Declaration of Authorship

We, Shifatur Rahman (134402) and Sakib Hasan(134406), declare that this thesis titled, **'Constraint-based Multicast Routing in Internet of Things'** and the works presented in it are our own. We confirm that:

- This work has been done for the partial fulfillment of the Bachelor of Science in Computer Science and Engineering degree at this university.
- Any part of this thesis has not been submitted anywhere else for obtaining any degree.
- Where we have consulted the published work of others, we have always clearly attributed the sources.

Submitted By:

Shifatur Rahman (134402)

Sakib Hasan (134406)

Constraint Based Multicast Routing In Internet of Things

Approved By:

Dr. Muhammad Mahbub Alam
Thesis Supervisor,
Head of the Department,
Department of Computer Science and Engineering,
Islamic University of Technology.

Abstract

Multicast routing that meets multiple quality of service constraints is important for supporting multimedia communications in the Internet of Things (IoT). Existing multicast routing technologies for IoT mainly focus on ad hoc sensor networking scenarios; thus, are not responsive and robust enough for supporting multimedia applications in an IoT environment. In order to tackle the challenging problem of multicast routing for multimedia communications in IoT, in this book, we analysed two algorithms for the establishing multicast routing tree for multimedia data transmissions. The proposed algorithms leverage an entropy-based process to aggregate all weights into a comprehensive metric, and then uses it to search a multicast tree on the basis of the spanning tree and shortest path tree algorithms. We went through the evolution of the problem from wired networks to wireless netoworks.The book shows the theoretical analysis and extensive simulations for evaluating the proposed algorithms. Both analytical and experimental results demonstrate that one of the proposed algorithms is more efficient than a representative multiconstrained multicast routing algorithm in terms of both speed and accuracy; thus, is able to support multimedia communications in an IoT environment. We believe that the results shown at the end of the algorithm description are able to provide in-depth insight into the multicast routing algorithm design for multimedia communications in IoT and also will speak for themselves of ensuring a better Multicast Routing for Multimedia networks in Internet Of Things based on multiple constraints.

Acknowledgements

First and foremost, we offer gratitude to the Almighty Allah (SWT) for giving us the capability to do this work with good health.

We are grateful to our research supervisor, Dr. Muhammad Mahbub Alam, for the support and guidance throughout our research at Islamic University of Technology (IUT). He created a nice research environment for which we were able to explore many ideas without constraint. We have gained a wealth of knowledge and experience in science and engineering through his direction that is beyond value to our future endeavor. For all of his efforts as our true mentor, we express our heartfelt gratitude to him.

We would like to thank all the faculty members of the department of CSE, IUT for their inspiration and help.

And last but not the least we are thankful to our family, friends and well-wishers for their support and inspiration. Without them it would never have been possible for us to make it this far.

Contents

1 Introduction and Background

2 Background Study

3 Entropy Weight Aggregation

4 Multiconstraint Multicast Routing In Wired Networks

5 Multicast Routing In Wireless Networks in IOT Environment

6 Experiment

1 Introduction and Background

1.1 Multicasting and IOT

The Internet of Things (IoT) has been envisioned as a key networking paradigm to bridge the gap between the cyber and physical world [1]. IoT, in general sense, is a network that consists of a wide variety of things or objects, such as RFID tags, sensors, actuators, and mobile devices, which are connected through wired and/or wireless networks to the Internet. Recent progress in the IoT has shown that it is leading toward a new digital context, which promotes various novel applications and services [2]. To successfully implement the IoT, network infrastructure, and protocols play crucial roles in providing effective and efficient communications among IoT objectives. Therefore, networking performance has a significant impact on IoT service performance [3]. Data communications in IoT has been an active research area. Earlier applications of IoT were mainly in data sensing and object actuating environments, where IoT devices with limited computing and battery capacity communicate with each other through wireless channels [4]–[6]. The main objectives of previous work for IoT data communications are to minimize network resource and power consumption while maintaining a (relatively low) level of quality of service (QoS). With the proliferation of IoT, it is being adopted into much more diverse application scenarios, including the recent trend of combining IoT with cloud computing and big data analytic services [7]–[10]. All these bring in new challenges to high performance networking for supporting various IoT services including multimedia applications.

Multimedia communications typically require networks to provide guarantees of multiple QoS metrics, for example, the minimum throughput, maximum delay, and maximum packet loss rate for data transmissions. The dynamic nature of networking in IoT makes such requirements more challenging. Again, in the vehicle network of an emergency response system, various network elements including routers, switches, base stations, even gateways to the Internet, may be hosted on moving vehicles; therefore, the network topology may change frequently. In addition, the wireless channel capacities between network nodes also vary due to vehicle mobility; thus causing variable link status in network topology. Supporting real-time multimedia communications in those dynamic IoT networks becomes a challenging problem that has to be addressed. Also, multimedia communications often require multicast data transmissions. Multimedia content delivery services are based on high performance data transmissions from a single data source to multiple data sinks [11], [12]. In the above emergency response system, multimedia incident information may be distributed from a cloud data center to the responders on multiple vehicles through multicast in the vehicle network. Therefore, multicast routing mechanisms that are able to guarantee multiple QoS constraints as well as adaptive to dynamic network topology play a crucial role in supporting high-performance multimedia communications in an IoT environment.

However, traditional QoS multicast routing technologies developed for wireline networks often assume relatively static network topology and unlimited amount of resources such as node capacity, which do not fully reflect the features in IoT. More efficient multicast routing algorithms are required in order to meet the requirements of multimedia communications in more dynamic IoT networking environments. Beside, previous work on multicast routing in IoT often assumed ad hoc sensor networks as the context. In such networking scenarios, communications are mainly for sensing data collection and actuation data transmission, which do not have strict QoS requirement such as the minimum throughput and maximum latency. The main objectives of the multicast routing algorithms developed in this area are to maximize network resource utilization and/or minimize network energy consumption, rather than meeting the multiple QoS constraints required by multimedia communications. Therefore, multicast routing for supporting multimedia communications in an IoT environment needs to be redesigned.

1.2 Real World Applications

As a real-world application case of multimedia communications in IoT, for example, an emergency response system comprises a group of rescue vehicles that communicate with each other to form an ad hoc vehicle network, which is connected to the Internet infrastructure through some gateway devices. Multimedia information about the task field, such as videos and images showing the nature and severity of the incident, may be collected via some sensing devices in the field and transmitted to a vehicle network gateway through the Internet infrastructure. Then the gateway may forward such multimedia data to a group of vehicles assigned to a certain task via multicast communications in the vehicle network. In this application case, multimedia communications with multicast routing capability are required in the IoT environment. Also some real time audio-video streaming in multiple devices in an IoT based environment needs multicast routing.Interactive remote recording and playback of multicast video-conferences in multimedia communication is also a sector of this implementation.

1.3 Thesis Objective

The sole objective of our thesis research is to evaluate algorithms proposed for the multicast routing in Internet of things for multimedia data communication. Of all the approaches we know of fall under the victim of packet loss, jitters, time delay and a whole lot of latency. Our aim is to ensure approaches so that these constraints gets reduced to the optimal level of the network condition on-demand. The steps involved in our research are as follows-

- Background study of algorithms already existing for ad-hoc and wired networks.This considers the theoretical analysis of [1][2][3] which considers a fixed number of constraints balancing.
- The next study was to analyse the approach to solve multicasting issues in wired networks considering more number of constraints to be exact more than or equal to 2.
- The following step involves our main interest which describes and therotically analyse two algorithms called FAST and FAMOUS for multicast routing in IOT environment taking more than 2 constraints in consideration.
- In the next step we formulated one of the algorithms in a sample environment and tried to simulate and find out what problems still remain in the solution approach.
- As our future work of interest we list those problems and conclude our work in this way.

Our whole research was to theoretically discuss and analyse the solutipn approaches to this multicasting problem in different networks and try to find the problems of that approach. After that we went on to our next study to find how the previous problems were approached to solve or reduce. In this way we advanced and conducted our research. At the last we found out some problems of the latest approach and intend to work on that in the future.

2 Background Study

2.1 Khuller's Algorithm

Khuller et. al. [1] proposed multicasting in wired networks(known topology). We give a simple algorithm to find a spanning tree that simultaneously approximates a shortest-path tree and a minimum spanning tree. The algorithm provides a continuous tradeoff: given two trees and a 7 > 0, the algorithm returns a spanning tree in which the distance between any vertex and the root of the shortest-path tree is at most 1 + x/27 times the shortest-path distance, and yet the total weight of the tree is at most 1 + ,~/2/~/ times the weight of a minimum spanning tree. Our algorithm runs in linear time and obtains the best-possible tradeoff. It can be implemented on a CREW PRAM to run a logarithmic time using one processor per vertex.

Solution :

The algorithm is given an e > 1, a minimum spanning tree, and a shortest-path tree rooted at a vertex r. It returns an (e, 1 + 2/(e - 1))-LAST rooted at r. The basic idea of the algorithm is to traverse the minimum spanning tree, maintaining a *current tree,* and checking each vertex when it is encountered to ensure that the distance requirement for that vertex is met in the current tree. If it is not met, the edges of the shortest path between the vertex and the root are added into the current tree. Other edges are discarded so that a tree structure is maintained. After all vertices have been checked and paths added as necessary, the remaining tree is the desired LAST. The final tree is not too heavy because a shortest path is only added if the path that it replaces is heavier by a factor of e > 1. This allows a charging argument bounding the net weight of the added paths.

Lackings :

The problems that were still prevailing in this solution approach are -

- Only 2 constraints(path cost & delay).
- Time delay is not fully optimal for various cases.
- Works for known topology(wired networks) only.

2.2 Parsa's Algorithm

Parsa et. al. [2] tried to improve the multicasting in wired networks proposing a new algorithm. The bounded shortest multicast algorithm (BSMA) is presented for constructing minimum-cost multicast trees with delay constraints. BSMA can handle asymmetric link characteristics and variable delay bounds on destinations, specified as real values, and minimizes the total cost of a multicast routing tree. Instead of the single-pass tree construction approach used in most previous heuristics, the new algorithm is based on a feasible search optimization strategy that starts with the minimum-delay multicast tree and monotonically decreases the cost by iterative improvement of the delay-bounded multicast tree. BSMA's expected time complexity is analyzed, and simulation results are provided showing that BSMA can achieve near-optimal cost reduction with fast execution.

Solution :

The DBMST problem can be approached as a feasible search optimization problem in which the feasible region consists of all trees that satisfy the delay-bound requirement. BSMA constructs a DBMST in two steps described as follows,

Initial step: Construct an initial tree with the minimum delays from the source to all destinations.

Improvement step: Iteratively minimize the cost of the tree while always satisfying the delay bounds.

To guarantee that a feasible solution is found that satisfies the given delay bound, the initial tree is the minimum delay tree, which is constructed using Dijkstra's shortest-path algorithm. In some cases the delay bounds given by DDF may be too tight, i.e., they cannot be met even in the minimum delay tree. In such cases some negotiation is required to relax the delay bounds of DDF before any feasible tree can be constructed, as shown in the BSMA flowchart. The bounds given by DDF must be relaxed until they can be met by the minimum-delay tree. The rest of this paper assumes that DDF assigns the delay bounds that can be met by the minimum-delay tree.

Lackings :

The problems that were still prevailing in this approach are-

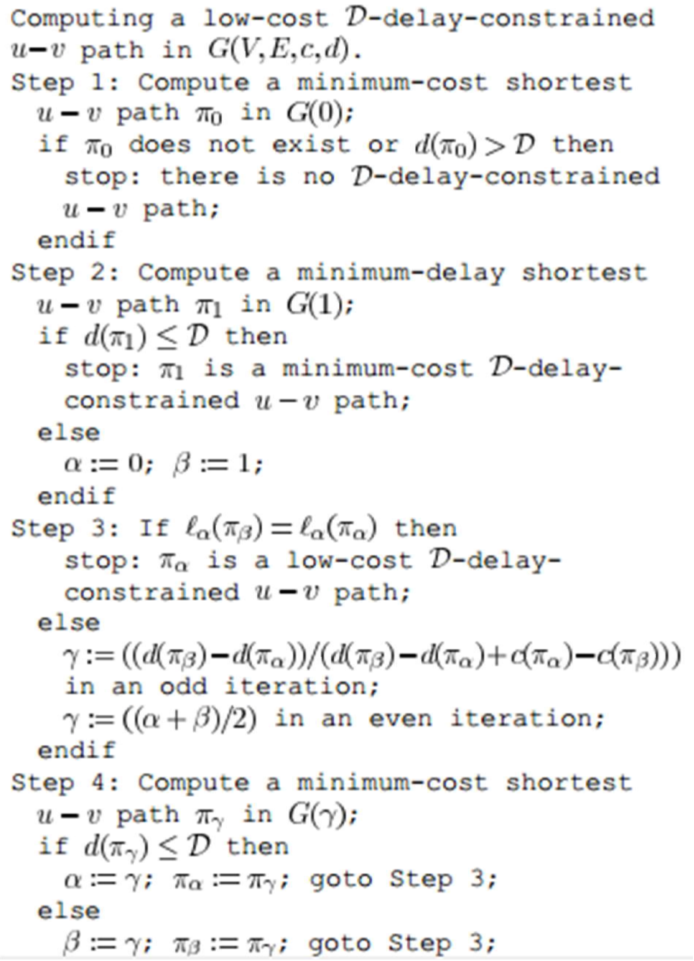
- Only 2 constraints(path cost & delay).
- Improves time delay but Execution time increases(new step).
- Works for known topology(wired networks).
- Simulated for small scaled networks.

2.3 Guoliang's Algorithm

Guoliang Xue [3] tried to improve previous two conditions of Time delay & Execution time in large scale. network can be modeled by an edge-weighted undirected graph ,where is the set of vertices, is the set of edges, is the *cost* of edge, and is the *delay* of edge . Note that this is a somewhat simplified model because in a realistic communication system, delays consist of delays due to propagation, link bandwidth, and queueing at intermediate nodes. This simplified model has been (and still is) widely used in the scientific literature on computer communications and networks because many efficient algorithms for the more complicated model are based on efficient algorithms for this simplified model *Multicasting* consists of concurrently sending the same data from a source to a group of destinations in a computer or communication network. Multicast service plays a more and more important role in computer or communication networks supporting multimedia applications. To support a large number of multicast sessions, a network must minimize the sessions' resource usage, while meeting their quality-of-service (QoS) requirements . To reduce resource usage, a packet from the source node to several destination nodes may share some communication links in the early stages before forking to the destinations. As a result, the packet is transmitted from the source to the destinations in a tree-like network known as a *multicast tree* . Algorithms for computing multicast trees are known as *multicast algorithms* . *Unicast* refers to the special case of a multicast where there is only one destination. Also known as *end-to-end routing* , unicast is another basic operation in communication networks and is often used as a subproblem in the computation of optimal multicast trees.

Solution :

A single algorithm to find the MST & SPT together.



```
Computing a low-cost $\mathcal{D}$-delay-constrained
$u-v$ path in $G(V,E,c,d)$.
Step 1: Compute a minimum-cost shortest
  $u-v$ path $\pi_0$ in $G(0)$;
  if $\pi_0$ does not exist or $d(\pi_0) > \mathcal{D}$ then
    stop: there is no $\mathcal{D}$-delay-constrained
    $u-v$ path;
  endif
Step 2: Compute a minimum-delay shortest
  $u-v$ path $\pi_1$ in $G(1)$;
  if $d(\pi_1) \leq \mathcal{D}$ then
    stop: $\pi_1$ is a minimum-cost $\mathcal{D}$-delay-
    constrained $u-v$ path;
  else
    $\alpha := 0$; $\beta := 1$;
  endif
Step 3: If $\ell_\alpha(\pi_\beta) = \ell_\alpha(\pi_\alpha)$ then
    stop: $\pi_\alpha$ is a low-cost $\mathcal{D}$-delay-
    constrained $u-v$ path;
  else
    $\gamma := ((d(\pi_\beta)-d(\pi_\alpha))/(d(\pi_\beta)-d(\pi_\alpha)+c(\pi_\alpha)-c(\pi_\beta)))$
    in an odd iteration;
    $\gamma := ((\alpha+\beta)/2)$ in an even iteration;
  endif
Step 4: Compute a minimum-cost shortest
  $u-v$ path $\pi_\gamma$ in $G(\gamma)$;
  if $d(\pi_\gamma) \leq \mathcal{D}$ then
    $\alpha := \gamma$; $\pi_\alpha := \pi_\gamma$; goto Step 3;
  else
    $\beta := \gamma$; $\pi_\beta := \pi_\gamma$; goto Step 3;
```

Lackings :

The problems that were still prevailing are-

- Only 2 constraints(path cost & delay).
- Works for known topology(wired networks).

3 Entropy Weight Aggregation

3.1 Introduction

Entropy plays a vital role in multicriteria decision areas and is widely used for criteria aggregation . Using this technology, the multicriteria decision problem can be naturally transformed to be a single-criterion decision problem. The main concept is that various constants are calculated with the given weights for each nodes of the graph. These weights are assigned to various constraints in normalized form. Then they are aggregated into a single one. So now each edge has only one normalized weight that defines it's overall cost.Entropy is the probability of randomness in a system. The weights in these graphs are random and naturally generated. That's the main reason the algorithm has the word Entropy at the start. The motivation of applying this algorithm here is to create a balance among all the network constraints prevailing to an accepted level so that the user and operator both remain satisfied as much as possible.

3.2 Algorithm

Algorithm 1: EWA

```
Input: $G(V, E)$
Output: $G^0(V, E)$
Initialize $X = [x_{ik}]_{m\times K}$; $f_k^U \leftarrow \max\limits_{1\le i\le m}\{x_{ik}\}$;
$f_k^L \leftarrow \min\limits_{1\le i\le m}\{x_{ik}\}$;
Compute $R = [r_{ik}]_{m\times K}$;
Calculate $P = \left[p_{ik}\right]_{m\times K}$, $\delta_k$, $d_k$, and $\alpha_k$;
$w^0(e) \leftarrow \sum\limits_{k=1}^{K} \alpha_k \cdot w_k(e)$;
return $G^0(V, E)$;
```

$$r_{ik} = \begin{cases} \dfrac{f_k^L}{x_{ik}}, & x_{ik} \text{ is the negative type weight} \\ \dfrac{x_{ik}}{f_k^U}, & \text{otherwise.} \end{cases} \tag{1}$$

Obviously, $0 \le r_{ik} \le 1$, the normalized matrix $P = [p_{ik}]_{m\times K}$ can be computed by

$$p_{ik} = \frac{r_{ik}}{\sum\limits_{i=1}^{m} r_{ik}}. \tag{2}$$

Then δ_k, d_k, and α_k are calculated by

$$\delta_k = -\frac{1}{\ln m} \cdot \sum_{i=1}^{m} p_{ik} \ln p_{ik} \tag{3}$$

$$d_k = 1 - \delta_k \tag{4}$$

$$\alpha_k = \frac{d_k}{\sum\limits_{j=1}^{K} d_k}. \tag{5}$$

4 Multiconstraint Multicast Routing in Wired Networks

4.1 Introduction & Analysis

The next proposed[4] two algorithms were considering multiple constraints in wired networks for multicast routing for multimedia communications.

Problems considered :

The criteria considered for this two algorithms were-

- Multiple constraints(>2) including Path cost & time delay
- Execution time minimize

Solution :

The solution approaches are done considering the following criterias-

- Two algorithms for approximating SPT & MST where constraints>2
- Takes Entropy Weight Aggregation into account
- Improves both constraint and efficiency concerns

Lackings :
The only few lackings were remaining though which are -

- Works for known topology i.e. not for wireless networks.

While the above mentioned works mainly focused on constructing an SPT with cost and delay constraints, the case with K constraints $(K > 2)$ has received limited attention.

Multicast routing with *K* constraints is a challenging issue and deserves thorough investigations since the newly emerged services require the constructed multicast trees satisfy three or even more QoS requirements. Two approximation algorithms were proposed for *K*-constrained multicast routing problem in the recent works[4]. However, these two algorithms were designed for regular wireline networks instead of IoT networks with more dynamic topologies and stricter resource constraints.

5 Mulitcast Routing in Wireless Networks in IOT Environment

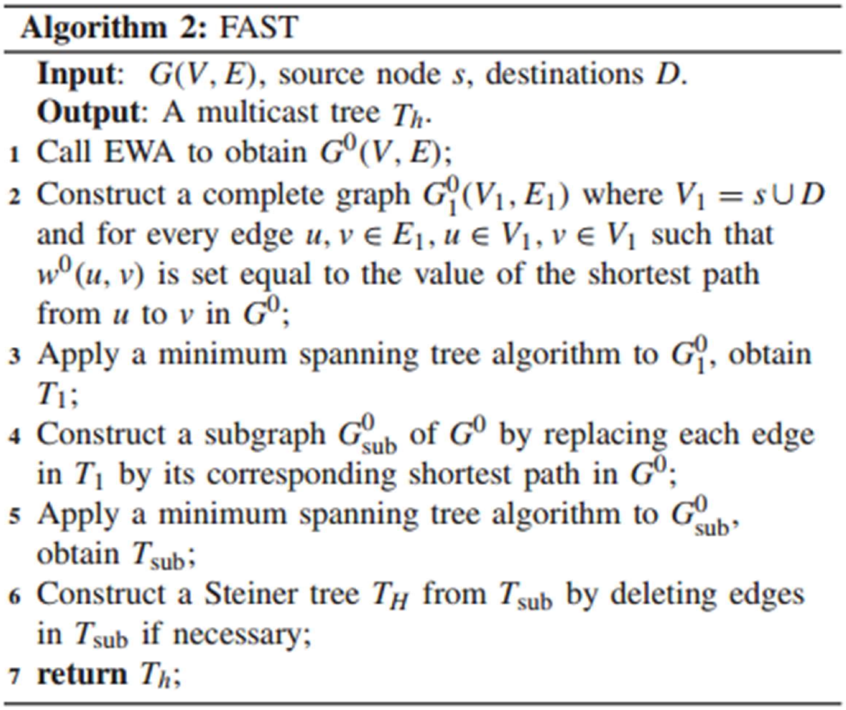
The next problem taken in consideration was to implement all the previous works in an wireless sensor network medium. The next approach was taken in[5] for IOT multimedia communication which requires multicast routing. The solution proposes two algorithms FAST and FAMOUS for estimating the constraints taken in consideration where K>2. The EWA algorithm is the heart of the two algorithms which reduces multiple constraint based network into a single criterion based network discovering the network topology following on-demand routing.Lets discuss the algorithms first.

5.1 FAST Algorithm

On the basis of EWA, a heuristic, called FAST, for MOMT can be derived immediately according to [33]. This algorithm is shown by Algorithm 2, in which EWA is called in the first step to simplify MOMT as a Steiner tree problem. Fig. 1 gives an illustrative example of FAST execution. s is the source and d1, d2, and d3 are destinations. Fig. 1(a) shows the original graph G(V, E) and the weights associated with each edge. Fig. 1(b) gives the graph G0(V, E) with the aggregated weight on each edge obtained by EWA. Fig. 1(c) depicts the graph G01. Fig. 1(d) shows the minimal spanning tree T1 of G0 1.
Fig. 1(e) shows the G0 sub, which is exactly the final solution Th. Notice in this example that lines 5 and 6 are executed without change of G0 sub due to G0 sub = Th. Besides, T1, G0 sub, and Tsub might be not unique in the algorithm, in this case, an arbitrary
one of each graph/tree can be picked for the next step. It can be inferred from FAST that in the worst case, calling EWA takes O(Km) time to calculate the aggregated weight for
each edge. Line 2 can be done in O(mn2) time, line 3 can be done in O(m2) time, line 4 can be done in O(n) time, line 5 can be done in O(n2), and line 6 can be done in O(n) time.

Overall, line 2 dominates the computational time. Therefore, the worst case time complexity of FAST is O(mn2).From [33] we know that FAST is very efficient algorithm with the approximation ratio of 2(1 -(1/l)) when K = 1. l is the number of leaves in the optimal tree. Since multicast routing in IoT calls for more fast MOMT algorithm, we propose the following algorithm. It is worth noting that the algorithms proposed in this section assumes the availability of current network topology and the QoS constraint information associated with network topology.

Such an assumption is common for typical QoS-constrained multicast routing algorithms. How to collect and update the information about network topology and QoS constraints in IoT, although is an important research topic, is out of the scope of the work presented in this paper. With the proliferation of software-defined networking (SDN), which enables a logically centralized controller that maintains a global view of the entire network domain, we expect that collecting and updating network topology and QoS constraint information may be facilitated by application of SDN technologies in the IoT environment.



```
Algorithm 2: FAST
    Input: $G(V, E)$, source node $s$, destinations $D$.
    Output: A multicast tree $T_h$.
1   Call EWA to obtain $G^0(V, E)$;
2   Construct a complete graph $G_1^0(V_1, E_1)$ where $V_1 = s \cup D$
    and for every edge $u, v \in E_1, u \in V_1, v \in V_1$ such that
    $w^0(u, v)$ is set equal to the value of the shortest path
    from $u$ to $v$ in $G^0$;
3   Apply a minimum spanning tree algorithm to $G_1^0$, obtain
    $T_1$;
4   Construct a subgraph $G_{\mathrm{sub}}^0$ of $G^0$ by replacing each edge
    in $T_1$ by its corresponding shortest path in $G^0$;
5   Apply a minimum spanning tree algorithm to $G_{\mathrm{sub}}^0$,
    obtain $T_{\mathrm{sub}}$;
6   Construct a Steiner tree $T_H$ from $T_{\mathrm{sub}}$ by deleting edges
    in $T_{\mathrm{sub}}$ if necessary;
7   return $T_h$;
```

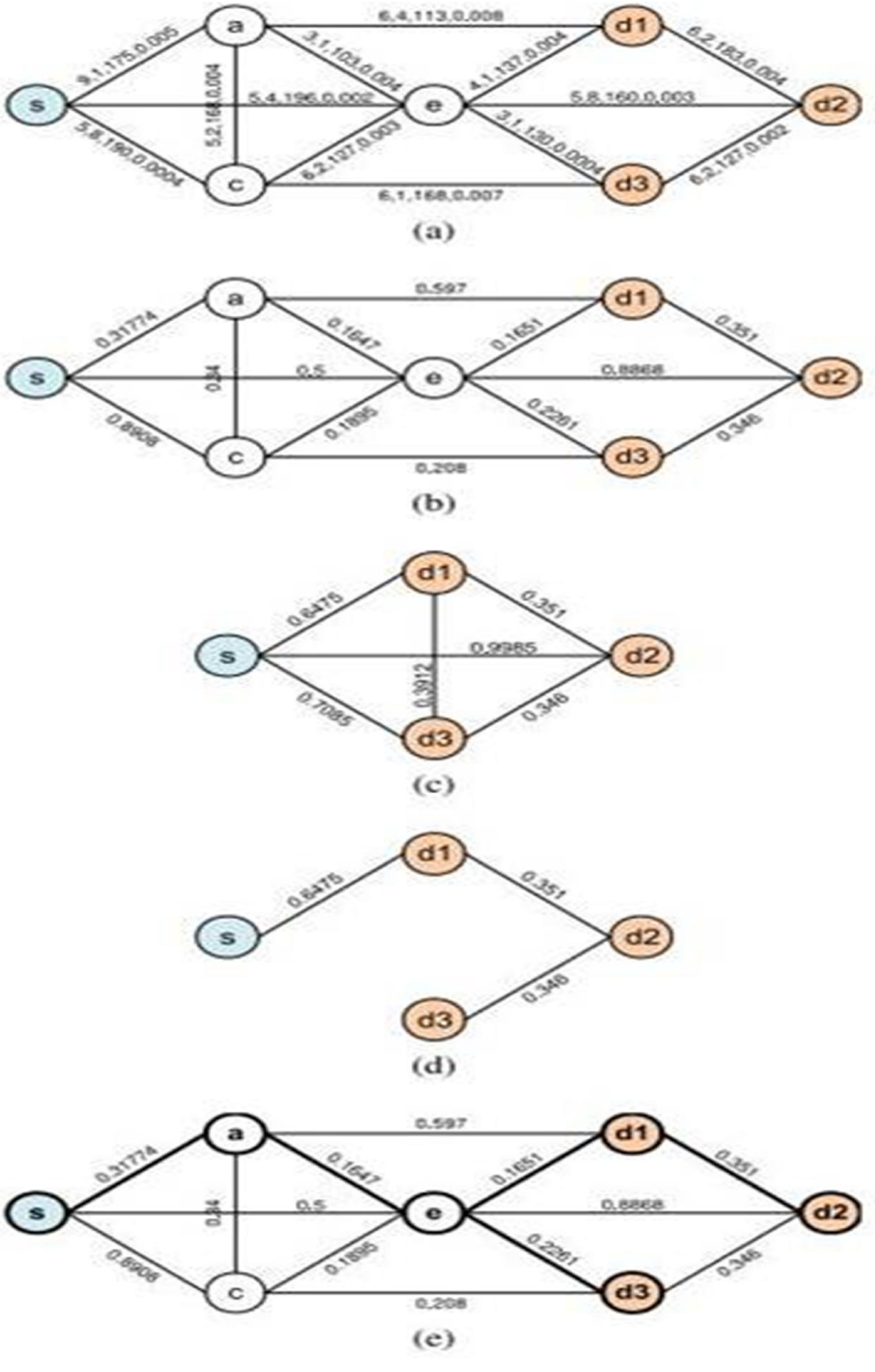


Figure : Illustrative example of FAST algorithm

5.2 FAMOUS Algorithm

As analyzed the above, FAST may not response enough though it can be trivially implemented. In this section, we present a faster algorithm for MOMT.

Algorithm Description

The proposed algorithm, termed FAMOUS (a fast multiconstrained multicast routing algorithm), is given in Algorithm 3, in which the EWA is called in the first step for transforming the problem to be a simple one. The remaining steps of FAMOUS follow the same philosophy of the SPT, where:

Adj[u]: the adjacent nodes of u;

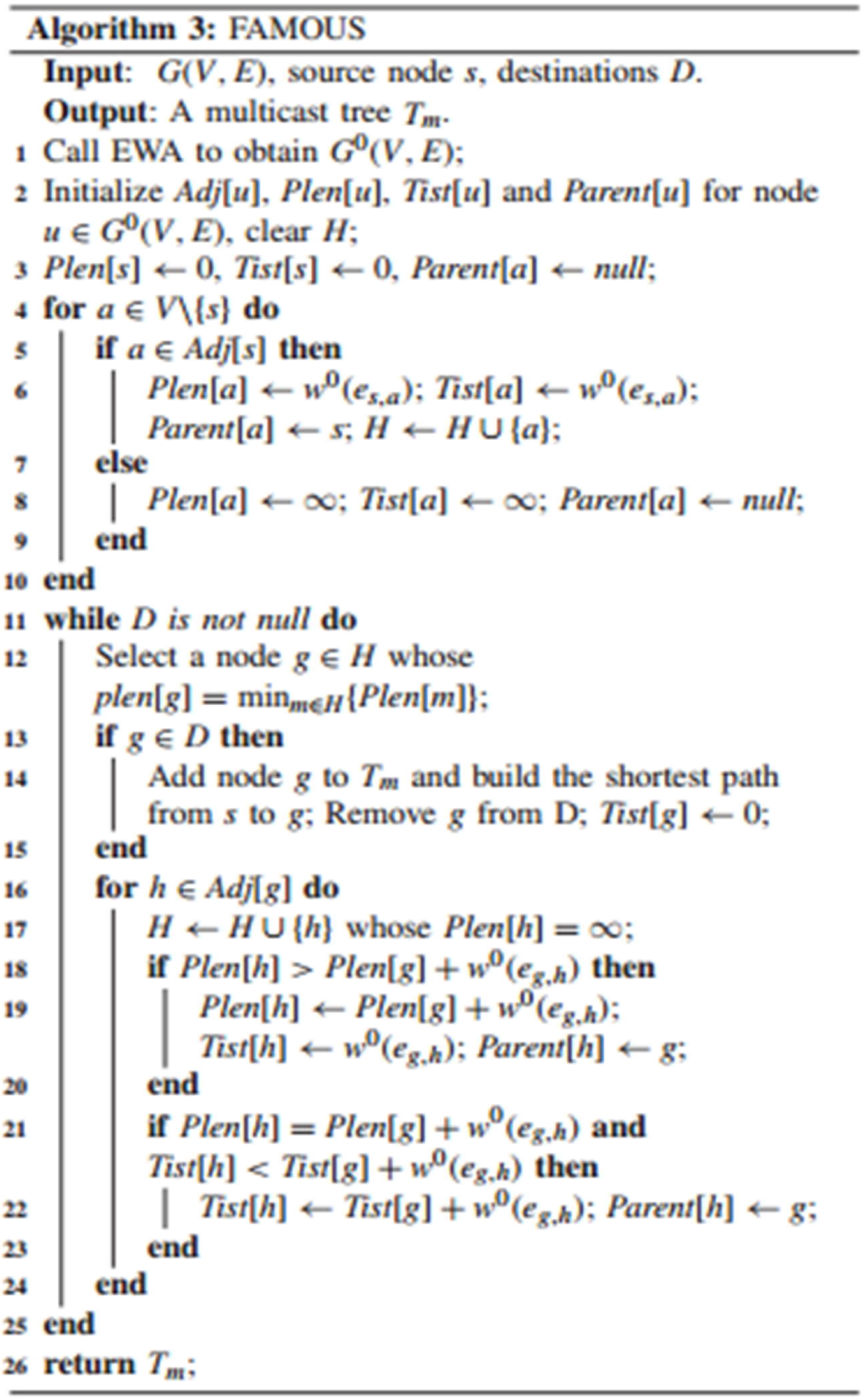
Plen[u]: the length of the shortest path from s to u;

Tist[u]: the shortest length from u to the current tree Tm;

Parent[u]: the parent node of u;

w0(es,a): the aggregated weight of edge (s, a).

It is interesting to notice that the obtained multicast tree Tm may be infeasible since the aggregation of the weights leads to some weight which is likely to exceed the constraint. To remedy this situation, we employ a request-filtering process executed in advance to assure that the request, i.e., the constraint is valid. Thus, it is guaranteed that Tm is a feasible multicast tree. Fig. 2 provides an illustrative example of the execution of FAMOUS, where s is the source and d1, d2, d3 are destinations. Fig. 2(a) shows the original graph G(V, E) and the weights associated with each edge. Fig. 2(b) gives the graph G0(V, E) with the aggregated weight on each edge obtained by EWA. Fig. 2(c) depicts the process of adding the destination node d1 to the multicast tree. It seems that Fig. 2(d) shows the situation of adding for d3 to join the tree. Fig. 2(e) describes how the last destination node d2 joins the tree whereby the multicast tree Tm is eventually formed.



Algorithm 3: FAMOUS

```
Input: $G(V, E)$, source node $s$, destinations $D$.
Output: A multicast tree $T_m$.
Call EWA to obtain $G^0(V, E)$;
Initialize $Adj[u]$, $Plen[u]$, $Tist[u]$ and $Parent[u]$ for node $u \in G^0(V, E)$, clear $H$;
$Plen[s] \leftarrow 0$, $Tist[s] \leftarrow 0$, $Parent[a] \leftarrow null$;
for $a \in V \backslash \{s\}$ do
    if $a \in Adj[s]$ then
        $Plen[a] \leftarrow w^0(e_{s,a})$; $Tist[a] \leftarrow w^0(e_{s,a})$; $Parent[a] \leftarrow s$; $H \leftarrow H \cup \{a\}$;
    else
        $Plen[a] \leftarrow \infty$; $Tist[a] \leftarrow \infty$; $Parent[a] \leftarrow null$;
    end
end
while $D$ is not null do
    Select a node $g \in H$ whose $plen[g] = \min_{m \in H}\{Plen[m]\}$;
    if $g \in D$ then
        Add node $g$ to $T_m$ and build the shortest path from $s$ to $g$; Remove $g$ from D; $Tist[g] \leftarrow 0$;
    end
    for $h \in Adj[g]$ do
        $H \leftarrow H \cup \{h\}$ whose $Plen[h] = \infty$;
        if $Plen[h] > Plen[g] + w^0(e_{g,h})$ then
            $Plen[h] \leftarrow Plen[g] + w^0(e_{g,h})$; $Tist[h] \leftarrow w^0(e_{g,h})$; $Parent[h] \leftarrow g$;
        end
        if $Plen[h] = Plen[g] + w^0(e_{g,h})$ and $Tist[h] < Tist[g] + w^0(e_{g,h})$ then
            $Tist[h] \leftarrow Tist[g] + w^0(e_{g,h})$; $Parent[h] \leftarrow g$;
        end
    end
end
return $T_m$;
```

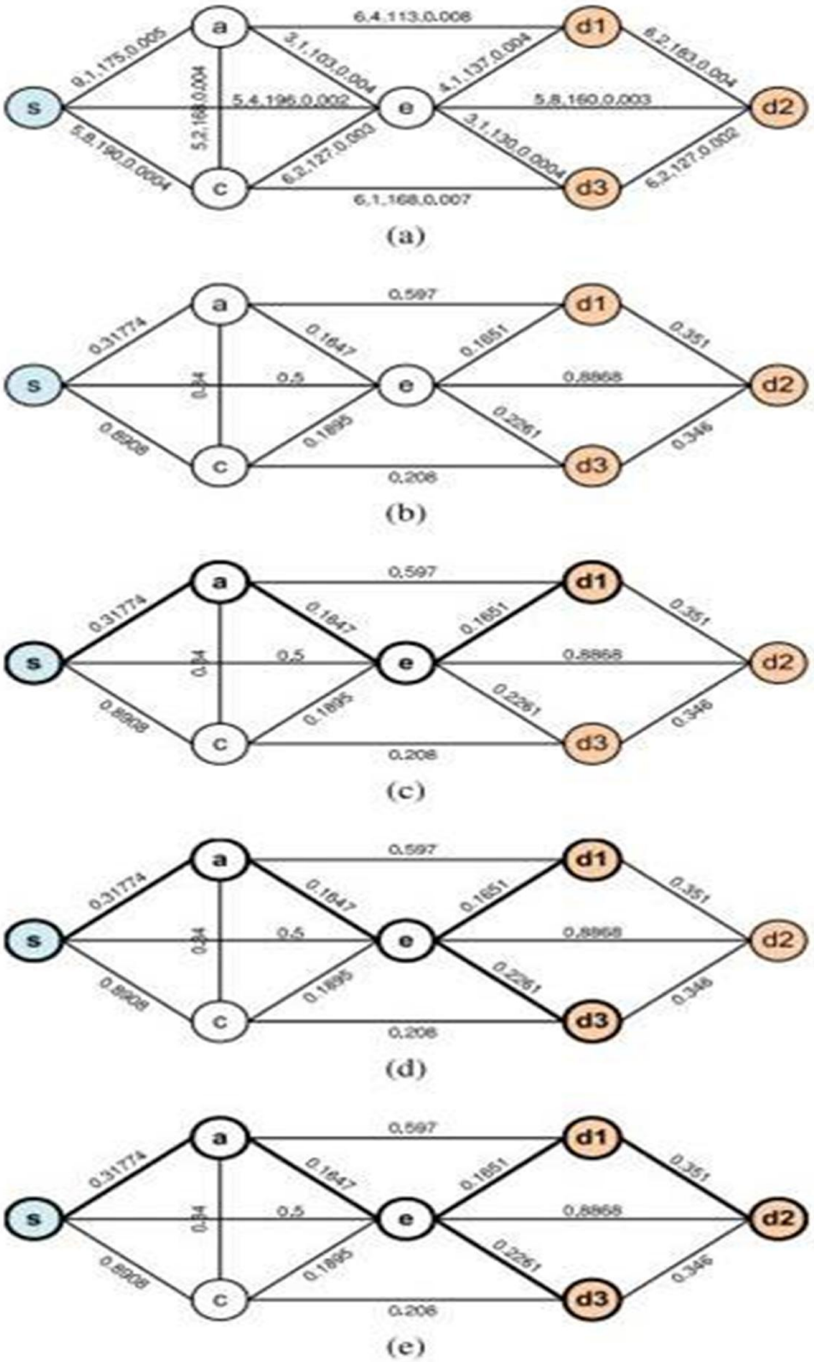


Fig. 2. Illustrative example of FAMOUS execution

5.3 Performance Metric

The performance metrics used in the rest of this section for evaluating FAMOUS and HeurMOMT are defined as follows.

Execution Time: It indicates the average running time of an algorithm by its one hundred independent runs. This metric is used for evaluating time cost performance of an algorithm, i.e., speed.

Average Weight: It denotes an aggregated weight of a tree returned by the algorithm, that is, for a tree *T* returned by an algorithm

$$\text{Average weight} = \frac{\left[\sum_{k=1}^{K} w_k^2(T)\right]^{\frac{1}{2}}}{|D|}.$$

This metric reflects the quality of solution, i.e., the accuracy of the algorithm where wk is the aggregated weight on each of the edges of the tree.

5.4 Setup and Evaluation

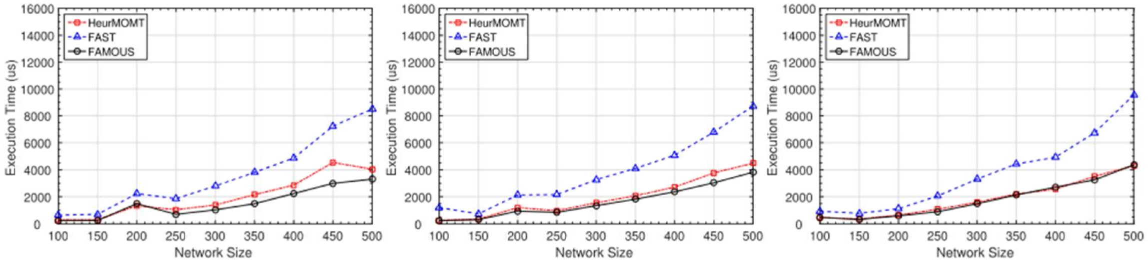
In this section, the performance of FAST and FAMOUS are compared[5] against that of a representative multicast routing algorithm HeurMOMT via extensive experiments. They[5] selected HeurMOMT as the baseline for comparison because FAST and FAMOUS as well as HeurMOMT fall into the same algorithmic category, that is, they are all heuristics. The comparison is conducted in two aspects: 1) the speed and 2) the accuracy of the algorithm.

Experiment Setup

We adopt the same experiment settings from in order to compare the performance unbiasedly. A set of random network topologies generated by Waxman model is adopted here: the nodes are randomly placed in a one-by-one square, and the probability of creating a link between node u and node v is $\alpha \cdot e\text{-}d(u,v)/\beta L$, where $d(u, v)$ is the distance between u and v, and L is the maximum distance between any two nodes. A and β are set to 0.6 and 0.4, respectively, to guarantee that each generated topology is a connected graph. The network size used in the experiment ranges from 100 nodes to 500 nodes, and the number of destinations is set to 5 and 10. Regarding QoS parameters, we set K to 2, 3, and 4, i.e., each edge is associated with two or three or four weights, which are uniformly generated in a given range [1, 100]. The constraint are all set to 1000. In our experiments, the average results are reported by running each test instance one hundred times independently, and all the experiments are run on an IBM P4 2.4 GHz PC with 4GB memory.

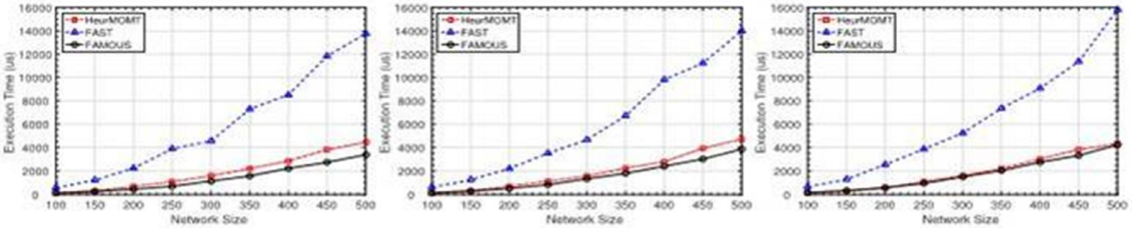
Results

Figs. 3 and 4 show the execution time of three algorithms with different network sizes when $|D| = 5, 10$, $K = 2, 3, 4$, respectively. It is obvious that the execution time of FAMOUS is the lowest among three curves and that of FAST is the highest. This observation indicates that FAMOUS is more time-efficient than other two algorithms. On the other hand, Figs. 5 and 6 delivers the comparison results of average weights. The results show that the average weights of FAMOUS is slightly lower than those of FAST and HeurMOMT, which means that the solution found by FAMOUS has the best quality compared with other two algorithms.



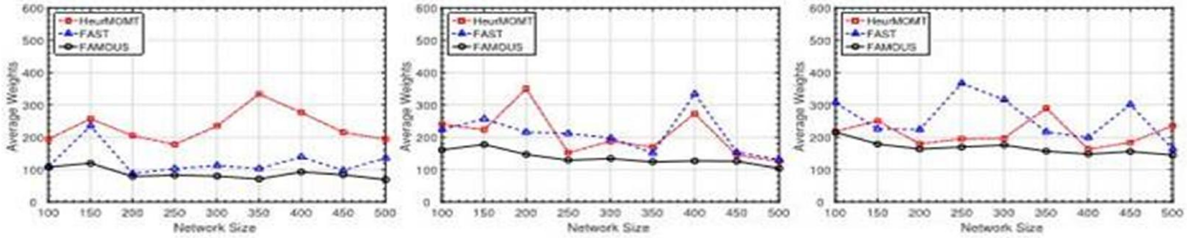
(a) (b) (c)

Fig. 3. Execution time comparison. (a) $|D| = 5, K = 2$. (b) $|D| = 5, K = 3$. (c) $|D| = 5, K = 4$.



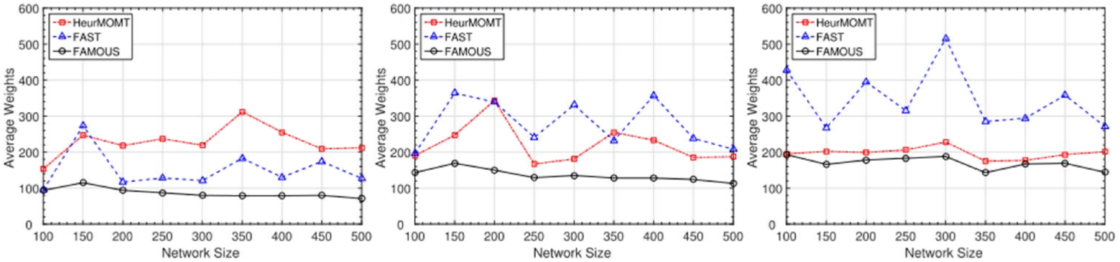
(a) (b) (c)

Fig. 4. Execution time comparison. (a) $|D| = 10, K = 2$. (b) $|D| = 10, K = 3$. (c) $|D| = 10, K = 4$.



(a) (b) (c)

Fig. 5. Average weights comparison. (a) $|D| = 5, K = 2$. (b) $|D| = 5, K = 3$. (c) $|D| = 5, K = 4$.



(a) (b) (c)

Fig. 6. Average weights comparison. (a) $|D| = 10, K = 2$. (b) $|D| = 10, K = 3$. (c) $|D| = 10, K = 4$.

6 Experiment

In our thesis work, we tried to simulate the FAMOUS algorithm described theoretically above[5].We used sample random mechanism in C++ environment to setup our graph and implemented the EWA first and called it from the FAMOUS algorithm code directly for a fixed case as the nodes were not dynamic in C++

6.1 Algorithm

Parameters

- Multiple constraints(2 to 4)
- Network size = 7
- Destinations 3

We implemented the EWA algorithm in C++ in the following way.The lines ccan be matched with the exact algorithm for better understanding.

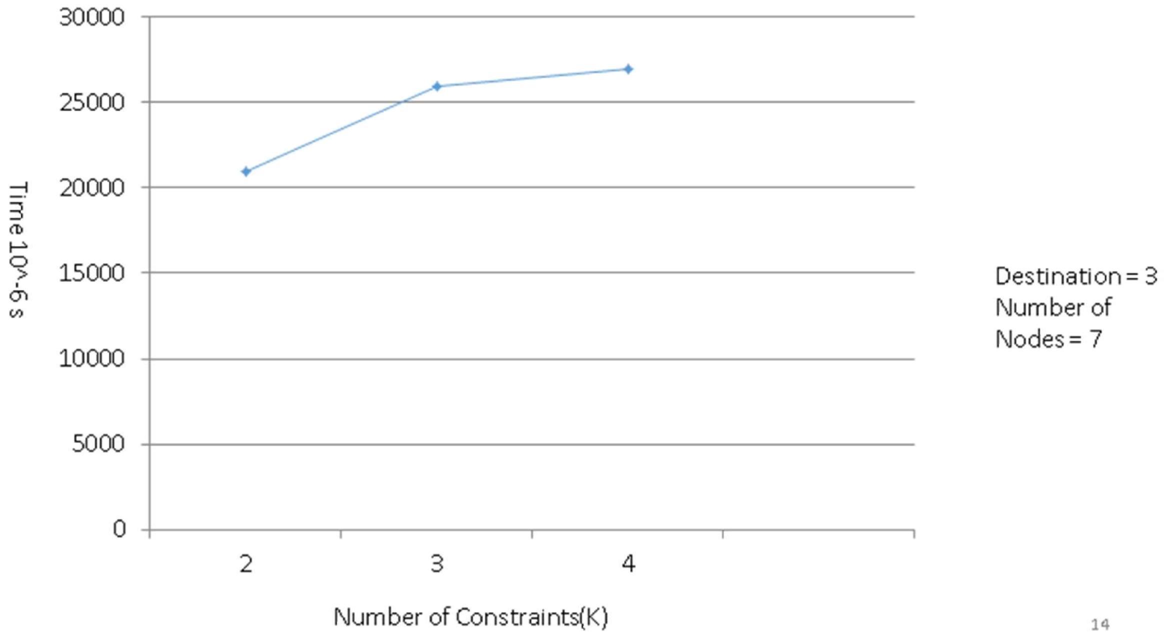
```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>

struct edges
{
char name[10];
double cost[5];
double w;
} x[13];

int main()
{
FILE *fp = fopen("1.txt","r");
int i,j,k;
for(i=0; i<13; i++)
fscanf(fp, "%s %lf %lf %lf
%lf",&x[i].name,&x[i].cost[0],&x[i].cost[1],&x[i].cost[2],&x[i].cost[3]);
```

```
fclose(fp);

double mx[5],mn[5];

for(j=0; j<4; j++)
{
mx[j] = -1;
mn[j] = 10000;
for(i=0; i<13; i++)
{
if(x[i].cost[j]>mx[j]) mx[j] = x[i].cost[j];
if(x[i].cost[j]<mn[j]) mn[j] = x[i].cost[j];
}
}

double r[13][5];

for(i=0; i<13; i++)
{
for(j=0; j<4; j++)
{
if(x[i].cost[j]<0) r[i][j] = mn[j]/x[i].cost[j];
else r[i][j] = x[i].cost[j]/mx[j];
}
}

double hor;

for(j=0; j<4; j++)
{
hor = 0;
for(i=0; i<13; i++)
hor += r[i][j];

for(i=0; i<13; i++)
{
r[i][j] = r[i][j]/hor;
}
}

double deltaK[5], dK[5], alphaK[5], sumDK=0, temp;
```

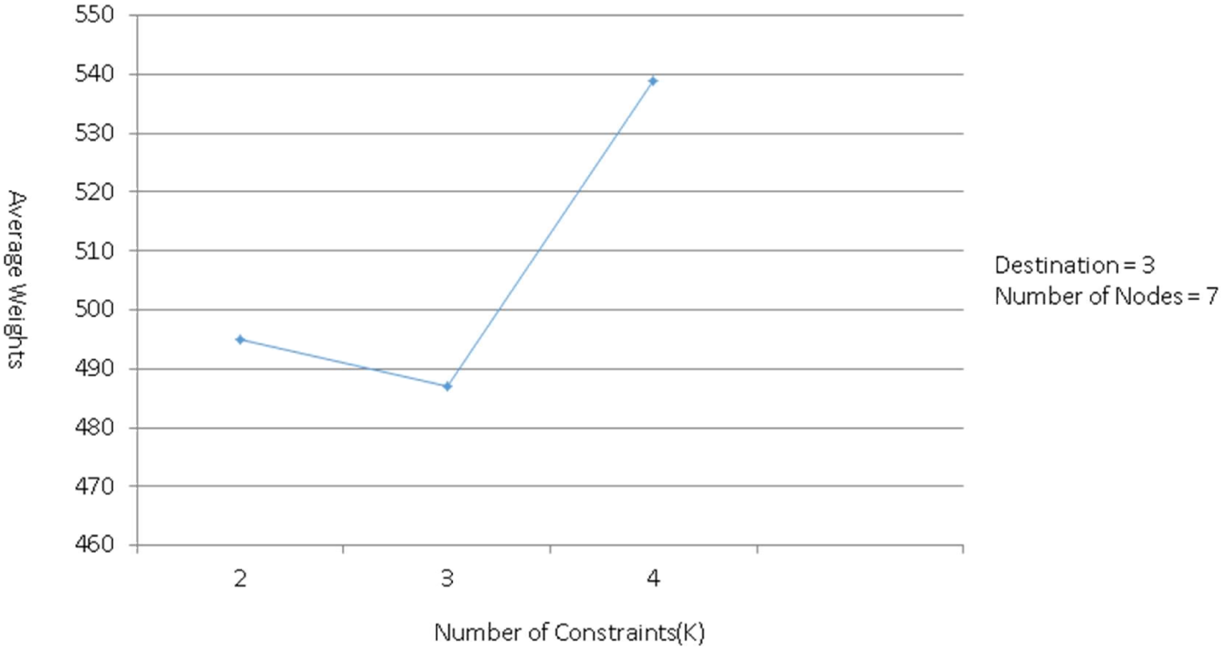
```
for(j=0; j<4; j++)
{
deltaK[j] = -(1/log(13));
temp=0;
for(i=0; i<13; i++)
{
temp += r[i][j]*log(r[i][j]);
}
deltaK[j] *= temp;
dK[j] = 1 - deltaK[j];
sumDK += dK[j];
}

for(j=0; j<4; j++)
{
alphaK[j] = dK[j]/sumDK;
//printf("%lf\n",alphaK[j]);
}

for(i=0; i<13; i++)
{
x[i].w = 0;
temp = 0;
for(j=0; j<4; j++)
{
x[i].w += alphaK[j]*x[i].cost[j];
temp += x[i].cost[j];
}
x[i].w = x[i].w/temp;
//printf("%s : %lf\n",x[i].name, x[i].w);
}

return 0;
}
```

6.2 Results

The result that we got are shown to compare taking varied number of constraints in the following graphs implying Aberage weight metric and Time of Execution metric.



Graph : Execution time vs K



Graph : Average weight vs K

6.3 Lackings

The experiment consists of very few nodes. Only 7 nodes are considered and 3 of them are destinations. So our results are not enough for large data sets.

Simulation of this in any TinyOS system is also very expensive as it requires a huge amount of sensor nodes to form a network and then define their factors. This research side is also fairly new, so there’s not enough data available for this.Thus the theoretical data that was described in [5] could not be exactly verified by our work here.

7 Conclusion and Future Work

To tackle the challenging problem of multicast routing for multimedia communication in the IoT, in [5] the proposed two algorithms with $K > 2$ constraints works theoretically very fine. By applying the entropy technique to aggregate multiple constraints into a comprehensive metric, the proposed algorithms dramatically reduce the complexity of multiconstrained multicast routing problem and enables application of some well-known algorithms to solve the problem. The theoretical analysis on the complexity and approximation of the proposed algorithms, and conducted extensive simulations to evaluate performance of the algorithm are demonstrated well in[5]. Both analytical and experimental results have demonstrated that the one of the proposed algorithms is superior to a representative multiconstrained multicast routing algorithm in terms of both speed and accuracy. The findings provide in-depth insight into the multicast routing algorithm design for multimedia communications in IoT.

The future works ahead with this could be as follows –

- Energy allocation for sensor nodes are a drastic factor in WSN based IOT environment. No measurements for that purpose is taken in the algorithms.
- No precautionary measures against a CONGESTED NODE found in the optimal path is taken. So a new constraint factor for the algorithms could well be CONGESTED NODES.
- The algorithms are not well aware of the new rising trend called “Wastage aware routing in Energy harvesting WSNs”. In this type of networks the sensor nodes harvests energy from nature for recharging. Considering this would give the algorithms more better result in future.
- Also we would like to simulate the algorithms in a small scale real life IOT environment and try to find the solution to the future works discussed above.

REFERENCES :

1) S. Khuller, B. Raghavachari, and N. Young, “Balancing minimum spanning trees and shortest-path trees,” Algorithmica, vol. 14, no. 4, pp. 305–321, 1995

2) M. Parsa, Q. Zhu, and J. J. Garcia-Luna-Aceves, “An iterative algorithm for delay-constrained minimum-cost multicasting,” IEEE/ACM Trans. Netw., vol. 6, no. 4, pp. 461–474, Aug. 1998

3) G. Xue, “Minimum-cost QoS multicast and unicast routing in communication networks,” IEEE Trans. Commun.,vol. 51, no. 5, pp. 817–824, May 2003.

4) J. Huang, Y. Tanaka, and Y. Ma, “On approximating a multicast routing tree with multiple quality-of-service constraints,” IEICE Trans. Commun., vol. E95-B, no. 6, pp. 2005–2012, Jun. 2012

5) “Multicast Routing for Multimedia Communications in the Internet of Thing” – Jun Huang, Senior Member, IEEE, Qiang Duan,Member, IEEE, Yanxiao Zhao, Member, IEEE, Zhong Zheng, Member, IEEE, and Wei Wang, Senior Member, IEEE; IEEE INTERNET OF THINGS JOURNAL,VOL. 4, NO. 1, FEBRUARY 2017