ROUTING OPTIMIZATION IN SDN USING DEEP REINFORCEMENT LEARNING

D. Naga Maheswari[1], C. B. Sujitha[2] & Dr. Kadiyala Ramana[3]

[1,2]*UG Student, Annamacharya Institute of Technology and Sciences, Rajampet, A.P., India.*

[3]*Associate Professor in CSE, Annamacharya Institute of Technology and Sciences, Rajampet India.*

maheswaridharmisetty@gmail.com , sujithareddy109@gmail.com , ramana.aits@gmail.com

Abstract: *To achieve routing optimization in the Software Defined Network (SDN) environment, deep reinforcement learning is placed on the SDN routing process and a method for optimizing routing is proposed based on deep reinforcement learning. This method will boost network performance such as delay, latency and in continuous time realize black-box optimization, which will certainly minimize network operation and maintenance costs. In addition, a set of experiments test the proposed routing optimization method. The experimental results show that there is strong consistency and efficiency in the proposed SDN routing optimization mechanism and can provide better routing configurations and performance stability than conventional routing protocols.*

Keywords: **We would like to encourage you to list your keywords in this section**

1. INTRODUCTION

Routing optimization and particularly traffic engineering are the most fundamental networking activity, and have therefore been studied extensively in a variety of contexts. Software Defined Networking (SDN) model evolving introduced new routing capabilities. In SDN, it distinguishes the control plane and the forwarding plane. A centralized controller is responsible for routing decisions in computing, thus reducing the complexity of network components. In addition, it is capable of tracking resource demands and availability globally; hence, it is capable of optimally balancing resource needs. However, both the size of the solution space and the complexity of the optimization problem are increased because the SDN model introduces degree of routing freedoms (flow-based forwarding vs destination-based forwarding) and the centralized controller has to solve the problem with a global view. New SDN routing approaches have been suggested as a consequence [1, 2]. In [3] the author proposed to monitor the network a Information Plane (KP), which is based on machine learning and cognitive techniques. The KP can bring many networking benefits, such as automation and suggestion, and it can lead to a paradigm shift in the way we run, handle, and optimize the data networks.

Machine learning (ML) techniques were adopted in a number of application areas and made breakthroughs. It is possible to classify ML algorithms into three categories: supervised learning (SL), unsupervised learning (USL), and reinforcement learning (RL). While SL and USL concentrate on classification or regression tasks, RL algorithms learn to determine the best series of actions to optimize a given objective (i.e. reward) function. ML's most important advantages are its ability to manage complicated issues; thus, it is intuitive to exploit ML in the network domain where the complex issues are common [4]. RL was verified in the sense of routing that it outperforms other ML techniques [5]; therefore, this paper focuses only on RL techniques.

RL techniques were used to address optimization of the routing [6]. They are also used in QoS routing [7]; however, table dependent RL agents are unable to provide efficient solutions for unseen network states. Deep Reinforcement Learning (DRL) is capable of providing solutions for unknown network states which conventional table-based RL agents cannot achieve. It also overcomes the iterative optimization and heuristics development process by providing a DRL agent delivering a near-optimal solution in a single stage. Recent breakthroughs in deep neural networks [8, 9] have enhanced DRL

performance; thus, paving a way for networking to implement DRL. Deep convolution neural networks [10] have proven their efficiency in a variety of applications, e.g. image processing; thus, the use of deep convolution neural networks in networking that give some benefits.

In [5], the authors analyzed the impacts of inputs and action spaces on machine learning output in connection with routing problems. The paper results verified learning the weights of the connections with the traffic matrix outperforms other approaches. The connection weights can have real positive values which the well-known Deep Q Network algorithm [11] cannot solve since it can handle only discrete and low-dimensional space of action [12]. By proposing a Deep Deterministic Policy Gradient (DDPG) algorithm the authors in [12] addressed the needs of continuous power. DDPG's routing problem solving capability has been verified in [13]. However, the author in [13] concentrated on reducing mean network latency and did not discuss multiple metrics (e.g., latency and packet loss rate) of the QoS routing issue.

2. LITERATURE REVIEW

Routing is a vital network function. In legacy networks, routing decisions are made by switches in decentralized manner. In SDN, the central controller makes routing policies by installing flow tables on the switches. Optimizing routing allows for effective use of network resources and QoS provision. The SDN architecture takes advantage of the controller's global view of the network to make efficient routing decisions.

Heuristic algorithms like ant colony optimization are one of the most effective routing optimization approaches [15]. Heuristic solutions give close to optimal solution, but they involve a high computational time cost and hence unable to calculate optimal solutions in real time [15]. This drawback of heuristic algorithms makes impractical to run such algorithms in the SDN controller which is responsible for calculating the route for flows. Machine learning operates on building relationships between features and does not need the exact network model as heuristic algorithms. Also, once a ML model has been trained, it is able to predict near-optimal solutions in a very short time making it effective for real time use. Several studies have been conducted on using ML especially Reinforcement learning to solve the routing optimization problem. Related studies have been presented below grouped based on the ML approach used.

[14] presents a routing framework called Neuroute. The Neuroute used a traffic matrix predictor based on a Long Short-Term Memory Recurrent Neural Network (LTSM-RNN). The model took in traffic matrix at time t and output the traffic for time t+1.LTSM-RNN was trained on sampled training data using backpropagation to build the predictor model. The predicted network traffic and networks states as inputs and their corresponding heuristic routing solutions as outputs are used to train a Deep Feed Forward Neural network to build a traffic routing unit. The architecture was implemented on the GEANT network topology. The work focused on effective routing of traffic volumes and necessarily meeting QoS requirements of specific traffic classes.

[15] presents another optimized routing solution based on supervised learning. They presented a framework with ML based meta-layer for routing in real-time. The architecture the output of a heuristic algorithm and its corresponding network information as input data to build a neural network that can provide heuristic-like routing decisions. Their proposal was applied on the NSTNET and compared to some other routing solutions and the running time was found to be faster.

[16] presented a reinforcement learning based distributed routing based on SDN. Though in an SDN topology, they implemented the routing protocol to run in a distributed manner. The RL algorithm was added to the OSPF routing protocol and tested. They compared it with the traditional OSPF routing protocol and their approach achieving

better QoS delays and jitter. [7] presented a routing approach based on RL learning algorithm. The optimization was implemented in a multi-layer hierarchical SDN architecture. The architecture choice was to reduce signaling delay. The SDN controllers were designed in three levels; the super, domain/master and slave controllers as shown in figure below. Each of these controllers had their unique set of responsibilities to reduce the burden on the super controller. The slave controllers provided read only access to the data plane and received port messages from them. The slave controllers also performed simple control functions as traffic admission, flow control etc. The domain controllers received flow set-up requests and installed flow rules on switches. The super controller had a global view of the network and controlled full network functions. A QAR algorithm with QoS based reward was implemented in the super controller to achieve optimal routing solutions. The proposal was deployed in a real Sprint GIP network and tested. The approached provided fast convergence as compared to existing learning solutions.

[17] also presented an approach based on deep reinforcement learning. The paper used a DRL mechanism called DDPG to determine optimal routing solutions. The optimization mechanism outlined in the paper is called DROM and experimental results proved that it had good convergence and effectiveness

[18] utilizes ML and SDN to improve scheduling and control in data centers. The framework involved two steps. The first step involved using Machine learning at the edge of the network to classify the traffic. The classification result is the used by the centralized SDN controller in conjunction with its global view of the network to implement optimized routing solutions. This paper did not provide specifics as to which algorithms and mechanisms will be used for the classification and routing. Another work that combines traffic classification and route optimization is presented in [19]. They proposed a network traffic classification based on a Deep Neural network (DNN) classifier in an SDN environment. A VNF is introduced in the architecture to reduce the burden on the SDN. When a flow arrives, the SDN switch routes the flow based on the flow rules installed. If there is no flow rule for the packet, it sends a request to the controller. The controller calculates the path to route the flow based on the network topology and installs these flows in the SDN switches. The controller also installs a flow rule to pass a copy of the packet to a port where the VNF will be listening. The VNF captures the traffic and extract features for traffic classification based on the trained DNN. The identified class information is tagged in the DSCP field and sent to the SDN controller. The SDN controller now searches for a path that will meet the QoS requirements of the identified traffic class and installs those flows with high priority in all switches along the path giving them priority over the initially installed flows. This work presents an approach to improve the network quality, but the work does not propose any optimization approach that as used by the SDN controller after classification to determine the path. The controller simply uses its default routing mechanism based on the global view to route the identified packet.

3. PROPOSED SYSTEM

The proposed RL agent is an off-policy, actor-critical, deterministic policy gradient algorithm[20] that interacts with its environment (i.e. the network) by exchanging the state, action and reward of the three signals[21]; Specifically, the state is represented by the Traffic Matrix (TM, being the bandwidth request for each source destination pair), the operation by a tuple of link-weights (that univocally defines the paths for all source-destination node pairs), and the reward is based on the mean network delay. The agent's objective is to evaluate the optimal behavior policy π mapping from the space of the S states to the space of actions A ($\pi : S \rightarrow A$) that maximizes the expected reward $r \in R$ (minimizes the delay of the network). It does so by improving its awareness of the

relationship between the three signals iteratively across two deep neural networks (one for the performer, one for the critics) [22].

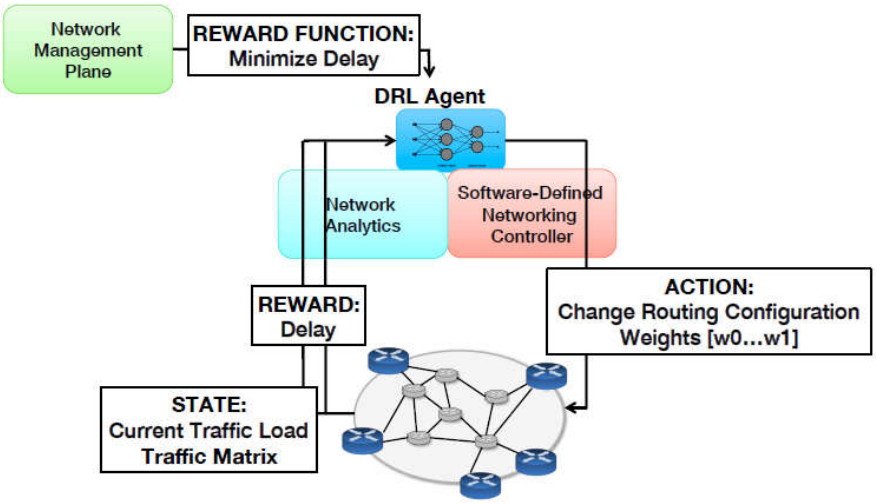

Figure 1. Proposed System

4. METHODOLOGY

We use a scale-free network topology of 25 nodes and 53 full-duplex links to assess the efficiency of the proposed DRL agent, with uniform connection capacities and average node degree of 3 as shown in Figure 2.

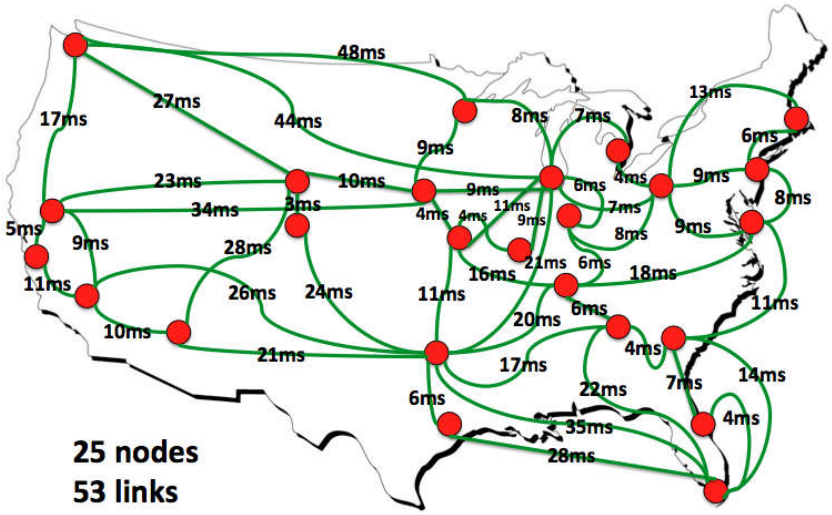

Figure 2. Backbone Network

We use 4 levels of traffic intensity (TI), ranging from 10 to 100 percent of total network capacity. We produce 100 TMs of equivalent total traffic for each TI using a gravity model [23], thus having 1000 distinct traffic configurations varying over both the total traffic volume and its distribution. We compare its output against 100,000 randomly generated routing configurations as an initial step towards evaluating our DRL agent. Those configurations are true in the sense that: (i) all nodes are accessible, (ii) no loops are open. Same 100,000 routing configurations are used for each combination of TI and TM. The DRL agent was trained with gravity-generated TMs in 100,000 steps. The agent itself defined the routing configurations (i.e. actions), as shown in Figure 3. We applied a stochastic rate of exploration to the behavior of the agent (overriding its deterministic

choices of action) to prevent it from getting trapped in local minima during the training [21]. Then, in a single step, we introduced each of the 1,000 test TMs to the trained DRL agent in order to get its expected solution. Throughout the experiment the OMNeT++ discrete event simulator (v4.6) [24, 25] was used to obtain the network delay under the specified traffic and routing. Both datasets and source code used on this paper can be found at [26].

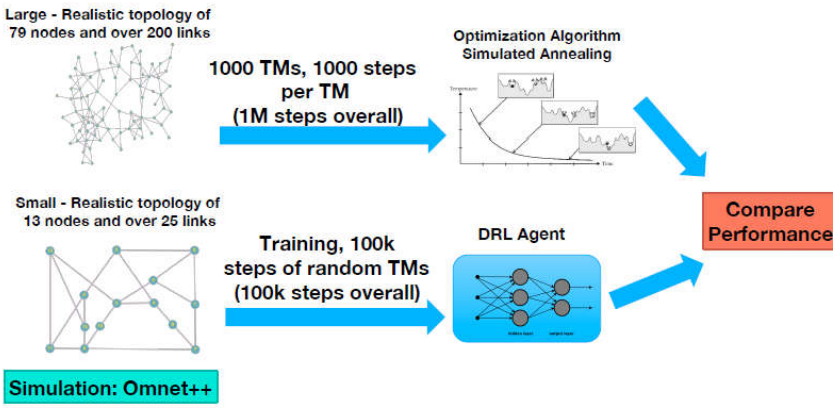

Figure 3. Proposed Methodology

5. EXPERIMENTAL RESULTS

We use four separate TLs in the experiment: 10 percent, 40 percent, 70 percent and 100 percent, respectively, of the total network bandwidth. We're generating 250 traffic matrices for every TL. We train DDPG agents with the same traffic matrices under different TLs for 2000, 4000, 10000, 20000, 50000, 80000 and 100000. Then, we test the DRL agent's output with 1000 test traffic matrices as the input and obtain the routing results for each TL from the qualified agent. Given the traffic matrix and the routing solution, we can obtain network output parameters directly from the OMNeT++, such as network delay. To ensure the validity of the data, we use the average results of 1000 experiments to reflect the training results.

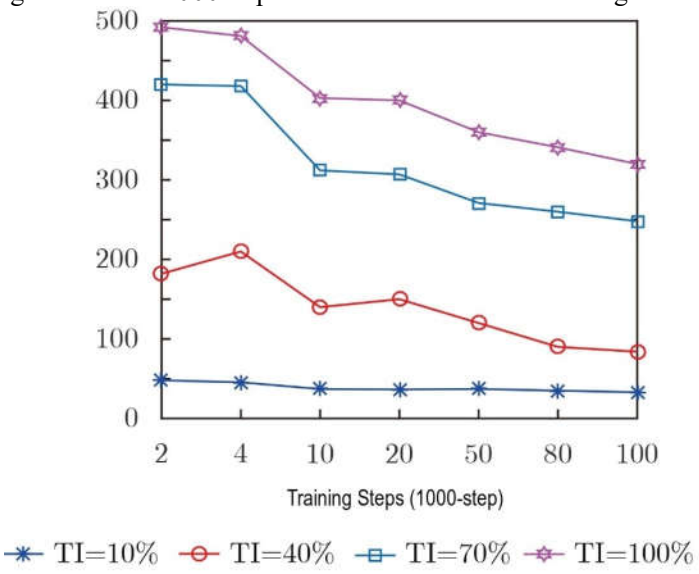

Figure 4. Network delay of training steps under different traffic loads

The experimental results are shown in figure 4. Within the figure the DRL agent decreases the network delay effectively. As training steps increase the delay in the

network decreases. Under TL = 70%, DRLs delay performance with 100,000 steps training improves 40.4%, which is lower than DRL's performance with 2000 steps training.

In Figure 5, the upper and lower sections of the rectangular represent the upper quartile and lower quartile values of the delays respectively, and the line in the center of the rectangular is the median of the delays. The upper and lower ends of the straight line that stretches from the rectangle represent, respectively, the maximum and minimum delay values. The outliers are not shown in the figure, for simplicity. Experimental findings show that DRL delays are less than the lower quartile delay of the randomly generated routing configurations, and DRL's minimum delay is very similar to the best outcome of the randomly generated routing configurations.

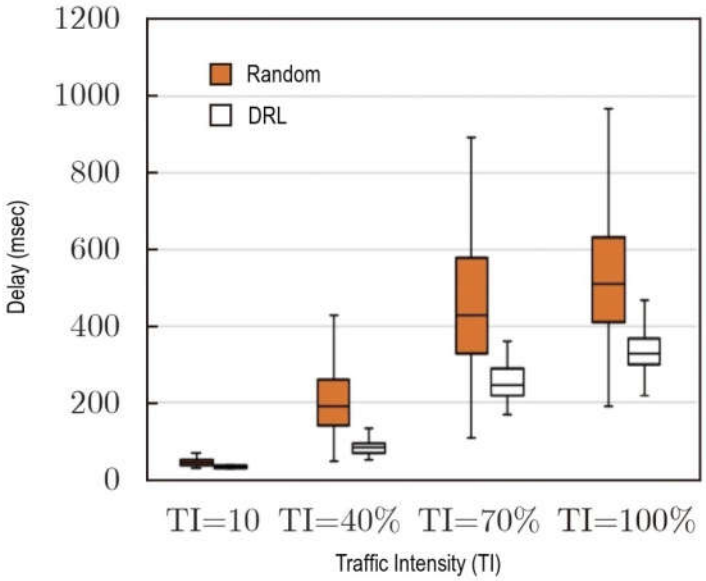

Figure 5. Comparison of DRL and random routing configuration under different traffic loads

7. CONCLUSION & FUTURE SCOPE

In this paper, we propose an SDN-based machine learning architecture, which uses a novel deep-reinforcement learning method called DDPG to optimize the SDN routing process. Based on the system, we are proposing a routing optimization method called DRL for continuous time realization of national, real-time, and custom network intelligence control and management. Results of the experiment show that DRL has a strong convergence and productivity, and DRL can boost network performance with reliable and superior routing services compared with existing routing solutions.

From the above results, we can see that with the various operation and maintenance approaches the DRL can achieve the customizable routing optimizations. The way a strategy is developed is still an open question. If we can combine the QoS-aware traffic classification and network measurement with the DRL, we can create a strategy that is adaptively created to make end-to-end transportation QoS-aware, reliable and efficient. As our future work, we leave this interesting issue.

REFERENCES

[1] *Akyildiz, I.F., Lee, A., Wang, P., Luo, M., Chou, W.: A roadmap for traffic engineering in SDN-open flow networks. Computer Networks 71,30 (2014).*

[2] *Layeghy, S., Pakzad, F., Portmann, M.: SCOR: software-defined constrained optimal routing platform for SDN. CoRR abs/1607.03243 (2016),*

[3] *Clark, D.D., Partridge, C., Ramming, J.C., Wroclawski, J.T.: A knowledge plane for the internet. In: Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications. pp. 3(10). ACM (2003)*

[4] *Wang, M., Cui, Y., Wang, X., Xiao, S., Jiang, J.: Machine learning for networking: Work flow, advances and opportunities. IEEE Network 32(2), 92{99 (March 2018).*

[5] *Valadarsky, A., Schapira, M., Shahaf, D., Tamar, A.: A machine learning approach to routing. CoRR abs/1708.03074 (2017),*

[6] *Boyan, J.A., Littman, M.L.: Packet routing in dynamically changing networks: A reinforcement learning approach. In: Advances in neural information processing systems. pp. 671-678 (1994)*

[7] *Lin, S., Akyildiz, I.F., Wang, P., Luo, M.: Qos-aware adaptive routing in multi-layer hierarchical software defined networks: A reinforcement learning approach. In: 2016 IEEE International Conference on Services Computing (SCC). pp. 25-33 (June 2016).*

[8] *Li, Y.: Deep reinforcement learning: An overview. arXiv preprint arXiv:1701.07274 (2017)*

[9] *Arulkumaran, K., Deisenroth, M.P., Brundage, M., Bharath, A.A.: A brief survey of deep reinforcement learning. CoRR abs/1708.05866 (2017), http://arxiv.org/abs/1708.05866*

[10] *Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. pp. 1097-1105 (2012)*

[11] *Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G., et al.: Human-level control through deep reinforcement learning. Nature 518(7540), 529 (2015)*

[12] *Lillicrap, T.P., Hunt, J.J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., Wierstra, D.: Continuous control with deep reinforcement learning. arXiv preprint arXiv:1509.02971 (2015)*

[13] *Stampa, G., Arias, M., Sanchez-Charles, D., Munt_es-Mulero, V., Cabellos, A.: A deep-reinforcement learning approach for software-defined networking routing optimization. CoRR abs/1709.07080 (2017), http://arxiv.org/abs/1709.07080*

[14] *Azzouni, A., Boutaba, R., & Pujolle, G. (2017). NeuRoute: Predictive dynamic routing for software-defined networks. In 2017 IEEE International Conference on Network and Service Management (CNSM), pp. 1-6.*

[15] *Yanjun, L., Xiaobo, L., & Osamu, Y. (2014). Traffic engineering framework with machine learning based meta-layer in software-defined networks. In 2014 IEEE International Conference on Network Infrastructure and Digital Content, pp. 121-125.*

[16] *Sendra, S., Rego, A., Lloret, J., Jimenez, J. M., & Romero, O. (2017). Including artificial intelligence in a routing protocol using software defined networks. In 2017 IEEE International Conference on Communications Workshops (ICC Workshops), pp. 670-674.*

[17] *Yu, C., Lan, J., Guo, Z., & Hu, Y. (2018). DROM: Optimizing the routing in software-defined networks with deep reinforcement learning. IEEE Access, 6, 64533-64539.*

[18] *Glick, M., & Rastegarfar, H. (2017). Scheduling and control in hybrid data centers. In 2017 IEEE Photonics Society Summer Topical Meeting Series (SUM), pp. 115-116.*

[19] *Xu, J., Wang, J., Qi, Q., Sun, H., & He, B. (2018). Deep neural networks for application awareness in SDN-based network. In 2018 IEEE International Workshop on Machine Learning for Signal Processing (MLSP), pp. 1-6.*

[20] *David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin Riedmiller. Deterministic policy gradient algorithms. In Proceedings of the 31st International Conference on Machine Learning (ICML-14), pages 387–395, 2014.*

[21] *Richard S Sutton and Andrew G Barto. Reinforcement learning: An introduction. MIT press Cambridge, 1998.*

[22] *Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. arXiv preprint arXiv:1509.02971, 2015.*

[23] *Matthew Roughan. Simplifying the synthesis of internet traffic matrices. ACM SIGCOMM Computer Communication Review, 35(5):93–96, 2005.*

[24] *András Varga. Discrete event simulation system. In Proc. of the European Simulation Multiconference (ESM'2001), 2001.*

[25] *András Varga and Rudolf Hornig. An overview of the omnet++ simulation environment. In Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems & workshops, page 60. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2008.*

[26] *Universitat Politècnica de Catalunya. Knowledge-defined networking training datasets. URL http://knowledgedefinednetworking.org.*