Real-Time Detection of Packet Drop Attacks on Synchrophasor Data

Seemita Pal, Biplab Sikdar and Joe Chow
Department of Electrical, Computer and Systems Engineering
Rensselaer Polytechnic Institute, Troy, NY, 12180

***Abstract*—The importance of phasor measurement unit (PMU) or synchrophasor data towards the functioning of real-time monitoring and control of power generation and distribution systems makes them an attractive target for cyber-attacks. An attack with potential for significant damage is the packet drop attack, where the adversary arbitrarily drops packets with synchrophasor data. This paper develops a real-time mechanism for detecting packet drop attacks on synchrophasor data carried over the Internet. The proposed solution is receiver-based, and uses the one-way packet delays to extract features that are used to detect attacks. The proposed attack detection mechanism leads to lower detection delays and greater accuracy as compared to existing mechanisms.**

I. Introduction

Measurements from synchrophasors facilitate a number of applications in smart grids such as real-time system monitoring, state estimation, disturbance monitoring, instability prediction, wide area protection and control, etc. [1], [2]. Since synchrophasor measurements are typically transferred over the public Internet, they are susceptible to a number of cyber-attacks, with potentially serious consequences. This paper addresses the problem of detecting packet drop attacks on synchrophasor data. In such attacks, the adversary drops packets from the stream of synchrophasor data, for example by gaining control of routers or by launching a denial-of-quality attack on a router. Our objective is to develop a real-time, zero-overhead mechanism for detecting packet drop attacks that does not rely on any network assistance.

The data reported by synchrophasors includes the frequencies, voltage and current phasors, with accuracy better than 0.1% [8]. The synchrophasor data have highly accurate time-stamps that are usually obtained through in-built global positioning system (GPS) receivers and have precision better than 1 μs. Loss of synchrophasor data results in inaccuracies in the estimated system state and can easily lead to loss of observability of the system. The performance of other applications that use synchrophasor data such as inter-area oscillation, wide area monitoring and control is also adversely affected by loss of data. Thus packet drop attacks on synchrophasor data are an attractive avenue for cyber-attacks on smart grids.

Existing work on the detection of packet drop attacks (also called gray-hole attacks) is primarily in the context of wireless ad hoc networks [3], [4], [5]. In wireless networks, the detection of malicious packet drops is usually achieved by monitoring the transmissions of neighboring nodes. The broadcast nature of wireless transmissions facilitates the development of mechanisms for the detection of packet drop attacks, for example by using statistics such as the fraction of packets forwarded by a node. For wired networks such as the core of the Internet, [6] presents a network-assisted mechanism for detecting packet drop attacks where routers in the network cooperate to provide real-time network data for attack detection. In addition to added overheads, this mechanism also has the drawback that the network information provided by the routers may be compromised during an attack. In [7] a mechanism for detecting packet drop attacks that does not need any network support has been proposed. However, this mechanism has relatively high detection delays.

This paper presents a real-time mechanism for detecting packet drop attacks on synchrophasor data carried in the Internet. The fundamental challenge in the detection of malicious packet drops is to distinguish such drops from those that occur naturally in the network (due to congestion). The proposed mechanism for detecting packet drop attacks uses features that are extracted from the one-way delays experienced by the synchrophasor data packets. The effectiveness of the proposed detection mechanism has been validated through extensive simulations.

The rest of this paper is organized as follow. Section II presents the background material and an overview of the system model. Section III presents the proposed packet drop attack detection mechanism and Section IV presents simulation results to validate the proposed detection mechanism. Finally, Section V concludes the paper.

II. Background and System Model

This section provides an overview of packet drop attacks as well as the network and threat models assumed in this paper.

A. Packet Drop Attacks

Cyber-attacks on computer networks can be broadly classified into two categories: active and passive. Passive attacks are more benign since in these attacks the data is not modified. Instead, the adversary is interested in learning the characteristics of the network or the data being transferred. On the other hand, an adversary in an active attack manipulates the data or the equipment in the network and thus has the potential to inflict greater damage. Packet drop attacks fall in the category of active attacks. As the name suggests, the adversary in a packet drop attack causes the drop of packets in the network. However, not all packets in a link are usually dropped, since such attacks are easily detected [9]. Thus packet drop attacks are usually associated with scenarios where only a subset of the packets is dropped, thereby making it more difficult to detect. Also, the packets to be dropped may be chosen arbitrarily.

The primary challenge in detecting packet drop attacks is to distinguish the malicious drops from those occurring naturally in the network due to congestion. In a large-scale network such as the Internet, the random variations in the number of flows and their data rates makes it difficult to distinguish malicious packet drops from those due to congestion. Due to the scale and the overheads involved, using explicit network assistance for attack detection is not an option. Consequently, the effective and real-time detection of packet drop attacks remains an open problem.

B. Network Model

This papers assumes a network with an an arbitrary topology and an arbitrary number of nodes. Each synchrophasor sends its data to a Phasor Data Concentrator (PDC) over the network. The synchrophasor data to the PDC passes through a number of routers in the network and each router may have a different number of flows passing through it at any given point in time. We assume that an arbitrary number of the routers in the network may have been compromised or subjected to attacks that lead to packet drops from the synchrophasor data streams. The data generated by the synchrophasor is periodic and each packet is of the same size. We assume that the synchrophasor data is transferred using the User Datagram Protocol (UDP) as the transport layer protocol. UDP is used as the transport layer protocol due to the time-sensitive nature of the synchrophasor data and unlike Transmission Control Protocol (TCP), UDP does not stop to recover lost packets or slow down its transmissions in response to congestion. If TCP were used for transferring the PMU data, in the event of a packet loss, the PDC would not receive any new data till the lost packet is recovered. For real-time applications with strict deadlines on the data arrival time, such delays are unacceptable and may lead to interruptions in the monitoring and control applications.

In addition to the synchrophasor data, the network also carries data from other sources. These sources of traffic may use either TCP or UDP as their transport layer and the number of flows may vary with time. The bottleneck link is thus not fixed and the level of congestion experienced by a synchrophasor data stream is different on each hop of its path. Also, since the different flows sharing a bottleneck link may have different round trip times, the reaction time of each flow to a congestion event may be different.

C. Threat Model

The threat model assumed in this paper is that the adversary has compromised one or more routers in the network and has the ability to arbitrarily drop any packet that passes through these routers. In addition to dropping synchrophasor data, the adversary may also drop packets from other flows in order to make the attack harder to detect. The synchrophasor data is assumed to be encrypted and the adversary does not alter the contents of the packets. The adversary may indirectly affect the behavior of the other flows in the network. For example, the adversary may be able to affect the rate of packet transmissions of TCP flows by dropping their packets.

Under the adversary model described above, the paper considers the following packet drop attack: in a network with arbitrary topology and traffic characteristics, the adversary compromises a set of routers and arbitrarily drops packets that pass through these routers. The accuracy and the effectiveness of the applications that use the synchrophasor data depends on the timely availability of the measured data. To maximize the damage to these applications, the objective of the adversary is to drop the largest possible number of packets from the synchrophasor flow without being detected. Our objective is to develop a real-time mechanism to detect packet drop attacks and the primary concern is to distinguish malicious packet drops from those dropped due to congestion.

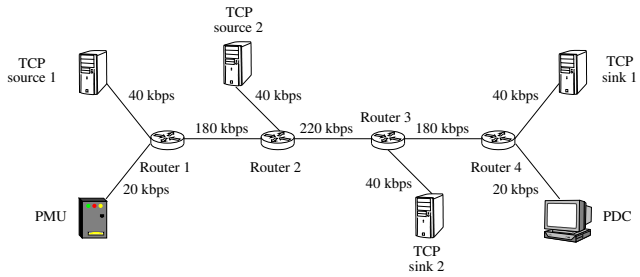


Fig. 1. Example Network Topology: one synchrophasor flow and two TCP flows.

III. Proposed Methodology for Detecting Packet Drop Attacks

In this section, we present our strategy to detect packet drop attacks. We first motivate the features used in detection mechanism and then present the details of the methodology for detecting packet drop attacks.

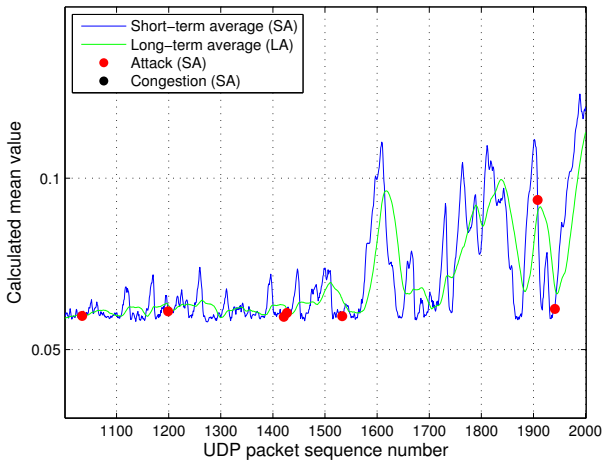
A. Correlation Between Congestion and Delay

Congestion related packet drops in the Internet are caused by overflowing buffers at the routers. As the buffer occupancy of the routers increases during the onset of congestion, a corresponding increase in the queuing delay is observed. On the other hand, such correlations are unlikely to occur when an attacker maliciously drops randomly selected packets from the synchrophasor data flow. This correlation in the packet delays and congestion forms the basis of the proposed mechanism for detecting packet drop attacks.

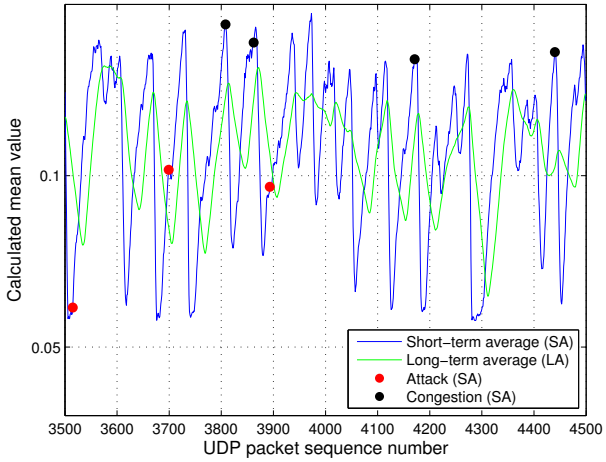
To capture the correlation between the packets delays and congestion, this paper uses three metrics, adapted from [10]. To illustrate these metrics, we use simulations conducted using the NS2 network simulator [11], using the topology in Figure 1. In this example scenario, the network has one synchrophasor flow that shares the network with other TCP flows. The number of TCP flows is varied to create networks with different congestion levels.

The first metric used in this paper is the Correlation Indication Metric (CIM) which counts the number of times the ratio of the short-term average $x_m(i)$ of one-way latencies of m packets preceding the i-th packet to the sum of the long-term average $x_n(i)$ and standard deviation $sdev_n(i)$ of preceding n packets is greater than 1, where $n>m$. In other words, CIM is the number of occurrences where the ratio, i.e.

$$\frac{x_m(i)}{x_n(i) + sdev_n(i)} \geq 1$$



(a) Attack drops in a lightly-congested network



(b) Congestion and attack drops in a congested network

Fig. 2. Short-term average and long-term average of the packet delays.

before a packet drop occurs due to congestion or attack.

The short-term average latency represents the instantaneous average value of the latencies and the difference between the short-term and the long-term averages is reflective of the change in the queueing delays observed by the packets at the routers of a network with dynamically changing traffic. As network traffic increases, the delays seen by packets keep increasing till one or more packets get dropped due to congestion. This results in less queuing time for the subsequent packets. In addition, the TCP clients trigger their congestion control mechanism which causes a sharp reduction in the delays observed by the later packets. Therefore, for a packet dropped due to congestion, the mentioned ratio is greater than 1 for a window of preceding packets. Thus, $CIM{>}\alpha$ for congestion drops and $CIM{<}\alpha$ for attack drops, where α is the CIM threshold that can be estimated emperically without difficulty. A sharp decrease in the ratio is also generally observed after packet drop due to congestion.

The CIM metric helps in quantifying how effectively the network state can be determined using packet latencies. Sample plots showing the variation of the short-term average and the

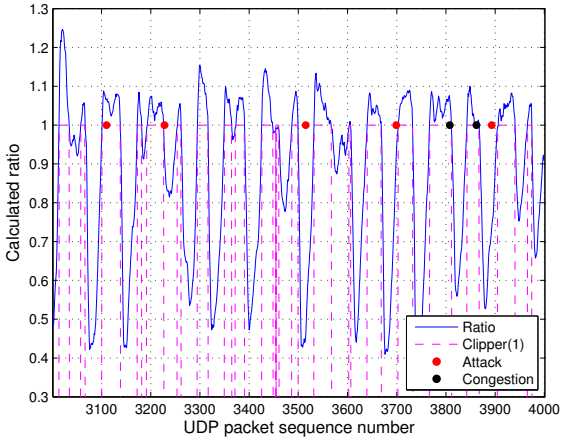


Fig. 3. CIM values observed in a congested network.

long-term average latencies with packet sequence number for both lightly and heavily congested networks are shown in Figure 2. The red circles indicate packets dropped by attacker while the black circles indicate congestion drops. It is clear from the figure that the packet drops caused by congestion occur at the various peaks of the short-term average curve and typically before the corresponding peak in the long-term average curve. Figure 3 plots the values of the ratio against the corresponding packet sequence numbers in the case of a congested network. The curve has been clipped at the value of 1 to provide an idea of the CIM values in a graphical manner. It is clear that CIM values for packets dropped by congestion are typically greater than that of packets dropped by attacker. Also, immediately after a packet drop, the ratio falls very rapidly.

The second metric used in this paper is the Loss Conditioned Delay Correlation (LCDC) metric that calculates the average and standard deviation of the one-way latencies of the PMU packets whose j^{th} preceding (j is negative) or following (j is positive) packet is dropped. An example of the LCDC metric is shown in Figure 4. As can be seen, the one-way packet latency starts increasing when network congestion sets in and reaches its peak at which point one or more packets get dropped due to buffer overflow. The packet losses trigger the congestion control mechanisms in the TCP sources, causing a sharp decrease in the subsequent latency values.

The third metric considered in this paper is the loss conditioned delay cumulative distribution function (CDF) metric. In this metric, the CDF of the short-term average latency is compared with the CDF of the long-term average latency, separately for packets dropped by congestion and those dropped by the attacker. A comparison of the CDFs is shown in Figure 5. It can be seen that packet drops due to congestion do not occur below a threshold value of observed packet latency. Any packet drop occurring when the receiver-observed latencies are low is an indication of the presence of a malicious attacker.

B. *Classification and Detection Mechanism*

In order to develop a detection mechanism that can accurately detect the presence of an attacker dropping synchrophasor data packets, we first need to classify the cause of a packet

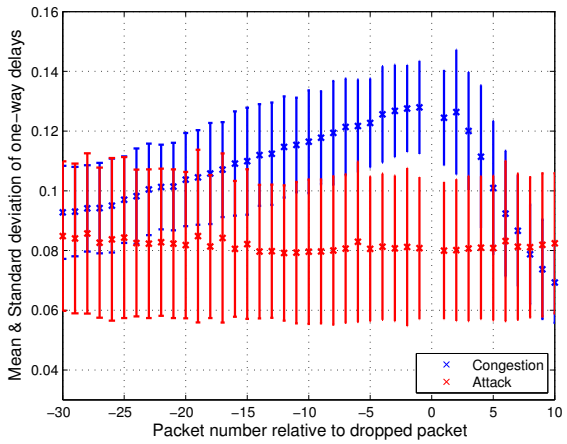


Fig. 4. LCDC for packet drops (the standard deviation is indicated by the error bars) caused by congestion and attack.

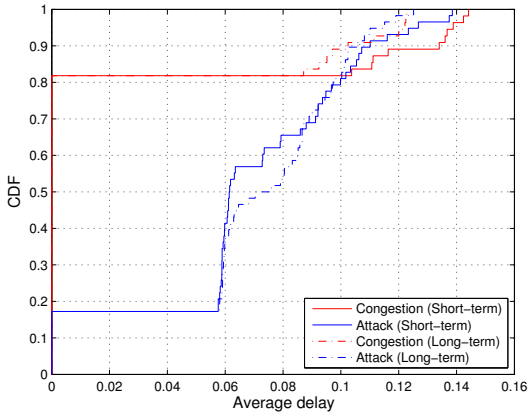


Fig. 5. Comparison of the CDFs of the short-term and long-term average latencies.

loss as either due to congestion or due to an attacker. We use the results of the above three metrics to develop a classifier which can effectively classify each lost packet according to cause even in a dynamically changing network. The proposed mechanism does not use any network support and is executed at the receiving end point of the synchrophasor data (such as a PDC).

At the PMU, each outgoing synchrophasor data packet is time-stamped by the GPS before transmission. On arrival at the receiving end or the PDC, the latencies of each of the incoming packets are calculated. The online detection mechanism based on these calculated latencies can enable us in early detection of packet drop attacks. In general, the first or an isolated occurrence of a packet loss that is classified as an attack is not sufficient to declare the onset of an attack with a high level of confidence (due to non-zero false positive rates of the classifier). Thus the attack detection mechanism relies on testing whether the number of dropped PMU packets classified as attack drops over a given interval of time exceeds a particular threshold. If this threshold is exceeded, only then a packet drop attack alarm is generated.

The proposed packet drop attack detection algorithm is shown in Algorithm 1. The proposed detection mechanism keeps a log of the individual one-way packet latencies as well as the short-term and long-term average latencies of packets preceding a particular packet (say i). If D_i is the latency of the i^{th} packet then the short-term average delay ($savg_i(m)$) and the long-term average delay ($lavg_i(n)$) with window sizes of m and n respectively can be defined as

$$savg_i(m) = \sum_{j=1}^{m}(D_{i-j}/m) \tag{1}$$

$$lavg_i(n) = \sum_{j=1}^{n}(D_{i-j}/n) \tag{2}$$

where, $m \leq n$.

Algorithm 1 Packet Drop Attack Detection Algorithm

initialize $x_n = 0$
function CLASSIFY(i)
  $slope_I(i) = (savg_i - savg_{i-k})/k$
  $slope_F(i) = (savg_{i+l} - savg_{i+p})/(l-p)$
  if $savg_i - lavg_i \leq (0.005 lavg_i)$ **then**
    $cause = attack$;
  else if $slope_I(i) - slope_F(i) < 0$ **then**
    $cause = attack$;
  else
    $cause = congestion$;
  end if
  if $cause == attack$ **then**
    ALERT()
  end if
end function

function ALERT()
  if $x_n = 0$ **then**
    start $timer$ for value t;
  end if
  update $x_n = x_n + 1$;
  if $timer$ has expired **then**
    if $x_n > \eta$ **then**
      generate alarm for "Packet Drop Attack";
    end if
    $x_n = 0$;
  end if
end function

loop
  for each packet arrival i **do**
    calculate delay D_i;
    consider n previous arrivals;
    calculate $lavg_i = \sum_{j=1}^{n}(D_{i-j}/n)$;
    consider m previous arrivals;
    calculate $savg_i = \sum_{j=1}^{m}(D_{i-j}/m)$;
    if out-of-sequence packet **then**
      {/* packet loss detected */}
      wait for l new arrivals;
      CLASSIFY(i)
    end if
  end for
end loop when session is terminated

On the arrival of each new packet, the algorithm calculates the above two quantities corresponding to that particular packet. In case a packet loss is detected (i.e. an out of sequence packet is received), the algorithm calculates the trend in the short-term average delays before the packet loss and waits for the next l packets. It uses the delay values associated with these succeeding packets to determine the trend in the short-term average delays after the loss. The slopes of the short-term average delay before and after the packet drop are denoted by $slope_I$ and $slope_F$ respectively and are defined as follows:

$$slope_I(i) = (savg_i - savg_{i-k})/k \tag{3}$$

$$slope_F(i) = (savg_{i+l} - savg_{i+p})/(l-p) \tag{4}$$

Keeping the value of p equal to m ensures that delays of preceding packets are not included in the computation of $slope_F$. Based on the average values and the calculated slopes $slope_I$ and $slope_F$, the classifier considers two cases and performs classification.

In a normal network scenario without any attacker, the onset of congestion is accompanied by a nearly steady increase in the packet latencies which will result in a positive value of $slope_I$. A negative $slope_I$, on the other hand, indicates that the instantaneous average delays experienced by the packets in transit was not increasing and possibly the network was not moving towards congestion.

When TCP flows lose packets due to congestion, they react by either decreasing the congestion window to half of the current value in case of triple duplicate acknowledgements (ACKs) or by decreasing it to 1 maximum segment size (MSS) in case of timeout. Thus, there is a sharp drop in the delays of the packets succeeding the dropped packet resulting in a negative value of $slope_F$. A positive value of $slope_F$, however, indicates that the delays kept increasing even after the packet drops occurred and indicates the involvement of a malicious attacker in causing the drops. Thus in case of congestion, the difference of $slope_I$ and $slope_F$ should be greater than zero.

Long-term average latency serves as the reference delay value in an ever-changing network. If the short-term average latency is much higher than this reference at the time of packet drop then a considerable increase in traffic is suggested, thus indicating possible congestion. Therefore, when a packet drop occurs due to congestion, the short-term average latency generally exceeds the corresponding long-term average by a certain threshold.

Thus, based on the values of $savg_i(m)$, $lavg_i(n)$, $slope_I$ and $slope_F$ we have the following cases:

1) If $savg_i - lavg_i \leq (0.005 lavg_i)$ then the cause of the packet loss is marked as attack.
2) If $slope_I(i) - slope_F(i) < 0$ then the cause of packet loss is marked as attack.
   Otherwise, it is classified as congestion.

Thus, based on the proposed classification scheme, we classify the cause of each of the lost synchrophasor packets as "congestion" or "attack". On the observation of the first packet classified as "attack", the system goes in the *alert mode* and initializes an alarm clustering timer of t seconds. In this alert mode, the system counts the number of dropped packets classified as attack, denoted by x_n. The presence of an attacker is confirmed and a system-wide packet drop attack alarm is generated if x_n exceeds the alarm threshold (η) before the expiration of the timer. If the threshold is not exceeded when the timer expires, then the classifier resets x_n, keeps monitoring packet drops, and repeats the entire process on observation of the next dropped packet classified as "attack". Thus,the maximum detection delay possible using this scheme is t sec in the worst case while the presence of false alarms is almost eliminated.

IV. Simulation Results

In this section, we present simulation results to evaluate the effectiveness of the proposed packet drop attack detection mechanism. The simulations were conducted using the Network Simulator 2 (NS2) simulation tool. For our results, we consider multihop network topology and vary the number of flows in the network to create different levels of congestion. In addition, we consider attacks of different intensities where the adversary drops a different fraction of the packets that traverse the compromised router. Each simulation scenario consists of one synchrophasor flow and a number of TCP flows. The synchrophasor flow uses UDP as the transport layer protocol and generates 20 packets per second and each packet is of 100 bytes. The length of each simulation run was kept at 1000 seconds and each reported result is for the average of 3 different runs for the same scenario. In these simulations, we assume that the adversary drops packets from the PMU as well as the TCP flows in order to make the detection of the attack more difficult. The simulations used $\eta = 3$, $n = 35$, $m = p = 3$, $k = 8$ and $l = 10$ for the detection algorithm. These values remain unchanged for the same topology inspite of the changing nature of the network. For different topologies, they can be further fine-tuned.

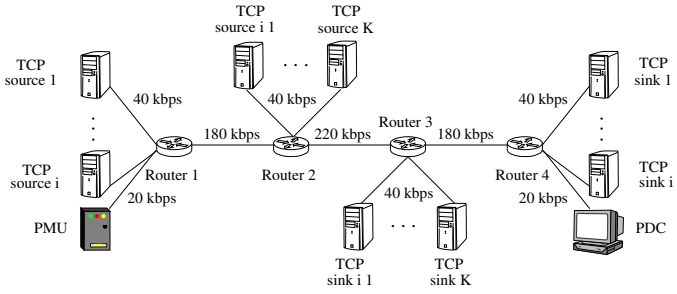


Fig. 6. Network topology with multiple bottlenecks: one synchrophasor flow and multiple TCP flows.

The performance of the proposed classification mechanism is evaluated in terms of its accuracy, false positive rates, and false negative rates. The false positive rate is the probability that a packet drop is categorized as an "attack" drop when the real cause of the drop is congestion. The false negative rate is the probability that the cause of a packet drop is categorized as "congestion" when the actual cause is attack. In addition, we also evaluate the accuracy of the proposed detection scheme. The accuracy of the system is defined as the fraction of packet losses whose cause is correctly classified.

The simulated network with multiple bottlenecks is shown in Figure 6. In this topology it is assumed that Router 2 is

TABLE I. ACCURACY, FALSE POSITIVE AND FALSE NEGATIVE PERCENTAGES OF PACKET DROP ATTACK DETECTION IN A NETWORK WITH 12 TCP FLOWS FOR DIFFERENT PACKET DROP-RATES AND DETECTION DELAYS. A: ACCURACY, FP: FALSE POSITIVE, FN: FALSE NEGATIVE

Drop rate	60 sec			90 sec			120 sec		
	A	FP	FN	A	FP	FN	A	FP	FN
0.00	100	0	0	100	0	10	100	0	0
0.005	91.33	0	8.66	97.88	0	2.12	100	0	0
0.0075	97.73	0	2.27	100	0	0	100	0	0
0.01	99.03	0	0.97	100	0	0	100	0	0
0.015	100	0	0	100	0	0	100	0	0
0.02	100	0	0	100	0	0	100	0	0

compromised and the attacker drops packets from all the flows that pass through it. The propagation time of all links in the network was 10ms, and each router had a buffer capacity of 60 Kbits. Scenarios with different levels of congestion are constructed by choosing the total number of TCP flows in the network (K in the figure) as 4, 6, 8, 10 and 12, and the corresponding number of long TCP flows (i in the figure) as 3, 4, 5, 6 and 7, respectively. In addition, we simulated different intensities of of packet drop attacks by simulating attacker drop rates of 0.005, 0.0075, 0.01, 0.015 and 0.02, and also the scenario when there is no attacker.

Table I presents the results of the detector for the case of 12 TCP flows in the network, for different values of the alarm clustering timer t. For all cases, we observe that the proposed classification scheme can classify, with a high level of accuracy, the cause of any packet drop in a network. The attack detection scheme based on this classification mechanism thus yields very accurate results. Similar results were obtained for the scenarios with other numbers of TCP flows and these results have been omitted. It is seen that as the attacker increases the rate of packet drop, the accuracy increases and the detection delay decreases. Since the attacker will try to drop the maximum possible number of packets in order to cause damage to the system, the proposed detection mechanism will be particularly accurate in real-life scenarios. For the chosen alarm threshold ($\eta = 3$), the average detection delay for the case of 0.005 drop rate (irrespective of the number of TCP flows) is observed to be approximately 30 seconds. As the attacker increases the drop rate, the detection delay becomes smaller and smaller, or in other words, detection is quicker.

TABLE II. OVERALL DETECTION RESULTS USING PROPOSED DELAY-BASED ALGORITHM AND PREVIOUS ALGORITHM. A: ACCURACY, FP: FALSE POSITIVE, FN: FALSE NEGATIVE

Timer value	Proposed algorithm			Algorithm in [7]		
	A	FP	FN	A	FP	FN
60 sec	97.93	0	2.47	90.35	4	5.65
90 sec	99.65	0	0.354	93.265	5.67	1.07
120 sec	100	0	0	99.27	0	0.89

The overall results of the detector for all the different TCP cases and attack intensities are presented in Table II. These results correspond to the averaged results for all the choices of K (the number of TCP flows) and the attacker drop rates. It is seen that the accuracy is 100 percent when the alarm clustering timer is set to 120 sec i.e. 2 min. Note that the detection scheme proposed in [7] has a detection delay of 5 minutes in order to obtain nearly 100% accurate results. The overall detection results for the same simulation settings for the scheme in [7] are also presented in Table II.

V. CONCLUSIONS

This paper presented a real-time mechanism to detect packet drop attacks in networks carrying sensitive synchrophasor data. The proposed methodology utilizes the analysis of one-way packet delays and the results of the three discussed metrics in order to develop an online packet drop detection mechanism that performs early detection without raising false alarms. Simulation results are presented to verify the performance of the proposed algorithm.

REFERENCES

[1] S. Horowitz, A. Phadke and B. Renz, "The Future of Power Transmission," *IEEE Power and Energy Magazine,* vol.8, no.2, pp.34-40, March-April 2010.

[2] R. Burnett, M. Butts and P. Sterlina, "Power system applications for phasor measurement units," *IEEE Computer Applications in Power,* vol. 7, no. 1, pp. 8-13, January 1994.

[3] D. Djenouri, L. Khelladi and A. Badache, "A survey of security issues in mobile ad hoc and sensor networks, *IEEE Communications Surveys and Tutorials,* vol. 7, no. 4, pp. 2-28, 2005.

[4] B. Kannhavong, H. Nakayama, Y. Nemoto, N. Kato and A. Jamalipour, "A survey of routing attacks in mobile ad hoc networks, *IEEE Wireless Communications,* vol. 14, no. 5, pp. 85-91, October 2007.

[5] J. Cai, P. Yi, J. Chen, Z. Wang and N. Liu, "An Adaptive Approach to Detecting Black and Gray Hole Attacks in Ad Hoc Network," *Proc. of IEEE AINA,* pp. 775-780, Perth, Australia, April 2010.

[6] A. Mizrak, S. Savage and K. Marzullo, "Detecting Malicious Packet Losses," *IEEE Transactions on Parallel and Distributed Systems,* vol. 20, no. 2, pp. 191-206, February 2009.

[7] S. Pal, H. Li, B. Sikdar and J. Chow, "A mechanism for detecting Gray Hole Attacks on Synchrophasor Data," *IEEE ICC 2014* (accepted).

[8] A. Armenia and J. Chow, "A Flexible Phasor Data Concentrator Design Leveraging Existing Software Technologies," *IEEE Transactions on Smart Grid,* vol. 1, no. 1, pp. 73-81, June 2010.

[9] X. Zhang, S. F Wu, Z. Fu, and T-L Wu, " Malicious Packet Dropping: How It Might Impact the TCP Performance and How We Can Detect It,"*Proc. of ICNP,* pp. 263-272, November 2000.

[10] J. Martin, A. Nilsson and I. Rhee, "Delay-Based Congestion Avoidance in TCP," *IEEE/ACM Transactions on Networking,* vol.11, no.3, pp.356-369, June 2003.

[11] The Network Simulator. Univ. California, Berkeley, CA. [Online]. Available: http://www-mash.cs.Berkeley.EDU/ns/