Multi-objective Heuristic Multicast Routing Algorithm in NDN

Xuming An[1,2], Yu Zhang[1,2(✉)], Yanxiang Chen[1], and Yadong Wang[1]

[1] Beijing Institute of Technology, Beijing, China
{2120160725,yuzhang}@bit.edu.cn
[2] The Science and Technology on Information Transmission and Dissemination in Communication Networks Laboratory, CETC-54, Shijiazhuang, China

Abstract. NDN naturally supports multicast better than the traditional Internet, and multicast plays an important role in NDN. Most researchs on multicast routing algorithms are focused on cost optimization without taking node cache into account. This paper constructs a mathematical model for joint optimization of delay and cost, which is more flexible in describing NDN than adding delay as a constraint to the model. Then, the heuristic multicast algorithm considering node cache for this model is proposed. Last, we analyze the delay performance of the algorithm by comparing it with the exact Algorithm and the classical STMPH algorithm.

Keywords: Multicast algorithm · Delay · Joint optimization · NDN

1 Introduction

Named Data Networking (NDN) [8]is a new network architecture. A user sends out an Interest packet whose name identifies the desired data. Routers forward the Interest packet based on their names. A Data packet with the matching name is returned to the requesting consumer. Because of node caching and explicitly naming data, NDN can realize multicast easier than TCP/IP.

There are several routing protocols for NDN: such as NLSR [3], CRoS [5], and OSPFN [4], but as best we know, there is no multicast algorithm in these protocols. The multicast routing problem can be interpreted as solving steiner tree, and there are many algorithms: the exact algorithm, such as STEA [7], TEA [7], DPA [7], LRA [7], the computation time of these algorithm increase exponentially as the network size increases; the approximate algorithm, authors of [1] propose a family of algorithms that achieves an approximation ratio of $i(i-1)k^{1/i}$ in time $O(n^i k^{2i})$ for any fixed $i > 1$, where k is the amount of destinations; the heuristic algorithm, such as KMB [7], MPH [7], ADH [7], KPP [6], BSMA [6], the

Supported by Science and Technology on Communication Networks Laboratory Foundation Project and Aerospace Field Pre-research Foundation Project (060501).

B. Li et al. (Eds.): IoTaaS 2018, LNICST 271, pp. 211–214, 2019.
https://doi.org/10.1007/978-3-030-14657-3_20

approximate algorithm and heuristic algorithm belong to the polynomial-time algorithm, and the calculated result is close to the optimal solution. Although KPP and other algorithms add delay constraints, this approach cann't describe NDN flexibly due to that the services in NDN have different QoS requirements. Therefore, this paper constructs a model for joint optimization of delay and cost to describe NDN more flexibly. Moreover, because there is no delay performance analysis for these algorithms, we analyze the delay performance of the algorithm.

2 Mathematical Model

In this section, we construct a mathematical model of joint optimization delay and cost in network:

$$\min \quad \lambda_1 * \sum_{arc(i,j)} c[i,j] * y[i,j] + \lambda_2 * D$$
$$\Lambda_+ = \{\lambda | \lambda \geq 0, \lambda_1 + \lambda_2 = 1\}$$
$$D = \sum_{arc(i,j)} x[i,j] * d[i,j]$$
$$\text{s.t.} \quad \sum_i x[i,j] - \sum_i x[j,i] = \begin{cases} K[s,i], & \text{if } j \neq s; \\ -\sum_t K[s,t], & \text{if } j = s. \end{cases} \tag{1}$$
$$BigM * y[i,j] \geq x[i,j]$$
$$y[i,j] \leq x[i,j] \tag{2}$$
$$y[i,j] * demand \leq cap[i,j] \tag{3}$$
$$x[i,j] \in \mathbb{N}$$
$$y[i,j] \in \{0,1\} \tag{4}$$

In this model, each link (i,j) has three parameters, namely bandwidth $b[i,j]$, cost $c[i,j]$ and delay $d[i,j]$. Moreover, the binary variable $K[s,t]$ indicates if the source s has data transmission to node t(1/0).

Formula 1 guarantees the flow conservation of the source node, the destination node, and the relay node, where the variable $x[i,j]$ represents the number of destination nodes that the content on link (i,j) flow to. Formula 3 ensures that the total bandwidth utilized on each link does not exceed its available bandwidth, where the variable *demand* denotes the requested bandwidth of the content. The formula 2 ensures that binary variable $y[i,j]$ controls the inclusion of link (i,j) in the solution. The values of λ_1 and λ_2 can be set according to the service type.

3 Multicast Algorithm

The steiner tree problem is NP-complete [7], the exact algorithm does not belong to the polynomial-time algorithm, it is impractical to solve route in NDN with the exact algorithm. Futhermore, the approximate algorithm and heuristic algorithm don't consider node cache. Thus, when designing multicast algorithm, we should consider node caching in NDN. The MOH (Multi-objective heuristic multicast routing algorithm) algorithm we proposed as follow:

Given a directed graph $G = (V, A)$, the source node S, dest set $M \subset V$, the requested bandwidth of the content: bw. Moreover, we define the link (i, j)'s distance: $\lambda_1 * c[i, j] + \lambda_2 * d[i, j]$. The multicast routing problem is that of finding a directed Steiner tree with S as the root $T = (V_t, A_t)$:

Step1: Start from the source node S, $k = 1$, taking S as T_1, then $T_k = T_1$, $V_k = V_1 = \{S\}$, M_k is the set of the destinations. Calculate the shortest path from the spanning tree T_k to all destinations in M_k. If the relay node has less than bw cache space, delete the relay node and look for the suboptimal path; if the available bandwidth of the link is less than bw, the link is deleted; Record the path and distance of the spanning tree T_k to each destination in M_k.

Step2: The nearest (the distance from the spanning tree to destiantion is the shortest) destination in M_k is selected, then the destination and all relay nodes that are in the shoretest path from the spanning tree T_k to the destination are added to T_k, after that, the destination is deleted from M_k.

Step3: The following procedure is performed for the new nodes in (2): Calculate the shortest distance from the new nodes to the remaining destiantions in M_k. If this distance is less than the remaining destiantion's distance from the spanning tree T_k, then this distance is taken as the distance from the spanning tree T_k to the destination, meanwhile the shortest path from the spanning tree T_k to the destiantion is also recorded.

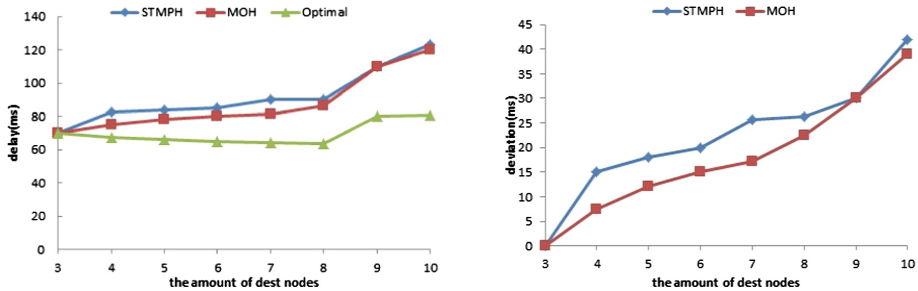
Step4: Repeat **step2** and **step3** until M_k is empty. When the set is empty, all the destiantions are in the spanning tree T_k. T_k is the multicast tree that we wanted.

4 Performance Analysis

In order to effectively evaluate the delay performance of multicast algorithm, we simulate three algorithms: the MOH algorithm we proposed, the exact algorithm, the classical STMPH algorithm [2], by using a network models. In this models, the network size is fixed at 100 nodes and the node's degree are 10. Figure 1(a) shows that the average delay vs the multicasting group size. Figure 1(b) show the deviation of the two heuristic algorithms from the optimal value in the above three scenarios.

From above figures, we can see that when the scale of network is fixed, with the increase of group size, the two deviation become more obvious. After analyzing the two algorithms, we believe that the reason for this phenomenon should be: each time a terminal added, it is always added to the tree through the shortest path in the two algorithms. It is possible to miss those paths that are not the shortest path or whose cost are also the shortest but search order is later.

These paths are likely to implement reduce the overall delay of the entire multicast group. On the other hand, the STMPH's deviation is greater than MOH's. This is due to the fact that when you add a terminal, STMPH connects the new multicast terminal to the terminal or the source node in the multicast tree. This way only considers the shortest path between the terminals that already in the tree and the remaining multicast terminals. Thus, STMPH's search depth is far less than MOH's, which results in missing more paths. Moreover, because of the way of STMPH adding new terminal, it is likely to lead multiple multicast terminals to be in a chain structure on one path. This will result in an increase of overall transmission delay. Therefore, our proposed algorithm's delay performance better than the classcial STMPH algorithm.



(a) delays vs group size (b) deviations vs group size

Fig. 1. Delay performance when the node's degree is 10

References

1. Charikar, M.: Approximation algorithms for directed Steiner problems, pp. 192–200 (1998)
2. Li, H., Yu, J., Xie, W.: Locally searching minimum path cost heuristic. Acta Electron. Sinica **28**(5), 92–95 (2000)
3. Hoque, A.K.M., Amin, S.O., Alyyan, A., Zhang, B., Zhang, L., Wang, L.: NLSR: named-data link state routing protocol. In: ACM SIGCOMM Workshop on Information-Centric NETWORKING, pp. 15–20 (2013)
4. Wang, L., Hoque, M., Yi, C., Alyyan, A., Zhang, B.: OSPFN: an OSPF based routing protocol for named data networking (2012)
5. Torres, J.V., Ferraz, L.H.G., Duarte, O.C.M.B.: Controller-based routing scheme for named data network (2013)
6. Wang, B., Hou, J.C.: Multicast routing and its QoS extension: problems, algorithms, and protocols. IEEE Netw. **14**(1), 22–36 (2000)
7. Winter, P.: Steiner problem in networks. In: Conference on Computer Networks (1987)
8. Zhang, L., et al.: Named data networking (NDN) project, vol. 1892, no. 1, pp. 227–234 (2014)