Adaptive and Predictive SDN Control During DDoS Attacks

Jagannadh Vempati
Department of Computer Science
Kettering University
Flint, USA
jvempati@kettering.edu

Ram Dantu
Department of Computer Science and Engineering
University of North Texas
Denton, USA
ram.dantu@unt.edu

Syed Badruddoja
Department of Computer Science and Engineering
Universirty of North Texas
Denton, USA
syedbadruddoja@my.unt.edu

Mark Thompson
Department of Computer Science and Engineering
Universirty of North Texas
Denton, USA
mark.thompson2@unt.edu

***Abstract*—While Distributed Denial of Service (DDoS) attacks continue to plague the Internet, Software Defined Networks (SDNs) offer the promise of configuration elasticity for all devices in the plant to mitigate these attacks and stabilize the network in real-time. Since it is difficult to distinguish between legitimate and attack traffic, we propose a closed-loop feedback control mechanism using a multi-loop proportional, integral (PI) controller and a model predictive controller (MPC) that will regulate the system characteristics with disturbance rejection. Results from our setup implemented in the SDN platform show that our dynamic and predictive control models provide for a graceful degradation of real-time services in the SDN environment in real-time.**

Keywords—feedback control, DDoS, SDN, network resilience

I. Introduction

A DDoS attack on October 21, 2016, served as a stark reminder of the perils of unregulated network traffic. This massive attack took down a significant portion of the Internet services on the U.S. east coast and in central Texas. As a result, various Internet services such as Twitter, Reddit, Amazon, and Spotify were not available to the users. The rate of the attack recorded was around 600 Gbps. A few months later, Microsoft's instant messaging service, Skype, suffered from connectivity issues due to an alleged DDoS attack [2]. This outage lasted for days, with users unable to communicate with each other. The immediate fallout of these attacks is apparent in lost revenue and loss of consumer loyalty, which are crucial to network service providers. To ensure a user's quality of experience (QoE), we need to be able to identify and mitigate these attacks quickly.

With attacks growing in number, the design of a resilient network capable of absorbing a disturbance and providing the desired level of service is critical. A desirable attribute of resilient networks is the ability to dynamically configure the network. The emerging paradigm of Software Defined Networking (SDN) that attracted attention in recent years promises to deliver elasticity in configuring various network devices, such as routers, switches, firewalls, and other devices.

With the deployment of SDN, network performance can be improved through effective traffic engineering. Microsoft designed a software-driven WAN that interconnects data centers and achieves high network throughput [3]. The flexibility of SDN architecture has convinced Google to adopt it as well in its data centers [4]. The SDN architecture also finds applications in various types of networks such as wireless, home, cellular, and enterprise.

II. Motivation

DDoS attacks are evolving, and hackers are unleashing new techniques to amplify them. The financial impact of these attacks is growing rapidly for many businesses. The average cost of a DDoS attack on an enterprise is over $2M per attack and is rising dramatically [5]. In addition, these attacks also damage the reputation of the organizations. Current mitigation techniques ranging from hours to days are completely unacceptable given the cost and inconvenience these attacks place on our society. Despite the existence of many cybersecurity tools and techniques, many networks are still vulnerable to malicious attacks.

The main problem with a DDoS attack is that it may be difficult to distinguish between legitimate traffic and attack traffic. During an attack, network services are impacted, and the service availability drops drastically. Since networks are complex and highly dynamic, static techniques such as rate limiting using Access Control Lists [6] provide an inadequate response to such types of attacks. In this regard, feedback control mechanisms play a crucial role to mitigating the attacks in real time [7]. Feedback control is about regulating the system characteristics with disturbance rejection.

We propose a closed-loop feedback mechanism applied to real-time services to limit the impact of an attack such that the QoS metrics of the network remains consistent with those specified in the service level agreements (SLAs). An SLA is a commitment guaranteed by the Internet service providers to the user and includes one or more service level objectives (SLOs), which is a criterion used to evaluate the performance of the service. These objectives include metrics such as availability, delivery time, response time, failure rate, delay, jitter, and various other scalability metrics. Enforcing these objectives requires the service providers to meet their requirements by efficiently utilizing sufficient resources. Hence, implementing SLOs becomes a control problem.

During an attack, the closed-loop feedback control mechanism detects and limits the impact of an attack on real-time services by maintaining the SLAs. The controller makes intelligent decisions to ensure the SLOs are met by dynamically regulating the configuration settings of the network devices. The first controller is a multi-loop proportional, integral (PI) controller. The second controller, a model predictive controller (MPC), is an exceptional feedback control system that uses a model to predict the future outputs of a process. Implemented in an SDN platform, results from our setup show that both feedback control models provide graceful degradation of the real-time services and stabilize the network in real-time.

III. Related Work

Control theory approach has been applied to a wide range of computing systems. Xiaoqi Yin et al. [8] propose a control theoretic approach for dynamic adaptive video streaming over HTTP (DASH) to ensure good quality of experience. The authors implement a model predictive control algorithm for bitrate adaptation. In addition, the authors also develop a robust MPC that handles errors in throughput predictions. The basic MPC algorithm implemented by the authors has three main steps: 1) predict the throughput by looking N steps ahead, 2) optimizing quality of experience (QoE) maximization problem, and 3) applying the methodology where the player downloads the chunks. In this approach, the bitrate of the video and buffer occupancy is applied as feedback. The authors implement their algorithms in the dash.js framework and evaluate the performance of their approach with the current existing rate and buffer based techniques. The authors claim that their proposed approach outperforms existing algorithms.

A client-side controller for DASH [1] was implemented Luca De Cicco et al. Their implementation differs from the conventional approach, which employs controllers to throttle the video level and the regulate the buffer. The authors use a single controller that throttles the video level to drive the buffer length. The control input is the buffer length, and the measured output is the video level. The authors use a feedback linearization technique to compute the control law. From various results, the authors show that their methodology performs flawless and provides high channel utilization even when the video flows share a bottleneck in the presence of TCP flows. To the best of our knowledge, the use of feedback control to build resilient networks is yet to be explored.

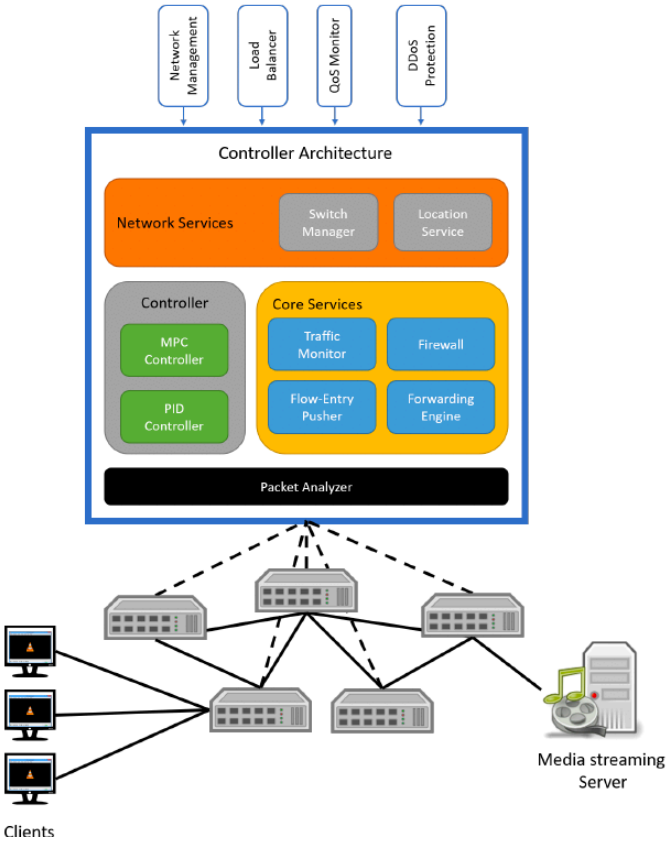

Fig. 1. Implemented Software Defined Network (SDN) Topology.

IV. System Overview

A. Mininet

Mininet [9] is an open-source network emulator used to build a virtual SDN network. It provides a simple and inexpensive way to design and implement a network including many virtual hosts, switches, and controllers. The hosts run standard Linux network software. The switches support OpenFlow for custom routing. This tool is used to simulate the SDN network and the topology as shown in Fig. 1.

B. Client

We use the VideoLAN client (VLC) media player [10] to play the streaming video and audio content. Multiple clients installed with VLC media players were used to play the same content from the streaming server.

C. Server

A standalone media server is used to host all the media content and stream to all the devices over the network. This server is equipped with an Intel Xenon processor (4 cores) and a 32 GB RAM. We consider three types of video qualities: a) 2K Video (high quality), b) 720p (high Definition), and c) 480p (standard definition).

D. Generation of Traffic

We use VLC media player using the Real Time Streaming Protocol (RTSP) to broadcast a stream. We consider three types of video qualities: a) 2K (high quality), b) 720p (high definition), and c) 480p (standard definition), as well as a high-resolution audio (320 Kbps). We use a tcpdump packet analyzer to sniff the packets at the server and client interfaces, which act as sensors. The traffic monitor module collects the configuration parameter settings of switches and other network devices present in data plane of the SDN Controller (SDNC). These settings impact the output of network QoS metrics.

To emulate a real-world attack, we use the hping3 [11] DDoS tool that is capable of generating a large number of UDP and TCP packets. The attack traffic can be modeled as a disturbance input into the plant. The severity of the attack is varied by altering the packet size.

V. Methodology

A. Closed Loop Feedback Control

The goal is to maintain real-time network services by enforcing SLAs, which include SLOs such as providing a response time of fewer than 2 seconds, network availability, and network latency of less than 30 ms. Another important objective is that the Internet service provider cannot afford to drop 911 emergency calls despite traffic congestion.

In order to reduce unnecessary costs, the SLOs must be met using available resources. To achieve these objectives, we propose two closed-loop feedback control mechanisms as shown in Fig. 2 and Fig. 3. These mechanisms guarantee a graceful degradation of real-time services, i.e., to maintain limited functionality of network services even when the network is disrupted by an attack or large amounts of traffic. During an attack, network services are impacted, and the Quality of Service (QoS) of legitimate traffic drops drastically. The purpose of the controllers is to limit the impact of an attack, such that the QoS metrics of the network is consistent with those specified in the SLAs. The SLOs are provided as a reference input for the desired value of the network's output.

B. Committed Information Rate (CIR)

The committed information rate (CIR) defines the number of tokens removed at each interval. The CIR parameter polices the excess traffic that does not comply with the policy, i.e., if the traffic flow reaches the configured CIR rate, the excess traffic is dropped. The CIR is considered as a control input which regulates the flow of traffic.

C. Control Approach 1

The closed-loop feedback architecture is shown in Fig. 2. Here, the multiloop PI controller is implemented to resolve service disruptions. The controller detects anomalies in the network and regulates the CIR configuration setting to maintain the network in the desired state. From several experiments, we observe that the configuration setting of the CIR affects the bitrate and the packet loss. Hence, each output (bitrate and packet loss) from the system is fed back to an individual PI controller, which determines an appropriate CIR value. The minimum value of the controller's output is downloaded into the congested switch to stabilize the network instantly.

The multi-loop PI controller is a combination of two single-input-single-output (SISO) models. One SISO model quantifies the relation between the bitrate and CIR configuration setting, while the other captures the relation between packet loss and CIR. The objective of the feedback control mechanism is to dynamically control the system to achieve the desired state as defined by reference input. In a PI feedback control mechanism, the measured system output is continuously fed back. The controller receives the control error and determines the appropriate setting of control input based on the proportional (P) and integral (I) terms. The control error is the difference between the measured output and the reference input.

The general equation of a PI controller is given below:

$$u(t) = K_p e(t) + K_i \int_0^{t'} e(t')dt' \quad (1)$$

The proportional (K_p) variable adjusts the system output proportionally to the error signal by controlling the proportional gain of the controller. Isolated use of a proportional controller might help in reducing the error, but the final output might result in oscillations such as an on-off signal. These oscillations can be reduced by the integral variable (K_i).

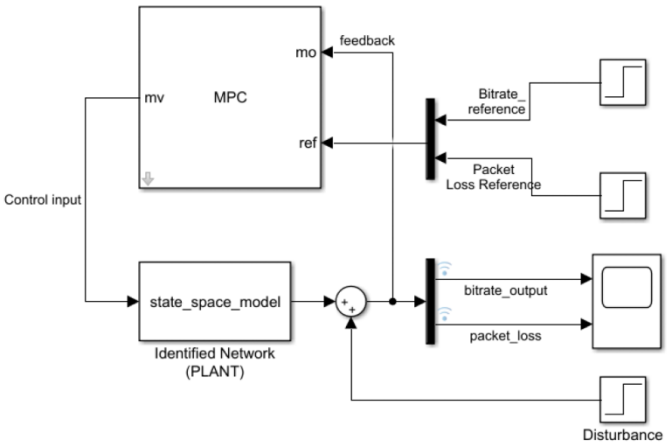

Fig. 3. Closed-loop feedback controller architecture with MPC controller, where *mv* (manipulated variable) is control input predicted by the controller.

D. Control Approach 2

Fig. 3 shows the model predictive controller that constitutes the closed-loop feedback architecture implemented in this approach. A model predictive controller [12] is an excellent feedback control algorithm that uses a model to predict the future outputs of a process. The controller achieves this by using an explicit dynamic model of the system response to the manipulated variables (MVs), or the control input. The controller uses this model to control the plant. MPC has been applied to many process control industries including chemical plants, oil refineries, and power plants and has been proven to be a better option to PI controller for complex systems [12]. Unlike a PI controller, an MPC controller provides the flexibility of controlling multiple system configurations that impact the system outputs.

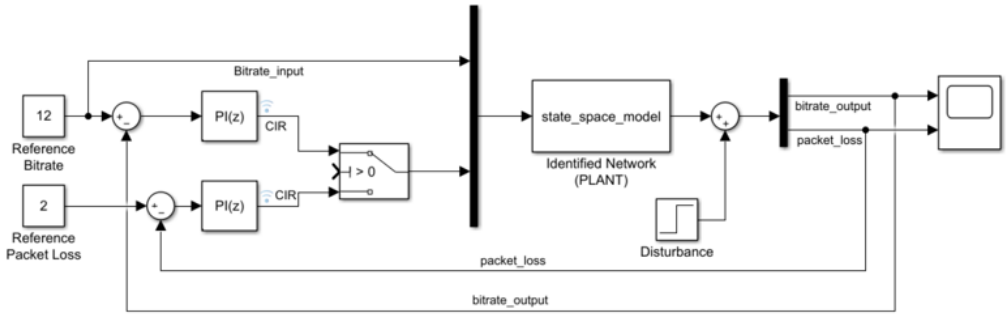

Fig. 2. Closed-loop feedback controller architecture with multi-loop PI controller. The PI controller regulates the process by adjusting the control input (CIR configuration setting).

The MPC controller uses the model of the plant to predict the future plant output behavior. The prediction horizon determines the number of future control intervals the controller must predict. The controller uses an optimizer to ensure the predicted output follows the desired reference input. It minimizes the error of the predicted output and the reference input by solving the online optimization problem at every control interval. The solution regulates the control input or the manipulated variables to be used in the plant until the next interval.

Integrated into the SDNC, the MPC controller utilizes the traffic statistics from the traffic monitor module. The traffic monitor feeds the network characteristics to the MPC controller. The controller detects the abnormalities of the network service and adjusts the control input to guarantee the service quality. Downloaded into the congested switch in the data plane, the control input is the CIR configuration setting that drops the excess traffic when the traffic reaches the configured rate. The congested switch is identified through link utilization by the SDN controller.

The state-space model of the plant contained inside the MPC controller is the second order discrete-time state space model shown below.

$$x(t+Ts) = A\,x(t) + B\,u(t) + K\,e(t) \quad (2)$$

$$y(t) = C\,x(t) + D\,u(t) + e(t) \quad (3)$$

where u is the input, x is the state, y is the output, and e is the error. A, B, C, D, and K are matrix coefficients.

VI. Results and Analysis

The SDN shown in Fig. 1 was implemented to study the performance of both feedback architectures. The switches in the data plane are connected in a mesh topology. These switches are connected to the controller, which has the ability to control the flow, configure the switch, and monitor the traffic using OpenFlow communication protocol. The data plane is set up using Mininet while the Ryu controller [13] is used as an SDNC. The switches update their flow statistics to the controller periodically. The PI controller and the MPC controller are implemented as a module in the SDNC. Traffic statistics such as number of flows, packets per second, and rate of packets in bytes per second are fed back to the controllers.

Audio and video traffic are streamed across the network using the VLC streaming server. The clients play the video using VLC client application. Both audio and video traffic are marked as high priority, i.e., DSCP=46. Three different types of video streaming resolutions are considered: a) 2k Video, b) 720p Video (high definition), and c) 480p Video (standard definition). The video played contains many fast moving objects, explaining the reason for a varying bitrate, while the bitrate of the audio stream is constant.

In order to emulate a real-world DDoS attack, the network is flooded with a large number of packets. The attack resembles a step input that is modeled as disturbance into the plant. When the network is under attack, the links get congested, resulting in dropped packets, increasing the delay and jitter of the services. The QoS of the network service drops drastically. Despite prioritizing both video and audio traffic, the attacks impact the high priority video and audio traffic as shown in Fig. 4 and Fig. 5, respectively.

A. PI Controller

The PI controller is designed as a multi-loop SISO controller as shown in Fig. 2. The controllers are connected to a switch that selects the minimum CIR value (control input), stabilizing the network. The QoS of the network services, or output of the system, is fed continuously into the controller. When the network is under attack, the QoS of the real-time traffic drops drastically. The PI controller detects the abnormalities in the network characteristics from the feedback signal and adjusts the CIR input to the system calculated from the error signal making the network resilient towards the attack. The CIR input provided by the controller controls the data/attack traffic.

Fig. 4 gives the output from the network shown in Fig. 2, collected from the PI controller scenario. A 720p resolution video is streamed across the SDN network. The variation in the bitrate is due to the content of the video containing many fast moving objects. The QoS metrics of the video, collected from the traffic monitor module present in the SDNC is fed to the controller. After nearly 40 seconds, the network is flooded with a significant amount of UDP packets, congesting the links. Due to the congested links, a large number of packets are dropped. The packet loss of the video traffic sharply increased to nearly 50% and the bitrate of the video drops to nearly 50 kbps, degrading the quality of the video. The PI controller detects this drop in the QoS of the video traffic and adjusts the control input, i.e., the CIR value of the congested node, to bring back the network to the desired state.

The setpoint values or the reference values define the desired state. The controller tries to match the measured system output to the provided reference values by adjusting the CIR input. Since PI controller is capable of controlling single input

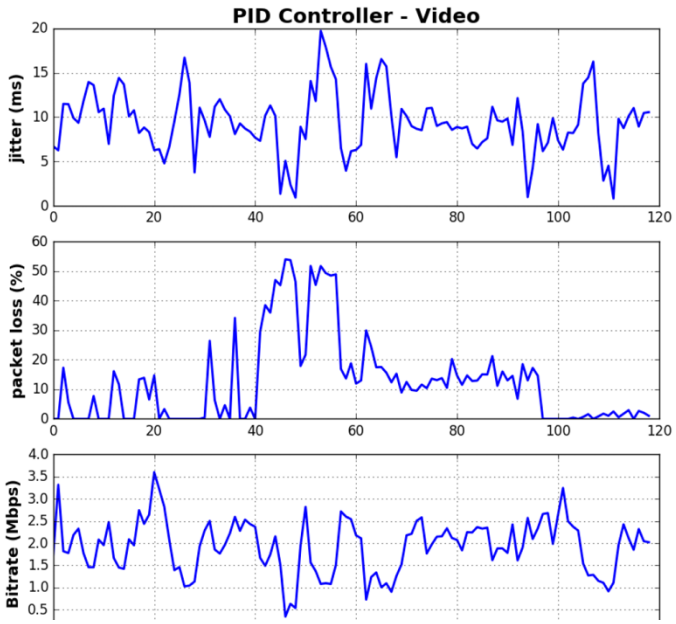

Fig. 4. Video (a) Jitter, (b) packet loss, and (c) bitrate measured at PI controller from SDN under attack. Attack at t=40 seconds shows severe deterioration of video, but impact is reduced after nearly 60 seconds.

and single output system, each PI controller tries to reduce the CIR value based on the corresponding reference input. The reference values of bitrate and packet loss are based on the SLA agreements of the network and are set to 7 Mbps and 3% respectively, i.e., the bitrate of the video should be maintained at 7 Mbps, and the network can afford up to 3% packet loss.

After nearly 60 seconds, the controller achieves its goal of stabilizing the traffic by minimizing the control error, which is the difference between the current measured output value and the reference input. The measured output of the system is fed periodically to the controller. The SDNC provides flexibility to the PI controller to dynamically set the CIR value so that it is gradually decreases until the measured system output converges to the reference value. The video is restored back at 98 seconds when the control objectives are met. The packet loss drops down to less than 5%, and the bitrate of the video matches the reference value. The settling time, which is measured as the period from the detection of the disruption in the video traffic to bringing the system to the desired state, is nearly 60 seconds.

Fig. 5 shows the QoS metrics of the network traffic collected while streaming high definition audio. It is evident that the bitrate of the audio traffic is very smooth, unlike the video traffic, and much lower than the video traffic. During an attack, the packet loss increases to nearly 70%. The controller performs a similar functionality in controlling the degradation of audio.

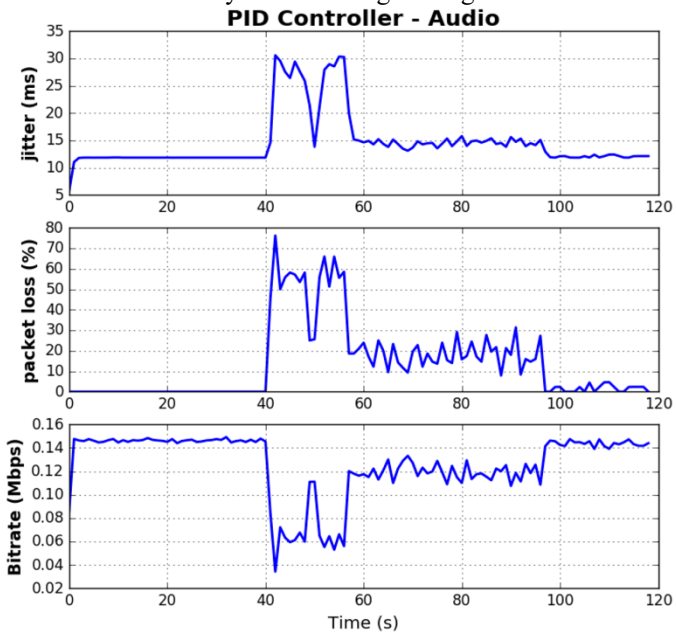

Fig. 5. Audio (a) Jitter, (b) packet loss, and (c) bitrate measured at PI controller from SDN under attack. Attack at t=40 seconds shows severe deterioration of audio, but impact is reduced after nearly 60 seconds.

B. *MPC Controller*

The MPC controller implemented in this scenario controls the network under attack by adjusting the CIR value, inspecting both the measured system outputs simultaneously. From the experiments conducted, we observe that both the CIR value setting and input bit rate of the video affect the behavior of the network. The MPC's inherent property of being able to control a MIMO system provides an accurate prediction of the CIR configuration parameter.

Fig. 6 shows the output of the MPC controller architecture implemented while streaming a high definition video stream (720p). This data represents the QoS metrics of the video traffic collected every second from the client. The reference values of the bitrate and the packet loss are set to 7 Mbps and 3%, respectively. After 40 seconds, the network is flooded with a large number of packets, congesting the network. Consequently, the QoS of the video traffic drops to nearly 50%, increasing the control error that is used to solve the online optimization problem by the MPC controller, and the solution determines the setting of the control input (CIR value). The impact of the attack on the video traffic is limited after nearly 20 seconds. Unlike the PI controller, the MPC controller quickly restores the quality of the video.

A similar approach was applied to the network while streaming high definition audio. The output of the QoS metrics of the audio stream is shown in Fig. 7. The quality of the audio is restored in nearly 15 seconds.

Fig. 8 shows the predicted control input (CIR values) of the designed MPC controller for various resolutions of video. The congested switches in the data plane are configured with the predicted CIR values periodically. We can observe that the predicted value is inversely proportional to the bitrate of the video. The higher the bitrate of the video, the lower is the predicted CIR value. Initially, CIR is set to 100. When the network is disrupted by an attack, the QoS metrics drop, thereby increasing the control error that is used by the plant model contained in the MPC architecture to predict the CIR values. Furthermore, it can be inferred from the figure that the predicted CIR values reveal the under-damped dynamics of the controller. After nearly 40 seconds, the oscillating CIR values begin to converge to a steady state.

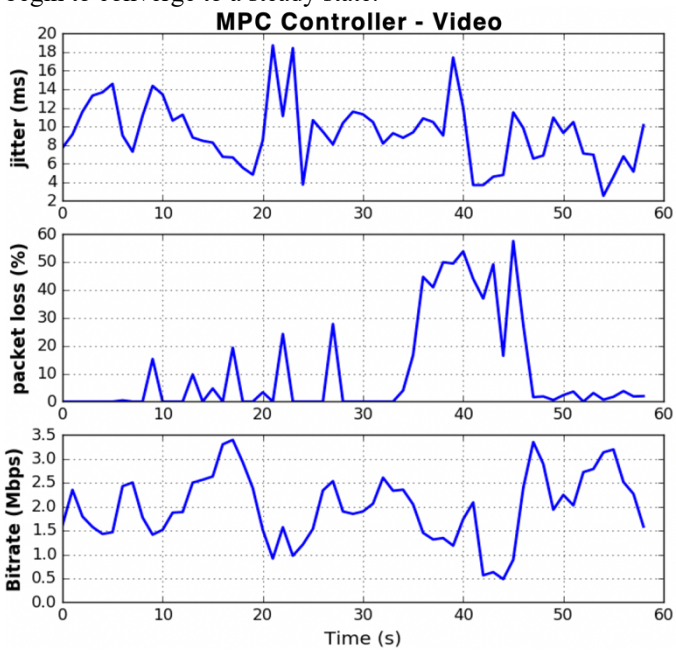

Fig. 6. Video (a) jitter, (b) packet loss, and (c) bitrate measured at MPC controller from SDN under attack. Attack at t=30 seconds shows deterioration that is restored after nearly 20 seconds.

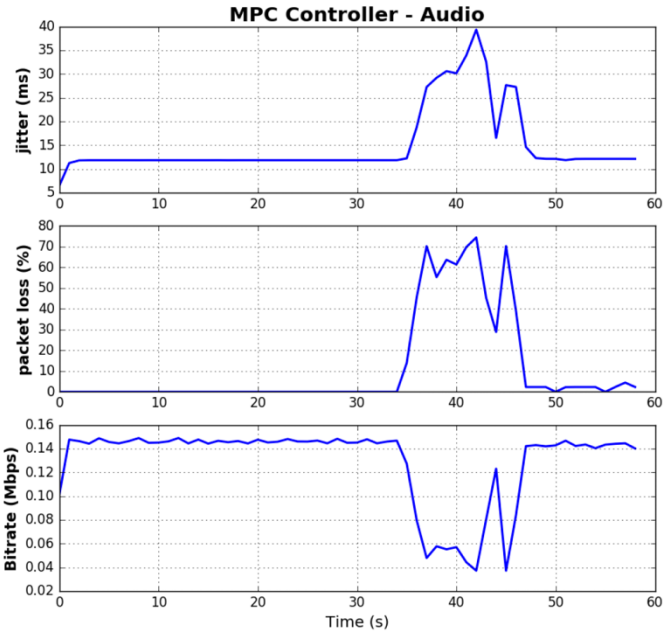

Fig. 7. Audio (a) jitter, (b) packet loss, and (c) bitrate measured at MPC controller from SDN under attack. Attack at t=30 seconds shows deterioration that is restored after nearly 15 seconds.

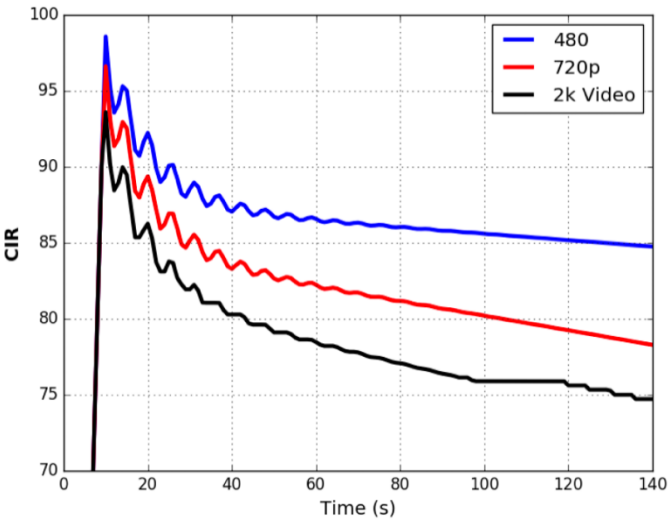

Fig. 8. The predicted control input (CIR values) of designed MPC controller for various video resolutions that reveals under-damped dynamics of controller. After nearly 40 seconds, oscillating CIR values begin to converge.

VII. Conclusion

Since attack traffic is not detected, the dropped traffic might include insignificant data traffic. This methodology holds well for graceful degradation of the real-time services and the network. This approach can be applied to mission-critical real-time applications such as surveillance traffic, 911 emergency calls, and air traffic control, where packet loss cannot be tolerated. Most streaming video applications use TCP as the underlying transport protocol, which intuitively handles network congestion.

The proposed approached steers the network towards desired performance as prescribed by the SLAs. It is important to note that the controllers designed are modeled for disturbance rejection. The limitation of a PI controller is that the controller has a longer settling time to perform the control actions, as evidenced by the results. Whereas, using the MPC controller the QoS of real-time traffic gracefully degrades the network within a shorter settling time. MPC controller has the flexibility of handling Multi-input-multioutput systems where multi-variable inputs control the outputs. On the other hand, PI controllers handle only SISO systems. However, PI controllers can process MIMO controller by using multiloop controllers, but this approach is not scalable.

This methodology scales well in response to any magnitude of the DDoS attack. UDP flooding was considered as a significant source for the DDoS attack. However, this principle also holds good for TCP based DDoS attacks such as SYN flooding, where the controller would tune the burst size parameter to minimize the number of TCP connections.

References

[1] L. De Cicco, V. Caldaralo, V. Palmisano, and S. Mascolo, "ELASTIC: A Client-Side Controller for Dynamic Adaptive Streaming over HTTP (DASH)," *20th International Packet Video Workshop*, San Jose, CA, pp. 1-8.

[2] Skype suffering connectivity issues, allegedly due to ddos attack, https://www.scmagazineuk.com/skype-suffering-connectivity-issues-allegedly-due-to-ddos-attack/article/670071/, (Accessed on 06/21/2018).

[3] Chi-Yao Hong, Srikanth Kandula, Ratul Mahajan, Ming Zhang, Vijay Gill, Mohan Nanduri, and Roger Wattenhofer, Achieving high utilization with software-driven wan, ACM SIGCOMM Computer Communication Review, vol. 43, ACM, 2013, pp. 15-26.

[4] Sushant Jain, Alok Kumar, Subhasree Mandal, Joon Ong, Leon Poutievski, Arjun Singh, Subbaiah Venkata, Jim Wanderer, Junlan Zhou, Min Zhu, et al., B4: Experience with a globally-deployed software defined wan, ACM SIGCOMM Computer Communication Review, vol. 43, ACM, 2013, pp. 3 - 14.

[5] Ddos breach costs rise to over $2m for enterprises finds kaspersky lab report| kaspersky lab us, https://usa.kaspersky.com/about/press-releases/2018_ddos-breach-costs-rise-to-over-2m-for-enterprises-finds-kaspersky-lab-report,(Accessed on 07/13/2018).

[6] Subramani rao Sridhar rao, Denial of service attacks and mitigation techniques: Real time implementation with detailed analysis, White paper (2011).

[7] Ram Dantu, Joao W Cangussu, and Sudeep Patwardhan, Fast worm containment using feedback control, IEEE Transactions on Dependable and Secure Computing 4 (2007), no. 2, pp. 119-136.

[8] A control-theoretic approach for dynamic adaptive video streaming over http, ACM SIGCOMM Computer Communication Review, vol. 45, ACM, 2015, pp. 325 – 338.

[9] Mininet: An instant virtual network on your laptop (or other pc) - mininet, http://mininet.org/, (Accessed on 07/13/2018).

[10] Vlc: O_cial site - free multimedia solutions for all os! - videolan, https://www.videolan.org/index.html, (Accessed on 05/20/2018).

[11] hping security tool - hping3 information, http://www.hping.org/hping3.html, (Accessed on 07/12/2018).

[12] Eduardo F Camacho and Carlos Bordons Alba, Model predictive control, Springer Science & Business Media, 2013.

[13] Ryu sdn framework, https://osrg.github.io/ryu/, (Accessed on 07/13/2018).