A Prediction-based Dynamic Resource Management Approach for Network Virtualization

Jiacong Li, Ying Wang, Zhanwei Wu, Sixiang Feng, Xuesong Qiu
State Key Laboratory of Networking and Switching Technology
Beijing University of Posts and Telecommunications
Beijing, China
Email: {huanshiweiyang, wangy}@bupt.edu.cn

***Abstract*—in network virtualization environment, multiple virtual networks share the same resource of a physical network. Since the physical resources of a substrate network is limited, it is necessary to improve the utilization of physical resources. Considering the resource requirement of a virtual network may change over its lifetime, we propose a prediction-based resource management mechanism. To increase the utilization of the substrate network, we can adjust the resource allocated to the virtual network based on the result of prediction. Additionally, in order to avoid the result of prediction deviates from the real requirement, we compare our prediction result with the collection of the resource utilization at real time to ensure the correctness of our result. The simulation results show that our approach can increase the utilization of the physical resource and improve the virtual network acceptance ratio while ensuring the requirement of the virtual networks.**

Keywords—network virtualization; prediction-based resource management; double exponential smoothing

I. INTRODUCTION

Network virtualization has been widely studied as a promising technology for the future Internet [1-3]. In network virtualization environment, the infrastructure provider (InP) manages the physical resource of the substrate networks (SNs), and the service provider (SP) builds virtual networks (VNs) by leasing virtual resources from InP [4]. Multiple VNs can be embedded in a shared substrate network.

In network virtualization environment, the main challenge is virtual network embedding (VNE) problem. VNE is to allocate substrate nodes and links to virtual nodes and links respectively. So far, there are many solutions of the VNE problem have been proposed. In some solutions, once a VN has been embedded successfully, the resource remains unchanged during its lifetime. These solutions do not consider the resource adjustment of the VNs after VNs have been embedded. To solve this problem, some researchers propose the dynamic VNE solutions, but they only adjust the VN embedding scheme when a new VN is embedded unsuccessfully. Therefore, the approaches above may not fully utilize the resources allocated to the VNs.

In this paper, we propose a prediction-based dynamic resource management method. In the proposed method, the VNE problem is resolved through two stages: formal VNE and dynamic resource management. When a VN arrives, the VN is firstly embedded by using a usual VNE algorithm. After the VN is embedded successfully, the actual resource utilization of the VN is recorded over the time. Based on the data recorded, the management framework predicts and adjusts the resource allocation of the VN in the next time episode. In order to avoid the result of prediction is wrong, we propose the check and backup mechanism to judge whether our prediction result is correct and allocate the backup resource when the prediction result is less than the real requirement.

This paper makes the following contributions: (1) We propose a dynamic resource management method based on prediction. Additionally, we also design the solution for the wrong result of prediction. (2) We apply a prediction algorithm based on DES algorithm to predict the resource requirement of the virtual networks. (3) Simulations are carried out to prove the proposed method can increase the utilization of resource and improve the virtual network acceptance ratio.

The rest of this paper is organized as follows. Section II discusses the related work. Section III describes the network model and defines the problem. Section IV describes the typical VNE and the prediction-based resource management approach. Section V describes our experimental methodology and evaluates the performance of our approach, and this paper is concluded in Section VI.

II. RELATED WORK

Many solutions have been proposed to resolve the VNE problem. These solutions can be classified into two categories: the static approaches [5-7], and the dynamic approaches [8-11].

The static approaches are used to allocate the resources of substrate networks for virtual networks statically. Yu et al. [5] advocated a different approach: rethinking the design of the substrate network to enable simpler embedding algorithms and making use of resources more efficiently. Houidi et al. [6] presented exact and heuristics optimization algorithms for the provisioning of virtual networks involving multiple infrastructure providers. Dietrich D et al. [7] presented a traffic matrix based VNE framework that enables VN request partitioning under LID, and conducted a feasibility study on VN embedding with LID compared to a "best-case" scenario where all information is available to VN Providers. In all these solutions, the resource allocation of a VN is static. However, the requirements of a VN will change during its lifetime in practical. As a result, these static solutions may waste the physical resource and cannot allow more virtual networks to be embedded.

In order to solve the problem mentioned above, researchers have proposed dynamic approaches of the VNE [8-11]. P. N. Tran et al. [8] proposed a reactive reconfiguration mechanism, which aims at minimizing the number of changed links and nodes to reduce the service disruption. M. R. Rahman et al. [9] developed a pro-active and a hybrid policy heuristic to solve VNE problem. N. F. Butt et al. [10] proposed a topology-aware VNE solution. M. F. Zhani et al. [11] designed a migration-aware dynamic virtual data center embedding framework. Although these methods can solve the problem of static approaches, they do not consider that the requirement of virtual network changes when VN embeds successfully.

To address this gap, we propose a prediction-based dynamic resource management method. The method is based on DES algorithm to predict the resource requirement of the VN to dynamically manage the resource. Additionally, we consider the situation that the result of the prediction is deviated from the real value to ensure the reliability of the network.

III. PROBLEM FORMULATION

A. Physical Network

The substrate network is represented by an undirected weighted graph $G_s = (N_s, L_s)$, where N_s and L_s donate the sets of physical nodes and physical links respectively. The attributions of each physical node are operation system (OS), location, and CPU capacity, and the attributions of each physical link are bandwidth and bandwidth price. Each physical network G_s belongs to an InP.

B. Virtual Network Request

The virtual network is represented by an undirected weighted graph $G_v = (N_v, L_v)$, where N_v and $L_v$donate the sets of virtual nodes and virtual links respectively. The requirements of each virtual node are OS, location and CPU capacity, and the requirements of each virtual link are bandwidth.

C. The Prediction-Based Resource Management

In this subsection, we propose a prediction-based resource management mechanism to manage the resource allocation for virtual networks dynamically. In this mechanism, the solution of VNE problem can be divided into two stages: typical VNE and dynamic resource management based on the predicting the real resource requirements of the VNs. The workflow of the prediction-based resource management method is described as Fig.1 shows.

When a VN request arrives, the VN is firstly embedded by using a typical VNE algorithm. The typical VNE stage is to allocate physical nodes and paths to the virtual nodes and links, according to the requirement of the VNs. In the stage of dynamic resource management, the allocated resources of the VN can be adjusted by predicting the resource requirement of the VN. After the VN is embedded successfully, the resource utilization of the virtual network are recorded. Based on the recorded data, we predict the resource requirements of the VN. Then, according to the prediction result, the resource allocated to virtual nodes and links are adjusted. To solve the situation that the result of prediction is wrong, we backup the resources of the difference between the request and the prediction result for some time. When we detect the utilization of resources is higher than the prediction, the backup resources will be used.

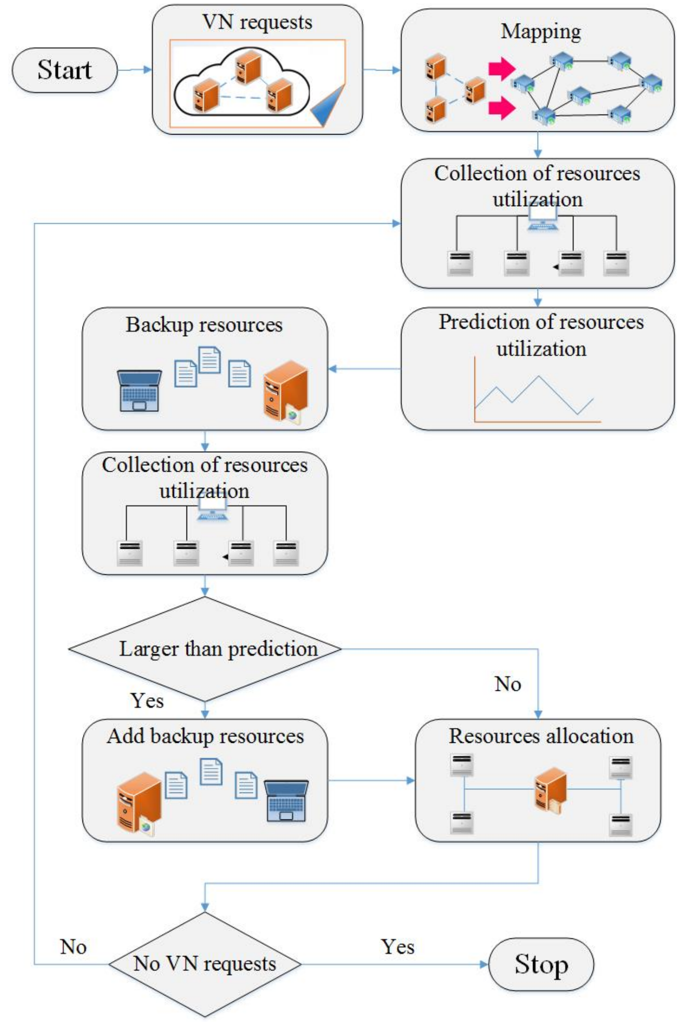

Fig. 1. The workflow of the prediction-based method

IV. THE PROPOSED MECHANISM DESIGN AND IMPLEMENTATION

A. The Typical Virtual Network Embedding

1) ILP for Formal Virtual Network Embedding

The Integer Linear Programming (ILP) formulation is given by function (1). N_p is the set of peering nodes. $C(n_v, n_s)$ denotes the cost of embedding virtual node n_v in peering node n_s. $C(l_{ij}, p_{mn})$ denotes the cost of embedding virtual link l_{ij}, m and n are the peering nodes that virtual nodes i and j are embedded in, p_{mn} is the path that l_{ij} is embedded in.

$$\text{Min} \sum_{n_v \in N_v, n_s \in N_p} C(n_v, n_s) + \sum_{i,j \in N_v, m,n \in N_p} C(l_{ij}, p_{mn}) \quad (1)$$

2) Algorithm for Particles Initialization

We propose a greedy algorithm to initialize the particles with resource awareness (IPRA). We defined a value $H(l_v)$ for each virtual link l_v to indicate the resources that the virtual link l_v is associated, and define $M(l_v)$ to denote the cost of embedding virtual link l_v. They are calculated as follows:

$$H(l_v) = \big(CPU(i) + CPU(j)\big) * BW(l_v) \quad (2)$$

$$M(l_v) = C(i) + C(j) + C(l_v) \quad (3)$$

where the i and j are the two endpoints of the virtual l_v. $CPU(i)$ and $CPU(j)$ are the required capacity of the virtual nodes i and j. $BW(l_v)$ is the required bandwidth of the virtual link l_v. $C(i)$ and $C(j)$ are the cost of embedding virtual node i and i. The IPRA used to initialize the particles is presented in Algorithm 1.

Algorithm 1: IPRA algorithm

1: Enqueue all the virtual link l_v of the VN to a queue L.
2: Put all the virtual nodes of the VN into a set N.
3: calculate the value $H(l_v)$ of all the links in the queue L.
4: Sort the virtual links in L in descending order $H(l_v)$.
5: Dequeue the first link in L, find its two endpoints i and j.
6: Query the set N, if both the two endpoints i and j are not in the set N, go to the step 5, else, embed the endpoints which are in the set N, and minimizing the cost $M(l_v)$.
7: Remove the link l_v from the queue L, and remove the endpoints which have been embedded from the set N.
8: If neither the L nor N is empty, go to step 5.

3) FVNE-PSO Algorithm Description

We provide a VNE approach named FVNE-PSO based on PSO [12]. The variants of PSO are used for discrete optimization problem [13-14]. The input of this algorithm are the information of a VN request and the virtual resource. The output is a VNE solution. This algorithm is presented in Algorithm 2.

Algorithm 2: FVNE-PSO algorithm

1: Set the parameters.
2: Initial the particles. Use the IPRA algorithm to initialize the position vector X of the first particle.
3: Initial P_Best and G_Best. Set the P_Best_i to equal to the initial X_i, and set the G_Best to equal to the vector X_j, where the value $f(X_j)$ of the particle j is the largest.
4: Update the velocity and position vector of the particles.
5: Calculate the fitness values f(X) of all the particles.
6: If the iterations finish, go to step 4, otherwise go to step7.
7: Output the G_Best.

B. The Prediction-based Resource Management

1) Double Exponential Emoothing Algorithm

Exponential smoothing (ES) algorithm was firstly proposed by Robert G [15]. Assume that the observation value at time t is Y_t, which can be seemed as a time series $\{Y_t\}$. The smoothed value S_t at time t is estimated by current observation value Y_t. It is given by the following Eq. (4).

$$S_t = \alpha * Y_t + (1-\alpha) * S_{t-1} \tag{4}$$

Where S_t is the smoothed value, Y_t is the actual measurement value at time t, and α is smoothing constant. In this paper, we use double exponential smoothing (DES) algorithm [16]. It is a variant of ES algorithm to predict the trend of virtual resource's usage. In this paper we only introduce the algorithm of the virtual node. DES algorithm constructs the prediction based on "levels mean" l_t and the "trend" T_t. They can be presented by the formulation as follows:

$$l_t = \alpha y_t + (1-\alpha)(l_{t-1} + T_{t-1}) \tag{5}$$

$$T_t = \beta(l_t - l_{t-1}) + (1-\beta)T_{t-1} \tag{6}$$

where, l_t denotes an estimate level of the physical resource used by the virtual node at time t, T_t represents an estimate of the slope of the series at time t, y_t is the observed value of the physical resource used by the virtual node. α and β are smoothing constants. Then, m period ahead predicting:

$$y'_{t+m} = l_t + mT_t \tag{7}$$

By substituting function (5) and (6) into (7), a predicting recursive equation for one period ahead is obtained by

$$y'_{t+1}(\alpha,\beta) = l_t + T_t$$

$$= (y_t - l_{t-1} - T_{t-1})\alpha + (l_t - l_{t-1} - T_{t-1})\beta + l_{t-1} + 2T_{t-1} \tag{8}$$

$$F(\alpha,\beta) = \sum_{t=3}^{N} f_t^2(\alpha,\beta) = \sum_{t=3}^{N} (y_t - y'_t(\alpha,\beta))\,\text{^}2 \tag{9}$$

The initial estimations are selected as follows:

$$l_1 = y_1 \tag{10}$$

$$T_1 = ((y_2 - y_1) + (y_3 - y_2) + (y_4 - y_3))/3 \tag{11}$$

By substituting function (11) into (5) and (6), l_t and T_t can be obtained, and then they are substituted into function (8). Thus, function (9) becomes a nonlinear equation. Then the smoothed value y'_{t+1} at time t+1 can be obtained.

2) Adjust the Resource Allocation for the Virtual Network

We use the function (12) and function (13) to compute the allocated resource $A(t+1)$ of the virtual node at time t+1.

$$\theta(t+1) = U(t) \tag{12}$$

$$A(t+1) = y'_{t+1}/\theta(t+1) \tag{13}$$

Where θ is the adjustment coefficient. $U(t)$ is the actual resource utilization of the virtual node at time t. The adjustment coefficient $\theta(t+1)$ is equal to the resource utilization $U(t)$. At time t+1, the amount of allocated resource is equal to the ratio of the predicted values y'_{t+1} and the ratio of the adjustment coefficient $\theta(t+1)$. Since the adjustment of the resource allocation is not necessarily optimal. We recalculate the adjustment coefficient $\theta(t+1)$ as function (14).

$$\theta(t+1) = (\Delta \frac{A(t)-y'_t}{A(t)+y'_t} + 1)U(t) \tag{14}$$

Where Δ the adjustment coefficient of the $\theta(t+1)$. The function (14) describes the mathematical relationship between the adjustment coefficient $\theta(t+1)$ and the resource utilization $U(t)$. Then, substituting the function (14) into (12). According to the function (14), the $\theta(t+1)$ can be adjusted in timely to adapt to the process the resource allocation. The θ is related to the ratio of the resource allocation and the resource utilization.

V. Performance Evaluation

A. Experimental Environment and Metrics

The network topologies are randomly generated via GT-ITM tool [17]. The size of a physical network ranges from 50 to 100 nodes. The node capacity varies from 200 to 300. The unit price of a node capacity varies from 1 to 10. Each pair of nodes is connected by an intra-domain link with a probability of 0.5. The link capacity ranges from 1500 to 3000. For a virtual network request, the number of virtual nodes varies from 5 to 15. The actual required capacity of a virtual node and the required bandwidth of each virtual link follows a Poisson distribution.

In this paper, we compare our dynamic method based on prediction (DMP) with the common static method (CSM) and the dynamic method are triggered by failed VNE (DMF).

In this evaluation, we use three metrics to compare our approach with other approaches. The first one is the acceptance ratio of virtual network. The definition is as the function (15).

$$acceptance\ ratio = \frac{the\ number\ of\ the\ VN\ emdedded}{the\ number\ of\ the\ VN\ arrived} \qquad (15)$$

The second metric is the resource utilization. It means the average utilization of nodes and links. The third metric is mean square error (MSE), which is used to analyze the accuracy of the prediction algorithm. The definition of the MSE is as function (16), where n is the number of samples.

$$MSE = \sqrt{\frac{\sum_{i=1}^{n}(actual\ utilization - allocated\ value)}{n}} \qquad (16)$$

B. *The Results of Evaluation*

In Fig.3, with number of VN request increases, the VN acceptance ratios are all decreases and the DMP decreases slower than the static method. The reason of the result is that DMP constantly predicts the resource requirement of the VNs, and based on the prediction to adjust the resource allocated.

In Fig.4, the resource utilization of CSM changes. Because that the VN is allocated with the fixed amount resource in CSM, and CSM cannot always make full use of the allocated resource. On the contrary, the resource utilization of DMP increases gradually and tends towards stable. The reason is that the DMP can reallocate the resource to match the actual requirements.

In Fig.5, the mean square error of resource utilization of the static method dynamically changes over time. However, the DMP gradually decreases and tends towards stable. Because that the DMP reallocate resource to VN according to the actual demand of the VN over the time. Therefore, the dynamic method can accurately allocate resource to the VNs.

In Fig.6, we investigate the prediction interval of the DMP. Prediction interval is the time interval of running the prediction algorithm. When the interval is small, the acceptance ratio is also low. Because that frequently start prediction algorithm will adjust the network resource allocation frequently. Thus, a proper prediction interval is important to the prediction algorithm.

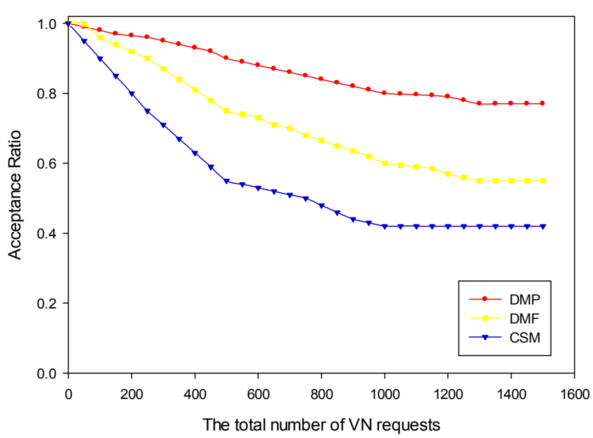

Fig. 3. The acceptance ratio of different methods

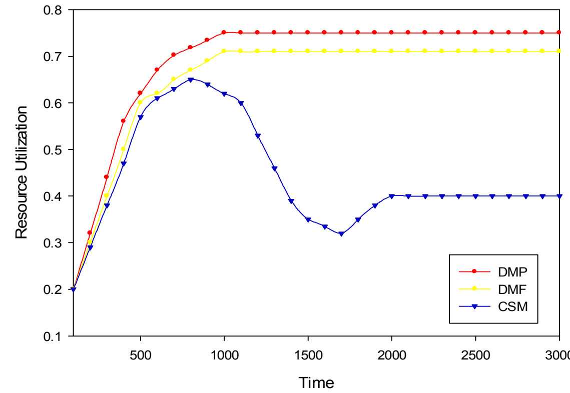

Fig.4. The resource utilization of defferent methods

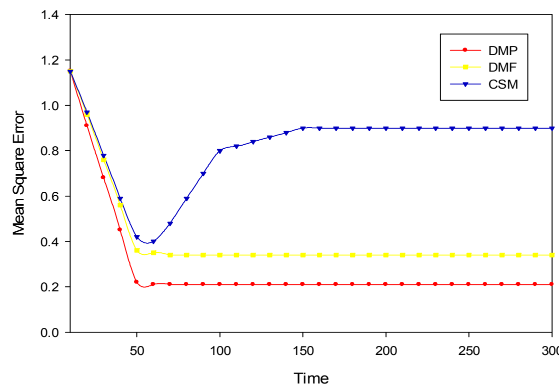

Fig. 5. The accuracy of the prediction algorithm

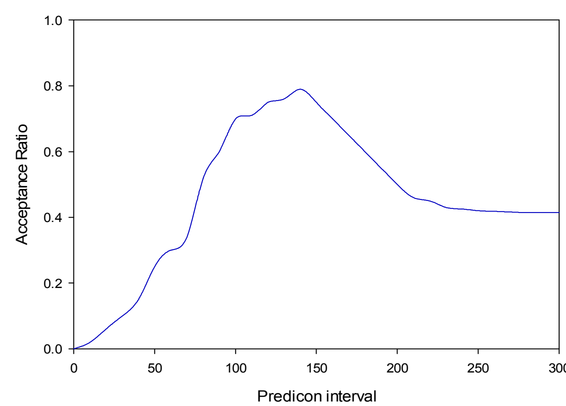

Fig. 6. The influence of the prediction interval

VI. CONCLUSION

We propose a prediction-based resource management method to manage the resource allocated to the VNs. Our method can be divided by two stages: typical VNE and dynamic resource management based on prediction. We mainly study the prediction algorithm, which is based on the DES. DES can be constructed by using a linear regression equation and compute data easily. The simulation results show that our method can increase the VN acceptance ratio and the resource utilization of SN. In the future, we plan to optimize the prediction algorithm and provide more experiments to prove the practicability of it.

ACKNOWLEDGMENTS

This work was supported by the National Natural Science Foundation of China (61501044).

REFERENCES

[1] T. Anderson, L. Peterson, S. Shenker, and J. Turner, “Overcoming the Internet impasse through virtualization,” Computer, vol. 38, no. 4, pp.34–41, 2005.

[2] J. Turner and D. Taylor, “Diversifying the Internet,” in GLOBECOM’05,vol. 2, 2005.

[3] N. M. M. K. Chowdhury and R. Boutaba, “Network virtualization: State of the art and research challenges,” IEEE Communications Magazine,vol. 47, no. 7, pp. 20–26, July 2009.

[4] [1] A. Fischer, J. Botero, M. Beck, H. De Meer, and X. Hesselbach,“Virtual network embedding: A survey”, IEEE Communications Surveys Tutorials, pp. 1-19, 2013

[5] M. Yu, Y. Yi, J. Rexford, and M. Chiang, “Rethinking virtual network embedding: substrate support for path splitting and migration,” Computer Communication Review, vol. 38, no. 2, pp. 17–29, 2008.

[6] I. Houidi, W. Louati, W. Bean-Ameur, and D. Zeghlache, “Virtual Network Provisioning Across Multiple Substrate Networks,” Computer Networks, 55(4), March 2011.

[7] Dietrich D, Rizk A, Papadimitriou P, “Multi-Domain Virtual Network Embedding with Limited Information Disclosure,” IFIP Networking Conference. 2013. 1-9.

[8] P.N.Tran, L.Casucci, A.Timm-Giel, “Optimal Mapping of Virtual Networks Considering Reactive Reconfiguration,” IEEE 1st International Conference on Cloud Networking, pp.35-40, 2012.

[9] M. R. Rahman and R. Boutaba, “SVNE: Survivable Virtual Network Embedding Algorithms for Network Virtualization”, IEEE Transactions on Network and Service Management, VOL. 10, NO. 2, pp. 105-118,

[10] G. Sun, H. Yu, V. Anand, L. Li, “A cost efficient framework and algorithm for embedding dynamic virtual network requests”, Future Generation Computer Systems 29, pp. 12651277, 2013.

[11] M. F. Zhani, Q. Zhang, G. Simon and R. Boutaba, “VDC Planner: Dynamic Migration-Aware Virtual Data Center Embedding for Clouds”, Proc. of the IFIP/IEEE Integrated Network Management Symposium (IM). Ghent (Belgium), pp. 18 - 25, 2013.

[12] J. Kennedy, R. Eberhart, et al., “Particle swarm optimization,” IEEE International Conference on Neural Networks 4,1995, pp. 1942–1948.J.

[13] K. Wang, L. Huang, C. Zhou, W. Pang, “Particle swarm optimization for traveling salesman problem,” IEEE International Conference on Machine Learning and Cybernetics, vol. 3, 2003, pp.1583–1585.

[14] Xiang Cheng, Sen Su, Zhongbao Zhang, et al., “Virtual network embedding through topology awareness and optimization,” Computer Networks, April 2012, pp. 1797-1813.

[15] Brown RG,Meyer RF. “The fundamental theorem of exponential smoothing”. Operations Research,1961, 9(5),673-685

[16] Charles C. Holt. Forecasting seasonals and trends by exponentially weighted moving averages. International Journal of Forecasting, 20(2004): 5-10.

[17] GT-ITM. http://www.cc.gatech.edu/projects/gtitm/