Intrusion Detection System for PS-Poll DoS Attack in 802.11 Networks Using Real Time Discrete Event System

Mayank Agarwal, *Student Member, IEEE*, Sanketh Purwar, Santosh Biswas, *Member, IEEE*, and Sukumar Nandi, Senior *Member, IEEE*

***Abstract*—Wi-Fi devices have limited battery life because of which conserving battery life is imperative. The 802.11 Wi-Fi standard provides power management feature that allows stations (STAs) to enter into sleep state to preserve energy without any frame losses. After the STA wakes up, it sends a null data or PS-Poll frame to retrieve frame(s) buffered by the access point (AP), if any during its sleep period. An attacker can launch a power save denial of service (PS-DoS) attack on the sleeping STA(s) by transmitting a spoofed null data or PS-Poll frame(s) to retrieve the buffered frame(s) of the sleeping STA(s) from the AP causing frame losses for the targeted STA(s). Current approaches to prevent or detect the PS-DoS attack require encryption, change in protocol or installation of proprietary hardware. These solutions suffer from expensive setup, maintenance, scalability and deployment issues. The PS-DoS attack does not differ in semantics or statistics under normal and attack circumstances. So signature and anomaly based intrusion detection system (IDS) are unfit to detect the PS-DoS attack. In this paper we propose a timed IDS based on real time discrete event system (RTDES) for detecting PS-DoS attack. The proposed DES based IDS overcomes the drawbacks of existing systems and detects the PS-DoS attack with high accuracy and detection rate. The correctness of the RTDES based IDS is proved by experimenting all possible attack scenarios.**

***Index Terms*—Fault detection and diagnosis, intrusion detection system (IDS), null data frame, power save attack, PS-Poll frame, real time discrete event system (DES).**

I. Introduction

WI-FI is ubiquitous now-a-days. As wireless communication happens over the air, eavesdropping Wi-Fi communication is easy. An attacker needs to be in the vicinity of the target STA in order to eavesdrop on its communication. Encryption schemes, like wired equivalent privacy (WEP), Wi-Fi protected access (WPA), WPA2, encrypt only data frames. The management and control frames are sent in clear-text even on encrypted networks. Sending clear-text frames makes processing easier and faster. The management and control frames can be easily spoofed and injected into the network using variety of tools like aircrack-ng suite [1], scapy [2] etc., various denial of service (DoS) attacks [3] on 802.11 Wi-Fi network like de-authentication DoS, dis-association DoS, power save DoS, etc., exploit the un-authenticated nature of management and control frames. In this paper we focus on the power save denial of service (PS-DoS) attack.

The power save feature of 802.11 standard [4] allows the STAs to enter into sleep state to conserve battery life. If any data arrives for the STA while it is in sleep state, the AP buffers the data and delivers to the targeted STA when it wakes up. An STA informs the AP about its change in power save state from sleep to awake and vice-versa using a PS-Poll or null data frame[1]. Null data frame does not contain any data. Encryption schemes like WEP, WPA, WPA2, etc., only encrypt the data payload but leaving the MAC header in plain-text.

An attacker can craft spoofed null data frame in order to fetch buffered frame(s) targeted at other STAs, which results in frame losses for the original recipient(s). An attacker repeats this process for different STAs which leads to severe frame losses for the STAs and finally results in the PS-DoS attack. In this paper, we have considered the case that the attacker causes frame losses to multiple STAs. This helps the attacker increase the potency of the PS-DoS attack. If the attack is restricted to only one STA then the PS-DoS attack can be trivially detected using standard anomaly based techniques [5]. Also, the impact of the PS-DoS attack on a single STA is limited and does not lead to DoS on the network. In order to have an higher attack impact and to evade detection, the attacker targets multiple STAs in the network. All the STAs whose buffered frames are retrieved by the attacker are termed as victim STAs.

Current methods to prevent the PS-DoS attack include encryption, up-gradation to newer standards which necessitate firmware upgrades. Encryption requires systematic arrangement of key establishment, key renewal, key revocation leading to increased deployment and maintenance costs. Up-gradation to newer standards require firmware upgrades on the STA as well as server, which often incurs high cost. Moreover, presence of legacy network(s) makes this task difficult. Received

Manuscript received September 29, 2014; accepted October 5, 2015. This work was supported by TATA Consultancy Servies (TCS) Research Fellowship Program, India. Recommended by Associate Editor Yilin Mo. *(Corresponding author: S. Biswas.)*

Citation: M. Agarwal, S. Purwar, S. Biswas, S. Nandi, "Intrusion detection system for PS-Poll DoS attack in 802.11 networks using real time discrete event system," *IEEE/CAA J. of Autom. Sinica*, vol. 4, no. 4, pp. 792−808, Oct. 2017.

M. Agarwal, S. Biswas, and S. Nandi are with the Department of Computer Science and Engineering, IIT Guwahati, Assam, India (e-mail: mayank.agl@iitg.ernet.in; santoshbiswas402@yahoo.com; sukumar@iitg.ernet.in).

S. Purwar is with the HP PPS R&D Lab in Bangalore, India (e-mail: sankethpurwar@gmail.com).

Digital Object Identifier 10.1109/JAS.2016.7510178

[1]Since null data and PS-Poll frame have the same functionality, in this paper we explain the concepts using null data frame only. The concept and theories that are applicable to null data frame are also applicable to PS-Poll frame. In case of exceptions, a clear distinction is made.

signal strength indicator (RSSI) based approaches can help detect the PS-DoS attack. However, the use of RSSI based techniques requires additional specialized hardware equipment which increases costs. So we can see that existing methods escalates deployment and maintenance costs. Signature based IDS uses predefined signatures while anomaly based IDS make use of statistics to detect attacks [6]. PS-DoS attack does not differ in semantics or statistics under normal and attack circumstances. As a result, generation of signature or statistics for such attacks is difficult. Usage of such IDSs may lead to high false positive rate. In this work we propose a real time discrete event system (RTDES) based IDS that helps to overcome the drawbacks listed above and detect the PS-DoS attack with high accuracy and low false positive rate.

DES theory has been used for fault detection and diagnosis (FDD) in various systems like HVAC, chemical reaction chambers, nuclear reactors, VLSI circuits [7]−[13] etc. The core idea is to develop the normal and failure model corresponding to normal and failure scenarios. Subsequently a diagnoser which is a state estimator is built using the states traversed in the normal and fault models. The diagnoser determines whether the system is operating under normal, failure or uncertain conditions. An overview of FDD using DES is discussed in [14], [15]. DES based FDD have various frameworks depending on the system under consideration. For example, in distributed systems Petri Net based frameworks is preferred [16], [17] while for system involving stochastic process a stochastic DES based framework is recommended [18]. In cases where the underlying system involves incomplete or partial observation, frameworks based on partial observation and learning need to be used [19]. For faults related to timing of events, an RTDES [20] based framework proves to be beneficial.

The attacks in networks are equivalent to failures. Both the failure and attack denote a deviation from normal activity. The use of DES for detecting network attacks has been studied in [21]. However, the DES based IDS proposed in [21] is designed for a wired local area network (LAN) attack. Attacks occurring in Wireless LAN are more difficult to detect than wired counterparts due to mobility, obstructions caused by obstacles in path, noisy medium, limited coverage of the AP and limitations of the processing power of wireless nodes. The proposed RTDES framework requires certain extensions over classical theory [22] and techniques, which have been used in [21], [23], [24]. The modifications are made as follows:

1) The PS-DoS attack manifests itself in time. As a result, timing (delay-deadline) information needs to be incorporated in the model. In order to incorporate timing information we have used the RTDES framework [20] instead of un-timed one.

2) The sequence of power save frame exchange is the same under normal and attack conditions. This leads to creation of identical models under normal and attack scenarios thwarting diagnosability. To create differences between normal and attack scenarios, active probing is used. It may be noted that active probing maintains 802.11 protocol standard.

3) We incorporate model variables in the RTDES based IDS in order to overcome the state explosion problem.

The summary of our contributions are:

1) We propose an RTDES based IDS[2] that detects the PS-DoS attack in Wi-Fi networks. The developed IDS adheres to the 802.11 standard. No protocol modification is required. We exploit the fundamental characteristics and properties of the 802.11 protocol standard to detect the PS-DoS attack.

2) The communication pattern with the AP under normal and the PS-DoS attack conditions are identical. In order to create difference between them, IDS uses active probing technique.

3) The only hardware requirement is a sensor capable of sniffing the wireless data. So the cost of the proposed scheme is low and can be readily applied to legacy as well as existing networks.

4) As DES is a formal paradigm, the correctness of the proposed scheme is verified by considering all possible attack scenarios based on the arrival timing of the PS-Poll and null data frames.

5) We incorporate model variables in the proposed RTDES based IDS to detect the PS-DoS attack. Model variables help to overcome the state explosion problem.

The rest of the paper is organized as follows. Section II describes the power save feature of 802.11 networks. We discuss the security vulnerabilities associated with Wi-Fi networks and the ways in which they can be exploited by an attacker to launch the PS-DoS attack. We also look into the current approaches to tackle the PS-DoS attack. In Section III, we introduce the concept of failure detection and diagnosis (FDD) using real time discrete event system (RTDES). We describe the RTDES based IDS to detect the PS-DoS attack in the same section. The correctness of the RTDES based IDS along with its detection rate, accuracy and network load of the IDS are described in Section IV. Section V concludes the work.

II. Background and Motivation

In this section we look at the power save feature and the vulnerabilities associated with it. We look at how these vulnerabilities can be exploited to launch the PS-DoS attack. Subsequently we look at the existing solutions to detect the PS-DoS attack and the motivation behind the work.

A. Power Save Feature in 802.11

Hand-held devices usually have limited battery, so an effective power saving mechanism is must. The 802.11 standard incorporates a power saving feature that enables an STA to enter into sleep state, which consumes much less power. When an STA is in sleep state, it can neither transmit nor receive frame(s). An STA enters into sleep(awake) state by sending null data frame with power Mgmt bit set to 1 (0). If any data arrives at the STA while it is in sleep state, the AP buffers the data on behalf of the STA. The AP sets the STA's association IDentifier (AID) bit to 1 in the traffic indication map (TIM) message of the successive beacon frames until the STA retrieves all the buffered frames at the AP. Beacon frames are periodically sent by the AP to help the STAs

[2]Henceforth in this paper, IDS would mean RTDES based IDS.

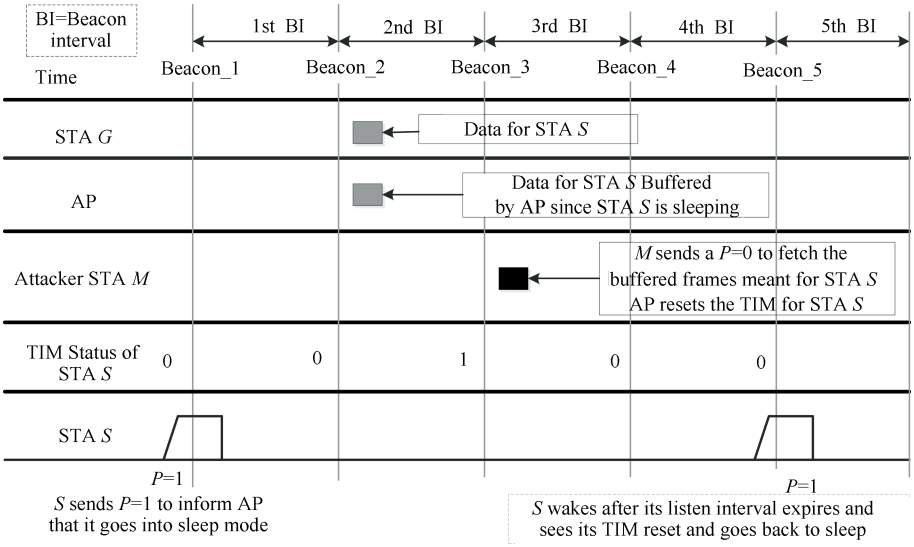


Fig. 1. Overview of the PS-DoS attack.

detect the presence of the AP. The beacon frame includes TIM information element which consists of a list of AID(s) of those STA(s) whose frame(s) is(are) buffered at the AP.

An STA in sleep state must periodically wake up and listen to the beacon frame. An STA uses the value of listen interval (LI) for waking up periodically. If an STA is in sleep state, it must wake up after LI number of beacons to check for the presence of buffered frames at the AP. For example, if the STA is in sleep state and its LI is 5, the STA needs to wake up every 5th beacon to check if any frame(s) is (are) buffered at the AP. The value of LI is decided during the association process. The LI is a 2 byte element in the association request frame sent by the STA to the AP. The AP needs to buffer the frame(s) for a sleeping STA at-least for the corresponding STA's LI number of beacon frames. A sleeping STA wakes up after the LI duration and reads the TIM element of the beacon frame. If the AID of the STA is set (reset) in TIM, it implies that data is buffered (not buffered) at the AP. The STA sends a null data frame with Power Mgmt bit set to 0 to retrieve the buffered frame(s) at the AP. If a STA continues to remain in sleep state even after its LI expires, the AP discards the buffered frame(s) for the corresponding STA.

B. Vulnerabilities in Power Save Feature of 802.11

Encryption schemes like WEP, WPA, WPA2 etc., encrypt only the data payload keeping other fields in clear-text. PS-Poll and null data frame does not contain any data, and is sent in clear-text. Being un-encrypted, spoofing these frames is trivial. The PS-DoS attack is explained by an example shown in Fig. 1. STAs M, S and G are associated with the same AP. M is the attacker while S and G are normal STAs. We assume beacon interval (BI) as 100 ms and S's LI as 5 for the purpose of explanation. The BI is available in all the beacon frames transmitted by the AP. M gets the LI and AID of the STA S by eavesdropping on the association process between the STA S and the AP.

The time-line shown in Fig. 1 is explained below:

1) [1st BI]: S sends null data frame with $PwrMgmt = 1$ to the AP to inform that it is entering sleep state. TIM for S is set to 0 as there are no outstanding frame(s) for S at the AP.

2) [2nd BI]: G sends one data frame to S while S is in sleep state. This frame is buffered by the AP as S is in sleep state. AID in the TIM for S is set to 1 in all successive beacon frames till all the buffered frame(s) is (are) retrieved by S.

3) [3rd BI]: M captures a beacon and finds the AID of S is set in TIM. The attacker sends a spoofed null data frame with source (SRC) MAC address set to MAC address of S and PwrMgmt bit set to 0. On receiving the null data frame the AP sends all the buffered frames to M. After all buffered frames targeted at S are delivered, the AP resets the AID of S in TIM of the successive beacons.

4) [4th BI]: No frame exchange for S. TIM = 0 for S.

5) [5th BI]: S's LI expires. It wakes up to read the TIM bit of the beacon frame. S finds that its AID in TIM is reset indicating absence of any buffered data at the AP. S goes back to sleep state by sending null data frame with $PwrMgmt = 1$ to the AP.

As seen above, the attacker M silently steals the buffered frame(s) targeted at S. S as a result becomes the victim STA as it does not receive the intended frame(s). G also indirectly becomes a victim as it assumes that frame(s) is (are) successfully delivered to S, but in fact they are delivered to M. So in the PS-DoS attack both the sender and the receiver are victimized. M should repeat this process for different STAs at arbitrary intervals in the network to cause frame losses to STAs resulting in the PS-DoS attack. The attacker has to choose arbitrary intervals to launch PS-DoS attack since performing the PS-DoS attack at regular intervals could lead to detection by an anomaly based IDS caused by regularity in attack intervals. As can be observed the PS-DoS attack can be easily launched using minimal resources and possesses the potential to cause high frame losses. In the following sub-section, we look into various approaches to detect and prevent the PS-DoS and describe the motivation behind our work.

C. Existing Approaches to Detect or Prevent the PS-DoS attack

1) Encryption based methods: An intuitive solution is to encrypt all the management and control frames along with the data frames as suggested by Bellardo *et al.* in [3]. Encryption requires an establishment of systematic secure key distribution, management, renewal and revocation system, which involves a huge overhead. As the PS-DoS attack involves spoofing of null data frame, encrypting all frames does prevent the PS-DoS attack. Management and control frames are used frequently for communication purposes. Encryption and decryption of all management and control frames would involve additional processing for both the AP and the STAs. This in turn would lead to faster draining off the batteries of the STAs. It also necessitates up-gradation of the STA and the AP firmware which is costly. Qureshi *et al.* [25] propose an encryption based scheme to prevent the PS-DoS attack by encrypting the AID portion of the PS-Poll frame using pre-established PTK. As PTK is used, this method is not applicable to open and WEP based networks. Null data frame does not contain the AID field, which limits this method to those STAs using PS-Poll frames for power save operations.

Meiners [26] proposes that the STAs must use non-empty data frame like ICMP/ARP for reporting the change in power save mode. The motive behind this solution is to eliminate all possible frame candidates that are vulnerable to the PS-DoS attack. As ICMP/ARP frame contain data, the data portion is encrypted, which prevents possible spoofing by an adversary. However, this solution has fundamental drawbacks. First, it breaks the AP compatibility with those STAs that use traditional frames like null data frames to inform about the change in power save mode. The null data frames are used by NICs for managing power, scanning channel and keeping the association awake [27]. Eliminating the processing of such frames would prevent the STA from performing these functions.

2) Received signal strength indicator (RSSI) based methods: Faria *et al.* [28] use the AP(s) as sensors and obtain the RSSI values for each STA within its range. They have shown that the signal prints of a STA and its physical location are closely correlated. Azimi *et al.* [29] and Chen *et al.* [30] also make use of physical layer countermeasures like RSSI values and RF signal print to detect and localize the attacker. In RSSI based techniques the approximate location of the sender is determined. Then the location of the source MAC address in the received frame is determined. If the difference in the location exceeds a certain threshold, the frame is not processed and tagged as spoofed. Though physical layer countermeasures are effective, they require specialized hardware and firmware changes in 802.11. If the STAs are located close to each other, their physical characteristics are similar, which may result in false positives, since a genuine STA might be classified as an attacker.

3) Up-gradation to newer IEEE standards: IEEE 802.11w standard provides protection for the management frames. However, the proposed standard does not include protection for control frames [31]. PS-Poll frame being a control frame, 802.11w leaves the PS-Poll frame unprotected. 802.11w standard enhances the security of Wi-Fi networks by mitigating dis-association and de-authentication attacks. However, it also involves additional deployment costs, firmware upgrades for both the STA and the AP. A large number of legacy devices exist today, which do not support 802.11w standard [32].

In brief, the drawbacks of the existing schemes to detect or prevent the PS-DoS attack are:

1) Requirement of Encryption;
2) Firmware up-gradation;
3) Up-gradation to newer standards;
4) Installation of specialized hardware.

From the above summary it is clear that a strategy to detect the PS-DoS attack is required having the following features.

1) No alteration in 802.11 protocol is required;
2) Hardware costs should not be exorbitant;
3) It must be easily deployable to the existing as well as new networks;
4) It must not require patching of underlying operating system or installation of new software;
5) If the IDS generates extra traffic, it should be as low as possible.

Signature and anomaly based IDS usually generate a lot of false positives when used for detecting the PS-DoS attack and related attacks, which do not alter frame semantics under normal and attack conditions. In such cases, DES based IDS have proven to be an effective mechanism for detecting network attacks without any need for protocol modification, encryption or installation of proprietary hardware. A DES based IDS can be formally proved to be correct. In this paper we propose RTDES based IDS for the PS-DoS attack that incorporates the features listed above and overcomes the drawbacks of the existing approaches.

III. Proposed Scheme: Application of RTDES for Detecting PS-DoS Attack

In this section we first present the principle to detect the PS-DoS attack. Then, the architecture of the proposed RTDES based IDS is explained followed by listing the assumptions of the attacker and the IDS along with their motivations. Subsequently, basic concepts related to fault detection and diagnosis using RTDES are described. The construction of normal and fault (attack) model, RTDES diagnoser which is instrumental for detecting the PS-DoS attack are described next.

A. Working Principle of the IDS for Detecting PS-DoS Attack

Principle: If the buffered frames for a STA are fetched before the expiry of its LI, IDS marks this activity as suspicious. Such activity cannot be directly marked as attack since the 802.11 standard does not prohibit or restrict the Wi-Fi STAs to wake up earlier than their scheduled waking up. This is plausible as it may happen that the STA may have to transmit some critical data for which it may have to break its sleep

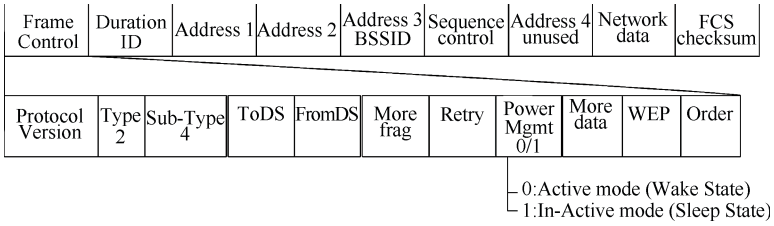


Fig. 2. Frame format of null data frame.

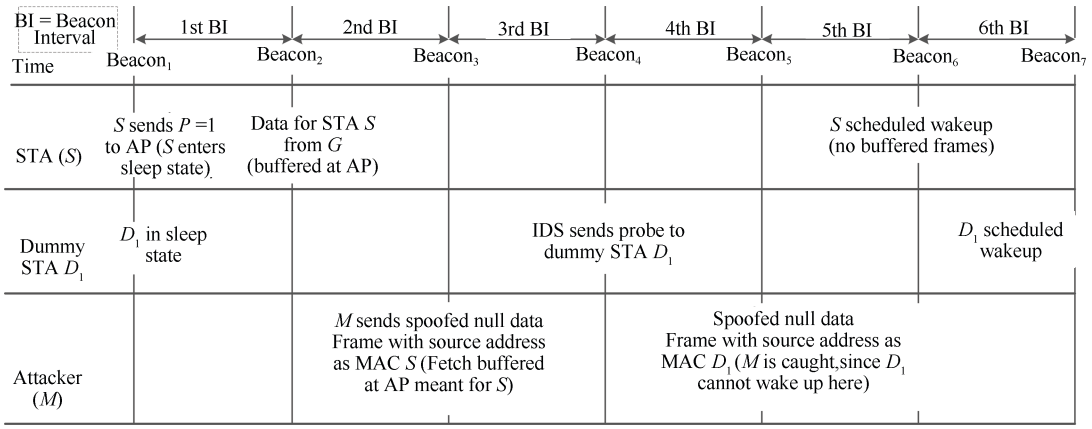


Fig. 3. Time-line of detecting the PS-DoS attack.

cycle. So, every STA that wakes up earlier than scheduled cannot be assumed to be an occurrence of the PS-DoS attack. However, in the proposed scheme we have the dummy STAs as part of detection methodology. The dummy STAs are software controlled and their communication is handled by the IDS. The IDS ensures that the dummy STAs never wake up before the expiry of their LI. Only the IDS possesses the dummy STA's MAC address and is regularly updated by the IDS. The need for updating the dummy MAC address and other characteristics of the dummy STAs are explained later. Upon observing an early wake up frame[3] for any STA(s) associated with the monitored AP, the IDS sends a power save probe to the dummy STA. As the dummy STA is in sleep state at the time IDS sends a power save probe, the frame is buffered at the AP. The power save probe is a simple 802.11 data frame sent from IDS destined to the dummy STA. If the buffered frame(s) meant for the dummy STA are fetched before the expiry of the LI of the dummy STA under question, the presence of the PS-DoS attack is confirmed.

Reason: The dummy STAs never fetch frame(s) before expiry of LI as their communication is handled by the IDS. So, it is not possible that the frame(s) for the dummy STA are retrieved before the expiry of the LI of the dummy STA. Consequently, fetching of buffered frame(s) meant for the dummy STA must have been by the attacker on behalf of the dummy STA in its quest to fetch the buffered frame(s) of the sleeping STAs.

Using the above principle we show an example of the PS-DoS attack detection technique with the help of the time-line shown in Fig. 3. M, S and G are associated with the same AP. M is the attacker, S and G are the normal STAs while D_1 is the dummy STA. The time-line has 7 distinct time slots. Each time slot corresponds to a beacon interval (BI).

1) [1st BI]: S goes to sleep. S's scheduled waking up is at BI_6. The dummy STA D_1 is in sleep state. D_1's scheduled waking up is at BI_7.

2) [2nd BI]: G sends data to S. As S is in sleep state, the data is buffered by the AP. The AP sets the AID for S in TIM bit for successive beacons till all buffered frame(s) is (are) retrieved by S.

3) [3rd BI]: M sends a waking frame spoofing as S to the AP. All buffered frame(s) targeted at S are retrieved by M. M acknowledges the AP on behalf of S.

4) [4th BI]: IDS observes early wake from the STA S. IDS sends a power save probe to the dummy STA D_1 to verify the early wake of S. The dummy STA's scheduled waking up is at BI_7. The AP resets the AID for S in the TIM bit for successive beacons as no buffered data is left at the AP.

5) [5th BI]: IDS sniffs a waking frame to retrieve buffered frame targeted at D_1. As D_1 never fetches frame before scheduled waking up, the request for fetching frame(s) must have been made by the attacker on behalf of the dummy STA. Thus, presence of the PS-DoS attack is detected.

6) [6th BI]: S's scheduled waking up time. S finds its TIM is reset and goes back to sleep.

7) [7th BI]: D_1's scheduled waking up time. D_1 goes back to sleep as no buffered frame(s) is (are) present for it.

As seen, S is the victim STA that lost it buffered frame(s) due to the PS-DoS by M. The sending of power save probes to D_1 helps to detect the presence of M in the network. The timing of the response of the power save probes is vital in ascertaining the presence of attacker. The block

[3]For simplicity of explanation of the formal (RTDES) modeling we refer null data frame with $PwrMgmt$ = 1 as **sleep** frame and null data frame with $PwrMgmt$ = 0 as **waking** frame.

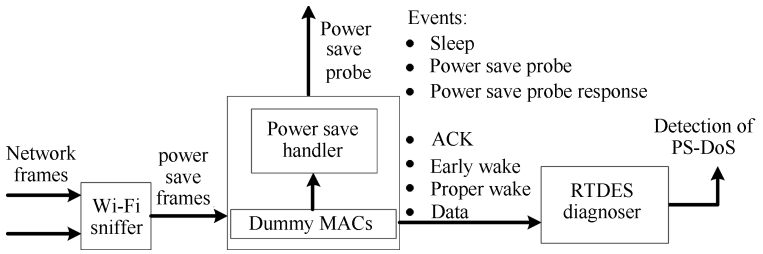


Fig. 4. Block diagram of proposed IDS.

diagram for detecting the PS-DoS attack is shown in Fig. 4. The components are described as follows:

1) Wi-Fi sniffer: The Wi-Fi sniffer works in promiscuous mode and captures all Wi-Fi frame(s) traveling in the network. Only those frame(s) destined to and from the monitored AP are sniffed. frame(s) to other APs are dropped. The Wi-Fi sniffer forwards the captured frame(s) to "power save handler".

2) Power save handler: The Power Save Handler is responsible for extracting vital information like LI, BI, AID, MAC address of the source (SRC) STA from frame headers. The Power Save Handler component is also responsible for the generation of events like *Sleep, ACK, Early Wake, Proper Wake, Power Save Probe, Power Save Probe Response, Data.* The Power Save Handler forwards the generated event (s) to RTDES Diagnoser. The RTDES diagnoser determines if the PS-DoS attack has occurred or not based on the events captured.

3) Dummy STAs: IDS maintains a list of the dummy STAs with it. The dummy STAs are software controlled and their communication is handled by the IDS. As already explained in the principle of attack detection, the 802.11 standard allows a STA to wake up before the expiry of its LI. In order to check the genuineness of the early waking frame sent, the IDS sends a power save probe destined to a dummy STA while it is in sleep state. As the communication of the dummy STAs is handled by the IDS, the dummy STAs never wake up before the expiry of their LI. So, in-case a waking frame is seen to fetch the frame(s) targeted at a dummy STA in the network before the expiry of the LI, the IDS raises an alarm indicating the occurrence of the PS-DoS attack. In case the power save probes are fetched after the expiry of the LI of a dummy STA the network is assumed to be operating under normal conditions. The traffic pattern of the dummy STAs and the associated network overhead are discussed in later sections.

4) RTDES Diagnoser: The RTDES diagnoser is actually implemented as a software module and used as attack detector. The RTDES diagnoser forms the crux of the detection methodology and is constructed from the DES model under normal and attack conditions. The construction and uses of the diagnoser are described in Section III-E.

B. Attacker and IDS Assumptions

Attacker Assumptions.

1) Attacker fetches buffered frame(s) at the AP immediately for the STAs that are in sleep mode.

An attacker fetches frame(s) buffered at the AP for the targeted sleeping STAs immediately.

Motivation: If the attacker delays in fetching the frame(s) buffered at the AP it is possible that the real STA may wake up and fetch the frame(s). This may reduce the impact of the PS-DoS attack. Also, those STAs staying awake persistently never have buffered frame(s) at the AP. So the PS-DoS attack is not possible on such STAs.

2) Attacker launches the PS-DoS attack on multiple STAs simultaneously at arbitrary intervals.

The attacker targets multiple STAs simultaneously. The attacker first does a passive scanning of the network to determine the list of associated STAs with the target AP. The attacker eavesdrops on the communication between the STAs and the AP to obtain useful parameters like the network's BI, STA's LI and AID. Using these parameters the attacker crafts spoofed null data frame in order to fetch the buffered frame(s) at the AP for the targeted sleeping STAs.

Motivation: For the success of the PS-DoS attack on a STA, two conditions must be satisfied: i) the STA is in sleep mode and ii) the AP has buffered frame(s) for the STA. Experimentally it has been observed that occurrence of these two scenarios is not very frequent. So, in order to increase the impact of the PS-DoS attack to a reasonable level, multiple STAs are targeted by the attacker. Also, if a single STA is attacked then the attacker needs to fetch the buffered frame(s) for that STA each time there is a buffered frame for it at the AP and the selected STA is in sleep mode. Such frequent early fetches by the attacker renders attack detection trivial by any standard anomaly IDS [5], where statistical deviation of network behavior when compared to normal profile (with regards to sleep and waking up schedule) is declared as attack. On the other hand, if multiple STAs are targeted then the attacker makes early fetches on each individual STA at arbitrary intervals so that sleep and wake-up profile is not significantly different from the normal scenario.

IDS Assumptions.

1) Monitoring of the authorized AP(s) only.

The IDS maintains a white-list of the MAC address of the AP that need to be protected. Only the frame(s) destined to and from these AP(s) are monitored.

Motivation: Frame(s) destined to other AP(s) are discarded as the primary goal of installing the IDS is to detect the attacks on the APs deployed by the administrator and not other external AP(s).

2) IDS keeps track of sleep and waking up schedules of the associated STAs.

Motivation: Wi-Fi STAs follow their sleep and wake schedule (s) so that they can conserve power by being in

sleep state when they have no data to transmit. In case of the PS-DoS attack the frequency of early wake by the STAs get increased as the attacker fetches the buffered frame(s) available at the AP on the behalf of the STAs while they are in sleep state. By keeping a track of the sleep and wake schedules, the IDS can identify those STAs that wake up earlier than their scheduled wakeup timing. STAs which are frequently seen to waking up before completion of their sleep schedule(s) may be the potential victims of the PS-DoS attack. This also helps the IDS to determine the number of dummy STAs to be kept in the network by following the philosophy — more potential victims there are, more dummy STAs are needed.

3) Proposed IDS has sniffing and injection capabilities.

The IDS sniffs the wireless frames traveling in the air. If the IDS observes an early waking frame it sends a power save probes destined to the dummy STA. The IDS maintains a list of AID, LI, sleep-wake timing of all the associated STAs for the AP under question.

Motivation: If the IDS does not have sniffing capabilities then early wake by the STAs could not be detected. If the IDS does not posses the frame injection capability, no probes would be sent for early wake by the STAs. So, the attacker would escape detection. Hence, the proposed IDS has both sniffing as well as probe injection ability in order to detect the presence of the attacker.

4) IDS regularly updates the dummy STA's MAC address list.

Motivation: IDS regularly updates the dummy STA's MAC address list to prevent possible learning of the dummy STA's MAC address by the attacker. To elaborate, most of the Wi-Fi STAs are mobile in nature. As a result, large number of STAs join and leave the network over a period of time. Even though the normal STAs have fixed MAC addresses, their joining and leaving pattern with respect to the network is highly dynamic. A normal STA may join the network and then leave the network in a few minutes or hours and may never return to the same network. Few of the STAs may join the network and leave after some time and may associate to some other AP in the same network. However, some of the STAs (like employees working for a firm) may show a static behavior in terms of joining and leaving, but they are very few. To mimic a similar dynamic trend with that of the normal STAs, the IDS regularly updates the dummy STA's MAC address list. So, even if the normal STAs have fixed MAC addresses the attacker cannot learn whether the STAs are normal or dummy by looking at whether the MAC address appeared in the historical data. In other words, if the list of dummy STA's MAC address is not dynamic in nature, an attacker could possibly find the list of static MAC address in the network by observing the historical data of the MAC addresses seen in the network. The attacker would not respond to any of the power save probes having MAC addresses belonging to these static STAs assuming them to be dummy MAC address thus evading attack detection. So, the list of dummy STA's MAC address needs to be updated regularly to prevent such possible learning by the attacker.

5) Dummy STAs never wake up before expiration of their LI.

Motivation: The normal STAs follow their sleep cycles according to their LI. However, the 802.11 standard does not prohibit or restrict the Wi-Fi STAs to wake up earlier than their scheduled waking up. This is also plausible as it may happen that the STA may have to transmit some critical data for which it may have to break its sleep cycle. So, every STA that wakes up earlier than its scheduled waking up cannot be marked directly as the occurrence of the PS-DoS attack.

However, in the proposed IDS the behavior of the dummy STAs is handled by the IDS. So, these dummy STAs never wake up before the expiry of their LI. So, if a STA is observed to wake up early and fetch buffered frame(s) targeted at the dummy STA from the AP, it is definitely an instance of PS-DoS attack. To elaborate, in this case the attacker is spoofing the dummy STA′s MAC address and sniffing the frame(s) originally buffered for the dummy STA in the AP. This phenomenon of the fetching of the frame(s) targeted at the dummy STAs forms the basis of the proposed IDS.

6) Dummy STAs mimic the behavior of the real nodes in terms of the traffic generated.

The traffic pattern generated by the dummy STAs is similar in nature to that generated by the normal STAs in the network.

Motivation: In our detection philosophy, the IDS sends a power save probe to a dummy STA. As the attacker is unaware of the presence of the dummy STAs it responds to the probe and eventually gets caught. If the traffic pattern of the dummy STAs significantly varies with the normal traffic pattern, an attacker can identify the presence of the dummy STAs using traffic flow(s). Once the dummy STAs are identified, the attacker would not fetch the frame(s) targeted at the dummy STAs at the AP. As a result, the attacker could evade detection. So, in order to make the dummy STAs behavior indistinguishable from the real STAs the traffic pattern generated by the IDS for the dummy STAs is similar to that of normal traffic pattern. Further, the number of dummy STAs in the network is kept to minimum and their MAC addresses are changed dynamically. So, not only their numbers is meager and dynamic but also their traffic pattern is similar to normal network traffic. These two characteristics of the dummy STAs makes them stealthy in nature and identifying them based on traffic pattern becomes non-trivial for an attacker. As a result, when the IDS sends a probe to a dummy STA, there are high chances that the attacker responds to the probe frame assuming it to be from a genuine STA, which leads to attack detection.

So, from the above discussion it is clear that timing of the various network activities(sending probe, fetching probe frames etc.) is important in order to detect the PS-DoS attack. As RTDES modeling is capable of capturing timing information into the model, we have chosen the RTDES framework for modeling the PS-DoS attack. We now look how RTDES framework that is capable of capturing the timing information and help in detecting the PS-DoS attack.

C. Fault Detection and Diagnosis Using Real Time Discrete Event System (RTDES)

The RTDES model G is defined as $G = \langle V, X, t, \Im, \theta \rangle$, where $V = \{v_1, v_2, \ldots, v_n\}$ is a finite set of discrete variables, X is a finite set of states, t is a clock variable, $\Im$ is a finite set of transitions and θ is the initial condition. Each variable

$v \in V$ ranges over binary values as its domain elements. Each state $x \in X$, is a mapping of each of the variables to one of its domain elements. The model has a clock variable t with $type(t) = \mathbf{N}$, the set of all natural numbers. The clock variable represents time on a global clock. A transition $\tau \in \Im$ from a state x to another state x^+ is an ordered seven-tuple $\tau = \langle x, x^+, \sigma, l_\tau, u_\tau, check(V), assign(V)\rangle$, where

1) x is the initial state of the transition, denoted as $initial(\tau)$.

2) x^+ is the final state of the transition, denoted as $final(\tau)$.

3) σ is the event on which the transition is fired.

4) l_τ, u_τ are the delay and deadline time bounds, denoted as $delay(\tau)$ and $deadline(\tau)$, respectively. Let $t_{c\tau}$ be the time instant when τ is enabled. A transition can take place at any time instant t when $t_{c\tau} + l_\tau \leq t \leq t_{c\tau} + u_\tau$, provided that the transition remains enabled throughout the interval $[t_{c\tau}, t]$. Therefore, a transition does not take place before the delay and must take place before the deadline.

5) $check(V)$ represents conditions on a subset of model variables. For firing τ, along with the enabling event σ, $check(V)$ should hold true.

6) $assign(V)$ represents a subset of model variables and assignment of values from their corresponding domain, when τ fires.

A transition τ from x to x^+ is denoted as $\tau : \langle x, x^+\rangle$ for brevity when its other components are clear from the context. The tick transition, or simply tick, denoted as η, is defined as $\eta = \langle x, x, true, -, -, -, -\rangle$. Each occurrence of tick results in an increment of the clock t by 1 and leaves the other variables unchanged. In fact, tick is the only transition that changes the value of t. *Tick occurs infinitely often and is not explicitly included in* $\Im$. No model variables are defined for the *tick transition.*

The initial condition θ specifies the set X_0 of initial states at $t = 0$. A trace of model G is a sequence of transitions generated by G denoted as $s = \langle \tau_1, \tau_2, \ldots\rangle$, where $initial(\tau_1)$ is an initial state, and the juxtaposition property holds, that is, $initial(\tau_{i+1}) = final(\tau_i)$, for $i \geq 1$. Henceforth, we assume the juxtaposition property for "sequence of transitions". A state x is said to be in a trace s, if $x = initial(\tau_i)$, for some $i \geq 1$. The set of all traces generated by G along with all their finite prefixes is the language of G, denoted as $L(G)$. $L_f(G)$ denotes the subset of $L(G)$ comprising the finite prefixes. For any trace $s = \langle \tau_1, \tau_2, \ldots\rangle$, $initial(s) = initial(\tau_1)$ and for a finite trace $s = \langle \tau_1, \tau_2, \ldots \tau_f\rangle$, $final(\tau_f) = final(s)$. Naturally, $L(G)$ is a subset of $\Im^w$, where $\Im^w$ is the set of all infinite sequences of $\Im$; $L_f(G)$ is a subset of $\Im^*$, the Kleene closure of $\Im$. The post language of G after a trace s, denoted as $L(G)/s$, is defined as

$$L(G)/s = \{t \in \Im^w \mid st \in L(G)\} \tag{1}$$

$L_f(G)/s \subset L(G)/s$ comprises finite prefixes of the infinite traces of $L(G)/s$.

D. RTDES Model: Measurement Limitations & Failure Diagnosis

Limitations of measurement give rise to uncertainty in transitions in the observed dynamics of the model. In this section, the notion of measurement limitation in the RTDES framework is formally introduced and the consequent uncertainty in transitions in G is characterized. For explanation of the DES terminologies using practical systems, the readers are directed to [33]. States and transitions belonging to normal (failure) model are denoted by the non-primed (primed) notations.

Definition 1 (Measurable and unmeasurable events/transitions): Any **event** that can be measured in the system using sensor(s) is a measurable event. Events that cannot be measured using sensors are unmeasurable events. A **transition** $\tau_i = \langle x, x^+, \sigma, l_\tau, u_\tau, check(V), assign(V)\rangle$ is said to be a measurable (unmeasurable) transition if σ is a measurable (unmeasurable) event. $\Im_m$ and $\Im_u$ denote the set of measurable and unmeasurable transitions.

Definition 2 (Measurement equivalent transitions and states): Two transitions $\langle x_1, x_1^+, \sigma_1, l_{\tau 1}, u_{\tau 1}, check_1(V), assign_1(V)\rangle$ and $\langle x_2, x_2^+, \sigma_2, l_{\tau 2}, u_{\tau 2}, check_2(V), assign_2(V)\rangle$ are equivalent if $\sigma_1 = \sigma_2$ (the same event), $l_{\tau 1} = l_{\tau 2}$ & $u_{\tau 1} = u_{\tau 2}$ (the same delay and deadline), $check_1(V) \equiv check_2(V)$ (the same equalities over the same subset of variables in V), and $assign_1(V) \equiv assign_2(V)$ (the same subset of model variables with the same assignment). If $\tau_1 \equiv \tau_2$ then the source states of the transitions are equivalent and so are the destination states, i.e., $x_1 \equiv x_2$ and $x_1^+ \equiv x_2^+$.

Definition 3 (Projection and inverse projection operator): A projection operator $P : \Im^* \rightarrow \Im_m^*$ is defined as: $P(\epsilon) = \epsilon(null\ string)$; $P(\tau) = \tau\ if\ \tau \in \Im_m$; $P(\tau) = \epsilon\ if\ \tau \in \Im_u$; $P(s\tau) = P(s)P(\tau),\ where\ s \in L_f(G), \tau \in \Im$. The function P erases the unmeasurable transitions from the argument finite trace. $P(s)$ is termed as the *measurable finite trace* corresponding to the finite trace s. An inverse projection operator $P^{-1} : \Im_m^* \rightarrow 2^{\Im^*}$ is defined as: $P^{-1}(s) = \{s' \in L_f(G) \mid sEs'\}$ Thus, $P^{-1}(s)$ encompasses all possible sequences of transitions that are equivalent to the finite trace s. The projection function P, the inverse function P^{-1} and the measurement equivalence E of finite traces can be extended to traces $\in \Im^w$, in a natural way.

Definition 4 (Measurable equivalent Traces): Two finite traces s and s' are said to be measurement equivalent if the following relation holds:

$$P(s) = \langle \tau_1, \tau_2, \ldots, \tau_n\rangle,\ P(s') = \langle \tau_1', \tau_2', \ldots, \tau_n'\rangle \text{ and } \tau_i E \tau_i', 1 \leq i \leq n.$$

We use the symbol E to denote measurement equivalence of finite traces as well as that of transitions, with slight abuse of notation. The equivalence of finite traces s and s' implies that if measurable transitions are extracted from s and s' by using operator P, then all the transitions are measurement equivalent.

Definition 5 (Normal G-state/G-transition and failure G-state/G-transition): States that are traversed by the system when operating without any faults are known as Normal G-state. X_N denotes the set of all normal states. A G-transition $\langle x, x^+\rangle$ is called a normal G-transition if $x, x^+ \in X_N$. States that are traversed by the system when operating under failure circumstances are known as Failure G-state. X_{F_i} denotes the set of all failure states. A G-transition $\langle x, x^+\rangle$ is called a failure G-transition if $x, x^+ \in X_{F_i}$.

Definitions 6 (Failure causing G-transition): A transition $\langle x, x^+\rangle$, where $x \in X_N$ and $x^+ \in X_{F_i}$, is called a $failure$ causing transition indicating the occurrence of some failure. Since failures are assumed to be $permanent$, there is no transition from any $x \in X_{F_i}$ to $x^+ \in X_N$.

E. RTDES Diagnoser

The diagnoser is a directed graph represented by $O = \langle Z, A\rangle$; where Z is the set of diagnoser nodes, called O-$nodes$, and $A \subseteq Z \times Z$ is the set of diagnoser transitions, called O-$transitions$. Each O-node $z \in Z$ corresponds to a set of G-states representing the uncertainty about the actual state. Similarly, each O-transition $a \in A$ of the form $\langle z_i, z_f\rangle$ is a set of measurement equivalent transitions representing the uncertainty about the actual measurable transition that occurs. The *unmeasurable successor* (set) of a set Y of states is defined as $\mathcal{U}(Y) = \bigcup_{x\in Y}\{x^+|\tau = \langle x, x^+\rangle \in \Im_u\}$. The *unmeasurable reach* of a set Y of states, denoted as $\mathcal{U}^*(Y)$, is the reflexive-transitive closure of unmeasurable successors of Y.

*D*iagnoser Construction: The states in X_0 are partitioned into equivalent subsets denoted as $X_{01}, X_{02}, \ldots X_{0m}$. For all i, $1 \leq i \leq m$, an initial O-node z_{0i} is obtained as the unmeasurable reach of X_{0i}, i.e., $z_{0i} = \mathcal{U}^*(X_{0i})$. The set of all initial O-nodes is denoted as $Z_0 = z_{01} \cup \ldots \cup z_{0m}$. The initial O-nodes capture the fact that the diagnoser can infer a set z_{0i} of possible initial system states (or their unmeasurable reach) by measuring the variables without waiting for the first measurable transition. Given any O-node z, the O-transitions emanating from z are obtained as follows. Let $\Im_{mz}$ denote the set of measurable G-transitions from the states $x \in z$. Let A_z be the set of all equivalence classes of $\Im_{mz}$ under E. For each $a \in A_z$, a successor O-node z^+ of z such that $z^+ = final(a)$ can be created as follows. Let $z_a^+ = \{final(\tau)|\tau \in a\}$; then $z^+ = \mathcal{U}^*(z_a^+)$ and a is designated as: $\langle z, z^+\rangle$. The set of the diagnoser transitions is augmented as: $A \leftarrow A \cup \{a\}$, and the set of O-nodes is augmented as: $Z \leftarrow Z \cup \{z^+\}$. Each $a \in A$ is an ordered pair $\langle z, z^+\rangle$, where $z = initial(a)$ and $z^+ = final(a)$. Thus, each O-node contains equivalent states. The detailed algorithm for diagnoser construction is shown in Algorithm 1.

Some definitions related to the diagnoser are discussed below:

Definition 7 (F_i-O-node and F_i-certain O-node): An O-node, which contains an F_i-G state, is called an F_i-O-node, denoted as z_{F_i}; the set of all F_i-O-nodes is denoted as Z_{F_i}. An F_i-O-node z is called an F_i-certain O-node if $z \subseteq X_{F_i}$. An F_i-O-node which is not F_i-certain is called F_i-$uncertain$.

Definition 8 (F_i-O-path (γ_{F_i})): A path of the diagnoser O is a sequence of O-transitions $\gamma = \langle a_1, a_2, ...\rangle$, with the consecution property $initial(a_{i+1}) = final(a_i), i \geq 1$. An F_i-O-path γ is an O-path in which every O-node is an F_i-O-node.

Definition 9 (F_i-uncertain cycle): An F_i-uncertain cycle is an F_i-O-cycle in which there is no F_i-certain O-node.

Algorithm 1: Algorithm for construction of diagnoser O for an RTDES model G

Input: RTDES model G
Output: RTDES Diagnoser

```
1  Partition X_0 into equivalent subsets X_01, X_02, ..., X_0m
2  for all i, 1 ≤ i ≤ m do
3  |  z_0i = U*(X_0i)
4  end
5  Z_0 ← z_01 ∪ ... ∪ z_0m
6  Z ← Z_0
7  A ← φ
8  for all z ∈ Z do
   |  /* Find the set of measurable G-transitions (ℑ_mz) outgoing from z */
9  |  ℑ_mz ← {τ|τ ∈ ℑ_m ∧ initial(τ) ∈ z}
   |  /* Find the set of all measurable equivalent classes A_z, of ℑ_mz */
10 |  for all a ∈ A_z do
11 |  |  z_a^+ = {final(τ)|τ ∈ a}
12 |  |  z^+ = U*(z_a^+)
13 |  |  Z = Z ∪ {z^+}
14 |  |  A = A ∪ {a}
15 |  end
16 end
```

Definition 10 (F_i-indeterminate cycle): An F_i-uncertain cycle γ in which the F_i-states contained in the O-nodes of γ form a cycle in G comprising transitions from γ, is called an F_i-*indeterminate cycle*. The equivalence between F_i-diagnosability and the absence of F_i-indeterminate cycles has been formally established for DES models [34]. Consider a sequence $\langle x_1, x_2, x_3, \ldots, x_1\rangle$ that corresponds to a normal cycle and measurement equivalent failure cycle $\langle x_1{}', x_2{}', x_3{}', \ldots, x_1{}'\rangle$. If the system is under normal conditions, the diagnoser moves into normal cycle and once a failure occurs it moves in the failure cycle. As both the normal and attack cycles are measurement equivalent, the normal and failure cycles are indistinguishable from one another. As a result of the presence of the indeterminate cycle it is not possible to predict whether the diagnoser is moving under normal or failure cycle. The presence of an F_i-indeterminate cycle leads to non-diagnosability.

The RTDES model G used to represent PS-DoS attack under normal and attack scenarios is shown in Fig. 5. For readability purposes Fig. 5 is annotated with transition number τ_i, delay-deadline of the transition $[delay, deadline]$ and the event due to which the transition τ_i is fired. The transitions of the RTDES model G shown in Fig. 5 are explained in Table I. States and transitions belonging to normal (attack) model are denoted by the non-primed (primed) notations. The various components of G are as follows:

$X = \{s_0, s_1, s_2, s_3, s_4, s_5, s_6, s_7, s_0{}', s_1{}', s_2{}', s_3{}', s_4{}', s_5{}'\}$

$\Sigma = \{$Sleep, ACK, Early Wake, Proper Wake, Data, Power Save Probe, Power Save Probe Response$\}$

$V = \{mac_{src}, mac_{dst}\}$ is the set of model variables. The domain of both mac_{src} and mac_{dst} is
$\{xx : xx : xx : xx : xx : xx|x \in [0 - F]\}$. mac_{src} and mac_{dst} holds source and destination MAC address contained in the frame respectively.

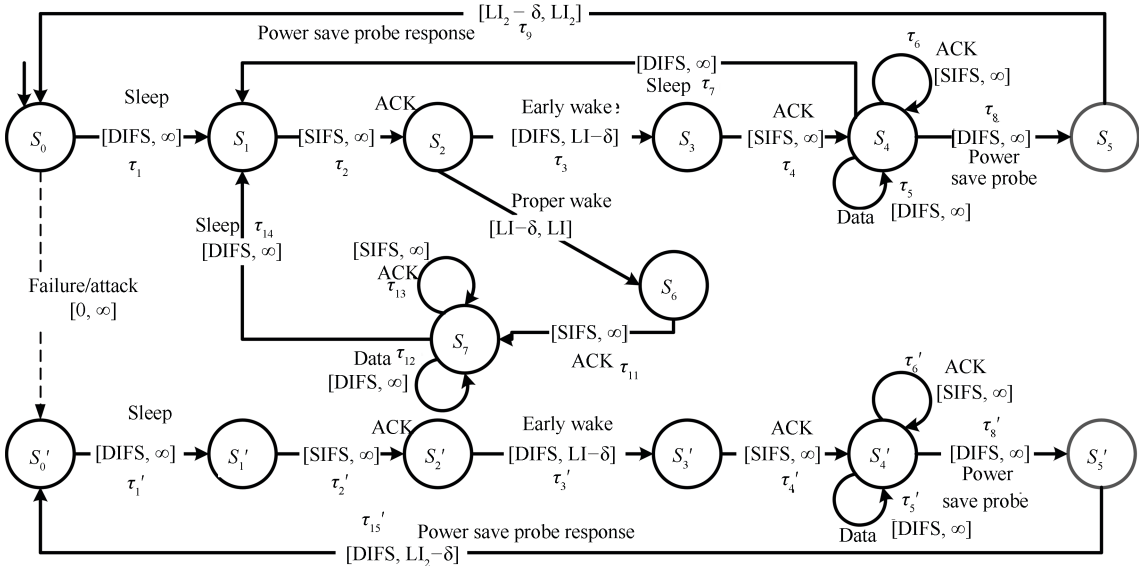


Fig. 5. Normal and attack model in PS-DoS attack.

TABLE I
TRANSITION TABLE FOR FIG. 5

Transition (τ)	Initial State (x)	Final State (x^+)	Event (σ)	Delay (l_τ)	Deadline (u_τ)	$check(V)$	$assign(V)$
τ_1	s_0	s_1	Sleep	DIFS	∞	–	$mac_{SRC} \leftarrow mac_{STA}$ $mac_{DST} \leftarrow mac_{AP}$
τ_2	s_1	s_2	ACK	SIFS	∞	$mac_{SRC} \equiv mac_{AP}$ $mac_{DST} \equiv mac_{STA}$	–
τ_3	s_2	s_3	Early Wake	DIFS	$LI - \delta$	$mac_{SRC} \equiv mac_{STA}$ $mac_{DST} \equiv mac_{AP}$	–
τ_4	s_3	s_4	ACK	SIFS	∞	$mac_{SRC} \equiv mac_{AP}$ $mac_{DST} \equiv mac_{STA}$	–
τ_5	s_4	s_4	Data	DIFS	∞	$mac_{SRC} \equiv mac_{AP}$ $mac_{DST} \equiv mac_{STA}$	–
τ_6	s_4	s_4	ACK	SIFS	∞	$mac_{SRC} \equiv mac_{STA}$ $mac_{DST} \equiv mac_{AP}$	–
τ_7	s_4	s_1	Sleep	DIFS	∞	$mac_{SRC} \equiv mac_{STA}$ $mac_{DST} \equiv mac_{AP}$	–
τ_8	s_4	s_5	Power Save Probe	DIFS	∞	–	$mac_{SRC} \leftarrow mac_{IDS}$ $mac_{DST} \leftarrow mac_{CD1}$
τ_9	s_5	s_0	Power Save Probe Response	$LI_2 - \delta$	LI_2	$mac_{SRC} \equiv mac_{D1}$ $mac_{DST} \equiv mac_{IDS}$	–
τ_{10}	s_2	s_6	Proper Wake	$LI - \delta$	LI	$mac_{SRC} \equiv mac_{STA}$ $mac_{DST} \equiv mac_{AP}$	–
τ_{11}	s_6	s_7	ACK	SIFS	∞	$mac_{SRC} \equiv mac_{AP}$ $mac_{DST} \equiv mac_{STA}$	–
τ_{12}	s_7	s_7	Data	DIFS	∞	$mac_{SRC} \equiv mac_{AP}$ $mac_{DST} \equiv mac_{STA}$	–
τ_{13}	s_7	s_7	ACK	SIFS	∞	$mac_{SRC} \equiv mac_{STA}$ $mac_{DST} \equiv mac_{AP}$	–
τ_{14}	s_7	s_1	Sleep	DIFS	∞	$mac_{SRC} \equiv mac_{STA}$ $mac_{DST} \equiv mac_{AP}$	–
$\tau_{15}{}^{'}$	$s_5{}^{'}$	$s_0{}^{'}$	Power Save Probe Response	DIFS	$LI_2 - \delta$	$mac_{SRC} \equiv mac_{STA}$ $mac_{DST} \equiv mac_{AP}$	–

$\Im = \{\tau_1, \tau_2, \tau_3, \tau_4, \tau_5, \tau_6, \tau_7, \tau_8, \tau_9, \tau_{10}, \tau_{11}, \tau_{12}, \tau_{13}, \tau_{14}, \tau_1{}^{'}, \tau_2{}^{'}, \tau_3{}^{'}, \tau_4{}^{'}, \tau_5{}^{'}, \tau_6{}^{'}, \tau_8{}^{'}, \tau_{15}{}^{'}\}$

Behavior Under Normal Conditions: States $\{s_0, s_1, s_2, s_3, s_4, s_5, s_6, s_7\}$ and transitions $\{\tau_1, \tau_2, \tau_3, \tau_4, \tau_5, \tau_6, \tau_7, \tau_8, \tau_9, \tau_{10}, \tau_{11}, \tau_{12}, \tau_{13}, \tau_{14}\}$ represent the system under normal conditions.

1) s_0: The model starts at state s_0. In state s_0, the STA is about to send a sleep frame to the AP.

2) $(s_0 \rightarrow s_1)$ At state s_0, the STA sends a sleep frame to the AP and the model reaches state s_1 by traversing transition τ_1. The initial $(\tau_1) \equiv s_0$, final $(\tau_1) \equiv s_1, \sigma \equiv Sleep$.

Here $check(V) = \{--\}$ and $assign(V) = \{mac_{src} \leftarrow mac_{STA}, mac_{dst} \leftarrow mac_{\rm AP}\}$. $mac_{src} \leftarrow mac_{STA}$ is the assignment of SRC MAC address to model variable mac_{src} and $mac_{dst} \leftarrow mac_{\rm AP}$ is the assignment of DST MAC address to model variable mac_{dst}. The sleep frame must be sent after the medium is free for at-least the distributed inter frame spacing (DIFS) duration. Hence the delay duration is DIFS. Depending on the length the transmission the deadline varies, hence the deadline is assumed to be infinite. The "power save handler" generates the "Sleep" event.

Note 1: If model variables (V) are not used then there would have been 2^{48} transitions (and states) from s_0, each representing a possible MAC address (as MAC address are 48 bit) sent in the request frame. As MAC address for every STA is unique, we need all possible MAC address combinations, each represented by a state. This would lead to state explosion problem.

3) $\tau_2 : (s_1 \rightarrow s_2)$ After receiving the sleep frame from the STA, the AP sends an acknowledgment (ACK) to the STA. This is indicated by the transition τ_2. The ACK must be received after short inter frame spacing (SIFS) duration has elapsed. The receipt of ACK frame allows the STA to enter sleep state. While a STA is in sleep state, any frame(s) destined for the STA are buffered at the AP. Here $check(V) = \{mac_{src} \equiv mac_{AP}, mac_{dst} \equiv mac_{STA}\}$. These conditions are checked to verify that the ACK belongs the sleep frame received in the previous step and does not belong to any other STA. As no assignment is done during this transition $assign(V) = \{--\}$. All ACK events have delay-deadline as $[SIFS, \infty]$ in model G shown in Fig. 5.

4) $\tau_3 : (s_2 \rightarrow s_3)$ *and* $\tau_{10} : (s_2 \rightarrow s_6)$ At state s_2 the sender sends a waking frame. Now there arises two possibilities. The STA wakes up before or after the expiry of LI. If the STA wakes up before the expiry of LI (denoted by event "early wake"), the RTDES model takes transition τ_3. The delay-deadline for transition τ_3 is $[DIFS, LI - \delta]$. δ is subtracted from LI as the STAs tend to wake up just before the expiry of LI [27]. If the STA wakes up after the expiry of LI (denoted by event "proper wake"), the RTDES model takes transition τ_{10}. In both the transitions, $check(V) = \{mac_{src} \equiv mac_{STA}, mac_{dst} \equiv mac_{AP}\}$ and $assign(V) = \{--\}$.

5) $\tau_4 : (s_3 \rightarrow s_4)$ *and* $\tau_{11} : (s_6 \rightarrow s_7)$ The transition τ_4, τ_{11} are the ACK frames sent by the AP to the STA. Their explanation is similar to transition τ_2 above. Here $check(V) = \{mac_{src} \equiv mac_{AP}, mac_{dst} \equiv mac_{STA}\}$ and $assign(V) = \{--\}$.

6) $\tau_5, \tau_6 : (s_4 \rightleftarrows s_4)$ *and* $\tau_7 : (s_4 \rightarrow s_1)$ At state s_4 the STA fetches the buffered frame(s) at the AP. The STA also responds with ACK frame for the data successfully received. At state s_4 the transition τ_5 represents the data received by the STA and transition τ_6 represents the ACK sent to the AP by the STA. After the communication between the STA and the AP ends, the STA sends a sleep frame to the AP. Transition τ_7 denotes the sending of sleep frame by the STA to the AP. The delay-deadline of transition τ_7 is $[DIFS, \infty]$. For transition τ_5, $check(V) = \{mac_{SRC} \equiv mac_{AP}$, $mac_{DST} \equiv mac_{STA}\}$ to check whether the data frame(s) is (are) being sent to the the same STA from which the waking frame is obtained in transition τ_3. For transitions τ_6, τ_7 $check(V) = \{mac_{SRC} \equiv mac_{STA}, mac_{DST} \equiv mac_{AP}\}$ checks whether the ACK frame received for the Data frame(s) sent in transition τ_5 belong to the same STA. The checks for transition τ_7 is the same as τ_6. $assign(V) = \{--\}$ for both τ_6, τ_7.

7) $\tau_8 : (s_4 \rightarrow s_5)$ The transition sequence $\langle \tau_3, \tau_4, \tau_5, \tau_6, \tau_7 \rangle$ represents the waking of the STA before the expiry of LI (early wake). As explained earlier this is a suspicious state. The IDS sends power save probe frame to determine if the early wake is initiated by the STA or the attacker. The power save probe consists of a data frame having SRC MAC address of IDS and DST MAC of a dummy STA in it. The delay-deadline of transition τ_8 representing the sending of power save probe is $[DIFS, \infty]$. Here $check(V) = \{--\}$ and $assign(V) = \{mac_{SRC} \leftarrow mac_{IDS}, mac_{DST} \leftarrow mac_{D1}\}$.

8) $\tau_9 : (s_5 \rightarrow s_0)$ The transition τ_9 represents the "power save probe response" to the power save probe sent by the IDS. The delay-deadline of transition τ_9 is $[LI_2 - \delta, LI_2]$. LI_2 is the LI of the dummy STA D_1. The delay-deadline $[LI_2 - \delta, LI_2]$ represents the time after the expiry of LI_2. As power save probe response is received after expiry of LI_2, network is under normal conditions. Here $check(V) = \{mac_{SRC} \equiv mac_{D1}, mac_{DST} \equiv mac_{IDS}\}$ to check that the power save probe response received is for the power save probe sent in transition τ_8. Here $assign(V) = \{--\}$.

9) $\langle \tau_{10}, \tau_{11}, \tau_{12}, \tau_{13}, \tau_{14} \rangle : (s_2 \rightarrow s_6 \rightarrow s_7 \rightarrow s_1)$ The transition sequence $\langle \tau_{10}, \tau_{11}, \tau_{12}, \tau_{13}, \tau_{14} \rangle$ represents waking of the STA after the expiry of LI (Proper Wake). As the STA wakes up after LI expiry, no power save probes are being sent and the network is assumed to operate under normal circumstances. $check(V)$ and $assign(V)$ are the same as the Early Wake scenario explained earlier.

Behavior Under Attack Conditions: For attack conditions we will discuss only those states and transitions that differ from normal scenario. States $\{s_0{}', s_1{}', s_2{}', s_3{}', s_4{}', s_5{}'\}$ and transitions $\{\tau_1{}', \tau_2{}', \tau_3{}', \tau_4{}', \tau_5{}', \tau_6{}', \tau_8{}', \tau_{15}{}'\}$ represent the system under $attack$ conditions.

1) $\tau_{15} : (s_5{}' \rightarrow s_0{}')$ The transition $\tau_{15}{}'$ represents the "power save probe response" to the power save probe sent by the IDS during transition $\tau_8{}'$. The delay-deadline of the transition $\tau_{15}{}'$ representing the power save probe response is $[DIFS, LI_2 - \delta]$. The delay-deadline $[DIFS, LI_2 - \delta]$ represents the time before the expiry of LI_2. Here $check(V) = \{mac_{SRC} \equiv mac_{D1}, mac_{DST} \equiv mac_{AP}\}$. The check is done to ensure that the response received for the power save probe is the one which is sent during the transition $\tau_8{}'$. As no assignment is done, $assign(V) = \{--\}$.

An important thing to note here is that the transitions $\tau_9, \tau_9{}'$ both represent the same event, "power save probe response". However, the delay-deadline are different for both. τ_9 has delay-deadline of $[LI_2 - \delta, LI_2]$ representing fetching of the frame(s) $after$ the expiry of the LI of the dummy STA. On the other hand, $\tau_9{}'$ has delay-deadline of $[DIFS, LI_2 - \delta]$ representing fetching of the frames $before$ the expiry of the LI of the dummy STA. This timing difference is successfully captured by the RTDES framework and proves vital in detecting PS-DoS attack as explained later.

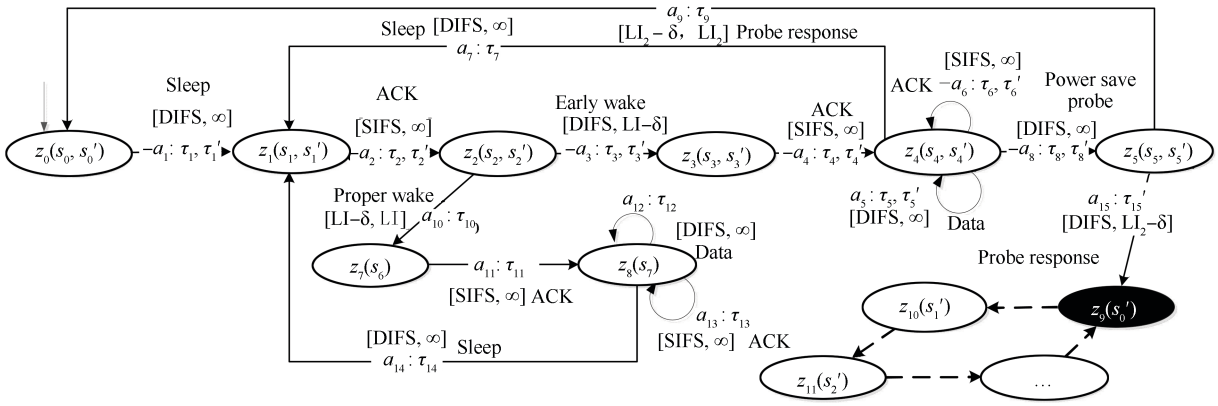


Fig. 6. Diagnoser for PS-DoS attack.

Diagnoser construction for PS-DoS attack:

Fig. 6 is the diagnoser for the RTDES model G shown in Fig. 5. The diagnoser is built following the steps listed in Algorithm 1. Some of the initial steps for this example are as follows.

1) The initial state of the diagnoser, i.e., Z_0 is obtained as follows. X_0 is partitioned into measurement equivalent subsets of G-states which in our case is one, i.e., $X_0 = \{X_{01}\}$; there is only a single initial G-state s_0 (as shown in Fig. 5) and $X_{01} = \{s_0\}$. Since X_0 can be partitioned into only one subset of measurement equivalent initial G-states, $Z_0 = z_{01}$ and $z_{01} = \mathcal{U}^*(X_{01}) = \mathcal{U}^*(s_0) = \{s_0, s_0{}'\}$. Thus, there is only one initial O-node z_0 (as shown in Fig. 6) and $z_0 = \{s_0, s_0{}'\}$. As $z_0 = \{s_0, s_0{}'\}$ is formed using measurement equivalent states $\{s_0, s_0{}'\}$ the state of the system is uncertain as s_0 corresponds to a normal state while $s_0{}'$ corresponds to attack state. So, given the diagnoser O-node z_0 it cannot be predicted whether the PS-DoS attack has occurred or not. Thus z_0 is an F_i-uncertain O-node.

2) The outgoing O-transitions from z_0 are obtained as follows. Here, $\Im_{mz0} = \{\tau_1, \tau_1{}'\}$ which are all the outgoing measurable transitions from O-node in z_0. Now, $A_{z_{01}} = \{\tau_1, \tau_1{}'\}$, as $\{\tau_1, \tau_1{}'\}$ forms a set of measurement equivalent transitions. Corresponding to $\{\tau_1, \tau_1{}'\}$ there is O-transition a_1.

3) The destination O-node corresponding to a_1 is obtained as follows. $z_{0a_1}^+ = \{s_1, s_1{}'\}$ as a_1 comprises G-transitions $\{\tau_1, \tau_1{}'\}$ and $final(\tau_1) = s_1$ and $final(\tau_1{}') = s_1{}'$. Further, $z_1^+ = \{s_1, s_1{}'\}$ as $\mathcal{U}^*(s_1) = \{s_1\}$ and $\mathcal{U}^*(s_1{}') = \{s_1{}'\}$ since there is no unmeasurable transition emanating from either state s_1 or state $s_1{}'$. Thus, the destination O-node of the O-transition a_1 is $z_1 : \{s_1, s_1{}'\}$.

In order for the diagnoser to detect PS-DoS attack, the diagnoser should reach the O-node z_9 which is an F_i-certain O-node (attack certain O-node). However, if the diagnoser gets stuck in an F_i-indeterminate cycle before reaching z_9 it leads to non-diagnosability. The RTDES diagnoser for PS-DoS attack shown in Fig. 6 does not have any F_i-indeterminate cycles. The cycle comprising of the O-transition sequence $\langle a_2, a_3, a_4, a_7 \rangle$ is an uncertain cycle and not an indeterminate cycle. The reason for that is the O-transition a_7 can occur only in normal conditions and not under attack conditions. So if the diagnoser moves in $\langle a_2, a_3, a_4, a_7 \rangle$ cycle, the network is operating under normal conditions. Similar explanation can be given for other two uncertain cycles comprising of transition sequence $\langle a_2, a_{10}, a_{11}, a_{14} \rangle$ and $\langle a_1, a_2, a_3, a_4, a_8, a_9 \rangle$. As the diagnoser does not have any F_i-indeterminate cycles, the diagnoser is diagnosable. Thus the RTDES framework successfully detects the PS-DoS attack in 802.11 networks.

F. An Example of Attack Detection Using Diagnoser

The diagnoser for the RTDES model G shown in Fig. 5 is shown in Fig. 6. The parameters taken for the explanation of PS-DoS attack are shown in Table II. The STA sends a sleep frame and receives an ACK from the AP. The O-transitions a_1, a_2 denote that. The IDS sniffs a waking frame from a STA at 1600 ms. As 1600 ms is less than the LI expiry time of 1950 ms, IDS treats this as suspicious activity. "Power Save Handler" generates the event "Early Wake" and the diagnoser reaches sate z_3. The AP responds with an ACK frame and the STA and the AP continue data exchange. "Power Save Handler" generates the events "ACK"and "Data" respectively. The diagnoser O-transitions a_5, a_6 denote the data exchange and ACK events respectively. Currently the diagnoser is in O-node z_4 which is composed of states s_4 and $s_4{}'$ which are measurement equivalent. As a result, at z_4 it cannot be ascertained whether an attack has taken place or not. To ascertain the presence of an attacker, the IDS sends a power save probe frame which has SRC MAC as IDS MAC address DST MAC address as the dummy STA's MAC address. "Power Save Handler" generates the event "Power Save Probe" and the diagnoser moves to O-node z_5. Now, assume that the IDS sniffs a waking frame meant to fetch the buffered frame(s) for the dummy STA at 3600 ms instead of the dummy STA's LI expiry interval of 4950 ms. "Power Save Handler" generates the event "Power Save Probe Response" for the event. Dummy STA can never wake up early for data fetch which means that the query to fetch the frame(s) is sent by the attacker. The early fetching is denoted by O-transition a_{15} and the diagnoser reaches O-node z_9 which is an F_i-certain O-node. Thus the diagnoser successfully determines the presence of the attacker. Once the attack is detected, the diagnoser traverses only $F_i$certain O-nodes sequence $\langle z_9, z_{10}, z_{11}, \ldots, z_9 \rangle$.

Now we look into an example corresponding to normal network conditions. Under normal network condition the diagnoser must not reach any F_i-certain O-node. The states and transitions till the diagnoser until reaching O-node z_2 are

similar to those explained above for the attack scenario. We assume the case of proper wake for normal network scenarios. The IDS sniffs a wake frame from the STA. The "Power Save Handler" generates the event "Proper Wake". As the waking frame is received after the expiry of the LI of the STA, the IDS does not send any power save probes. The STA receives an ACK from the AP. The "Power Save Handler" generates the event "ACK". The diagnoser traverses O-transition sequence $\langle a_{10}, a_{11}\rangle$ during this process. At O-node z_8 the STA receives buffered frame(s) from the AP and sends ACK back to the AP. The "Power Save Handler" generates the events "Data" and "ACK" respectively. The O-transitions for "Data" and "ACK" events are a_{12}, a_{13} respectively. After the buffered frame(s) is(are) retrieved, the STA sends a sleep frame to the AP to inform the AP that it is entering into sleep state. The "Power Save Handler" generates the event "Sleep". The O-transition for "Sleep" event is a_{14}. So, the O-nodes sequence traversed during normal conditions by the diagnoser is given as $\langle z_0, z_1, z_2, z_7, z_8\rangle$, none of which are F_i-certain O-node. So diagnoser reports it correctly as normal activity.

TABLE II
PARAMETERS VALUES FOR BEACON INTERVAL AND LISTEN INTERVALS

Parameter	Real STA	Dummy STA
Beacon Interval (BI)	100 ms	100 ms
Listen Interval (LI)	20	50
Delta (δ)	50 ms	50 ms
BI * LI	2000 ms	5000 ms
BI * LI $-\delta$	1950 ms	4950 ms
Early Wake (For this example)	1600 ms	3600 ms

IV. RESULTS AND DISCUSSIONS

In this section, we discuss the accuracy, detection rate and network load of the proposed IDS followed by proving the correctness of the RTDES diagnoser.

The setup for the proposed the PS-DoS attack is shown in Fig. 7. We have an open AP with SSID "FreeWiFi", IDS, three STA machines and one attacker machine. The IDS has two network interfaces. The attacker machine (M) is equipped with BackTrack 5R3 [35]. The IDS is implemented in C language

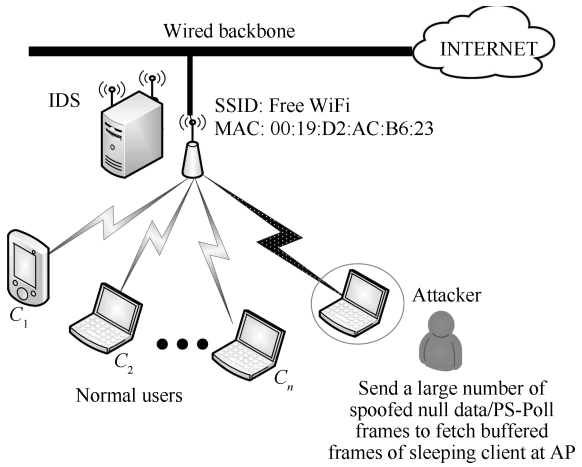


Fig. 7. Experimental setup.

on Ubuntu 12.04 machine. Scapy is used for injecting power save probes in the network. The three STAs (C1, C2 and C3) have Ubuntu 10.04, Windows XP and Windows 7. The experiment has run for a duration of 3 hours and the attack is launched randomly on different systems. The network load under normal traffic and under probing environment is shown in Fig. 8.

A. Accuracy, Detection Rate and Network Load of Proposed IDS

The metrics used for measuring the performance of IDS are accuracy and detection rate.

Accuracy is the proportion of the total number of attacks that are correctly detected. It is determined using the expression:

$$Accuracy = \frac{TP}{TP + FP}.$$

Detection rate is defined as the number of attacks detected by the IDS to the total number of attacks actually present.

$$Detection\ Rate = \frac{TP}{TP + FN}$$

here TP is true positive, FP is false positive, FN is false negative. A TP is a instance, which is actually an attack and is declared as attack by the IDS. An FP is a case when IDS treats a normal activity as attack activity. An FN arises when the IDS treats an attack activity as normal.

The memory and CPU consumption are important parameters when considering the scalability of IDS. However, these parameters are secondary parameters and do not have any effect on the detection rate and accuracy. The CPU and memory consumption of an IDS depend heavily on the implementation procedure followed. Hence in this work we only concentrate on accuracy and detection rate of the RTDES based IDS [36].

An interesting observation can be made from Table III that the detection rate is 99 % even when 100 % power save probes are sent. This is due to that fact that Wi-Fi is a lossy medium. It is quite possible that the IDS fails to capture the spoofed PS-Poll frame sent by the attacker to the AP. As the spoofed PS-Poll frame is not captured, no probes are sent by the IDS. In such cases, even if the IDS captures all other PS-Poll frames successfully and sends probes for the captured PS-Poll frames (so 100 % percent probing from IDS perspective), the detection rate would be still less than 100 % (99 % in this case) due to the failure in capturing the (spoofed) PS-Poll frame. In general, various kinds of frame losses may lead to a reduced detection rate.

The sending of power save probes increases the network traffic. However, there is a trade-off between the sending of probes and detection rate. As seen in Table III, the detection rate falls with the reduction in the probes sent. This is because as the number of probes sent are reduced, the chances of the attacker escaping from the detection increases. For example, consider an early waking frame sent by the attacker for which no power save probe is sent. The attacker fetches the buffered frame(s) using the spoofed waking frame but is not caught as no power save probe frame is sent. Although, sending of lesser

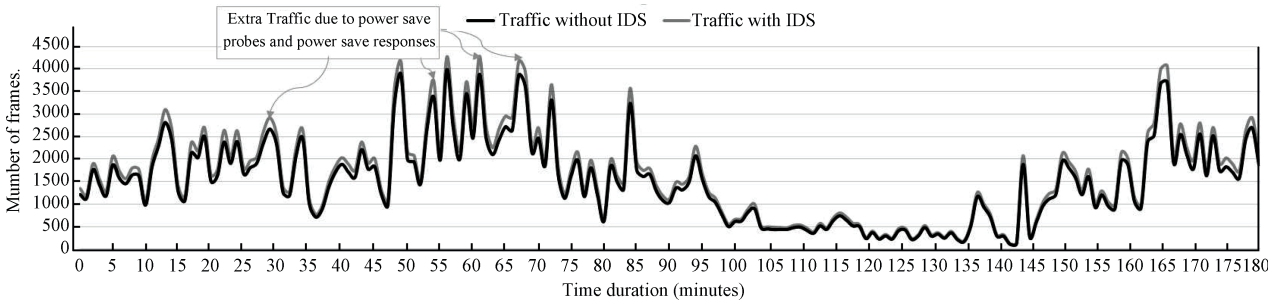


Fig. 8. Network traffic with and without use of IDS.

probes conserves network bandwidth, it affects the detection rate. However, no proportional drop is observed when the percentage of probes is reduced.

From Table III, it can be seen that the detection rate of the IDS is 59 % even in those cases when the probe is sent only 10 % of the time. This is explained as follows: as the proposed IDS is a detection system and not a prevention system, it behaves aggressively once the presence of the attack is ascertained. To be specific, once an STA is detected to be under attack, if the STA under question ever wakes up before its sleep schedule and fetches buffered frame(s), IDS sends an alert. As already discussed, the STAs do not usually wake up before its sleep schedule and so any such observation can be notified as attack once an attack on that STA is confirmed in the past. This approach taken by the IDS seems plausible as the attacker is assumed to fetch all the buffered frame(s) for a STA at the AP while the STA is in sleep state. Hence, the detection rate is higher even under the case when the probes sent are low. For example, consider the network setup shown in Fig. 7. Assume that the attacker is under the process of fetching buffered frame(s) for the STA C2 of the network. As the IDS injects few probes into the network, once it is ascertained that the attacker is fetching the buffered frame(s) for C2 (using power save probe frame), all the subsequent frame(s) fetched using the MAC of the STA C2 are marked as frame(s) captured by the attacker without sending any additional probes.

TABLE III
DETECTION RATE AND ACCURACY OF IDS

% Probes	# of Attacks Launched	Detection Rate (%)	Accuracy (%)
10	400	59	100
20	400	62	100
30	400	66	100
40	400	72	100
50	400	80	100
60	400	84	100
70	400	89	100
80	400	92	100
90	400	97	100
100	400	99	100

B. Network Load due to Power Save Probes

Power save probes and the dummy STAs are required for detecting the PS-DoS attack, which lead to an increase in the network traffic. Power save probes are sent only in those cases when a sleeping STA has buffered frame(s) and is (are) fetched before the expiry of the STA's LI. The graph in Fig. 8 shows the network traffic with and without the use of IDS during a three hour test run. In the test run, the IDS varies the dummy STAs from one to five depending upon the frequency of improper wakes by the STAs in the three hour test run. As explained earlier, the number of dummy STAs are decided by the IDS. Different runs may have different number of dummy STAs depending on the characteristics of the network and the number of observed early wakes. So, it can be observed that at few points the overhead is slightly above the normal traffic. At places where the frequency of improper wakes is low, the traffic with and without IDS almost overlaps. This is due to the fact that when the frequency of improper wakes by the STAs is low, fewer power save probes are sent to the dummy STAs by the IDS and also the number of dummy STAs is small. In the test run, an average of 1.14 % to 6.23 % increase in the network traffic is observed due to overhead involved as a result of the dummy STAs.

Now, we discuss intuitively why the traffic overheard due to the power save probes and presence of the dummy STAs is not high. Under normal circumstances when the STAs do not wake up early, the IDS does not send any power save probes to the dummy STAs. So, the proposed scheme does not add any kind of overhead under normal network conditions. Under attack conditions, a power save probe is sent only when buffered frame(s) is (are) present at the AP and the STA wakes up early. Even under those circumstances, only 1 power save probe frame is sent per STA irrespective of the number of buffered frame(s) present at the AP for that STA. So, the overhead caused by sending probes is low.

In the proposed scheme, a minimum number of dummy STAs are present in the network at a given instant. Though, this feature is implementation specific, a bare minimum of one dummy STA is required for the detection scheme to work correctly. By observing the frequency of improper sleep and wake cycles, the IDS can increase or decrease the number of dummy STAs in the network. This dynamic behavior for the presence of the dummy STAs adopted by the IDS coupled with the sending of only 1 power save probe for every early wake per STA leads to sparse overhead of the traffic generated by the dummy STAs. So, the overhead induced by the dummy STAs in the network is low.

C. Advantages and Correctness of RTDES Diagnoser

The RTDES modeling helps us to formalize a given system and check for correctness and completeness [34]. In this

section we show the correctness and completeness of the proposed scheme by considering all the exhaustive cases of the PS-DoS attack. For each and every case considered, we have shown that the proposed scheme successfully detects the presence of the PS-DoS attack. The proving of correctness and completeness by considering all the exhaustive cases helps to ascertain that the attacker does not escape from detection thus making the detection scheme robust. If formal modeling is not taken into consideration, the system might contain certain loopholes unobserved, which an attacker could exploit in order to evade detection. Formalism helps us to verify all the possible states a system may traverse during its lifetime thereby ensuring no such loopholes are left.

There are a number of ways in which the PS-DoS attack can be launched. In order to prove the correctness of the diagnoser shown in Fig. 6, we consider 6 possible cases of the arrival of null data frame which are depicted in the tree shown in Fig. 9. We denote S as the victim STA, M as the attacker, G as the genuine STA and D_1 as the dummy STA.

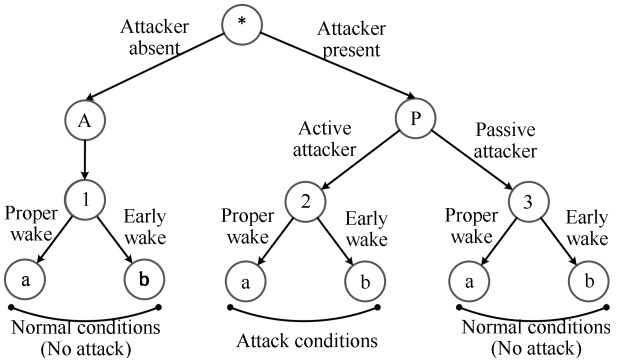


Fig. 9. Six possible cases of arrival of null data frames.

1) Attacker absent (normal network conditions):

a) Proper Wake. In this case S wakes after the expiry of its LI. Initially S sends a sleep frame to the AP and subsequently receives an ACK from the AP. The diagnoser takes O-transition sequence $\langle a_1, a_2\rangle$. S sends a waking frame after expiry of LI and receives an ACK from the AP. The diagnoser takes O-transition sequence $\langle a_{10}, a_{11}\rangle$. At state z_8, S fetches the buffered frame(s) from the AP and sends ACK back to the AP. This is indicated by O-transition sequence $\langle a_{12}, a_{13}\rangle$. After exchanging the frame(s), S sends a sleep frame to the AP and subsequently the diagnoser reaches state z_1. None of the states traversed by the diagnoser contains an F_i-certain (attack) state. Hence the diagnoser treats the scenario as normal network activity.

b) Early Wake. Here S sends a waking frame before the expiry of its LI. This waking frame could be a spoofed one. To verify the genuineness of the waking frame power save probe is sent to D_1. The initial sleep and ACK frames are represented by the O-transition sequence $\langle a_1, a_2\rangle$. The early wake is denoted by the O-transition a_3. The O-transition a_4 is the ACK sent by the AP in response to early wake. At state z_4 S fetches the buffered frame(s) from the AP and sends back the ACK to the AP. The O-transition sequence $\langle a_5, a_6\rangle$ denotes the fetching and ACK events. The sending of power save probe frame is denoted by O-transition a_8. The power save probe response is obtained $after$ the expiry of the LI of D_1. M is assumed to fetch every buffered frame(s) available instantly. As M did not fetch the frame(s) available for D_1 the early waking frame sent by S must be genuine. The diagnoser takes the O-transition a_9. This completes the detection cycle of IDS and it returns to start state and monitors again for the PS-DoS attack. Even though an early waking frame is received, no states traversed by the diagnoser is an F_i-certain (attack) state, which exemplifies that the network is under normal conditions.

2) Attacker present (active attacker)

M needs to send a wake frame before the expiry of LI of S in order to successfully launch the PS-DoS attack. The diagnoser must reach an attack certain state to ascertain that the attacker is caught.

a) Proper Wake. In this case, M sends a spoofed waking frame after the expiry of the LI of S. S also wakes up after expiry of its LI to check for buffered frame(s) at the AP. Both S and M are awake simultaneously and both receive the buffered frame(s) from the AP. The diagnoser follows the O-transition sequence $\langle a_1, a_2, a_{10}, a_{11}, a_{12}, a_{13}\rangle$. As none of the nodes are F_i-certain (attack) the diagnoser treats this as normal activity. As both S and M are awake simultaneously M does not cause any frame losses to S defeating the purpose of the PS-DoS attack. Technically, the PS-DoS attack is launched by M but has not caused any frame losses to S. Due to this, the diagnoser treats this as normal activity as it causes no frame losses.

b) Early Wake. The O-states and O-transitions till z_5 are as explained in Early Wake scenario of Case 1. M is assumed to fetch every buffered frame(s) and does not possess the knowledge of the MAC address of the D_1. Due to this M fetches the buffered frame(s) targeted at D_1 $before$ the expiry of LI of D_1. This is indicated by the O-transition a_{15} which takes the diagnoser to state z_9 which is an F_i-certain (attack) state. As an F_i-certain (attack) state is reached the diagnoser detects the PS-DoS attack in the network. As failures are assumed to be permanent, the diagnoser moves into an attack certain cycle $\langle z_9, z_{10}, z_{11} \ldots, z_9\rangle$.

3) Attacker present (passive attacker).

In this case, M only eavesdrops on the network. It does not send a waking frame in spite of buffered frame(s) at the AP. As attacker does not respond, no PS-DoS attack is possible.

a) Proper wake: Same as proper wake of Case 1.

b) Early wake: Same as early wake of Case 1.

The network remains under normal circumstances in-spite of the presence of M. The diagnoser correctly labels this condition as normal activity of the network.

V. Conclusion

In this paper, RTDES framework has been adapted and used for detecting the PS-DoS attack in 802.11 networks. In the PS-DoS attack an attacker fetches the buffered frame(s) of the genuine STAs while they are in sleep state by injecting spoofed null data or a PS-Poll frame. As per 802.11 standard null data frame or a PS-Poll frame are sent in clear text, which make them easier to spoof. The PS-DoS attack causes frame losses to the victim STAs. Existing schemes for the

PS-DoS attack detection and prevention suffer from expensive setup, maintenance, scalability, deployment issues and lack of formal frameworks for design of the attack detector etc. The proposed RTDES based IDS helps to detect the PS-DoS attack with high accuracy and low false positive rate and overcomes the shortcomings of existing approaches. The proposed IDS make use of active probing in order to create differences between normal and attack scenario. Though injecting power save probes increases network traffic, but with slight increase in overall traffic presence of the PS-DoS attack is detected accurately. Another major advantage of the RTDES based IDS is that it does not require protocol modifications, use of any encryption algorithms or firmware upgrades either on the AP or on the end user side. Besides this, the proposed methodology can be applied on legacy as well as modern day systems.

REFERENCES

[1] Aircrack-ng Suite. [Online]. Available: http://www.aircrack-ng.org/.

[2] Scapy-A powerful interactive packet manipulation program.[Online]. Available: http://www.secdev.org/projects/scapy/

[3] J. Bellardo and S. Savage, "802.11 denial-of-service attacks: Real vulnerabilities and practical solutions," in *Proc. 12th Conf. USENIX Security Symposium*, Berkeley, CA, USA, 2003.

[4] "Information technology-telecommunications and information exchange between systems-Local and metropolitan area networks-Specific requirements-Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," IEEE Std 802.11-2007, pp.C1–1184, 2017.

[5] R. Bansal, S. Tiwari, and D. Bansal, "Non-cryptographic methods of MAC spoof detection in wireless LAN," in *Proc. 16th IEEE International Conf. Networks*, New Delhi, India, 2008, pp. 1−6.

[6] P. García-Teodoro, J. Díaz-Verdejo, G. Maciá-Fernández, and E. Vázquez, "Anomaly-based network intrusion detection: Techniques, systems and challenges," *Comput. Secur.*, vol. 28, no. 1−2, pp. 18−28, Feb. 2009.

[7] S. Gayaka and B. Yao, "Fault detection, identification and accommodation for an electro-hydraulic system: An adaptive robust approach," *IFAC Proc. Vol.*, vol. 41, no. 2, pp. 13815−13820, Jul. 2008.

[8] A. Alaghi, N. Karimi, M. Sedghi, and Z. Navabi, "Online NoC switch fault detection and diagnosis using a high level fault model," in *Proc. 22nd IEEE International Symposium on Defect and Fault-Tolerance in VLSI Systems, DFT 0́7.*, Rome, Italy, 2007, pp. 21−29.

[9] L. F. Gonçalves, J. L. Bosa, T. R. Balen, M. S. Lubaszewski, E. L. Schneider, and R. V. Henriques, "Fault detection, diagnosis and prediction in electrical valves using self-organizing maps," *J. Electron. Test.*, vol. 27, no. 4, pp. 551−564, Apr. 2011.

[10] S. Hong and S. Kim, "Lizard: Energy-efficient hard fault detection, diagnosis and isolation in the ALU," in *Proc. IEEE International Conf. Computer Design (ICCD)*, Amsterdam, the Netherlands, 2010, pp. 342−349.

[11] X. Yu and J. Jiang, "Hybrid fault-tolerant flight control system design against partial actuator failures," *IEEE Trans. Control Syst. Technol.*, vol. 20, no. 4, pp. 871−886, Jul. 2012.

[12] C. F. Chien, C. Y. Hsu, and P. N. Chen, "Semiconductor fault detection and classification for yield enhancement and manufacturing intelligence," *Flex. Serv. Manuf. J.*, vol. 25, no. 3, pp. 367−388, Sep. 2013.

[13] S. J. Youk, S. S. Yoo, C. Y. Lee, J. H. Kho, and G. Lee, "Development of fault detection system in air handling unit," in *Proc. International Conf. Convergence and Hybrid Information Technology*, Daejeon, Korea, 2008, pp. 287−292.

[14] S. Lafortune, "Diagnosis of discrete event systems," in *Encyclopedia of Systems and Control*. London, UK: Springer, 2014, pp. 1−10.

[15] J. Zaytoon and S. Lafortune, "Overview of fault diagnosis methods for discrete event systems," *Ann. Rev. Control*, vol. 37, no. 2, pp. 308−320, Dec. 2013.

[16] C. Mahulea, C. Seatzu, M. P. Cabasino, and M. Silva, "Fault diagnosis of discrete-event systems using continuous petri nets," *IEEE Trans. Syst., Man Cyber. A: Syst. Hum.*, vol. 42, no. 4, pp. 970−984, Jul. 2012.

[17] M. P. Fanti, A. M. Mangini, and W. Ukovich, "Fault detection by labeled petri nets in centralized and distributed approaches," *IEEE Trans. Automat. Sci. Eng.*, vol. 10, no. 2, pp. 392−404, Apr. 2013.

[18] M. Chang, W. Dong, Y. D. Ji, and L. Tong, "On fault predictability in stochastic discrete event systems," *Asian J. Control*, vol. 15, no. 5, pp. 1458−1467, Sep. 2013.

[19] R. H. Kwong and D. L. Yonge-Mallo, "Fault diagnosis in discrete-event systems: Incomplete models and learning," *IEEE Trans. Syst. Man Cyber. B: Cyber.*, vol. 41, no. 1, pp. 118−130, Feb. 2011.

[20] P. Bhowal, D. Sarkar, S. Mukhopadhyay, and A. Basu, "Fault diagnosis in discrete time hybrid systems-a case study," *Inf. Sci.*, vol. 177, no. 5, pp. 1290−1308, Mar. 2007.

[21] N. Hubballi, S. Biswas, S. Roopa, R. Ratti, and S. Nandi, "LAN attack detection using discrete event systems," *ISA Trans.*, vol. 50, no. 1, pp. 119−130, Jan. 2011.

[22] C. G. Cassandras and S. Lafortune, *Introduction to Discrete Event Systems*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.

[23] R. Sekar, M. Bendre, D. Dhurjati, and P. Bollineni, "A fast automaton-based method for detecting anomalous program behaviors," in *Proc. 2001 IEEE Symposium on Security and Privacy*, Oakland, CA, USA, 2001, pp. 144−155.

[24] S. Whittaker, M. Zulkernine, and K. Rudie, "Towards incorporating discrete-event systems in secure software development," in *Proc. Third International Conf. Availability, Reliability and Security, 2008. ARES 08.*, Barcelona, Spain, 2008, pp. 1188−1195.

[25] Z. I. Qureshi, B. Aslam, A. Mohsin, and Y. Javed, "A solution to spoofed PS-poll based denial of service attacks in IEEE 802.11 WLANs," in *Proc. 11th Conf. 11th WSEAS International Conf. Communications*, vol. 11, pp. 7−11, Jul. 2007.

[26] L. F. Meiners, "But...my station is awake! (Power Save Denial of Service in 802.11 Networks)," [Online]. Available: http://corelabs.coresecurity.com/ index.php?module=Wiki&action= view&type= publication&name=WiFiPowerSaveDoS.

[27] W. J. Gu, Z. M. Yang, D. Xuan, W. J. Jia, and C. Que, "Null data frame: A double-edged sword in IEEE 802.11 WLANs," *IEEE Trans. Parallel Distrib. Syst.*, vol. 21, no. 7, pp. 897−910, Jul. 2010.

[28] D. B. Faria and D. R. Cheriton, "Detecting identity-based attacks in wireless networks using signalprints," in *Proc. 5th ACM Workshop on Wireless Security*, New York, NY, USA, 2006, pp. 43−52.

[29] B. Azimi-Sadjadi, A. Kiayias, A. Mercado, and B. Yener, "Robust key generation from signal envelopes in wireless networks," in *Proc. 14th ACM Conf. Computer and Communications Security*, Alexandria, Virginia, USA, 2007, pp. 401−410.

[30] Y. Y. Chen, W. Trappe, and R. P. Martin, "Detecting and localizing wireless spoofing attacks," in *Proc. 4th Annual IEEE Communications Society Conf. Sensor, Mesh and Ad Hoc Communications and Networks, 2007. SECON0́7.*, San Diego, CA, USA, 2007, pp. 193−202.

[31] J. Wright, "How 802.11w will improve wireless security," [Online]. Available: http://www.networkworld.com/article/2312251/network-security/ how-802-11w-will-improve-wireless-security.html. Accessed on: May 2006.

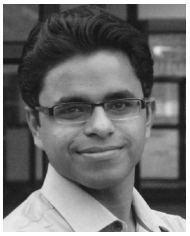
[32] CWNP, "Wireless LAN security and IEEE 802.11w."[Online]. Available: http://www.cwnp.com/cwnpwifiblog/wireless-lan-securityand-ieee-802-11w/.

[33] S. Biswas, D. Sarkar, and S. Mukhopadhyay, "Diagnosability of delay-deadline failures in fair real time discrete event models," *Int. J. Syst. Sci.*, vol. 41, no. 7, pp. 763−782, Jul. 2010.

[34] M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen, and D. Teneketzis, "Diagnosability of discrete-event systems," *IEEE Trans. Automat. Control*, vol. 40, no. 9, pp. 1555−1575, Sep. 1995.

[35] "BackTrack." [Online]. Available: http://www.backtrack-linux.org/.

[36] N. J. Puketza, K. Zhang, M. Chung, B. Mukherjee, and R. A. Olsson, "A methodology for testing intrusion detection systems," *IEEE Trans. Softw. Eng.*, vol. 22, no. 10, pp. 719−729, Oct. 1996.



Mayank Agarwal is a Ph.D candidate at the Indian Institute of Technology, Guwahati. He received his bachelors degree from Sardar Patel Institute of Technology, Mumbai in 2009. His research interests include wireless & network security, discrete event system modeling.

Sanketh Purwar is a Senior R&D Engineer at HP PPS R&D laboratory in Bangalore. He received his bachelors degree from Indian Institute of Technology, Guwahati in Computer Science in 2013. His research interests include application of networks, embedded systems in Internet of things.

Santosh Biswas received the B.E. degree from NIT, Durgapur, India, in 2001. He has completed his M.S. and Ph.D from IIT Kharagpur, India, in 2004 and 2008, respectively. He works as an Associate Professor in the Department of Computer Science and Engineering, IIT Guwahati. His research interests include networking, VLSI testing and discrete event systems.

Sukumar Nandi received his B Tech (Applied Physics) in electrical engineering (Specialization in Instrumentation) from Calcutta University. He received his M Tech in computer science from Calcutta University and Ph. D. in Computer Science and Engineering from Indian Institute of Technology Kharagpur. He is a Professor in the Department of Computer Science and Engineering, IIT Guwahati. He is also involved in several international conferences as member of advisory board/ Technical Programme Committee. He has published more than 150 Journals/Conferences papers. He is Senior Member of IEEE and Fellow of the Institution of Engineers (India). His research interests include computer networks, network security, and data mining.