A Multicast Routing Algorithm Under the Delay-Restricted Network Environment

Jinpeng Wang(✉), Xin Guan, Yang Zhou, Fan Cao, and Nianyu Zou

Information Science and Engineering College,
Dalian Polytechnic University, Dalian 116034, China
wangjp@dlpu.edu.cn

Abstract. Nowadays, the network is expected to service much more multimedia data with the improvement of the network communication bandwidth and development of the processing capacity. In the multicasting communication system, once every receiving end separately sends the data packets, the network resources intend to be wasted, and the calculating stress on the nodes is also going to be increased. A distributed delay-restricted multicast route heuristic method DMPH (delay-constrained minimal-cost path heuristic) is presented in this article. This algorithm proposed in the article can achieve a convergence speed, back up dynamic multicast, and offer the pretty good the network overhead performance. In order to gain the better performance of the network cost, the article firstly proposes a mathematical topological model of the problem over the delay-constrained multicast routing, and then presents a dynamic multicast delay-restricted multicast route algorithm which is called DMPH (Delay-Constrained Minimal-Cost Path Heuristic), fast-convergent and distributed. The computing simulation results suggest that the method pro-posed can gain the better performance of the network cost. In addition, the article just addressed the operation of the addition and quit of the nodes on the dynamic varying of the multicast group, but not concerned the over-head optimal. These problems mentioned above are all our future research work.

Keywords: Multicast · Delay-restricted multicast · Routing · Topology

1 Introduction

Nowadays, the network is expected to service much more multimedia data with the improvement of the network communication bandwidth and development of the processing capacity [1]. And the network is required that they should have perfect capacity on multicast by most of these services, especially for many multimedia services, just like the video/audio meeting online [2], the exchanging mock simulation, the game for multiplayer, and the distributed database [3] etc.

In the multicasting communication system, once every receiving end separately sends the data packets, the network resources intend to be wasted, and the calculating stress on the nodes is also going to be increased.

X. Jiang and P. Li (Eds.): GreeNets 2020, LNICST 333, pp. 343–349, 2020.
https://doi.org/10.1007/978-3-030-62483-5_35

In common meaning, it is pretty difficult to deal with the problem about the routing on QoS multicast [4]. The most important thing to a delay-constrained Steiner problem is to address the cost-optimization multicast tree under the delay-restricted environment [5]. Because the NP-complete is the key to address the optimization delay-restricted problem [6], the increase of the algorithm computing complexity is not able to be illustrated via the polynomials with the increase of the network ends. While the network is big, finding a Steiner tree which is unsuitable for network multicast applications intends to use a long time. Despite the heuristic methods are not able to achieve the optimal Steiner tree in most instances [6], they are also able to the quasi-Steiner tree with near-optimal cost in a relatively shorter time span. And it has the merits on the low complexity, easy application and is more reasonable in the multicast cost optimization.

A distributed delay-restricted multicast route heuristic method DMPH (delay-constrained minimal-cost path heuristic) is presented in this article. This algorithm proposed in the article can achieve a convergence speed, back up dynamic multicast, and offer the pretty good the network overhead performance.

2 The Mathematical Computing Model for the Delay-Restricted Multicasting Route Problem

Supposed a graph $G = (V, E)$, every edge $e \in E$ has its own two weight power functions: $C(e)$ and $D(e)$, inside $C(e)$ symbolize the positive real overhead of the link e, and $D(e)$ expresses the delay which is taken for transmitting message over the link e. For the graph G, supposed a source node $s \in V$ and a destination node set $D \subseteq V - \{s\}$. Then the delay-restricted Steiner tree T is a tree which is rooted in s and able to cover the all destination nodes. Once the condition which the delay constraint is met, the network overhead of the tree could be minimized as below:

$$\forall v \in D,\ \text{while} \sum_{e \in P(s,v)} D(e) \prec \Delta \text{ is met}, \sum_{e \in T} C(e) \text{ is minimized} \tag{1}$$

In above, Δ is a positive real number, representing the boundaries of delay constraints, and $P(s, v)$ being the path from the source node s to the destination node v in the multicast tree.

3 Description of the Algorithm Steps

Based on the MPH arithmetic, DMPH algorithm is proposed in this paper, and its main idea is about: In the networks, there is not going to be many paths to d_i to meet the delay requirement in common if the shortest delay time from the source end s to the destination end d_i is somewhat close to the upper limit of Δ. However, once the shortest delay time to d_i is relatively long to the upper limit of Δ, then there will be more d_i paths to meet the delay requirements. Consequently, when we build a delay-restricted multicast tree, if we firstly add the destination ends with relatively big minimum delay to the multicast tree, then the destination ends with smaller minimum delay are connected to the multicast tree via sharing some paths of the current tree without violating the delay restrictions. These cases could come true, and the multicast tree also is optimized with much less overhead. On the contrary, once we firstly choose the ends with the smaller delay to connect to the tree, then the ends which meet the shortest time delay may give up the method of connecting to the multicast tree by sharing some paths of the current tree because of the limitation of delay constraint. It even needs to establish a path to connect itself to the group separately. So to plant trees is not easy to optimize the multicast tree cost.

Before describing the DMPH method, a tree to destination data structure T2D (shown in Table 1) and a parameter D_v that we need to use in the algorithm are defined firstly.

The table T2D has $|D|$ rows, every row has five columns: d_i, T2D $[d_i]$.cost, T2D $[d_i]$.trenode, T2D $[d_i]$.order and T2D $[d_i]$.tag. They represent respectively: the destination end di, the minimum overhead path $P(d_i, T)$[1] from node d_i to the current tree T under the condition of satisfying delay, the node which $P(d_i, T)$ can access in the tree T, the sequence number of destination node d_i plan to connect to the tree T, the flag whether d_i has been added to the tree T (yes: means Di has been added to the multicast tree; no: means Di has not been added to the multicast tree).

Table 1. Data structure of T2D

d_i	T2D$[d_i]$.cost	T2D$[d_i]$.trenode	T2D$[d_i]$.order	T2D$[d_i]$.tag
d_1				
……				
$d_{\|D\|}$				

Parametric D_v represents the path delay from source node s to node v in a multicast tree, and is kept as an information for each node v in the tree. The algorithm can be described below:

Step 1. To initialize the table T2D. Firstly, the tag of every destination node is set to be NO, and the trenode of each access end being s. Computing the minimum delay of every destination end D and the minimum cost which can meet the delay requirement, arranging these destination ends in non-ascending order withe the shortest delay they reach, but the nodes with the same minimum delay are arranged in a non-descending order of the minimum cost. The value of T2D$[d_i]$.order is the location of the node Di in the sorted node set. The location also is the order in which it connects to the multicast tree, and supposing that the order of the sorted destination nodes is $d_1, d_2, \cdots, d_{|D|}$. T2D$[d_i]$.cost is the minimum overhead which can meet the delay requirement from the source node to the destination node. It should be noted that if the shortest delay from one node to another is bigger than the limit Δ, there is no multicast tree that meets the delay requirement, so the process should be stopped.

[1] $P(d_i, T)$ It can be calculated from a delay-limited unicast routing algorithm, which does not mean absolute optimization.

Step 2. To assume $T = (\{s\}, \varnothing)$ and $i = 1$. Establishing a shortest overhead path $P(d_i,s)$ which meets the delay requirement from d_i to s, Then adding $P(d_i, s)$ into T, and updating T2D[d_i].tag to yes. If $P(d_i, s)$ also pass through other destination ends, then T2D [d_i].tag of this destination node also is updated to yes. Updating T and T2D.

Step 3. To assume that the access node from d_i to the tree is u_i, for every node on the newly added path $P(\ u_i, d_i)$. From the access end u_i to the destination node d_i, the following operations are performed step by step (taking one of the nodes v as an example):

To compute the smallest overhead path $P^{'}(v, d_j)$ from the node v to every node d_j of the non-tree respectively under the condition of the delay limit $\Delta - D_v$, and if meeting the condition: cost[$P^{'}(v, d_j)$] $\prec$ T2D[d_j].cost, then

$$\begin{cases} \text{T2D}[d_j].\text{cost} = \text{cost}[P'(v, d_j)] \\ \text{T2D}[d_j].\text{trenode} = v \end{cases} \tag{2}$$

In the Eq. (2), cost[$P^{'}(v, d_j)$] being the overhead of the path $P^{'}(v, d_j)$, and the node v transfer the T2D to the next node of the tree. Therefore, at destination node d_i, all non-tree destination nodes find the minimum overhead path and access node that can satisfy the delay requirement from the current tree, and then T2D stays at d_i.

Step 4. Let $i = i + 1$**, and start from step 4 If the T2D [di].** To let tag is yes. Otherwise, d_{i-1} sends the message inform[2] to the node T2D [d_i].trenode. Once T2D [d_i].trenode receives the information, the smallest overhead path P(T2D [d_i].trenode, d_i) to d_i which meets the delay requirement is constructed. Then this path will be accessed into the tree T, updating the tree T, and T2D [d_i].tag to yes. If the path P(T2D [d_i].trenode, d_i) also pass through other destination nodes, the T2D [d_i].tag of the destination end is updated to yes.

Step 5. To keep the Operation. If $i \neq |D|$, then go to the step 3; the algorithm will not stop until $i = |D|$. T2D is established at the source node, with the addition of new paths to the tree, T2D will go through every new joined tree node (including destination nodes and non-destination nodes) one by one, which makes it easy that every new joined tree node can get the delay of the path from the source node to its tree.

From the algorithm above, the method DMPH is completely distributed, the source node only takes the responsibility for adding 3 into the multicast tree, and the rest of routes will be done via other nodes. This way allow that the two steps (route[3] 和connection configuration[4]) to establish the multicast connection could be done at the same time,

[2] Inform contains T2D and a message informing T2D [d_i]. tree node to establish a minimum overhead path to node d_i that satisfies the delay.

[3] Route: to look for a multicast routing tree initiated from a source node that covers all destination nodes;

[4] Connection Configuration: New connections are configured for each node on the tree, including reserving network resources and registering a new connections in a switching table.

it is much easier. However, in the centralized algorithm, when the source node (or a central node) calculates the routing, a connection configuration stage also is needed to be done separately. In addition, the centralized method is much more complex than the distributed algorithm. In the whole convergence time of this algorithm, DMPH intends to take at most $|D|$ times to connect the destination nodes from the tree ends to the multicast trees, up to $|D| - 1$ inform packets are sent to the tree node to notify the establishment of a path to connect a destination node to the tree, and a complete packet is needed to inform the source node the message that the tree construction has been completed. All in all, DMPH only take at most $|D|$ times to transfer the packets in the whole convergence time of this algorithm.

4 Addition and Quitting of the Nodes

Many multicast applications require the supports for dynamic multicasting, because the members of multicast group often change, the participating network nodes can join or leave the multicast group at any time, and the communication members are dynamic, with the addition and quitting of the multicast members, this type of the application needs to change the current multicast tree. So the multicast problem is the dynamic multicast routing, and DMPH addresses the dynamic multicast routing problem via using the methods below:

4.1 Addition of the Nodes

When a node d_q requests to be added into the current multicast group, if d_q is the Steiner node of the multicast tree (except the source node and destination node), no additional operation is needed to be performed. But if d_q is not on the multicast tree, the following operations below intends to be done:

The node d_q sends a request packet to the source node s for joining. When s receives the request information from d_q, it creates a require group with three items of information (new-node, cost and trenode), and sets new-node, cost and trenode to d_q, minimum overhead under delay constraint from s to d_q, and s respectively. s Sends require packets to each of its downstream nodes separately, the downstream node compares the minimum overhead of require packets to d_q under the requirement of time delay. If the overhead is less than the cost of require packets, cost is set to the overhead, and tree node is set to the downstream node. This downstream node transmits the require packets which are processed by it to every its downstream ends for performing the operations above. Therefore, when the require packet reaches at the all leaf nodes, each leaf end transmits the processed packet to the source node s. After the source node s receives the all require packets feed backed, the node which is the closest to d_q under the delay restriction will be found. Then the source end s sends the create packet to the node which is the closest to d_q to construct a minimum overhead path under the delay constriction, and add d_q to the tree.

4.2 Quitting of the Nodes

The treatment for the quitting of the nodes is so easy. If the quitting node is not the leaf node, there is no operation. If the quitting node is the leaf node, then this node and its upstream node will be deleted. The treatment for the quitting of the nodes is so easy. If the quitting node is not the leaf node, there is no operation. If the quitting node is the leaf node, then this node and its upstream node will be deleted until it meets a multicast node or a node with more than 2°.

5 Experimental Cases

The Fig. 1 shows a graph with 18 ends, the parameters of its each edge is (overhead, delay), the source node is1, and its destination nodes set is {7, 14, 16, 18}. Figure 2 and 3 are the calculating results under the delay constriction $\Delta = 9$ and $\Delta = 25$ respectively, the tree overhead is 86 and 65, and the maximum delay from the source node to the destination end is 8 and 12.

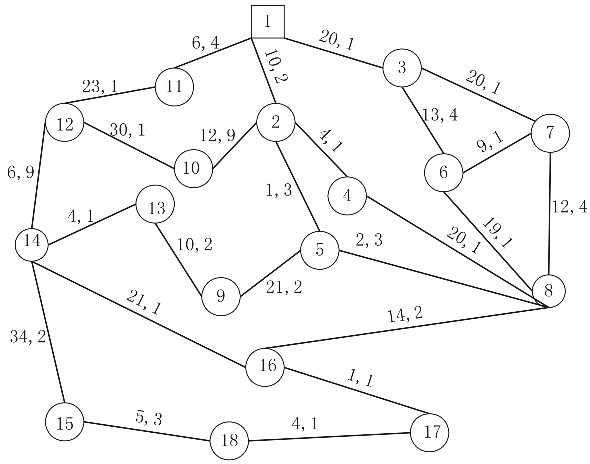

Fig. 1. A graph with 18 nodes

Now we could compare the algorithm to the minimum delay path tree method. When the algorithm starts, a shortest delay path from the source node to every destination nodes will be built, then these paths are combined, and combined result is exactly the final result.

The tree overheads are all 114 while the delay constriction $\Delta = 9$ and $\Delta = 25$. However, in the traditional method, the destination nodes which are the closest to the current tree could be linked to the tree, so that the overhead of the multicast tree equals 126 under $\Delta = 9$ and a circle is constructed. At the same time, the overhead of the multicast tree is 65 under $\Delta = 25$. Consequently, the overheads of the multicast tree which is computed via the DMPH algorithm proposed in this article are all not more than the traditional method.

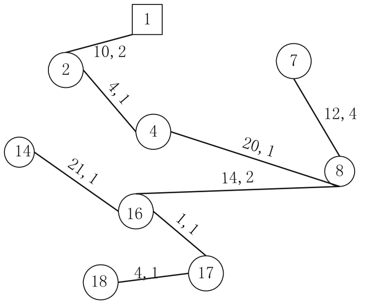

Fig. 2. Tree of $\Delta = 9$

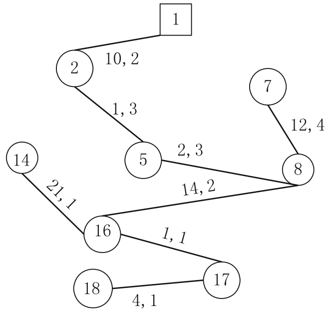

Fig. 3. Tree of $\Delta = 25$

6 Conclusion

This paper proposed delay-restricted multicast route heuristic method DMPH (delay-constrained minimal-cost path heuristic). The algorithm has the characteristics of distributed, fast convergence and dynamic multicast support, and achieves the good performance over the network overhead. In the article, we only talked about the algorithm under the delay constriction, and supposed the same delay-restricted limit for the all destination nodes. Virtually, there are different QoS limit, and each destination node has their own different QoS requirement. So a multicast routing algorithm with QoS limit intends to be more complicated. In addition, the article just addressed the operation of the addition and quit of the nodes on the dynamic varying of the multicast group, but not concerned the overhead optimal. If the optimal problem is involved in, the method will be more and more complicated. These problems mentioned above are all our future research work.

Acknowledgements. This research was financially supported via Project of the National Natural Science Foundation of China (61402069), the 2017 Project of the Natural Science Foundation of Liaoning province (20170540059), the General project of Liaoning education department in 2016 (2016J205).

References

1. Lee, S.-J., Su, W., Gerla, M.: On-demand multicast routing protocol in multi hop wireless mobile networks. Mob. Netw. Appl. **7**, 441–453 (2002)
2. Wang, J., Zhengpeng, Y., Gillbanks, J., Sanders, T.M., Zou, N.: A power control algorithm based on chicken game theory in multi-hop networks. Symmetry **11**(5), 718 (2019)
3. Sahasrabuddhe, L.H., Mukherjee, B.: Multicast routing algorithms and protocols: a tutorial. IEEE Netw. **1**(2), 90–102 (2000)
4. Lee, S.-J., Su, W., Gerla, M.: Ad hoc wireless multicast with mobility prediction.: Book title. In: Proceedings of IEEE ICCCN 1999, no. 7, pp. 4–9 (1999)
5. Wang, J., Zhang, S., Zhang, J.: Multi-hop maimal ratio combining (MHMRC) diversity based on virtual cellular network. Jilin Univ. **41**, 533–536 (2011)
6. Wang, J., Zou, N., Zhang, Y., Li, P.: Study on downlink performance of multiple access algorithm based on antenna diversity. ICIC Express Lett. **9**, 1221–1225 (2015)