An Optimized Link Selection Function for Solving Delay-constrained Multicast Routing Problem

Yuanchen Li[1, a], Weiqun Liu[2, b]

[1]Academy of Information Technology, Luoyang Normal University,Luoyang,471022,China

[2] Academy of Information Technology, Luoyang Normal University,Luoyang,471022,China

[a]email: yuanchen_li@126.com, [b]email:liuweiqun1@163.com

Keywords: Routing Optimization; Delay-constrained; Link Selection Function

Abstract. In multicast routing messages are sent from the source node to all destination nodes. In order to meet QoS requirements an optimizing algorithm is needed. We propose an optimization algorithm based on the KPP and QDMR algorithm to do so. This algorithm uses optimized link selection function to modify the paths from the source node to the destinations, the nodes whose delay are smaller than the average achieves a new delay as big as possible and the nodes whose delay are above the average achieves a new delay as small as possible. This can give an effective optimal solution to multicast routing problem.

Introduction

As the rapid development of the Internet, applications such as multimedia is becoming more and more diverse, and much more wide range of real-time multimedia communications, multicast is becoming an important requirement to support these multimedia services. To support this multicast communications, networks have to guarantee an upper bound on the end-to-end delay from the source node to each of the destination nodes. So, a network must minimize the resource consumption, while meeting their quality of service(QoS) requirements. The current approach for efficiently supporting a multicast session in a network consists of establishing a multicast tree along which session information is transferred. Algorithms are needed to structure multicast tree. A good multicast routing algorithm is often able to shorten the time consumption and meet certain QoS constraints on the minimum network bandwidth consumption, and thus use of network resources efficiently. Finding the minimum cost multicast routing tree can be formalized as the Steiner tree problem. Solving the Steiner tree problem is NP-hard problem[1], therefore the researchers put forward to calculating the Steiner heuristic tree, there have been some mature algorithms at present. The BSMA algorithm[1] constructs a minimum delay tree by using the Dijkstra algorithm to obtain low-cost path under the delay constraint. Kou, Markousky and Berman had found out any meet the delay constraint minimum cost path between network nodes, and thus they built a heuristic algorithm called KMB algorithm to produce the multicast tree[2]. In view of the above questions, this paper studies the problem of establishing Steiner tree under the end-to-end delay constrains and put forward a new heuristic multicast routing algorithm based on above method and MPH algorithm. Both theoretical proof and simulation results indicate that this algorithm has the performance of well delay and lower cost.

Network Model And Problem Pacification

A network can be modeled as a graph $G = (V, E)$, where V is the set of nodes and E is the set of edges. The nodes in V represent the hosts or the routers in the network. The edges in E represent the communication links connecting the routers. $|V|$ are the nodes of the graph, $|E|$ are the number of edges or the link numbers in G ($e=(x,y), e\in E, x, y\in V$). Defining two positive real functions on every edge e： cost function($cost(e) : E\rightarrow R^+$) and delay function($delay(e) : E\rightarrow R^+$). In a multicast communication network, there is a source node $s\in V$ and a member of destination nodes $M\subseteq V$. s is the node corresponding to the source that generates the data. The nodes in M , called the multicast group members correspond to the destinations that receive the data. The delay constraint Steiner tree *T* is rooted as node *s* and can reach all the destination nodes, and under the delay constraint, make the network cost minimum. Now we are ready to formally define our problem as follows.

Definition 1 P (*u*,*v*)

P(*u*,*v*) indicates that the path from node $u \in V$ to node $v \in V$, path delay is defined as the sum of the delay of each link, path cost is defined as the sum of the cost of each link.

Definition 2 Delay and cost

According Definition 1,Delay of the *P*(*u*,*v*) is defined as total delay of the link and cost as the total cost of the link of *P*(*u*,*v*), respectively written for follows:

$$D(p(u,v)) = \sum_{e \in p(u,v)} D(e) \tag{1}$$

$$C(p(u,v)) = \sum_{e \in p(u,v)} C(e) \tag{2}$$

Definition 3 Steiner tree

Given graph *G*=(*V*,*E*),a source node *s* ,destination node sets $M \subseteq V - \{s\}$. If there is a minimum spanning tree *T* and

$$\min\left\{ \sum_{v \in D, e \in T} C(e) \right\} \tag{3}$$

So the tree is defined as Steiner tree.

Definition 4 Delay constraints Steiner tree

Given a delay constraint (or delay bound)*Δ* in Steiner tree *T*, if *T* is minimized subject to the following constraint:

$$\sum_{e \in p(s,v), v \in D} D(e) \le \Delta \tag{4}$$

Where *Δ* is a positive real number, indicating from the source node to any node in the path that must be met end to end delay constraints, *p*(*s*,*v*) indicates that the path from *s* to *x* in the multicast tree. This tree is called Delay constraints Steiner tree.

Work background

KPP[1] algorithm uses the idea of the minimum spanning tree algorithm to deal with the cost of the path and uses the shortest path tree algorithm to deal with the delay of the path, so that the cost of the link path added to the multicast tree each time is the minimum. It's link selection function is defined by following:

$$f(u,v) = \begin{cases} cost(u,v)/(\Delta - p(u) - delay(u,v), - p(u) - delay(u,v) > \Delta \\ \infty, \text{otherwise} \end{cases} \tag{5}$$

Where *p(u)* is the delay from the source node *s* to node *u*, *cost(u,v)* is the cost of *p(u,v)*, *delay(u,v)* is the delay of *p(u,v)* .

Using the greedy idea, DDMC[3] algorithm defines the new link selection function:

$$Cost(v) = I_D(u) Cost(u) + Cost(u,v) \tag{6}$$

Where *cost*(*v*) is the cost of *p(s,v)*,*cost*(*u*,*v*) is the cost of *p(u,v)*, $v \notin T$， $u \in T$. I_D is the indicator function(*D* or *M* are the set of destination nodes)，defining by

$$I_D(u) = \begin{cases} 0, u \in D \\ 1, u \notin D \end{cases} \tag{7}$$

QDMR[4] modified the indicator function of DDMC, using

$$I_D(u) = \begin{cases} Delay(u) / \Delta, u \in D \\ 1, u \notin D \end{cases} \tag{8}$$

The algorithms above can construct certain delay constrained multicast tree under the condition of constraints of the multicast tree and be used in actual network. But they all exist obvious flaws, namely node needs to gather the global information network, especially in g a higher complexity multicast tree. Therefore, we will design a new algorithm to solve multicast tree problem.

The Algorithm We Proposed

Based on the above analysis, selection function in algorithm not only consider to give priority to the destination nodes, but also to give priority to the path with lower delay, therefore, *cost(v)* should be the compromise of the path with lower cost and the path has bigger surplus delay. Thus, if choose the node with lower value of *cost(v)* to join the multicast tree, the choice is not only the path with lower cost, but also has bigger surplus delay. So it can reduce the probability of constructing the multicast tree by the minimum delay path, and improve the efficiency.

1.Definition link selection function

After a comprehensive consideration and according KPP,DDMC and QDMR algorithm ,this paper presents the following link selection function:

$$Cost(v)=\begin{cases}\dfrac{I_D(u)Cost(u)}{(\Delta-(p(u)+Delay(u,v)))^{\lambda}}, p(u)+Delay(u,v)<\Delta \\ \infty, otherwise\end{cases} \quad (9)$$

Where, *Cost*(*v*) is the cost of p(s,v),*Delay*(*u*,*v*) is the delay of p(u,v),*P*(*u*) is the delay of p(s,v), I_D is the indicator function. $\lambda > 0$ as the balance parameter between the path cost and path delay. When λ is smaller, Cost(v) is bigger, stress to the cost optimization; When λ is bigger, Cost(v) is smaller, stress to the delay optimization.

In actual network environment, (9) usually written by

$$Cost(v)=\begin{cases}log(I_D(u)Cost(u))-\lambda\, log(\Delta-(p(u)+Delay(u,v))), p(u)+Delay(u,v)<\Delta \\ \infty, otherwise\end{cases} \quad (10)$$

The indicator function this paper is defined by

$$I_D(u)=\begin{cases}p(u)/(\Delta+p(u)), u\in D\cup D' \\ 1, otherwisw\end{cases} \quad (11)$$

2. Algorithm description

Our algorithm procedure is as follows:

Step 1 initialization. *T*=Ø，*S*={*s*}，*cost*(*s*)=0，*delay*(*s*)=0； ***Step 2*** Use the Dijkstra algorithm to calculate the minimum delay tree from the source node *s* to any destination node, then judge whether there is a low cost of delay-constrained multicast tree; if there is not exist, then exit immediately; ***Step 3*** Calculate the cost from the source node *s* to each of the node *u* connecting with it, and then choose the node which has the minimum *cost(u)* value to join the multicast tree, at the same time make *S*=*S*∪{*u*}; ***Step 4*** According to the path selection function and the instruction function(10) and (11), calculate the cost(*v*) from the node u which has been added to the multicast tree to the node *v* which connecting with it, then choose the minimum value of cost(*v*) to join the multicast tree, at the same time make *S*=*S*∪{*v*} .Modifies *delay(v)*; ***Step 5*** for all *u*∈*p(s,v)*,calculating *cost*(*u*) and selecting *u*∈min(*cost*(*u*)) to tree *S*，that is *S*=*S*∪{*u*}; ***Step 6*** repeat step(3), step(4)and step(5),add as many as destination nodes to the multicast tree, if all the destination nodes are added to the multicast tree, then hot, otherwise turn to the step(7); ***Step 7*** add the minimum delay path from the source node to the destination node to the multicast tree, if there are loops, just simply delete the parent node that forms the loop edges, at the same time delete the leaf nodes which are not the destination nodes. output T.

Algorithm Analysis

The simulation used in this paper is proposed in [5]. The connectivity between nodes is determined by:

$$p(u,v)=\beta\exp\left(\frac{-d(u,v)}{\alpha L}\right) \quad (12)$$

Where *d*(*u*, *v*) is the distance between *u* and *v* , *L* is the max distance between two arbitrary nodes, α and β are the parameters to regulate the network map feature. In order to make the random network model near to the realistic, this paper select α=0.4,β=0.2 , λ= 20. The dates used in the simulation experiment are the average result of 100 times, different source nodes and receive nodes are in the same conditions.

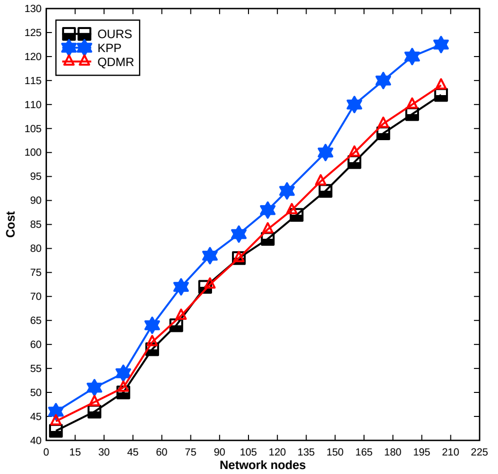

Fig. 1 The relationship between the cost and the network nodes

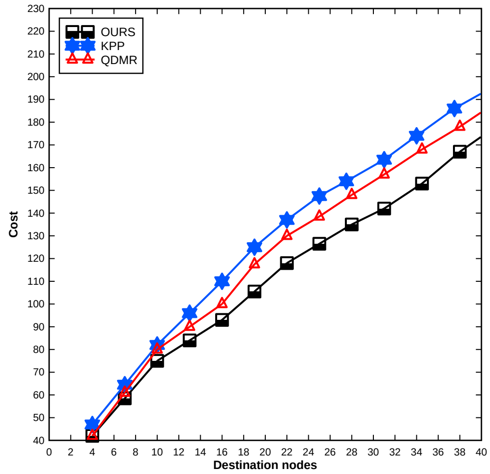

Fig. 2 the relationship between the cost and the destination nodes

Fig. 1 shows the relationship between the cost and the network nodes. The network node increase from 5 to 210, and each time the number of nodes in the network increased by 15, delay constraint remains 0.02s. From the figure we can see the multicast tree cost generated by our algorithm is lower than KPP and QDMR algorithm. Fig. 2 shows the relationship between the cost and the destination nodes, delay constraint remains 0.02s, the destination nodes in this experiment group from 1 to 40, from the figure can be seen, the algorithm we proposed is better than KPP and QDMR algorithm.

Summary

The proposed algorithm in this paper is based on the idea of the KPP and QDMR algorithm. The algorithm can use lower time complexity and cost to construct the multicast tree, and the multicast tree constructed has superior performance, so as to meet the need of real-time multimedia applications effectively. Simulation results show that the algorithm has low cost and good comprehensive performance.

Acknowledgement

This work was financially supported by the Henan Natural Science Foundation (112102310527) , (142102210473) and (14B520016).

References

[1] Kompella V P, Pasqual J C,Polyzos G C. Multicast routing for multimedia communication [J].IEEE/ACM Trans on Networking ,1993,1(3):286-292.

[2] L kou, G markowsky, L berman. A fast algorithm for Steiner trees in graphs[J].Acta Informatica,1981,15(2):141-145.

[3] Shaikh A, Shin K G. Destination-driven routing for low-cost multicast[J].IEEE Journal on Selected Areas in Communications,1997,15(3):373-381.

[4] L guo, I matta. QMDR: An efficient QoS dependent multicast routing Algorithm[C]. Proc of the 5th IEEE Real-Time Technology and Applications Symposium,1999:213-222.

[5] Waxman B. Routing of multipoint connections [J].IEEE Journal on Selected Area in Communications, 1988, 6(6):1617-1622.