PDR: A Prevention, Detection and Response Mechanism for Anomalies in Energy Control Systems

Cristina Alcaraz and Meltem Sönmez Turan

National Institute of Standards and Technology, Gaithersburg, MD
{mariacristina.alcaraz,meltem.turan}@nist.gov

Abstract. Prevention, detection and response are nowadays considered to be three priority topics for protecting critical infrastructures, such as energy control systems. Despite attempts to address these current issues, there is still a particular lack of investigation in these areas, and in particular in dynamic and automatic proactive solutions. In this paper we propose a mechanism, which is called PDR, with the capability of anticipating anomalies, detecting anomalous behaviours and responding to them in a timely manner. PDR is based on a conglomeration of technologies and on a set of essential components with the purpose of offering situational awareness irrespective of where the system is located. In addition, the mechanism can also compute its functional capacities by evaluating its efficacy and precision in the prediction and detection of disturbances. With this, the entire system is able to know the real reliability of its services and its activity in remote substations at all times.

Keywords: Wide-Area Situational Awareness, Prevention, Detection, Response, Energy Control Systems, Industrial Wireless Sensor Networks, MANET and the Internet.

1 Introduction

Modernisation of our critical energy control infrastructures is bringing a set of unexplored and unsolved challenges. Most of them are mainly related to the need to find a desirable trade-off between operational performance in (almost) real-time, and protection against serious threats. These threats do not necessarily have to be cyber-attacks [1]. They can be associated with unforeseen or abrupt changes registered within the system, such as a power surge in generators or a voltage reduction in transmission lines. If these unexpected situations are not controlled properly, they may trigger a serious effect that may lead to local, regional or national outages and/or blackouts, with the possibility of spreading on its own to other countries. This is the case of the well-known blackout of the August 14, 2003 that occasioned an economic and social crisis between two countries; U.S. and Canada. Unfortunately, this kind of event has not been the only one that has happened in recent years [2].

Considering the application domain and its sensitive nature, this protection should consist of proactive and reactive solutions based on dynamic and automatic methods. The reason lies in that the vast majority of energy control subsystems (e.g. substations) are distributed at distant-geographic locations in which the control is normally limited to a few human operators in the field. This need was also identified by NIST in [3], and NIST classified this need as one of the eight priorities to be taken into account when protecting Critical Infrastructures (CIs). This priority, known as *Wide-Area Situational Awareness* (WASA), focuses on supervising and controlling the performance of underlying systems located over large geographic areas in (almost) real-time. This includes anticipating, detecting and responding to problems before they can cause disruptions.

Given this, we present a dynamic solution that tries to cover some of the stated points for WASA, such as prevention, detection and response. The proposed approach, called here as PDR, is based on four main technologies; Industrial Wireless Sensor Networks (IWSNs), Mobile Ad-Hoc Networks (MANETs), the Internet and the ISA100.11a standard [4]. We have selected these technologies as each one of them offers an attractive set of benefits for local and remote protection [5, 6]. Moreover, the architecture suggested for PDR is also able to evaluate by itself the level of precision of the schemes proposed for detection and prevention. In this way, the Supervisory Control and Data Acquisition (SCADA) Center is made aware of the accuracy and functionality of the approach, and remotely control the situation at all times and any time.

The paper is organised as follows. Section 2 introduces the general architecture of PDR together with the technologies that play a special role within our approach. We describe in detail the components that comprise the architecture in Section 3, whilst three proactive and reactive methods are discussed in the remainder of the same section. Section 4 concludes the paper and outlines future work.

2 PDR: General Architecture and Technologies

The architecture proposed for PDR is mainly based on IWSNs, MANETs, the Internet, and ISA100.11a. Figure 1 illustrates an example of the collaboration and cooperation of such technologies for control and supervision of energy generation and distribution systems. This figure also represents, in a general way, the operation in field and energy distribution from bulk generations systems (e.g. nuclear systems, hydroelectric systems, wind farms, and others) to urban areas [3].

For electricity production, the majority of generation systems have to be connected to generators to induce mechanical energy into electric energy to a low voltage. For increasing the level of voltage and its transmission over long distances, the system makes use of large electricity transmission lines with transformers (transmission system). To distribute the power to urban areas, the voltage load must be downloaded into substations reducing its level of voltage (distribution system). Both transmission and distribution substations are based on transformers, control devices (e.g. Remote Terminal Units (RTUs)), industrial meters, sensors and industrial engineering devices. Any activity in field must be supervised, either locally or remotely, and any information produced or sensed must be sent to a centralised system for purposes of control and register. All of this control system is commonly known as a SCADA System.

Unfortunately, this complex circuit of power generation and distribution is quite sensitive to unexpected events. This means that one fault registered in a local point of the system could trigger a change in its normal behaviour, probably leading to a cascading effect towards other CIs [7]. To control and coordinate these types of unforeseen

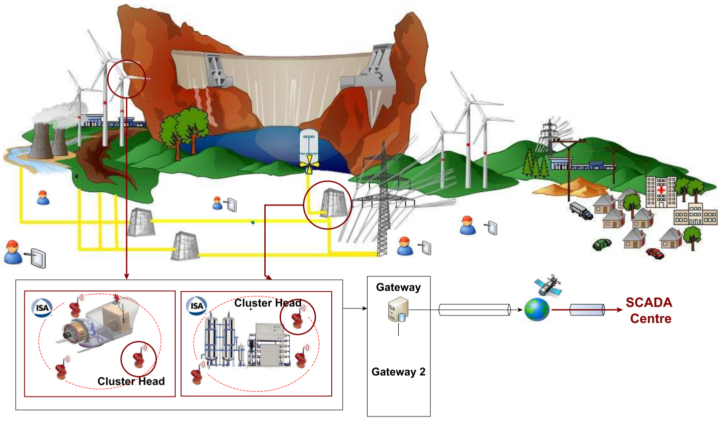

Fig. 1. General Architecture of the PDR mechanism

situations, we distribute IWSNs and MANETs throughout the entire system and close to its more sensitive parts, such as energy generators, motors, turbines, industrial engines, transformers, and others. In order to understand their functionalities in field, we describe in detail their particular characteristics and services below.

An IWSN [8] is composed of small and smart devices with the capability (4MHz-32MHz micro-processor, 8KB-128KB RAM, 128KB-192KB ROM) for sensing real states of an object or its surroundings. These states are associated with physical events in the context, such as temperature, pressure, voltage, vibration, etc. To measure these types of events, sensor nodes should be deployed close to the supervised target, for example, generators or motors of wind turbines (See Fig. 1). As conventional sensor nodes, industrial sensor nodes are also autonomous devices capable of processing and transmitting information to a base station. In our case, this base station is a powerful gateway device. Industrial sensor nodes can also offer services of auto-configuration, auto-organisation, self-monitoring and self-healing, detection, warning and tracking of anomalous behaviours or threatening situations such as peaks in voltage in electrical pylons or abrupt changes of temperature registered in industrial engines, as well as querying and reporting on-demand.

All of these features and services have encouraged both industry and government to modernise their CIs. Indeed, the industrial sector is aware of the advantages and opportunities of this technology to increase its levels of competitiveness, productivity and efficiency [8]. To the contrary, the government needs the technology to find a way to protect many of our CIs. According to the last report of the American Recovery and Reinvestment Act (ARPA) of 2009 [9], the U.S. already aims to invest in new information and communication systems in order to automate, for example, substations with smart sensors. The reasoning behind this investment is to find the way to avoid or mitigate disturbances and instabilities generated in remote locations.

Having commented this, we are not saying that IWSNs pretend to replace traditional wired industrial systems, such as RTUs. Instead, they try to offer a complementary tool for maximizing automation tasks and ensuring protection. As mentioned above, this protection includes all of the potential capabilities for prevention, detection and response against anomalous events of the system. An anomalous event is defined by the Oxford dictionary as "*something that deviates from what is standard, normal, or expected*" [10]. For our case, we identify two types of anomalies; *infrastructural anomalies* and *control anomalies*. The former type is related to the deviations associated with normal behaviour of the observed infrastructure, such as high/low voltage level, strong stress, high/low temperature, corrosion, gas/oil leaks, etc. In contrast, the latter type refers to the normal behaviour of the control network; i.e. the IWSN. Given that the control of our architecture is basically centralised on this kind of technology, the control anomalies could be de-synchronisation (deSync) in the communication channels, loss of information, or exhaustion of energy. Note that IWSNs are quite sensitive to these types of threats due to their mesh topology and their wireless-channels, in where harsh industrial conditions (e.g. vibration or noise) could break their links and cause unreliable communication.

Continuing with the architecture of PDR, it also includes MANET networks as a self-configuring technology of mobile devices connected by wireless links. This kind of communication enables human operators to locally manage the systems, allowing their mobility in field and collaboration with other human operators. Any information acquired from sensors can be visualised by their hand-held interfaces (e.g. a PDA). These interfaces facilitate the automation tasks by managing; (i) measurements, i.e. physical events, (ii) alarms with relevant data on real states from the observed infrastructure, or (iii) commands with a particular action. For communication from/to sensors, it is currently possible to apply wireless industrial communication protocols, such as ZigBee PRO [11], WirelessHART [12] or ISA100.11a. We focus our attention on the ISA100.11a standard for several reasons. First of all, it is an extended version of WirelessHART and was intentioned for industrial environments. Thus, it provides a set of useful services to address the coexistence with other technologies, communication reliability (e.g. use of hopping and blacklisting methods) and alarm management based on priorities. Second, it improves some of the security services of the ZigBee PRO, such as the key negotiation process in commissioning phase [6]. Another advantage of ISA100.11a is its flexibility for configuring wireless networks.

We believe that a good approach for our architecture is a hierarchical configuration; i.e. a network based on clusters of sensors. The reason lies in that this conformation of clusters does not only reduce computational costs in sensors, but it also facilitates a rapid location of a problem by knowing the sensor deployment and the affected area. For each cluster, a trustworthy sensor is selected, which is known as the Cluster Head (CH) with a unique ID_{CH}. This CH is responsible for (i) filtering and aggregating measurements, (ii) receiving alarms from its sensors, and (iii) resending any information to the gateway. Here, the gateway acts as a powerful interface between the acquisition world and the SCADA Center, with the capability for processing data and translating different types of messages. For reasons of simplicity, we assume that the communication link 'sensor-sensor' and 'sensor-gateway' are protected by using security services of ISA100.11a, and communication 'gateway-hand-held' and 'gateway-SCADA Center' are protected through security services of the TCP/IP standard and/or virtual private networks.

Table 1 summarises the advantages of building a proactive and reactive system based on IWSNs, MANETs, ISA100.11a and the Internet. Note that this table is based on the needs identified for WASA and on the studies done on WSNs, MANETs and the Internet in [5]. When combining technologies, different types of advantages are obtained, such as *monitoring*, *prevention*, *detection*, *alerting*, *response*, *collaboration* and *mobility*. Given this, the next step is to present the approach using mentioned technologies.

Technologies		Monitoring	Prevention	Detection	Alert	Response	Collaboration	Mobility
A	*IWSN & ISA100.11a*	$\surd - local$	$\surd$	$\surd$	$\surd$		$\surd - local$	
B	*MANETs*	$\surd - local$				$\surd - local$	$\surd - local$	$\surd$
C	*The Internet*	$\surd - remote$					$\surd - remote$	
D	*A & B & C*	$\surd - local/remote$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd - local/remote$	$\surd$

Table 1. Advantages of using IWSNs, MANETs, ISA100.11a and the Internet for WASA

3 PDR in Wide-Area Situational Awareness

As mentioned in Section 2, the sensor network follows a hierarchical configuration where CHs take on a special role within our approach. Each CH is configured with four main modules; a *Normalisation* module, a *Behaviour Pattern* module, a *Filtration-Aggregation* module and an *Alarm Manager* module (See Fig. 2). Any reading value of voltage, $volt_i$, from sensors must be normalised by the Normalisation module in order to format and standardise their contents. The normalised message is later processed by the behaviour Pattern module so as to identify normal or abnormal states. Normal states refer to those acceptable voltage reading values that are inside permitted thresholds, $[V_{min}, V_{max}]$. For these states, each CH has to (i) filter and aggregate the new value through the Filtration-Aggregation module, and (ii) send the aggregated values to the gateway. When the message is received by the gateway, it re-sends the message to the SCADA Center for supervision purposes, accountability or future analyses.

For unacceptable states ($volt_i \notin [V_{min}, V_{max}]$), it is essential to differentiate and classify different kind of states that could happen in our application context. One way to classify it would be to (i) consider the five levels of priority offered by the ISA100.11a standard [4], such as *normal, journal, low, medium, high* and *urgent* signalled with 0 to 5 respectively (such a value is denoted here as v_i), and (ii) define priority thresholds for each state. These thresholds not only depend on the security policies, but also the established policies for each country/organization.

Depending on the v_i and priority thresholds, the CH, through the Alarm Manager, will have to generate a particular type of alert with a specific label; *journal, low, medium, high* and *urgent*. The alert has to be sent to the gateway. For generation of the

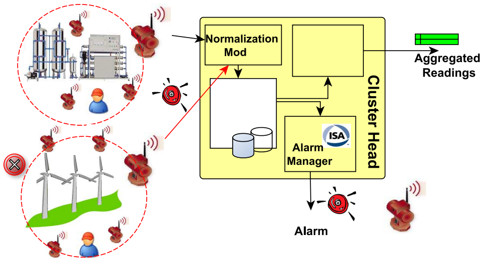

Fig. 2. Cluster Head: Dissemination and Warning

alert, the manager makes use of ISA100.11a objects. In particular, these objects come from the ARMO (Alert Reporting Management Object) class, and they have to be received and processed by a unique device in the system (in our case, the gateway), which contains the ARO (Alert Receiving Object) class of ISA100.11a.

When alarms arrive to the gateway, it is expected that the system responds to them properly and in a timely manner. For this reason, we deal with the prevention, detection and response in this section. Our intention is to anticipate infrastructural anomalies, detect suspicious behaviours in the control network and provide a rapid response to face incidents. These three activities will be configured inside the gateway using a set of interconnected modules (See Fig. 3). In particular, five main modules; an *ARO Manager* module, a *Prevention* module, a *Detection* module, a *Diagnostic* module, and an *Alarm Manager* module.

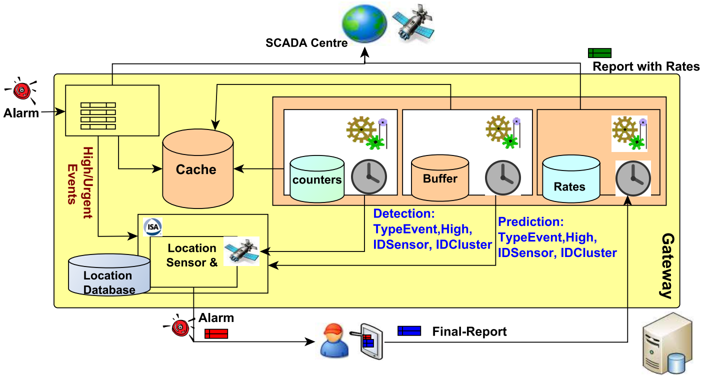

Fig. 3. Architecture of the Gateway: Incidents Management and Warning

Any incident from the control network has to be received by the ARO Manager. It is in charge of queuing incidents according to their priorities and handling critical alarms

[4-5]. These alarms have to be forwarded to the Alarm Manager Module so that it can locate the nearest staff in field immediately. For localisation of human operators, it is necessary to depend on the global positioning technologies and an updated database with information relative to deployment knowledge of sensors and human operators' availability according to their contracts. Both tasks are performed by the sub-module *Location Sensor & Operator*. Lastly, and continuing with the ARO Manager, any non-critical evidence ([0-5]) must be temporally stored in a cache memory for purposes of prevention, detection and response. Given that these three aspects are relevant topics within our approach, we will discuss them in-depth in the following sections.

3.1 Prevention of Infrastructural Anomalies through a Forecasting Model

Deviations in system attributes such as temperature or voltage levels are the main indicators of infrastructural anomalies. In this section, we propose a forecasting model to prevent these anomalies, particularly focusing on the voltage measurements. However, the model can easily be extended to other attributes as well.

As mentioned in previous sections, ISA100.11a classifies voltage measurement using six criticality levels ([0-5]). Receiving voltage measurements with level 4 or 5 requires the attention of the operators within a short time period. The forecasting algorithm may detect an anomaly before receiving a critical alarm (e.g. about 20 minutes ahead), and this enables the operator to have more time to resolve the problem.

The forecasting model aims to predict the occurrence of the critical alarms based on the assessment of previously received signals. To this end, the CH$_s$ collect the signals from sensors and send them to the gateway to be temporarily stored in a cache memory, which stores the voltage measurements received over the last $\Delta_{Twindow}$ minutes. Note that the Prediction module exports all the information with priority [0-3] from this cache to a internal buffer stored by ID$_{CHj}$ and ID$_{si}$. This buffer is applied for analyzing the behaviour the system in the following minutes. Exportation is done each time period (denoted here as $\Delta_{T diagnosticPrevention}$), the value of which is defined by the security policies.

Evaluation of the values in cache memory is done independently for each sensor. When the system is stable, we assumed that these measurements follow an independent discrete probability distribution with $Pr(v = i) = p_i$ for $i = 0, 1, \ldots, 5$, with $p_0, p_1, p_2, p_3 \geq 0$ and $p_4 = p_5 = 0$, where v is the voltage measurement level. It should be noted that the distribution of v should be estimated based on previous signals received when the system is in stable position. Moreover, the estimated distribution should be tested periodically, especially after making some infrastructural changes to the system. When there is an incident in the system, the distribution of v starts to deviate from the original distribution. The measurements tend to increase and eventually the sensors generate critical alarms, i.e. p_4 and p_5 are no longer zero.

Let $(v_1, v_2, \ldots, v_k)$ be the measurements corresponding to a particular sensor in the temporary cache memory that are received in the last $\Delta_{Twindow}$ minutes. The evaluation of the forecasting algorithm is based on the summation of v_i values, $S_k = v_1 + v_2 + \ldots + v_k$. Whenever the summation is greater than a threshold value, the algorithm sends an early warning alert for the corresponding sensor. The threshold value T is selected so

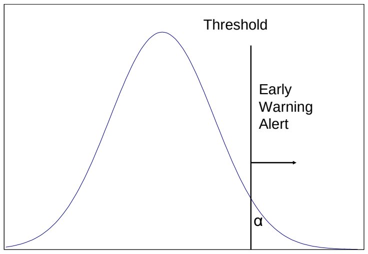

Fig. 4. Early warning alert and threshold value

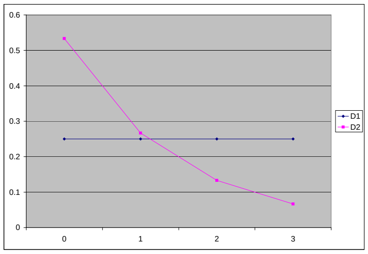

Fig. 5. The example distributions D_1 and D_2

that $Pr(S_k \geq T) \approx \alpha$, where α is an acceptable false alarm probability, i.e. the probability that the forecasting unit incorrectly outputs an early warning (See Fig. 4). Calculation of this probability requires the distribution of the S_k, which can be determined by induction, using the facts (i) the distribution of S_1 is equal to the distribution of v_i's, and (ii) the distribution of $S_k = S_{k-1} + v_k$. Table 2 provides threshold and false alarm probabilities for two example distributions, D_1 and D_2, where D_1 is the discrete uniform distribution i.e., $p_i = 0.25$ for $i = 0, \ldots, 3$ and D_2 satisfies $p_i \approx 2p_{i+1}$, for $i = 0, 1, 2$, i.e., $p_0 = 0.5335$, $p_1 = 0.2667$, $p_2 = 0.1333$ and $p_3 = 0.0665$ (See Fig. 5).

	α	T **for** D_1	T **for** D_2		α	T **for** D_1	T **for** D_2
$k = 5$	0.01	14	9	$k = 10$	0.01	15	23
	0.05	11	7		0.05	12	20
	0.10	10	6		0.10	11	19

Table 2. False alarm probabilities and corresponding threshold values for D_1 and D_2 depending on k previous values.

Taking into account the functions and variables defined in Table 3[1]. We describe the functionality of the Prediction module in a pseudo-code. The pseudo-code of the forecasting model is given below.

```
$systemTime = SystemTime();$
$nextDiagnostic = \Delta_{T diagnosticPrevention} + systemTime;$
WHILE ($ReceivingMessageFromARO()$) DO
    $message = NormalisedMessageCache();$
    $ID_{si} = IDSensor(message);$
    $ID_{CHj} = IDClusterHead(message);$
    $systemTime = SystemTime()$
    IF ($nextDiagnostic \leq systemTime$) THEN
        $StoreInBuffer(message, ID_{CHj}, ID_{si});$
        IF ($BufferIsComplete()$) THEN
            FOR ($j = 1; j \leq numCH_s; j++$) DO
                FOR ($i = 1, i \leq numSensorj; i++$) DO
                    $buffer = ObtainBuffer(j, i)$
                    $sum = CalculateSum(buffer);$
```

[1] It is important to mention that the rest of variables are defined throughout the text. In addition, part of these functions and variables are also used in Section 3.2.

Function	Description
$SystemTime()$	Obtain the real time of the system
$InitialiseAllCounters()$	Set all counters to 0
$ReceivingMessageFromARO()$ $NormalisedMessageCache()$	Read the cache memory while ARO is able to receive readings Obtain a message (the first) from the cache memory and normalise it
$IDSensor(message)/IDClusterHead(message)$	Obtain the IDs of sensor from the received message
$TimeReceived(message)/TimeStamp(message)$	Obtain values stored in message, such as the reception time of the message
$ExpectedTime(ID_{si})$	Obtain from the configuration of the sensor the reception time of a reading same node at the same time (using the cache)
$WasPreviouslySent(ID_{si}, timeStamp_{si}, message)$	Verify on the cache whether the message was previously sent by the same node at the same time
$IncreaseCounter(ID_{si}, Counter_{x})$	Increase the $Counter_x$ for the ID_{si}
$CounterX(ID_{si})$	Check the value of the $Counter_x$ for the ID_{si}, where X is Relay, DeSync, Lost
$WarningToAM(ID_{CHj}, ID_{si}, event, priority)$	Warn the Alarm Manager of a situation with a type of event and priority
$SensorIsNotInCache(ID_{si})$	Verify whether a sensor is active (alive) by checking its activity within the cache memory
$InitialiseCounterRelay\_DeSync\_Lost()$	Set the counters C_{relay}, C_{deSync} and $C_{lossInf}$ to 0
$StoreInBuffer(message, ID_{CHj}, ID_{si})$	Export all information with priority [0-3] from cache to the buffer
$BufferIsComplete()$	Verify whether the buffer is complete
$ObtainValuesSequence(ID_{CHj}, ID_{si})$	Obtain the sequence of values corresponding to the ID_{CHj} and ID_{si}
$CalculateSum(buffer)$	Calculate the sum of the values v_i
$Threshold(CHj, si)$	Calculate the threshold value depending on the buffer size of the CHj and si

Variable	Description
$systemTime$	Real time of the system
$nextDiagnostic$	Indicator of the frequency of the diagnostic
$buffer$	Temporal buffer of the Prediction module
$numCHs$ and $numSensorj$	Number of cluster heads and number of sensors in CH$_j$
sum	Total value of $\sum v_i$
$timeReceived$ and $timeStamp_{si}$	Time of reception of a message and its time-stamp
T_{time}	Estimate the period of time of reception of a message

Table 3. Description of Functions and Variables used in the Pseudo-codes of the Prediction and Detection modules

```
                    IF  $(sum > Threshold(SizeBuffer, CHj, si))$  THEN
                        $WarningToAM(ID_{CHj}, ID_{si}, \text{‘ prevention ’}, High)$
                    END
                END
            END
        END
        $nextDiagnostic = \Delta_{TdiagnosticPrevention} + systemTime;$
    END
END
```

3.2 Detection of Control Anomalies

As was mentioned above, the Detection module is in charge of detecting suspicious anomalies in the sensor network. These anomalies are related to relays, deSync and loss of sensitive information, as well as the presence of dead nodes. To control these threatening situations, we use four counters for each sensor; C_{relay}, C_{deSync}, $C_{lossInf}$ and $C_{deadNode}$. These counters should be frequently initialised when a given time for diagnosis, $\Delta_{TdiagnosticDetection}$, is attained.

For diagnostic, the Detection module needs to evaluate the time-stamp of each message received. If the time-stamp of a specific message is outside of an established maximum time for receiving messages (T$_{MAX}$), then the module may deduce that such a message was lost within the network, increasing the value of the counter $C_{lossInf}$. It is also possible that the time-stamp is within the required time, but a relay threat or a deSync threat are happening in the field. To detect a relay threat, a correlation process should be carried out so as to check evidence streams with information stored in the cache, the entries of which should be ordered by the time-stamp so as to speed up the process of search and correlation of values. In this way, if a specific sensor s$_i$ with ID$_{si}$ already sent a message with time-stamp$_{si}$ in the past, then the Detection module may infer that a relay attack is starting within the system, increasing its C_{relay}. Similarly, we require configuration information related to each sensor, such as the expected time to receive an evidence, to detect a deSync threat. If a sensor s$_i$ with ID$_{si}$ sends messages outside of said expected time, the Detection module increases the counter C_{deSync}. This also means that it is important to take into account the network configuration, as ISA100.11a offers the possibility of configuring the time division multiple access with specific a time-slot for the data link layer, in addition to providing a customizable hopping method for 16 channels. Note that two further situations may arise when a deSync threat is frequently produced within the network; (i) hardware or software problems, or (ii) the presence of a delay attack. A delay attack refers to forwarding information in a desynchronized manner in order to provoking delays in the reception of messages.

However, none of the previous measures control the presence of a dead node, which could be caused by a physical attack, energy exhaustion or a Denial of Service (DoS) attack. To this end, we use a diagnostic procedure, which is frequently executed when $\Delta_{TdiagnosticDetection}$ is reached. This procedure checks the cache memory in order to see whether a particular sensor s$_i$ with ID$_{si}$ temporally stopped its activity in field. If so, the Detection module has to update the counter $C_{deadNode}$. When the four counters exceed their respective prescribed thresholds, the Detection module will have to warn of the situation immediately. The notification must include, at least; CH$_j$-ID$_{CHj}$, s$_i$-ID$_{si}$, the type of event and the priority of the detected event. The events can range from ‘relay-threat’, ‘deSync-threat’, ‘lossInf-threat’ to ‘dead-node’. In order to understand the functionality of the Detection module, a pseudo-code is provided below (cf. the functions and variables defined in Table 3).

```
$systemTime = SystemTime();$
$nextDiagnostic = \Delta_{TdiagnosticDetection} + systemTime;$
$InitialiseAllCounters();$
WHILE $(ReceivingMessageFromARO())$ DO
    $message = NormalisedMessageCache();$
    $ID_{si} = IDSensor(message);$
    $ID_{CHj} = IDClusterHead(message);$
    $timeReceived = TimeReceived(message);$
    $timeStamp_{si} = TimeStamp(message);$
    $T_{time} = SystemTime() - timeStamp_{si};$
    IF $(T_{time} \leq Time_{MAX})$ THEN
        IF $(timeReceived \approx ExpectedTime(ID_{si}))$ THEN
            IF $(WasPreviouslySent(ID_{si}, timeStamp_{si}, message))$ THEN
                $IncreaseCounter(ID_{si}, C_{relay});$  $//Relay\ Threat$
                IF $(CounterRelay(ID_{si}) \geq Threshold_{relay})$ THEN
                    $WarningToAM(ID_{CHj}, ID_{si}, `relay-threat', High);$
                END
            END
        ELSE
            $IncreaseCounter(ID_{si}, C_{deSync});$  $//deSync\ Threat$
            IF $(CounterDeSync(ID_{si}) \geq Threshold_{deSync})$ THEN
                $WarningToAM(ID_{CHj}, ID_{si}, `deSync-threat', High);$
            END
        END
    ELSE
        $IncreaseCounter(ID_{si}, C_{lossInf});$  $//LostMessage$
    IF  $(CounterLost(ID_{si}) \geq Threshold_{lossInf})$  THEN
      $WarningToAM(ID_{CHj}, ID_{si}, `lossInf - threat', High);$
    END
  END
  $systemTime = SystemTime()$
  IF  $(nextDiagnostic \leq systemTime)$  THEN
    FOR  $(j = 1; j \leq numCH_s; j++)$  DO
      FOR  $(i = 1; i \leq numSensorj; i++)$  DO
        IF  $(SensorIsNotInCache(ID_{si}))$  THEN
          $IncreaseCounter(ID_{si}, C_{deadNode});$  $//Dead\ Node$
          IF  $(CounterDeDeadNode(ID_{si}) \geq Threshold_{deadNode})$  THEN
            $WarningToAM(ID_{CHj}, ID_{si}, `dead - node', High);$
          END
        END
      END
    END
    $nextDiagnostic = \Delta_{TdiagnosticDetection} + systemTime;$
    $InitialiseCounterRelay\_DeSync\_Lost();$
  END
END
```

3.3 Response to Anomalies and Evaluation

After the resolution of incidents in field, human operators should provide the system with enough feedback on the situation to be able to evaluate the level of precision (either of the prediction module or the detection module) (cf. Section 3.1 and Section 3.2). This feedback consists of three simple values; *good*, *bad* and *undetected*, and they have to be introduced through authorised hand-held interfaces and sent back to the gateway. When this feedback is received by the gateway, it has to be managed by the Diagnostic module to rate the final behaviour of the Prediction module and the Detection module. Given that we predict infrastructural anomalies and detect control anomalies, such feedback also has to include the type of resolution; i.e. an infrastructural issue or a control issue. With all of this information, the Diagnostic module has to compute a set of counters, which are declared as follows:

- Two counters of *True Positive* ($C_{PredictionTP}$ and $C_{DetectionTP}$): It refers to that a suspicious threat was properly predicted/detected by the system, and the human operator's feedback indicates a 'good' value.
- Two counters of *False Positive* ($C_{PredictionFP}$ and $C_{DetectionFP}$): It means that a suspicious threat was not correctly predicted/detected, and the human operator's feedback signals it as a 'bad' value.
- Two counters of *False Negative* ($C_{PredictionFN}$ and $C_{DetectionFN}$): It refers to that the human operator's feedback indicates the presence of an undetected critical situation (an 'undetected' value), and the approach was not able to detect it.
- Two counters of *True Negative* ($C_{PredictionTN}$ and $C_{DectionTN}$): An valid situation (e.g. $volt_i \in [V_{min}, V_{max}]$) happens within the system and it was properly classified by the system as innocuous.

Considering all these variables, the Diagnostic module has to find the way for evaluating the precision of our mechanism throughout its entire life-cycle. To this end, a set of metrics and measures of contingency described in [13] have been considered for our mechanism. These metrics consist of estimating the 'precision' by eventually computing the equations of Table 4.

Rate	Prevention	Detection
True Positive	$\frac{C_{PredictionTP}}{C_{PredictionTP}+C_{PredictionFP}}$	$\frac{C_{DetectionTP}}{C_{DetectionTP}+C_{DetectionFP}}$
False Positive	$\frac{C_{PredictionFP}}{C_{PreventionFP}+C_{PreventionTN}} \leq T_{PredictionFP}$	$\frac{C_{DetectionFP}}{C_{PreventionFP}+C_{PreventionTN}} \leq T_{DetectionFP}$
False Negative	$\frac{C_{PredictionFN}}{C_{PreventionFN}+C_{PreventionTP}} \leq T_{PredictionFN}$	$\frac{C_{DetectionFN}}{C_{PreventionFN}+C_{PreventionTP}} \leq T_{DetectionFN}$

Table 4. Precision Levels of the Prevention and Detection modules

Table 4 also shows us a set of thresholds, which should be defined to control the real level of precision of the modules. Note that the threshold for false negative rates should be much more restrictive with respect to the rest, since it is unacceptable that a control system is not able to predict/detect undesirable situations. Thus, when a false negative rate (either $C_{PredictionFN}$ or $C_{DectectionFN}$) is higher than its prescribed threshold, a report should be generated to warn the SCADA Center of the situation immediately. In this case, the organisation will have to analyse, for example, the possibility of extending the value of $\Delta_{Tdiagnostic}$ for detection or changing the probabilities of the transition between states for prediction. In contrast, a high false positive rate is not really a problem for critical environments given that this fact does not imply a loss of critical warnings.

It is worth stressing that the Prediction and Detection modules maintain a narrow relationship each other. If the Detection module is not able to detect a delay attack, it is possible that the Prediction module increases its $C_{PredictionFN}$, since critical alarms may be delayed. Similarly, if a relay attack appears within the network, the values sequence may change the value of $C_{PredictionFN}$ or the $C_{PredictionFP}$ by re-sending messages with priority [0-2] or [3], respectively.

4 Conclusion

In this paper we have modelled a preventive and reactive system based on four main types of technologies: IWSNs, MANETs, the Internet and the ISA100.11a standard. With this, we aim to show the capabilities of these technologies for prevention, detection and response in critical environments, and of course, cover some still pending challenges for WASA. As a result, the proposed system is able to warn of an emergency situation in advance, detect anomalous behaviours and respond against crisis situations in order to minimise security risks and avoid a cascading effect as far as possible. On the other hand, the solution, called here PDR, is also able to evaluate by itself the level of precision of its components of prevention and detection. This process will help the SCADA Center to maintain an exhaustive report corresponding to the functionality and reliability of the control service in the field and at any time.

Lastly, it is essential to continue advancing in this research area since there are a lot of open issues that need to be dealt with, such as security and connectivity problems when heterogeneous devices are being connected. For this reason, our next goal is to research how to connect sensors to the Internet [14] when gateways are not working, and how alarms and measurements can reach the SCADA Center in emergency situations in a secure manner. Likewise, it would also be interesting to provide location privacy mechanisms so as to protect the deployment of sensors and their visibility with respect to external threats [15].

References

1. C. Alcaraz and J. Lopez. Analysis of Requirements for Critical Control Systems. In *Sixth IFIP WG 11.10 International Conference on Critical Infrastructure Protection*, National Defense University, Washington DC (USA), 2012.
2. A. Atputharajah and T.K. Saha. Power System Blackouts - Literature Review. In *International Conference on Industrial and Information Systems (ICIIS)*, pages 460–465, 2009.
3. NIST. NIST Framework and Roadmap for Smart Grid Interoperability Standards, Release 2.0. NIST Special Publication 1108R2, February, 2012.
4. ANSI/ISA-99.02.01-2009 Standard. Security for Industrial Automation and Control Systems Part 2: Establishing an Industrial Automation and Control Systems Security Program, 2009.
5. C. Alcaraz, J. Lopez, J. Zhou, and R. Roman. Secure SCADA Framework for the Protection of Energy Control Systems. *Concurrency and Computation Practice & Experience*, 23(12):1414–1430, 2011.
6. C. Alcaraz and J. Lopez. A Security Analysis for Wireless Sensor Mesh Networks in Highly Critical Systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 40(4):419–428, 2010.
7. J. Peerenboom and R. Fisher. Analyzing Cross-Sector Interdependencies. *IEEE Computer Society, HICSS, IEEE Computer Society,*, pages 112–119, 2007.
8. V.C. Güngör, Bin Lu, and G.P. Hancke. Opportunities and Challenges of Wireless Sensor Networks in Smart Grid. *Industrial Electronics, IEEE Transactions on*, 57(10):3557 –3564, 2010.
9. The White House. Office of the Press Secretary, President Obama Announces $3.4 Billion Investment to Spur Transaction to Smart Energy Grid, October 2009.
10. Oxford Dictionary. Anomaly. `http://oxforddictionaries.com/definition/anomaly`, Retrieved on March 2012.
11. ZigBee Alliance. ZigBee PRO. `http://www.zigbee.org/`, Retrieved on March 2012.
12. HART. WirelessHART Technology. `http://www.hartcomm.org`, Retrieved on March 2012.
13. F. Salfner. *Event-based Failure Prediction An Extended Hidden Markov Model Approach*. PhD thesis, Humboldt-Universittzu Berlin, 2008.
14. J. Lopez C. Alcaraz, P. Najera and R. Roman. Wireless Sensor Networks and the Internet of Things: Do We Need a Complete Integration? In *First International Workshop on the Security of The Internet of Things (SecIoT 2010)*, Tokyo, Japan, 2010.
15. W. Zhu, Y. Xiang, J. Zhou, R. Deng, and F. Bao. Secure Localization with Attack Detection in Wireless Sensor Networks. *International Journal of Information Security*, 10:155–171, 2011.