2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering

Uninterrupted Video Surveillance 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

Mark Thompson
Computer Science and Engineering
University of North Texas
Denton, USA
Email: mark.thompson2@unt.edu

***Abstract*—Distributed denial of service (DDoS) attacks continue to plague businesses and consumers alike, and due to an ever-growing digital landscape, these attacks are expected to grow in size and complexity. Current mitigation techniques ranging from hours to days are completely unacceptable given the cost and inconvenience these attacks place in our society. This paper puts forth three feedback control mechanisms to minimize the effects of DDoS attacks on real-time traffic. The first, called differentiated services code point (DSCP) Markdown, is a passive approach that uses micro firewall rules to lower the priority of out-of-profile packets while a second mechanism actively drops the out-of-profile packets based on rate and burst size parameters. The third technique uses parallel links when feedback is applied to stabilize the network after an attack has been detected. Results from all three techniques have shown to have a positive effect on real-time traffic. The first two approaches were able to stabilize network traffic in real-time, while the parallel links technique resulted in a slight delay. We validate the feedback mechanisms with our model that was generated using the system identification technique. Results show that the feedback architecture provides a fit accuracy with positive results.**

***Keywords*-Resilient, DDoS, real-time services, micro-firewall, feedback control, QoS, DSCP;**

I. Introduction

Distributed denial of service (DDoS) attacks continue to be problematic for websites and service providers and thanks to the Internet of Things (IoT) supporting a rapidly growing number of network-connected devices from refrigerators to thermostats, and so on, this trend is expected to continue [1]. During the second quarter of 2017, for example, the number of DDoS attacks increased by 28 percent, with the United States being targeted over 122.4 million times [2]. One high profile DDoS attack during this period involved Microsoft's instant messaging service Skype where many users lost connectivity to the application and were unable to send or receive messages for several days with lingering connectivity issues. In response to this attack, Stephanie Weagle, VP of Corero Network Security, stated that "proactive, automated protection is required to keep the internet-connected business available in the face of DDoS attacks" [3]. In order to mitigate these growing DDoS attacks, we need to be able to identify and respond to malicious traffic immediately.

A. Motivation

Clearly, mitigating the effects of a DDoS attack in a few hours, much less in a few days, is unacceptable given the monetary impact on businesses and consumers each and every DDoS attack has. The average damage of a single DDoS attack on business has now increased to more than $2.5 million per incident [4] while the cost to launch a DDoS attack ranges from a measly $2,000 to $7,275 [5]. What's more is that DDoS attacks are quickly evolving and taking a life of their own as they are growing larger and more complex than ever. In addition, there is a burgeoning market for DDoS-as-a-Service as the sales of botnets, and DDoS tools have grown into a sizable business [1].

In our previous work, we developed a robust feedback design to maintain the stability of the network despite attacks using non-real-time (NRT) traffic [6]. We presented a passive approach to minimize the impact of DDoS on web services. As evidenced by the results, the feedback mechanism provided a positive effect on the network and stabilized the network in less than 60 seconds. The results in [6] also show a fit accuracy of approximately 75% after the feedback was provided, bringing back the unstable network to a stable state.

In this paper, we consider real-time traffic, such as streaming video, which is very sensitive to delay, packet loss, and jitter. Slight disruptions in the traffic can deteriorate streaming video. Understanding this unclear nature of real-time traffic and generating a corresponding model can be very challenging. A simple model is not adequate to identify the dynamics of such traffic. We are thus motivated to look to system identification [7] techniques to design and analyze a robust model. We select autoregressive-moving average with exogenous terms model to identify the network. We then validate the model with our robust feedback strategies. We use a micro-firewall rule that identifies and prioritizes the legitimate traffic. The firewall rule acts as feedback controller that is used to detect anomalies in the quality of the video. The importance of this mechanism is that it provides a real-time and scalable solution to deliver seamless video streaming in spite an attack.

2324-9013/18/31.00 ©2018 IEEE
DOI 10.1109/TrustCom/BigDataSE.2018.00121

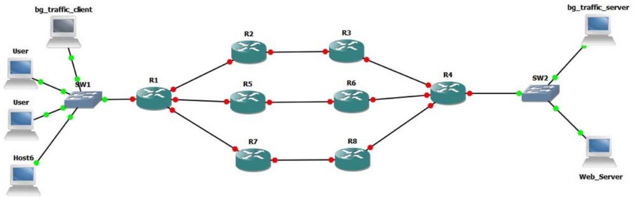

Figure 1. Experimental Network Topology: Network Elements Connected in Parallel

B. Related Work

A feedback control approach has been applied to a wide range of network systems [8]. There has been a significant amount of research on performance modeling of video streaming [7], [9]–[13]. A feedback control architecture was designed by Luca De Cicco et al., [14] for adaptive live video streaming. They propose a quality adaption controller for live adaptive video streaming. In their proposed design, the controller is fed with the length of the sender buffer as input. The sender buffer selects the video quality. The authors claim that their architecture is able to control the video level with a transient time of 30 seconds.

Chao Chen et al., [15] presented a Hammerstein-Wiener model for predicting the time-varying subjective quality (TVSQ) of rate-adaptive videos transmitted over HTTP. The authors claim that their model predicts TVSQ for the HTTP-based video streaming in real time. They also claim that their model achieves an outage rate of less than 3.4%. A non-linear autoregressive model with exogenous outputs model was proposed for the prediction of streaming video quality of experience (QoE) in [16]. Their model is driven by 3 inputs – objective measure of video quality, rebuffering-aware information and QoE memory descriptor.

Guibin Tian and Yong Liu [9] developed a video adaption algorithm for Dynamic Adaptive Streaming over HTTP (DASH). Their algorithms use client-side buffered video time as the feedback signal. They use a PI controller driven by deviation in buffered video time as the feedback signal.

These methods focus on maintaining the quality of the video at the server-side or at the client-side. To the best of our knowledge, the use of feedback control to mitigate the effect of denial of service (DoS) attack has to be yet explored for real-time traffic. Our approach has a solid defense mechanism. The feedback mechanism implemented in our approach stabilizes the network and makes the network function robustly despite the attack. Also, we prove that the QoS of the video remains the same even after the network is disrupted.

II. Architecture

A. Network Topology

We implement the same network topology used in [6]. Specifically, we implement a network connected in parallel using eight Cisco Catalyst devices configured to use the Open Shortest Path First (OSPF) protocol as shown in Fig. 1., with the default topology supporting one link R1-R2-R3-R4 connected in series. When our feedback mechanism is applied, the full network of three parallel links, each comprised of four routers connected in series, become enabled to respond to disruptions or failures in the network. Each link is configured to share the load equally, with the R4 router utilizing the load-balance feature to distribute the packets based on the destination address.

B. Client

We use the VideoLAN client (VLC) media player [17] to view the streaming videos. Using the option to stream video from the network, the streaming server's network IP is entered as the network URL. Multiple clients installed with VLC media player were used to emulate real-world scenarios.

C. Server

A standalone machine is used to host a video-streaming server to deliver real-time video content over the Internet to a user with a connected device. This system uses an Intel Xenon processor (4 cores) with 32 GB RAM. H.264/MPEG-4 AVC is used as the video coding format.

III. Methodology

A. System Identification

We use a black-box approach to identify and analyze the dynamics of the network. We consider the entire network made up of clients, a streaming server, and network devices such as routers, switches, etc. We then model this network using the system identification approach. This model describes the relationship between the measured input and output.

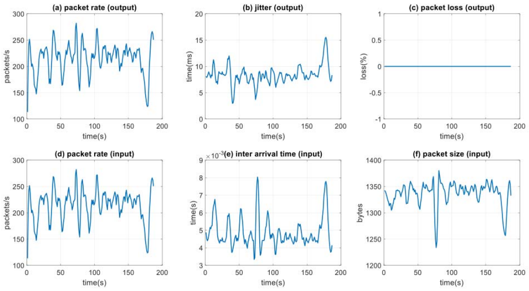

Figure 2. Input-output profile. This data is used to model the network. Figure (d), (e) (f) are input to the network and (a), (b) and (c) are the outputs collected from the network

Real-time traffic, such as video streaming, is very sensitive to jitter, delay, and packet loss. The inter-arrival time and the rate of incoming packets directly affect the QoS metrics. Hence, we select the average inter-arrival time and the rate of packets measured from the streaming server as inputs. The input is represented as u(t). The outputs of the system, represented as y(t), are the QoS metrics such as end-to-end jitter, packet loss, and the rate of RTP packets between the client-streaming server pair.

This Multi-Input Multi-Output (MIMO) black-box model is described using the System Identification Toolbox present in Matlab [18], which constructs an analytical model of the dynamic network from the observed data. This system identification technique is widely used in control engineering.

We select an autoregressive moving average with exogenous terms (ARMAX) model structure to identify the black-box model. ARMAX models are more flexible in handling disturbances [8] and are encouraged to use for time series modeling. This model structure is useful when load disturbances are present in the input.

For parameter estimation of the ARMAX model, we initially collect the input-output data for a sampling period of one second, as shown in Fig. 2. This data is split into two components; the first half used to generate the model and last to validate the model. We then use the ARMAX model order range and estimate the parameters using the System Identification Toolbox. Using the fit criteria, we select the best model and then validate it using a different set of the data sample. The order of the model is selected based on the fit accuracy.

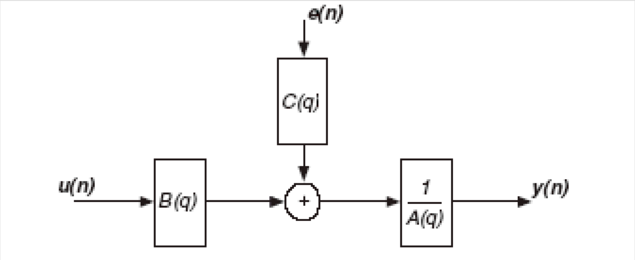

Figure 3. ARMAX general architecture

The general difference equation of ARMAX structure as shown in Fig. 3 is:

$$A(q)y(t) = B(q)u(t - nk) + C(q)e(t) \tag{1}$$

where,

y(t)— Output at time t. The orders of the A, B, and C polynomials are na, nb, and nc respectively

n_a - Number of poles.

n_b - Number of zeroes plus 1.

n_c - Number of C coefficients.

n_k - Number of input samples that occur before the input affects the output, also called the dead time in the system.

u(t - n_k) ... u(t - n_k - n_b + 1) - Previous and delayed inputs on which the current output depends.

e(t - 1) ... e(t - n_c) - White-noise disturbance value.

The parameters n_a, n_b, and n_c are the orders of the ARMAX model, and n_k is the delay. q is the delay operator. Using the ARMAX model jitter, the rate of RTP packets and packet loss can be predicted in the near future. The jitter and the arrival rate could be disturbed by the unpredictable cross traffic, making it difficult to predict in the long term. The ARMAX model generated is:

Model for output "Rate":

$A(z)y_1$(t) = - A_i(z)y_i(t) + $B(z)u(t)$ + $C(z)e_1$(t)

$A(z)$ = 1 + 0.7408z^{-1} + 0.0006638z^{-2}

A_2(z) = 0.01649z^{-1} - 0.01265z^{-2}

A_3(z) = 0

B_1(z) = 0.9892 + 0.7505z^{-1}

B_2(z) = -201.8 + 176.7z^{-1}

B_3(z) = -0.005782 + 0.006175z^{-1}

$C(z)$ = 1 + 0.8317z^{-1}

Model for output "Jitter":

$A(z)y_2$(t) = - A_i(z)y_i(t) + $B(z)u(t)$ + $C(z)e_2$(t)

$A(z)$ = 1 - 1.613z^{-1} + 0.7593z^{-2}

A_1(z) = 0.01649 z^{-1} - 0.01265 z^{-2}

A_3(z) = -7.007$e{-}25$ z^{-1} - 2.38$e{-}25$ z^{-2}

B_1(z) = -0.03476 - 0.04866z^{-1}

B_2(z) = 681.9 - 704.9z^{-1}

B_3(z) = 0.03101 - 0.02925z^{-1}

$C(z)$ = 1 + 0.03994z^{-1}

The orders of n_a, n_b, n_c, and n_k are 2, 2, 1, and 0, respectively. The mean square error of the model is 0.8733. The model yielded a fit accuracy of 98% and 78% for the rate of packets and jitter outputs, respectively.

B. Generation of traffic

We use VLC media player to broadcast a stream. We select the Real Time Streaming Protocol (RTSP) as the streaming method with the H.264 video compression codec and MP4 container format. The selected video was streamed across the network shown in Fig. 1. The tcpdump packet analyzer is used to sniff the packets at the server's interface, thus acting as the 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.

C. Feedback

We implement and analyze three different feedback mechanisms

1) DSCP Markdown: Differential Services Code Point (DSCP) is simply a measure of the QoS level of a packet. As a passive approach, this rule still allows the attack traffic but lowers the priority of the out-of-profile packets by marking them with a different QoS level and prioritizes the real-time in-profile packets.

2) Drop Out-of-Profile Traffic: As an active approach, this rule drops out-of-profile traffic based on the rate and burst size parameters. The rate defines the number of packets removed at each fixed 0.125 milliseconds interval while the burst size is the maximum number of packets that can be held by the bucket to determine whether a packet is in profile or out-of-profile.

3) Parallel links: In this passive approach, the attack traffic is still allowed, but parallel links provide additional bandwidth to stabilize the network when an attack is detected.

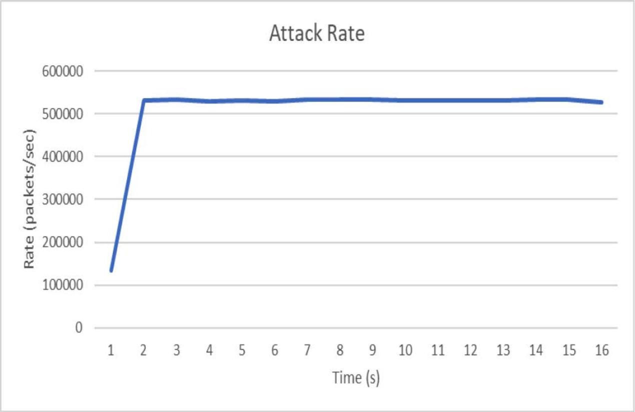

Figure 4. Arrival rate of the DoS attack. This resembles a step input

IV. Analysis of results

The feedback to the network is applied by adding a micro-firewall rule that contains two parameters rate and burst size that control the operation of policing. These parameters are selected based on the measured traffic rate. Two types of policing actions can be performed if the traffic complies with the specified profile. They are (i) dropping the out-of-profile packet and (ii) marking down the DSCP value of the packet to the one with a lower priority.

According to Cisco [19], QoS policing in the Catalyst 3550 device complies with the leaky bucket concept to determine whether a packet is considered in-profile or out-of-profile. That is, if there are enough tokens available for a packet to be transmitted, it is considered to be in-profile; otherwise, it is considered to be out-of-profile. In effect, the number of packets proportional to the traffic packet sizes is placed in a bucket so that at regular intervals, the tokens derived from the configured rate are then removed. If there is no place in the bucket to accommodate a packet, the packet is considered as out-of-profile and is dropped or marked down. These actions were applied dynamically to the network as a feedback mechanism when the network moves to an unstable state.

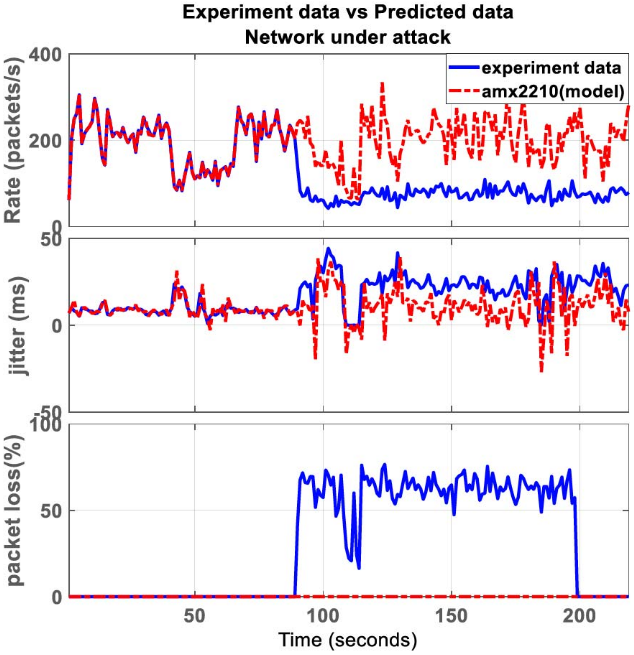

Figure 5. The experimental data does not follow the predicted data when the network is under attack. The network is attacked at t = 80s. Due to the attack, jitter and packet loss increase, resulting in the drop of the packet rate.

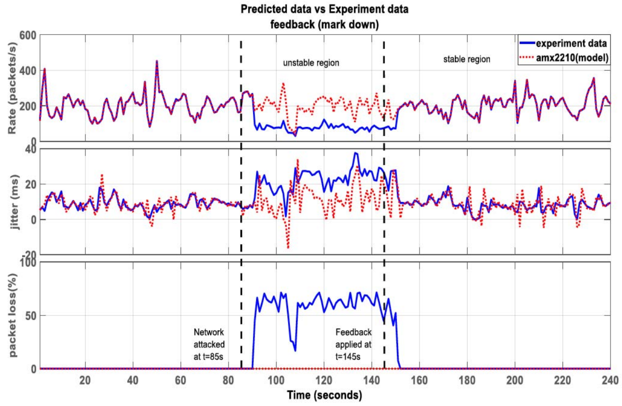

Figure 6. (a) rate of packets, (b) jitter and (c) packet loss measured from the RTP stream before and after the attack. The network is attacked at t=85s. The feedback marking-down rule is applied at t = 145s to the network under attack. We can observe, the model follows the data initially when the network is in stable state. During the attack the predicted response does not match with the experiment data. The model again closely follows the data after the feedback is introduced. This shows that the network goes to stable state from the unstable state after the feedback is applied.

A. Marking Down DSCP

The feedback rule applied in this scenario changes the DSCP value of the out-of-profile traffic to a lower priority and prioritizes the real-time traffic. Fig. 6. shows the output collected from the network before and after the attack. The

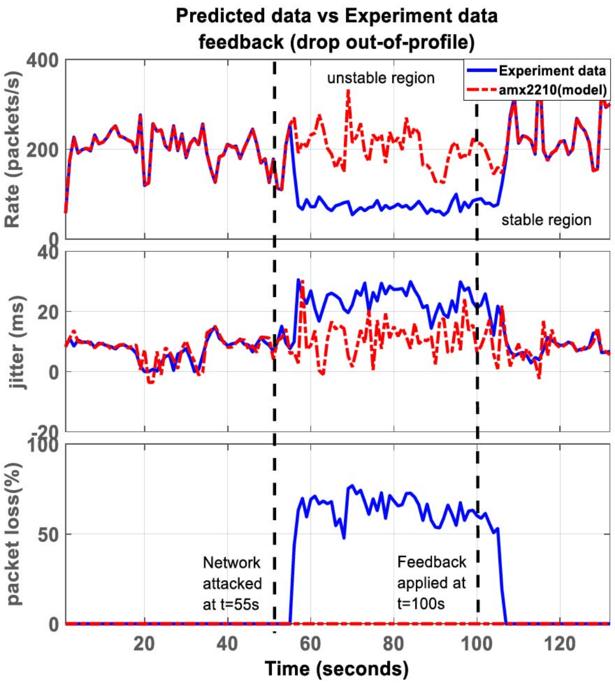

Figure 7. (a) rate of packets, (b) jitter and (c) packet loss measured from the RTP stream before and after the attack. The network was attacked at t=55s. The feedback applied which drops the traffic that does not comply with the rule, is applied at t = 100s. The model closely follows the data after the feedback is introduced and yields better fit. This shows that the network goes to stable state from the unstable state after the feedback is applied.

network is attacked after 85 seconds. The rate of attack is approximately 530,000 packets per second as shown in Fig. 4. that resembles a step input. The attack is the disturbance to the plant. The comparison of the 1-step predicted data with the experimental data when the network under attack is shown in Fig. 5.

From Fig. 6, we can observe that the jitter measured has erratic spikes and is around 38 ms. This jitter value causes severe deterioration in the quality of the video. Fig. 6. also shows the measured packet loss and the rate of RTP packets before and after the attack, respectively. When the network is under attack, the nodes appear congested. As a result, several packets would be dropped, indicating that the network is unstable. We can observe from Fig. 6. that 70 percent of the packets are dropped during the attack. Due to a very high drop in the percentage of the packets, we can observe the jitter increased by approximately 50 percent. After applying the rule of marking down out-of-profile traffic and prioritizing the real-time traffic, the network returns to a stable region after being unstable. We can observe that, when feedback is applied to our network, the jitter dropped down to nearly 12 ms with no packets lost. The feedback is applied at 145 seconds in the scenario. The network then goes from an unstable state to a stable state within 10 seconds.

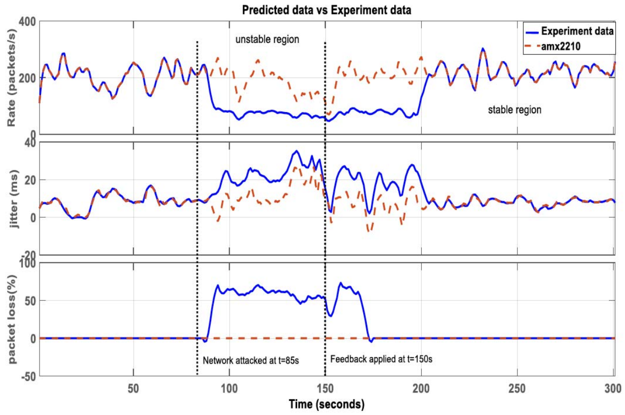

Figure 8. (a) rate of packets, (b) jitter and (c) packet loss measured from the RTP stream before and after the attack. The network is attacked at t=85.The feedback adding a parallel path is applied at t = 150s. We can observe that after the application of the feedback the network takes about 60s for the network to restore back to stable state

B. Dropping Out-of-Profile Traffic

The feedback rule applied in this scenario drops the out-of-profile traffic. Fig. 7. shows the traffic profile of the network before and after the feedback is applied. The feedback applied to the network under attack drops the traffic that does not comply with the specified profile. The real-time traffic matches with the specified rule and gets placed into the bucket. The attack traffic with a very high rate and burst size gets dropped, reducing the congestion at the router. The network is restored back to a stable state in less than 10 seconds.

C. Parallel Path

Fig. 8. shows the output traffic profile of our network before and after feedback is applied. With our network initially connected in series, feedback is applied to the network once the system becomes unstable, causing the network configuration to change from series to parallel. When the network is connected in parallel, the traffic load is shared among the links, causing a lower jitter value as well as only a minor percentage of packet loss. As a result, the quality of the video restores back to the original. The feedback is applied after 150 seconds. We can observe that after the application of the feedback the network does not restore back to the stable state, right away, as in the cases of marking down DSCP and dropping out-of-profile packets. Since it takes approximately 60 seconds to establish a link, the network restores back to stable state after t=200s

V. Conclusion

Our feedback mechanism has a positive effect on the real-time traffic. The network goes to the stable state in real-time in all the scenarios except for the parallel path where there is a slight delay. The delay is due to the time taken for the link to be enabled. In this approach, we did not consider the RTP delay, which is an important quality of service parameter for video streaming, as we did not observe a drastic change in the delay. We used a UDP flood attack to emulate the DDoS attack. In our approach, the micro firewall rule detects the known traffic and allows the video streaming seamlessly despite the attack. This approach is not suitable for data traffic as the rate of data is unpredictable; also, prioritizing the data traffic is not an acceptable approach. Although we did not actively stop the attack, except for dropping the out-of-profile traffic scenario, we were able to use a passive approach to bring the network back to a stable state.

A. Future Work

In the future, we will design the controller to automate the feedback process. The controller will detect the irregularities in the output and add the corresponding rule to ensure seamless video streaming. We will test this feedback mechanism with several other types of DDoS attacks such as SYN flood and other application layer attacks, similar to the one considered in [6], on the Real-Time services. We would like to extend this feedback mechanism in the Software Defined Networks (SDN) architecture where control theory proves to be more effective.

References

[1] "Three ways ddos attacks are evolving in 2017 — liquid web," https://www.liquidweb.com/blog/three-ways-ddos-attacks-evolving-2017/, (Accessed on 05/20/2018).

[2] 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).

[3] C. Page, "Skype outage to blame on 'ddos attack', hacking group claims responsibility," https://www.theinquirer.net/inquirer/news/3012275/skype-outage-ddos-attack, (Accessed on 05/20/2018).

[4] 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).

[5] K. Coleman, "Ddos attacks grow with a new twist! — 2017-07-10 — security magazine," https://www.securitymagazine.com/blogs/14-security-blog/post/88141-ddos-attacks-grow-with-a-new-twist, (Accessed on 05/20/2018).

[6] 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.

[7] 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.

[8] "Selecting a model structure in the system identification process - national instruments," http://www.ni.com/white-paper/4028/en/, (Accessed on 05/20/2018).

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

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] C. Chen, L. K. Choi, G. de Veciana, C. Caramanis, R. W. Heath, and A. C. Bovik, "Modeling the time—varying subjective quality of http video streams with rate adaptations," *IEEE Transactions on Image Processing*, vol. 23, no. 5, pp. 2206–2221, 2014.

[16] C. G. Bampis, Z. Li, and A. C. Bovik, "Continuous prediction of streaming video qoe using dynamic networks," *IEEE Signal Processing Letters*, vol. 24, no. 7, pp. 1083–1087, 2017.

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

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

[19] "Understanding qos policing and marking on the catalyst 3550 - cisco," https://www.cisco.com/c/en/us/support/docs/switches/catalyst-3550-series-switches/24800-153.html, (Accessed on 05/20/2018).