IOP Conference Series: Materials Science and Engineering

PAPER • OPEN ACCESS

Intelligent Routing Scheme for SDN Satellite Network Based on Neural Network

To cite this article: Weichao Sun *et al* 2019 *IOP Conf. Ser.: Mater. Sci. Eng.* **563** 052087

IOP Conf. Series: Materials Science and Engineering **563** (2019) 052087 doi:10.1088/1757-899X/563/5/052087

Intelligent Routing Scheme for SDN Satellite Network Based on Neural Network

Weichao Sun*, Jun Liang, Nan Xiao, Ran Ding and Zhenhao Zhang

Information and Navigation college of Air Force Engineering University, Xi'an, China

E-mail: 1037008913@qq.com

Abstract. In view of the existing SDN-based satellite network system, the tri-state content-addressable memory (TCAM) space occupied by the flow table is increasing, and the search and matching of flow entries are complicated, resulting in reduced routing and forwarding efficiency, which cannot meet the needs of diverse application requirements. Therefore, an intelligent routing scheme of SDN satellite network based on neural network (IRSSN) is proposed. The controller acquires the transmission mode of the data stream by training the neural network (NN), and replaces the flow table with the trained NN. An intelligent routing mechanism based on the orthogonal polynomial (Chebyshev) neural network is proposed. The switch predicts its forwarding path according to the traffic type of the data stream to meet the QoS requirements of the satellite network application. The simulation results show that IRSSN significantly reduces the occupied TCAM space and improves routing efficiency.

1. Introduction

Many scholars have made some explorations on the introduction of SDN in satellite networks [1]. However, most literatures all use the OpenFlow protocol as the SDN southbound programming interface [2]. With the advancement of the OpenFlow version [3], the expanding "multi-level flow table" and complex flow table item lookup and matching problems bring great challenges to the storage and processing capabilities of the switch.

Zhang C [4] firstly proposes a method of replacing flow tables with radial basis function (RBF) neural network in SDN network. However, it is difficult to perform parallel operations on the satellite [5]. It also needs complex inverse matrix operations, which makes it difficult to meet the requirements of on-board routing [6]. Chebyshev neural network function based on orthogonal polynomials has excellent approximation performance, and the orthogonality of the basis function ensures that the weight value of NN at the nodes of polynomials can be solved quickly [7].

Based on the above analysis, an intelligent routing scheme (IRSSN) for SDN satellite network based on Chebyshev neural network is proposed. The transmission mode of data stream is acquired by training neural network to predict routing. At the same time, NN is used instead of flow table to save TCAM storage space and improve routing and forwarding efficiency of data stream.

2. System Architecture

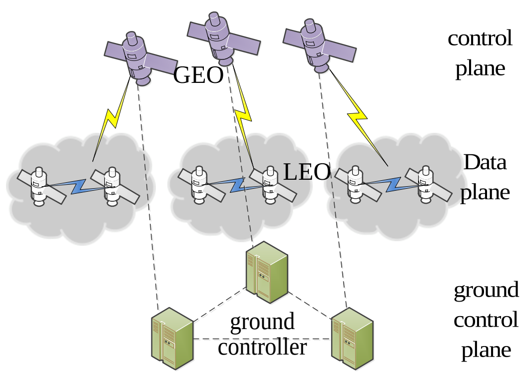

Figure 1. system architecture.

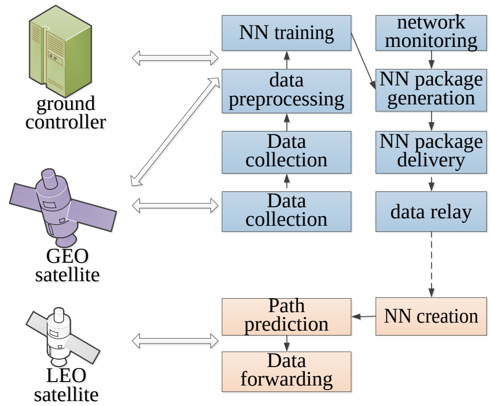

Figure 2. functions of planes.

We use the single-layer satellite multi-layer controller architecture (SSMC), the SSMC is divided into three layers: GEO satellite control plane, LEO data plane plane and ground control plane. The new function modules in each plane are shown in Figure 2.

2.1. Control Plane

The control plane is composed of GEO satellite control plane and ground control plane. Its main function is to collect and process a large amount of network flow information and acquire data flow transmission mode by training NN with its powerful computing ability.

The core function modules of the ground controller are described as follows:

- Network monitoring module. Responsible for real-time monitoring of LEO satellite network global state and topological changes.
- Data collection module. Responsible for collecting routing request messages from GEO satellite, extracting source, destination, traffic type and path information of data stream from them for training NN.
- Data preprocessing module. Responsible for preprocessing the collected data and constructing the sample set for NN training and testing. It includes preprocessing of data stream path, construction of training set and test set, normalization of sample set, etc.
- NN training module. Responsible for training the NN to obtain the transmission mode of the data stream.
- NN package generation module. Responsible for creating NN packages using NN parameters. See section 1.4 for the format.
- The NN package delivery module. Responsible for sending NN packets to GEO satellite controllers managing source and destination nodes, and to all relevant LEO satellites.

In addition to the above functional modules, the GEO satellite controller also has a data relay module as a relay node for the ground controller and LEO satellite. In addition, the network monitoring and data collection modules of GEO satellite controller are all oriented to LEO satellite local network within its own coverage area.

2.2. Date Plane

It is composed of LEO satellite switch, which is equivalent to SDN switch. Its core function is to forecast the path of data stream by using the NN sent by the controller, and route and forward the data stream. Its core function modules are described as follows.

- NN creation module. After receiving the NN packet, LEO satellite switch creates the NN according to the structure parameters stored in the NN packet.

- Path prediction module. After receiving the business data sent by users, three tuple information (source, destination, traffic type) is extracted from the data as input of NN for path prediction.
- Data forwarding module. The data packet is forwarded according to the path predicted by the path prediction module.

2.3. NN Preprocessing

Before training NN, it needs to be preprocessed, including the normalization of sample set and the construction of training set and test set.

When a data flow passes through the network, the controller extracts the source node (Src), the destination node (Dst), and the service type ($Type$) from the data flow for constructing the sample set, and records the transmission path ($Path$) of the data flow. When performing NN preprocessing, the sample $(Src, Dst, Type)$ is used as the input of the NN, and the Route is used as the output. In this paper, 30% of the samples are randomly selected as the training set, and the remaining 70% are used as the test set[8].

After the sample set is normalized, the controller trains the NN with the training set until the predicted success rate reaches the preset criteria and the NN training ends. The controller then creates a NN packet based on the trained NN and sends the NN packet to the associated LEO satellite switch.

3. Chebyshev Neural Network Intelligent Routing Mechanism

3.1. Chebyshev Neural Network Model

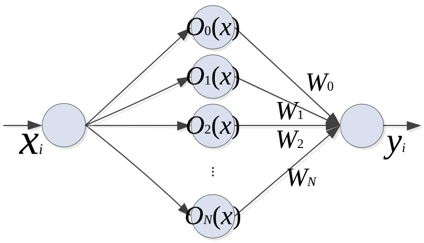

Figure 3. Monadic Chebyshev Neural Network Model.

Chebyshev NN consists of an input layer, a hidden layer and an output layer. Its unary model is shown in Figure 3. Among them, $O_i(x)$ represents the orthogonal polynomial of degree i. Its system constitutes the single hidden layer structure of NN, and W_i is the weight of $O_i(x)$. The goal of NN training is to use the known orthogonal polynomial system to approximate the unknown linearly through the training set. That is to say, $T_N(x)$ makes:

$$f(x_i) = T_N(x_i) = \sum_{j=0}^{N} W_{ij} O_j(x_i) \tag{1}$$

In this paper, $(Src, Dst, Type)$ is used as the input of Chebyshev NN, and the path of data stream (Route) is obtained as the output of NN. Therefore, consider the three-dimensional Chebyshev NN model, the input is a three-dimensional vector $[x_1, x_2, x_3]^T$, and the output is an accurate approximation of the three-dimensional function $y_i = f(x_1, x_2, x_3)$. it has the following forms:

$$f(x_1, x_2, x_3) = O_{N_1 N_2 N_3}(x_1, x_2, x_3) = \sum_{j_1=0}^{N_1} \sum_{j_2=0}^{N_2} \sum_{j_3=0}^{N_3} (W_{j_1 j_2 j_3} \prod_{j=1}^{3} O_{jk_j}(x_j)) \tag{2}$$

Among them, f is a function to be simulated, O is used to approximate the k_j-th polynomial of x_j, and $W_{j_1 j_2 j_3}$ is the weight of the combination of the orthogonal polynomials.

In addition, the output of the neural network is one or a group of scalar values, and the output of the NN in this paper should be expressed as an optimal path, so a mapping method is needed to form a one-to-one mapping between the scalar value and the optimal path.

3.2. Routing Algorithms

When the new data stream arrives, the switch first checks whether the NN has been constructed. If the NN has been built, the switch extracts $(Src, Dst, Type)$ from the incoming data stream and uses them as the input of the NN to predict the path and route it.

If the NN is not constructed, the data flow related information is encapsulated in the routing request message and sent to the controller. When the controller receives the request, it sends the recently trained N to all switches. After receiving the NN packet, the switch constructs the NN according to the structure parameters obtained from the NN packet, and then uses the extracted flow information as the NN input to predict the path and route the data flow.

In the training process of controller for NN, the weight value can be directly determined by using the orthogonality of Chebyshev polynomial [8].

In the process of path prediction, the traffic type of data stream is regarded as an input vector of NN. According to the user's requirements in bandwidth, delay, delay jitter and bit error rate, network resources can be allocated effectively, so that the path can better meet the quality of service requirements of different users.

4. Simulation and performance evaluation

4.1. simulation settings

Based on the lightweight test platform Mininet developed by Stanford University, the improved ONOS controller was selected as the experimental controller. In order to avoid interference between Mininet and ONOS, the two are deployed on two physical devices in the form of virtual machines. Choose 6 experimental machines with the same configuration. The machine configuration is Intel Core i5 3.3GHz, 4GB RAM, 2Gbp network card, based on Ubuntul14.04 LTS system. The experimental results were analyzed by means of Matlab tools. Using the simulation scenario similar to reference [9], it consists of 3 GEO satellite controllers, 48 LEO satellite switches, and 3 ground controllers.

4.2. simulation results analysis

The performance of IRSSN is evaluated comprehensively by the following indicators.

- Storage space occupied by NN packages (SNP). In the proposed scheme, flow tables are replaced by NN packages, and the storage space occupied by flow tables is compared with that occupied by NN packages.
- Data Packet Transfer Rate (PDR). It refers to the proportion of packets that successfully arrive at the destination node.
- Packet Loss Rate (PLR). It refers to the proportion of lost data packets to all data packets.
- Packet Delay (PDL). Refers to end-to-end packet delay.

Each simulation experiment was repeated 10 times and its average value was taken as a measure.

Experiment 1: Storage space

In OpenFlow-based SDN networks, data streams are routed and forwarded according to flow tables. Flow tables are stored in TCAM. However, in the proposed routing scheme, data streams are routed and forwarded according to the NN, and do not need to be stored in TCAM. To facilitate storage space comparison, it is assumed that each switch in SDN network contains 1000 stream table entries. In OpenFlow V1.1.0, the maximum length of the matching field of the stream table entries is 356 bits. Therefore, the TCAM storage space occupied by streammeter entries is about 444.57 KB. As shown in Figure 4, Chebyshev NN takes up much less TCAM storage space than 444.57 KB, even when considering storing NNs in TCAM.Compared with NNIRSS using RBFNN, the storage space is further reduced due to the reduced space complexity.

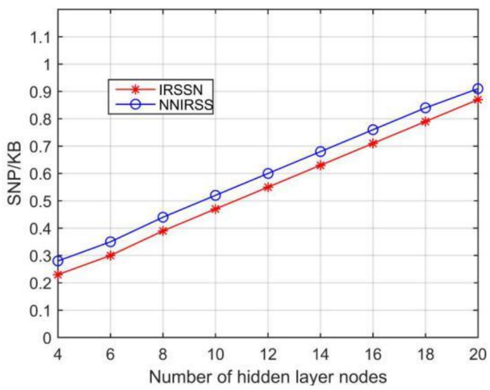

Figure 4. storage space occupied by IRSSN.

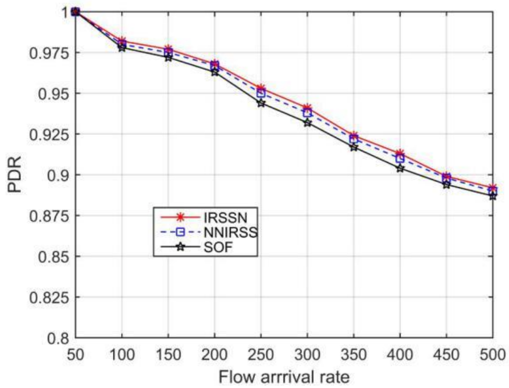

Figure 5. Comparison of PDR.

Therefore, it can be concluded that the proposed IRSSN can significantly reduce the occupied TCAM storage space.

Experiment 2: PDR

Figure 5 shows the comparison of PDR between IRSSN, NNIRSS and SOF. As the rate of arrival of traffic increases, the total number of packets in the network increases, causing network congestion. Due to the limited processing power of the switch, some packets in the stream are dropped and cannot be successfully routed and forwarded to the destination. We can observe that regardless of IRSSN, NNIRSS or SOF, the corresponding PDR becomes smaller as the rate of arrival of traffic increases.

However, when the traffic arrival rate becomes larger, the PRS of the IRSSN is higher than NNIRSS and SOF because the SOF uses the shortest path strategy for the data stream. In SOF, packets in a stream are routed along the shortest path and forwarded, causing packets from the same data stream to be routed along the same path. In IRSSN and NNIRS, data packets in a flow are routed and forwarded according to their respective service types, that is, different types of data packets are routed and forwarded along different paths in IRSSN and NNIRS, and compared with IRSSN of Chebyshev NN. NNIRSS solves faster and processes faster.

Therefore, the total number of packets transmitted to the destination is greater than NNIRSS and SOF.

Experiment 3: PLR

Figure 6 shows the PLR comparison results between IRSSN, NNIRSS and SOF. As the rate of arrival of the traffic increases, the PLR of the three becomes higher, and the PLR of the IRSSN is smaller than the PLR of the NNIRSS and the SOF. The reason is similar to the above PDR. As the rate of arrival of traffic increases, a large number of packets are discarded in the SOF due to network congestion caused by too many packets routed on the same path, and the IRSSN using Chebyshev NN is faster and faster than NNIRSS. Can reduce congestion. Therefore, the total number of dropped packets in the IRSSN is smaller than the dropped packets in the NNIRSS and SOF.

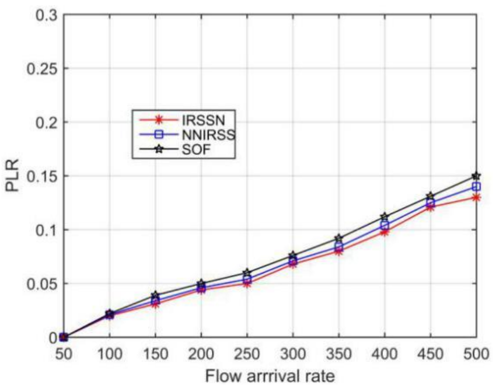

Figure 6. Comparison of PLR.

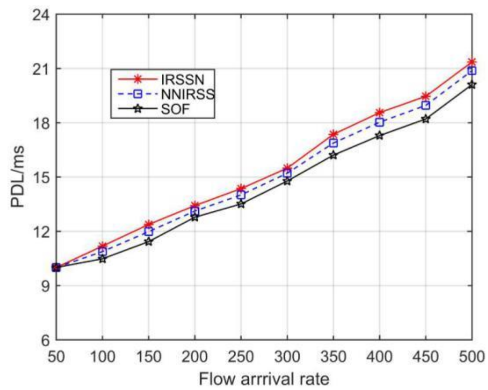

Figure 7. Comparison of PDL.

Experiment 4: PDL

Figure 7 shows the PDL comparison results of IRSSN, NNIRSS and SOF at different traffic arrival rates. Regardless of the IRSSN, NNIRSS, or SOF, as the rate of arrival of traffic increases, the total number of packets in the stream becomes larger. Excessive packets can cause network congestion. The more severe the network congestion, the longer the packet will be transmitted to the destination node.

In addition, we can observe that the length of PDL of SOF and NNIRSS is longer than that of IRSSN. Although the shortest path algorithm is adopted in SOF, as the traffic arrival rate increases, network congestion becomes more serious, and the PDL of SOF becomes longer. On the other hand, in IRSSN and NNIRSS, the routing and forwarding process of data packets does not involve complex flow table lookup and matching processes. Compared with SOF, NN path prediction takes less time, and Chebyshev NN's IRSSN is compared to NNIRSS. The solution is faster and faster.

In summary, IRSSN is superior to SOF and NNIRSS in TCAM storage space, PDR, PLR and PDL performance.

5. Conclusion

In order to solve the problem of insufficient storage space of TCAM caused by stream table expansion in OpenFlow protocol, an intelligent routing scheme for SDN satellite network based on NN is proposed, which uses NN instead of stream table to reduce storage space and routing time overhead of TCAM and improve routing efficiency. NN packages do not need to be stored in TCAM, which greatly reduces the storage space and corresponding hardware costs of TCAM, eliminates the complex flow table generation process, and no longer needs to find and match flow table entries. Secondly, an intelligent routing mechanism based on Chebyshev NN (IRSSN) is proposed. LEO satellites can predict the trained Chebyshev NN path according to the traffic type of data stream, which meets the application's QoS requirements. Finally, the performance of the proposed IRSSN is evaluated by simulation experiments. The simulation results show that the proposed routing mechanism can significantly reduce the occupied TCAM space and improve the routing forwarding efficiency.

References

[1] Xu S, Wang X W and Huang M 2018 “Software-Defined Next-Generation Satellite Networks: Architecture, Challenges, and Solutions,” *IEEE Access*, vol. 6, no. 1, pp. 4027-4041.

[2] Diego K, Ramos M V, Fernondo M V and Paulo E V 2015 “Software-Defined Networking: A Comprehensive Survey,” *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14-76.

[3] Azzouni A, Braham O and Trang N T M 2016 “Fingerprinting OpenFlow Controllers: The First Step to Attack an SDN Control Plane,” in *Proc. IEEE Int. Conf. Commun*., pp. 841-843.

[4] Zhang C, Wang X and Li F 2018 “NNIRSS: neural network-based intelligent routing scheme for SDN,” *Neural Computing and Applications*, vol. 1, no. 3, pp. 1-17.

[5] Karakus M, Durresi A 2017 “Quality of Service (QoS) in Software Defined Networking (SDN): A survey,” *Journal of Network and Computer Applications*, vol. 80, no. 12, pp.200-218.

[6] Folkes R, Lahav O and Maddox S J 2018 “An Artificial Neural Network Approach to the Classification of Galaxy Spectra,” *Monthly Notices of the Royal Astronomical Society*, vol. 283, no. 2, pp. 651-665.

[7] Liu H Y, Sun F C 2013 “Satellite network QoS routing algorithm based on orthogonal polynomials neural network,” *J Tsinghua Univ (Sci & Tech)*, vol. 53, no. 4, pp.556-561.

[8] Montijano E, Montijano J I and Sagues C 2013 “Chebyshev Polynomials in Distributed Consensus Applications,” *IEEE Transactions on Signal Processing*, vol. 61, no. 3, pp. 693-706.

[9] Qi X G, Ma J L and Liu L F 2018 “Routing optimization based on topology control in satellite network,” *Journal on Communications*, vol. 39, no. 2, pp. 11-20.

[10] Du P, Nazari S and Mena J 2016 “Multipath TCP in SDN-enabled LEO Satellite Networks,” in *Proc. IEEE Int. Conf. Commun*., pp. 354-359.