Network Management Research Group
Internet-Draft
Intended status: Informational
Expires: January 3, 2019

M-S. Kim
Y-G. Hong
ETRI
Y-H. Han
KoreaTech
T-J. Ahn
KT
K-H. Kim
ETRI

July 2, 2018

Intelligent Network Management using Reinforcement Learning
draft-kim-nmrg-rl-03

Abstract

This document describes intelligent network management system to autonomously manage and monitor using machine learning techniques. Reinforcement learning is one of the machine learning techniques that can provide autonomously management with multi-agent path-planning over a communication network. According to intelligent distributed multi-agent system, the main centralized node called by the global environment should not only manage all agents workflow in a hybrid peer-to-peer networking architecture and, but transfer and share information in distributed nodes. All agents in distributed nodes are able to be provided with a cumulative reward for each action that a given agent takes with respect to an optimized knowledge based on a to-be-learned policy over the learning process. The optimized and trained knowledge would be involved with a large state information by the control action over a network. A reward from the global environment is reflected to the next optimized control action autonomously for network management in distributed networking nodes. The Reinforcement Learning(RL) Process have developed and expanded to Deep Reinforcement Learning(DRL) with model-driven or data-driven technical approaches for learning process. The trendy technique has been widely to attempt and apply to networking fields since Deep Reinforcement Learning can be used in practical networking areas beyond dynamics and heterogeneous environment disturbances, so that in the technique can be intelligently learned in the effective strategy.

Status of This Memo

This Internet-Draft will expire on January 3, 2019.

Copyright Notice

Table of Contents

1. Introduction

In large infrastructures such as transportation, health and energy systems, collaborative monitoring system is needed, where there are special needs for intelligent distributed networking systems with learning schemes. Agent reinforcement learning for intelligently autonomous network management, in general, is one of the challengeable methods in a dynamic complex cluttered environment over a network. It also needs the development of computational multi-agents learning systems in large distributed networking nodes, where the agents have limited and incomplete knowledge, and they only access local information in distributed networking nodes.

Reinforcement Learning can become an effective technique to transfer and share information among agents via the global environment (centralized node), 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 automatically and autonomously by trial and error.

Reinforcement Learning is one of the machine Learning techniques that will be adapted to the various networking environments for automatic networks[S. Jiang]. Thus, this document provides motivation, learning technique, and use case for network machine learning.

Deep reinforcement learning recently proposes that the extended reinforcement Learning algorithm could emerge as more powerful model-driven or data-driven techniques over a large state space to overcome the classical behavior reinforcement Learning process. The deep reinforcement learning technique has been significantly shown as successful models in playing Atari games [V. Mnih]. The deep reinforcement learning provides more effective experimental system performance in a complex and cluttered networking environment.

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.

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. Motivation

3.1. General Motivation for Reinforcement Learning

Reinforcement learning is a system capable of autonomous acquirement and incorporation of knowledge. It can continuously self-improve learning process with experience and attempts to maximize cumulative reward to manage an optimized learning knowledge by multi-agents-based monitoring systems[Teiralbar]. The maximized reward can be increasingly optimizing of learning speed for agent autonomous learning process.

3.2. Reinforcement Learning in networks

Reinforcement learning is an emerging technology in terms of monitoring network system to achieve fair resource allocation for nodes within the wire or wireless mesh setting. Monitoring parameters of the network and adjusts based on the network dynamics can demonstrate to improve fairness in wireless environment Infrastructures and Resources[Nasim].

3.3. Deep Reinforcement Learning in networks

Deep reinforcement learning is a large state model-driven or data-driven approach on an intelligently learning strategy. The intelligent technique represents learning models successfully to

train knowledge for control policy directly from high-dimensional sensory input using reinforcement learning with Q-value function in a convolutional neural network [Mnih]. The model repeatedly estimates reward using the defined reward function depending on the current states, to acquire more effective and optimized control action in following next steps. The deep reinforcement learning can be widely-adopted in routing optimization to attempt minimizing the network delay [Stampa].

3.4. Motivation in our work

There are many different networking management problems to intelligently solve, such as connectivity, traffic management, fast internet without latency and etc. We expect that machine-learning-based mechanism such as reinforcement learning will provide network solutions with multiple cases against human operating capacities even if 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 between agents in a distributed memory node to be transferred over a communication network.[Minsuk]

4. Related Works

4.1. Autonomous Driving System

Recently, 5G network and AI are new trend and future research areas, so that a lot of business models have been developed and appeared in the networking fields. Autonomous vehicle has been simultaneously developed with 5G and AI. Autonomous vehicle is capable of self-automotive driving without human supervision depending on optimized trust region policy by reinforcement learning that enables learning of more complex and special network management environment. Such a vehicle provides a comfortable user experience safely and reliably on interactive communication network [April] [Markus].

4.2. Network Defect Prediction

Nowadays, the networking equipment handles a variety of services such as Internet, IPTV, VoIP in a single device. As the performance of the equipment improves, even if there is an advantage to construct the equipment to be separately constructed in a single device, the probability of the service failure of network equipment might be increasing. For that reason, the equipment failure risk over a network poses a major networking carriers, so that there is growing need to prevent disturbances by detecting network failure in advance. Machine learning such as deep learning or reinforcement learning emerged the preferred solutions to manage and monitor the networking

equipment (LTE core, router and switch) prevented by the networking failure risk.

4.3. Wireless Sensor Network (WSN)

Wireless sensor network (WSN) consists of a large number of sensors and sink nodes for monitoring systems with event parameters such as temperature, humidity, air conditioning, etc. Reinforcement learning in WSNs has been applied in a wide range of schemes such as cooperative communication, routing and rate control. The sensors and sink nodes are able to observe and carry out optimal actions on their respective operating environment for network and application performance enhancements[Kok-Lim].

4.4. Routing Enhancement

Reinforcement learning is used to enhance multicast routing protocol in wireless ad hoc networks, where each node has different capability. Routers in the multicast routing protocol are determined to discover optimal route with a predicted reward, and then the routers create the optimal path with multicast transmissions to reduce the overhead in reinforcement learning[Kok-Lim].

4.5. Routing Optimization

Routing optimization as traffic engineering is one of the important issues to control the behavior of transmitted data in order to maximize the performance of network [Stampa]. There are several attempts to be adopted with machine learning algorithms in the context of routing optimization. Deep reinforcement learning is recently one of solutions for unseen network states that cannot be achieved by traditional table-based reinforcement learning agent [Stampa]. Deep reinforcement learning can provide more improvement to optimal control routing configuration by given-agent on complex networking.

4.6. Game Theory

The adaptive multi-agent system, which is combined with complexities from interacting game player, has developed in a field of reinforcement learning. In the early game theory, the interdisciplinary work was only focused on competitive games, but reinforcement learning has developed into a general framework for analyzing strategic interaction and has been attracted field as diverse as psychology, economics and biology.[Ann] AlphaGo is also one of the game theories using reinforcement learning, developed by Google DeepMind. Even though it began as a small learning computational program with some simple actions, it has now trained on

a policy and value networks of thirty million actions, states and rewards.

5. Intelligent Machine Learning Technologies

5.1. Reinforcement Learning (RL)

Agent reinforcement learning is machine-learning-based unsupervised algorithms based on an agent learning process. Reinforcement learning is normally used with a reward from centralized node (the global environment), 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]

5.2. Deep Learning (DL)

The rule-based network equipment failure for judgment/prediction should have been described as a correct rule for equipment or case, and continuously updated when a new failure pattern occurs. Deep Learning (DL) techniques such as Convolution Neural Network(CNN), and Recurrent Neural Network(RNN) can be adapted to learn new patterns occurred by the networking faults. We are able to judge and predict a fault condition in these models. The deep learning models has advantages in terms of maintenance and expandability, since it can automatically learn features under the patterns without needing to describe the detailed rules.

5.3. Deep Reinforcement Learning (DRL)

Nowadays, some of advanced techniques using reinforcement learning encounter and combine to deep learning technique in Neural Network(NN) that has made it possible to extract high-level features from raw data in compute 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, which have problematic issue due to complex and cluttered networking structure, can be used with large amounts of labelled training data.

Recently, 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. 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 [Mnih].

5.4. 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 though 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.

5.5. 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.

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].

5.6. Policy using Distance and Frequency

Distance and Frequency algorithm uses the state occurrence frequency in addition to the distance to goal. It avoids deadlocks and lets the agent escape the Dead, and it was derived to enhance agent optimal learning speed. Distance-and-Frequency is based on more levels of agent visibility to enhance learning algorithm by an additional way that uses the state occurrence frequency.[Al-Dayaa]

5.7. Distributed Computing Node

Autonomous multi-agent learning process for network management environment is related to transfer optimized knowledge between agents on a given local node or distributed memory nodes over a communication network.

5.8. Agent Sharing Information

This is a technique how agents can share information for optimal learning process. The quality of agent decision making often depends on the willingness of agents to share a given learning information collected by agent learning process. Sharing Information means that an agent would share and communicate the knowledge learned and acquired with or to other agents using RL.

Agents normally have limited resources and incomplete knowledge during learning exploration. For that reason, the agents should take actions and transfer the states to the global environment under RL, then it would share the information with other agents, where all agents explore to reach their goals via a distributed reinforcement reward-based learning method on the existing local distributed memory nodes.

MPI (Message Passing Interface) is used for communication way. Even if the agents do not share the capabilities and resources to monitor an entire given large terrain environment, they are able to share the needed information to manage collaborative learning process for optimized management in distributed networking nodes.[Chowdappa][Minsuk]

6. Proposed Architecture

6.1. Architecture for Reinforcement Learning

The architecture using reinforcement learning describes a collaborative multi-agent-based system in distributed environments as shown in figure 1, where the architecture is combined with a hybrid architecture making use of both a master and slave architecture and a peer-to-peer. The centralized node(global environment), assigns each slave computing node a portion of the distributed terrain and an initial number of agents.

```
         +-------------+
         |             |                       +-----------------+
         |             |<...... node 1 .......>|    terrain 1    |
         |             |                       +-----------------+
         | Global env. |                           +        |
         |  (node 0)   |                           |        |
         |             |                           |        +
         |             |                       +-----------------+
         |             |<...... node 2 .......>|    terrain 2    |
         |             |                       +-----------------+
         +-------------+
```

Figure 1: Hybrid P2P and Master/Slave Architecture Overview

Reinforcement Learning (RL) actions involve interacting with a given environment, so the environment provides an agent learning process with the elements as followings:

- Agent control actions, large states and cumulative rewards

- Initial data-set in memory

- Random or learning process in a given node

- Next, optimamization in neural network under reinforcement learning

Additionally, agent actions with states toward its goal as below:

- Agent continuously control actions to earn next optimized state based on its policy with reward

- After an agent reaches its goal, it can repeatedly collect the information collected by the random or learning process to next learning process for optimal management

- o Agent learning process is optimized in the following phase and exploratory learning trials

6.2. Architecture for Deep Reinforcement Learning

In shown as Figure2, we illustrate the fundamental architecture for relationship of an action, state and reward, and each agent explores to reach its goal(s) under deep reinforcement learning. The agent takes an action that leads to a reward from achieving an optimal path toward its goal.

```
                            DRL Network
                            +----------------------------------+
                            |Q-Value1|                         |
                            |--------+    +-------+    +------+|
          ......Action......|Q-Value2|----|Network|----|States||<...
          .                 |--------+    +-------+    +------+|   .
          .                 |Q-Value3|                         |   .
          .                 +----------------------------------+   .
          .                                                        .
        +---------+----------+                                     .
        | Global Environment |                                     .
        +---------+----------+                                     .
          .                                                        .
          .                                                        .
          .           +-------------------+                +----------+
          ...........>+ Large State Space +....States.....>+ D-Memory +
                      +-------------------+                +----------+
```

Figure 2: DRL work-flow Overview

Deep Reinforcement Learning network can provide a convolutional neural network to overcome the problematic issues of reinforcement Learning for successfully learning control policy from raw data in a complex environment. It is also used with an experience replay memory that randomly samples previous transitions, and thereby smooths the training distribution over many past behaviors [V. Mnih].

7. Use case of Reinforcement Learning

7.1. Distributed Multi-agent Reinforcement Learning (RL): Sharing Information Technique

In this section, we deal with case of a collaborative distributed multi-agent, where each agent has same or different individual goals in a distributed environment. Since sharing information scheme among the agents is problematic one, we need to expand on the work described by solving the challenging cases.

Basically, the main proposed algorithm is presented by distributed multi-agent RL as below:

```
+--------------------------------------------------------------------+
| Proposed Algorithm                                                 |
+--------------------------------------------------------------------+
| (1) Let Ni denote the number of node (i= 1, 2, 3 ...)              |
|                                                                    |
| (2) Let Aj denote the number of agent                              |
|                                                                    |
| (3) Let Dk denote the number of goals                              |
|                                                                    |
| (4) Place initial number of agents Aj, in random position (Xm,     |
| Yn)                                                                |
|                                                                    |
| (5) Initialization of data-set memory for neural network           |
|                                                                    |
| (6) Copy neutal network Q and store as the data-set memory         |
|                                                                    |
| (7) Every Aj in Ni                                                 |
|                                                                    |
| -----> (a) Do initial exploration (random) to corresponding Dk     |
|                                                                    |
| -----> (b) Do exploration (using RL) for Tx denote the number of   |
| trial                                                              |
+--------------------------------------------------------------------+
```

Table 1: Proposed Algorithm

Random Trial
(1) Let Si denote the the current state
(2) Relinquish Si so that the other agent can occupy the position
(3) Assign the agent new position
(4) Update the current state Si -> Si+1

Table 2: Random Trial

Optimal Trial
(1) Let Si denote the the current state
(2) Let ACj denote a contorl action
(3) Let DRm denote discount reward
(4) Choose ACj <- Policy(Si, ACj) in neural network
(5) Update and copy the network for learning process in the global environment
(6) Update the current state Si < Si+1-
(7) Repeat a available network control action

Table 3: Optimal Trial

Multi-agent reinforcement learning in distributed nodes can improve the overall system performance to transfer or share information from one node to another node in following cases; expanded complexity in RL technique with various experimental factors and conditions, analyzing multi-agent sharing information for agent learning process.

7.2. Intelligent Edge Computing technique 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.

7.3. Edge computing system in a field of construction works using Reinforce 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 (DL, RL) 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, when any alert should be detected due to the dangerous elements, the streaming should be converted to high quality streaming data to rapidly show and defect the dangerous situation. To approach technically, DL is one of the solutions to automatically detect these kinds of dangerous situations with prediction in an advance. It can provide the transform data including with the high-rate streaming video and quickly prevent the other risks. RL is additionally important role to efficiently manage and monitor with the given dataset in real time.

[TBD]

7.4. Fault prediction for core-network using Deep Learning

EPC equipment such as PGW, SGW, MME, HSS and PCRF in the LTE core network send/receive messages using interfaces based on the 3GPP standard specification. These EPC equipment could create training data and model to predict/detect features of the precursor symptoms occurring before the networking failure when a specific equipment and LTE network service failures are discovered. In the addition, Deep Learning (DL) can predict various network faults such as in/out traffic, resource information of CPU/Memory and QoS performance in the case of IP core network equipment.

[TBD]

8. IANA Considerations

There are no IANA considerations related to this document.

9. Security Considerations

[TBD]

10. References

10.1. 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>.

10.2. 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 con- volutional 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.

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

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

Phone: +82 42 860 6557
Email: yghong@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

Tae-Jin Ahn
Korea Telecom
70 Yuseong-daero 1689 Beon-gil Yuseung-Gu
Daejeon 305-811
Korea

Phone: +82 42 870 8409
Email: Taejin.ahn@kt.com

Kwi-Hoon Kim
ETRI
161 Gajeong-Dong Yuseung-Gu
Daejeon 305-700
Korea

Phone: +82 42 860 6746
Email: kwihooi@etri.re.kr