2018 IEEE 4th International Conference on Collaboration and Internet Computing

Automatic Feedback Control for Graceful Degradation of Real-Time Services in the Face of an Attack

Jagannadh Vempati
Computer Science and Engineering
University of North Texas
Denton, USA
Email: jagannadhvempati@my.unt.edu

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

***Abstract*—Distributed denial of service (DDoS) attacks continue to pose a serious threat to various businesses and consumers. With the growth in the number of devices connected to the internet, these attacks continue to grow in number. Despite the availability of security tools, the attacks continue to happen and are causing various businesses to sweat. These tools may take anywhere from a few hours to a few days to counter the attacks, which is unacceptable. In this paper, we put forth a novel feedback control mechanism to minimize the effect of volumetric attacks such as DDoS. During an attack, the feedback control model detects and reduces the impact of the attack by maintaining the service level agreements (SLA) of the network service. The controller makes intelligent decisions to ensure the quality of service (QoS) metrics are gracefully degraded by tuning the micro-firewall rules such as the committed information rate and burst size. A proportional Integral (PI) controller is used as a closed-loop feedback controller to maintain the stability of the network in spite of an attack. This proposed architecture is verified in a lab setup, and we observe that we are able to minimize the degradation of the real-time service so that the user's quality of experience (QoE) is preserved. We validate the proposed architecture with a model generated by using the system identification technique. Results from the setup show that the closed-loop feedback control model stabilizes the network in real-time.**

I. Introduction

Distributed denial of service (DDoS) attacks are one of the biggest cyber threats for many businesses such as financial sectors, IT services, and telecom services and with the growing number of Internet of Things (IoT) devices connected to the internet, this trend is expected to continue [1]. After the fourth quarter of 2017, DDoS attacks increased by 91% [2]. According to a Verisign DDoS trends report [3], the average attack peak size increased to 850%. According to the same report, out of the different types of DDoS, UDP floods topped the chart with 42%.

On February 28th, 2018, Github suffered from the most massive DDoS attack ever [4] with a jaw-dropping amount of traffic. The rate of the attack recorded was around 1.2 Tbps. Another high profile attack was on October 21, 2016, that disrupted several services on the internet. The attack rate 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 [5]. The outage lasted for days, and the users were unable to communicate with each other. These DDoS attacks make services unavailable to the users. To ensure user's quality of experience (QoE), we need to be able to identify and mitigate these attacks. However, distinguishing between legitimate traffic and attack traffic is challenging.

A. Motivation

DDoS attacks are evolving, and hackers are unleashing new techniques to amplify the attack. The financial impact of these attacks is growing. The average cost of a DDoS attack on an enterprise is over $2M per attack and is rising dramatically [6]. These attacks not only impact the financial costs but also damage the reputation of the organizations. With attacks increasing rapidly, the design of resilient systems is of utmost importance. Resilience is the ability of a system to withstand when provoked by an external disruption. Disruption or disturbance is an abnormal activity that hinders the normalcy of the system. Resilient networks try to provide the desired level of service, despite challenges such as malicious attacks, and misconfigurations.

In our previous work, we developed a robust closed-loop feedback mechanism to maintain the stability of a web service in the face of an attack [7]. We presented a passive approach of distributing the load to multiple links and reducing the impact of an attack, using feedback control. The results proved that the proposed feedback mechanism provided a positive effect on the system and the web services were restored in real time. The impact of the attack was reduced in about 60 seconds, and the user quality of experience was maintained, making the network stable.

We chose a very high order linear transfer function model in [7]. A higher order model is very unstable, and the model could result in overfitting. Also, the model failed to capture the non-linearities of the network. To understand the complex nature of the real-time traffic and also to capture the non-

978-1-5386-9502-9/18/$31.00 ©2018 IEEE
DOI 10.1109/CIC.2018.00-37

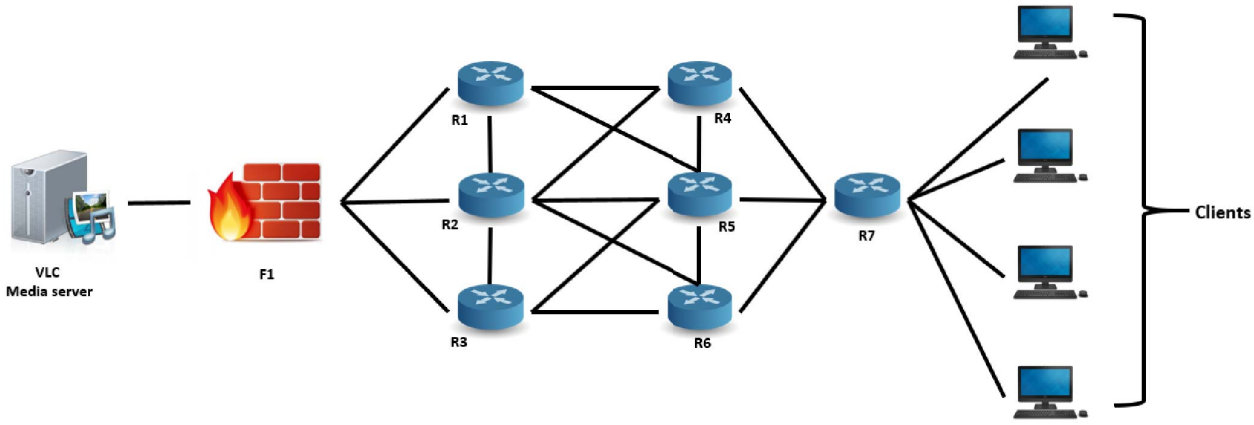

Fig. 1. Experimental Network Topology: Network Elements Connected in a mesh topology. All the network devices (F1: firewall, and R1-R7: routers) are configured with open shortest path first (OSPF) routing protocol to route the packets. The routers are configured with a round robin load balancing feature to share the load equally among the links.

linearities of the network we chose a second order state-space model to identify the network.

Our work focuses on designing a robust closed-loop feedback control mechanism to protect the network and provide resilience to a network when triggered by attacks and faults. We develop this feedback control mechanism for real-time traffic, such as streaming video and audio, which is very sensitive to delay, packet loss, and jitter. Slight disruptions in the network can hinder the performance of the real-time traffic. Network traffic such as the real-time traffic is highly dynamic, complex and very unpredictable. Understanding this complex nature of the network traffic is critical in being able to design an accurate model. In our approach, we look to system identification to build and validate a model for the complex and dynamic network.

Current mitigation techniques ranging from hours to days are entirely unacceptable given the cost and inconvenience these attacks place on our society. Our mechanism provides a real-time and scalable solution to maintain the service level agreements and to deliver video and audio streaming seamlessly smooth in spite of an attack. After designing the dynamic feedback controller, the question arises can we quantify resilience, if so, how? We look to control system theory to define the metrics of the resilient system.

B. Related Work

Feedback control theory has been applied to many computing and networking systems. However, there is not much research on the use of feedback control to detect and control the attacks. Ram Dantu et al., [8] propose an Automated Defense System based on feedback control theory to control the spreading of a worm in a network. They design a state-space model that detects and controls the virus by measuring the velocity of the number of new connections. The authors design a PI controller that limits the number of connections. The authors claim that they were able to limit the infection to less than 5% of the hosts.

Feedback control theory has been applied to various areas including performance modeling of video streaming [9]–[15]. A Quality Adaptation Controller for live adaptive video streaming based on feedback control theory was presented by Luca De Cicco et al., [16]. This controller throttles the video level. The controller tracks the queue length and outputs the preferred bitrate of the video to match the available bandwidth. The authors claim that they achieve this in less than 30 seconds.

Guibin Tian and Yong Liu [9] designed and implemented a receiver-driven adaptation algorithm for Dynamic Adaptive Streaming over HTTP (DASH). They design a PID controller driven by deviation in the client-side buffered video time. The authors claim that their algorithm provides a balance between the needs of video rate smoothness and high bandwidth utilization.

The approaches as mentioned earlier focus on performance and regulating the bitrate of video streaming. To the best of our knowledge, the use of feedback control to minimize the impact of an attack has to be yet explored especially for real-time traffic. Our approach decreases the degradation of real-time services. The feedback mechanism implemented in our approach minimizes the impact of the attack and restores back the QoE of the network service.

II. Architecture

A. Network Topology

The experimental testbed implemented in our lab is shown in Fig 1. The network comprises a firewall, routing devices,

media streaming server, and clients. We implement a mesh network using the Cisco Catalyst devices and the firewall. These devices are configured to use Open Shortest Path First (OSPF) protocol to route the packets. We also set the routers with a load-balancing feature such that the load is shared equally among the links.

In this architecture, a centralized controller collects information such as arrival rate, inter-arrival time, jitter, packet loss and various other metrics. This controller is aware of the network topology, configurations of multiple network devices, security policies and the service level agreements. This controller is a logical function and can be implemented in any network device such as routers, load-balancers, and also in any security devices such as firewalls and intrusion detection systems (IDS). In our proposed approach we implement the closed loop feedback mechanism in the firewall which is capable of controlling the micro-firewall rules to stabilize the network under attack. The measurements such as packet loss and, jitter are fed back into the firewall to understand the current state of the network.

B. Client

We use the VideoLAN client (VLC) media player [17] to play the streaming video and audio content. Multiple clients installed with VLC media players were used to playing the same content from the streaming server.

C. Server

A standalone media server is used to host all the media content and to 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 480p (Low quality).

III. Methodology

A. System Identification

To be able to design a robust feedback control architecture and to identify the characteristics of the complex network a good system model is required. Due to various working conditions of the network, building an accurate model to capture entire dynamics of the network is challenging. However, based on the observations of the network in certain conditions, we can dynamically predict a model. We are thus motivated to look into system identification technique to determine the dynamics of the complex network comprising of clients, servers, routers, firewalls, and so forth.

System identification is a process of deriving mathematical relations between input and output data [19]. We use a black box approach [19] [20] to identify the model. We consider the network comprising of routers, firewall, clients and media server as a plant. We model the plant using the system identification technique. The inputs considered are the bitrate of the video and the committed information rate (CIR). The output measured from the plant are the QoS metrics such as packet loss, jitter, and bit-rate of the video. Control input

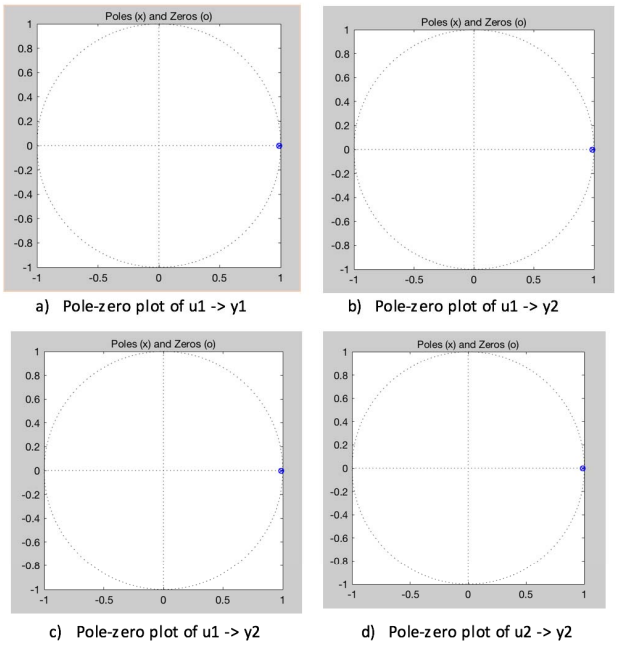

Fig. 2. Pole-zero plots of the identified model. According to control theory [18], location of poles and zeros define the stability of the system. Here, the poles and zeros lying within the unit circle indicate the model is stable.

is a parameter which can be dynamically adjusted and can affect the behavior of the system. Hence, we chose Committed information rate (CIR) as the control input. Details of the selection of control input are discussed in section III B.

State space models have provided better representations of various classes of engineering, computing and several biological systems and processes [18]. State space models are used to characterize the system, i.e., how the system functions or performs based on the state variables which explain the dynamics of the system [18]. State space models are also scalable and can be applied to non-linear systems. Hence, we decided to use a state space model for modeling our network.

The general format of a discrete-time state space model is:

$$x(t+Ts) = Ax(t) + Bu(t) + Ke(t) \quad (1)$$

$$y(t) = Cx(t) + Du(t) + e(t) \quad (2)$$

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 and must have these characteristics:
A must be an n-by-n matrix, where n is the number of states.
B must be an n-by-m matrix, where m is the number of inputs.
C must be an r-by-n matrix, where r is the number of outputs.
D must be an r-by-m matrix.

We use the the system identification toolbox of matlab [21] to construct the state-space model for the network system. The following are the parameters estimated using the toolbox:

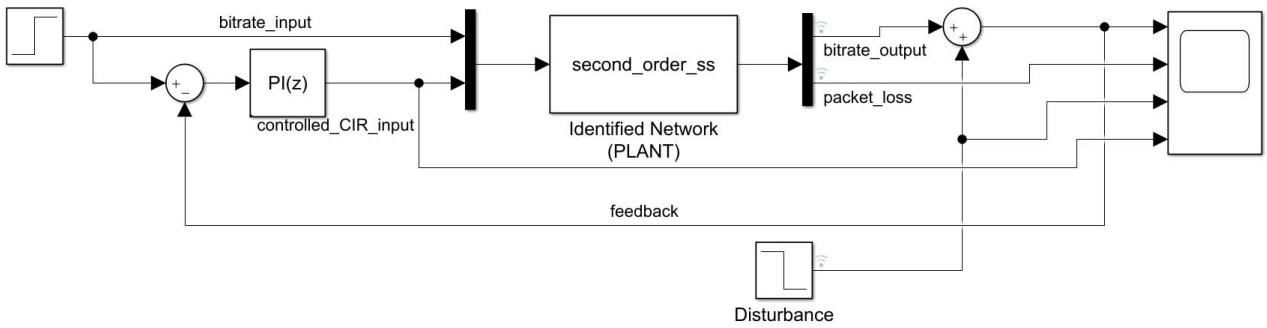

Fig. 3. Simulink model of our closed-loop feedback control architecture. The plant is the entire network comprising of all the devices including firewall, routers, clients, and servers. The output from the plant, i.e., the QoS metrics such as bitrate and packet loss are fed back into the PI controller. The PI controller designed is the brain of this system, that regulates the process. When the network is disturbed by an attack, the controller provides a controlled input, i.e., CIR, to the plant. Bitrate and packet loss are provided as a reference input to the controller. The disturbance is a negative step input added to the bitrate.

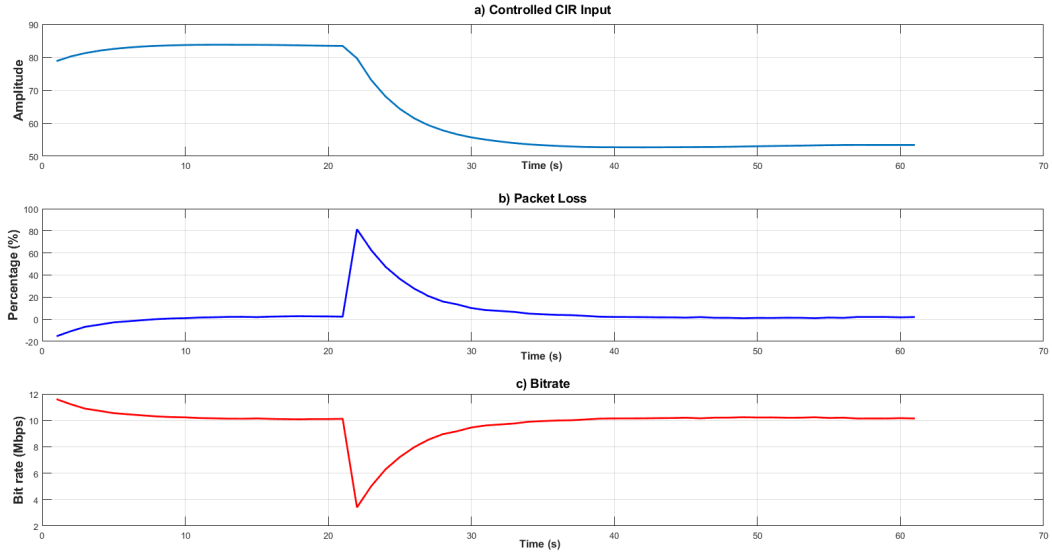

Fig. 4. Results from the simulation of the PI controller to control the impact of an attack and maintain the QoS. We can observe that when the system is excited with an attack, the PI controller regulates the CIR value, ensuring the packet loss and bitrate are maintained at the desired range, in just about 15 seconds.

$$A = \begin{bmatrix} 0.9895 & 0.000192 \\ -0.002089 & 0.9885 \end{bmatrix}$$

$$B = \begin{bmatrix} 2.266e-06 & -3.122e-07 \\ 1.347e-05 & -1.857e-06 \end{bmatrix}$$

$$C = \begin{bmatrix} 8.822 & 1.949 \\ -102.6 & -31.67 \end{bmatrix}$$

$$D = \begin{bmatrix} 2.438 & -0.2546 \\ -19.38 & 3 \end{bmatrix}$$

$$K = \begin{bmatrix} 0.4056 & 0.02522 \\ -1.341 & -0.1155 \end{bmatrix}$$

According to control theory [18], the stability of a model is defined by the location of the poles and zeros. In a transfer function, poles and zeros are the roots of the numerator and denominator polynomials respectively. The pole-zero plot of the identified model is shown in Fig 2. We can observe from the figure that the poles and zeros lie within the unit circle indicating the model is stable [18].

B. Feedback Control

Our goal is to make the network stable and resilient by maintaining the QoS and the SLA of the network service, in spite of an attack. During an attack, the network is congested, due to which the QoS of the real-time traffic drops drastically

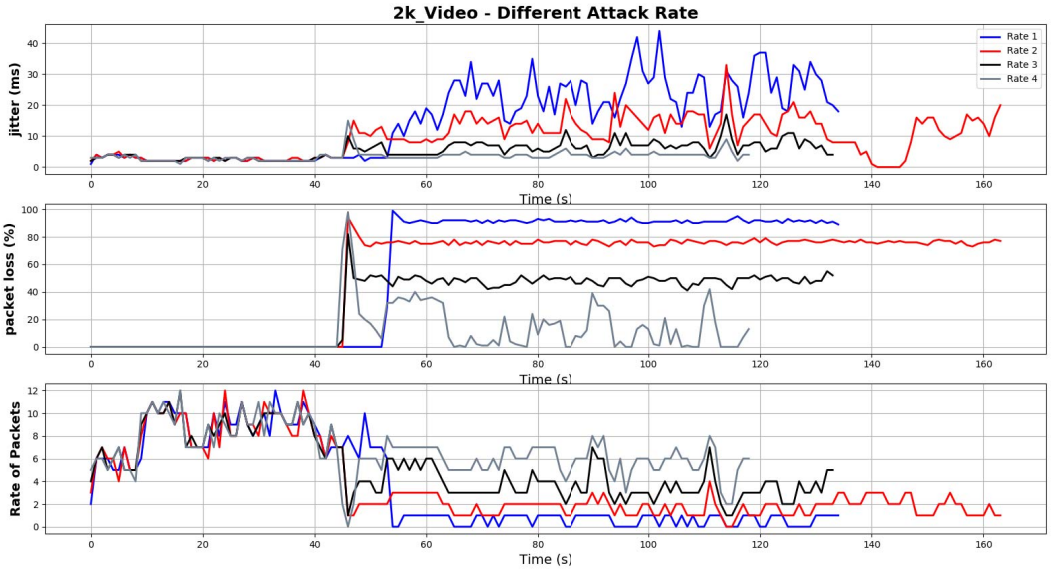

Fig. 5. (a) Jitter, (b) packet loss and (c) bitrate measured from the network before and after the attack while streaming video. Under normal conditions, i.e., from t = 1 to t = 40 seconds, the network is stable, and the video streaming is seamlessly smooth. After 40 seconds, we flood the network with different rates of attack as shown in Fig 6. We can observe the degradation of the video traffic against various rates of attack. As the rate of attack increases the QoS of video traffic drops drastically. All the attacks deter the video streaming.

as shown in Fig 5 and Fig 8. From the figures, we can observe that as the rate of attack increases the quality of both audio and video degrades progressively. To achieve our goal, the attack must be managed. We propose a robust closed-loop feedback control approach to limit the degradation of the real-time traffic gracefully. While designing a controller, we address two critical questions:

1) What can be done to control the traffic rate
2) When is the right time to introduce feedback

The answer to the first question lies in regulating the attack traffic by using a micro-firewall rule known as committed information rate (CIR).

Drop Out-of-Profile Traffic: This rule drops out-of-profile traffic based on the committed information rate (CIR) and burst size parameters. The CIR defines the number of tokens removed at each interval and the burst defines the maximum amount of packets (in megabytes) the bucket can hold at any time. The CIR parameter polices or drops 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. We consider the CIR as a control input which regulates the flow of traffic. We now understand that by tuning the CIR value we can control the traffic rate. However, the most important questions arise, i.e., question two, when is the right time to introduce feedback and what can be done to control the CIR parameter dynamically. The answer to this question lies in designing a suitable controller that offers to regulate the attack traffic.

Controller Design: A proportional-integral-derivative controller (PID controller) is a closed-loop control mechanism often used in many industrial chemical and computing systems. In a closed loop control system the current output measured from the system, also known as a process variable, is fed back to the controller along with the desired reference value. The reference value is also called as a setpoint. The controller continuously calculates the error e(t) measured from the process variable and the setpoint and applies a correction to the system based on the P, I and D terms. These three variables determine the controller's behavior. The general equation of a PID controller is given below:

$$u(t) = K_p e(t) + K_i \int_0^{t'} e(t')dt' + K_d \frac{de(t)}{dt} \qquad (3)$$

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). The derivative term (K_d) estimates the future trend of the error based on the current rate of change of the output. However, after applying the integral term the error signal converged with the output, indicating a steady state, which led us to chose a proportional-integral (PI) controller.

The PI controller implemented is shown in Fig 3. This controller controls the attack traffic by regulating the CIR

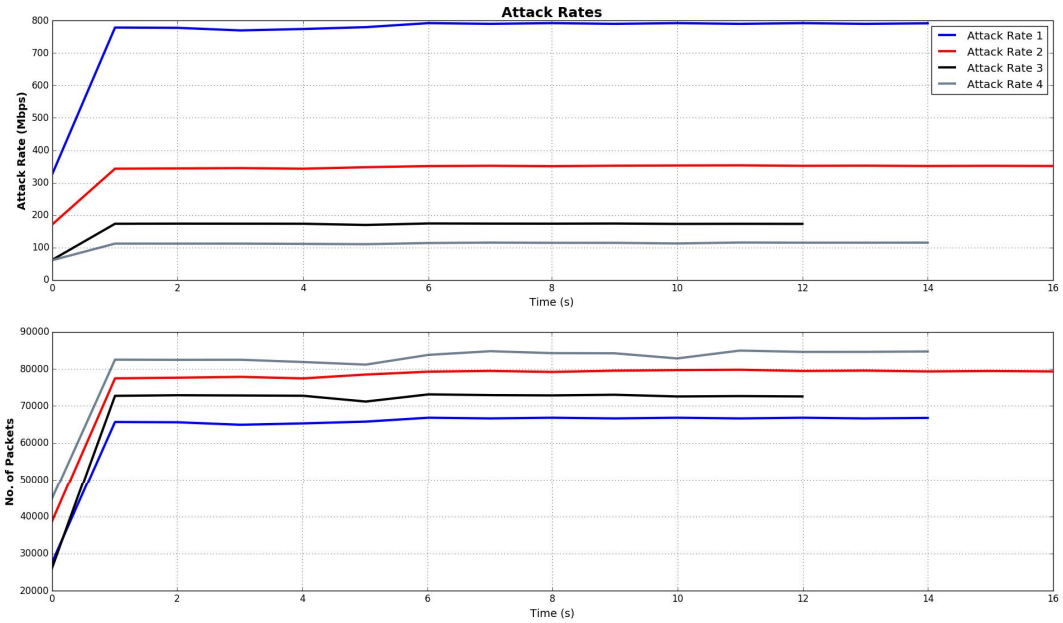

Fig. 6. Attack rates used to emulate the real world DDoS attack (a) rate of packets or bitrate in Mbps, and (b) No. of packets. The attack rates are varied by varying the packet size.

parameter. The QoS metrics measured are fed back into the PI controller, which uses this feedback to detect anomalies. When the network is under attack, the QoS of the network service drops drastically, and the network becomes unstable. This change in the QoS values such as bitrate, packet loss, and jitter is detected by the controller, which determines an appropriate CIR value to bring back the network to the preferred QoS values. During simulation, we use a negative step input as an output disturbance to the plant which reduces the QoS values of the real-time traffic.

Tuning a PI controller means selecting best values for the P and I variables to achieve the desired results. In our experiment, we want to contain the attack traffic when an attack is detected. Fig 4 shows the results of a simulation of applying a PI controller to the network designed in Matlab. We can observe that during an attack the bitrate of the video drops to less than 1 Mbps. When the feedback is provided using P = 0.047 and I = 0.094 the system converges to the setpoint value with an initial overshoot.

A logical architecture of the feedback controller implemented for the entire network is shown in the figure 7. Each wide area network (WAN) is equipped with a PI controller which provides the network with a CIR value. The controller receives the control error e_i (where i =1, 2, ..., N), which is the difference between the reference value (desired QoS) and the currently measured output QoS. When the network is under attack, the controller detects anomalies in the real-time services and regulates the CIR configuration setting to maintain the service in the desired state.

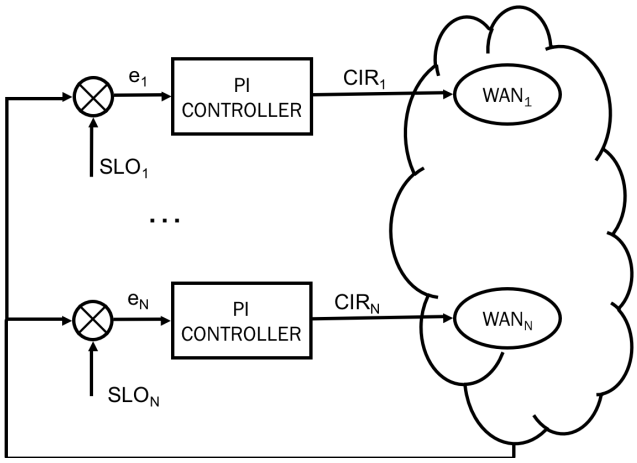

Fig. 7. The PI controller architecture implemented in the network. The PI controller detects the error from the desired reference value, i.e., Service level Objective (SLO) and provides an appropriate CIR value accordingly to reduce the impact of the attack.

C. Generation of traffic

We use VLC media player to broadcast a stream. We select the Real Time Streaming Protocol (RTSP) as the streaming method. We consider three types of video qualities a) 2K (High quality) b) 720p (high definition) and c) 480p (standard

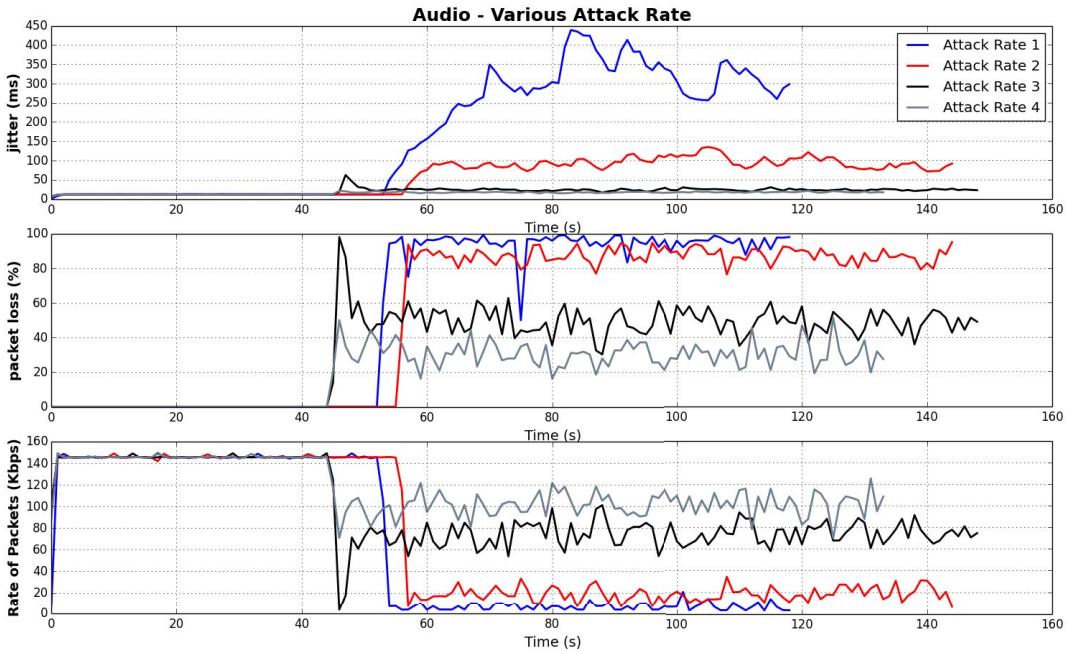

Fig. 8. (a) Jitter, (b) packet loss and (c) bitrate measured from the network before and after the attack while streaming audio. Under normal conditions, i.e., from t = 1 to t = 40 seconds, the audio is seamlessly smooth. After 40 seconds, we flood the network with different rates of attack as shown in Fig 6. We can observe the degradation of the audio towards various rates of attack. During a very high rate attack, the bitrate of the audio drops down to 1-2 Kbps. The packet loss percentage increases to 98% leading to a very high jitter value of 400 ms. We can observe that as the rate of attack decreases, the QoS metrics, i.e., packet loss percentage and jitter decreases but unacceptable.

definition) and a high-resolution audio (320 Kbps). The chosen video was streamed across the network shown in Fig 1. We use the tcpdump packet analyzer to sniff the packets at the server and client's interface, which acts as a sensor. The data such as rate of packets, inter-arrival time, etc. is collected periodically from the packet capture. We use this collected data as input into our model.

The attack rates as shown in figure 6 were used, emulating the real-world DDoS attacks as considered in [22]–[24]. The rate of the attacks is varied to exhibit the impact of the video traffic towards low-rate and high rate DDoS attacks. Some of the real DDoS attacks [24]–[28] also motivated us to model the attacks. A major amplification attack recorded by cloudflare is shown in [29]. The average packet size was 300 bytes.

IV. Analysis of Results

We study the performance of the feedback controller architecture by emulating a real-world network. The firewall and the routers are connected in a mesh topology as shown in Fig 1 and configured with OSPF protocol. We attempt to model the steady state of the network from the input-output data collected when the network is stable, i.e., when the network is under normal conditions, and any attacks are not disrupting the network services.

We then flood the network with a large number of UDP packets. The attack rate is varied by varying the packet size. Fig 6 shows the different rates of attack used to stress the network. We model these attack rates that resemble a step input as a disturbance input to the plant. When the network is under attack, the network service is hindered resulting in the degradation of the QoS. Fig 5 and Fig 8 show the effect of these attacks on video and audio traffic simultaneously. In spite of both video and audio traffic being prioritized, the attacks impact the legitimate traffic. We can observe from the Fig 5 that when the attack rate is very high, the bitrate of the video reduces resulting in more than 95% of packet loss increasing the jitter. Similarly as shown in Fig 8 the audio traffic can also be impacted by the attack increasing the jitter to nearly 400ms which is unacceptable.

A. Applying feedback

To preserve the service level agreements of the network service the feedback to the network is applied by adding a micro-firewall rule that controls the rate of traffic during an attack. Two different types of network services are considered to investigate the behavior of the feedback controller: 1) Video and 2) Audio. We also test the controller performance with various video and audio resolutions. We consider bit-rate of the video as a reference input to the plant. Bit-rate of real-time

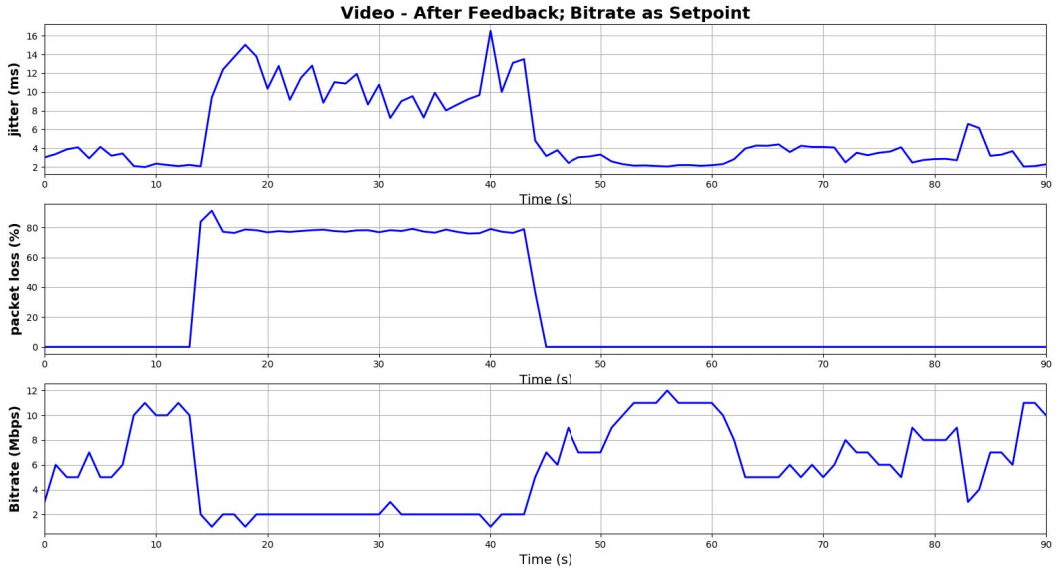

Fig. 9. (a) Jitter, (b) packet loss and (c) bitrate measured from the network before and after providing feedback. The network is attacked after 15 seconds. We can observe, when the network is under attack, the percentage of packet loss increased to 80%. Due to this high packet loss, the jitter increased by nearly 83%. At t = 40s we apply feedback to the network under attack. After the feedback is applied, the service returns to normalcy, maintaining the QoE and stabilizing the network under attack. The reference input considered in this scenario is Bitrate.

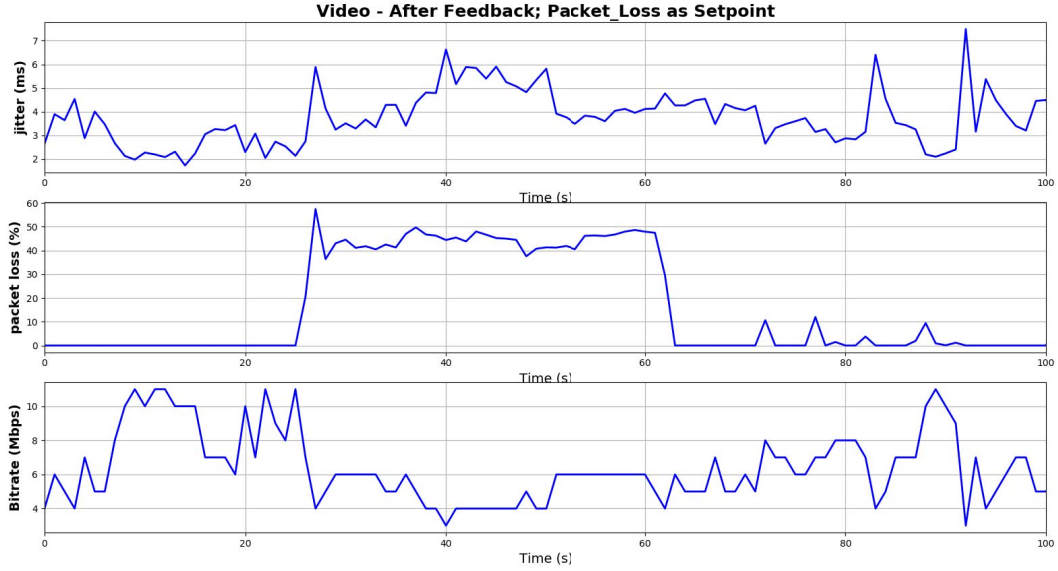

Fig. 10. (a) Jitter, (b) packet loss and (c) bitrate measured from the network before and after providing feedback. The network is attacked after 20 seconds. We can observe, when the network is under attack, the percentage of packet loss increased to 50%. Unlike in the previous scenario Fig 9, the QoS of the real-time traffic dropped by nearly 50% which is unacceptable. The feedback is applied after 60 seconds to the network under attack. In this scenario, packet loss is set as a reference input. After the feedback is applied, the service is restored.

traffic such as audio and video corresponds to the quality of the audio/video traffic, higher the bit-rate, better is the quality of the real-time traffic.

We conducted experiments by providing two different reference inputs to the controller 1) bitrate and 2) Packet Loss. We present a detail discussion about the two scenarios in the next sections.

B. Bitrate as setpoint

In this section, we explain the performance of the PI controller when the bitrate is set as a reference input to the controller. Fig 9 shows the QoS metrics (output) of video traffic collected from the network before and after introducing the feedback to the network. After 25 seconds we flood the network with a large number of packets, to emulate a DDoS attack. When the network is under attack, the nodes are congested resulting in the dropping of packets. During the attack, we can observe from the Fig 9 (b) that the measured percentage of packet loss increased to 80%. Due to a very high drop in the packets, the jitter increased by approximately 83%. This sharp increase in the percentage of packet loss and jitter causes severe deterioration of the video. We can also observe from the Fig 9 (c) that the bitrate of the video dropped to nearly 2 Mbps from 11 Mbps.

We apply the feedback to the controller after 40 seconds. We can observe that, when feedback is provided to the network after 40 seconds, the network returns to the stable state in less than 10 seconds. After implementing the feedback, the PI controller detects the drop in the bitrate of the video traffic. The controller then tunes the CIR parameter to a lower value to ensure the network service is maintained at the desired bitrate. After the controller updates the micro firewall rule into the firewall, we can observe the attack is contained and the video is restored back in real time.

C. Packet loss as setpoint

In this section, we explain the performance of the PI controller when the packet loss is set as a reference input to the controller. Fig 10 shows the output collected from the network excited with a different rate of attack. In this experiment, we flood the network with a low rate attack after 25 seconds. We then provide the feedback after 55 seconds. The controller detects the drop in the QoS values and controls the non-prioritized traffic by tuning the CIR to a lower value. In this experiment, the reference input to the controller was set to maintain the packet loss below 2%. Hence, from the Fig 10 (b) we can observe a very few percentages of packets are lost after providing the feedback.

V. Conclusion

The proposed feedback mechanism provides a real-time and scalable solution to make the network persistent during an attack and delivers video streaming without any degradation during an attack. In the conducted experiments the PI controller maintained the network service in a stable state in spite of an attack in less than 20 seconds. The closed-loop feedback mechanism designed proved that in every attack scenario, the service level agreements of the real-time services are not violated. The PI controller is designed as a disturbance rejection controller. The PI controller designed is a multiple loop SISO controller. We look to improve the controller and make the system more robust by developing a MIMO controller.

The technique implemented in the previous approach [15] brought the network back to a stable state in about 120 – 150 seconds. The current technique maintained the network service in a steady state in spite of an attack in less than 30 seconds.

The type of DDoS attack implemented in this approach is UDP flooding; such as UDP amplification attack, where a large number of UDP packets are flooded across the network, congesting the network. In our approach, the attack is not detected, although the impact of the attack is detected by the controller, i.e. the drop in the bitrate and a sharp increase in the packet loss. The video traffic is identified by the classifier in the router. The router classifies the traffic into data, voice and video traffic by using the packet classifiers and identifying a class of service parameters.

In our approach, the micro-firewall rule detects the prioritized real-time traffic. This approach is not suitable for an attack on data traffic, as the mechanism is only capable of identifying prioritized traffic and prioritizing non-real-time traffic is not an acceptable method. In our approach, we did not consider end-to-end delay and latency of the real-time traffic, which are an essential quality of service metrics because we observed a lot of artifacts due to high packet loss. Also, we only considered UDP flooding as a significant source for the DDoS attack. This approach 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.

Since attack traffic is not detected in this approach, the dropped traffic might include insignificant data traffic such as web services. This methodology holds good 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, and 911 emergency calls where packet loss cannot be tolerated.

A. Future Work

In the future, we will fine-tune the model to reduce the overshoot of the controller and also look to design more stable controller. We will also test the controller's performance with various other types of attacks such as worm propagation, and SYN flooding. We also look to design a predictive controller such as a model predictive controller (MPC) which provides the intelligence of predicting the attack. We would like to extend this closed-loop feedback mechanism in Software Defined Networks.

REFERENCES

[1] M. Moore, "Ddos attacks increase by 28 percent in q2 2017," https://betanews.com/2017/08/23/ddos-attacks-q2-2017/, (Accessed on 05/20/2018).

[2] "Ddos attacks increased 91% in 2017 thanks to iot - techrepublic," https://www.techrepublic.com/article/ddos-attacks-increased-91-in-2017-thanks-to-iot/, (Accessed on 06/21/2018).

[3] "Ddos attack trends," https://www.verisign.com/assets/infographic-ddos-trends-Q42017.pdf, (Accessed on 06/21/2018).

[4] "Github hit with the largest ddos attack ever seen — zdnet," https://www.zdnet.com/article/github-was-hit-with-the-largest-ddos-attack-ever-seen/, (Accessed on 06/21/2018).

[5] "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).

[6] C. Osborne, "The average ddos attack cost for businesses rises to over $2.5 million — zdnet," https://www.zdnet.com/article/the-average-ddos-attack-cost-for-businesses-rises-to-over-2-5m/, (Accessed on 05/20/2018).

[7] J. Vempati, M. Thompson, and R. Dantu, "Feedback control for resiliency in face of an attack," in *Proceedings of the 12th Annual Conference on Cyber and Information Security Research*. ACM, 2017, p. 17.

[8] R. Dantu, J. W. Cangussu, and S. Patwardhan, "Fast worm containment using feedback control," *IEEE Transactions on Dependable and Secure Computing*, vol. 4, no. 2, pp. 119–136, 2007.

[9] G. Tian, Y. Liu, G. Tian, and Y. Liu, "Towards agile and smooth video adaptation in http adaptive streaming," *IEEE/ACM Transactions on Networking (TON)*, vol. 24, no. 4, pp. 2386–2399, 2016.

[10] L. De Cicco and S. Mascolo, "A mathematical model of the skype voip congestion control algorithm," *IEEE Transactions on Automatic Control*, vol. 55, no. 3, pp. 790–795, 2010.

[11] ——, "An experimental investigation of the akamai adaptive video streaming," in *Symposium of the Austrian HCI and Usability Engineering Group*. Springer, 2010, pp. 447–464.

[12] R. Kuschnig, I. Kofler, and H. Hellwagner, "An evaluation of tcp-based rate-control algorithms for adaptive internet streaming of h. 264/svc," in *Proceedings of the first annual ACM SIGMM conference on Multimedia systems*. ACM, 2010, pp. 157–168.

[13] L. De Cicco and S. Mascolo, "An adaptive video streaming control system: Modeling, validation, and performance evaluation," *IEEE/ACM Transactions on Networking (TON)*, vol. 22, no. 2, pp. 526–539, 2014.

[14] X. Yin, A. Jindal, V. Sekar, and B. Sinopoli, "A control-theoretic approach for dynamic adaptive video streaming over http," in *ACM SIGCOMM Computer Communication Review*, vol. 45, no. 4. ACM, 2015, pp. 325–338.

[15] J. Vempati, R. Dantu, and M. Thompson, "Uninterrupted video surveillance in the face of an attack," in *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*, Aug 2018, pp. 843–848.

[16] L. De Cicco, S. Mascolo, and V. Palmisano, "Feedback control for adaptive live video streaming," in *Proceedings of the second annual ACM conference on Multimedia systems*. ACM, 2011, pp. 145–156.

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

[18] J. L. Hellerstein, Y. Diao, S. Parekh, and D. M. Tilbury, *Feedback control of computing systems*. John Wiley & Sons, 2004.

[19] L. Ljung, "ljung," in *Signal analysis and prediction*. Springer, 1998, pp. 163–173.

[20] "Black-box modeling - matlab & simulink," https://www.mathworks.com/help/ident/ug/black-box-modeling.html, (Accessed on 06/21/2018).

[21] "System identification toolbox - matlab," https://www.mathworks.com/products/sysid.html, (Accessed on 05/20/2018).

[22] M. H. Bhuyan, D. Bhattacharyya, and J. Kalita, "An empirical evaluation of information metrics for low-rate and high-rate ddos attack detection," *Pattern Recognition Letters*, vol. 51, pp. 1 – 7, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S016786551400244X

[23]

[24] C. Joslyn, S. Choudhury, D. Haglin, B. Howe, B. Nickless, and B. Olsen, "Massive scale cyber traffic analysis: a driver for graph database research," in *First International Workshop on Graph Data Management Experiences and Systems*. ACM, 2013, p. 3.

[25] "Udp port 1900 ddos traffic - sans internet storm center," https://isc.sans.edu/forums/diary/UDP+port+1900+DDoS+traffic/18577/, (Accessed on 08/24/2018).

[26] "sflow: Ddos," https://blog.sflow.com/2013/03/ddos.html, (Accessed on 08/24/2018).

[27] "Dns flood ddos attack hit video gaming industry with 90 million requests per second," https://thehackernews.com/2014/06/dns-flood-ddos-attack-hit-video-gaming.html, (Accessed on 08/24/2018).

[28] "What is a udp flood — ddos attack glossary — incapsula," https://www.incapsula.com/ddos/attack-glossary/udp-flood.html, (Accessed on 08/24/2018).

[29] "Reflections on reflection (attacks)," https://blog.cloudflare.com/reflections-on-reflections/, (Accessed on 08/24/2018).