

A Network Coding architecture base on OpenFlow network

Jian Di & Jingtao Dong

School of control and Computer Engineering, North China Electric Power of University, BaoDing, Hebei Province, China

ABSTRACT: By allowing the exchange node to operate data, network coding can effectively improve the network throughput, ease network congestion and improve the network performance. Due to the closure of switching device in the current network structure and the cost of coding and decoding, the network coding can not be widely promoted and applied in real network environment. The OpenFlow network structure is programmable and the controller can know the global network topology, it make the possibility that apply network coding in the OpenFlow network structure. Thus, this paper try to join the network coding in OpenFlow network, and design SDNC, a framework to enable networking coding in OpenFlow networks, and this paper propose a optimized routing algorithm CBBSP, to improve the network performance by optimizing the network topology. Finally, we set up the SDNC system upon Mininet platform. And do a lot performance evaluation to verify our scheme can effectively improve the network performance.

KEYWORD: OpenFlow; Network Coding; Routing Algorithm; Mininet

1 INTRODUCTION

Since Ahlswede et al first proposed network coding (NC) in 2000(Ahlswede, R), it has attracted wide attention from the international academic community. Network coding by allowing the network switching node to the data operation, can effectively improve the network throughput, improving load balance, reduce transmission delay, save the energy of the sensor nodes and enhance the network robustness and can be widely used in Ad_Hoc networks, sensor networks, P2P content distribution, distributed file storage and network security, etc. However, due to the closure of the current network equipment, it is difficult to expand the coding function, and unable cope with real-time changes in the topology and other issues, resulting in network coding is difficult to be applied in the actual network (J. Krigslund, 2015).

In recent years, the concept of software defined network (SDN), which is considered as a major change in the network world, has been widely concerned by the academic community. And openflow protocol as the basis of SDN, have separation of data plane and control plane, programming, and the controller is known global topological structure characteristics, make the possibility that apply network coding in sdn.

In (Felicián Németh, 20121), Felicián Németh et al confirmatory in OpenFlow switch added encoding

and decoding function, first proposed the concept of combining sdn with network coding.(J. Hansen, 2015) apply the technology of combining sdn with network coding to the 5G network of the future, and prove that this will increase the transmission efficiency of many times. (D. Zhu, 2015) in Wireless Mesh Networks (WMNs) combine sdn and flow within the network coding technology, propose a new OpenCoding protocol, which can simplify the deployment and management of the network and improve the performance of the network.(Liu Jie, 2015) designed a new encoding control method on FiWi network architecture based on SDN, namely in the controller to realize the source code package filtering and coding control, improve the coding efficiency and network performance in the FiWi network. (Jun Yang, 2015) propose a NC-OF, a framework to enable networking coding in SDN networks, as well as in NC-OF apply MMF-NC coding strategy to improve the performance of the network, and put forward a view when the link of broadband enough don't need for network coding to reduce the consumption. (J. Wang, 2014) In the information-centric network (CCN) proposed a cache management framework based on sdn, and propose an efficient buffer management algorithm based on network coding.

In this paper, based on the research results of the author of the above literature, propose a SDNC , a

framework to enable networking coding in OpenFlow networks, use floodlight controller to achieve, and proposed a optimized routing algorithm that the Dijkstra algorithm combine with the compact box burning algorithm (CBB), which is called the compact box combustion shortest path algorithm (CBBSP), to improve performance and coding efficiency of the network.

2 DESIGN AND IMPLEMENTATION OF THE NCOS FRAMEWORK

According to partition functions of openflow network structure, the responsibility of controller is control the whole network, decision the forwarding behavior of openflow switch, OpenFlow switch just according to the strategy that is allotted from controller and perform simple data forwarding. In view of this, this paper design the SDNC architecture to carry on the division of work style of the original OpenFlow network structure. That is, for the newly added network coding function, decision how to apply network coding by the floodlight controller, and OpenFlow switch just according to the strategy that development of the controller to perform the packet encode/decode actions.

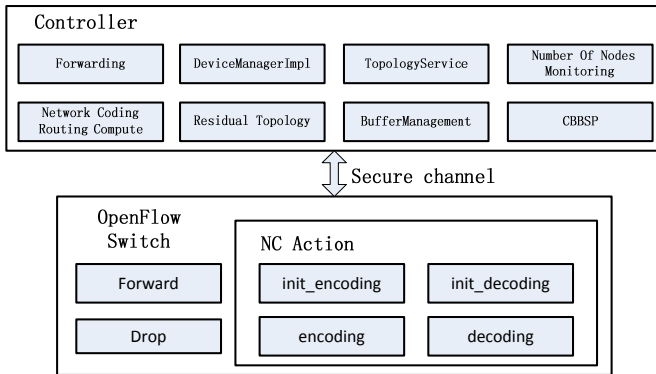


Figure 1. The SDNC framework.

In order to realize the function of network coding, the network coding and routing module is designed on the controller floodlight, which the routing algorithm is optimized, and design five new module: Number Of Nodes Monitoring module, the Residual Topology module, CBBSP module, Buffer Management module, Network Coding Routing Compute module. And apply to the original Topology in the floodlight controller: Forwarding module, DeviceManagerImpl module, TopologyService module. In addition, we use the XOR encoding method for network encoding. Because the XOR method is simple and does not require any coding coefficients, it can effectively reduce the length of the coding data header, and can reduce the time delay of coding. In order to make the OpenFlow switch has the ability to encode and decode, we add four actions in the

switch: init_encoding, init_decoding, encoding, decoding, and expand of the OpenFlow protocol correspondingly. the specific architecture shown in Figure 1.

2.1 CBBSP MODULE

Because of the SDN can obtain a global network topology, which is a fundamental difference between SDN network and traditional network. When the global topology can be obtained, we can completely according to the topology information, to optimize the routing lookup. In the traditional graph theory, and in most of the controller, find the shortest path algorithm based on the topology is Dijkstra algorithm, the time complexity of this algorithm is $O(V+E\log E)$. That is, the performance of the algorithm is related to the number of nodes and edges, so long as the number of V and E is reduced, it is clear that the performance of the algorithm can be improved.

The network fractal thought of the complex network just fit our demand for reducing the size of the network nodes and edges, so we use complex network box cover algorithm thought, propose a SDN routing algorithm based on the box cover algorithm, optimize the traditional Dijkstra algorithm from topological structure.

In general, first, the whole network is divided into several subnets by using the box covering algorithm, each subnet that box cover algorithm in the box. Then take the different subnet network as a node, between subnet and subnet using the Dijkstra algorithm, find a shortest path based on subnets, after that find the shortest path inside each subnet on the path. Finally, combine those paths together, is what we request routing. Among the many boxes covering algorithm, we choose Compact Box Burning algorithm (CBB) as dividing Box algorithm, the algorithm steps is simple, easy to implement, and requires less computation time and can effectively reduce the time delay of the whole process of encoding. The Compact Box Burning Shortest Path (CBBSP) algorithm steps are as follows:

- (i) Construct the set C of all yet uncovered nodes, according to the topology G .
- (ii) Choose a random node p from the candidate set C and remove it from C .
- (iii) Remove from C all nodes i whose distance from p is $l_{pi} \geq l_B$, since by definition they will not belong in the same box.
- (iv) Repeat steps (ii) and (iii) until the candidate set is empty.
- (v) The collection of nodes were chosen constitute a compact box. Repeat the above steps, until all nodes are constituted a compact box, that is, all points are covered, forming a new optimized topology G .

- (vi) The box which contain the source node and destination node as G source node and destination node, and use Dijkstra algorithm to find the shortest path R .
- (vii) Perform Dijkstra algorithm separately in the each box which is contained by path R , find out each box of shortest path.
- (viii) Stitching These shortest path together is what we are looking for the shortest path in the topology graph G .

This paper according to the CBBSP algorithm implements a Demo, and run in the different size of the network, and compare with the traditional Dijkstra algorithm. Because most of the computer network is a scale-free network, this article generate a random scale-free network, then on each node in the network calculation to all other nodes route, respectively using the Dijkstra algorithm and CBBSP algorithm for the shortest path, and compare the results, the results are shown in Figure 2. It can be seen that, when the number of nodes is more, the computation time of CBBSP algorithm is less, compare to the Dijkstra algorithm has superior performance.

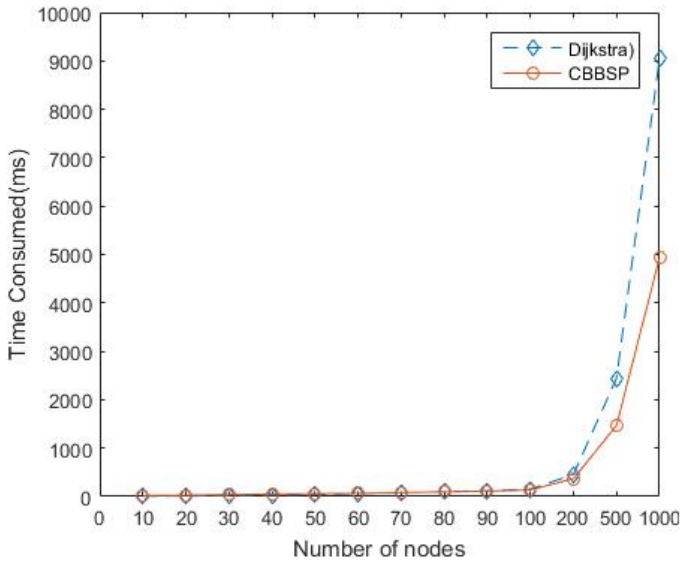


Figure 1. CBBSP and Dijkstra the time consumed.

Table 1. Algorithm result comparison.

The number of nodes	Dijkstra	CBBSP
10	0.0060572	0.02060
50	0.0384808	0.05744
70	0.08048	0.08054
100	0.152447	0.13640
200	0.45474	0.36858
500	2.442844	1.46568
1000	9.046405	4.93387

And it can be seen from Table 1, when the number of nodes is less, the computation time of CBBSP algorithm is greater than the Dijkstra algorithm, but with the increase of the number of nodes, the gap is gradually reduce. When the number of nodes is 70,

the computation time of two algorithms is almost equal. After that, with the number of nodes increase of the continuously, the advantages of CBBSP algorithm becomes more and more obvious. Therefore, in order to further improve the performance of SDNC framework, we put the number of nodes for 70 as a watershed, when the node number is less than 70, SDNC use the Dijkstra algorithm, when the number of nodes is greater than 70, SDNC use CBBSP algorithm to improve the performance. So we added a Number Of Nodes Monitoring module, a decision mechanism, according to the number of nodes in the network to decide which algorithm to compute routes

2.2 Floodlight default module

TopologyService, Forwarding, and DeviceManager-Impl modules are the default module in the controller floodlight. The main function of TopologyService module is to maintain the topology information for the floodlight controller, and to find the routes in the network. The main function of forwarding module is to transmit data packets between the two devices, the routing algorithm use Dijkstra algorithm. The DeviceManagerImpl module learns about devices through PacketIn requests. It takes information from the PacketIn and classifies the device according to how the entity classifier is setup.

2.3 Residual Topology

The current can be used the network topology is stored in Residual topology module, that is, the global topology in TopologyService subtract the links which already had occupying. Floodlight compute routes for unknown flow will call Residual Topology module, and search for routing path in the remaining topology. When forwarding module have calculated the routing path for the unknown flow, it will update the remaining topology, namely modify the network topology in the Residual Topology module

2.4 Buffer Management

The Buffer Management module is used to manage the buffer in the OpenFlow switch, which is used to buffer the receiving data packets for encoding or decoding. In order to meet the different requirements of the network, we add the QoS to SDNC. We use the MPLS label strategy to configure QoS service, by adding a priority number in the packet header, so that the coding queue can be further divided and the position of packet in the queue can be adjusted according to this priority such as figure 3.

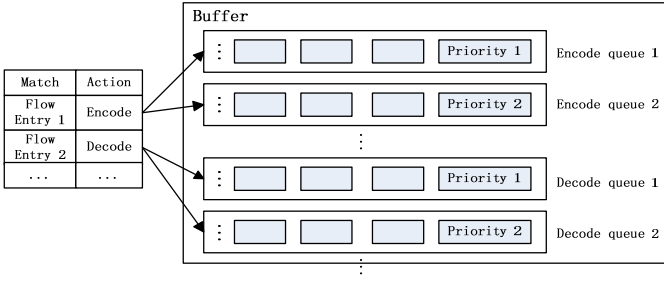


Figure 3. Apply MPLS label to QOS in buffer

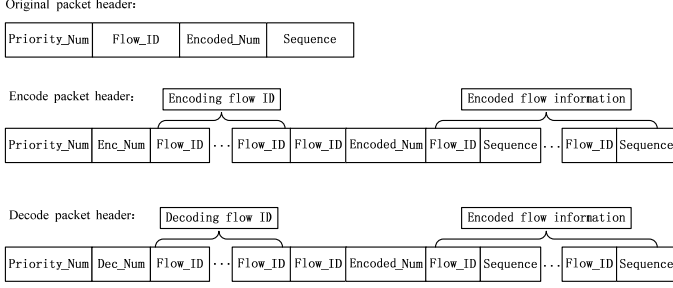


Figure 4. NC encode/decode packet header in OpenFlow

2.5 Network Coding Routing Compute

This module is used to calculate the code routing for the unknown flow. When the Forwarding module can not calculate the routing in the residual topology, the Network Coding Routing module is called to calculate the coding routing. Network Coding Routing module according to the judgment of the Number Of Nodes Monitoring module, to decide which algorithm to calculate the routing paths in the global network topology, Dijkstra algorithm or CBBSP algorithm. If we can get the code routing path, indicating that this unknown flow has a common path with the flow is transmitting. After that, if the cure path can be obtained by the Dijkstra algorithm in the topology that the remaining topology subtract the routing path of just has calculated, the routing policy generate a flow table entry, configure the OpenFlow switch, and update the Residual Topology module.

2.6 Network Coding Packet Header

As shown in Figure 4, we use MPLS to transmit corresponding coding information within the coding packets. We need add three new field for the original data packages (i.e., three MPLS label): the serial number of the flow (Flow_ID), the number of flows that already have encoded (Encoded_Num), the serial number of packets on the flow (Sequence). Encode packet also is added the multiple fields: the number of flows that will be encoded (Enc_Num), the participating encoded flow ID (Flow_ID), serial number of this encode flow (Flow_ID), the number of packets of already have encoded (Encoded_Num), etc. The decode packet header is similar to the encoded packet header.

In summary, the OpenFlow switch with encode and decode function support encode with both original flows, also support encode with the flow of already encoded and the original flow; In the process of decode, it support partial decode, also support fully decode. In short, the OpenFlow switch is proposed in this paper have complete encode/decode function.

3 IMPLEMENTATION AND EXPERIMENTS

We use Mininet as the experimental platform of the SDNC system, the controller use the Floodlight. We didn't use large network topology because we want to focus on the process of network coding, try to ignore the influence of the intermediate node forwards. We use Mininet custom topology function build test topology as shown in Figure 5, two flows (the size of packet in the flow is the same), in the network , respectively are flow 1 from client h1 to h4, flow 2 from client h2 to h3. Then setting the switch port rate is 0.96Mbit/s, set message rate in flow 1 and flow 2 is per second sent 50 packets, total send five minutes. When the packet size of the flow 1 and flow 2 are changed from 100B to 1800B with the growth rate of 100b. The bandwidth of OpenFlow switch port will change as the send packet size. We convert the data packet size to the bandwidth of the occupancy (transmission load) to compare conveniently, the results of the two flow transmission delay is shown in Figure 6.

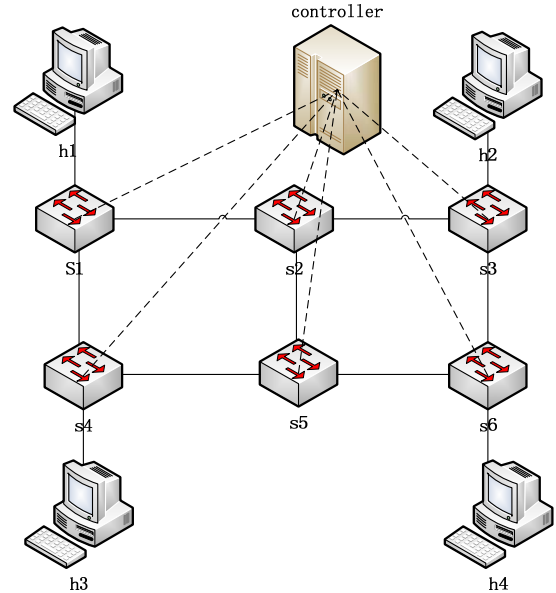


Figure 5. Test topology

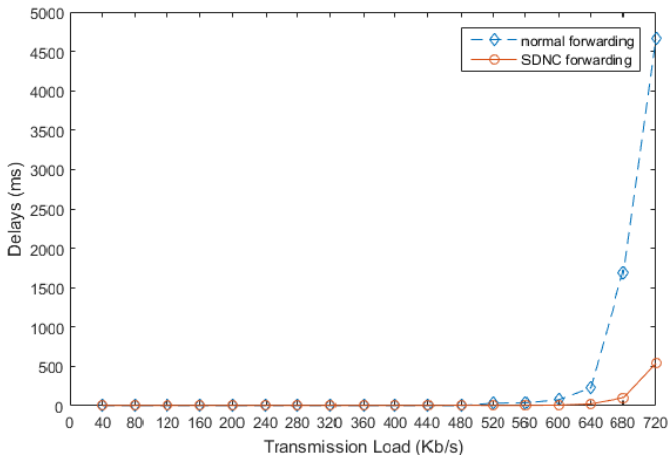


Figure 6. The flows transmission delay change with the transmission load

It can be seen from the experimental results, that the average transmission delay of SDNC forwarding transmission delay is higher than the normal forwarding transmission delay when the link bandwidth is sufficient. This is because the code strategy need encode/decode packet and increase transmission delay. However, with the lack of bandwidth, the normal forwarding will be congested, and the transmission delay will increase dramatically, while SDNC forwarding transmission delay increases slowly. Although network coding can increase link capacity and improve network performance, but also increase the transmission delay and increase the computation amount. It has both advantages and disadvantages.

4 CONCLUSION

This article focus on implementing network coding function in OpenFlow network architecture, and we propose a SDNC, which is a complete network coding network architecture systematically. We introduce the details of SDNC with each module and the coding process, and put forward a CBBSP algorithm which a shortest path routing algorithm based on the box cover algorithm, to improve the network performance of the SDNC under the condition of too many nodes. Finally, we implemented SDNC framework on Mininet to evaluate its performance. It can be seen from the experimental results that the network coding can effectively improve the network throughput and bring better network performance to the OpenFlow network structure. But also because of the encode/decode packet operation lead to increase transmission delay, and when the network topology changes caused by transmission delay is increased and other issues, we will try to solve these problems in the future.

REFERENCES

- Ahlsweide, R. & Cai, Ning & Li, S. Y. R. & Yeung, R. W. 2000. Network information flow. *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1204-1216.
- D. Zhu, X. Yang, P. Zhao and W. Yu, "Towards Effective Intra-Flow Network Coding in Software Defined Wireless Mesh Networks," 2015 24th International Conference on Computer Communication and Networks (ICCCN), Las Vegas, NV, 2015, pp. 1-8.
- Felicián Németh, Ádám Stipkovits, Balázs Sonkoly and András Gulyás, "Towards SmartFlow: Case Studies on Enhanced Programmable Forwarding in OpenFlow Switches", *ACM SIGCOMM'12*, 2012.
- J. Krigslund, J. Hansen, D. E. Lucani, F. H. P. Fitzek and M. Medard, "Network Coded Software Defined Networking: Design and Implementation," *European Wireless 2015; 21th European Wireless Conference; Proceedings of*, Budapest, Hungary, 2015, pp. 1-6.
- J. Hansen, D. E. Lucani, J. Krigslund, M. Medard and F. H. P. Fitzek, "Network coded software defined networking: enabling 5G transmission and storage networks," in *IEEE Communications Magazine*, vol. 53, no. 9, pp. 100-107, September 2015.
- Jun Yang, Bin Dai, Lu Lv, and Guan Xu, "Coding Openflow Enable Network Coding in SDN Networks," *International journal of Computer Networks & Communications* 09/2015; 7(5):29-38.
- J. Wang, J. Ren, K. Lu, J. Wang, S. Liu and C. Westphal, "An optimal Cache management framework for information-centric networks with network coding," *Networking Conference, 2014 IFIP, Trondheim, 2014*, pp. 1-9.
- Liu Jie, Guo Feng, Li Wenzhu, Ren Danping, Liu Xin, "Network coding control scheme in software-defined fiber wireless access networks", *Study On Optical Communications*, 2015, 04:7-9.