Mitigating TCP Congestion: A Coordinated Cyber and Physical Approach

Zeyu Mao, *Student Member, IEEE,* Abhijeet Sahu, *Student Member, IEEE,*
Patrick Wlazlo, *Student Member, IEEE,* Yijing Liu, *Student Member, IEEE,* Ana Goulart, *Member, IEEE,*
Katherine Davis, *Senior Member, IEEE,* and Thomas J. Overbye, *Fellow, IEEE*

***Abstract*—The operation of the modern power grid is becoming increasingly reliant on its underlying communication network, especially within the context of the rapidly growing integration of Distributed Energy Resources (DERs). This tight cyber-physical coupling brings uncertainties and challenges for the power grid operation and control. To help operators manage the complex cyber-physical environment, ensure the integrity, and continuity of reliable grid operation, a two-stage approach is proposed that is compatible with current ICS protocols to improve the deliverability of time critical operations. With the proposed framework, the impact Denial of Service (DoS) attack can have on a Transmission Control Protocol (TCP) session could be effectively prevented and mitigated. This coordinated approach combines the efficiency of congestion window reconfiguration and the applicability of physical-only mitigation approaches. By expanding the state and action space to encompass both the cyber and physical domains. This approach has been proven to outperform the traditional, physical-only method, in multiple network congested scenarios that were emulated in a real-time cyber-physical testbed.**

***Index Terms*—Cyber-physical system (CPS), TCP Congestion, Industrial Control System, DNP3, Reinforcement Learning**

2021 North American Power Symposium (NAPS) | 978-1-6654-2081-5/21/$31.00 ©2021 IEEE | DOI: 10.1109/NAPS52732.2021.9654541

I. Introduction

The increasing complexity of large interconnected power systems makes it ever-more difficult to ensure normal grid operation. Automatic controls, e.g. automatic generation control, remedial action scheme, have been deployed to help operators maintain the power grid's reliability and security. The grid's underlying communication infrastructure enables timely real-time monitoring and actuation of these controls. The concern is that the addition of these infrastructure components brings with it additional challenges, such as network congestion, cyber intrusions, and increased misconfigurations, that can prevent timely actuation of operations. Hence, traditional controller and ICS protocols may not be sufficient to still maintain the timeliness requirement, especially for more remote operations.

Various work on automatic control in power systems under a normal cyber environment have been conducted, such as market decision [1], transient angle stability [2], Automatic Generation Control (AGC) [3], Automatic Voltage Regulator (AVR) [4], microgrid controls [5], demand control [6], etc. Similarly, other publications that discuss cyber-based incidence response have explored RL based solutions such as automated incident handling against network-based attack [7]. In 2018, an OpenAI Gym integration into NS-3 [8] was developed which was then used to train a communication network controller to take action such as controlling the TCP congestion window and various wireless communication parameters, such as Signal to Noise Ration (SNR) thresholds. The question that naturally stems from combining the two prior research groups works is; is it possible to combine studies from both sides to guarantee autonomous controls' that are resilient enough for both complex cyber and complex physical environments?

In order to mitigate the impact of an adversarial event in a cyber-physical power system, most previous works either focus on only physical state and action space, taking actions only on the available physical devices, or reconfiguring only the cyber networks, such as by adjusting the routing path. Based on the intensity of the attack, the effectiveness and efficiency of these methods may vary. In addition, the cyber approaches tend to be easier and less time-consuming to deploy, while the physical approaches are generally more tolerant to the application condition. In this paper, a coordinated approach is proposed to effectively and efficiently mitigate the TCP congestion-related events. This two-stage approach will adjust the congestion window size first, to mitigate the impact of TCP congestion in a cyber based efficient and effective technique, then to give alternative control suggestions to the operator if congestion window adjustment is not sufficient. The main contributions of this paper are:

1) A federated simulation environment for data-driven cyber security applications in power systems is proposed.
2) The proposed method expands the state and action space to cyber and physical domain to mitigate the DoS/DDoS events. Both congestion window reconfiguration (cyber action) and providing alternative control suggestions (physical action) are supported.
3) The process to integrate the approach into DNP3 and other TCP-based ICS protocol is presented.
4) The proposed framework is validated in a real-time interactive cyber-physical testbed and outperform the physical-only method on efficiency.

The rest of the paper is organized as follows: Section II introduces the command flow of wide-area monitoring and controls, and the issues of TCP congestion on ICS protocols. Section III systematically presents the typical procedure of the remote control command delivery process and the process for integrating the proposed response engine. In section IV, we present the design of the training and validation framework for the proposed method. Section V discusses the results from the WSCC 9 bus cyber-physical model on the proposed method and the baseline method. Finally, Section VI summarizes the contributions of this paper.

II. Problem Description of the TCP Congestion

A. Wide-area Monitor and Control (WAMC): from Source to Destination

In the modern Energy Management Systems (EMS), when an operator issues a remote command through the human-machine interface (HMI) in the control center, the control command will be sent to the connected Supervisory Control And Data Acquisition (SCADA) system, where depending on the ICS protocol that SCADA system is using, the command will be encoded and then transmitted through the utility communication network to its target devices. For example, Fig. 1 shows how Distributed Network Protocol 3 (DNP3) master and outstation communication through the utility network.

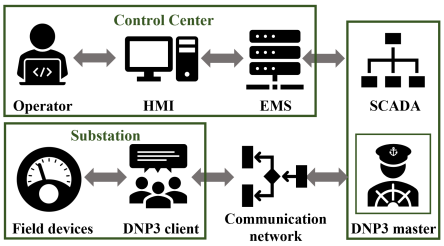

Fig. 1. Typical Information Flow for Wide-Area Monitor and Control

There are situations where commands from control center could not reach the remote terminal unit (RTU), for various reasons. Most protocols handle these situations by number of re-attempts, as shown in Algorithm. 1, where DNP3 master will return a failure response if the number of retry exceeds a pre-defined value. Other protocols may return an undecided indicator and wait for a response from their target client.

An optimal maximum number of attempts is hard to determine and normally it is viewed as an engineering problem. If the value is too low, the communication under weak connection tends to have too many failures. If the value is too high, then it will spend much longer time to determine whether the command delivery is successful, which can then affect the timeliness of the system functions. Besides, these ICS protocols do not provide enough feedback to operators about the communication failure; this is largely due to the fact that in the traditional view of control system, IT and OT are separate domains. During the occurrence of an adversarial cyber event, the current mechanism provides limited information to operators, which affects their decisions and may cause relative long time to recover.

B. TCP Congestion

Congestion of TCP might come from a variety of sources. The most common are from a denial of service (DoS) or distributed denial of service attack (DDoS) attacks, that burden intermediate routers and services with a large quantity of inordinately long packets, or the packets travel along paths with limited bandwidth. In both DoS/DDoS attacks, a volume of ICMP, UDP, or TCP packets are sent to a router or server in short succession causing them to drop both DoS/DDoS and legitimate traffic. It can also happen when a router along the path a packet is traveling is busy servicing other UDP or TCP connections. Or this could be the result of a low-bandwidth connection only being able to service a limited number of TCP sessions. Low-bandwidth communications, like microwave, are still a significant portion of the total communication media (e.g. power line carrier (PLC), fiber optic cable, or low-capacity radio) in the modern power system communication network [9]. The congestion control algorithms purpose is to either to prevent the congestion events listed above or to remove congestion after it has happened. The amount of traffic a sender must send depends on two windows: receiver window and the sender's congestion window. TCP congestion window limits the amount of data a sender can inject into the network.

The potential impact of the TCP congestion can result in ICS protocols, like DNP3, ICCP, MODBUS, or IEC61850, being delayed in transmission or dropped. In order to ensure the timeliness, most of these protocols have a timeout mechanism. Once the acknowledgement period exceeds the KeepAlive timeout interval, the connection will be be labeled as "collapsed", thus the operational commands cannot not be delivered - which might severely impact the power system operation.

III. Coordinated Cyber Physical Response Engine targeting TCP Congestion

To improve the operational resilience against TCP congestion, a two-stage response framework was developed to fortify the power grid against cyber attacks and to help mitigate their impact. The response engine is designed to be compatible with the TCP-based ICS protocols. There are two triggering conditions for the engine, as shown in Algorithm. 2. One is the alert from IDS indicating any intrusion or compromise. In this situation, the WAMC communication may still be uninterrupted, thus preventive action could be taken to improve the communication resilience. The other triggering condition is the timeout signal from the client in the connection stage or during the acknowledgement period. Once the triggering condition is met, the cyber-physical response engine determines the congestion intensity from the monitored cyber data, and then determine whether the congestion could be fixed by adjusting the congestion window size. If it is determined to be unsolvable, then it will provide suggestions on available alternative controls which would bypass the affected area. The network congestion caused due to DoS can be escalated by other ways such as isolating the affected zone, packet re-routing, or manual operation. The current approach prioritizes to regulate the congestion window over other approach for faster response.

The response engine requires the controllability over TCP sessions, thus it could be used with TCP-based ICS protocols. Algorithm. 3 shows how the framework could be integrated into the EMS and work with other components. The response engine analyzes the real-time cyber data and the topology information, and then based on the congestion intensity, it either gives the optimal settings of congestion window to the TCP client (e.g. DNP3 master and clients), or it provides area congestion information to help determine the alternative controls that not involve the affected components. The data from the state estimator would be fed into the response engine to support its model to generate alternative control suggestions.

Algorithm 1 Typical remote control command delivery process (DNP3 as example).

CF(Connected_Flag) := False, *RC*(Retry_Count) := 0, *REM*(Retry_Exceeds_Maximum) := False
The operator issues a remote command through the EMS human-machine interface
The integrated DNP3 master encodes the command with IEEE P1815 protocol header
if DNP3 master is running in the polling mode **then**
    Connect to the target DNP3 client
else
    if DNP3 master in running in the Multi-connection mode **then**
        Check whether the connection time exceeds the KeepAlive threshold. If yes, re-connect to the DNP3 client
    end if
end if
while !(*CF* | *REM*) **do**
    Attempt to connect to the target client
    if Connection time > timeout duration **then**
        RC = *RC* + 1
    else
        CF = True
    end if
end while
Once the connection between DNP3 master and client is established, the command packet will be sent to the target
The target will send back an acknowledgement to confirm the packet is received
if an acknowledgement is received after the configured timeout **then**
    the command is discarded
end if

IV. Implementation and Validation of the Response Engine in a Cyber Physical Testbed

The problem this work helps to solve can be divided into two sub-problems: the optimal congestion window size searching could be viewed as a classical control problem, and the alternative physical action could be formulated as a Markov Decision Process (MDP) problem with cyber and physical constraints. Timeliness is a major concern of ICS systems. Hence, a Deep Reinforcement Learning (DRL)-based approach is used to meet the efficiency and the effectiveness requirement for real-time reliable operation.

As shown in Fig. 2, a combined cyber and physical simulation environment is created in the RESLab testbed [10] to interact with the DRL agent. For the cyber system simulation, NS-3 Gym is used, along with the ESA [11] Python package that interacts with PowerWorld for power system simulation. The NS-3 Gym environment is developed to provide a platform to train the RL agent for optimal decision making in networking and communication systems such as deciding channel allocation schemes, TCP congestion control, backoff and retransmission for congestion avoidance, etc. It makes use of Google *protobuf* to allow the environment to interact with the NS-3 simulation. This environment is used to both provide the communication network parameters and states, and to execute actions on adjusting network configurations.

A real-time interactive cyber physical environment is developed to test and validate the response engine. As shown in Fig. 3, the communication network is emulated in the Common Open Research Emulator (CORE). A DNP3 master has been integrated in the control center emulator, which will send remote control commands through the emulated network to the DNP3 client in the DS [12, 13]. The interactivity from

Algorithm 2 Response Engine Triggering Logic

Inputs: IDS log **AND** DNP3 timeout (connection and acknowledgement)
Initialize the response vector V
Parse the latest alert from IDS
if the alert indicates any compromise or attack on component u **then**
    $V_u = 1$
end if
if the device w connection time exceeds the timeout **OR** the response time exceeds the acknowledgement timeout **then**
    $V_w = 1$
end if

Algorithm 3 TCP congestion prevention & mitigation process

if $||V|| > 0$ **then**
    Find all the affected DNP3 clients
    Modify the TCP congestion control parameters based on the result from the response engine for both DNP3 master and clients
    Re-send the remote command if applicable
    if $||V|| > 0$ **then**
        Notify the operator communication issues and provide alternative control suggestions based on the result from the response engine
    end if
end if

DS provides a real-time controllability over generators, loads, shunts, branches, etc. [14, 15], while CORE provides on-the-fly network parameter adjustment capability. Hping3 is used to emulate the DoS attack with varying intensity and different packet size. The response engine interacts with the environment, adjusts the congestion window and alerts the operator with alternative controls when necessary.

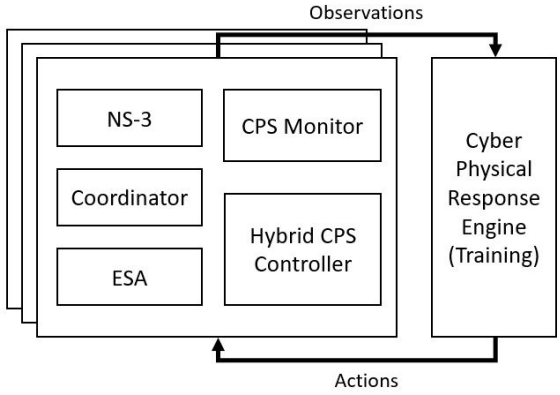

Fig. 2. Proposed Training Framework

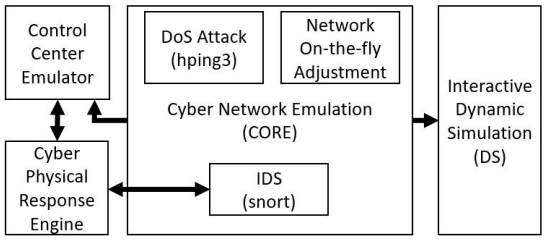

Fig. 3. Proposed Validation Framework

There are many states that can be used or tracked in NS-3 Gym. The variables that are being tracked to mitigate the TCP congestion are listed in Table I. The *AckInter* or acknowledgement packet interval is the time it takes between two sequential DNP3 response packets to arrive at the DNP3

client. The *RTT* or round trip time is the time it takes for a operate packet from the client to be sent to the server, plus the time it takes for the server to send a response packet back to the client. The *THR* is the maximum bandwidth threshold of a link between the client and sever. The *SST* is the slow start threshold that gradually increased in order to find the maximum throughput of a link, while the *CWND* or congestion window is the maximum amount of TCP packets that have been successfully transmitted.

TABLE I
VARIABLES BEING TRACKED IN CYBER SYSTEMS

Observable	Controllable
AckInter (Acknowledgement Packet Interval)	CWND
RTT (Round Trip Time)	N/A
THR (Throughput of Link)	N/A
SST (Slow Start Threshold)	N/A
CWND (Current Congestion Window)	N/A

TABLE II
VARIABLES BEING TRACKED IN POWER SYSTEM

Observable	Controllable
Power Network Topology (transmission line and transformer status)	Breaker
Bus voltage	Shunt
Generator reactive power output	Generator reactive power setpoint

Unlike existing penetration tools like Nessus, metasploit, and other tools in Kali Linux toolkit, NS-3 does not provide a platform for simulating intrusions. A pseudo intrusion scenario is created which mimics the DoS attack by running high-volume background traffic between pairs of nodes to create congestion for some bandwidth-limited links in the network. After that, the autonomous agent will decide either to decrease its congestion window to control the intended physical device or switch to another device.

The purpose of the reinforcement learning agent is to find the optimal *CWND* state that will send the maximum number of TCP packets through a given link. To do so, the agent can either decrease or increase the congestion window by different amounts to interact with the environment. After training, the agent tends to choose the action which maximizes the accumulated rewards.

V. CASE STUDY

As shown in Fig. 4, a three-substation network is created based on the WSCC 9 bus case [16], which consists of 4 broadcast domains, one each for the substation and one for the main control center. The congestion will be emulated in the four links of the network simulation. Using Monte Carlo simulation, 10,000 scenarios are created with different load profiles, generation profiles, and TCP congestion scenarios. These scenarios are randomly split in a 80-20 ratio for training and testing purposes.

A. Cyber Exploration

With the network configured as shown in Fig. 4 in NS-3 OpenGym, the bandwidth and delay of each link is varied to show how the congestion window responds. There were 1,600 different combinations of cases to train the RL agent on how

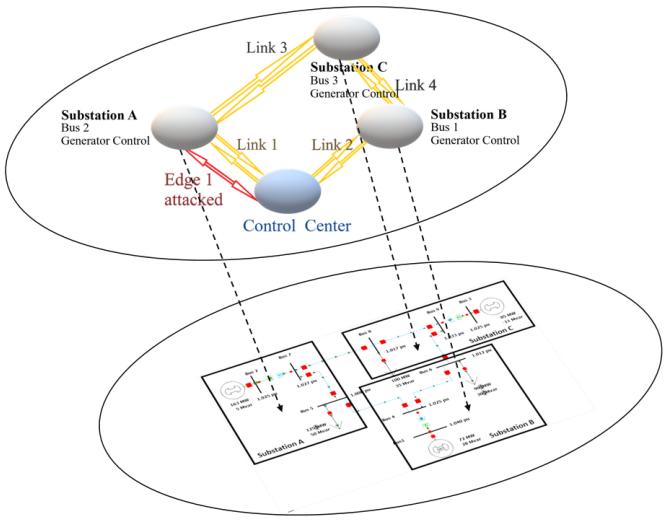

Fig. 4. WSCC 9-Bus Cyber Physical Model

to properly adjust its congestion window in order to minimize the round trip time (RTT) to meet the DNP3 requirement. For each trial, a different set of bottleneck links were targeted of the possible 16 different combinations. Then the banwidth of the bottleneck links were varied between: (0.8, 0.9, 1.0, 1.5, 2.0, 2.5, 5, 10, 15, and 20) Mbps. Additionally, the delay of the bottleneck is varied between: (5, 10, 15, 20, 25, 50, 100, 150, 200, and 250) ms.

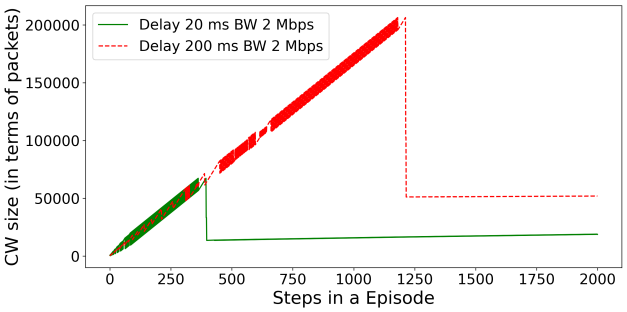

Fig. 5. Impact on Congestion Window with different Bottleneck Link Delay

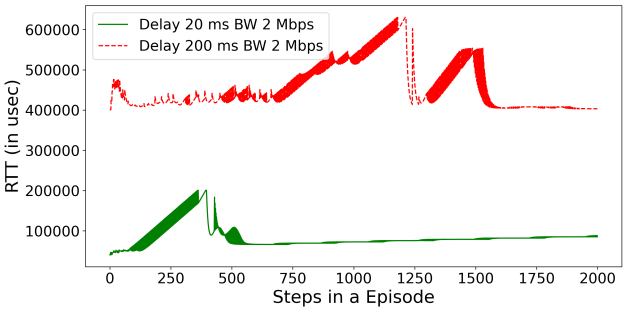

Fig. 6. Impact on RTT with different Bottleneck Link Delay

To illustrate the difference between each case, a baseline bottleneck cases was defined to have a bandwidth of 2 Mbps

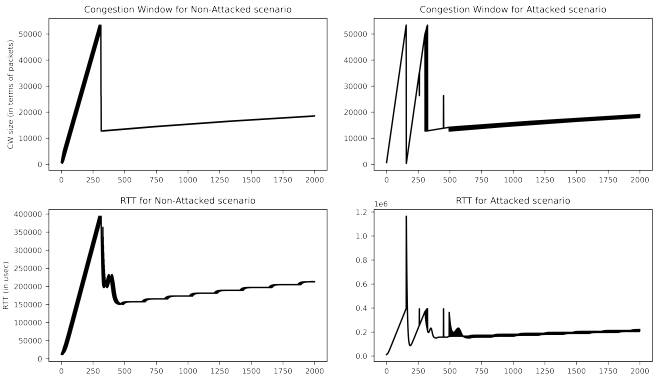

Fig. 7. DoS Attack Explorationn

with a transmission delay of 20 ms. These parameters were used since they are typical for a control center to substation link. Next, a case transmission delay increases to 200 ms with the bandwidth remaining constant at 2 Mbps. This would generally happen with a link is congested by other network traffic as in a DoS attack. A similar case was conduced where the bandwidth is increased to 1 Gbps with the delay remaining at 20 ms. Then to visualize the difference between attacking two different edges, two case were examined. First, the edge between the control center and substation was targeted, which would impact the flow of traffic between the control center substations A and C. Then, the edge 2 between substation A and substation C was targeted, which - in theory - would only affect traffic between the control center and substation C.

B. Results from Congestion Window Adjustment

The impact of varying the bottleneck bandwidth and delay on the congestion window size and RTT displayed in Figs. 5 and 6. In order to automatically adjust the congestion window size for different network conditions, the reward function for the automatic CW adjustment is designed as follows:

$$r_t = \begin{cases} 2, & \text{if } CW_{t+1} = CW_t + 1 \\ -30, & \text{if } CW_{t+1} = CW_t - 1 \end{cases} \tag{1}$$

In the base case, the congestion window is shown to increase to a maximum of approximately 70,000 frames then drop to a steady state of 20,000 frames. At the same time, RTT has a relatively small amount of variance of about 300 ms. When the delay time is increased, the maximum congestion window size increased to about 200,000 frames. This is primarily due to the bandwidth of the link not able to handle the DoS attack traffic along with its normal traffic, with the delay time varying drastically between 400 ms and 1,000 ms. When the bandwidth was increased, the congestion window grew more steadily over time to 200,000 frames with the RTT remaining relatively low at 40 ms. This demonstrates that the delay time has a greater effect on the congestion window size than the bandwidth of the bottleneck link. The DoS attack on edge 1 and 2 are relatively similar, both had high RTT. When edge 2 was targeted, the RTT spiked to 1200 ms, as opposed to when no edge were targeted and the RTT was 400 ms, as shown by Fig. 7.

C. Results from Alternative Physical Control

In order to provide the reliable alternative control suggestions in a complex cyber-physical environment when the intensity of the cyber attack exceeds the capability of the congestion window reconfiguration, the response engine is developed to achieve two main objectives: 1) to avoid any voltage violations and branch overloading; 2) to issue commands that will not be affected by the congested area. To achieve these objectives, the reward function is designed as follows:

$$r(v,u) = \begin{cases} -100 * v, & \text{if } v > 0 \text{ or } u > 0 \\ 200, & \text{otherwise} \end{cases} \tag{2}$$

where *v* is the number of all the violations and *u* is whether the control has conflicts with the cyber constraints. Fig. 8 shows the moving average of attempts during the training process, which indicates that the DRL model in the response engine gradually learns the optimal policy against the changing environment with cyber constraints. At the starting stage (episodes 0-2k), due to the high exploration rate, the agent tends to explore different combination of actions, thus the efficiency to solve the system violations is relatively low. This can be compared to the close-to-end stage (episodes 4k-6k), where the exploration rate has decayed to a minimal value (0.001), thus the actions with the highest possible cumulative rewards are always selected. This explains why almost only one attempt is required by the agent to find the alternative control candidate at the end of the training episodes.

TABLE III
HYPERPARAMETERS OF THE DRL MODEL

Hyperparameter	Value
NN Structure	300x150x125 dense
Activation	*relu*
Optimizer	Adam
Discount rate	0.99
Initial exploration rate	1.0
Final exploration rate	0.001
Exploration decay	0.9992
Replay memory	300
Target update weight	0.5

The performance of the DRL model in the response engine is evaluated in 2000 test scenarios which differ from the training scenarios. Table. IV shows the results from the testing dataset. Compared to the DRL-based model[17] that does not consider the cyber constraints, the model performs better under a cyber environment with TCP congestion. The agent only needs 1.05 (average) attempts to solve the voltage violation and bypass the congested area, while the benchmark model needs to take 2.89 times to do that. Consider the similarity between the underlying DRL models, the difference on performance on mainly caused by the state space (cyber-physical vs physical) and the reward function design. This result proves the effectiveness and timeliness of the proposed approach under the scenarios where resizing congestion window is not sufficient.

TABLE IV
AVERAGE ATTEMPTS FOR THE TEST DATASET

Method	Average Attempt
Baseline	2.89
Proposed method	1.05

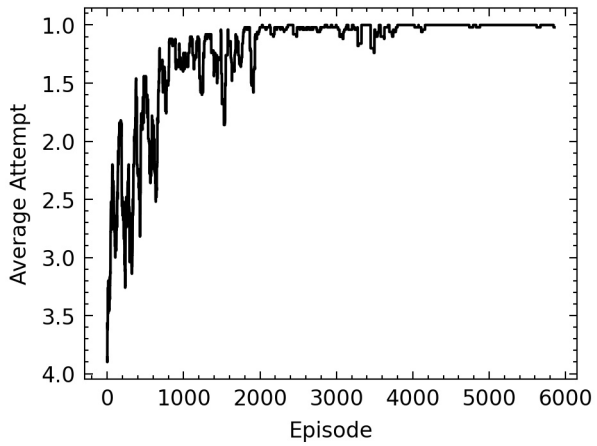

Fig. 8. Average attempts per episode during the training process (processed by moving average with length 50)

VI. CONCLUSIONS

In this work, a cyber-physical approach to efficiently prevent and mitigate the harm that can be caused to the grid by TCP congestion is proposed. The approach utilizes real-time data from both cyber and physical systems, and based on the intensity of the congestion, it selects the more effective action by adjusting the congestion window and providing alternative control actions that can bypass the congested area. Both actions are found to be effective from the experiments in different scenarios. The approach is designed to be compatible with multiple ICS protocols. Compared to other RL-based CPS methods which generally focus on physical-only state and action space, this two-stage approach is capable of adjusting the congestion window size to mitigate the impact of DoS attacks. By expanding the state space to consider both the cyber and physical domains, the approach could also efficiently provides the alternative control suggestions to operators that avoid the communication-interrupted area. In the future work, we will further investigate the impact induced by processing latency of this approach on system dynamics.

ACKNOWLEDGMENT

This work was supported by the National Science Foundation under Award Number ECCS-1916142, the U.S. Department of Energy (DOE) under award DE-OE0000895 and the Sandia National Laboratories' directed R&D project #222444.

REFERENCES

[1] M. Rahimiyan and H. R. Mashhadi, "An adaptive q-learning algorithm developed for agent-based computational modeling of electricity market," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 40, no. 5, pp. 547–556, 2010.

[2] D. Ernst, M. Glavic, F. Capitanescu, and L. Wehenkel, "Reinforcement learning versus model predictive control: a comparison on a power system problem," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 39, no. 2, pp. 517–529, 2008.

[3] T. Yu, B. Zhou, K. W. Chan, L. Chen, and B. Yang, "Stochastic optimal relaxed automatic generation control in non-markov environment based on multi-step q (λ) learning," *IEEE Transactions on Power Systems*, vol. 26, no. 3, pp. 1272–1282, 2011.

[4] Y. Xu, W. Zhang, W. Liu, and F. Ferrese, "Multiagent-based reinforcement learning for optimal reactive power dispatch," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, pp. 1742–1751, 2012.

[5] G. K. Venayagamoorthy, R. K. Sharma, P. K. Gautam, and A. Ahmadi, "Dynamic energy management system for a smart microgrid," *IEEE transactions on neural networks and learning systems*, vol. 27, no. 8, pp. 1643–1656, 2016.

[6] Q. Shi, H. Cui, F. Li, Y. Liu, W. Ju, and Y. Sun, "A hybrid dynamic demand control strategy for power system frequency regulation," *CSEE Journal of Power and Energy Systems*, vol. 3, no. 2, pp. 176–185, 2017.

[7] S. Ossenbühl, J. Steinberger, and H. Baier, "Towards automated incident handling: How to select an appropriate response against a network-based attack?" in *2015 Ninth International Conference on IT Security Incident Management & IT Forensics*. IEEE, 2015, pp. 51–67.

[8] P. Gawłowicz and A. Zubow, "Ns-3 meets openai gym: The playground for machine learning in networking research," in *Proceedings of the 22nd International ACM Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, 2019, pp. 113–120.

[9] M. Soetan, Z. Mao, and K. Davis, "Statistics for building synthetic power system cyber models," in *2021 IEEE Power and Energy Conference at Illinois (PECI)*. IEEE, 2021, pp. 1–5.

[10] A. Sahu, P. Wlazlo, Z. Mao, H. Huang, A. Goulart, K. Davis, and S. Zonouz, "Design and evaluation of a cyber-physical testbed for improving attack resilience of power systems," *IET Cyber-Physical Systems: Theory & Applications*, 2021.

[11] B. L. Thayer, Z. Mao, Y. Liu, K. Davis, and T. Overbye, "Easy simauto (esa): A python package that simplifies interacting with powerworld simulator," *Journal of Open Source Software*, vol. 5, no. 50, p. 2289, 2020.

[12] T. J. Overbye, Z. Mao, K. S. Shetye, and J. D. Weber, "An interactive, extensible environment for power system simulation on the pmu time frame with a cyber security application," in *2017 IEEE Texas Power and Energy Conference (TPEC)*, 2017, pp. 1–6.

[13] T. J. Overbye, Z. Mao, A. Birchfield, J. D. Weber, and M. Davis, "An interactive, stand-alone and multi-user power system simulator for the pmu time frame," in *2019 IEEE Texas Power and Energy Conference (TPEC)*, 2019, pp. 1–6.

[14] Z. Mao, H. Huang, and K. Davis, "W4ips: A web-based interactive power system simulation environment for power system security analysis," in *Proceedings of the 53rd Hawaii International Conference on System Sciences*, 2020.

[15] D. Wallison, M. Gaskamp, Z. Mao, Y. Liu, K. S. Shetye, and T. Overbye, "Design considerations for operational power system simulation scenarios," in *2020 52nd North American Power Symposium (NAPS)*, 2021, pp. 1–6.

[16] I. C. for a Smarter Electric Grid (ICSEG), "Wscc 9-bus system."

[17] J. Duan, D. Shi, R. Diao, H. Li, Z. Wang, B. Zhang, D. Bian, and Z. Yi, "Deep-reinforcement-learning-based autonomous voltage control for power grid operations," *IEEE Transactions on Power Systems*, vol. 35, no. 1, pp. 814–817, 2019.