VizNet – Dynamic Visualization of Networks and Internet of Things

Zoran Constantinescu, Monica Vlădoiu, Gabriela Moise
Department of Computer Science and Information Technology
UPG University of Ploiesti
Ploiesti, Romania
{zoran, mvladoiu, gmoise@upg-ploiesti.ro}

***Abstract* — Both the increasing complexity of networked devices and the sheer number of IoT devices raise the problem of proactively knowing when something goes wrong in their normal functioning. How do we know when there is a failure or a malfunction in the network? How do we know if the network is overloaded and how can we evaluate the infrastructure's performance? How do we pinpoint the cause of the problems? How do we organize and present this information to the user in a simple and intuitive way? We introduce here our prototype 3D visualization system for real-time monitoring of both the status of networked devices (wired, wireless, IoT devices) and the network's dynamics ("pulse") (e.g. configuration, load, traffic, abnormal events, suspicious connections, failed IoT devices, etc.). The user can visualize the current status of the networks of interest, in a very simple and intuitive way, from anywhere on the Internet (even from a mobile device). He can also receive alerts whenever something significant happens in the network by means of short text or instant messages.**

Keywords — network monitoring and visualization, network topology discovery, network status and dynamics, IoT devices

I. INTRODUCTION

Development of communications and networking has raised new problems with respect to network administration and security. Network issues are more and more complicated due to growing network complexity (size, structure, using different technologies, as more often the networks are hybrid with different protocols having to work together, and so on). Network complexity nowadays is proven also by the large range of existing routing algorithms and techniques, i.e. shortest path algorithm, flooding, distance vector routing, link state routing, hierarchical routing, broadcast, multicast, anycast routing, routing in ad-hoc networks, etc. [1]. The threats and vulnerabilities that networks have to handle nowadays are also on an increasing trend with respect to both their sheer number and diversity. The Internet of Things (IoT) paradigm adds up to this complexity by its specific issues (a very large variety of smart sensors, devices, and objects, different embedded technologies and communication protocols, bidirectional interaction, etc.) [1].

A large variety of network threats makes possible numerous attack scenarios in a network, such as traffic intercepting and spying, sniffing and scanning network, social engineering, phishing attacks, piggybacking, tailgating, superzapping, scavenging attacks, Trojan horses, all kinds of viruses, trap doors attacks, spyware, adware, etc. [3]. The new generation of 5G wireless technology will provide for a totally mobile and connected society that benefits from multiple gigabits per/second speeds and latencies under 1ms [4][5]. Moreover, handling upcoming ad-hoc networks (Mobile Ad hoc Network - MANET, Vehicular Networks - VANET) raises extra challenges in networking due to their particular characteristics (dynamics, security, limited bandwidth, limited energy resources, special security threats, etc.) [6][7][8].

All these aspects will increase further the complexity of network administration and security assurance. Therefore, there is an increasing need for tools that allow network visualization and monitoring and that are easy to use for solving both typical and extraordinary issues that are encountered daily in such multifaceted networks.

We introduce here our early results in the development of **VizNet**, a 3D visualization system for *monitoring real-time status of networked devices*. These devices can be of different types and from different producers, such as network equipment (routers, switches, gateways), wireless access points (APs), wired devices (PCs, servers, etc.), wireless devices (laptops, mobiles), and IoT devices. The system will also capture the dynamics or "pulse" of the network (i.e. current configuration, load, traffic, events, suspicious connections, failed devices, etc.). The user can visualize, in a very simple and intuitive way, the current status of the networks of interest from anywhere on the Internet, by using a simple web browser from a desktop, a laptop, a tablet, or even a smart phone. The visualization is shown as a dynamic, animated 3D virtual environment. In addition, the user can receive alerts whenever something significant happens in the network by means of short messages, as text messages over the phone, or as text and image messages using instant messaging (for example, Facebook, telegram, or similar systems). The main goals of this paper are as follows: to describe briefly the VizNet system and its advantages when compared to related work, to provide the reader with an overview of both its capabilities and architecture, and to present some aspects of the 3D visualization and alerts messaging interface, along with some real life use examples. However, the description of VizNet is far from exhaustive, many implementation details being left out.

The structure of this paper is as follows: the next section includes the related work. In Section III we present the

architecture of the proposed visualization and monitoring system, focusing on its main components. Section IV presents the VizNet interface, both the 3D visualization and the alerts messaging. Throughout Section III and Section IV we point out the current status of the implementation that is also illustrated with some examples. The final section is dedicated to both conclusions and ideas for further development.

II. Related Work

All the complex network issues (infrastructure, architecture, security, dimension, and dynamics) have lead to an increased necessity for development of tools for network analysis and visualization (that are usually called *monitoring and visualization network tools*). The monitoring tools generate a huge amount of data about the network and the traffic within, while the visualization tools help the entitled users to make sense of this data, to draw some conclusions, and to act upon them. Such tools are useful for network management, monitoring, and security ensuring, because visualization helps explaining how route analytics works and how it can be used to enhance network monitoring and troubleshooting by adding real-time visibility into routing operations, as well as network-wide traffic flows, leading to operations, engineering, and business costs savings [9].

These monitoring and visualization tools are based on different protocols presented beneath:

- The Simple Network Management Protocol (SNMP) offers both a standard framework and a common language for administration and monitoring of network devices using the TCP/IP protocol suite. Despite SNMPv1's large use, SNMPv3 is preferable due to the advanced security characteristics it provides [1];
- The Internet Control Message Protocol (ICMP) is specified in RFC 792/4443 (for IPv4/v6) with later updates. ICMP is used by hosts and routers to send each other control data, typically error messages [1];
- NetFlow is a protocol for collecting and monitoring data flow in networks, such as IP flow information. Network elements (routers and switches) gather flow data that provides fine-grained metering for highly flexible and detailed resource usage accounting [10];
- The Internet Protocol Flow Information Export (IPFIX) defines how the IP flow information can be exported from routers and other network elements [10]. The IPFIX Collecting Process allows receiving the flow information passing through multiple network elements [11].

Several monitoring and visualization tools that are largely used and/or include innovative approaches are presented further on.

Based on the retrieval of the network traffic flow information from network devices (NetFlow and SNMP) and from local traffic network (by packet sniffer- Host-bed/Local traffic flow information), there are eight main tool groups: Network Monitoring Platforms (NMP), Monitoring Tools Integrated with NMP, Commercial Monitoring Tools not Integrated with an NMP, Public Domain Network Monitoring Tools, Web Tools, Auxiliary Tools to Enable Monitoring, Analysis, and Report Creation or Simulation [13]. For commercial network monitoring tools, there are eight subgroups: Analyzer/Sniffer, Application/Services monitoring, Flow monitoring, FTP, Network security, SNMP tools, Topology, and VOIP (Voice Over IP) [13]. Also, for public network monitoring tools, there are fourteen subgroups: Application Monitoring, Finger Printing, FTP (File Transfer Protocol), Mapping, Monitoring Infrastructures, Packet Capture, Path Characterization, Ping, RRDtool (Round Robin Database Tool), SNMP, Throughput tools, Traceroute [13].

A network monitoring tool that uses a Traffic Dispersion Graph (TDG) to monitor, analyze, and visualize the host interactions is presented in [14]. The authors propose TDGs as a different way of modeling traffic behavior that, in their opinion, have characteristic structure and provide visualizations that can distinguish the nature of some applications.

One popular tool in the open source community is Netflow Sensor (NfSen), a graphical Web-based front end for the nfdump netflow tools. NfSen allows displaying, navigating, and processing the netflow data (flows, packets, and bytes), creating history and continuous profiles, setting alerts based on various conditions, and writing specific plugins for processing netflow data regular intervals [15]. The most used free monitoring and visualization tools are as follows: Cacti, Nagios, Icinga, NeDi, Ntop, Zabbix, Observium, Microsoft Network Monitor, OpenNMS, Advanced IP Scanner, Fiddler, NetworkMiner, Pandora FMS, Zenoss Core, PRTG Network Monitor Freeware, etc. [16][24].

Commercial systems such as IBM Aurora1, NetQoS Reporter Analyzer, Caligare Flow Inspector, and Arbor Peak-flow often include methods for intrusion detection that provides for examination of generated alerts through interactive reports. However, the used statistical charts and diagrams only scale to a limited number of alerts or highly aggregated information [17]. One particular tool to be mentioned is ANEMOS (Autonomous Network MOnitoring System), which is able to evaluate user-specified rules on the collected data (in real time or in a batch process) and to issue alarms when the rule conditions are triggered [18].

Development of monitoring and visualization tools has shown lately a new trend, i.e. the integration of innovative visualization tools that make it easy for users to understands what is going on inside the networks they manage [20]. For example, The Spinning Cube of Potential Doom (SCPD) constantly scans any computer connected to the Internet to determine any potential security vulnerability and attacks [21]. SCPD has been developed to make the participants at the 2003 Supercomputing Conference aware of the treats in the conference network [22]. Within the CyberNet project, research on the usability and effectiveness of 3D techniques and virtual reality interfaces for system and network management has been performed [19]. CyberNet is a distributed object framework for network administration that provides for handling of a large variety of tasks, ranging from collecting the network data to 3D visualization. The authors

have benefited from using of distributed object technology to tackle some important issues: the facts that the data is distributed and dynamic and the separation of the computationally intensive GUI from other parts of the system.

A very comprehensive survey, based on a systematic literature review on information visualization and network and service management, on a very large time period (1980 to 2013), is available in [23]. This overview emphasize a set of challenges of network and IoT visualization, such as:

- new approaches that included dynamic, positioning, and linked filters, 3D layout, hierarchy of 3D surfaces, support for queries, 3D virtual worlds, geographic views, and visualization for security;
- new tools for visualization of huge networks, in which Internet graph includes, aside from the management and service network functions, performance measuring, anomaly and intruder detection;
- integrated views on the network resources that support administrators in their decision making;
- AI techniques, as a promising approach of intruder identification and of attack visualization, which can help increasing network security;
- IoT devices' dynamic and monitoring, trust management, IoT privacy and security, Cloud Computing; Big Data, software defined networking and human-centered evaluation [20].

As we can see from the related work, there are many tools available for network monitoring and visualization. However, most of them are focused only on some aspects and challenges presented above. Below we summarize what differentiate our approach from the related work and its main advantages.

With VizNet, we propose a solution for *user-friendly* monitoring and visualization of what happens in a network, our final goal being to present the user with the "pulse" of a network, *in real-time*, by means of an easy to use 3D dynamic visualization tool. However, the visualization is just the frontend of the system, the backend part being the network monitoring, data storage, and alerting. Our design approach is *modular* and *customizable* (with different visualization interfaces available - rich 3D virtual environments or simpler, 2D interfaces). An important capability is *IoT monitoring*, given the complexity and constraints of such networks. The VizNet system can be used as a *research tool* that helps its users to analyze and understand networks and to *make sense* of what happens within, and also as an *educational support tool* for those who learn about networks and IoT.

III. VizNet Architecture

The VizNet system is a distributed framework for network monitoring and visualization that handles all the tasks necessary to collect and persistently store both network data and information about network devices and their status, to discover new devices, to keep an up to date connectivity status, to monitor network and device events, and to present visualization data.

The VizNet architecture includes components dedicated to device discovery, device connectivity discovery, device monitoring, and visualization and alerting modules (Fig.1).

One of the main components of VizNet is dedicated to *topology discovery*. It keeps a database of the existing network devices and their interconnectivity. These can be any type of network devices, IoT devices, wired or wireless, with permanent or temporary network connections. Another component is dedicated to the *devices' status monitoring* and it keeps track of each device's status by storing this information in a database for later analysis. The system can also send different alerts to users by using the *user notification* component. The *visualization interface*, which allows users to view in real-time different aspects of the monitored network, is also an important component of VizNet that will be presented in further detail in the next section.

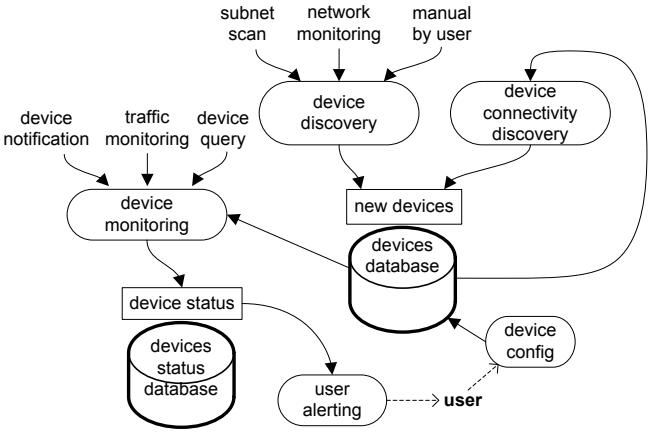

Fig. 1. Architecture of the VizNet System

A. Topology Discovery

The network topology discovery component is responsible for detection of both the network devices and their interconnections. To accomplish its goals it uses two modules. The first module, *device discovery*, is able to detect new devices in the network and also to collect information about them. Different techniques can be used for discovering new modules:

- *network scanning* – given a number of subnets, the module will scan all the possible IP addresses in each subnet and try to detect each device's type; this approach works when the devices are on-line and respond to network requests (ping, icmp, etc.);
- *network monitoring* – by sniffing network traffic on certain network interfaces, new devices can be detected as well; this is possible for devices which have an IP address and that are using IPv4 or IPv6 based protocols (network devices, IoT devices using wireless 6LoWPAN protocols, etc.);
- *manual* – the user can add a set of IP addresses associated to different network devices; the module will scan these addresses and will try to detect each device's type; if not, the user can add detailed information;

- *special device query* – in case of wireless devices which connect to the Internet via a gateway (access points, IoT gateways, LoRa gateways and concentrators, etc.) it could be possible to directly interrogate such gateways and to find the currently connected wireless devices (provided that they offer such an interrogation service).

The second module, *device connectivity discovery*, based on the detected devices and their types, tries to establish how they are interconnected using the following approaches:

- *wired device discovery protocols* – such protocols enable directly connected devices to discover information about each other by advertising information about each device over every interface; the most used protocols are CDP (Cisco Discovery Protocol) and LLDP (Link Layer Discovery Protocol); in order to build a full connectivity graph of the interconnected devices, one needs to run the used protocol on most of the devices, which should not be a problem since the protocols are fairly simple, most of the existing network devices already support at least one of them, and, moreover, there are free/commercial implementations for servers or desktop computers;
- *wireless discovery* – the connectivity of wireless devices that use WiFi protocols can be requested from each access point.

Using the data obtained from both the *device discovery* module and the *device connectivity discovery* module, a database with the existing devices and their connectivity with each other (either wired or wireless) can be built. Each device has a set of attributes which describes it, the most common being *device ID, type, name, short name, description, brand, model, physical location, ports, MAC address, IP addresses, OS, CPU, memory, etc*. These attributes change very rarely, usually only due to upgrades (more memory, newer software, etc.) or device reconfiguration (e.g. different IP address).

B. Status Monitoring

The network status monitoring component is responsible for monitoring the status of all the devices in the network. A device's status is defined by a number of parameters, which usually vary in time, such as CPU or other loads, current network traffic on each port, battery status, device temperature, etc.

The monitoring of the devices' status is done depending on each device's type. Most of the network devices have already implemented standard network management protocols like SNMP. By enabling the SNMP agent on them, the system is able to request status information about each device. For each type of device a set of relevant parameters is established and their values are periodically stored and updated. Such standard management protocols are also available on desktop PCs and servers. However, in the case of complex servers, some other management software could be further installed and interrogated directly from our system. This is the case of monitoring different software services on the servers and knowing their status.

In the case of very low cost IoT devices with very limited resources (hardware, software, energy), SNMP like network management protocols are mostly nonexistent directly on the devices. Such networks of IoT devices usually communicate with the Internet using a more powerful gateway, which can have SNMP support, but it will provide status information only about the gateway. For these kinds of IoT devices, one way to obtain information about their functionality is to passively monitor their data traffic and to extract status information from this data. Another way is to add some lightweight messaging protocol into the device that is able to report some simple status information to the monitoring system. Such protocol could be, for example, MQTT (Message Queuing Telemetry Transport), a publish/subscribe protocol that is used for wireless sensor networks. The protocol could be integrated into the device provided that access to the software source code is possible.

C. User Notifications

The user notifications component of the system generates different *real-time* alert messages based on the current status of each individual device (Fig. 2). It can also respond to requests from the user about a specific parameter of a particular device.

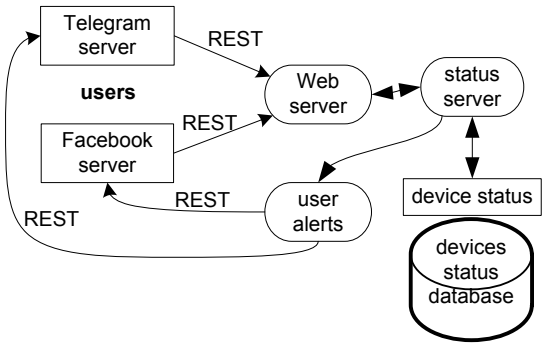

Fig. 2. User notifications

In case of events, different methods can be used for notification as follows:

- *mobile phone short messages (SMS)* – this is one of the simplest way of notifying users about events happening in the network. When an event occurs, based on a set of rules, a short text message is generated and sent to the users using either a GSM modem attached to a server or an Internet based SMS service;
- *instant messages using Facebook messenger or Telegram* – different data from different devices can be queried using the Facebook or Telegram instant messaging application, as shown in Fig. 3a and 3b. These can be simple queries for a specific parameter or aggregated data, like, for example, a visual representation of the network traffic for a period of time. The user can also automatically receive instant messages about alerts or notifications about significant events (ordinary or out of the ordinary ones).

Both instant messaging tools mentioned are used by means of their provided API (Facebook messenger API and Telegram API). Each time an alert is triggered, which requires a new

message to be sent to the user, an API method is triggered in the form of a REST (Representational State Transfer) call to the appropriate instant messaging server, with the message and user ID. The messenger server then dispatches the message to the user. The user has also the possibility to interrogate the VizNet system by means of simple messages sent via the messenger application. This could be simple commands that query, for example, the status of a device. When the messenger server receives such a message, the appropriate webhook on the VizNet server is invoked and a similar REST call is made with the message (command) and user ID. The system verifies the user ID, obtains the necessary information from the status server, then creates a response to the REST call, and, finally, the messenger server generates a response to the user.

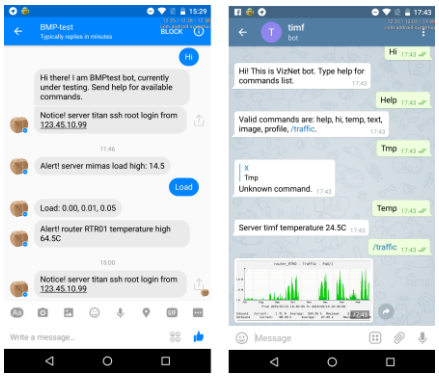

Fig. 3. Instant messenger notifications
a) with Facebook, b) with Telegram

IV. VizNet Visualization Interface

The 3D visualization interface is done using a WebGL enabled Web browser. Most of today's modern browsers (Mozilla Firefox, Chrome, Internet Explorer, and Opera) natively support WebGL. The monitored network is represented as a 3D virtual environment in the browser. The network is divided into so called *3D worlds*, each world consisting of a number of devices or other worlds. The idea of *a world* is similar to that of *a view*, a subset of the network. Devices can belong to multiple worlds at the same time, allowing the user to define worlds based on different criteria such as location-based (e.g. main datacenter, office1, laboratory1, etc.), function-based (e.g. main communication backbone, Internet connection, etc.), device-based (e.g. IoT devices of a certain type, wireless devices, etc.), or any other suitable criterion.

Tthe visualization module is illustrated in Fig. 4. The browser connects to a Web server configured to read information from the device database and it creates a 3D representation of each world, by generating a webpage with JavaScript. It uses the REST (Representational State Transfer) protocol for obtaining detailed information about each device and its connectivity to other devices. Further, the script from that page connects to the status server via a websocket to obtain real time information about any changes in the status of the devices present in the 3D representation of the world.

An example of the 3D visualization for a desktop browser is presented in Fig. 5. Objects in the 3D world (i.e. devices)

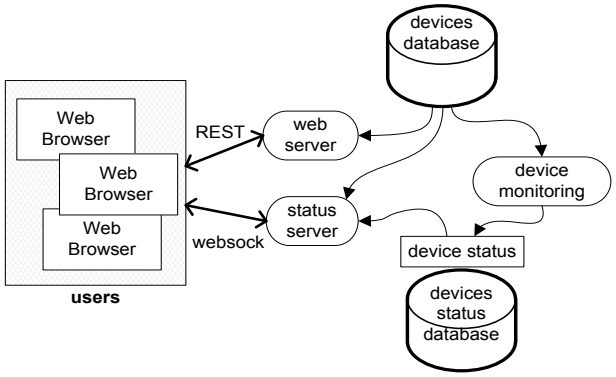

Fig. 4. Visualization module

can easily be rotated or moved around by the user in the browser, thus allowing her to arrange them as necessary. The new position of each device is automatically saved on the server, allowing future sessions to have the same view. Each device type has a different 3D representation, making easy for the user to recognize different devices. Also different colors provide visual clues about the status of each device or of different device components. For example, a red sphere might represent an unconnected switch port, while a green one a connected port to another device. Connections are represented as lines (or 3D pipes), where their thickness could mean, for example, either the amount of traffic through that connection or the maximum bandwidth allowed by the physical connection. VizNet allows simultaneous users to connect to the 3D visualization module. Each user can monitor different worlds, i.e. different aspects of the network. Each user can have its own personalized representation for each world. The interface can also be used on mobile devices (Fig. 6).

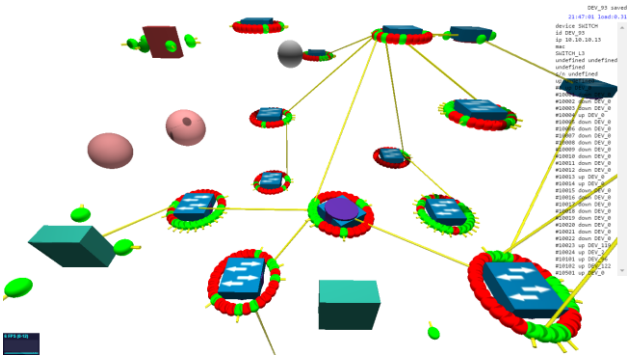

Fig. 5. 3D visualization on browser

V. Conclusions and Future Work

Modern networks include an ever increasing number of complex technologies, protocols, and devices. The recent explosion of IoT devices and the usage of different connecting technologies have increased the complexity and size of such networks, making their management even more complicated. We introduced here the prototype of VizNet, a system that allow its users (either administrators or other authorized users) to analyze and understand different aspects of the network infrastructure and to make sense of what happens within it, by using advanced 3D dynamic visualization techniques, but simpler options are available as well (2D or text only). Our main goal has been to show each user a compact real time

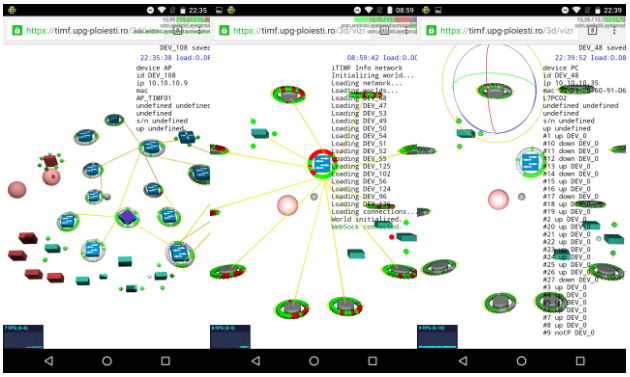

Fig. 6. 3D visualization on mobile phones

view of a relevant subset of the network, along with its status and dynamics in time. The system provides also for IoT monitoring, being able to tackle specific constraints and difficulties. VizNet has been designed to be both friendly and qualitatively useful to the user. The system can be used from anywhere on the Internet by using a Web browser on a variety of devices (laptops, mobile phones, tablets, etc.). It can provide early notifications and visualization of events in the network.

For the time being, the system is still in an early development stage. We have implemented most of the discovery, status, and notification components, and also a WebGL based 3D interface. We have performed various use experiments as well. For future development, we are looking into using a time series database for storing device status data, thus allowing the user to inspect historical data later, after different events have already occurred. Also, from our experience, the browser based visualization using WebGL is relatively slow compared to other native 3D visualization engines. This can be due to the additional layers for image rendering present in the browser. Therefore, we are currently experimenting with a game engine based visualization module, in hope of a better 3D rendering performance.

We will continue to improve both the VizNet architecture and its implementation in order to obtain a stable version that can be released as *free software*. Thus, a broader community can explore the concepts presented here, can contribute to the system, and also may benefit from using VizNet.

References

[1] A.S. Tanenbaum, D.J. Wetherall, Computer Networks, 5th ed., Prentice Hall, 2011, pp.362-392.

[2] Gartner IT Glossary, Internet of Things. http://www.gartner.com/it-glossary/internet-of-things/ (accesed August 2016).

[3] G. Moise, Z. Constantinescu, M. Vladoiu, M. Dumitru, Networking and Security, UPG University of Ploiesti Publishing House, 2015.

[4] Verizon, , "State of the market: Internet of Things 2016," April 2016. https://www.verizon.com/about/sites/default/files/state-of-the-internet-of-things-market-report-2016.pdf (accesed August 2016).

[5] GSMA Intelligence, "Analysis Understanding 5G: Perspectives on future technological advancements in mobile," December 2014. https://www.gsmaintelligence.com/research/?file=141208-5g.pdf (accesed August 2016)

[6] S. Olariu, T. Hristov, and G. Yan, "The Next Paradigm Shift: From Vehicular Networks to Vehicular Clouds", in Mobile Ad Hoc Networking: Cutting Edge Directions, 2nd ed., S. Basagni, M. Conti, S. Giordano and I. Stojmenovic, Eds. John Wiley & Sons, Inc., Hoboken, NJ, USA, 2013, pp. 645-700.

[7] Z. Constantinescu, M. Vladoiu, "Challenges in Safety, Security, and Privacy of Vehicle Tracking Systems, International Workshop on Systems, Safety and Security for Automotive, Passengers and Good Protection", 17th International Conference on System Theory, Control and Computing Joint Conference ICSTCC, Sinaia, Romania, pp. 607 – 612, October 2013.

[8] C. de Morais Cordeiro, and D. P. Agrawal, "Mobile Ad hoc Networking", Center for Distributed and Mobile Computing, ECECS, University of Cincinnati, 2002. (accessed August 2016). http://eecs.ceas.uc.edu/~cordeicm/course/survey_ad_hoc.pdf

[9] K. B. Miller and E. R. Brandon, "Improving Network Monitoring and Security via Visualization," preprint https://arxiv.org/pdf/1511.08795 (accesed August 2016).

[10] B. Claise, "Cisco Systems NetFlow Services Export Version 9," Request for Comments – RFC 3954.

[11] B. Claise, "Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of IP Traffic Flow Information," Request for Comments – RFC 5101.

[12] E. Boschi, L. Mark, J. Quittek, M. Stiemerling, P. Aitken, "IP Flow Information Export (IPFIX) Implementation Guidelines," Request for Comments – RFC 5153.

[13] C. So-In, A Survey of Network Traffic Monitoring and Analysis Tools, http://www.cs.wustl.edu/~jain/cse567-06/ftp/net_traffic_monitors3/ Report, (accesed August 2016).

[14] M. Iliofotou, P. Pappu, M. Faloutsos, M. Mitzenmacher, G. Varghese, "Network monitoring using traffic dispersion graphs (TDGs)," IMC '07 Proceedings of the 7th ACM SIGCOMM conference on Internet measurement, pp. 315-320, October 2007.

[15] NfSen - Netflow Sensor, http://nfsen.sourceforge.net/ (August 2016).

[16] P. Venezia, "7 free tools every network needs. From device discovery to visibility into systems, networks, and traffic flows, these free open source monitoring tools have you covered," October 2014, http://www.infoworld.com/article/2825120/network-monitoring/7-free-open-source-network-monitoring-tools.html (accesed August 2016).

[17] F. Fischer, F. Mansmann, D. A. Keim, S. Pietzko, M. Waldvogel, "Large-Scale Network Monitoring for Visual Analysis of Attacks," in Visualization for Computer Security, Lecture Notes in Computer Science vol. 5210, September 2008, pp. 111-118. [VizSec '08 Proceedings of the 5th international workshop on Visualization for Computer Security].

[18] A. Danalis and C. Dovrolis, "ANEMOS: An Autonomous NEtwork MOnitoring System," In Proc. of 4th Passive and Active Measurements Workshop, San Diego, CA, USA, 2003.

[19] P. Abel , P. Gros , D. Loisel, C. R. Dos Santos , J. P. Paris, "CyberNet: A framework for managing networks using 3D metaphoric worlds," Annales Des Télécommunications, vol. 55, issue 3, pp 131-142, March 2000.

[20] D. W. H. Ten, S. Manickam, S. Ramadass, H. A. Al Bazar, "Study on Advanced Visualization Tools In Network Monitoring Platform," Computer Modeling and Simulation, 2009. EMS '09. Third UKSim European Symposium on, pp. 445 – 449, 25-27 Nov. 2009

[21] S. Lau, "The Spinning Cube of Potential Doom," Magazine Communications of the ACM - Wireless sensor networks, Volume 47 Issue 6, June 2004, Pages 25-26, ACM New York, NY, USA

[22] G. Conti, Security Data Visualization: Graphical Techniques for Network Analysis, No Starch Press, 2007, pp. 172.

[23] V. T. Guimaraes, C. M. Dal Sasso Freitas, R. Sadre, L. M. Tarouco, L. Z. Granville, "A Survey on Information Visualization for Network and Service Management," IEEE Communications Surveys & Tutorials, vol 8, issue 11, pp. 285-323, July 2015.

[24] A. Tabona, "The Top 20 Free Network Monitoring and Analysis Tools for Sys Admins, May 2015, http://www.gfi.com/blog/the-top-20-free-network-monitoring-and-analysis-tools-for-sys-admins/ (accesed August 2016.