Powering the Internet of Things With 5G Networks

Vasuky Mohanan
Universiti Sains Malaysia, Malaysia

Rahmat Budiarto
Albaha University, Saudi Arabia

Ismat Aldmour
Albaha University, Saudi Arabia

A volume in the Advances in Wireless Technologies and Telecommunication (AWTT) Book Series

Published in the United States of America by
IGI Global
Information Science Reference (an imprint of IGI Global)
701 E. Chocolate Avenue
Hershey PA, USA 17033
Tel: 717-533-8845
Fax: 717-533-8661
E-mail: cust@igi-global.com
Web site: http://www.igi-global.com

Library of Congress Cataloging-in-Publication Data

Names: Mohanan, Vasuky, 1970- editor. | Budiarto, Rahmat, 1961- editor. | Aldmour, Ismat, 1962- editor.
Title: Powering the internet of things with 5G networks / Vasuky Mohanan, Rahmat Budiarto, and Ismat Aldmour, editors.
Description: Hershey, PA : Information Science Reference, [2018] | Includes bibliographical references.
Identifiers: LCCN 2017010780| ISBN 9781522527992 (hardcover) | ISBN 9781522528005 (ebook)
Subjects: LCSH: Internet of things. | Mobile communication systems.
Classification: LCC TK5105.8857 .P69 2018 | DDC 004.67/8--dc23 LC record available at https://lccn.loc.gov/2017010780

This book is published in the IGI Global book series Advances in Wireless Technologies and Telecommunication (AWTT) (ISSN: 2327-3305; eISSN: 2327-3313)

British Cataloguing in Publication Data
A Cataloguing in Publication record for this book is available from the British Library.

Chapter 8
Smart Real-Time Internet-of-Things Network Monitoring System

Adil Fahad Alharthi
Albaha University, Saudi Arabia

Deris Stiawan
Universitas Sriwijaya, Indonesia

Mohammed Yahya Alzahrani
Albaha University, Saudi Arabia

Muhammad Fermi Pasha
Monash University Malaysia, Malaysia

Ismat Aldmour
Albaha University, Saudi Arabia

Rahmat Budiarto
Albaha University, Saudi Arabia

ABSTRACT

The network traffic of the Internet became huge and more complex due to the expansion of the Internet technology in supporting the convergence of IP networks, Internet of Things, and social networks. As a consequence, a more sophisticated network monitoring tool is desired in order to prevent an enterprise network from malware attacks, to maintain its availability as high as possible at any time, and to maintain the network's healthiness. This chapter offers a development of real-time network monitoring tool platform. The research component of this chapter attempts to answer the challenges of making the monitoring tool become smarter and more accurate by applying artificial intelligence techniques. In addition, a research on buffering techniques to speed up the traffic data acquisition process and micro-controller unit design for sensor-based applications are also carried out. In the development component, some ground works has already been done such as network traffic packets capturing modules, and packets decoding modules. The system development uses Java Eclipse platform.

DOI: 10.4018/978-1-5225-2799-2.ch008

INTRODUCTION

In today's world, not only computers are connected to the Internet, other devices such as smartphones, smart cars, sensors, home appliances, and so on, are also connected to the Internet. This, so called Internet of Things, makes the network traffics become more complex and vulnerable. Computer networks provide the shared resources, accounting, e-mail, Internet and Intranet that is used within organizations. It helps business to reduce cost, streamlines processes, and facilitates the sharing of information and the same time opens new vulnerabilities.

Most computer networks provide a lot of features that can be used to help the running of a business however, if a problem occurs within the network itself, the productivity of the company is severely affected. Therefore, it is important to find the cause of the problem as soon as possible. Such a task can normally be very tedious in a complex network.

Many commercial network monitoring tools and software are available today, vary from as simple as only monitoring segments of network up to systems with sophisticated capabilities such as visualization of nodes activities, IDS and intelligent engine to analyze the traffic as well as to predict requirements for future system development.

The traffic on the network may be generated by thousands of devices and thousands of software drivers and applications. Without the proper tools that can interpret, analyze and display network traffic and any related problems, a network administrator is limited to the time-consuming trial and error method to try to identify a problem. With a network analyzer application, such problems can be immediately detected and resolved. Nonetheless, a simple network analyzer application is no longer enough. To keep a network performs at top-notch condition, a network administrator needs a tool that has ability to

- Have intelligence built in.
- Even of tracking and resolving some of the problems on its own.
- Detect network viruses and provide the early warning needed.
- Point out the sources of the virus, and close it if possible.
- Provide intrusion detection and warning.
- Work in all IP platforms, including IPv4 and IPv6.
- Cross platform that support any Operating systems.
- Capture and monitor traffic from devices attached to Internet.

Autonomous intrusion agents, commonly referred to as `worms', are fast becoming a popular method of network and system compromise. The most famous start to the history of network worms is the Morris worm, which quickly crippled a substantial portion of the 1988 Internet. Worms have been a persistent security threat on the Internet, though for most of this history they focused on Windows hosts.

A real-time smart network monitoring and security platform will be implemented as a product named InstaMon.

InstaMon performs real time data collection of network traffic that flows on a local area network (LAN) segment and analyzes the data that is decoded, performs statistical calculation and displays the analyzed data. The objective is to create an intelligent tool to assist network and system administrators by anticipating and giving intelligent information for preventive measures to be taken so that damages as a result of system or network down time that can be very costly is minimized. Real-time network analysis helps to detect and resolve network faults and performance problems quickly. It even has the power to analyze multi-topology, multi-protocol networks—automatically.

BACKGROUND

With the additional traffic generated by the Internet of Things, computer networks traffic are growing at a drastic rate and thus network administrators can no longer monitor network problems by only relying on the traditional method such as Simple Network Management Protocol (SNMP) and Remote Monitoring (RMON). What network administrators need is the latest passive monitoring approach. Because as the number of hosts increase number of SNMP agents increase as well, which will result in a massive amount of traffic pumped to the network due to the fact that SNMP based tools are active in nature.

There are few of ways to monitor/protect network segments. InstaMon will initially tap into a network segment using passive capturing method (Matthews, Cottrell, & Salomoni, 1999). This will give the administrator access to traffic statistics of a network segment. Secondly for network devices with an internal RMON (Remote MONitoring) probe, the RMON Extension™ gives an access to the information gathered internally by the network devices. Finally for network devices with internal SNMP agents, the SNMP Extension™ gives an access to SNMP alerts (traps) and current status of the network devices. InstaMon uses the first method to monitor a network segment. The reason is that it is the most compatible method supported by all the switches.

Not all features used by the network analyzer to monitor a network are built into network devices.

The reasons are:

1. As for SNMP 1, it reports only whether devices are functioning properly. This can prove to be too vague and does not pinpoint the problem-causing device. Industry has attempted to define a new set of protocols called SNMP 2 that can provide additional information upon recognition of a problem. However, standardization efforts have not been successful.
2. Although, RMON can prove to be a very useful network-monitoring tool, it is still plagued with various problems besides high costs. Even with the introduction of RMON 2, a problem with incompatibility between vendor implementations has still not been alleviated. Because of this, RMON tool vendors have been adding propriety extensions to their products to make them more attractive to network managers who are demanding more functionality. There is still much risk of becoming dependent on a single vendor.
3. The passive approach, which is what InstaMon uses, does not increase the traffic on the network for the measurements, unlike SNMP and RMON where the polling required to collect the data and the traps and alarms (SNMP and RMON) all generate network traffic, which can be substantial. The general passive approach is extremely valuable in network trouble-shooting, however it is limited in certain ability to emulate error scenarios or isolating problems arising from other networks. Thus, in order to overcome these limitations, InstaMon uses an advanced passive approach. That is why an advance network traffic monitoring application such as InstaMon is needed to overcome the problem faced today.
4. With the Internet of Things coming into the picture, the current Internet traffic become more complex. More intelligent technique to detect anomaly in the network is needed. This chapter considers intelligent agent, evolving connectionist system, and computer forensics techniques to be incorporated into the Instamon in order to provide more accuracy in anomaly detection strategy.

INTERNET OF THINGS MONITORING SYSTEM

As technologies are changing and advancing rapidly, new technologies have emerged which were not anticipated previously. These new technologies are gaining popularity, and are predicted to create the de facto standards in their respective areas. These include Wi-Fi or WLAN (IEEE 802.11b) and Internet Protocol version 6- IPv6 (Lohith et al. 2011). As a potential commercial product, InstaMon must encompass all these technologies in order to be a successful product. InstaMon will be using a modular

architecture as a measure to allow it to modularly incorporate rapidly changing network technologies as shown in the diagram in Figure 1. This modularity would allow changes to be confined to an affected module instead of the whole product, minimizing the development cost and reducing the time to release the product.

Instamon consists of 4 main modules: Real-time network monitoring, Intelligent engine to detect anomaly, Network Security (Intrusion detection and network forensics), and Internet of Things Applications and monitoring. The overall structure of the project is shown in Figure 2.

Real-Time Network Monitoring

Packet Driver, a low-level capturing component for network monitoring application, which interacts directly with the Network Interface Card (NIC) running in the operating system kernel level, provides interface for user level application as shown in the diagram in Figure 3.

The packet driver is the major component of network monitoring application such as InstaMon, basically it provides high level Application Program Interface (API) for capturing wireless packet (A set of routines provides to network monitoring application such as InstaMon to direct the performance of packet capturing procedures by a computer's operating system). One disturbingly powerful aspect of packet drivers is their ability to place the hosting machine's network adapter into "promiscuous

Figure 1. Overall architecture of the proposed Instamon

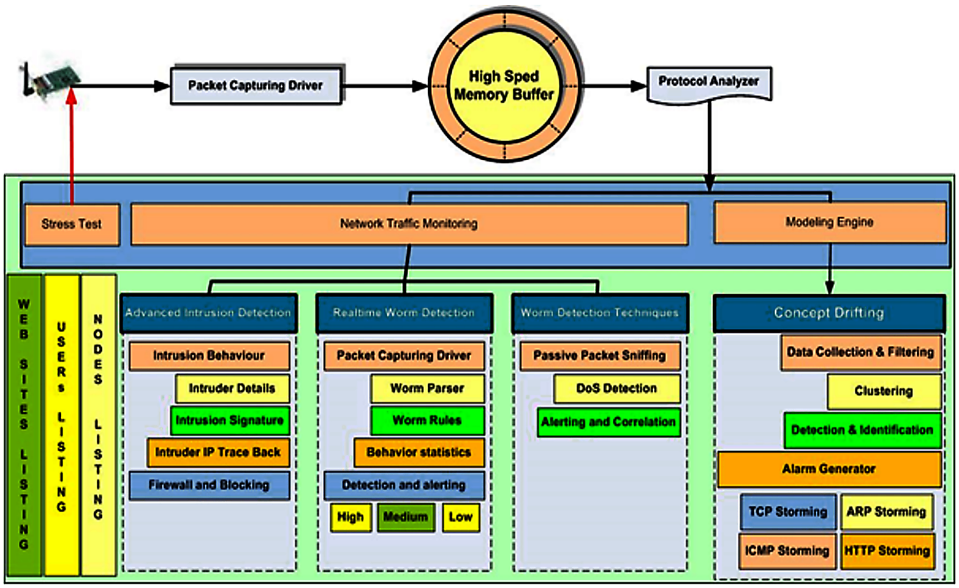


Figure 2. Development of InstaMon: a real-time smart network monitoring and security platform

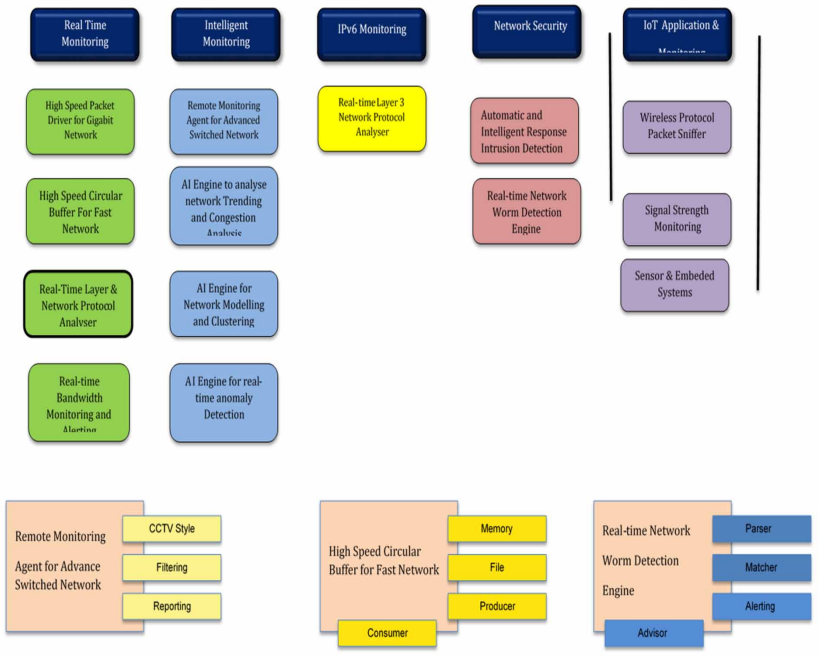
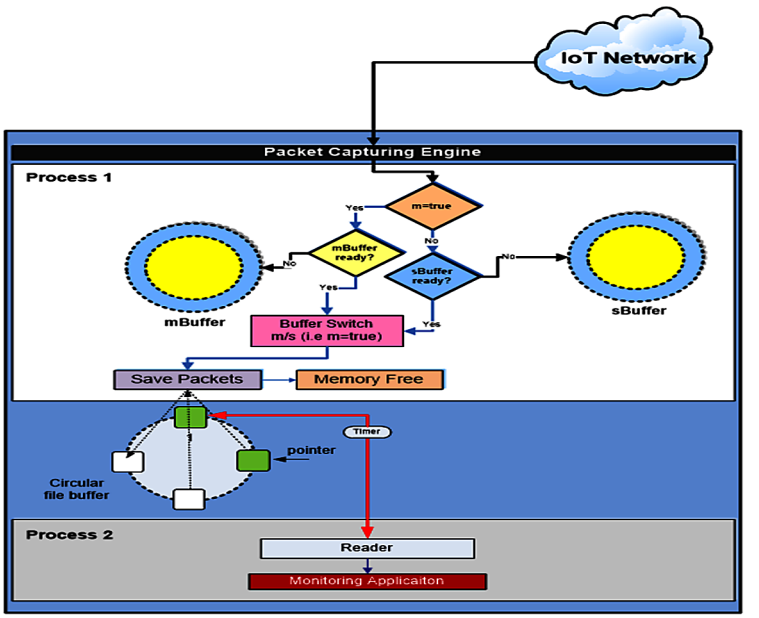


Figure 3. Packet drivers in Instamon



mode". Network adapters running in promiscuous mode receive not only the data directed to the machine hosting the sniffing software, but also all the traffic on the physically connected local network. It must provide high-speed packet capturing mechanism, which allows network-monitoring application to capture gigabits of information flowing in the computer network (3Com, 1990).

This module also focuses on the implementation of a circular buffer technology to enhance the capability of the capturing engine. A circular buffer is an efficient method of temporary storage allocation which entails the rotation of data through an array of buffer positions. In a circular buffer, the data writer advances one step every time new data is entered into the buffer. Once the end of the buffer is reached, this process is restarted once again from the beginning of the buffer. Data reading is done in the exact same manner. A circular buffer holds several advantages when compared to a conventional buffer. Firstly, it ensures approximately constant-time insertion and removal of data values. In addition, it also avoids the producer-consumer conundrum by enabling the packet analyzer to read up the packets from the circular file buffer in a smooth and efficient manner. This process is done concurrently with the insertion of data by the packet capturing engine. Careful calibration is done to ensure that the buffer writing process is done marginally faster than the packet analysis to avoid buffer overflow (Parameswar, 1996). All these will help to ensure a highly efficient capturing engine. Instamon is also designed to have capability to monitor IPv6 traffic.

Some concerns on the modules include: Network Driver Interface Specification (NDIS) version 5 whereby it is the de facto standard of interacting with NIC (Note: NDIS is a standard defined by Microsoft and 3Com), high-speed memory buffer to handle the large amount of data stream coming from the NIC, high speed packet filter to reduce the information passing from the NIC to computer's main memory, and the bottleneck of the I/O bus or Peripheral Component Interconnect (PCI) bus in the handling Gigabits of data (McCanne & Jacobson, 1992).

Intelligent Engine to Detect Anomaly

In years people have dream of one day, when computer handle all daily activities using intelligence as similar to human intelligent. The dream is realized by the arrival of Artificial Intelligent (AI). People are now applying AI technologies to the solution of difficult problems across a variety of application domains such as Network Trending and Congestion Analysis to achieve greater automation in network maintenances and supports. Basically, an Engine with AI technologies will tracks historical data on network traffic and graphs it. Automatically, compare current operations with an earlier benchmark, track improvement or deterioration

of performance over a period of months or years, and predict when additional bandwidth or server resources will be needed.

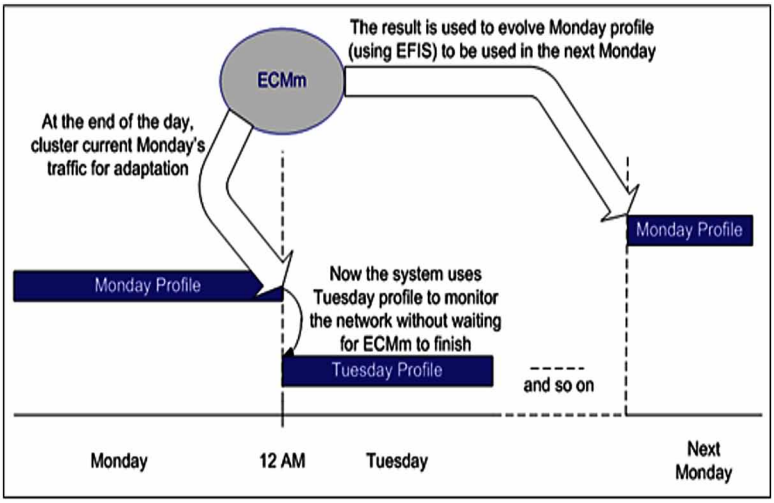
It will also detect problem conditions to a network automatically. Using trending data, it can give early warning of a possible problem. Automatically define triggers with each with many adjustments and sub-trigger settings, as well as an offset-based, user-defined trigger. Each trigger can have an associated action, which includes pop-up message windows, activating captures, starting/appending logs, executing external programs (for example, page or send email), etc. In short, AI engines apply the technology of AI to generate a network trend using recorded network traffic information such as network utilization, broadcast traffic percentage and protocol distribution. Using artificial intelligent again, current network trend is compare against history trend, and with the help of predefined probabilistic model it can alert the network administrator in the situation of network anomalies such as network congestion, broadcast abnormality.

Concerns to be considered include: neural network paradigms, knowledge-based systems, evolutionary algorithms, and optimization techniques in providing above mentioned feature; graphing and data recording technique to provide network trend presentation; software alarm for network error using AI techniques

The Evolving Clustering Method (ECM) is an online clustering method that performs well on one-pass partitioning of an input space. ECMc is a method to partition scarce input. ECMm is a clustering method for the purpose of clustering network traffic data streams. ECMm algorithm is a combination of ECM algorithm and its extension ECMc so that it can use the number of cluster created in previous process and optimize its cluster center in online mode. By having the advantages of ECM with a fast one-past online clustering and ECMc which optimize the cluster center, ECMm can perform online network traffic clustering with optimum results. There are 2 scenarios to show how ECMm work in clustering network traffic data input stream. The first scenario is the condition where ECMm is first-time run in the particular network and no profile has been created before. In this condition, ECMm works in one-pass clustering by creating and updating clusters in online mode. The ECMm is used to detect outliers if the current traffics collection is noisy and not attack-free. The second scenario is the condition where all the profiles have been created, where ECMm normally does after its first running.

ECMm is focused on component-level design and input-output data format. The ECMm algorithm is constructed into several components such as Objective function, Optimization function, Update cluster function, Main function, Create Cluster function. These components are representing some repeating process in the algorithm itself and each component is implemented as a separate function. The designs of these functions are following the theme of low coupling and high cohesion. Figure 4 shows the work flow of the ECM.

Figure 4. ECMm work flow



As for the input-output format, for the input, ECMm is fed with cumulative information every 5 minutes which contains the time when the traffic is captured. Total network traffic data (packets) at that time interval, and its total byte. This information is extracted from the collected network traffic data streams. As for the output, information about the cluster center and its radius, a matrix that maps the inputs index into the cluster where it belongs, the objective value and the input itself after being normalized is included. Figure 5 illustrates the clustering process and Figure 6 shows the snapshots of clustering results

Figure 5. Clustering process

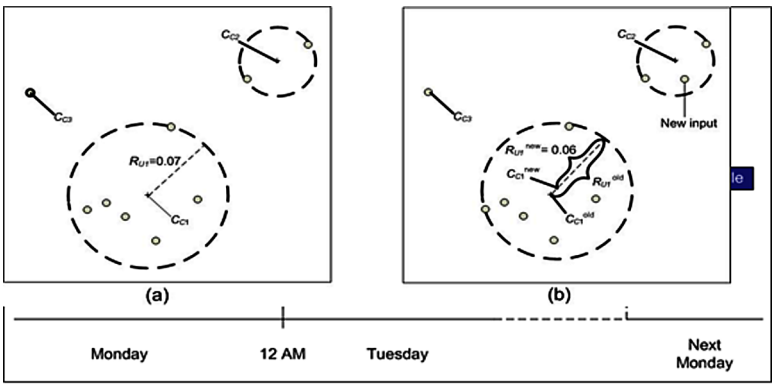
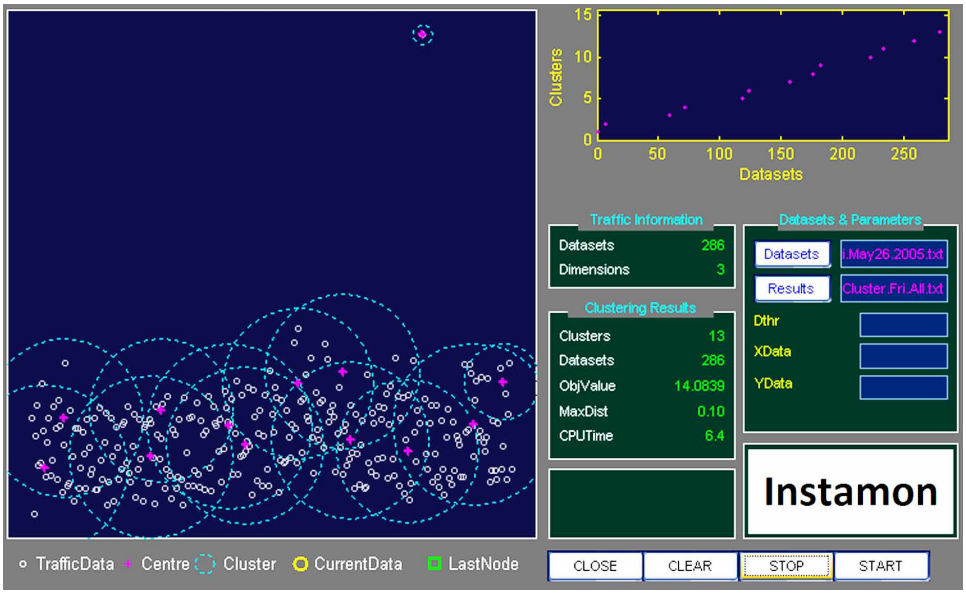


Figure 6. Snapshot of clustering results



IoT Network Security

Malina et all. (2016) states that an IoT security is the area of endeavor concerned with safeguarding connected devices and networks in the Internet of things (IoT). Much of the increase in IoT communication comes from computing devices and embedded sensor systems used in industrial machine-to-machine (M2M) communication, home and building automation, vehicle to vehicle communication and wearable computing devices.

The idea of networking appliances and other objects is relatively new, security has not always been considered in product design. IoT products are often sold with old and unpatched embedded operating systems and software. Furthermore, purchasers often fail to change the default passwords on smart devices -- or if they do change them, fail to select sufficiently strong passwords. To improve security, an IoT device that needs to be directly accessible over the Internet, should be segmented into its own network and have network access restricted. The network segment should then be monitored to identify potential anomalous traffic, and action should be taken if there is a problem (Tankard, 2015; Razzaque et al., 2016).

The main motivation behind this security module is the existence of a big number of attacks and viruses that severely affect the performance of computer networks. Such attacks include overloading the network with dummy packets which will then destroy the efficiency of the network. In general, network security systems have

some common objectives to fulfill. This chapter focuses on certain areas which are part of these common objectives. This section looks at the area of security policy which has to be adopted by any organization that are implementing and maintaining computer networks. Based on the security policy, an intrusion detection system that will perform the following tasks:

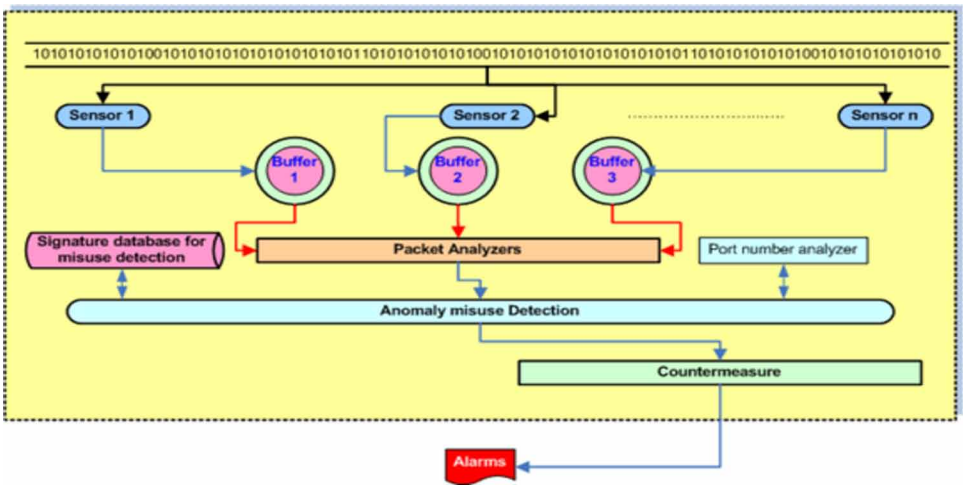
- Monitor and capturing the traffic on the network to be used later.
- Filter the traffic based on the source and destination IP addresses and the content of the packet (data).
- Detect attacks to the system using the Misuse and Anomaly Detection method.
- Employing neural network techniques to minimize false positive and maximizing the ability to learn from detected scenarios.
- Adopting fuzzy logic algorithm for abnormal activities to minimize the possibilities of intruders.

AI Engine for Real-Time Anomaly Detection

The explosion number of IoT applications also raises new issues, where various devices with multi-platform converge into one centralized, interconnected, shared, multi user, multi devices and flexible network. Research works by Microsoft (1991) and Stalling (1999) declare main issues on IoT. Growing number of internetwork network establish a heterogeneous network which is more complex than before. With various devices attached in the network, it will rise up technical problems in monitoring, managing, surveying and early detection of the network itself.

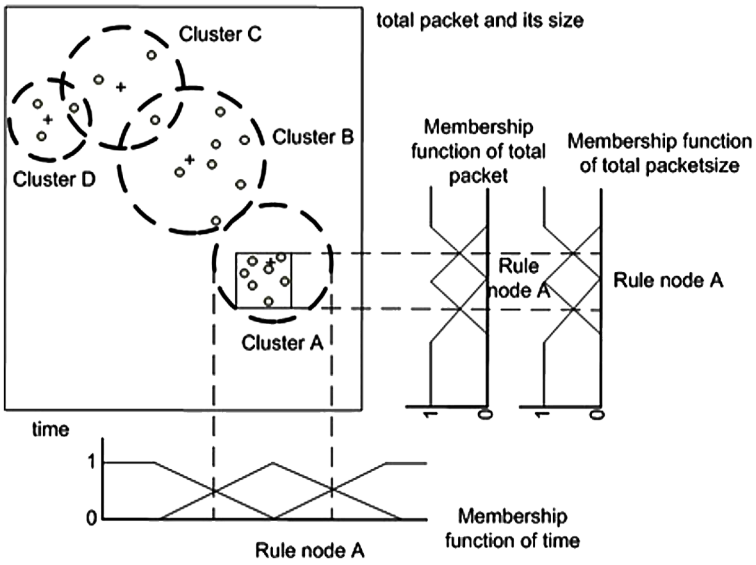
This sub-module utilizes EFIS: An Evolvable-Neural Based Fuzzy Inference System for Network Traffic Anomaly Detection. EFIS is an evolvable-neural based fuzzy inference system for profile creation and network traffic anomaly detection in online mode. It utilizes the evolving connectionist system framework which makes it able to evolve in open space to deal with concept drift problem and continuously monitor the network traffic to detect anomaly in online and lifelong mode. The idea of EFIS comes from a combination of features and structure of HyFIS and DENFIS model so that it is more suitable for network traffic data. The main feature of EFIS is that it is adaptable and the membership functions of the fuzzy predicates and the fuzzy rules can be adapted if necessary. EFIS structure has 2 main parts. The first one is the Profile Creation and Management (PCM) module which created and extracts rules from ECMm results, and the second one is the Neuro-Fuzzy Model (NFM) module which is a 5 layer neural network-based fuzzy system. Figure 7

Figure 7. Intrusion detection system model



depicts the intrusion detection system model. The Monitor process consists of Capturing, Buffering and Analyzing Network packets. The sensors monitor and capture every packet in the network. For the buffering, the packets will be saved in a circular buffer for fast processing and overcoming any packet lose and to enhance the system performance. Figure 8 illustrates the rule extraction process in the EFIS

Figure 8. Rule extraction process in EFIS



SNMP-Based Anomaly Detection

There are some solutions from a various IoT Infrastructure vendors with various standards that should be integrated. Unfortunately, due to the reason of incompatibility, not all of the things be able to adapt because they use their own proprietary technologies even though these technologies are claimed to be multiplatform support. These various technology also has encouraged the appearance of heterogeneous network information. Research works conducted conducted by (Zhenhui et al. (2014); Hyunho et al. (2014); Sakakibara et al. (2009)) mentioned that the heterogeneous IoT network must have services, with the following characteristics: (i) network transparency, (ii) transparency on the location of the service, (iii) transparency of data formats, and (iv) transparency of control protocols.

A problem related to the fact where each IoT device has a different Simple Network Management Protocol (SNMP) versions have been discussed in Yongqi et al. (2013) and Sanchez et al. (2013). The SNMP Protocol is used for capturing inbound-outbound packet load to a monitoring application. However, the existing monitoring applications only support monitoring in a single version of the SNMP protocol. An IoT network with monitoring and early anomaly detection system can prevent system failure which in turn will increase the reliability of the IoT network itself. Authors in Sanchez et al. (2013) and Tavares et al. (2014), proposed a network monitoring application with SNMP trap. The application is already informative but yet be able to perform the monitoring task if the SNMP protocol used is different versions and it merely focuses on network traffic. Besides, works by Aydin et al. (2009) and Wang et al. (2009) confirmed that the profiles of the system network activity (user, host, server and last mile connections) can be also as an indicator to any conditions occured in the network, such as: (i) utilization reaches 95% of the total traffic in a long period of time, which is typically only 40% in the peak time, (ii) increased use of memory continuously on the main Server, (iii) data access on a server outside normal hours, (iv) some devices attempt to connect and synchronize.

In the development of IoT monitoring system for detecting failures, the SNMP protocol is the main issues, where several heterogeneous network devices use different versions of SNMP which have different characteristics and features (SNMPv1, SNMPv2 and SNMPv3). That is the reason why messages format from the SNMP become the main attention. Extracting raw data from SNMP must be done to get the “Object Identifier” of the device.

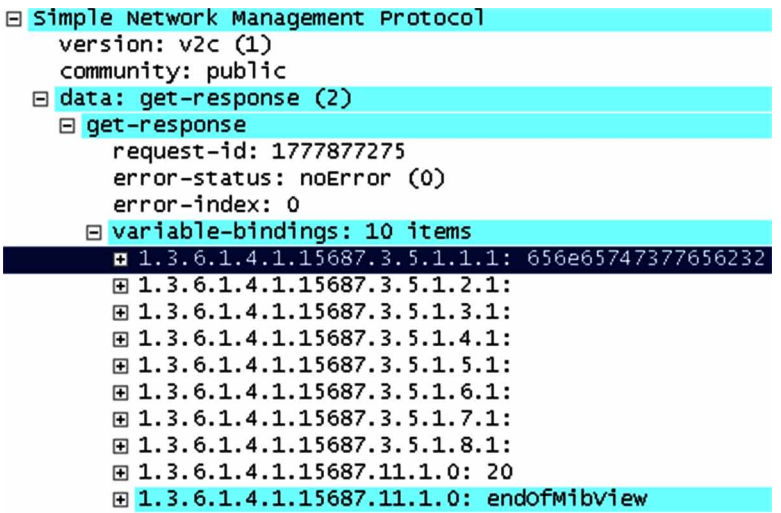
The design consists of two phases:

1. Deploying a network monitoring system for observing IoT network with heterogeneous devices,
2. Deploying early system for detecting errors based on device profiles and activities. In fact, there are several stages to design monitoring and detection system, including: data agent input, trapping the agent, storing that data into database, repeating on trapping the agent steps in case of data error and finally displaying data.

The first step is by paying attention to the monitoring of the process of "Get-Request" between managers and agents which getting some messages that occur in the process. The format of the messages from the "request and response" are: (1) Version, (2) Community name, (3) a Command (4) Request ID, (5) Error Status, (6) Error Index, and (7) value of the variable from the object.

This work refers to research works done previously by Yongqi et al. (2013) and Tavares t al. (2014) that focused on trapping traffic in SNMP and used the approaching method of SNMP messages format. SNMP has a Protocol Data Unit (PDU) as part of that message, and has five types of PDU: GetRequest PDU, GetNextRequest PDU, SetRequest PDU, GetResponse PDU and Trap PDU. Figure 9 shows example of the raw data packets that are successfully extracted from the traffic on the experimental network. SNMP is able to be installed in Raspberry "sysORDescr = STRING: The MIB module for SNMPv2 entities".

Figure 9. SNMP raw data



Experiment

A network environment is setup at our computer network Laboratory as illustrated in Figure. 10. The network consists of two components: network peripherals and devices, in order to represent an IoT network. The topology of the experimental network is set up in such a way, so an application system to monitor and to detect anomaly on multi-platform devices attached to an IoT network can be deployed.

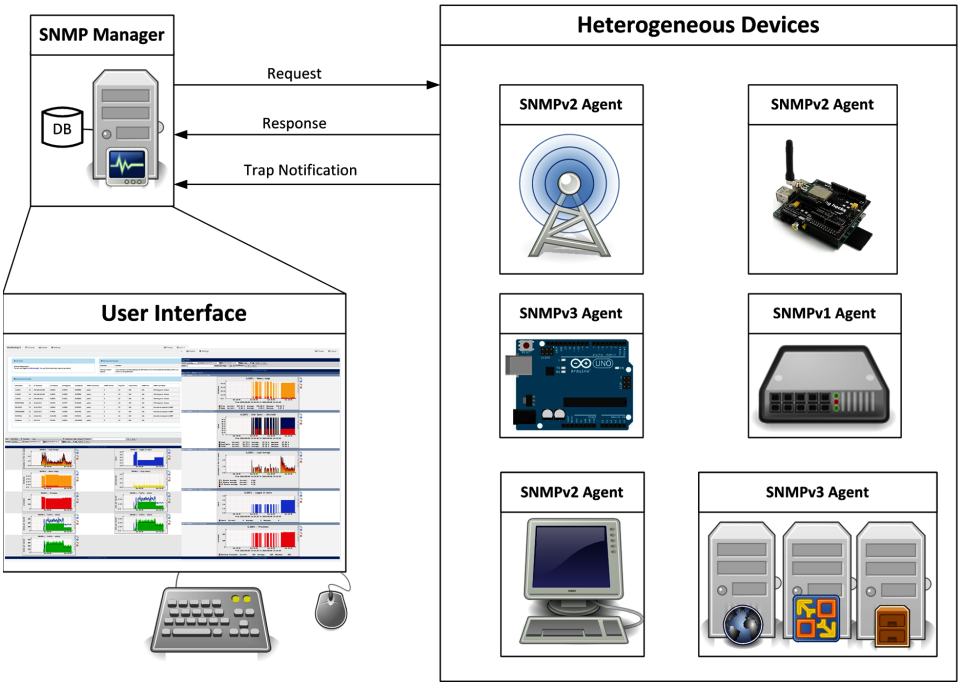
Device Specification and Configuration

The network pheriperals are: (1) two Cisco routers as a data packets forwarder at layer 3 (Network Layer), (2) two switches, and (3) one wireless access point router (WiFi 2.4 Ghz with AP-OpenWRT 802.11g) to provide WiFi service access from mobile devices.

The devices attached to the experimental network include:

1. Two servers to run multiple applications including Web server, database server, and other application servers with the following specifications: Intel Core

Figure 10. Experimental setup



2 duo, 4048MB RAM, 320 GB Storage, Operating system: Ubuntu Server 14.04.3 LTS 64bit.

2. Three PCs Workstation users which do the user profile with Windows 7 Operating System (OS): Intel Core 2 Quad Processor Q9500, 2048MB RAM DDR3, 500GB HD.
3. One server as network monitoring MIB host,
4. One cloud computing server for running virtual hosts running on Debian OS (Proxmox): Intel Xeon, 12048 MB, 1TB HD
5. Sensors to sensing room condition:
    a. Two Raspberry Pi: ARM Processor, Storage MMCard 8GB, 1 port Ethernet, Raspbian OS.
    b. Three Xbee S1 module: 3.3V @ 50mA, 250kbps Max data rate, 1mW output (+0dBm)
    c. Three M2303 Sensor and smoke sensor

Experiment Scenario

- Perform ping, tracer command and access to multiple servers
- Perform servers testing in running its daemon and enable by restarting it every time testing
- Measure traffic load to determine performance threshold values either in time or number of data packets
- Three PC users access to Web and application servers as well as cloud server. Set two times of treatment activities and performing (i) access separately to that three servers, (ii) access simultaneously, and (iii) random access with a specified time interval
- Memory and CPU usage will be used as measurement to enable a trap in the traffic by the agent,
- Running application of the monitoring system to receive data packages from any agent,
- Perform filtering to distinguish normal traffic from a failure, by separating and dividing the data traffic in several stages; based on time, target machine and used tools
- Pumping in data packet into the experimental network using Packet Generator (packgen) and real active users achieve a normal real world traffic
- Capturing raw traffic data using TCPdump to produce pcap files
- Enabling all services/daemons in target machines and restarting them before each test to ensure the same starting conditions

- Configuring sensors' option configuration LINUX-RASPI: Downed Device Detection SNMP Uptime, Timeout Value 400, Retry 1, SNMP Ver 3, Community public, Port 161, Timeout 10,
- Monitoring the network traffic using IPtraf and Collasoft Capsa.

Experimental Results

A network mapping matrix of interconnection traffic and summary data information are shown in Figure 11 and Figure 12. This traffic obtained during the experimental observation.

Figure 13 shows Raspberry device status from SNMP agent and displayed with graph. The system will inform real-time with elapse time of 5 seconds (adjustable) after the agent cannot response request from SNMP agent. Meanwhile, if the status is down, automatically will be updated in the system. Similarly, the graph will show up and down graphic to depict traffic activity from those devices.

The summary status visualization depicts for all device sensors that are most indicative of a pending failure, and the predictive strength of each devices.

Figure 14 shows a graph which describes about the traffic profiles of a user and CPU processing usage percentage in the experimental IoT network. The same graphic flow between the incoming and outgoing traffic of CPU/memory usage on

Figure 11. Overall traffic

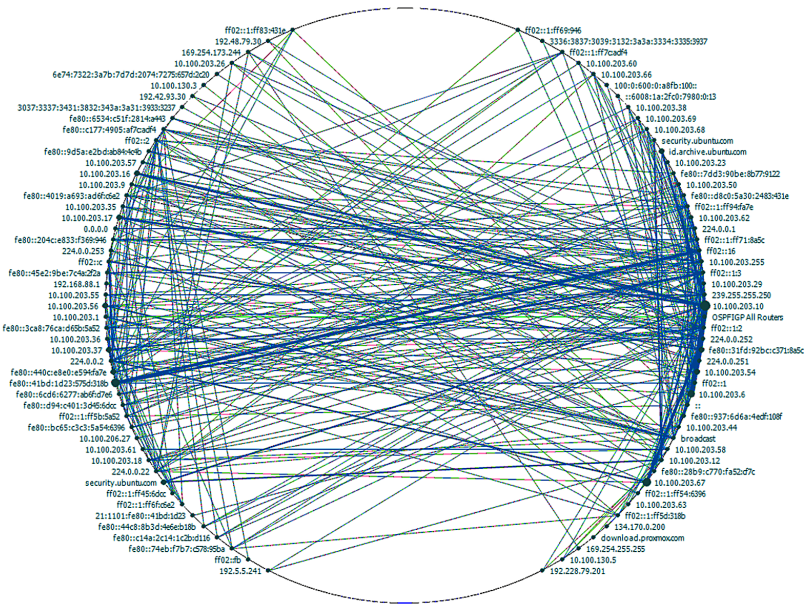


Figure 12. Overall protocol used

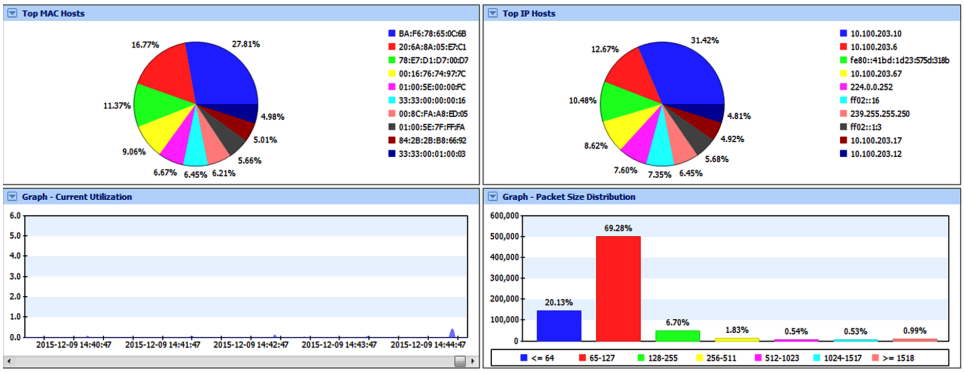


Figure 13. Device status vizualization

the monitored server are analyzed. Thus, it is possible to set a basic threshold value to trigger an allert in the notification system. For example, the network administrator may set the value of the threshold value network traffic as if “the traffic load < 500 Kbps or > 150 Mbps = usage processor > 45 percent” then the system will trigger an alert as an anomaly notification.

Network Forensics

What is unique about InstaMon as compared to other tools is that it does not only provide a centralised troubleshooting system: Centralised Monitoring Console, it also provides remote agent to monitor each remote segment and thin clients to protect each individual computer. This interesting and unique distributed architecture

Figure 14. Snapshot of traffic profile in a period of an observation time

creates a complete protected environment for the organisations servers, devices and the network itself. InstaMon is also a cross platform analysis and troubleshooting tool that means even if an organisation uses a mix of Windows and Linux based computers, devices and servers, Instamon would still be able to fully monitor and protect the network. The remote agent will constantly monitor the remote segment providing the CCTV Style to network administrators and enabling them to review the history at any time and at anywhere. For instance if administrator detected something went wrong at any specific time let say at 3 am, the network administrator will still be able to login and review what happened at that specific time and date.

FUTURE RESEARCH DIRECTIONS

The future of IoT and its impact on networking and network performance monitoring focus on concerns around IP addressing space and the need for Internet Service Providers (ISPs) to switch to Interent Protocol version 6 (IPv6) on a large scale. This is a perfectly valid concern – by some estimates, 30 billion devices will be online by 2020. This is the only main factor to consider when it comes to how IoT will impact the performance of networks around the world. The customer's expectations of network service is constantly changing as new technologies emerge that facilitate faster, more reliable connectivity. The following are some area of researches and development to enhance the capability of the future IoT network monitoring system to keep high availabitily of the network.

1. **Buffering System:** With the tremendous increase of traffic produced by billions of device connect to the network, it is a challange to avoid packet dropped by the packet capturing module. A fast and efficient buffering mechanism is one of the research opportunity to be carried out.
2. **Distributed Monitoring:** With the aims to not increasing the traffic with additional traffic to monitor, a distributed monitoring mechanism is recommended. The mechanism will keep local traffic in local probe and regulary will update the main/center monitoring system. The challenge are include: autonomousity, bottle-neck traffic in some segments, and synchronization.
3. **Layer-2 IDS:** Security is the main concern in WSNs especially in WSNs application designed for military and healthcare. Securing WSNs is a great challenge since broadcast nature of wireless commination, limited resources, unattended environment where sensor nodes vulnerable to physical attack (Abduvaliyev et al. (2013); Wang & Lu (2014)). Prevention countermeasures

like authentication, cryptography and other key management that known as first line of defiance can enhance the security of WSNs. Nevertheless, prevention solution cannot be stand alone to prevent all possible attacks Wang & Lu (2014). Thus Intrusion Detection System in Layer-2 (Data Link Layaer). that known as second line of defiance is extremely needed in order to detect attacks that pass prevention solutions (Abduvaliyev et al. (2013); Miranda et al. (2014)). Examples of attacks are given as follows.

a. **Collision:** When to node try to send out at identical frequency at the same time, the data portion will be changed due to packet collision. As consequence, packet will be rejected because of mismatch checksum (Buch (2011)).
b. **Exhaustion:** In this attack, adversary consumes all energy resources of the targeted node and disturbs the media access control protocol (MAC), by engaging the channel to send and receive unnecessary data. (Miranda et al. (2014); Khan (2014))
c. **Sybil Attack:** A malicious node in this attack spoofs the identity of other legitimate nodes (either MAC or IP). WSNs nodes work cooperatively and malicious node in this attack disturbs this cooperation. (Patel et al. (2013); Salehi et al. (2013))

4. **Adaptive System:** Due to the facts that new services and applications in IoT keep coming, the profile or the network traffic may change more frequently. The monitoring mechanism in the future also needs to be able to adapt to a new trend of the traffic quickly.

CONCLUSION

This chapter presents the development of smart real-time IoT network monitoring tool platform. The chapter is divided into 2 components: Research and Development. The research component attempts to answer the challenges of making the monitoring tool becomes smarter and more accurate by applying artificial intelligent techniques. The IoT carries a tsunami of data. IoT rollouts bring a proliferation of cheap, distributed sensors – resulting in a huge volume of data in a short amount of time. Thus, a research on buffering techniques to speed up the traffic data acquisition process and micro-controller unit design for sensor-based applications are introduced. The proposed system utilizes available existing elementary components such as network traffic packets capturing, packet filtering and packets decoding modules.

Malwares are becoming smart and sophisticated. Now, the same malwares may attack smart TV connected to Internet. This matter triggers the need of a system

that is capable to monitor in real time fashion to detect any anomalies in the network and to pinpoint the sources of the malwares. The proposed monitoring system uses intelligent techniques for clustering the network traffic to distinguish anomalies from normal traffics and machine learning for adaptively learning the network traffic changes and adapt accordingly. An insider attack is one of the biggest threats faced by modern enterprise networks. Thus, the proposed monitoring system is also acts as CCTV. The monitoring system uses Java Eclipse platform.

Experiments using an IoT testbed are conducted to reveal attacks/anomalies patterns.

REFERENCES

Abduvaliyev, A., Pathan, A.-S. K., Zhou, J., Roman, V., & Wong, W.-C. (2013, January). On the vital areas of intrusion detection systems in wireless sensor networks. *IEEE Communications Surveys and Tutorials*, *15*(3), 1223–1237. doi:10.1109/SURV.2012.121912.00006

Aydın, M. A., Zaim, A. H., & Ceylan, K. G. (2009). A hybrid intrusion detection system design for computer network security. *Computers & Electrical Engineering*, *35*(3), 517–526. doi:10.1016/j.compeleceng.2008.12.005

Buch, D., & Jinwala, D. (2011). Detection of wormhole attacks in wireless sensor network. *Proceedings of the 3rd International Conference on Advances in Recent Technologies in Communication and Computing (ARTCom 2011)*. doi:10.1049/ic.2011.0042

3. Corn/Microsoft LAN Manager, Network Driver Interface Specification (NDIS) Version 2.01 (FINAL). (1990).

Hyunho, P., Ho, L.-H., & Seung-Hwan, L. (2014). IEEE 802 standardization on heterogeneous network interworking. *Proceedings of the 16th International Conference on Advanced Communication Technology (ICACT)*, 1140-1145.

Khan, F. (2014). Secure communication and routing architecture in wireless sensor networks. *Proceedings of IEEE 3rd Global Conference on Consumer Electronics (GCCE)*. doi:10.1109/GCCE.2014.7031298

Lohith, Y. S., Brinda, M. C., Anand, S. V. R., & Hegde, M. (2011). 6PANVIEW: A Network Monitoring System for the Internet of Things. *Proceedings of the Asia-Pacific Advanced Newtork*, *32*(0), 106–109. doi:10.7125/APAN.32.13

Malina, L., Hajny, J., Fujdiak, R., & Hosek, J. (2016, June). On perspective of security and privacy-preserving solutions in the internet of things. *Computer Networks*, *102*, 83–95. doi:10.1016/j.comnet.2016.03.011

Matthews, W., Cottrell, L., & Salomoni, D. (2001, April). *Passive and Active Monitoring on a High Performance Research Network*. Academic Press.

McCanne, S., & Jacobson, V. (1992, December). *The BSD Packet Filter: A New Architecture for User-level Packet Capture*. Lawrence Berkeley Laboratory.

Miranda, J., Gomes, T., Abrishambaf, R., Loureiro, F., Mendes, J., Cabral, J., & Monteiro, J. (2014). A Wireless Sensor Network for collision detection on guardrails. *Proceedings of IEEE 23rd International Symposium on Industrial Electronics (ISIE)*. doi:10.1109/ISIE.2014.6864824

Parameswar, S. K., & Pooch, U. W. (1996). *Universal Packet Analyser - A Network Packet Filtering tool.* Department of Computer Science, Texas A&M University, Technical Report 96-008 (TR 96-008).

Patel, V., Taghavi, M., Bakhtiyari, K., & Júnior, J. C. (2013). An intrusion detection and prevention system in cloud computing: A systematic review. *Journal of Network and Computer Applications*, *36*(1), 25–41. doi:10.1016/j.jnca.2012.08.007

Razzaque, M. A., Milojevic-Jevric, M., Palade, A., & Clarke, S. (2016). Middleware for Internet of Things: A Survey. *IEEE Internet of Things Journal*, *3*(1), 70–95. doi:10.1109/JIOT.2015.2498900

Sakakibara, H., Nakazawa, J., & Tokuda, H. (2009). PBN: A seamless network infrastructure of heterogeneous network nodes. *Proceedings of the Sixth International Conference on Networked Sensing Systems (INSS)*. doi:10.1109/INSS.2009.5409912

Salehi, S. A., Razzaque, M., Naraei, P., & Farrokhtala, A. (2013). Detection of sinkhole attack in wireless sensor networks. *Proceedings of IEEE International Conference on Space Science and Communication (IconSpace)*. doi:10.1109/IconSpace.2013.6599496

Sánchez, R., Herrero, Á., & Corchado, E. (2013, October). Visualization and clustering for SNMP intrusion detection. *Cybernetics and Systems*, *44*(6-7), 505–532. doi:10.1080/01969722.2013.803903

Stalling, W. (1999). SNMP, SNMPv2, SNMPv3 and RMON 1 and 2 (3rd ed.). Addison Wesley Longman.

Tankard, C. (2015, September). The security issues of the Internet of Things. *Computer Fraud & Security*, *2015*(9), 11–14. doi:10.1016/S1361-3723(15)30084-1

Tavares Guimaraes, V., Lessa dos Santos, G., da Cunha Rodrigues, G., Zambenedetti Granville, L., & Rockenbach Tarouco, L. M. (2014). A collaborative solution for SNMP traces visualization. *Proceedings of International Conference on, 2014, International Conference on Information Networking (ICOIN)*, 458-463. doi:10.1109/ICOIN.2014.6799724

Wang, C. Y., Chou, S.-T., & Chang, H.-C. (2009. Emotion and motivation: understanding user behavior of Web 2.0 Application. *Proceedings of IEEE Computer Society Seventh Annual Communication Networks and Services Research Conference*, 1341-1346. doi:10.1109/ITNG.2009.205

Wang, L., & Lu, F. (2014). Intrusion detection system based on integration of neural network for wireless sensor network. *Journal of Software Engineering*, *8*(4), 225–238. doi:10.3923/jse.2014.225.238

Wong, E. (1997, August). *Network Monitoring Fundamentals and Standards*. Academic Press.

Yongqi, H., Yun, Z., Taihao, L., & Liying, C. (2013). Research of network monitoring based on SNMP. *Proceedings of Third International Conference on Instrumentation, Measurement, Computer, Communication and Control (IMCCC)*, 411-414.

Zhenhui, Y., Keeney, J., Van Der Meer, S., Hogan, G., & Muntean, G. M. (2014). Context-aware heterogeneous network performance analysis: Test-bed development. *Proceedings of IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*, 472-477.

KEY TERMS AND DEFINITIONS

Anomaly Traffic: A deviation from the normal traffic pattern for example a flood of UDP packets or a new service appearing on the network.

Intrusion Detection System: A system (usually in the form of device or software application) that monitors a network for malicious activity or policy violations. The system reports the detected anomaly or violation to system administrator or centrally collected using a security information and event management (SIEM) system.

Network Traffic: The amount of data moving across a network at a given point of time. Network data is mostly encapsulated in network packets, which provide the load in the network. Network traffic is the main component for network traffic measurement.

Traffic Visualization System: A system to display the captured network traffic, so system administrator can view current situation in the network.