Cognitive Virtual Network Topology Reconfiguration Method Based on Traffic Prediction and Link Importance

YUZE SU1, XIANGRU MENG1, ZHENHUA YU2, AND QIAOYAN KANG1

1College of Information and Navigation, Air Force Engineering University, Xi’an 710077, China
2Institute of Systems Security and Control, College of Computer Science and Technology, Xi’an University of Science and Technology, Xi’an 710054, China

Corresponding author: Zhenhua Yu (zhenhua_yu@163.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61873277, Grant 71571190, and Grant 61871313, and in part by the Key Research and Development Program of Shaanxi Province under Grant 2019GY-056.

ABSTRACT With the increase of network services, how to avoid link congestion and make full use of limited bandwidth resources in network virtualization environment have become the key challenges. In this paper, we introduce the cognitive method into virtual network topology reconfiguration and modify the connectivity to reach the target topology by sensing the traffic. Then, we formulate an optimization problem to maximize the ratio of cumulative saved resources to the square of changed link number (CSR/SCLNR) and determine which virtual link to be added or deleted. Finally, a heuristic algorithm called cognitive virtual network topology reconfiguration method based on traffic prediction and link importance (CVNTRM-TPLI) is proposed to solve this optimization problem. In the CVNTRM-TPLI, the link importance is presented in link deletion as the topological factor to avoid the ping pong effect. Also, a hybrid traffic prediction algorithm based on optimal parameter selection is put forward, where immune optimization is introduced to select the optimal parameters and the virtual network topology can react promptly to the traffic fluctuation without adding or deleting virtual link frequently. Simulation results show that the CVNTRM-TPLI not only has the highest CSR/SCLNR, but also solves the link congestion and makes full use of the limited bandwidth resources.

INDEX TERMS Virtual network topology, reconfiguration, cognitive, immune optimization, link importance, traffic prediction.

I. INTRODUCTION

As an important technology in next generation Internet, the network virtualization allows multiple heterogeneous virtual networks (VNs) to share substrate network (SN) and offer flexible manageability [1]–[3]. The characteristics of network virtualization facilitate the introduction of new types of network services (e.g., visual telephone and high-definition video distribution) that require a large amount of traffic and cause traffic fluctuation frequently [4]. The static virtual network topology (VNT) needs to spend more resources to adapt changes and restrict the development of network virtualization [5], [6]. The dynamic VNT can adapt to changes in the direction of the traffic, which has become a vibrant research area.

To support the dynamic VNT, a VNT reconfiguration is proposed, serving as a typical type of VN reconfiguration. The VN reconfiguration can be divided into two policies [7]. In the first policy, the VN is only reconfigured in exceptional cases, such as failures. It is usually permanently established and widely used in survivable VN against node failure [8]–[10], link failure [11] and hybrid failure [12], [13]. However, the bandwidth used reaches the maximum traffic value in a virtual link.

The alternative policy is also called VNT reconfiguration that reconfigures the VNT according to the current network status and traffic demand. This case is used to solve the link congestion and improve the link utilization [14]–[23]. To perceive the current status and make full use of the network resources, the cognitive method has been used in network. A cognitive network is defined as “A network with a process that can perceive current network conditions, and then plan,
decide, and act on those conditions. The network can learn from these adaptations and use them to make future decisions, all while taking into account end-to-end goals."[24]. Hence, the cognitive method is also introduced into VNT reconfiguration in this paper. In cognitive VNT reconfiguration, the traffic data can be collected to support the reconfiguration process and determine whether the current virtual topology should be kept or reconfigured. The virtual link can be added to solve the link congestion and deleted to release the bandwidth resources of an underutilized virtual link to embed more VNs or add more virtual links. The traffic is changed dynamically and this VNT reconfiguration is active to adapt it. While a virtual link may be deleted to handle the underutilized status in current traffic situation, but it may be added in the next traffic situation. This case is denoted as ping pong effect [15].

To avoid the ping pong effect, all new added virtual links cannot be deleted in next several periods [16]. Although this method can weaken the ping pong effect, many resources are occupied by all new added virtual links that may cause the waste of resources. Another way to solve the ping pong effect is to introduce traffic prediction methods and select their prediction results as inputs of VNT reconfiguration [17]–[19]. However, most traffic prediction methods are large scale. Also, traffic prediction requires much time and it cannot adapt the fluctuations on small scale traffic in a prediction period.

In this paper, a cognitive virtual network topology reconfiguration method based on traffic prediction and link importance (CVNTRM-TPLI) is proposed to avoid the ping pong effect and make full use of the limited bandwidth resources dynamically. At first, we introduce the cognitive method into VNT reconfiguration and formulate the VNT reconfiguration as an optimization problem. Then, a heuristic algorithm is proposed to solve this optimization problem. In CVNTRM-TPLI, the link importance and hybrid traffic prediction algorithm based on optimal parameter selection (HTPA-OPS) are presented in link deletion. The HTPA-OPS is a small scale traffic prediction method that can predict the fluctuations on small scale traffic within a prediction period. In HTPA-OPS, the optimal combination parameter selection algorithm based on immune optimization (OCPSA-IO) is introduced to select the optimal parameters and the VNT can react promptly to the traffic fluctuation without adding or deleting virtual link frequently.

The main contributions of this paper can be summarized as follows.

(i) We introduce the cognitive method into VNT reconfiguration. In cognitive VNT reconfiguration, the current traffic is monitored and obtained by the network controller. The link addition and deletion are triggered according to the change of current traffic. It is formulated into an optimization problem with some constraints and we propose a heuristic algorithm called CVNTRM-TPLI to solve it.

(ii) We regard the link importance as topological factor in link deletion. The virtual link with high link importance can become the busy path by routing and it cannot be deleted frequently to avoid the ping pong effect.

(iii) We propose the HTPA-OPS to predict the future traffic that can improve the adaptability to the traffic fluctuation. As a small scale traffic prediction method, the HTPA-OPS is only triggered in link deletion when the resources of virtual link are underutilized. In the HTPA-OPS, the local projection and phase space reconstruction are used to denoise the traffic and restore its inherent chaos characteristic. Then the RBF neural network is used to predict the traffic.

(iv) We propose the OCPSA-IO to select the optimal combination parameters with the help of immune optimization method in the HTPA-OPS. The OCPSA-IO is not only used to select the initial parameters, but also adjust the parameters when the prediction error is higher than the threshold to ensure the accuracy of the prediction results.

The rest of this paper is organized as follows. In Section II, we discuss the related work. In Section III, we present the problem statement and optimization model. The CVNTRM-TPLI is presented and its details are shown in Section IV. In Section V, we evaluate the proposed algorithm through extensive simulations and experiments. We conclude this paper in Section VI.

II. RELATED WORK

In network virtualization, the VN reconfiguration is a typical method to improve the survivability of VN. The VN reconfiguration can be triggered by exceptional cases, such as failures. In this case, the VN reconfiguration is complex that includes the virtual node migration and virtual link re-embedding. Also, the VN reconfiguration can be triggered by traffic cases, such as link congestion and link underutilization. This case can be called VNT reconfiguration that adapts the VNT by link addition and deletion.

In common, the current traffic status is as the input of the VNT reconfiguration. In [15], a VNT adaptation for wavelength division multiplexing (WDM) mesh networks under dynamic traffic was proposed. It adapted the optical links according to the actual traffic load continuously and reacted promptly to the fluctuations on the traffic by adding or deleting lightpath. However, the ping pong effect was controlled by adjusting the watermark values in this paper and the performance was limited. In [16], a VNT reconfiguration for mixed-line-rate optical WDM networks under dynamic traffic was proposed to follow the changes in traffic without a priori knowledge of the future traffic pattern. It could optimize resource utilization and network traffic performance by adjusting, adding or deleting one or more lightpaths. In link deletion, all new added virtual links could not be deleted in next time to avoid the ping pong effect. Consequently, more resources were wasted if the bandwidth utilization was low in next time. In [20], a gradually reconfiguring VNT based on estimated traffic matrices was proposed. It reconfigured the VNT gradually by dividing it into multiple stages and limiting the number of optical layer paths reconfigured in each stage to reduce the estimation errors. This algorithm adapted the
current bandwidth traffic status. However, it could not predict the future bandwidth utilization and solve the ping pang effect. In [21], a VNT reconfiguration with adaptability to traffic changes was proposed. A new index called flow inclusive relation modularity was introduced to reduce the number of optical paths, which had to be added when there were significant traffic changes. This method was proposed to avoid adding a large number of optical paths, it could not solve the ping pang effect and adapt the traffic in next time. In [22], a noise-induced method was proposed to adapt to traffic changes and accommodate traffic demand. It repeatedly reconfigured a VN that led to over-reconfiguration and network services disruption. In [23], a VN reconfiguration framework based on the Bayesian attractor model was proposed that used certain patterns of incoming and outgoing traffic at edge routers to characterize the traffic situation. The over-reconfiguration could be reduced by identifying the stored traffic situation that was closest to the current one and retrieving a suitable VN. However, this method could not solve the traffic situations that were not in the obtained VN candidate set well and the ping pang effect was not avoided.

The VNT reconfiguration is very relevant to current traffic and a VNT may become inappropriate to it after a certain time. Hence, the VNT reconfiguration may frequently adapt to the changing traffic and the ping pang effect may occur. To avoid the ping pang effect, a traffic prediction solution called autoregressive integrated moving averages technique was introduced, and a new transition method was also proposed to reduce the impact of unstable routing tables during a reconfiguration process [17]. The ARIMA technique was a typical large scale traffic prediction method and could not predict the small scale traffic. In [18], a VNT reconfiguration approach based on data analytics for traffic prediction was proposed. The artificial neural network was used to provide robust and adaptive traffic models. The VNT was regularly reconfigured based on the current and predicted traffic. Although the performance of artificial neural network was better than ARIMA technique, it could not predict the small scale traffic well. In [19], big data analytics was applied for IP traffic prediction. Predicted traffic was used as input for VNT re-optimization. Machine learning algorithms were employed to predict traffic conditions periodically (e.g., every hour). However, the prediction period was too long to adapt the fluctuations on the small scale traffic. As can be seen above, although the traffic prediction methods have gradually introduced into VNT reconfiguration, most of them are large scale traffic prediction methods and the traffic is predicted at each period that consumes more time and resources. Also, the VNT reconfiguration with large scale traffic prediction methods cannot adapt the fluctuations on small scale traffic within the prediction period.

### III. PROBLEM STATEMENT AND OPTIMIZATION MODEL

In this section, we formulate the VNT reconfiguration problem and design an optimization model.

---

**A. PROBLEM STATEMENT**

The SN is modeled as a graph $G_S = (N_S, E_S)$, in which the substrate node set and substrate link set are represented by $N_S$ and $E_S$, respectively. Similar to SN, the VN can be modeled as a graph $G_V = (N_V, E_V)$. $N_V$ represents the virtual node set and $E_V$ represents the virtual link set.

The following problems should be solved in VNT reconfiguration.

- Whether the current VNT is efficient for the current traffic.
- Whether the current VNT should be changed.
- How to change the current VNT by link addition or deletion.

In VNT reconfiguration, the first objective is to save more resources. Bandwidth resources are limited and relevant to the number of VNs that can be embedded successfully. By sensing the traffic and link utilization, the underutilized links can be deleted and the saved resources are used in link addition or VN embedding. Another objective is to minimize the cost that is related to the total number of additional and deleted links. Link addition and deletion lead the disruption of network service and consume the node resources for calculation and memory. Therefore, adding or deleting links frequently increase the total cost.

**B. OPTIMIZATION MODEL**

1) **NOTATIONS**

$s$ and $d$ denote the source and destination of a traffic flow.

$i$ and $j$ denote the originating and terminating nodes of a virtual link.

$m$ and $n$ denote the originating and terminating nodes of a substrate link.

The traffic matrix $T = \{T_{ij}\}$ denotes the traffic values between every virtual node pair. $T_{ij}$ denotes the traffic flow between virtual nodes $i$ and $j$.

The link hop matrix $H = \{h_{ij}\}$ denotes the hop counts of virtual links.

The bandwidth matrix $B = \{B_{ij}\}$ denotes the bandwidth resources of virtual links. The minimum value of $B$ is denoted by $B_{\text{min}}$.

The high threshold of link utilization is denoted as $W_H$. If the link utilization is higher than $W_H$, this link is considered to be overloaded or congested. $W_L$ is the low threshold of link utilization. If the link utilization is lower than $W_L$, this link is considered to be underutilized. The threshold of virtual link importance is $W_I$.

The link utilization matrix is denoted as $U = \{u_{ij}\}$. $u_{ij}$ is the utilization of virtual link $v(i, j)$ and it is denoted by the ratio of current traffic values between $i$ and $j$ to bandwidth.

$v$ is the number of prediction traffic within a prediction period.

$R_c$ is the consumed bandwidth resources in link addition. $R_f$ is the released bandwidth resources in link deletion. $R_s$ is the saved bandwidth resources in the VNT reconfiguration and $R_s = R_f - R_c$. 
2) VARIABLES

Current VNT \( \mathbf{V} = \{V_i\} \), where \( V_i \) is the binary value. \( V_i = 0 \) if there is no virtual link between nodes \( i \) and \( j \), and 1 otherwise.

New VNT \( \mathbf{V}' = \{V'_i\} \), which is similar to \( \mathbf{V} \). \( V'_i = 0 \) if there is no virtual link between nodes \( i \) and \( j \) in new VNT, and 1 otherwise.

Link embedding binary variable \( p(ij, mn) = 1 \) if virtual link \( vl(i, j) \) is embedded on substrate link \( sl(m, n) \), and 0 otherwise.

Node embedding binary variable \( x(i, m) = 1 \) if virtual node \( i \) is embedded on substrate node \( m \), and 0 otherwise.

Current traffic routing binary variable \( r(sd, ij) = 1 \) if traffic flow \( T_{sd} \) traverses the virtual link \( vl(i, j) \), and 0 otherwise.

New traffic routing binary variable \( s'(sd, ij) = 1 \) if traffic flow \( T_{sd} \) traverses the virtual link \( vl(i, j) \) in new virtual topology, and 0 otherwise.

The virtual link importance factor \( CR_{ij} = \sum_{e,f} \frac{\sum_{p \in e,f} \psi_p^{ij}}{P_{ef}} \). In which, \( P_{ef} \) is the shortest set between virtual nodes \( e \) and \( f \). The shortest link means the link has the shortest hop counts between the originating and terminating nodes. \( \psi_p^{ij} \) is the number of shortest links that traverse the virtual link \( vl(i, j) \). The \( CR_{ij} \) denotes the importance of the virtual link \( vl(i, j) \). If the virtual link with high \( CR_{ij} \) is deleted, it will have a high probability to be re-added. Hence, the \( CR_{ij} \) is important for avoiding the ping pong effect.

3) OBJECT

The objective function is to maximize the ratio of cumulative saved resources to the square of changed link number (CSR/SCLNR) that takes the resource and cost into consideration.

\[
\text{Max CSR / SCLNR} = \frac{\sum_{t} R_s(t)}{CN_C^2(t)}
\]

\( \sum_{t} R_s(t) \) is the cumulative saved resources in the VNT reconfiguration. \( CN_C^2(t) \) is the cumulative square number of changed links in the VNT reconfiguration. The saved resources can be used to embed more VNs or add more virtual links. Also, the cost of VNT reconfiguration is directly related to the \( CN_C \).

4) CONSTRAINTS

\[
\sum_{n} p(ij, mn) \leq 1
\]  
\( (1) \)

\[
\sum_{m} p(ij, mn) \leq 1
\]  
\( (2) \)

\[
\sum_{m} p(ij, ml) - \sum_{m} p(ij, lm) = 0, \text{ if } i \neq l, j \neq l
\]  
\( (3) \)

Equations (1)-(3) are the SN topology constraints. In (1), at most one outgoing substrate link of the source node is assigned to one virtual link. In (2), at most one incoming substrate link of the destination node is assigned to one virtual link. In (3), the number of incoming and outgoing links reserved for a substrate link of any intermediate node is equal.

\[
\sum_{i} \gamma(sd, iq) - \sum_{i} \gamma(sd, qi) = \begin{cases} 
1, & \text{if } q = d \\
0, & \text{if } q \neq s \\
-1, & \text{if } q = s
\end{cases}
\]  
\( (4) \)

Equation (4) is the flow conservation constraint used for routing traffic flows on virtual links.

\[
V_i' = \begin{cases} 
1, & \text{if } T_{sd} \leq B_{ij} \cdot (W_H - u_y) \\
0, & \text{otherwise}
\end{cases}
\]  
\( (5) \)

Equation (5) is the link addition constraint and the rerouted traffic flow \( T_{sd} \) traverses the virtual link \( vl(i, j) \) in the new topology without congestion.

\[
V_i' = \begin{cases} 
0, & \text{if } B_{ij} \cdot u_y \leq B_{ef} \cdot (W_H - u_{ef}) \\
1, & \text{otherwise}
\end{cases}
\]  
\( (6) \)

Equations (6)-(8) are the link deletion constraints. In (6), the virtual link \( vl(i, j) \) cannot be deleted if its destination link of traffic transfer \( vl(s, d) \) does not have more available bandwidth resources than total traffic values of \( vl(i, j) \). It ensures that the deletion of the virtual link should not cause congestion of other virtual links. In (7), the virtual link cannot be deleted if its \( CR_{ij} \) is less than \( W_1 \). In (8), we predict the future traffic of the virtual link to decide whether it should be deleted. If at least one future link utilization is higher than \( W_1 \), the virtual link cannot be deleted to avoid the ping pong effect.

IV. COGNITIVE VIRTUAL NETWORK TOPOLOGY RECONFIGURATION METHOD BASED ON TRAFFIC PREDICTION AND LINK IMPORTANCE

Solving this optimization model is computationally intractable. Most researchers solve this optimization model by proposing a corresponding heuristic algorithm that has the short computational time and gets an approximate optimal solution [15], [18], [20]. Therefore, we propose a novel heuristic algorithm called CVNTRM-TPLI to solve the optimization model of cognitive VNT reconfiguration. The block diagram is shown in Fig. 1.

As seen from Fig. 1, the network virtualization technology allows the SN to be shared by multiple heterogeneous VNs and provides different kinds of services. A network controller monitors VNs to obtain the current traffic and link utilization \( u_{ij} \). Then, the link utilization \( u_{ij} \) is compared with
the thresholds $W_H$ and $W_L$. If $W_L \leq u_{ij} \leq W_H$, there is no change to the VNT. If $u_{ij} > W_H$, another virtual link is added. If $u_{ij} < W_L$, we predict its future traffic with the help of HTPA-OPS and calculate the link utilization to decide whether this virtual link should be deleted. If none of the future link utilization is higher than $W_H$ and other constraints of link deletion are satisfied, this virtual link is deleted. Finally, the network controller controls the topology based on the information from VNT design.

Consequently, link addition, HTPA-OPS and link deletion are three important algorithms in CVNTRM-TPLI. Link addition is designed to solve the link congestion. HTPA-OPS and link deletion are used to increase the link utilization by deleting underutilized virtual links. The HTPA-OPS is introduced to predict the traffic and the prediction results are used as the inputs of link deletion algorithm to reduce the ping pang effect. The details of these three algorithms are shown as follows.

### A. LINK ADDITION

In CVNTRM-TPLI, link addition and link deletion are the core parts. The main idea of link addition is to decide how to transform traffic flows from a congested link to other available link. The main steps of the link addition are shown as follows.

In Algorithm 1, the overloaded traffic is calculated and traffic flows that traverse the congested virtual link are selected in lines 1-4. In lines 5-15, if there is only one hop count between nodes $e$ and $f$, add the bandwidth resources of the link and re-embed it. In lines 16-23, if there is more than one hop count between nodes $e$ and $f$, add the direct virtual link between the originating and terminating nodes. In lines 24-31, the matrices $B$, $U$ and $H$ are updated. If there is no virtual link that has the higher utilization than $W_H$, stop this process.

For example, three traffic flows traverse the virtual nodes $D$ and $E$ in Fig. 2. If the utilization of virtual link $vl(D,E)$ is higher than $W_H$ and the traffic flow $TF_{AF}$ has the highest traffic value, the direct virtual link between $A$ and $F$ is established and $TF_{AF}$ traverses $vl(A,F)$. Then the utilization of virtual link $vl(D,E)$ is lower than $W_H$.

### B. HTPA-OPS

1) **THE BLOCK DIAGRAM OF HTPA-OPS**

To delete underutilized virtual links accurately without causing the ping pang effect, traffic prediction is necessary before link deletion. In this paper, we propose the new traffic prediction method called HTPA-OPS by considering characteristics of small scale traffic, such as nonlinearity, mutation and chaos. The block diagram of HTPA-OPS is shown in Fig. 3.

---

**Algorithm 1 Link Addition Algorithm**

**Input:** $T$, $B$, $V$, $U$, $H$, $W_H$

**Output:** $T$, $B$, $V'$, $U$, $H$

1. Save the congested virtual links into set $CVL$
2. For each virtual link $vl(i,j)$ in $CVL$
3. Calculate the overloaded traffic $T_y$
   $$T_y = B_{ij} \cdot u_{ij} - W_H \cdot B_{ij}$$
4. Select the traffic flows that traverse $vl(i,j)$ and save them into $TF$
5. For $m = 1:length(TF)$
6. Select the $T_f(e,f)$ in $TF$ that has the maximum traffic value
7. If $h_{ef} = 1$
   \[ B_{ef} = B_{ef} + (B_{\min} + T_y/W_H) \]
   Re-embed the new virtual link $vl(e,f)$
8. If the re-embedding of $vl(e,f)$ is successful
   \[ T_y = T_y - T_f \]
9. Else
   \[ B_{ef} = B_{ef} - (B_{\min} + T_y/W_H) \]
   Delete the $T_f(e,f)$ from $TF$
10. End if
11. Else
12. Add the direct $vl(e,f)$ between originating and terminating nodes
   \[ B_{ef} = B_{\min} + T_y/W_H \]
13. If the embedding of $vl(e,f)$ is successful
   \[ T_y = T_y - T_f \]
14. Else
   Delete $T_f(e,f)$ from $TF$
15. End if
16. End if
17. Update $B$, $U$ and $H$
18. If $T_y < 0$
19. Break
20. Else
21. Continue
22. End if
23. End for
24. End if
25. End for
As seen from Fig. 3, the main steps of the HTPA-OPS can be summarized as follows.

Step 1: Collect the historical traffic samples and initialize the parameters. The original traffic flow is sampled with available intervals to find the historical traffic sequence. Then, we normalize the historical traffic sequence.

Step 2: Denoise the historical traffic sequence with the local projection method. The chaos characteristic of network traffic is usually affected by the high-dimensional noise. The local projection is a classical technique to denoise the network traffic and restore its inherent chaos in the nonlinear dynamic system.

Step 3: Reconstruct the historical traffic sequence with the phase space reconstruction method. Network traffic has complex dynamic characteristics and it is hard to describe with traditional low-dimensional coordinates. As an important method in chaotic time series prediction, the phase space reconstruction can describe the evolution of hidden chaotic attractor accurately and incorporate the existing values into a descriptive framework. Based on the chaos of network traffic, we transform the network traffic prediction into a nonlinear time series prediction problem. For the traffic prediction problem:

$$y = x_{q+1} = f \left(x_{q-(b-1)\tau}, x_{q-(b-2)\tau}, \ldots, x_{q-\tau}, x_q\right)$$  \hspace{1cm} (9)

where $x_{q+1}$ is the predictive result, \{ $x_{q-(b-1)\tau}, x_{q-(b-2)\tau}, \ldots, x_{q-\tau}, x_q$ \} is a historical traffic sequence, $b$ is the mapping dimension, and $\tau$ is the delay time. Assuming that the length of traffic data is $q$, after phase space reconstruction, training samples can be obtained.

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{q-(b-1)\tau} \end{bmatrix} = \begin{bmatrix} x_1 & x_1+\tau & \cdots & x_1+(b-1)\tau \\ x_2 & x_2+\tau & \cdots & x_2+(b-1)\tau \\ \vdots & \vdots & \ddots & \vdots \\ x_{q-1-(b-1)\tau} & x_{q-1-(b-2)\tau} & \cdots & x_{q-1} \end{bmatrix}$$ \hspace{1cm} (10)

Step 4: Select the optimal samples. In chaotic time series, the Euclidean distance is used to describe the relativity between the prediction value and the training sample. We select $k$ training samples that have the nearest Euclidean distances to the predicted traffic value.

Step 5: Select the initial optimal parameters with OCPSA-IO. Parameters $k$, $b$ and $\tau$ are related to the performance and complexity of the traffic prediction method. To select the optimal parameters, the OCPSA-IO is introduced and its details are described in next section.

Step 6: Train the traffic samples with the RBF neural network and predict the future traffic.

Step 7: Output the prediction samples with the RBF neural network and predict the future traffic.

Step 8: If the prediction error is higher than the threshold, update the optimal parameters with OCPSA-IO.

2) OCPSA-IO

The parameters $b$ and $\tau$ in phase space reconstruction and the parameter $k$ in optimal sample selection have important effect on the performance of prediction results. In order to select the optimal combination parameters, the OCPSA-IO is proposed. The block diagram of OCPSA-IO is shown in Fig. 4.

As seen from Fig. 4, the main steps of the OCPSA-IO can be summarized as follows.

Step 1: Initialize the network traffic information. The range of parameters $b$, $\tau$ and $k$ are set to $M$, $N$ and $K$, respectively.

Step 2: Generate the initial antibody population $A_n$. Every antibody $D_i \in A_n$ indicates one scheme of parameter selection. We set $D_i$ as follows.

$$D_i = [d_{i1}, d_{i2}, d_{i3}], \quad d_{i1} \in M, \ d_{i2} \in N, \ d_{i3} \in K$$ \hspace{1cm} (11)

where $d_{i1}$, $d_{i2}$ and $d_{i3}$ denote the values of parameters $b$, $\tau$ and $k$, respectively.
Step 3: Calculate the fitness value $A_{D_i}$, antibody affinity $S_{D_i,D_j}$ and the concentration $C_{D_i}$ of $D_i$.

$$A_{D_i} = \frac{1}{\sum_{y=1}^{v} |y(t) - \hat{y}_{D_i}(t)|} \quad (12)$$

$$S_{D_i,D_j} = \frac{\hat{S}_{D_i,D_j}}{|N_C|} \quad (13)$$

$$C_{D_i} = \frac{1}{|A_n|} \sum_{D_j \in A_n} S_{D_i,D_j},$$

$$S_{D_i,D_j} = \begin{cases} 1, & \hat{S}_{D_i,D_j} > \omega \\ 0, & \text{otherwise} \end{cases} \quad (14)$$

Among them, $y(t)$ denotes the current traffic values in one prediction period, $\hat{y}_{D_i}(t)$ denotes the predicted traffic values in one prediction period. $v$ is the number of prediction traffic in one prediction period. $\hat{S}_{D_i,D_j}$ denotes the number of the same elements between $D_i$ and $D_j$. The length of $D_i$ is $|N_C|$, $|A_n|$ denotes the total number of antibody population. $\omega$ is the threshold.

Step 4: Calculate the reproduction probability constant of antibody $P(D_i)$ and select the elitist.

$$P(D_i) = \varepsilon \frac{A_{D_i}}{\sum A_{D_i}} + (1 - \varepsilon) \frac{C_{D_i}}{\sum C_{D_i}} \quad (15)$$

where $\varepsilon$ is a constant.

Step 5: Produce the new antibody. The crossover, selection and mutation of antibody are used to produce the new antibody population. Then return to Step 2 until it satisfies the threshold $Er$.

C. LINK DELETION

After link addition, the utilization of each virtual link is lower than $W_H$. To decrease the number of underutilized virtual links, the link deletion is proposed and the underutilized virtual link can be deleted to release its possessed resources. To avoid the ping pong effect, the link importance and traffic prediction called HTPA-OPS are introduced into link deletion. The main steps of link deletion are shown in Algorithm 2.

In lines 1-6, the traffic values and the future utilization are predicted by the HTPA-OPS to decide whether this link should be deleted. In lines 7-9, the link importance is calculated to avoid the ping pong effect. In lines 10-27, the total traffic value in deleted link must be less than the residual bandwidth in the link that is traversed by the traffic flows of deleted link in a new VNT.

In this paper, the optimization object is to maximize the CSR/SCLNR. The CVNTRM-TPLI is proposed to solve the optimization model. It contains link addition, HTPA-OPS and link deletion algorithms. In link addition, the traffic flows that traverse the congested virtual link are selected and the direct virtual link between the originating and terminating nodes is added to reduce the resource consumption in link addition. The traffic prediction results of HTPO-OPS and virtual link importance factor $CR_{ij}$ are introduced into link deletion to
avoid the ping pong effect and satisfy the constraints (7)-(8). The underutilized virtual link that satisfies the constraints is deleted to reduce the cost of VNT reconfiguration. Therefore, the CVNTRM-TPLI can obtain the approximate optimal performance.

V. SIMULATION
To validate the performance of the CVNTRM-TPLI proposed in this paper, four comparative experiments are established in this section. The performance of the CVNTRM-TPLI is compared with the other three VNT reconfiguration algorithms in the first simulation experiment. Next, we simulate the performance of our traffic prediction method called HTPA-OPS. Then, we simulate the effect of different thresholds. Finally, we evaluate the effect of link importance and traffic prediction factors.

Initial VN: In this paper, the VNT is generated by the improved Salam network topology random generation algorithm. The VN is composed of 20 nodes and 80 links. The bandwidth of virtual link is [0.4, 0.6] Gbps.

Traffic: The traffic flow used in this paper is selected from real traffic data LBL-tcp-3.tcp [25]. The sample interval of original traffic flow is 1 second. The traffic sequence is obtained and normalized to get the network traffic used in this simulation. The traffic flow between each node pair is selected from above traffic series randomly with a continuous sequence. This selection is not vital for the CVNTRM-TPLI and the CVNTRM-TPLI should work for any input traffic. The routes over the VN are calculated by the constrained shortest path first algorithm.

Parameters: The range of $b$ is set to [1, 30]. The range of $\tau$ is set to [1, 10] and the range of $k$ is set to [10, 400]. The high threshold $W_H$ is set to 0.8 and the low threshold $W_L$ is set to 0.2. The link importance threshold $W_I$ is set to 3. The number of prediction traffic values $v$ is set to 3 and the traffic prediction error $Er$ is set to 0.05.

Simulation Environment: The computer in our simulations is a Lenovo Tianyi 510Pro with the Windows 10 operating system. The hardware platform is composed of an Intel Core i7-7700 3.6 GHz processor with 8 GB of RAM. The analysis software is Matlab R2007a.

In all simulation cases, the results are averaged over 50 simulations, and we show the margin of error with a 95% confidence level.

| Algorithm    | Description                                                                 |
|--------------|------------------------------------------------------------------------------|
| CVNTRM-TPLI  | The CVNTRM-TPLI is proposed in this paper. In link addition, the bandwidth of added link is $(B_{add}+T/W_0)$. In link deletion, the HTPA-OPS and link importance are used to decide whether the virtual link can be deleted. |
| VENTURE      | The VENTURE is proposed in [18]. An artificial neural network is used to provide robust and adaptive traffic models. In VNT reconfiguration, the traffic prediction result is taken as the input and the traffic flow can split into two parts. |
| IW           | The IW is proposed in [16]. The traffic adjusting algorithm is introduced. In link addition, the bandwidth of added link is $B_{add}$. In link deletion, all new added virtual links cannot be deleted in next time to avoid the ping pong effect. |
| GRVNT-ETM    | The GRVNT-ETM is proposed in [20]. In link addition, the bandwidth of added link is $(T/W_0)$. In link deletion, it does not consider ping pong effect. |
A. COMPARISON OF DIFFERENT VNT RECONFIGURATION ALGORITHMS

In this paper, we compare the CVNTRM-TPLI with the other VNT reconfiguration algorithms and their details are listed in Table 1. All these algorithms use the same VNT.

As seen from Figs. 5(a) and 5(b), the GRVNT-ETM saves more resources than the other three algorithms and it has the highest cumulative number of changed links. In link deletion, the link importance and traffic prediction are not taken into consideration in the GRVNT-ETM and it deletes more virtual links to release the bandwidth resources. The resources saved by IW are the lowest. Comparing with the other typical VNT reconfiguration methods, it does not delete the new added virtual links to solve the ping pong effect. Although it wastes some resources that can be released, it also reduces the cumulative number of changed links. In VENTURE, the integer linear programming and corresponding heuristic algorithm are proposed to minimize both the unserved traffic and used transponders. Also, the artificial neural network is employed to predict the traffic and decide whether the current VNT needs to be reconfigured. It saves more resources and decreases the cumulative number of changed links. However, the artificial neural network can predict the large scale traffic and it cannot solve the ping pong effect well. The CVNTRM-TPLI saves more resources than IW. In link deletion, the virtual links are deleted accurately with the help of the link importance. Also, the CVNTRM-TPLI has the lowest cumulative number of changed links. As seen from Fig. 5(c), the CVNTRM-TPLI has the highest CSR/SCLNR and the GRVNT-ETM is the lowest. It demonstrates that the CVNTRM-TPLI has the best performance.

B. THE PERFORMANCE OF TRAFFIC PREDICTION ALGORITHM

As seen from Fig. 6, the prediction result and actual traffic are accurately fitted. In the HTPA-OPS, the high-dimensional traffic prediction algorithm is used to predict the traffic and decide whether the current VNT needs to be reconfigured. It saves more resources and decreases the cumulative number of changed links. However, the artificial neural network can predict the large scale traffic and it cannot solve the ping pong effect well. The CVNTRM-TPLI saves more resources than IW. In link deletion, the virtual links are deleted accurately with the help of the link importance. Also, the CVNTRM-TPLI has the lowest cumulative number of changed links. As seen from Fig. 5(c), the CVNTRM-TPLI has the highest CSR/SCLNR and the GRVNT-ETM is the lowest. It demonstrates that the CVNTRM-TPLI has the best performance.
noise in network traffic is filtered by local projection to restore its chaos. The phase space reconstruction, optimal parameter selection and RBF neural network are all used to improve the performance of traffic prediction. In Fig. 7, the probability of prediction error is mainly distributed around zero. The probability is about 0.79 when the prediction error is zero. When the absolute error of prediction is above 0.02, the probability is quite small and even can be ignored. Consequently, the predicted traffic value can be used in the virtual link deletion.

C. THE EFFECT OF DIFFERENT THRESHOLDS

1) THE EFFECT OF $W_H$

The effect of changing the value of $W_H$ on the performance of the CVNTRM-TPLI for a fixed value of $W_L = 0.2$ is shown in Fig. 8. The $W_H$ and $W_L$ are denoted by $(W_H, W_L)$ in Fig. 8.

As seen from Figs. 8(a) and 8(b), with the increase of $W_H$, both the consumed bandwidth resources and cumulative number of added links decrease. When the parameter $W_H$ increases, the number of congested virtual links decreases which reduces the number of added virtual links. From Fig. 8(c), with the increase of $W_H$, the CSR/SCLNR increases. In this case, the cumulative number of added links has become the main factor affecting the CSR/SCLNR.
2) THE EFFECT OF $W_H$

The effect of changing the value of $W_L$ on the performance of the CVNTRM-TPLI for a fixed value of $W_H = 0.8$ is shown in Fig. 9. The $W_H$ and $W_L$ are denoted by $(W_H, W_L)$ in Fig. 9.

With the increase of the $W_H$, the released bandwidth resources and cumulative number of deleted links both increase. The virtual links that meet constraints in link deletion increase gradually and the number of deleted virtual links also increases. As seen from Fig. 9(c), with the increase of $W_L$, the CSR/SCLNR decreases.

D. THE EFFECT OF LINK IMPORTANCE AND TRAFFIC PREDICTION FACTORS

In this section, we simulate the performance of different VNT reconfiguration algorithms with different factors. To make a fair comparison, two baseline algorithms called CVNTRM-TPLI1 and CVNTRM-TPLI2 are proposed. The CVNTRM-TPLI1 is developed from the CVNTRM-TPLI, and it does not take the link importance and traffic prediction factors into consideration. The CVNTRM-TPLI2 is developed from the CVNTRM-TPLI, and it only takes the traffic prediction factor into consideration by introducing the HTPA-OPS.

From Figs. 10(a)-10(c), the CVNTRM-TPLI1 adds and deletes the virtual links frequently. It has the lowest CSR/SCLNR. The CVNTRM-TPLI2 reduces the cumulative number of changed links and saves more resources than CVNTRM-TPLI1. Its CSR/SCLNR is higher than CVNTRM-TPLI1 that evaluates the effect of HTPA-OPS. The CVNTRM-TPLI uses the HTPA-OPS and link importance and its CSR/SCLNR is higher than CVNTRM-TPLI2. It evaluates the effect of link importance factor.

VI. CONCLUSION

In this paper, we propose the VNT reconfiguration method to modify the topology by introducing the cognitive method. We formulate it as an optimization problem to determine which virtual link to be added or deleted. A heuristic algorithm called CVNTRM-TPLI is proposed to solve this optimization problem. In CVNTRM-TPLI, the link importance is proposed in link deletion as the topological factor to avoid the ping pong effect. Also, the HTPA-OPS is presented to predict the future traffic accurately. In HTPA-OPS, the OCPSA-IO is introduced to select the optimal parameters and the VNT can react promptly to the traffic fluctuation without adding or deleting virtual link frequently. Finally, four experiments are designed to demonstrate the performance of CVNTRM-TPLI. The first experiment results show that the performance of the proposed CVNTRM-TPLI is better than that of the other typical VNT reconfiguration methods. The second experiment assesses that the HTPA-OPS has good performance of traffic prediction. The third experiment assesses the effect of different thresholds. The last experiment evaluates the influence of link importance and traffic prediction factors on CVNTRM-TPLI. The link importance and traffic prediction factors are both useful to improve the performance of the CVNTRM-TPLI. The next step is to study the VNT reconfiguration method based on deep learning method in complex traffic environment.

REFERENCES

[1] P. Zhang, H. Yao, and Y. Liu, “Virtual network embedding based on computing, network, and storage resource constraints,” IEEE Internet Things J., vol. 5, no. 5, pp. 3298–3304, Oct. 2018.
[2] Y. Su, X. Meng, Q. Kang, and X. Han, “Survivable virtual network link protection method based on network coding and protection circuit,” IEEE Access, vol. 6, pp. 67477–67493, 2018.
[3] H. Cao, L. Yang, and H. Zhu, “Novel node-ranking approach and multiple topology attributes-based embedding algorithm for single-domain virtual network embedding,” IEEE Internet Things J., vol. 5, no. 1, pp. 108–120, Feb. 2018.
[4] Cisco System Inc., “Cisco visual networking index: Forecast and trends, 2017–2022 white paper,” New York, NY, USA, White Paper 155129690191003, Feb. 2019. [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-741490.html.
[5] F. Z. Yousaf, M. Bredel, S. Schaller, and F. Schneider, “NFV and SDN—Key technology enablers for 5G networks,” IEEE J. Sel. Area. Commun., vol. 35, no. 11, pp. 2468–2478, Nov. 2017.
[6] R. Misumi, J. Serrat, J.-L. Gorricho, N. Bouten, F. De Turck, and R. Boutaba, “Network function virtualization: State-of-the-art and research challenges,” IEEE Commun. Surveys Tutts., vol. 18, no. 1, pp. 236–262, 1st Quart., 2016.
[7] W. Golab and R. Boutaba, “Policy-driven automated reconfiguration for performance management in WDM optical networks,” IEEE Commun. Mag., vol. 42, no. 1, pp. 44–51, Jan. 2004.
[8] L. Peng, “A topology-awareness virtual network reconfiguration algorithm,” J. Sichuan Univ., vol. 47, no. 5, pp. 110–115, May 2015.
[9] N. Shahriar, R. Ahmed, A. Khan, S. R. Chowdhury, R. Boutaba, and J. Mitra, “ReNoVaE: Recovery from node failure in virtual network embedding,” in Proc. 12th Int. Conf. Netw. Service Manage. (CNSM), Montreal, QC, Canada, 2016, pp. 1–9.
[10] N. Shahriar, R. Ahmed, S. R. Chowdhury, A. Khan, R. Boutaba, and J. Mitra, “Generalized recovery from node failure in virtual network embedding,” IEEE Trans. Netw. Service Manage., vol. 14, no. 2, pp. 261–274, Jun. 2017.
[11] A. M. Ghaleb, T. Khalifa, S. Ayoubi, and K. B. Shaban, “Surviving link failures in multicast VN embedded applications,” in Proc. IEEE/IPCC/IFIP Netw. Oper. Manage. Symp., Istanbul, Turkey, Apr. 2016, pp. 645–651.
[12] Y. Su, X. Meng, Q. Kang, and X. Han, “Dynamic virtual network reconfiguration method for hybrid multiple failures based on weighted relative entropy,” Entropy, vol. 20, p. 711, Sep. 2018.
[13] Z. Qiang, W. Qiang, F. Sheng, and L. Wu, “Heuristic survivable virtual network embedding based on node migration and link remapping,” in Proc. IEEE 7th Joint Int. Inf. Technol. Artif. Intell., Chongqing, China, Dec. 2014, pp. 181–185.
[14] H. Zhou, L. Tan, Q. Zeng, and C. Wu, “Traffic matrix estimation: A neural network approach with extended input and expectation maximization iteration,” J. Netw. Comput. Appl., vol. 60, no. 6, pp. 220–232, Jan. 2016.
[15] A. Gencata and B. Mukherjee, “Virtual-topology adaptation for WDM mesh networks under dynamic traffic,” IEE/ACM Trans. Netw., vol. 11, no. 2, pp. 236–247, Apr. 2003.
[16] D.-R. Din and C.-W. Choo, “Virtual topology reconfiguration for mixed-line rate optical WDM networks under dynamic traffic,” Photon. Netw. Commun., vol. 30, no. 2, pp. 290–308, Oct. 2015.
[17] N. Fernández, R. J. D. Barroso, D. Siracusa, A. Francescon, I. de Miguel, E. Salvadori, J. C. Aguado, and R. M. Lorenzo, “Virtual topology reconfiguration in optical networks by means of optimization: Evaluation and experimental validation,” J. Opt. Commun. Netw., vol. 7, no. 1, pp. A162–A173, Jan. 2015.
[18] F. Morales, M. Ruiz, L. Gifre, L. M. Contreras, V. Lopez, and L. Velasco, “Virtual network topology adaptability based on data analytics for traffic prediction,” IEEE/OSA J. Opt. Commun. Netw., vol. 9, no. 1, pp. A35–A45, Jan. 2017.
[19] L. Gifre, L. M. Contreras, V. Lopez, and L. Velasco, “Big data analytics in support of virtual network topology adaptability,” in Proc. Opt. Fiber Commun., Anaheim, CA, USA, 2016, pp. 1–3.
[20] Y. Ohsita, T. Miyamura, S. Arakawa, S. Ata, E. Oki, K. Shiomoto, and M. Murata, “Gradually reconfiguring virtual network topologies based on estimated traffic matrices,” IEEE/ACM Trans. Netw., vol. 18, no. 1, pp. 177–189, Feb. 2010.

[21] M. Yoshinari, Y. Ohsita, and M. Murata, “Virtual network reconfiguration with adaptability to traffic changes,” J. Opt. Commun. Netw., vol. 6, no. 6, pp. 523–535, Jun. 2014.

[22] T. Ohba, S. Arakawa, and M. Murata, “Virtual network reconfiguration in elastic optical path networks for future bandwidth allocation,” IEEE/OSA J. Opt. Commun. Netw., vol. 8, no. 9, pp. 633–644, Sep. 2016.

[23] T. Ohba, S. Arakawa, and M. Murata, “Bayesian-based virtual network reconfiguration for dynamic optical networks,” J. Opt. Commun. Netw., vol. 10, no. 4, pp. 440–450, Apr. 2018.

[24] R. W. Thomas, D. H. Friend, L. A. DaSilva, and A. B. MacKenzie, “Cognitive networks: Adaptation and learning to achieve end-to-end performance objectives,” IEEE Commun. Mag., vol. 44, no. 12, pp. 51–57, Dec. 2006.

[25] P. Danzig, J. Mogul, and V. Paxson, Traces Available in the Interact Traffic Archive. Accessed: Jun. 15, 1995. [Online]. Available: http://ita.ee.lbl.gov/html/contrib/LBL-TCP.html

YUZE SU was born in Jinan, Shandong, China, in 1990. He received the B.S. degree in communication engineering from the Changsha University of Science and Technology, Changsha, in 2013, and the M.S. degree in information and communication engineering from Air Force Engineering University, Xi’an, China, in 2015, where he is currently pursuing the Ph.D. degree in information and communication engineering. His research interests include network virtualization and survivable virtual networks.

XIANGRU MENG was born in Lantian, Shaanxi, China, in 1963. He received the B.S., M.S., and Ph.D. degrees from Xi’an Jiaotong University, China, in 1985, 1988, and 1994, respectively.

From 1995 to 1997, he was a Visiting Scholar with the University of Electronic Science and Technology, Chengdu, China. He is currently a Professor with Air Force Engineering University, Xi’an, China. His research interests include the next generation Internet, network virtualization, and survivable networks.

ZHENHUA YU was born in Rushan, Shandong, China, in 1977. He received the B.S. and M.S. degrees from Xidian University, China, in 1999 and 2003, respectively, and the Ph.D. degree from Xi’an Jiaotong University, China, in 2006. He is currently a Professor with the Institute of System Security and Control, College of Computer Science and Technology, Xi’an University of Science and Technology, Xi’an, China. He has authored more than 20 technical articles for conferences and journals, and holds two invention patents. His research interests include cyber-physical systems and network virtualization.

QIAOYAN KANG was born in Yongchun, Fujian, China, in 1980. She received the B.S., M.S., and Ph.D. degrees from Air Force Engineering University, Xi’an, China, in 2001, 2004, and 2008, respectively, where she is currently an Assistant Professor. Her research interests include network management and survivable networks.