the given network can be presented as graph G = (V, E). Vertices and edges are considered as the group of vertices and the group of edges which are based on their minimum weight and their distance between two nodes in the network. E=(x,y) denotes the different set of edge values in the graph G that fall within minimum weight value range A, the results of the shortest paths edges are {x,y} . If the range of the weight value is exceeding to minimum weight value range A then the network is not considered for shortest path estimation. Therefore Union-Find operation is added which combines two different edges results for MST Kruskal Algorithm and its procedure is given as

Kruskal

(E,A: MST with minimum weight with sequence of edge, U:Union find)

Sorting of E by edge weight increasing

for each {x,y} ∈ E do

if U have different components like x,y then add edge{x,y} to A

join the partition of x and y in U

Procidure

Mesh

kruskal (E,A: sequence of edge, U: union find)

if ɱ ⩽ (ŋ, | E | , | A |) then Kruskal (E,A,U) \\ kruskal threshold\\

else

Mesh Kruskal (E>, A,U)

Function Mesh (E,U: UnionFind)

Return

[{x,y} ∈ E, x,y are in different components of U]

In mesh kruskal algorithm, once the shortest route path from source to destination node is established then three routing tables such as neighbour, graph and path exploration are created to manage the data distribution. The neighbour table maintains information of all nodes which is created after the initialization of each mesh network. Graph table has the record of route information of earlier and next node to reach destination based on the graph ID for each nodes. The path exploration table stores the sources information from source node i to reach destination node j and the information about the middle nodes in the route selected results from mesh kruskal algorithm. With reference to Hybrid artificial bee colony (HABC) algorithm [17] the cost of the route is based on the fitness value fit_i and its function (bf_ij^d) which is added to graph table for further content distribution as

$$BF_ij^d \leftarrow \phi_{ij}\, fit_{ij}^d + (1-\phi_{ij})\,(c^j_i)^{-1}$$

Where ϕ_{ij} is regularization parameter for route examination and the cost of route value results from source node i to middle node j, which can express by following equation

$$fit_i^d = c^j_i = d_i^j$$

d_i^j is the distance between nodes i and j. With respect to cost function in similar to MST, the closest node approach i selects node j that makes dij minimum, where dij is the cost of edge ij. The cost function based propagation tree can be explained by given figure 23. Let in the example of any network wired or wireless connection, the resultant propagation tree with the five nodes 1 to 5 join in order and attach to the closest node with the video distribution source node s. Let s is the video source and the node i select node j that makes dij + djs minimum.

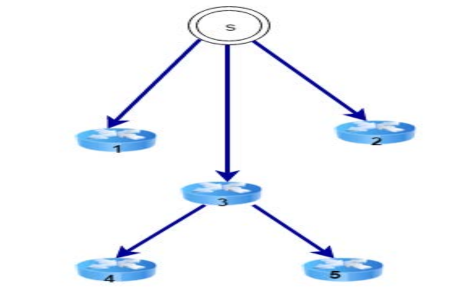


Figure 23.Resultant propagation tree

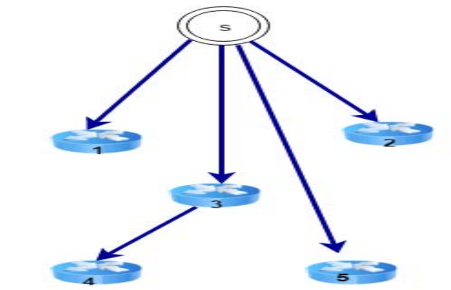


Figure 24.Resultant tree join in order to the closest node

If the propagation tree is within the cost function, then the new node may cause existing nodes to change into new tree optimum as shown in figure 24 in which nodes 1 to 5 join in order and attach to route node. The Kruskal algorithm finishes when either all nodes have been spanned or all edges have been processed.

If the selected edge (E) connects to the same tree vertices (V), then edge is rejected as a cycle would be produced. The resultant is single tree while running this algorithm with time calculation as

$$O(E \log E) \text{ or } O(E \log V)$$

Where E is at most V^2 and $\log V^2 = 2\log V$ If the isolated vertices are ignored then the edges are sorted by weight using a comparison sort in O(E log E) time.

To achieve O(log E) bound, V⩽ E+1, so log V is O(log E).

For this purpose an edge is removed from S in constant time with minimum weight and to keep track of vertices the disjoint-set data structure (Union &Find) is used.

A simple disjoint-set data structure in O(E log V) time can perform O(E) operations , so the resultant time is

$$O(E \log E) = O(E \log V).$$

The Krukal's Algorithm defines to add edges in increasing weight, skipping those whose addition would create a cycle because without cycle the MST become lower cost tree.

By skipping the cycles via Krukal's Algorithm in the IPTV mesh network, the overall cost can be reduced as shown in figure 25.

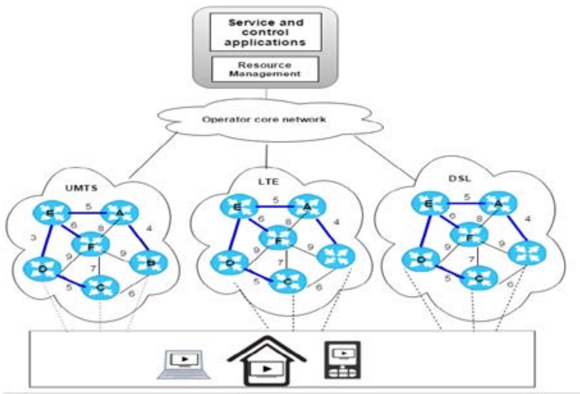


Figure 25.Kruskal's implementation for mesh network in IPTV network

The UMTS has its own assisted mesh network with different nodes such as central control (SC), gate way (GW), mesh node (MN) and mesh client (MC) as IPTV setup. Implementation of Kruskal algorithm also supports the UMTS mesh architecture as shown in figure 26.

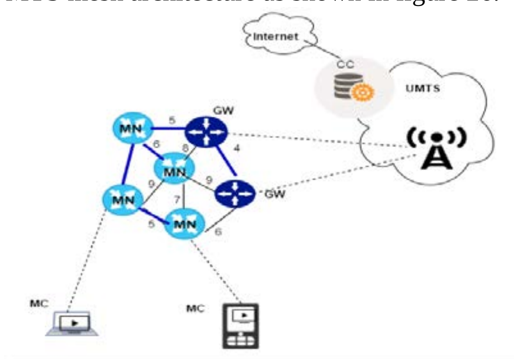


Figure 26.Kruskal algorithm mapping over UMTS supported mesh network

The LTE-Advance with macro cell mesh network mobile users is shown in figure 27.

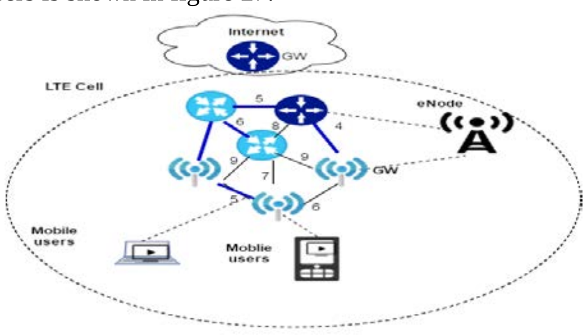


Figure 27.Kruskal algorithm mapping over LTE mesh network

The base station is labeled with eNB, the mobile users MUs with WiFi APs interfaces and internet gateways.

This implementation presents a new dynamic multicast routing for mesh networks associated with access networks that satisfies low cost constraints. The resultant graph shows that Kruskal's Algorithm results the content delivery ratio as higher than others with low end to end delay and minor routing overhead. The comparison between Kruskal's Algorithm and Legacy Algorithm is given in next section for validation.

5. Results and Comparisons

5.1 Comparison between Kruskal's Algorithm and Legacy Algorithm

Legacy STP is also a spanning tree algorithm with convergence properties to calculate cycle free subset of backup network topology. This algorithm is designed to manage direct link and indirect link failure however there is a drawback of Legacy STP Algorithm that the excessive amount of unicast flooding is used. Therefore the comparison between Kruskal's Algorithm and Legacy Algorithm is important for the implementation of reliable network.

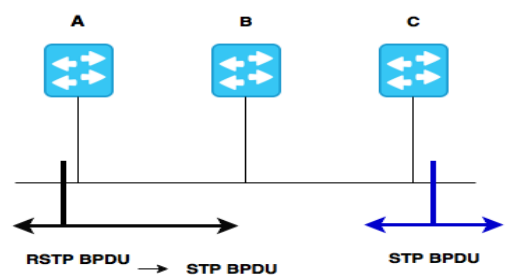


Figure 28.Comparison of RSTP & STP with Legacy STP

The legacy algorithm selects a routing path with a minimum metric of link load where the selected path is not changed. For rapid spanning-tree protocol (RSTP) and multiple spanning-tree protocol (MSTP), Cisco is offering legacy bridges to enable the passage and interoperability as shown in figure 28. To run RSTP two bridges A and B are used, A is acts for the segment and the B acts for legacy STP, bridge C is new on this link [18]. With the addition of bridge C, bridge protocol data units (BPDUs) frames can be run on this link. Such as A accepts BPDUs with migration delay measured in seconds. Bridge A alternates its mode only for this port and this mode has been changed into 802.1D. Therefore Bridge C accepts A as the selected bridge on higher priority. If legacy STP Bridge C is replaced then mode of Bridge A will be STP mode with Bridge B because Bridge A has no information about the removal of Bridge C from the segment. Therefore on Bridge A for this segment the protocol migration has to restart manually [18]. In the case of legacy multicasting, routing tree is based on the network elements, which would be constructed or cut off. To describe the multicast routing algorithm, for mathematical formulations let consider the given network as T (N, L, B). N is set of nodes, L is set of links and B is bandwidth of the links. Let the number of nodes is labbled with "n" and number of links with "m" demands for bandwidths are integral. The nodes are considered as the set by (a, T, b) for multicast connection. The node b is integral units for the bandwidth. Let node-link matrix (M) is n × m and each row corresponds to a node in this matrix and in the graph. The links are labelled in any arbitrary order in a sequence. Let the bandwidth of each link which is required for multicasting is given as

$$x\ aT = (x1, \cdots, x\ m\ )\ t$$

In above equation t represents transfer vector, m × 1 corresponding to (a, T). The cross ponding vector m × 1 for link (u,v) can be as

$$Y = (Y1, \ldots.\ Y\ m\ )\ t$$

The capacity of the link is mentioned by ɉ, so each component capacity Y j presents a residual capacity. For initial stage, the rate of Y can be set as

$$Z = (z, \ldots., z)t$$

Let Ꝋ aT is a vector n × 1 as branch node with an integer ƙ to distribute ƙ information for outgoing links. If the number of available paths from the previous setup is established, and if at the incoming interface for multicast stream is transferred at outgoing interface in to three streams, then integer ƙ is established as 2+1 for node a position. Another condition can be observed that if node position (a) could not mentioned as a branch node then −1 is positioned for destination nodes in group T. Therefore the value of all components summation is

$$Ꝋ\ aT = 0$$

For the maximizing available paths Max ꬵ1 K(xaT) with the feasible set ꬵ1 and K(xaT) presents the number of xaT in ꬵ1 which can be explained as given equation.

ꬵ1 = { xaT : M xaT = z Ꝋ aT , xaT ⩽ Y, 0 ⩽ $\varkappa_i$ϵ{ 0, z}}

To increase the link capacity summation with another feasible set ꬵ 2 can be explained as

$$\max_{ꬵ\,2} \sum_{j=1}^{m} Y\,j$$

To prove the given formulation, let the element of T (N, L, B) is specified by

(a, T, b) = (e1, {e3, e4}, 2).

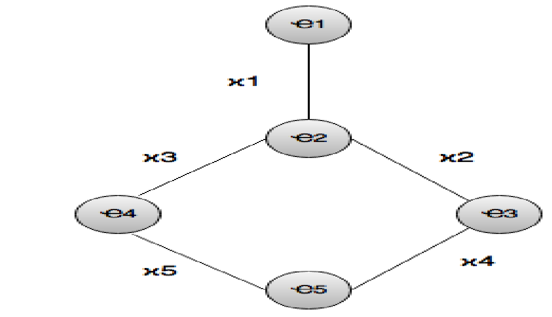


Figure 29.Illustrative the formulation for Node-link matrix graph

The Ꝋ aT can be presented as (1 1–1 -1 0) and xaT can be presented as (2 2 2 0 0) or (2 0 2 2 2). In a result, the initial path is recovered with accurate calculations than the 2nd one path, because the first path has 3 by calculating the hop count while the hop count for the second path is 4 as shown in figure 29.

The major type of legacy algorithm is Minimal Spanning Tree-Steiner Algorithm Residual (MSTSAR) which is used for residual capacities on all the links. The input for this algorithm is a graph T (N,L,R) where R presents the set of residual capacities on all the links. The output of this algorithm is a tree between a and T giving the capacity of b units. The steps of MSTSAR algorithm are given as [19]

1: Convert the graph T(N,L,R) to a new graph T′ (N′,L′,R′)

It shows that the links are eliminated less than b units.

2: T'_L(N'_G, L'_G,C) C is loop hop count of L'_G .

3: Find a MST Ŧ$_L$′$_G$ on T'_L

4: The multicast tree starts with empty

5: for links l ϵ L'_G do

6: find a shortest path P on T_L

7: add a path on the tree

This MST is made for the minimum hop count which is called Steiner tree. In this tree each path has the minimum hop and this minimum hop count is added in to the resultant final tree. MSTSAR algorithm has O (n × n^2_T) time complexity. However the fast channel switching it is difficult to manage with the legacy algorithm particularly for IPTV network. It is also observed that the legacy algorithm is not designed to support for the effective deployment by using the core network resource.

5.2 Comparison results of Kruskal's Algorithm with Dijkstra Algorithm and Bellma-ford Algorithm

Randomly 9 nodes are created as an input for the comparison between Dijkstra, Bellman-ford and Kruskal's algorithm to calculate shortest route without cyclic of loop property while sending data from source node 1 to receiving node 9. The individual graphs for the Dijkstra shortest distance route, Bellman shortest distance route and Kruskal's time per edge for random graph with random edge weigh are shown.

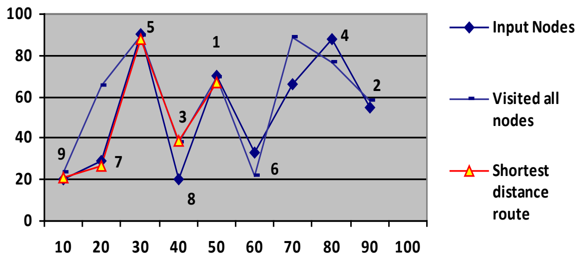


Figure 30.Dijkstra shortest route

Referring to figure 30, the randomly created nodes as input are 9 and which are assume to send data from source node 1 to receiving node 9 and all nodes have to visit for newly inserted node to pick a vertex not in source node with the lowest label and add it to source again, so this practice have to repeat until the destination or receiving node is in source. So the resultant shortest path has just 5 nodes out of 9 which are node 1 to 3, node 3 to 5, node5 to 7 and node 7 to node 9.

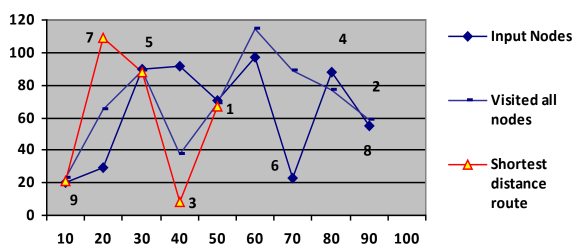


Figure 31.Bellman shortest route

The calculated shortest path in figure 31 for the Bellman routing algorithm gives a shortest distance from node 1 to node 3 and node 3 to node 5 and node 5 to 7 and then finally node 7 to node 9.

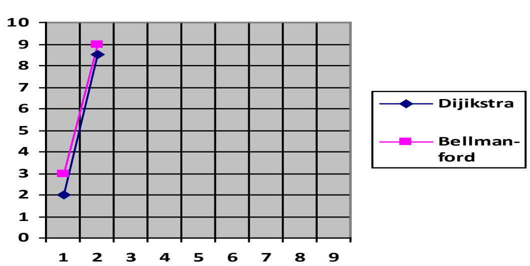


Figure 32. Reduction between Dijkstra and bellman

The number of shortest path nodes is same as Dijkstra but the node to node distance is different. The reduction between Dijkstra and bellman ford algorithm is shown in figure 32.

In the given figure 32 the Dijkstra preforms better due to distance increments than Bellman-ford because Dijkstra has faster response takes less convergence time in term of throughput and traffic loads. The measurement of total time spent for data received in Kruskal's Algorithm the graph between total time and communication is shown in figure 33. Kruskal's Algorithm has good performance as its cost of communication is much less than other algorithms, however the computation is observed slow which is improved by the Union Finds functions. The routing cost for average degree of Kruskal's and Dijkstra Algorithm is shown in figure 34, which is a graph between routing cost and average degree per vertex by varying the number of nodes form 10to 50.

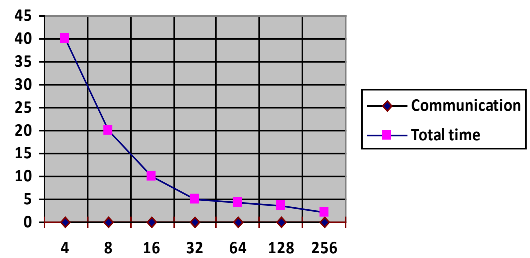


Figure 33.Ratio between process and total time

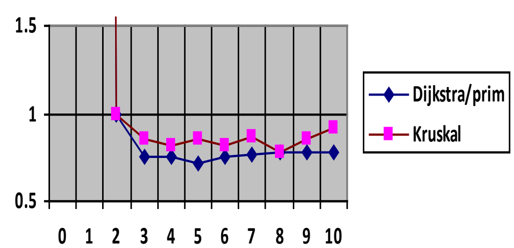


Figure 34.The routing cost for Kruskal and Dijkstra Algorithm

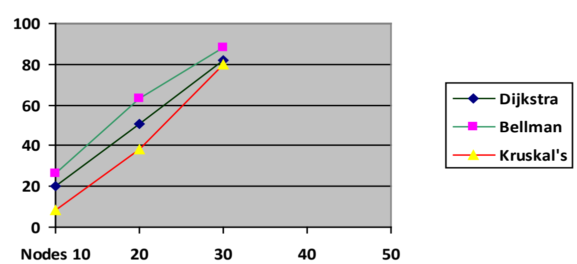


Figure 35.Resultant graph between number of nodes and time

In figure 35, for Kruskal's Algorithm the time per random edge weight, the graph between number of nodes and overhead in number of data shows the comparison between Dijkstra, Bellman and Kruskal's routing algorithm for reliability. The number of nodes and the time taken in milliseconds graph shows that for mesh network, Kruskal's algorithm takes less time to cover the required nodes than both Dijkstra and Belman-ford algorithm. The simulation results show that Dijkstra algorithm is best solution for multicast connections however it does a blind search therefore a lot of time waste and also it cannot handle negative edges. Therefore it leads to acyclic graph, which cannot achieve the right shortest path.

6. Conclusion

The aim of this study is to evaluate and implement Algorithms for reliable content transmission over IPTV via access mesh networks routing. The ability of a network to carry out a robust to failures communication between heterogeneous source node and terminal node is based on routing algorithms. In case of reliable routing this study compares Dijkstra, Legacy STA algorithm and Bellma-ford algorithm with the Kruskal's Algorithm. The implementation of Kruskal's Algorithm shows a higher packet delivery ratio, lesser end to end delay and lesser routing overhead than the existing routing algorithms particularly for IPTV multicasting and mesh networks. So the suggested Kruskal's Algorithm assures the load balancing with its cycle and also promises to achieve highly reliable communication. As future works, there is potential to analyse the features of network routing via implementing the cost effective algorithms to survive the efficient routing for both stable and mobile networks, as the implementation of MST K algorithm which easily applicable to topological changes avoiding any failure in network.

References

[1] M.Akram , C. A. Rahim , A.H. Zahid, "AV Encryption Algorithm to Protect Audio visual Content for IPTV", International Journal of Computer Science and Information Security (IJCSIS),Vol. 14, No. 6, June 2016

[2] P.Arberg, T. Cagenius, Olle V. Tidblad, M. Ullerstig ,P.Winterbottom, "Network infrastructure for IPTV", Ericsson Review No. 3, 2007

[3] M.Bauer, A.Dekorsy, P. Schefczik,M.Soellner, "Converged Access Networks Unified Resource Management and Network Architecture", Alcatel-Lucent, Bell Labs Europe – Nuremberg, Mai, 2007

[4] DexiangXie, Haibo Zhu, Lin Yan, Si Yuan and JunqiaoZhang " An improved Dijkstra algorithm in GIS application" Proceedings of 2010 Conference on Dependable Computing, 2010

[5] DechuanKong ,Yunjuan Liang, Xiaoqin Ma, Lijun Zhang "Improvement and Realization of Dijkstra Algorithm in GIS of Depot", IEEE, Institute of Electrical and Electronics Engineers, Inc, 2011

[6] Harika Reddy, "PATH FINDING - Dijkstra's and A* Algorithm's", December 13, 2013

[7] http://www.cisco.com/c/en/us/support/docs/lan-switching/spanning-tree-protocol/12014-18.html

[8] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, Introduction to Algorithms, Second Edition. The MIT Press and McGraw-Hill Book Company, 2001.

[9] S.Vutukury, j.j.Garcia Luna Aceves, "An Algorithm for Multipath Computation using Distance-Vectors with Predecessor Information" University of California at Santa Cruz,Department of Computer Engineering,Santa Cruz,CA,95064,1999

[10] Z. Jelacic, H. Balasko, and M. Grgic (2008) End-to-End Real-Time IPTV Quality Monitoring, ELMAR 2008. 50th International Symposium, pp. 73-77.

[11] H. Joo, H. Song, D. Lee, and I. Lee. (2008) An Effective IPTV Channel Control Algorithm Considering

[12] T.Elhourani, A.Gopalan, S.Ramasubramanian, "IP Fast Rerouting for Multi-Link Failures" , IEEE Conference on Computer Communications, IEEE INFOCOM 2014

[13] F. Poppe, S. van den Bosch, P. de la Vall´ee-Poussin, H. van Hove, H. de Neve, and G. Petit, "Choosing the objectives for traffic engineering in IP backbone networks based on Quality-of-Service requirements" In Proceedings of First COST 263 International Workshop, QofIS, pages 129–140, Berlin, Germany, Sept. 2000.

[14] Amos Korman, David Peleg, "Dynamic Routing Schemes for Graphs with Low Local Density", ACM Transactions on Algorithms, Vol. 4, No. 4, Article 41, Publication date: August 2008.

[15] Dorit , S. Hochbaum, "The Pseudoflow Algorithm Graph Algorithms and Network Flows", Lecture Notes for IEOR 266, 2008

[16] BOOK Design Analysis and Algorithm By Hari Mohan Pandey , university science press ,1st edition 2008

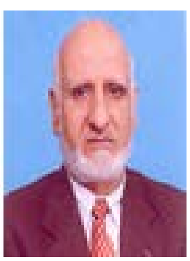
[17] Xiangyu Kong, Sanyang Liu, Zhen Wang, "A new hybrid artificial bee colony algorithm for global optimization", International Journal of Computer Science Issues, Vol. 10, Issue 1, No 1, January 2013

[18] Cisco AVVID Network Infrastructure: Implementing 802.1w and 802.1s in Campus Networks, Implementation Guide April, 2003

[19] Sungkwan Youm, Taeshik Shon ,Eui-Jik Kim, "Integrated multicast routing algorithms considering traffic engineering for broadband IPTV services", EURASIP Journal on Wireless Communications and Networking,Vol.1,Issue 127, 2013

Muhammad Akram is a TV Broadcast Engineer by profession and currently working in Virtual University of Pakistan's TV channels as Engineering Incharge. He received B.E degree in Telecommunication Engineering accredited by Pakistan Engineering Council (PEC) from Institute of engineering and Technology, Allama Iqbal Open University Islamabad Pakistan. He is enrolled for degree of M.Phil Computer Science from The Institute of Management Sciences, Lahore Pakistan. His area of interest is research in field of advance broadcast engineering, Audio Video processing over IP with security and implementation of algorithms for different networks and interfaces.

Amjad Hussain Zahid is M.Phil Computer Science scholar with renewed experience in academia as faculty member. He is currently working as Assistant Professor in Department of Computer Science of FC College (A Chartered University), Pakistan. He is registered student for PhD in Computer Science from University of Engineering and Technology, Lahore, Pakistan. Being a faculty member his favorite subjects for teaching are computer Networks, Compiler Construction, Object Oriented Programming, Computer Graphics, Design and Analysis Algorithms. He is also member of National Curriculum Revision Committee (NCRC), Higher Education Commission Pakistan.



Prof. Dr. Abdul Rahim Chaudhry has recently joined the Institute of Management Sciences (IMS), Lahore Pakistan as Dean of Academies. He has earned a PhD in Physics from University of Salford, UK. His area of research is Computer architecture and assembly language. His work experience spans over four decades. He has worked as Professor for over twenty five years at Punjab University, Lahore.