A Network Coding Optimization Algorithm for Reducing Encoding Nodes

Limin Meng and Yangtianxiu Hu(✉)

College of Information Engineering, Zhejiang University of Technology, Hangzhou 310023, China
2111603007@zjut.edu.cn

Abstract. Network coding can effectively improve the transmission efficiency of the network, but compared with the traditional forwarding nodes, the participation of network encoding nodes will bring resource consumption. In this paper, we propose an improved algorithm of the max-flow based on the shortest path, which combines the concept of path capacity summation to achieve the maximum flow of the network, and the shortest path guarantees the minimal number of encoding nodes. The simulation results based on random network show that this algorithm can effectively reduce the encoding nodes and the consumption of network resources on the basis of realizing the maximum flow of the network.

Keywords: Network coding · Encoding nodes · The shortest path Max-flow · Capacity summation

1 Introduction

The traditional network intermediate node has long played a role of routing forwarding until 2000 Ahlswede et al. [1] proposed the concept of network coding to break the limitations of the understanding of intermediate nodes. Intermediate nodes not only play the role of storage and forwarding, but also can encoded the information, so that the multicast network can achieve the maximum transmission traffic thus increasing the transmission rate.

But the encoding nodes are brought to the resource consumption, affecting the transmission efficiency. So, in case of maximum throughput (maximum flow), the fewer encoding are, the less resource consumption is. In the traditional network coding algorithm, encoding nodes are the intermediate nodes whose number of input links is larger than the number of output links in the work, however, in the guarantee of the maximum flow is not all such intermediate nodes need to be encoded, so there is a lot of unnecessary resource consumption. Therefore, how to ensure that maximum flow on the basis of the minimal coding node become a hot issue.

In graph theory, there are many algorithms for calculating the maximum flow of the network, which can be applied to the network coding to achieve maximum throughput. In recent years, more and more scholars have studied this aspect in more depth. Liu et al. [2] combined Dijkstra routing [3] and generic linear network coding [4] to reduce the complexity of network coding algorithm, and explained the possibility of applying

L. Meng and Y. Zhang (Eds.): MLICOM 2018, LNICST 251, pp. 265–272, 2018.
https://doi.org/10.1007/978-3-030-00557-3_27

the knowledge of graph theory to network coding. Tao et al. [5] proposed a minimal cost network coding algorithm based on critical link, and the Dinic algorithm [6] in graph theory is applied to search of the augmented path. In [7], the Floyd algorithm [8] is applied to the path cluster search process of the polynomial time algorithm [9]. And Zhu et al. [10] proposed an algorithm to reduce the number of coding nodes on the basis of Ford-Fulkerson algorithm [11]—the maximum flow algorithm of graph theory with the new understanding of reused links and super critical nodes.

In this paper, we proposed a shortest path maximum flow network coding algorithm based on the capacity summation. When the max-flow is achieved, the number of encoding nodes can be minimized because the shortest path makes the minimal passing intermediate nodes and the selection principle of coding nodes is improved.

The rest of this paper is organized as follows. We will review some related knowledge in Sect. 2 and give our algorithm steps in Sect. 3. In Sect. 4, we will present an example to compare this improved algorithm with the Ford-Fulkerson algorithm. In Sect. 5, simulation results will be given and we can know the advantage of the improved algorithm in reducing encoding nodes. At last, Sect. 6 serves as conclusions.

2 Related Works

In a network $N = (V, L, C)$ [12], V represents the set of nodes, including the source node set S, the sink node set T, and the intermediate node set W; L represents the set of links between nodes and C represents the capacity set. For link $l \in L, f(l)$ is called the flow on the link l. If $l = (v_i, v_j)$, $f(l)$ can also be recorded as f_{ij}. And for $v \in V$, $f^+(v) = \sum_{v_i = v} f_{ij}$ is the output flow of node $V, f^-(v) = \sum_{v_j = v} f_{ij}$ is the input flow of node V.

2.1 The Shortest Path

The shortest path in the graph theory is the shortest path between two mutually distinct nodes,where each link has a corresponding length. In this paper, in order to simplify the calculation, the length of each link will be united, so the shortest path from source to sink is to find a path with the minimal links in all possible cases, and thus the number of passing node is the minimal.

2.2 Capacity Summation

For a network, the links between each node have the corresponding capacity. If there is no link between two nodes, then there is no capacity and the capacity value is 0. The sum of all links' capacity on a path is added to obtain capacity summation.

2.3 Feasible Flow

If flow f satisfies:

$$0 \leq f(l) \leq C(l), \forall l \in L \tag{1}$$

$$f^{+}(v) = f^{-}(v), \forall v \in V\{s,t\} \tag{2}$$

such a f is called a feasible flow and f always exists.

2.4 Encoding Node

The encoding node will only appear in the intermediate nodes, and the encoding node of previous algorithm judges that such a node may be an encoding node when the number of input links is greater than the number of output links– $num^{-}(v) > num^{+}(v)$. But in this paper, in addition to this encoding node conditions, we also consider that transmission information on the input link is different from that on the output link; and for each $v \in V, \sum_{i=1}^{n} f_i > \sum_{i=1}^{n} c_i$, $\sum_{i=1}^{n} f_i$ represents the total flow of all input links, and $\sum_{i=1}^{n} c_i$ represents the total capacity of output links.

3 Algorithm Process

We propose an improved algorithm based on some of the above concepts to achieve the maximum flow of the network, and the following are the implementation steps of the algorithm:

(1) Generate a more regular network structure (single source network) by making some changes to the random network topology [13], where the source node is denoted as S, the sink node is denoted as $T_n(n = 1, 2, 3\ldots)$, and the intermediate node is denoted as $V_i(i = 1, 2, 3\ldots)$.
(2) Initialization: Suppose that the capacity of each link in the network is $C_{ij} > 0$ (if there is no link between two nodes, then $C_{ij} = 0$), and the value of the feasible flow f_{ij} is initialized to be 0.
(3) Augmented process: The source node S is sent to the adjacent node information, and then find the shortest path for augmentation from the source node S to the sink nodes T_n.If there are multiple paths of the same length, then find the path of the maximum capacity summation for augmentation (If all the capacity summation is the same, then choose one to augment arbitrarily). If there is such an augmenting path, then go to (4), otherwise re-enter (3) until finding such an augmenting path.
(4) Adjustment Process: The augmenting flow $\varepsilon = min_{\left(v_i, v_j \in V\right)}\left\{C_{ij}\right\}$. If the link is the forward link, then the residual flow value of the link minus ε. If the link is the backward link, then the residual flow value of the link plus ε. If a link has reached saturation after the flow adjustment, the link is not considered in the following augmenting path search.
(5) After the end of the flow adjustment, and then re-enter (3) to search for the next augmenting path.

(6) The above process is iterated until all the augmenting paths in the network are found to reach the maximum flow, and the information of the input edge and the output edge of each intermediate nodes is recorded, and then the encoding node is judged according to the definition of the encoding node.

The complexity of the algorithm is analyzed as follows:

Assume that the number of vertices is m and the number of links is k in the network graph. When searching for an augmenting path, since the number of vertices of the network is m, the number of links included in the augmented path found by the algorithm is at most m. So from the source node to the sink node, it takes at most m steps. And because the number of links is k, searching for an augmentation path takes at most k times, so the complexity of finding an augmenting path is $O(k * m)$. Then in the process of modifying the flow for the augmenting link that is sought, the traffic on each edge needs to be modified and this complexity is $O(m)$. So the complexity of this algorithm is $O(k * m^2)$.

4 Example Analysis

Suppose there is such a network, the source node is denoted as S and the sink node is denoted as T, and the intermediate node is denoted as $1, 2, 3, 4, 5, 6$. The connection between nodes represents a link, and the number on the link represents capacity. The network topology is shown in Fig. 1.

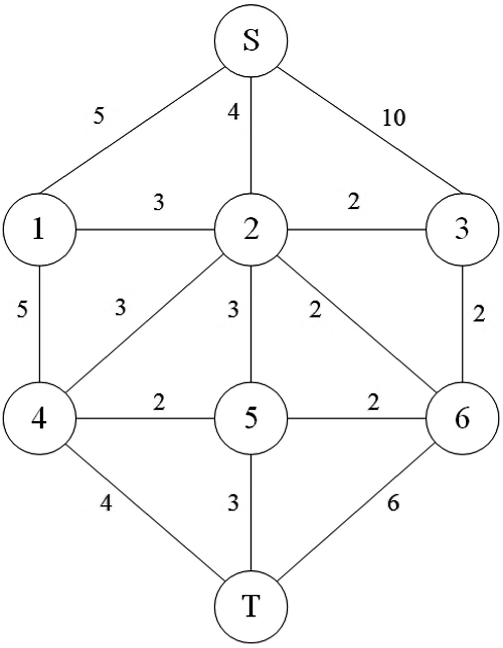

Fig. 1. Network topology

The implemental of the shortest path maximum flow algorithm based on the capacity summation in the network is as follows:

Find the shortest path from source node to the sink node, respectively $S \to 1 \to 4 \to T$, $S \to 2 \to 4 \to T$, $S \to 2 \to 5 \to T$, $S \to 2 \to 6 \to T$, $S \to 3 \to 6 \to T$, where the path of the maximum capacity summation is $S \to 3 \to 6 \to T$, so this path is augmenting path and the augmenting flow $\varepsilon_1 = 2$. For the saturated link, will not be considered in the next augmenting path search. So, we can get all of the following augmentation processes, as shown in the Table 1.

Table 1. Augmengting process of the improved algorithm

Serial number	Augmenting path	Augmenting flow
2	$S \to 1 \to 4 \to T$	$\varepsilon_2 = 4$
3	$S \to 2 \to 6 \to T$	$\varepsilon_3 = 2$
4	$S \to 2 \to 5 \to T$	$\varepsilon_4 = 2$
5	$S \to 1 \to 2 \to 5 \to T$	$\varepsilon_5 = 1$
6	$S \to 3 \to 2 \to 4 \to 5 \to 6 \to T$	$\varepsilon_6 = 2$

Finally, all the maximum flow paths from the source node to the sink node are obtained. The maximum flow $f_{max} = \varepsilon_1 + \varepsilon_2 + \varepsilon_3 + \varepsilon_4 + \varepsilon_5 + \varepsilon_6 = 2 + 4 + 2 + 2 + 1 + 2 = 13$.

So, for a multi sink multicast network, we can extend to find out all the maximum flow paths from the source node S to the sink nodes $T_n(n = 1, 2, 3\ldots)$. And the encoding nodes in all the maximum flow paths can be found according to the definition of the encoding node.

If we do not use the improved shortest path maximum flow algorithm but use the traditional maximum flow algorithm—Ford-Fulkerson. Ford-Fulkerson algorithm is an important max-flow algorithm in the graph theory and achieves the max-flow of the network through the residual network. And then we can know the augmenting process of this network is shown in Table 2.

Table 2. Augmengting process of Ford-Fulkerson algorithm

Serial number	Augmenting path	Augmenting flow
1	$S \to 1 \to 2 \to 4 \to 5 \to 6 \to T$	$\sigma_1 = 1$
2	$S \to 1 \to 4 \to 5 \to 6 \to T$	$\sigma_2 = 1$
3	$S \to 1 \to 4 \to T$	$\sigma_3 = 3$
4	$S \to 2 \to 4 \to T$	$\sigma_4 = 1$
5	$S \to 2 \to 6 \to 5 \to T$	$\sigma_5 = 2$
6	$S \to 2 \to 5 \to T$	$\sigma_6 = 1$
7	$S \to 3 \to 2 \to 5 \to 6 \to T$	$\sigma_7 = 2$
8	$S \to 3 \to 6 \to T$	$\sigma_8 = 1$

The max-flow $f_{max} = \sigma_1 + \sigma_2 + \sigma_3 + \sigma_4 + \sigma_5 + \sigma_6 + \sigma_7 + \sigma_8 = 1 + 1 + 3 + 1 + 2 + 1 + 2 + 2 = 13$.

Although the Ford-Fulkerson algorithm can achieve the maximum flow, the number of augmenting path is larger than the number of augmenting path of the shortest path maximum flow algorithm based on the capacity summation. And with the increase of nodes, the advantages of improved algorithm are more obvious. This is because the Ford-Fulkerson algorithm is not targeted to find nodes when looking for augmenting path and needs to continue to contiguous nodes labeled, so there is redundancy and blindness. The improved algorithm is based on the principle of the shortest path—passing the least nodes—and the maximum capacity summation to select the augmenting path, so that redundancy can be reduced, and because the number of passing intermediate nodes is minimal, then the corresponding number of intermediate nodes that can become the encoding node will be reduced.

5 Simulation Results

In this paper, the adopted simulation tool is MATLAB 2015a, the first set of experiments is to verify the relationship between the number of augmenting paths and the number of nodes of the improved algorithm and the Ford-Fulkerson algorithm in the same case, as shown in Fig. 2. We can see that in the case of fewer nodes, the path exists in the network is limited, so the difference between the two algorithms is not

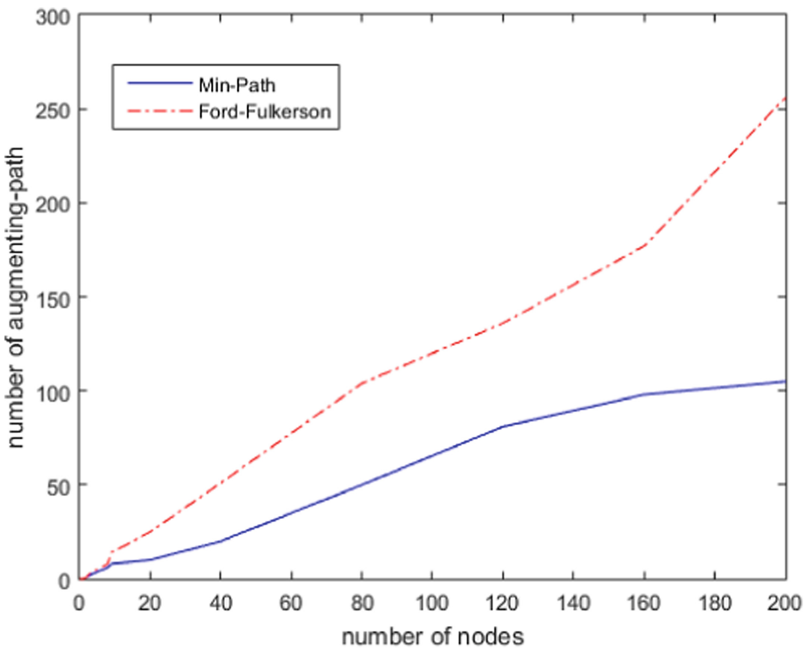

Fig. 2. The relation between the number of nodes and the number of augmenting paths

significant. However, as the number of nodes increases, the number of augmenting path of two algorithms is increased. But the curve of the improved algorithm is slower than that of the Ford-Fulkerson algorithm, which indicates that the larger the network size is, the more available paths are. The improved algorithm reduces redundancy so that maximum flow can be achieved with fewer augmenting paths.

In the second experiment, the relationship between the number of encoding nodes and the number of nodes is simulated, as shown in Fig. 3. Though the re-definition of encoding nodes, it can be seen that under the same conditions (the same network topology and the same capacity value on the link), the number of encoding nodes of the improved algorithm is smaller than the traditional network encoding nodes. And with the increase of the network size (the number of network nodes), this advantage is more prominent. Firstly, this is because that the shortest path means the minimal number of passing nodes, and then the definition of encoding nodes is stricter so that some unnecessary encoding nodes can be reduced.

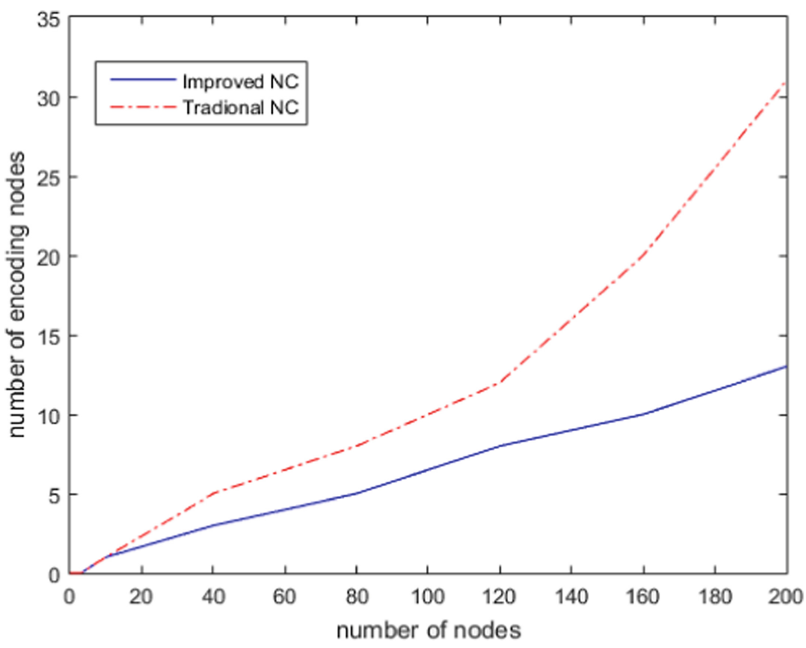

Fig. 3. The relationship between the number of nodes and the number of coded nodes

6 Conclusion

This paper aims to solve the problem of reducing the number of unnecessary encoding nodes in the network coding process, so we propose the shortest path maximum flow algorithm based on the capacity summation. Not only can achieve the maximum network flow makes the maximum throughput, but also reduces the amount of computation for encoding nodes of the network. Compared with the traditional algorithm, it has more advantages in reducing the encoding nodes, reducing the overhead of nodes, and making network coding more practical.

Acknowledgement. Thanks for the support of Zhejiang Provincial Key Laboratory of Communication Network Applications, National Natural Science Foundation of China (61372087) and Natural Science Foundation of Zhejiang Province (LY18F010024).

References

1. Ahlswede, R., Cai, N., Li, S.Y.R., et al.: Network information flow. IEEE Trans. Inf. Theory **46**, 1204–1216 (2000)
2. Hu, J.X., Liu, S.Y.: Improved multicast network coding algorithm. Comput. Eng. Appl. **47** (15), 116–118 (2011)
3. Dijkstra, E.W.: A note on two problems in connection with graphs. Numer. Math. **1**(1), 269–271 (1959)
4. Yeung, R.W., Li, S.Y.R., Cai, N., et.al.: Network Coding Theory. Now Publishers Inc. (2006)
5. Tao, S.G., Huang, J.Q., Yang, Z.K., et al.: An improved algorithm for minimal cost network coding. Huazhong Univ. Sci. Tech. (Natural Sci. Edit.) **36**(5), 1–4 (2008)
6. Yefim, D.: Algorithm for solution of a problem of maximum flow in a network with power estimation. Dokl. Akad. Nauk SSSR **11**, 1277–1280 (1970)
7. Liu, J.F., Zhou, J.: Minimize coding node algorithm based on polynomial time algorithms. In: The 23rd National Conference on New Computer Science and Technology and Computer Education (2012)
8. Floyd, R.W.: Algorithm 97: shortest path. Commun. ACM **5**(6), 345 (1962)
9. Jaggi, S., Sanders, P., Chou, P.A., et al.: Polynomial time algorithms for multicast network code construction. IEEE Trans. Inf. Theory **51**(6), 1973–1982 (2005)
10. Zhu, Y.Y., Cao, Z., Zhu, L.X.: An improved encoding nodes reduction algorithm for network. Electron. Electro—opt. Syst. **3**, 47–52 (2012)
11. Ford, L.R., Fulkerson, D.R.: Maximal flow through a network. Can. J. Math. **8**, 399–404 (1956)
12. Wang, H.Y., Huang, Q., Li, C.T., et al.: Graph algorithm and its MATLAB implementation. Beijing University of Aeronautics and Astronautics Press, Beijing (2010)
13. Waxman, B.M.: Routing of multipoint connections. IEEE J. Sel. Areas Commun. **6**(9), 1617–1622 (1988)