

Network Management Research Group
Internet-Draft
Intended status: Informational
Expires: September 12, 2019

M-S. Kim
ETRI
Y-H. Han
KoreaTech
Y-G. Hong
ETRI
March 11, 2019

**Intelligent Reinforcement-learning-based Network Management
draft-kim-nmrg-rl-04**

Abstract

This document presents intelligent network management scenarios based on reinforcement-learning approaches. Nowadays, a heterogeneous network should usually provide real-time connectivity, the type of network management with the quality of real-time data, and transmission services generated by the operating system for an application service. With that reason intelligent management system is needed to support real-time connection and protection through efficient management of interfering network traffic for high-quality network data transmission in the both cloud and IoE network systems. Reinforcement-learning is one of the machine learning algorithms that can intelligently and autonomously provide to management systems over a communication network. Reinforcement-learning has developed and expanded with deep learning technique based on model-driven or data-driven technical approaches so that these trendy techniques have been widely to intelligently attempt an adaptive networking models with effective strategies in environmental disturbances over variety of networking areas.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 12, 2019.

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1.	Introduction	3
2.	Conventions and Terminology	3
3.	Theoretical Approaches	4
3.1.	Reinforcement-learning	4
3.2.	Deep-reinforcement-learning	4
3.3.	Advantage Actor Critic (A2C)	4
3.4.	Asynchronously Advantage Actor Critic (A3C)	5
4.	Reinforcement-learning-based process scenario	5
4.1.	Single-agent with Single-model	6
4.2.	Multi-agents Sharing Single-model	6
4.3.	Adversarial Self-Play with Single-model	6
4.4.	Cooperative Multi-agents with Multiple-models	6
4.5.	Competitive Multi-agents with Multiple-models	7
5.	Use Cases	7
5.1.	Intelligent Edge-computing for Traffic Control using Deep-reinforcement-learning	7
5.2.	Edge computing system in a field of Construction-site using Reinforcement-learning	7
5.3.	Deep-reinforcement-learning-based Cyber Physical Management Control system over a network	8
6.	IANA Considerations	9
7.	Security Considerations	9
8.	References	9
8.1.	Normative References	9
8.2.	Informative References	9
	Authors' Addresses	11

1. Introduction

Reinforcement-learning for intelligently autonomous network management, in general, is one of the challengeable methods in a dynamic complex and cluttered network environments. With the intelligent approach needs the development of computational systems in a single or large distributed networking nodes, where these environments involve limited and incomplete knowledge.

The reinforcement-learning can become a challenge-able and effective technique to transfer and share information via the global environment, as it does not require a priori-knowledge of the agent behavior or environment to accomplish its tasks [[Megherbi](#)]. Such a knowledge is usually acquired and learned repeatedly and autonomously by trial and error. The reinforcement-learning is also one of the machine learning techniques that will be adapted to the various networking environments for automatic networks [S.Jiang].

Deep-reinforcement-learning recently proposes has been extended from reinforcement-learning that can emerge as more powerful model-driven or data-driven model in a large state space, to overcome the classical behavior reinforcement-learning process. However, the classical reinforcement-learning slightly has a limitation to be adopted in networking areas, since the networking environments consist of significantly large and complex components in fields of routing configuration, optimization and system management, so that deep-reinforcement-learning can provide much more state information for learning process.[[MS](#)]

There are many different networking management problems to intelligently solve, such as connectivity, traffic management, fast Internet without latency and etc. Reinforcement-learning-based approaches can surely provide some of specific solutions with multiple cases against human operating capacities although it is a challengeable area due to a multitude of reasons such as large state space, complexity in the giving reward, difficulty in control actions, and difficulty in sharing and merging of the trained knowledge in a distributed memory node to be transferred over a communication network.[[MS](#)]

2. Conventions and Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [[RFC2119](#)].

3. Theoretical Approaches

3.1. Reinforcement-learning

Reinforcement-learning is an area of machine learning concerned with how software agents should take actions in an environment so as to maximize some notion of cumulative reward.[Wikipedia] The reinforcement-learning is normally used with a reward from centralized node (the global brain), and capable of autonomous acquirement and incorporation of knowledge. It is continuously self-improving and becoming more efficient as the learning process from an agent experience to optimize management performance for autonomous learning process.[[Sutton](#)][Madera]

3.2. Deep-reinforcement-learning

Some of advanced techniques using reinforcement-learning encounter and combine with deep-learning in neural networks that has made it possible to extract high-level features from raw data in computer vision [A Krizhevsky]. There are many challenges under the deep-learning models such as convolution neural network, recurrent neural network and etc., on the reinforcement-learning approach. The benefit of the deep learning applications is that lots of networking models, but the problematic issue is complex and cluttered networking structures used with large amounts of labelled training data.

Recently, the advances in training deep neural networks to develop a novel artificial agent, termed a deep Q-network (deep-reinforcement-learning network), can be used to learn successful policies directly from high-dimensional sensory inputs using end-to-end reinforcement learning [V.Mnih].

The deep-reinforcement-learning (deep Q-network) can provide more extended and powerful scenarios to build networking models with optimized action controls, huge system states and real-time-based reward function. Moreover, the technique has a significant advantage to set highly sequential data in a large model state space. [[MS](#)] In particular, the data distribution in reinforcement-learning is able to change as learning behaviors, that is a problem for deep learning approaches assumed by a fixed underlying distribution [V. Mnih].

3.3. Advantage Actor Critic (A2C)

Advantage Actor Critic is one of the intelligent reinforcement-learning models based on policy gradient model. The intelligent approach can optimize deep neural network controller in terms of reinforcement-learning algorithms, and show that parallel actor-learners have a stabilizing effect on training and they can be

allowing all of the methods to successfully train neural network controllers [Volodymyr Mnih]. Even if the prior deep-reinforcement-learning algorithm with experience replay memory tremendously has performance in challenging of the control service domains, it still needs to use more memory and computational power due to off-policy learning methods. To make up for this algorithms, a new algorithm has appeared.

The Advantage Actor Critic (consisting of actor and critic) method would implement generalized policy iteration alternating between a policy evaluation and a policy improvement step. Actor is a policy-based method that can improve the current policy for available the best next action. Critic in the value-based approach can evaluate the current policy and reduce the variance by a bootstrapping method. It is more stable and effective algorithm than the pure policy-based gradient methods.[\[MS\]](#)

[3.4.](#) Asynchronously Advantage Actor Critic (A3C)

Asynchronously Advantage Actor Critic is the updated algorithm based on Advantage Actor Critic. The main algorithm concept is to run multiple environments in parallel to run the agent asynchronously instead of experience replay. The parallel environment reduces the correlation of agent's data and induces each agent to experience various states so that the learning process can become a stationary process. This algorithm is a beneficial and practical point of view since it allows learning performance even with a general multi-core CPU. In addition, it can be applied to continuous space as well as discrete action space, and also has the advantages of learning both feedforward and recurrent agent.[\[MS\]](#)

A3C algorithm is possibly a number of complementary improvement to the neural network architecture and it has been shown to accurately produce and estimate of Q-values by including separate streams for the state value and advantage in the network to improve both value-based and policy-based methods by making it easier for the network to represent feature coordinates [Volodymyr Mnih].

[4.](#) Reinforcement-learning-based process scenario

With a single agent or multiple agents trained for intelligent network management, a variety of training scenarios are possible, depending on how agents are interacted and how many models are linked to the agents. The followings are possible RL training scenarios for network management.

4.1. Single-agent with Single-model

This is the traditional scenario of training a single agent who tries to achieve one goal related to network management. It receives all of information and rewards from a network (or a simulated network), and decides its appropriate action for the current network status.

4.2. Multi-agents Sharing Single-model

In this scenario, multiple agents share a single model and a single goal linked to the model. But, each of them is connected to an independent part of network or an independent whole network, so that they receive different information and rewards from such an independent one. The multiple agents experience differently on their connected networks. However, it does not mean their training behavior for network management will diverge. Each of their experience is used to train the single model. This scenario is a kind of parallelized version of the traditional 'Single-Agent with Single-Model' scenario, which can speed-up the RL training process and stabilize the single model's behavior.

4.3. Adversarial Self-Play with Single-model

This scenario contains two interacting agents with inverse reward functions linked to a single model. This scenario makes an agent have the perfectly matched opposing agent: itself, and trains the agent to become increasingly more skilled for network management. Inverse rewards are used to punish the opposing agent when an agent receives as positive reward, and vice versa. The two agents are linked to a single model for network management, and the model are trained and stabilized while both agents interact in a conflicting manner.

4.4. Cooperative Multi-agents with Multiple-models

In this scenario, two or more interacting agents share a common reward function linked to multiple different models for network management. In this scenario, a common goal is set up and all agents are trained to achieve the goal together that is hard to be achieved alone. Usually, each agent has access only to partial information of network status and determines an appropriate action by using its own model. Each of actions will be independently taken in order to accomplish a management task and collaboratively achieve the common goal.

4.5. Competitive Multi-agents with Multiple-models

This scenario contains two or more interacting agents with diverse reward function linked to multiple different models. In this scenario, agents will compete with one another to obtain some limited set of network resources and try to achieve their own goal. In a network, there will be tasks that have different management objectives. This leads multi-objective optimization problems, which are generally difficult to solve analytically. This scenario is suitable for solving such a multi-objective optimization problem related to network management by allowing each agent solve a single-objective problem, but complete with each other.

5. Use Cases

5.1. Intelligent Edge-computing for Traffic Control using Deep-reinforcement-learning

Edge computing is a concept that allows data from a variety of devices to be directly analyzed at the site or near the data, rather than being sent to a centralized data center such as the cloud. As such, edge computing will support data flow acceleration by processing data with low latency in real-time. In addition, by supporting efficient data processing on large amounts of data that can be processed around the source, and internet bandwidth usage will be also reduced.

Deep-reinforcement-learning would be useful technique to improve system performance in an intelligent edge-controlled service system for fast response time, reliability and security. Deep-reinforcement-learning is model-free approach so that many algorithms such as DQN, A2C and A3C can be adopted to resolve network problems in time-sensitive systems.

5.2. Edge computing system in a field of Construction-site using Reinforcement-learning

In a construction site, there are many dangerous elements such as noisy, gas leak and vibration needed by alerts, so that real-time monitoring system to detect the alerts using machine learning techniques can provide more effective solution and approach to recognize dangerous construction elements.

Representatively, to monitor these elements CCTV (closed-circuit television) should be locally and continuously broadcasting in a situation of construction site. At that time, it is in-effective and wasteful even if the CCTV is constantly broadcasting unchangeable scenes in high definition. However, the streaming should be

converted to high quality streaming data to rapidly show and detect the dangerous situation, when any alert should be detected due to the dangerous elements. To approach technically deep-reinforcement-learning can provide a solution to automatically detect these kinds of dangerous situations with prediction in an advance. It can also provide the transform data including with the high-rate streaming video and quickly prevent the other risks. Deep-reinforcement-learning is an important role to efficiently manage and monitor with the given dataset in real-time.

5.3. Deep-reinforcement-learning-based Cyber Physical Management Control system over a network

With the nonlinear control system such as cyber physical system provides an unstable system environment with initial control state due to its nonlinear nature. In order to stably control the unstable initial state, the prior-complex mathematical control methods (Linear Quadratic Regulator, Proportional Integral Differential) are used for successful control and management, but these approaches are needed with difficult mathematical process and high-rate effort. Therefore, using deep-reinforcement-learning can surely provide more effective technical approach without difficult initial set of control states to be compared with the other methods.

The ultimate purpose of the reinforcement-learning is to interact with the environment and maximize the target reward value. Observing the state in the step and the action by the policy are performed, and the reward judge a value through the compensation given in the environment. Deep-reinforcement-learning using Convolutional Neural Network (CNN) can provide more performing learning process to make stable control and management.

As part of the system, it shows how the physical environment and the cyber environment interact with the reinforcement-learning module over a network. The actions to control the physical environment, delivered to the Enhanced Learning model based on DQN, transfer to data to the physical environment using networking communication tools as below.

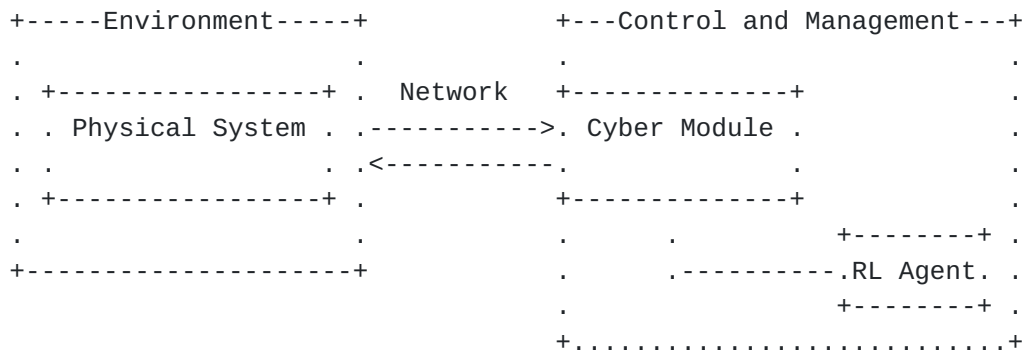


Figure 1: DRL-based Cyber Physical Management Control System

6. IANA Considerations

There are no IANA considerations related to this document.

7. Security Considerations

[TBD]

8. References

8.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.

8.2. Informative References

[I-D.jiang-nmlrg-network-machine-learning]
Jiang, S., "Network Machine Learning", ID [draft-jiang-nmlrg-network-machine-learning-02](#), October 2016.

[Megherbi] "Megherbi, D. B., Kim, Minsuk, Madera, Manual., A Study of Collaborative Distributed Multi-Goal and Multi-agent based Systems for Large Critical Key Infrastructures and Resources (CKIR) Dynamic Monitoring and Surveillance, IEEE International Conference on Technologies for Homeland Security", 2013.

[Teiralbar]

"Megherbi, D. B., Teiralbar, A. Boulenouar, J., A Time-varying Environment Machine Learning Technique for Autonomous Agent Shortest Path Planning, Proceedings of SPIE International Conference on Signal and Image Processing, Orlando, Florida", 2001.

[Nasim]

"Nasim ArianpooEmail, Victor C.M. Leung, How network monitoring and reinforcement learning can improve tcp fairness in wireless multi-hop networks, EURASIP Journal on Wireless Communications and Networking", 2016.

[Minsuk]

"Dalila B. Megherbi and Minsuk Kim, A Hybrid P2P and Master-Slave Cooperative Distributed Multi-Agent Reinforcement Learning System with Asynchronously Triggered Exploratory Trials and Clutter-index-based Selected Sub goals, IEEE CIG Conference", 2016.

[April]

"April Yu, Raphael Palefsky-Smith, Rishi Bedi, Deep Reinforcement Learning for Simulated Autonomous Vehicle Control, Stanford University", 2016.

[Markus]

"Markus Kuderer, Shilpa Gulati, Wolfram Burgard, Learning Driving Styles for Autonomous Vehicles from Demonstration, Robotics and Automation (ICRA)", 2015.

[Ann]

"Ann Nowe, Peter Vrancx, Yann De Hauwere, Game Theory and Multi-agent Reinforcement Learning, In book: Reinforcement Learning: State of the Art, Edition: Adaptation, Learning, and Optimization Volume 12", 2012.

[Kok-Lim]

"Kok-Lim Alvin Yau, Hock Guan Goh, David Chieng, Kae Hsiang Kwong, Application of Reinforcement Learning to wireless sensor networks: models and algorithms, Published in Journal Computing archive Volume 97 Issue 11, Pages 1045-1075", November 2015.

[Sutton]

"Sutton, R. S., Barto, A. G., Reinforcement Learning: an Introduction, MIT Press", 1998.

[Madera]

"Madera, M., Megherbi, D. B., An Interconnected Dynamical System Composed of Dynamics-based Reinforcement Learning Agents in a Distributed Environment: A Case Study, Proceedings IEEE International Conference on Computational Intelligence for Measurement Systems and Applications, Italy", 2012.

[Al-Dayaa]

"Al-Dayaa, H. S., Megherbi, D. B., Towards A Multiple-Lookahead-Levels Reinforcement-Learning Technique and Its Implementation in Integrated Circuits, Journal of Artificial Intelligence, Journal of Supercomputing. Vol. 62, issue 1, pp. 588-61", 2012.

[Chowdappa]

"Chowdappa, Aswini., Skjellum, Anthony., Doss, Nathan, Thread-Safe Message Passing with P4 and MPI, Technical Report TR-CS-941025, Computer Science Department and NSF Engineering Research Center, Mississippi State University", 1994.

[Mnih]

"V.Mnih and et al., Human-level Control Through Deep Reinforcement Learning, Nature 518.7540", 2015.

[Stampa]

"G Stamp, M Arias, etc., A Deep-reinforcement Learning Approach for Software-defined Networking Routing Optimization, cs.NI", 2017.

[Krizhevsky]

"A Krizhevsky, I Sutskever, and G Hinton, Imagenet classification with deep convolutional neural networks, In Advances in Neural Information Processing Systems, 1106-1114", 2012.

[Volodymyr]

"Volodymyr Mnih and et al., Asynchronous Methods for Deep Reinforcement Learning, ICML, arXiv:1602.01783", 2016.

[MS]

"Intelligent Network Management using Reinforcement-learning, [draft-kim-nmrg-rl-03](#)", 2018.

[Ju-Bong]

"Deep Q-Network Based Rotary Inverted Pendulum System and Its Monitoring on the EdgeX Platform, International Conference on Artificial Intelligence in Information and Communication (ICAIIIC)", 2019.

Authors' Addresses

Min-Suk Kim
Etri
161 Gajeong-Dong Yuseung-Gu
Daejeon 305-700
Korea

Phone: +82 42 860 5930
Email: mskim16@etri.re.kr

Youn-Hee Han
KoreaTech
Byeongcheon-myeon Gajeon-ri, Dongnam-gu
Choenan-si, Chungcheongnam-do
330-708
Korea

Phone: +82 41 560 1486
Email: yhhan@koreatech.ac.kr

Yong-Geun Hong
ETRI
161 Gajeong-Dong Yuseung-Gu
Daejeon 305-700
Korea

Phone: +82 42 860 6557
Email: yghong@etri.re.kr

