МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
Факультет кібербезпеки, комп'ютерної та програмної інженерії
Кафедра комп'ютерних систем та мереж

"ДОПУСТИТИ ДО ЗАХИСТУ"
Завідувач кафедри
_____ Жуков І.А.
"_____"_____2020 р.

ДИПЛОМНА РОБОТА

(ПОЯСНЮВАЛЬНА ЗАПИСКА)

випускника освітнього ступеня "МАГІСТР"

спеціальності 123 «Комп'ютерна інженерія»

освітньо-професійної програми «Комп'ютерні системи та мережі»

на тему: **"**Аналіз ефективності безпроводових мереж**"**

Виконавець: _____Олійник Д.А.

Керівник: _____Надточій В.І.

Нормоконтролер: _____Надточій В.І.

Засвідчую, що у дипломній роботі
немає запозичень з праць інших авторів
без відповідних посилань

_____Олійник Д.А.

Київ 2020

MINISTRY OF EDUCATION AND SCIENCE OF UKRAINE
NATIONAL AVIATION UNIVERSITY
Faculty of Cybersecurity, Computer and Software Engineering
Computer Systems and Networks Department

"PERMISSION TO DEFEND GRANTED"

The Head of the Department

_____ Zhukov I.A.

"_____"_____2020

MASTER'S DEGREE THESIS

(EXPLANATORY NOTE)

Specialty: 123 Computer Engineering

Educational-Professional Program: Computer Systems and Networks

Topic: **"**Efficiency analysis of wireless networks**"**

Completed by: _____Oliinyk D.A.

Supervisor: _____Nadtochii V.I.

Standard's Inspector: _____Nadtochii V.I.

Kyiv 2020

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет кібербезпеки, комп'ютерної та програмної інженерії
Кафедра комп'ютерних систем та мереж
Освітній ступінь: «Магістр»
Спеціальність: 123 «Комп'ютерна інженерія»
Освітньо-професійна програма: «Комп'ютерні системи та мережі»

"ЗАТВЕРДЖУЮ"

Завідувач кафедри
_____ Жуков І.А
"_____"_____2020 р

ЗАВДАННЯ
на виконання дипломної роботи
Олійника Данила Андрійовича
(прізвище, ім'я та по-батькові випускника в родовому відмінку)

1. Тема дипломної роботи: "Аналіз ефективності безпроводових мереж"
затверджена наказом ректора від 25.09.2020 р. № 1793/ст.
2. Термін виконання роботи (проекту): з 1 жовтня 2020 р. до 25 грудня 2020 р.
3. Вихідні дані до роботи (проекту*):* Комп'ютерні мережи, сенсорні мережи, протоколи, програмні середовища: Python, Cacti, Pygal, Matplotlib, WiFi IEEE 802.11.
4. Зміст пояснювальної записки: Вступ, огляд видів, стандартів, взаємодій безпроводових мереж. Розрахунки оцінки ефективності енергоспоживання безпроводових мереж. Огляд протоколу SNMP як спосіб транспортування даних для подальшого аналізу їх ефективності.
5. Перелік обов'язкового графічного (ілюстративного) матеріалу: Результати представлені у виді графіків у практичній частині роботи (4), базуючись на даних роботи мережевого обладнання.

NATIONAL AVIATION UNIVERSITY

Faculty of Cybersecurity, Computer and Software Engineering
Department: Computer Systems and Networks
Educational Degree: "Master"
Specialty: 123 "Computer Engineering"
Educational-Professional Program: "Computer Systems and Networks"

"APPROVED BY"
The Head of the Department
_____Zhukov I.A
"_____"_____2020

Graduate Student's Degree Thesis Assignment

Oliinyk Danylo Andriyovich

1. Thesis topic: "Efficiency analysis of wireless networks**"**

approved by the Rector's order of 25.09.2020 р. № 1793/ст

2. Thesis to be completed between з 1 жовтня 2020 р. до 25 грудня 2020 р.

3. Initial data for the project (thesis): Computer networks, sensor networks, protocols, software environments: Python, Cacti, Pygal, Matplotlib, WiFi IEEE 802.11.

4. The content of the explanatory note (the list of problems to be considered): Introduction, review of types, standards, interactions of wireless networks. Calculations for assessing the energy efficiency of wireless networks. Overview of the SNMP protocol as a way to transport data for further analysis of their effectiveness.

5. The list of mandatory graphic materials: The results are presented in the form of graphs in the practical part of the work, based on the data of the network equipment.

6. Календарний план-графік

№ пор.	Завдання	Термін Виконання	Підпис керівника
1	Узгодити технічне завдання з керівником дипломної роботи	1.10.20-6.10.20	
2	Виконати пошук та вивчення науково-технічної літератури за темою роботи	7.10.20-11.10.20	
3	Аналіз матеріалу на предмет ефективності	12.10.20-16.10.20	
4	Проаналізувати відомі підходи та методи щодо плану практичної реалізації	17.10.20-01.11.20	
5	Розробити прототип аналізу даних	02.11.20-03.12.20	
6	Порівняти очікуваний результат із фактичним результатом	04.12.20	
7	Виконати аналіз результатів, розробити рекомендації … та оформити пояснювальну записку.	05.12.20-08.12.20	
8	Оформити графічну частину записки та подати матеріали работи на антиплагіатну перевірку матеріалів	09.12.20-14.12.20	
9	Отримати рецензію та відгук керівника. Надати матеріали роботи на кафедру.	15.12.20-18.12.20	

7. Дата видачі завдання: “1” жовтня 2020 р.

Керівник дипломної роботи _____Надточій В.І.
(підпис керівника)

Завдання прийняв до виконання _____Олійник Д.А.
(підпис випускника)

6. TIMETABLE

#	Completion stages of Degree Project (Thesis)	Stage Completion Dates	Signature of the supervisor
1	Agree on the terms of reference with the thesis supervisor	1.10.20-6.10.20	
2	Perform a search and study of scientific and technical literature on the topic of work	7.10.20-11.10.20	
3	Material efficiency analysis	12.10.20-16.10.20	
4	Analyze the known approaches and methods for the practical implementation plan	17.10.20-01.11.20	
5	Develop a prototype of data analysis	02.11.20-03.12.20	
6	Compare the expected result with the actual result	04.12.20	
7	Analyze the results, develop recommendations and draw up an explanatory note	05.12.20-08.12.20	
8	Make a graphic part of the note and submit the materials of the work for anti-plagiarism	09.12.20-14.12.20	
9	Get a review and feedback from the manager. Provide work materials for the department.	15.12.20-18.12.20	

7. Assignment issue date: 1.10.2020

Diploma Thesis Supervisor _____Nadtochii V.I.
(Signature)

Assignment accepted for completion _____Oliinyk D.A.
(Student's Signature)

ABSTRACT

Explanatory note to the thesis "Efficiency analysis of wireless networks ":

103 pp., 37 figures, 1 table, 23 references.

Object of research: Methods and technologies to solve the problem of efficiency analysis of wireless network via modulation of problem structure and solving via modern programming language and other monitoring tools.

Purpose: Investigate effective methods techniques and realize, implement them to make qualified efficiency analysis of wireless networks.

Research methods: Processing of scientific and technical literature sources, comparative analysis.

The results of the master's work are recommended to be used during scientific research and in the practical activity of network administrators that dealing with network equipment and control it’s performance.

RELYABILITY, AVAILABILITY, POWER CONSUPTION, READINESS FACTOR, EFFICIENCY ANALYSIS.

CONTENT

LIST OF SYMBOLS, ABBREVIATIONS, TERMS

BSS - Base Station System;

BSC - Base Station Controller;

WAN – Wide Area Network;

IEEE - Institute of Electrical and Electronics Engineers;

EMCU – Energy Management Control Unit;

LAN – Local Area Network;

MIMO - Multiple Input Multiple Output;

WiMAX - Worldwide Interoperability for Microwave Access;

MIB - Management Information Base;

SNMP - Simple Network Management Protocol;

TCP/IP - Transmission Control Protocol and Internet Protocol;

ISO - International Organization for Standardization;

DSL – Digital Subscriber Line;

OID – Object Identifier;

DN - Distinguished Name;

OSI - Open Systems Interconnection;

ADSL - Asymmetric Digital Subscriber Line;

SVG - Scalable Vector Graphics.

INTRODUCTION

Actuality of theme. The increasing interest in network efficiency analysis tasks is determined by the necessity of automation in either control functions. Therefore, today the search and implementation of efficiency network principles for human function by means of computer systems.

One of the most perspective directions of solving the given problem is based on the usage of modern protocols, programming language and monitoring tools.

This problem is solved by means of choosing the correspondent architecture and learning method. The analysis demonstrates that there is still no model, which could be sensitive to all types of distortion.

The purpose of the thesis is designing test network, testing current network and analyse simulated network for purpose of efficiency network analysis.

Research methods. The thesis covers diverse approaches to solve efficiency network analysis tasks. The main attention is paid to the implementation of wireless networks and their learning methods for the mentioned tasks.

Particularly, such wireless network is regarded in the thesis with advantages and disadvantages of the working method. Part 4 of the thesis is devoted to the designing and analysing data that was collected during the test of wireless network followed by the results of the developed network system so solve efficiency analysis tasks. Recommendations and methods to improve the efficiency of the working system are also proposed.

Scientific novelty of the obtained results. Methods and techniques of efficiency analysis network technologies were developed and improved, which allowed more effective usage of particular devices, network and also increase the speed-work as well as decrease the hardware expenses.

The practical significance of the thesis results lies in the possibility to implement the received results for the efficient exploitation of wireless networks analysis to solve it's tasks. In addition, it is worth noting that the proposed basic model architecture can be altered and improved to obtain better efficiency results.

PART 1. WIRELESS SENSOR NETWORK NODE POWER CONSUMPTION ANALYSIS

1.1. Setting the problem

One of the main requirements for BSS is their autonomy, which can be fulfilled by reducing the power consumption of each node.

BSS (Base Station System) is a system of base stations for cellular communication of the GSM standard. The main task of the BSS is to establish, maintain and destroy the connection between the MS (Mobile station) and the NSS (Network Switching System), as well as between the MS and the packet data network. Includes BSC (Base Station Controller) - base station controller, BTS (Base Transceiver Station) - base station, as well as TRAU / TC (Transcoding Rate and Adaptation Unit) - transcoder.

The following methods are used to solve this problem:

- Definition and optimization of the transmission turn-on time;
- Multi-link transmission, i.e. sending messages through intermediate nodes instead of direct long-distance transmission;
- Pre-processing and reducing the amount of data required for transmission.

Devices without internal power supplies must receive power directly from the environment. Some devices of this type are equipped with solar panels, while others convert mechanical vibrations into electricity. In the case of ZigBee wireless devices, the use of radio wave energy is the most promising.

However, for any type of BPS node with or without a power supply, it is important to improve its energy efficiency. To solve this problem, it is necessary to conduct a number of studies to analyze the energy consumption of the node and the FSU router. It is also advisable to optimize the operation of the transceiver and preliminary processing of data at the network node.

1.2. Literary review

Aspects of power consumption by the BSS nodes to optimize network operation were considered in the following works. However, no detailed analysis of the power consumption of the node or the entire network was made in these works.

It should be noted that microminiaturization and development of microelectronics made it possible to create transceivers and microcontrollers that can run on battery power for several years. However, this service life of the BCC node is typical for its use in ultra low power modes. This mode also provides short time to receive and transmit messages within the network. Therefore, for information gathering networks with intensive information exchange it is advisable to take measures to increase energy efficiency.

Zelenin A. has paid attention to the analysis of energy cycles of BSS nodes in his works. N., Vlasova V. A. Some aspects of energy efficiency improvement in FSS are considered in the paper. The proposed ratios for calculating the energy consumption of the BSS nodes allow estimating the battery charge consumption using primary parameters at various functions performed by means of a combination. At the same time, the paper does not answer the question: "How does hardware implementation of the FSU node affect its energy cycle?

The American company Texas Instruments pays great attention to the analysis and solution of problems in the development of energy-efficient FSU nodes, as well as the creation of software and hardware for them. Experimental data from the research of this company are taken for analysis of power consumption of the wireless sensor network node in this work.

1.3. IEEE 802.15.4 specification and BSS

According to the IEEE 802.15.4 specification, BSS is divided into three types of devices: terminal devices (OUs), routers and the only coordinator that manages and collects all information from the network. According to the standard, any router must take over the role of the coordinator in case of its failure. However, this happens very rarely due to the

fact that the BCC coordinator, as a rule, has a stationary power supply and is quite often still connected to a gateway, such as ZigBee-Ethernet. In practice, the only autonomous power supplies have their own OU and routers. Therefore, the analysis of power consumption should be done only with them.

When considering the peculiarities of wireless sensor networks implementation on the basis of ZigBee top level network protocols specification, data compression was suggested as one of the ways to preprocess and reduce the volume of transmitted node data in work.

Energy balancing of the transmission route was considered in paper Y. Chen. The proposed new approach to EBMR (Energy-Balancing Multipath Routing) routing is based on taking into account the energy constraints of power supply sources for FSU nodes with energy balancing of the route.

The article gives an overview of various options for building wireless sensor networks based on MeshLogic technology. The technique of calculation of average power consumption of nodes is given.

the lifetime of their batteries. Given that the technology MeshLogic is a set of hardware and software, which implements a set of network protocols for packet data transmission between any devices in the network and is a de facto universal base for the creation of FPGA, it is possible to adapt the methodology for calculating the average value of power consumption of nodes on networks built on other hardware and software platforms.

The principles of data encoding and multiplexing, which can be applied to reduce the amount of data transmitted by the node, are described in the literature.

Among the factors constraining the development of FSS, it should be noted that there is insufficient development of methods to assess the energy efficiency of algorithms, as well as the lack of information and communication technologies.

measuring systems, which allow to control energy consumption parameters of functioning systems. The existing works in the field of energy consumption mainly concern theoretical issues of building optimal algorithms of data packet routing under conditions of significant and variable traffic volume in BSS. At the same time, the work devoted to the

construction of an information and measurement system for monitoring and analysis of power consumption of wireless sensor systems seems to be relevant today.

It was noted in [12, 13] that specific energy density of modern chemical current sources is growing very slowly, and an additional constraint is the issue of safe operation of high energy-intensive batteries.

It can be concluded that the life time of the BSS node is limited by the life time of the power source,

The task of reducing energy consumption is more important than ever, and the efficiency of its solution directly affects the further development of wireless sensor networks.

1.4. Energy Analysis

Wireless sensor network node power consumption analysis

To analyze the power consumption of a wireless sensor network node, let us consider its structure. It contains a sensor that receives data from the external environment, a microcontroller, memory, radio transceiver, autonomous power source in Figure 1.1.

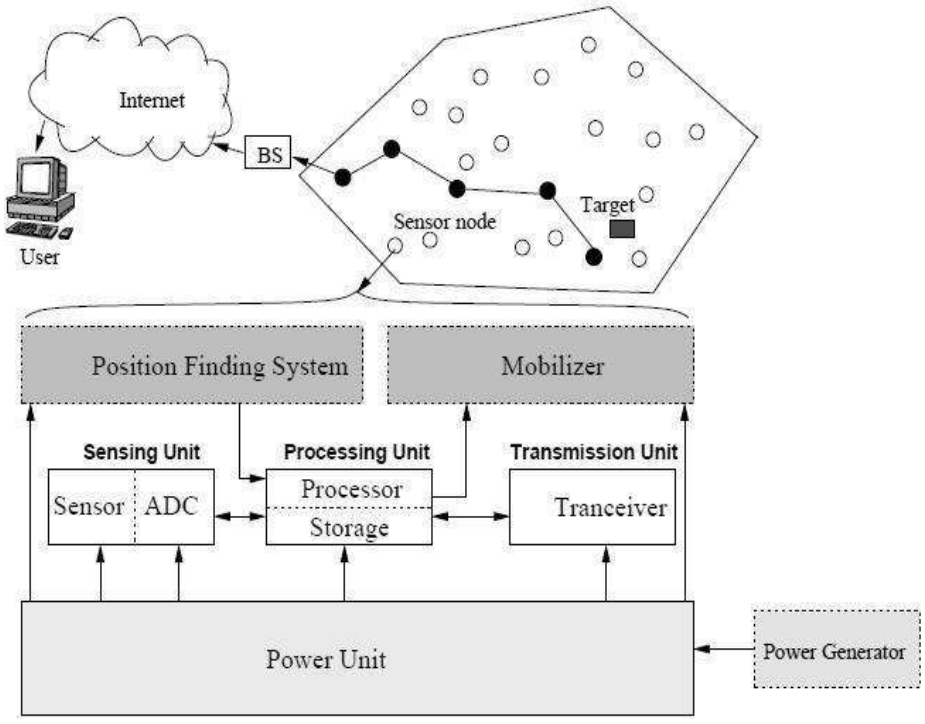

Fig. 1.1 Wireless sensor network node schematic summary

As can be seen from Fig. 1, the energy of the power supply (BSS node) is spent to power the sensor(s), the microcontroller with memory, which performs processing of the received information from the sensor (sensor), as well as on the radio-receiver. It can be assumed that the main resources that need to be protected in the FSU are the power supply and the capacity of the FSU node.

Having analyzed the power consumption of the node shown in Figure 1.2, when the address is set and the coordinator is found, we can see that most of the energy is spent on communication (receiving, listening and transmitting data), not on data processing or storage.

The power consumption of the BSS node built on the basis of SS2530 transceiver is shown.

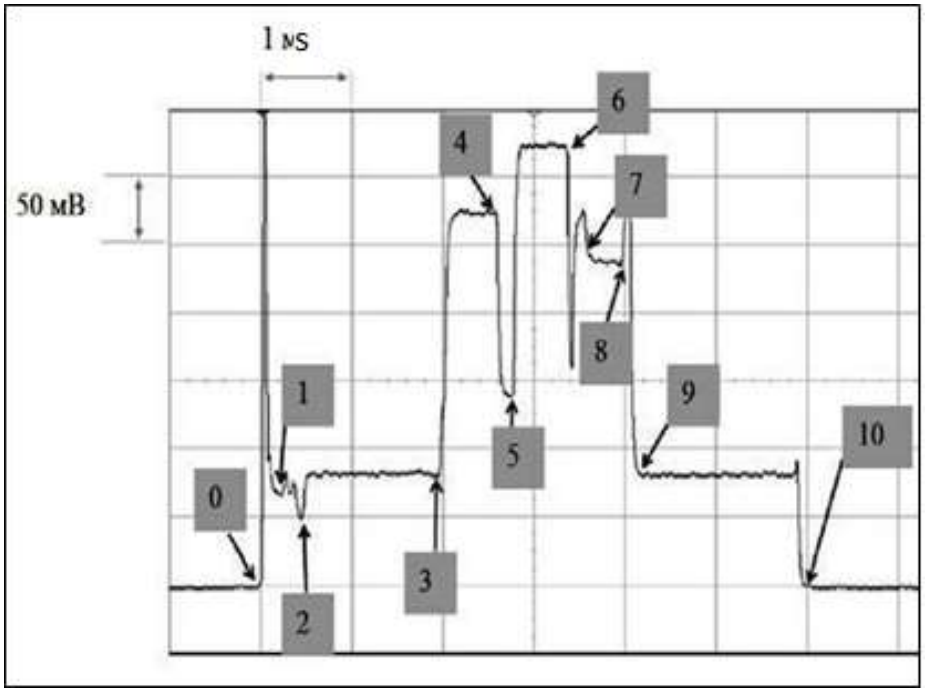

Fig. 1.2. Energy consumption of the BSS node, built on the basis of transceiver SS2530

Transition from point 0 to point 1 shows the start mode of the transceiver, with the energy consumption of 12 mA for 0.2 ms. Further from point 1 to point 2 the microcontroller operates at 16 MHz on the external quartz resonator for 0.25 μs with energy consumption of 6 mA, when switching to 32 MHz the external quartz resonator (points 2-3) energy consumption increases to 7.5 mA for 1.7 ms. After the microcontroller enters the operating mode, listening to the air (RX mode) to search for a signal BSS coordinator for 1.2 ms with a significantly higher energy consumption of 27 mA. Next, the node needs to contact the

network coordinator and he goes into the transmission mode, but not instantly, and after a while, equal to 0.2 ms with an energy consumption of 14 mA (points 4-5 in Fig. 2). At message transmission, the most energy is consumed by the power source, namely 32 mA during 0.5 ms. After the transmission, there is a switch from the transmission mode to listen to the air (points 6-7) for 0.2 ms with the cost of energy equal to 25 mA. Then, the air is again listened to with the energy consumption equal to 23 mA during 0.35 ms. Points from 8 to 9 show the process of ending listening to the air and setting the address of the node. When switching from point 9 to point 10, the node goes to sleep mode. In hibernation mode the SS2530 spends 1 μA in 1 s. Thus, you can see that most of the energy is spent on receiving, listening and transmitting data, rather than on processing or saving data.

An example of an oscillogram of current consumption by a sensor node in a long mode of operation, which is possible with large amounts of transmitted data, is shown in the figure. 3.

From Figure 1.3. shows that the current consumption by the sensor node, depending on the phase of processing the request, is uneven for a long period of time. As it can be seen from the oscillogram, during the internal processing the node consumes about 15 mA. When listening to the air, consumption increases to 20 mA, and when transmitting a message to 22 mA.

All energy consumption values of the BCS node will be depend on the internal architecture of the node itself. BSS node consists of 5 main components. In this case microcontroller, transceiver and memory can to be made on one crystal that contributes as miniaturization of the BSS node itself, as well as reducing its power consumption.

To unambiguously answer which modes work should be kept to a minimum, consider nominal power consumption transceiver SS2530, which is a receiving device… transmitter and microcontroller (MCS-51 core) on one crystal, table 1.

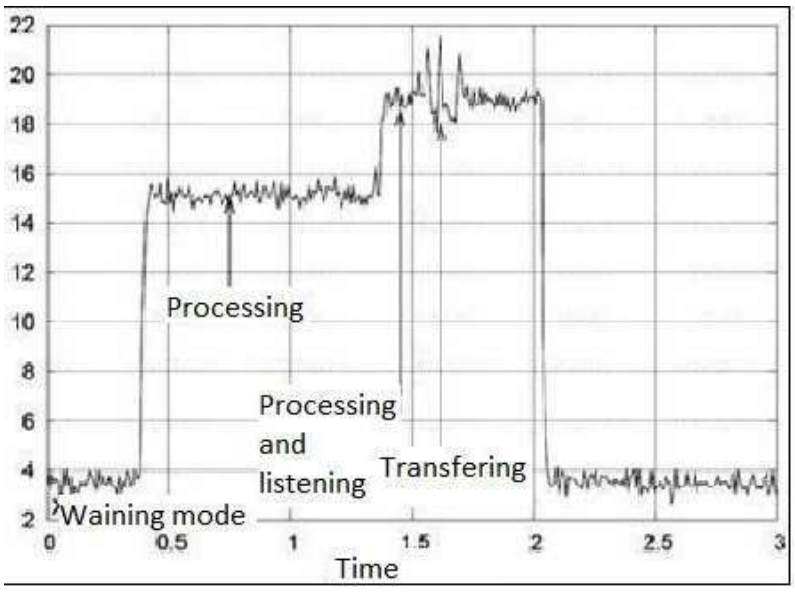

Fig. 1.3. BCS unit

Rated power consumption of transceiver SS2530 in different modes of current sensory node depending on the phase of inquiry processing

Table 1

Transmitter and microcontroller (MCS-51 core) on one crystal

Type of operation mode	Listening to the air (Mode RX)	Sending messages (Mode TX to 1dBm)	Microcontroller operation	Wake-up mode (for 4μs)	Sleep mode
Consumption, mA	24	29	9	0,2	from 0,0004 to 0,001

Power consumption of Ee node in one cycle, Based on the analysis performed, it can be defined as the sum of the energy consumption: ESleep EMCU Ercv Etrans Eather, where Eslep is the energy consumed by the node during the sleep phase;

EMCU - the energy consumed by the node during the operation of the microcontroller or the processing core of the transceiver when it is absent; Ercv - the energy consumed by the node during the operation of the microcontroller or the processing core of the transceiver. energy consumed by the node during reception; Etrans - energy consumed by the node during transmission; Eather - energy consumed by the node in other modes (wake-up mode, etc.).

To find out the power consumption of a wireless sensor router, you need to determine how it differs from the FSU endpoint. FSU router acts as a "mini-coordinator" within the area R.

From Figure 1.4 you can see that the router can collect information directly or through intermediate nodes and coordinate its transmission. Therefore, the principle of its power consumption is the same as that of the MA, but more as a result of more active interaction with the nodes of the FSU.

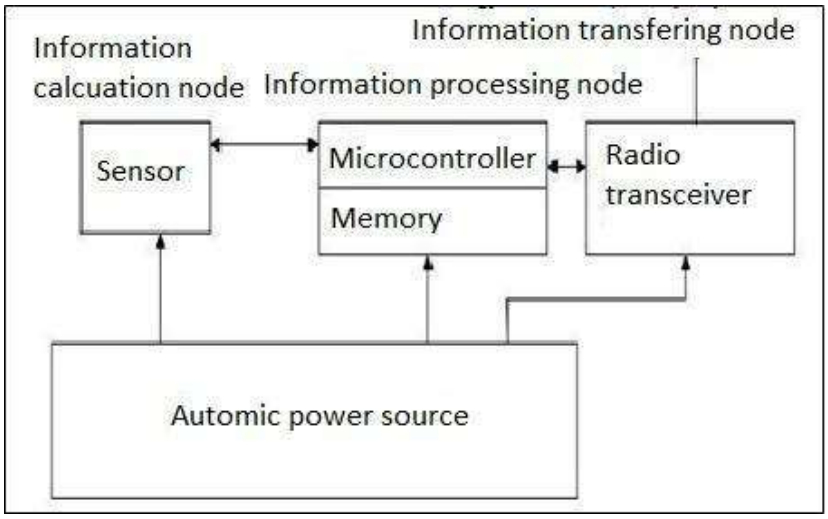

Fig. 1.4. R network area where the router coordinates the work of nodes

1.5. Energy consumption

The power consumption of the router, in one cycle, can be determined from expression that is represented in Figure 1.5.:

$$E_r \quad E \; {}_k E_e \; {}_m E_m \; {}_a E_a \;,$$

Fig. 1.5. Expression example of the power consumption of the router in one cycle

Where Ee is the energy consumed by the router to communicate with the coordinator; Em is the energy consumed by the router to communicate with its subordinate nodes in m; Ea - router power consumption for communication with a network agitator. In expression (2), the value of R will mainly affect Em . The more R the more Em will be spent

to connect the remote node to the router. To minimize the power consumption in the BSS data aggregation method can be used. From . rice. 4 you can see that apart from the router and the endpoints nodes it is possible to have a network aggregator in the R area. In the case when the coordinator needs identify integral characteristic for any part of the network, one of the nodes of this The area is assigned by the aggregator. The aggregator collects on the other hand nodes site individual values of the defined characteristic, calculates aggregate function and passes this value to the network coordinator. In this case, the total cost of information transfer is significantly lower than when there is no aggregator. From fig. 4 shows that some nodes of BSS are simpler send a message to the network aggregator than to transmit his router. Calculating average power consumption nodes and determining their effectiveness To determine the average power consumption nodes in the network, it is necessary to define work cycles, that are present in the BSS.

There are a number of others, but based on the simulation modeling and practical studies their impact on the power consumption of the entire network can be neglected. To assess the effectiveness of the algorithms of information collection and routing protocols used in the BSS, it is necessary to know the life time and power consumption of the node and the network as a whole. The amount of power consumption depends on many factors, therefore, in order to estimate the lifetime of the network, power consumption models are used, which can realistically describe the network energy consumption. The energy consumption of the whole network is represented by an expression that is represented in Figure 1.7.

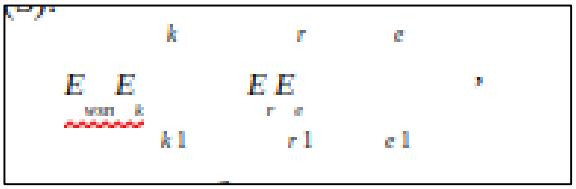

Fig. 1.7. The energy consumption of the whole network

Where Ek - energy consumed by the FSU coordinator, k=1; Er - energy consumed by the routers r; r - number of routers in the network; Ee - energy consumed by the end devices e; e - number of end devices in the network. Expressions (1) - (3) show what power sources, end devices, routers and all FSUs spend energy on. Analyzing the figure. 5, you can see that the main node cycle is repeated. As a rule, the transmission of the 1st message

passes in one cycle during the network activity cycle. Let us determine the power consumption of the network taking into account the activity cycles and their operating time that is represented in Figure 1.8.

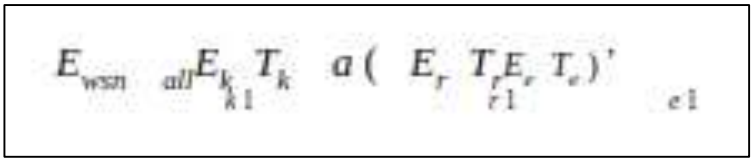

Fig. 1.8. Power consumption of the network

Where a - number of network activity cycles; Tk - time spent on the coordinator's work; Tr - time spent on the router's r-th work during the activity cycle; Te - time spent on the e-th endpoint work during the activity cycle. Usually, the lifetime of a network is determined by the lifetime of the device that will fail first. Let's calculate the network life time via formula that is represented in Figure 1.9.

Fig. 1.9. Network life time

Where Ta is the time of network activity cycles.

Approbation of research results Mechanisms to reduce energy consumption The BSS node depends on the model of data collection. Also, the energy consumption of the node is affected by the algorithm works nodes, built by at models collections information. The results of the research represent interest precisely from the point of view of describing the real network consumption. Experimental data works were taken from sources [3,5]. Amount . power consumption depends on . arrays factors, that is why, for togo in order to evaluate time lives networks, use . models power consumptions that can realistically describe network consumption. Suggested ideas are tested on adequacy of c via imitation simulations в system Castalia. Also proposed approach to energy consumption analysis BSC was approved by youth forum "Radio electronics and youth in the 21st century" (2013) and received a positive feedback. With the help of the proposed approach to the analysis of power consumption of nodes of BSS and simulation can be assessed the characteristics of the network without expensive in-situ simulation and determine the applicability of modules from different manufacturers for the projected network.

Energy consumption is a key parameter of FSU performance quality, so the question of its calculation when creating such systems arises as one of the first. The article analyzes the power consumption of wireless sensor network nodes. The method of calculating the power consumption of terminal nodes and router is given, as well as the calculation of network life time. This work will be useful when choosing the hardware for building FOSS.

1.6. WiMAX standard

WiMAX (Worldwide Interoperability for Microwave Access) telecommunications technology designed to provide universal wireless communication over long distances for a wide wide range of devices (from workstations and laptops to cell phones). It is based on the IEEE 802.16 standard, also Wireless MAN (WiMAX should be considered a slang name, as technology, but the name of the forum where the Wireless MAN was agreed upon). Scheme about how WiMAX works is represented in Figure 1.10.

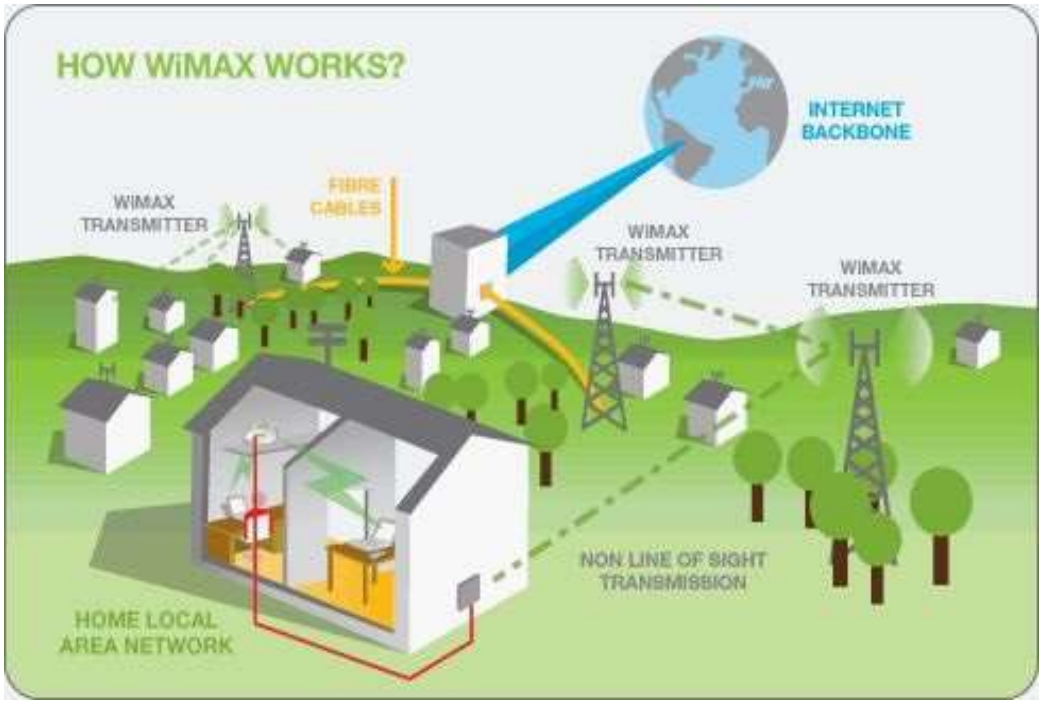

Fig. 1.10. Representation of WiMAX

WiMAX is suitable for the task of connecting Wi-Fi access points to each other and other segments of the Internet, as well as providing wireless broadband access as an alternative to leased lines and xDSL. WiMAX allows you to access the Internet at internet

access at high speeds, with much greater coverage than Wi-Fi networks. This makes it possible to use the technology as a backbone, this allows you to use the technology as trunk circuits, for which are the continuation of traditional leased and xDSL lines, as well as local networks. As a result, this approach allows you to create scalable high-speed networks within cities.

WiMAX is a long-range system that covers kilometers of space, which typically uses licensed frequency spectrums (although the use of unlicensed frequencies is also possible) to to provide a point-to-point Internet connection by an ISP to the end user. Different standards in the 802.16 family provide different standards in the 802.16 family provide different types of access, from mobile (similar to data transmission in mobile phones) to fixed access (an alternative to wired access in which the user's wireless equipment, in which the user's wireless equipment is tied to a location).

Unlike WiMAX, Wi-Fi is a shorter-range system typically covering tens of meters, which uses unlicensed frequency bands to provide network access. Wi-Fi is typically used by users to access their own local network, which may not be connected to the Internet. If WiMAX can be compared to mobile communications, Wi-Fi is more like a landline cordless phone (radiotelephone).

WiMAX and Wi-Fi have completely different Quality of Service (QoS) mechanism. WiMAX uses a mechanism based on establishing connection between the base station and the user's device. Each connection is based on a special scheduling algorithm that can guarantee a QoS parameter for each connection. Wi-Fi, in turn, it uses a QoS mechanism similar to the one used in Ethernet in which packets are prioritized differently. This approach does not guarantee the same QoS for each connection.

A set of benefits is inherent in the entire WiMAX family, but its versions differ significantly from each other. The developers of the standard have been looking for optimal solutions for both fixed and mobile to reconcile all requirements within a single standard. Although a number of basic requirements coincide, the focus of technologies on different market niches has led to the creation of two separate versions of the standard (or rather, they can be considered two different standards). Each of WiMAX specification defines its own operating frequency bands, bandwidth bandwidth, radiation power, transmission and access

methods, signal coding and modulation methods, radio frequency reuse principles radio frequency reuse principles, and other metrics. Therefore, WiMAX systems, based on the IEEE 802.16 e and d versions of the standard are virtually versions of the IEEE 802.16 standard are virtually incompatible.

The main difference between the two technologies is that fixed WiMAX allows you to serve only static subscribers, while the mobile is oriented to work with users travelling at speeds up to up to 150 km/h.

Mobility means roaming functions and seamless between base stations when the subscriber moves (as is the case with cellular networks). occurs in cellular networks). As a special case, mobile WiMAX can be used for fixed-line users as well. With the invention of mobile WiMAX, there is an increasing emphasis on development of mobile devices, including special telephone handsets (similar to a conventional mobile smartphone), and computer peripherals (USB-radio modules and PC cards). The equipment for using WiMAX networks is available from several manufacturers and can be Installed both indoors (devices the size of a conventional xDSL modem), and outside of it. It should be noted that the equipment designed for indoor installation and does not require professional skills installation, of course, is more convenient, but it is able to work at significantly shorter distances from the base station than professionally installed externally. Therefore, equipment installed Therefore, equipment installed indoors requires a much larger investment in network infrastructure.

In general, WiMAX networks consist of the following main parts:

- Base stations and subscriber stations, as well as the equipment that connects;
- Base stations with each other, with the service provider and to the Internet.

The structure of networks of IEEE 802.16 standards family is similar to traditional GSM networks (base stations operate at distances of up to tens of tens of kilometers, for their installation is not necessary to build towers - it is allowed installation on the roofs of houses in compliance with the conditions of direct visibility between stations). WiMAX is used both to solve the problem WiMAX is used to solve the "last mile" problem, and to provide network access to office and district networks.

To connect the base station to the subscriber uses high frequency range of radio waves from 1.5 to 11 GHz. Under ideal conditions The data transfer rate can reach up to 70 Mbit/s and does not require direct line of sight between the base station and the receiver. Connections (line-of-sight) are established between the base stations using the frequency range of the between the base stations (line-of-sight) using a frequency range of 10 to 66 GHz, communication speeds of up to 140 Mbit/s are possible. data rates of up to 140 Mbit/s are possible. At least one base station is connected to the provider's network. at least one base station is connected to the provider's network using classic wired connections. However, the greater the number of BSs connected to the provider's network, the higher the data transfer rate and reliability of the network as a whole.

1.7. Routing methods in wireless networks

There are three types of routing - simple, fixed and adaptive. The fundamental difference between them is the extent to which they take into account changes in the topology and load of the network when solving the problem of selecting a route. Also there is difference between routing and flooding that is represented in Figure. 1.11.

Routing	Flooding
--> Routing table is required.	--> No routing table is required.
--> May give shortest path.	--> Always gives shortest path.
--> Less reliable.	--> More reliable.
--> Traffic is less.	--> Traffic is high.
--> No duplicate packets.	--> Duplicate packets are present

Fig. 1.11. Difference between routing and flooding

Simple routing is characterized by the fact that in route selection does not neither changes in network topology nor changes in its state (load). It does not provide directional

packet transmission and has low efficiency. Its advantages are simple implementation routing algorithm and ensures the stable operation of the network in case of failure of failure of its individual elements. Some practical application received varieties of simple routing: random and avalanche routing.

The peculiarity of random routing is that for transmission package from the communication node is selected one, randomly chosenfree direction. The packet "wanders" through the network and with a finite probability ever reaches the addressee. In doing so, neither the optimal packet delivery time or efficient use of network bandwidth network capacity.

Avalanche routing (or: packet flooding of all available output directions) involves passing a packet from a node in all directions other than where the packet came from in that node. Since this happens in every node, the phenomenon of packet "multiplication" takes place, which dramatically degrades the bandwidth utilization of the network. To prevent this to prevent this from happening, it is necessary to mark a copy of the packet and to destroy the repeatedly passing through each node. node, the duplicates re-passing through it. The main advantage of this method is to guarantee optimal delivery time of a packet to the addressee, because of all directions, on which a packet is transmitted, at least one provides this time. The method can be used in unloaded networks, when the requirements to minimize the time and reliability of packet delivery are sufficiently high.

Fixed routing is characterized by the fact that when selecting route takes into account changes in the topology of the network and does not take into account changes in its load. For each destination node, the direction of transmission is selected according to a route table (directory) that determines the shortest paths. The directories are compiled in the network control center. They are compiled anew and modified when the network topology changes. Lack of adaptation to changes in load results in latency on the network. A distinction is made between single-path and multi-path variations of fixed routing. The former is based on a single packet transmission path between two subscribers, which is prone to failure and congestion, while the The second is based on several possible paths between two subscribers, from which a preferred path is chosen. of which the preferred path is

chosen. Fixed routing is used in networks with little varying topology and established packet streams are established.

Adaptive routing is when the decision on the direction of packet transmission is based on changes in both topology as well as network load. There are several modifications of adaptive routing that differ in what kind of information is used in route selection. Such modifications have become widespread modifications such as local, distributed, centralized and hybrid routing.

Local adaptive routing is based on the use of information available in a given node and includes: a routing table, which defines all directions of packet transmission from this node; information about the the state of the output links (operating or not operating); the queue length packets waiting to be transmitted. Information about the state of other communication nodes is not used. A route table identifies the shortest routes, which ensures that a packet reaches its destination in the shortest amount of time. An advantage of this method is that a route selection decision is made using the most recent state of a node status. The disadvantage of the method is its "short-sightedness", because the route choice route choice is made without taking into account the global state of the whole network. Hence, there is always the danger of transmitting a packet through an overloaded route.

Distributed adaptive routing is based on the use of information specified for local routing and data received from neighboring nodes in the network. Each node generates a table routes (directory) to all destination nodes, where the routes with the minimum packet delay time. Prior to network operation, this time is estimated based on the topology of the network. During the operation of the network, the nodes periodically exchange with neighboring nodes, so-called delay tables, which indicate the load (packet queue length) node. After the delay tables are exchanged, each node recalculates the delays and adjusts the routes based on the incoming data and the queue lengths in node itself. Delay tables can be exchanged not only periodically, but also asynchronously in case of sudden changes in load or network topology. Taking into account the state of neighboring nodes when choosing a route considerably increases the efficiency of routing algorithms, but it is achieved at the expense of increasing the loading of the network with service information. In addition, the

information about the changes in the state of the nodes propagates through the network is relatively slow, so the route choice is made on the basis of somewhat outdated data.

Centralized adaptive routing is characterized by the fact that routing task for each node in the network is solved in the routing center routing center (CM). Each node periodically generates a message about its status (queue lengths and availability of communication lines) and sends it to the CM. According to these data in the CM for each node is compiled table of routes. Naturally, the transmission of messages to the CM, the formation and distribution of route tables - all this is associated with time delays, hence the loss of efficiency of such a especially when the load ripple in the network is large. In addition, there is the risk of losing control of the network if the CM fails.

Hybrid adaptive routing is based on the use of route tables sent by the CM to the nodes of the network, combined with analysis of queue lengths in the nodes. Consequently, the principles of centralized and local routing principles are implemented here. Hybrid routing compensates for the disadvantages of centralized routing (routes, generated by the center are somewhat outdated) and local ("short-sightedness" of the method) and perceives their advantages: the routes of the center corresponds to the global state of the network, and consideration of the current state of the node ensures the timeliness of the solution to the problem.

Conclusions on Part 1

Devices without internal power supplies must receive power directly from the environment. Some devices of this type are equipped with solar panels, while others convert mechanical vibrations into electricity.

According to the IEEE 802.15.4 specification, BSS is divided into three types of devices: terminal devices (OUs), routers and the only coordinator that manages and collects all information from the network. According to the standard, any router must take over the role of the coordinator in case of its failure.

All energy consumption values of the BCS node will be depend on the internal architecture of the node itself. BSS node consists of 5 main components. In this case

microcontroller, transceiver and memory can to be made on one crystal that contributes as miniaturization of the BSS node itself, as well as reducing its power consumption.

There are three types of routing - simple, fixed and adaptive. The fundamental difference between them is the extent to which they take into account changes in the topology and load of the network when solving the problem of selecting a route.

PART 2 JUSTIFICATION OF THE DESIGN OF DEVELOPMENT WIRELESS COMMUNICATION NETWORKS BASED ON IEEE STANDARDS 802.11

2.1. Trends in the development of telecommunication wireless networks

The relevance of this topic is that for the development of society, it is necessary to implement innovative systems. This is due to the fact that humanity is moving to a new level of communication and transmission of information. Now you don't have to be close to send a message. It is possible to transmit information from around the world. Therefore, communication systems have a very strong impact on all areas of human life. The development of communication at the beginning of the XXI century is generally characterized by the following concepts: universalization, integration, intellectualization - in terms of technical means and in terms of networking; globalization, personalization - in terms of services. Progress in the field of communications is based on the development and development of new telecommunications technologies, as well as on the further development and improvement of existing technologies that have not yet exhausted their potential.

The rapid development of telecommunications, based on the achievements of microelectronics, has dramatically increased the efficiency of transportation, distribution, processing, storage of information, as well as the throughput of systems and transmission media. The main feature of modern telecommunications is the transmission and processing of signals in digital form. Digitalization has allowed to build cost-effective digital communication systems with a wide range of services, compared to analog.

Deepening the informatization of society necessitates the creation of an effective system of information dissemination, and its constant improvement. Advanced development of telecommunications is a necessary condition for the creation of business infrastructure, the formation of a favorable investment climate, the development of modern information technology.

The choice of switching method in telecommunication networks is determined by the requirements for communication quality. Thus, the transmission of speech requires a minimum time delay of signals, and some errors in the signals - not so critical. And data transmission, on the contrary, is highly sensitive to errors and much less - to delays in the transmission process. Therefore, public telephone networks use channel switching without complex error protection mechanisms. And in data networks use packet switching, as well as complex but effective mechanisms for protection against errors. The theory of Wi-Fi evolution of 802.11 standard is represented in Figure .2.1.

	802.11 (Legacy)	802.11b (Legacy)	802.11a (Legacy)	802.11g (Legacy)	802.11n (HT)	802.11ac (VHT)	802.11ax (HE)
Year Ratified	1997	1999	1999	2003	2009	2014	2019 (Expected)
Operating Band	2.4 GHz/IR	2.4 GHz	5 GHz	2.4 GHz	2.4/5 GHz	5 GHz	2.4/5 GHz
Channel BW	20 MHz	20 MHz	20 MHz	20 MHz	20/40 MHz	20/40/80/160 MHz	20/40/80/160 MHz
Peak PHY Rate	2 Mbps	11 Mbps	54 Mbps	54 Mbps	600 Mbps	6.8 Gbps	10 Gbps
Link Spectral Efficiency	0.1 bps/Hz	0.55 bps/Hz	2.7 bps/Hz	2.7 bps/Hz	15 bps/Hz	42.5 bps/Hz	62.5 bps/Hz
Max # SU Streams	1	1	1	1	4	8	8
Max # MU Streams	NA	NA	NA	NA	NA	4 (DL only)	8 (UL & DL)
Modulation	DSSS, FHSS	DSSS, CCK	OFDM	OFDM	OFDM	OFDM	OFDM, OFDMA
Max Constellation / Code Rate	DQPSK	CCK	64-QAM, 3/4	64-QAM, 3/4	64-QAM, 5/6	256-QAM, 5/6	1024-QAM, 5/6
Max # OFDM tones	NA	NA	64	64	128	512	2048
Subcarrier Spacing	NA	NA	312.5 kHz	312.5 kHz	312.5 kHz	312.5 kHz	78.125 kHz

Fig. 2.1. Theory of Wi-Fi evolution of 802.11 standard

Contradictory requirements for speech and data transmission is one of the reasons for the existence of different networks, such as telephone networks, data networks and others.

The high rate of development of the telecommunications market around the world is due to several factors, the main of which are technological and socio-economic. The ever-increasing demand for voice and high-speed data services encourages telecom operators to crown the capacity of their networks by upgrading them and switching to new, more advanced technologies. Advances in technology are leading to the emergence of a growing range of communication services in the global telecommunications market. At the same

time, there is an increase in consumer demand for quality of these services. In recent decades, there has been a migration of telecommunications technologies in two main directions: from voice services to the transmission of large data streams over high-speed channels; from fixed to mobile, which can only provide wireless communication.

The area that includes these two areas of telecommunications technology is called the multiservice network, it is inextricably linked to broadband wireless access networks. The term broadband access usually refers to the organization of a high-speed channel from a few Mbps from the subscriber to any public resource, such as the Internet, public telephone network, etc. 16 It is also very important that broadband access provides the subscriber with the integration of various services (Internet, specialized data, video, voice, etc.).

Until recently, existing broadband systems had significant drawbacks. Yes, they worked only in conditions of direct visibility from the subscriber to the base station, which greatly narrows the scope. Some systems did not have sufficient characteristics for high-quality service to a large number of subscribers, while other systems could not provide high-quality data services with high speed. But the most important thing is that almost all systems had insufficient frequency efficiency, ie they could not provide high speeds for each subscriber in conditions of limited frequency resource.

In connection with these problems, there is a need to create a new class of systems - broadband wireless access systems with service integration.

It was necessary to create cheap terminal devices for mass use, which would not require direct visibility to the base station.

The beginning of the development of broadband technologies occurred in the mid-90's. There are now more than 100 million broadband users in the world, access to which is organized by various technical means: DSL, ADSL, cable, satellite channel, terrestrial radio channel, etc. Types of DSL are reptoduced in Figure ****. Almost all analysts estimate an annual increase in the number of users by 30-40%, which means that maintaining this rate in 7-8 years can expect a significant increase in the number of users of multiservice networks, with the main increase expected from customers in Southeast Asia. Example and types of DSL is represented in Figure 2.2.

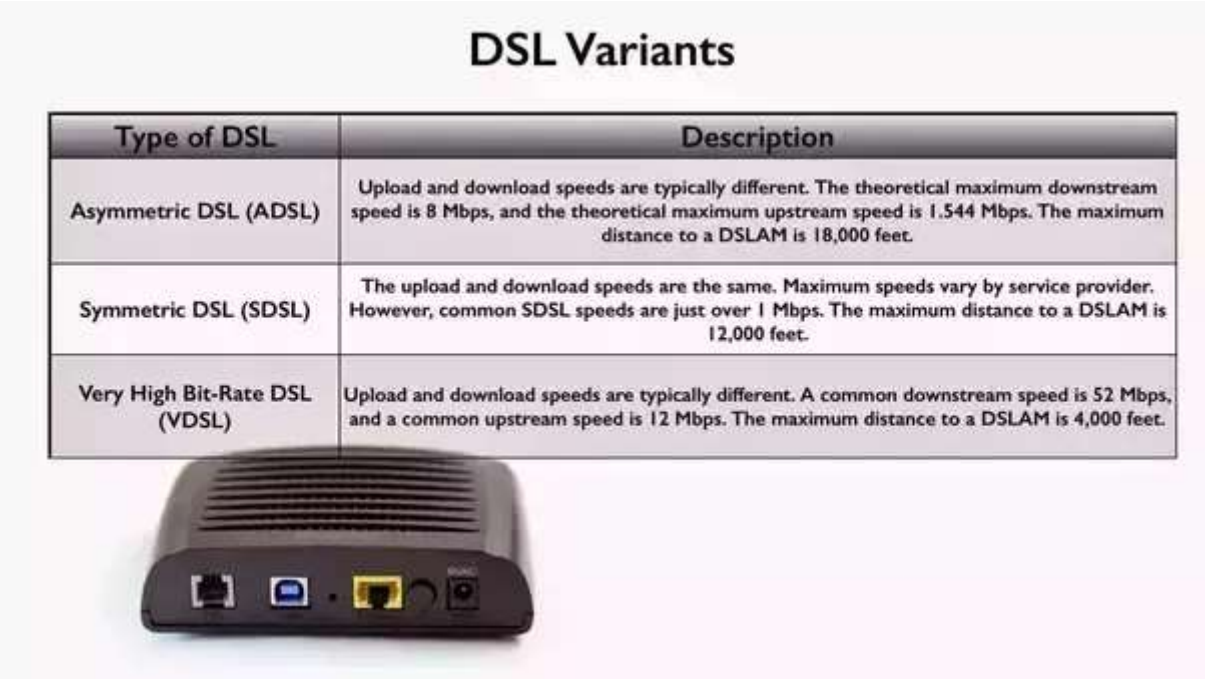

Fig. 2.2. Example and types of DSL

In terms of growth, broadband access can be compared to the growth of the Internet as a whole, only with a delay of about a decade. It is possible that the leaders in broadband will be countries with less developed structure of the Internet, the period of development of which will be the largest growth of this technology.

To assess the quality of broadband access, qualitative and quantitative indicators are usually used, which include: transmission speed, channel reliability, 17 quality set of services. Access speeds per subscriber have grown from 64 to 2048 kbit / s and above over the past few years. The quality of services is supplemented over time by telephony, video and a wide range of information and business applications. Prices are constantly falling.

Wireless broadband has better prospects. Its share of broadband access should increase from 7-8% now to 15% in five years on pessimistic forecasts and 25% on optimistic ones.

The development of wireless data transmission networks in Ukraine and around the world, which is often talked about as a wireless revolution in the field of data transmission, is associated with such advantages as:

- the ability to dynamically change the network topology when connecting, moving and disconnecting mobile users without significant loss of time, ie flexibility of architecture;
- high speed information transfer (1-1000 Mbps and above);

- speed of design and deployment; high degree of protection against unauthorized access;
- refusal of expensive, and not always possible, laying or rent of fiber or copper cable.

2.2. Analysis of modern wireless multiservice networks.

Wireless multiservice networks, being a well-known technology 10 years ago, are now becoming the mainstream of development. And at conferences devoted to television broadcasting, there is an increasing opinion that new wireless technologies will soon "cut the cable".

Yes, say supporters of this view, it will not happen tomorrow, but in the next five years - for sure. The struggle for the so-called digital dividend between television and cellular operators is intensifying. And while the former's main trump card is its wide coverage and much more efficient use of spectrum in mass broadcasting, the latter emphasize the availability of interactive or, as they now say, nonlinear services.

If we recognize that networks need to be classified, then wireless networks are distinguished by the distance of access and coverage of the service area: from individual network (PAN) and area (LAN, Local area network) to the global network (GAN, Global area network). Between them are metropolitan (MAN, Metropolian area network), campus (Campus area network) and regional (WAN, Wide area network) networks. Obviously, in some cases the boundaries between these classifications are very blurred.

A personal network covers communication between personal devices of use, usually the connection of the gadget to a desktop computer via Bluetooth or infrared. LAN is a home or office network. Campus is the same, but larger; it usually includes several local area networks. Urban - the same, but within the city or urban agglomeration. The regional network connects several cities, the global - covers the whole world. The distribution is quite conditional, but a general idea of the technologies used and services provided, he can give.

According to the topology, multiservice networks are divided into "point-to-multipoint" and "point-to-point". In terms of content - for corporate and operator. The first

are created in the interests of business customers, the second - to provide services and receive a subscription fee for them. Broadband wireless access systems have come a long way to transform from a scattering of different technologies into a single network, united by common standards and a single common technology. Representation of point-to-point connection and multipoint connection is shown in Figure 2.3.

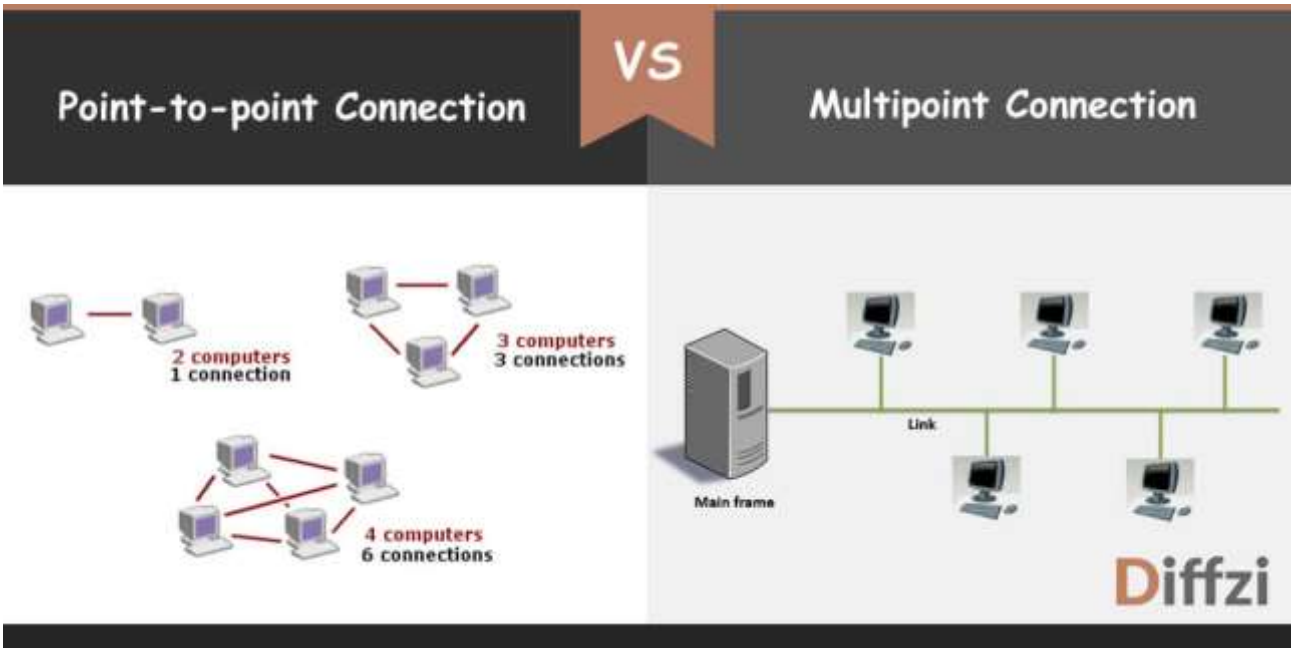

Fig. 2.3. Representation of point-to-point connection and multipoint connection

This process is far from complete, but the main direction of development is beyond doubt. Those wireless standards and technologies that have survived in this struggle, solve local niche problems. Now, if we talk about at least some mass application, all existing networks and operating standards are striving for multiservice: DVB is trying to somehow refine to establish nonlinear broadcasting, in the cellular standards hastily implement refinements for mass multicast. Of course (especially in the corporate sector), 19 remain separate data networks, including networks for the provision of simultaneous data and voice (VoIP) services, the lease of high-speed Internet access channels (LMDS - Local Miltipoint Distribution System), and radio bridges between networks in their standards and on their frequencies, and trunk channels at relay stations. But corporate customers increasingly prefer mass standards, primarily for the reason that the more common this standard, the cheaper the equipment for it.

The IEEE 802.11 standard, better known as Wi-Fi, was born in 1999 as the optimal solution to the problem of the latter is not even a mile, but rather an inch. Wireless radio

access, which requires no cable or sophisticated signal modulation algorithms, almost immediately gained a huge number of fans. A significant role was played by the fact that Intel has introduced an 802.11 adapter in the Centrino platform. Example of IEEE 802.11 standard schemelayers is represented in Figure. 2.4.

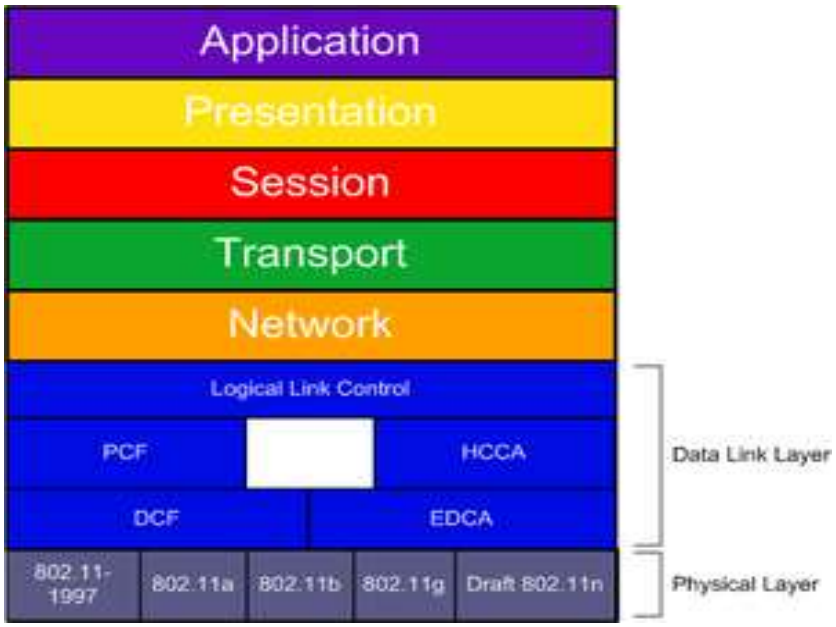

Fig. 2.4. Example of IEEE 802.11 standard schemelayers

A large number of laptops with Wi-Fi adapters appeared on the market, which, in turn, gave impetus to the development of networks. The first basic standard for wireless local area networks (Wi-Fi) IEEE 802.11 was developed in 1997, in 1999 the first mass version appeared - IEEE 802.11b. 802.11b adapters operate in the unlicensed 2.4 GHz band and provide theoretically a maximum transmission speed of 11 Mbps for a distance of up to 300 meters (also, of course, the theoretical limit). Its development is the 802.11g standard, which is fully compatible with 802.11b, operates in the same frequency range, the maximum transfer rate in the 802.11g standard is 54 Mbps. The 802.11n standard, adopted in 2009, operates in the 2.4-2.5 or 5 GHz bands, and the theoretically possible data transfer rate has been increased to 600 Mbps. Cisco already sells 1 Gbps wireless LAN equipment (802.11ac standard; 5 GHz). There is 802.11ac equipment that can operate from 450 Mbps to 6.93 Gbps (802.11n).

The biggest explosion in wireless networks in the past 10 years was made by cellular operators. And now, when we talk about multiservice wireless networks, we first have the term 4G, which generally characterizes the situation.

What awaits us in the future? And in the future we will have the next generation - now no one doubts that the 5G standard will dominate the market of wireless multiservice networks. The picture is simple: ITU has not yet defined the criteria for the standard, and manufacturers, operators and experts already have a strong opinion on this issue.

In 2016, the US Federal Communications Commission allowed the installation of 5G base stations without additional approval and allocated frequencies: 28 GHz (27.5-28.35 GHz), 37 GHz (37-38.6 GHz), 39 GHz (38 , 6 -40 GHz), 64-71 GHz (leaving the possibility to add frequencies above 95 GHz in the future). Also last year, Vodafone and Huawei tested 5G, overclocking the network to 20 Gbps for a single device at 71-76 GHz, 81-86 GHz and 92-95 GHz. South Korean operator SK Telecom has promised to organize a 5G connection for the 2018 Winter Olympics, and Russia's Megafon is also talking about test launches during the 2018 FIFA World Cup.

2.3 Support of MU-MIMO technology

The MIMO technology implemented in the 802.11n standard allows simultaneous transmission/reception of data between devices on the network. However, at a given point in time only one device can send and receive data while others are waiting their turn. The 802.11as standard greatly improves this situation. Multi-User Multiple-Input, Multiple-Output (MU-MIMO) technology has been implemented within the standard.

MU-MIMO creates a multi-threaded transmission channel, in which other devices do not wait their turn.

MU-MIMO-enabled devices can transmit up to four data streams simultaneously (up to four clients). This enables more efficient use of the wireless network and reduces the latency (latency to service) that occurs when the number of clients in the network increases significantly. Representation of MIMO is represented in Figure. 2.5.

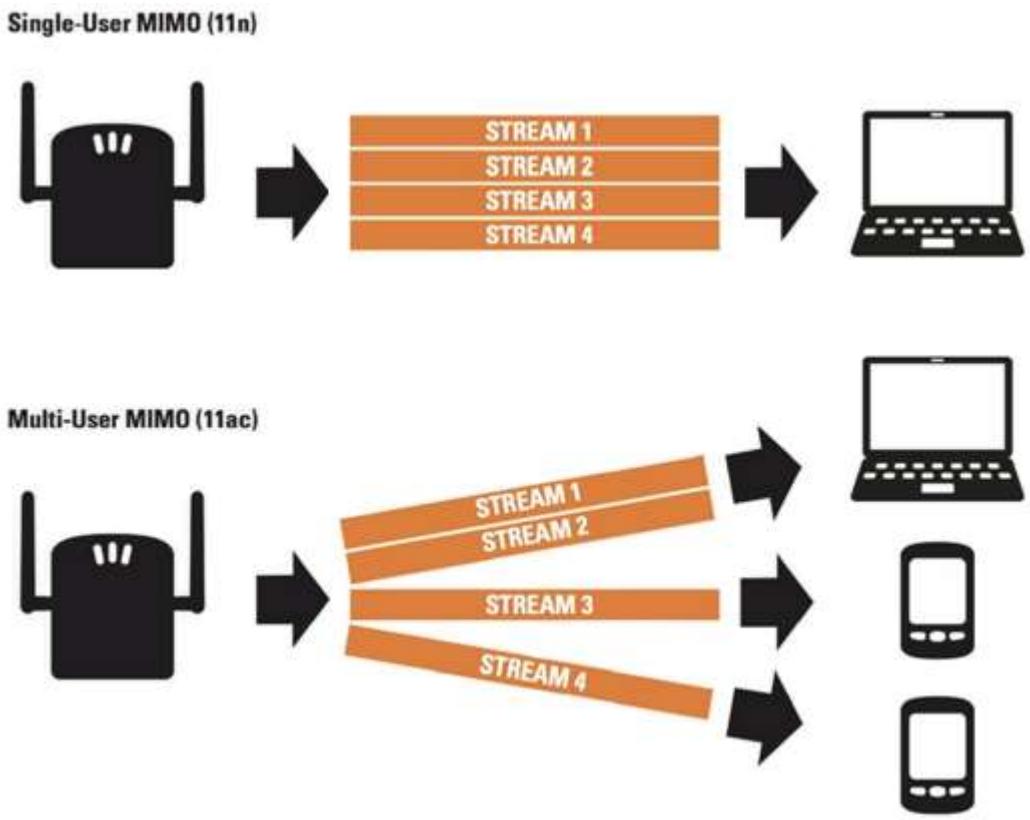

Fig. 2.5. Representation of MIMO

We, in turn, understand that miracles do not happen, and speeds are achieved primarily by increasing bandwidth. The number of threads is also increasing. MU-MIMO technology allows you to simultaneously transmit multiple independent data streams to different users. You can either increase the overall transmission speed, or increase the stability of the channel. MIMO is used in both LTE and Wi-Fi, but in mass models the number of streams does not exceed two. Massive MIMO - a technology developed for 5G, theoretically allows you to place dozens of small antennas in mobile devices and hundreds - in the transmitting station. For mass use, the expected number of streams used is 8 or 16. 5G operation is planned in the 4.5 GHz band, ie the waves have a lower penetrating power compared to 2 GHz in 3G and 2G, and base stations will have to be put more often.

For example, there are now about 200,000 BSs installed in the United States, and more than a million need 5G deployments. There is a plus in this: the situation of BS overload will be eliminated, when a large number of subscribers download heavy content through it. But in real urban conditions, 5G prototypes do not yet show results that satisfy potential customers. In addition, non-technical aspects emerge: the transition to small cells seriously exacerbates 21 the problem of numerous approvals at the installation site. And this

can greatly increase the price and increase the deployment time. Another feature of 5G is the support of the Internet of Things, the technology of which we will return to in this dissertation.

M2M services have been working through cellular networks of different generations for a long time. But it is assumed that in 5G it will not require the creation of separate services and applications, which will dramatically expand the market. Of course, the Internet of Things is at the same time one of the brightest 5G stickers for attracting consumers. Most subscribers now have enough spare capacity for existing 4G networks, which have just appeared en masse in Ukraine, so you can make them pay for new technologies only with a fundamentally new service, or at least what you can imagine. Currently, the main competition in the market is between cellular networks and Wi-Fi. The latter is expanding its presence, especially in large cities, based on the local level to the city. In New York, 7,500 phone booths will soon be replaced by gigabit Wi-Fi (802.11ac) points. In London, British Telecom is planning the same with 750 booths. For greater convenience of subscribers, these hotspots will provide the ability to recharge devices in order to promote Wi-Fi technology even more. The European Commission is currently examining the provision of 120 million euros for the WIFI4EU project - the deployment of free Wi-Fi networks in public places across the European Union. The distance that the 5G signal penetrates in urban development, compared to the Wi-Fi penetration zone, makes competition between standards not so pointless. The main difficulty of Wi-Fi in this competition is the fact that the speed of 1 Gbps in 802.11ac networks is really extremely difficult to achieve. However, research has shown that the brakes are not the shortcomings of the standard, but the low bandwidth of the cable that connects the point to the highway.

Google Fiber promoted this project in line with a similar project AT&T Gigapower. And only the transition to wireless technology, according to Google Fiber, will allow them to compete with AT&T, primarily by gaining deployment time, and appropriate testing of high-speed wireless radio access technology has allowed Google Fiber and other experts to talk about the emergence of such a phenomenon as wireless wireless cable.

2.4. Prospects for the use of Wi-Fi 802.11n technology

Prospects for the use of Wi-Fi 802.11n technology, taking into account other data transmission standards

Every year, the requirements for data networks are growing. Computers are increasingly immersed in a sea of multimedia, video conferencing is becoming increasingly necessary. The requirement for data transfer speed and user mobility is growing in direct proportion. Cisco has taken an important step in entering the market for basic high-performance wireless networks by announcing new wireless access points - Cisco Aironet, which operates under the 802.11n standard and is shown in Figure 2.6.

Fig. 2.6. Cisco AirNet, which operates under the 802.11n standard

It is worth recalling that for a long time, in order to develop mobility in enterprises, the company supports Cisco Motion, a product to support mobile services - Mobility Services Engine, which is a link needed to connect wireless networks to enterprise computing systems.

Prior to that, the corporate wireless network was usually isolated from other networks.

It had its own access and security systems, separate control protocols and terminals running incompatible applications. Meanwhile, the rapid spread of wireless and mobile devices - smartphones, tablets - in the consumer market creates the preconditions for growing demand for mobile features and in the corporate environment. Eventually, mobility should become an integral part of the corporate information technology infrastructure. For this idea to become a reality, Cisco experts believe, traditional incompatible wired and wireless network architectures must give way to a single, unified network architecture that provides employees with secure access to mission-critical business applications, including customer databases and inventory management systems. , anywhere and anytime.

This transition from wireless technology to mobility, explains Cisco Wireless Marketing Vice President Maciej Kranz, is at the heart of Cisco Motion technology. However, says Kranz, if corporations want to reap the full benefits of mobility - higher productivity, increased team cohesion, etc. - they must first build a more reliable and high-performance wireless infrastructure that can support more than just existing mobile applications. , but also applications that will appear on the market in the coming years. This is the purpose of the Cisco Aironet access point and Cisco Motion technology.

Designed for the office environment, the Cisco Aironet access point, with the latest configurations from 2016, is a compact, easy-to-install and energy-efficient device. Its real bandwidth, according to Kranz, exceeds 150 Mbps. We will not be completely tied to Cisco's strategy on this issue. What remains unchanged is the integral evolution of 802.11 standards that has not stopped since the end of the twentieth century. Below is table 1.1, which compares the IEEE 802.11n standard with its predecessors. As a result, we can conclude that the prospects for the use of Wi-Fi 802.11n technology are inextricably linked to the fact that the specifications of this standard have improved from year to year, leaving this technology very attractive, including for multiservice.

Conclusions on Part 2

The relevance of this topic is that for the development of society, it is necessary to implement innovative systems. This is due to the fact that humanity is moving to a new level of communication and transmission of information.

The rapid development of telecommunications, based on the achievements of microelectronics, has dramatically increased the efficiency of transportation, distribution, processing, storage of information, as well as the throughput of systems and transmission media. The main feature of modern telecommunications is the transmission and processing of signals in digital form. Digitalization has allowed to build cost-effective digital communication systems with a wide range of services, compared to analog.

In terms of growth, broadband access can be compared to the growth of the Internet as a whole, only with a delay of about a decade. It is possible that the leaders in broadband will be countries with less developed structure of the Internet, the period of development of which will be the largest growth of this technology.

In the future we will have the next generation - now no one doubts that the 5G standard will dominate the market of wireless multiservice networks.

MU-MIMO creates a multi-threaded transmission channel, in which other devices do not wait their turn.

MU-MIMO-enabled devices can transmit up to four data streams simultaneously (up to four clients). This enables more efficient use of the wireless network and reduces the latency (latency to service) that occurs when the number of clients in the network increases significantly.

PART 3. MANAGEMENT SYSTEM STANDARDS

3.1. Standardized elements of the management system

When formalizing the "manager-agent" scheme, the following aspects of its operation can be standardized:

- Agent - manager interaction protocol;
- The "agent - manageable resource" interface;
- The "agent - managed resource model" interface;
- The "manager - managed resource model" interface;
- Reference system on the availability and location of agents and managers, which simplifies the construction of a distributed control system;
- The managed resource model description language, i.e. the MIB description language;
- Scheme of inheritance of object model classes (inheritance tree), which allows building models of new objects based on models of more general objects, e.g. router models based on generalized communication device model;
- Scheme of hierarchical relations of managed object models (inclusion tree), which allows to reflect relations between separate elements of the real system, for example, belonging of switching modules to a certain switch or separate switches and hubs of a certain subnetwork.

Existing management system standards differ in that they may not standardize all of the above aspects of the manager-agent scheme.

In the standards of control systems, at least some way of formal description of models of managed objects is standardized, and the protocol of interaction between the manager and the agent is defined.

In practice today, there are two families of standards for network management - Internet standards, based on SNMP (Simple Network Management Protocol), and international standards ISO/ITU-T, using as a protocol of interaction between agents and managers protocol CMIP (Common Management Information Protocol).

Standards for management systems based on SNMP formalize the minimum aspects of the management system, while ISO/ITU-T standards - the maximum aspects, like most standards developed by ITU-T. Traditionally, SNMP-based management systems are mostly used in LANs and corporate networks, while ISO/ITU-T standards and CMIP are used in telecommunication networks.

3.2. Standards for SNMP-based management systems

Language describing the models of MIB and SNMP messages - language of abstract syntactic notation ASN.1 (standard ISO 8824:1987, recommendations ITU-T X.208);

Several specific MIB models (MIB-I, MIB-II, RMON, RMON 2), whose object names are registered in the ISO standards tree. Everything else is left to the developer of the management system. SNMP and closely related concept of SNMP MIB were developed to manage routers on the Internet as a temporary solution. But as is often the case with all temporary solutions, the simplicity and effectiveness of the solution ensured the success of this protocol, and today it is used to manage almost any kind of hardware and software computing networks. And although there is a steady trend of using ITU-T standards, which includes CMIP protocol, there are quite a few examples of successful use of SNMP management. SNMP agents are built into analog modems, ADSL modems, ATM switches, etc.

SNMP is an application layer protocol designed for the TCP/IP stack, although there are implementations for other stacks as well, such as IPX/SPX. SNMP is used to obtain information from network devices about their status, performance, and other characteristics, which are stored in the Management Information Base (MIB). The simplicity of SNMP is largely due to the simplicity of SNMP MIB, especially their first versions of MIB I and MIB II. In addition, the SNMP protocol itself is also very simple.

There are standards that define the structure of MIB, including a set of types of its objects, their names and allowable operations on these objects (for example, read").

MIB tree structure contains obligatory (standard) subtrees and also it can contain private subtrees allowing the manufacturer of an intelligent device to manage some specific device functions on the basis of MIB specific objects.

An agent in the SNMP protocol is a processing element that provides access to the values of the MIB variables for the managers located at the network control stations, thus allowing them to implement control and monitoring functions for the device.

The main management operations are placed in the manager, and the SNMP agent plays mostly a passive role, transmitting to the manager at its request the values of the accumulated statistical variables. In this case the device works with minimal costs to maintain the control protocol. It uses almost all of its processing power to perform its main functions of a router, bridge or hub, and the agent collects statistics and values of variables of the device state and transmits them to the manager of the management system.

3.3. SNMP primitives

SNMP is a request-response protocol, that is, for each request from the manager, the agent must transmit an answer. The peculiarity of the protocol is its extreme simplicity - it includes only a few commands.

The Get-request command is used by the manager to get the value of an object by its name from the agent.

The GetNext-request command is used by the manager to retrieve the value of the next object (without specifying its name) while sequentially browsing the object table.

Using the Get-response command, an SNMP agent sends a response to Get-request or GetNext-request commands to the manager.

The Set command is used by the manager to change the value of some object. The Set command is used to actually control the device. The agent must understand the meaning of the values of the object used to control the device, and based on these values perform the actual control action - disable a port, assign a port to a specific VLAN, etc. The Set command is also suitable for setting a condition, on fulfillment of which an SNMP agent is to send a corresponding message to the manager. Reaction to such events as agent initialization, agent

restart, connection failure, connection restoration, incorrect authentication and loss of the nearest router can be defined. If any of these events occur, the agent initializes an interrupt.

The Trap command is used by the agent to notify the manager when a special situation occurs.

SNMP v.2 adds the GetBulk command to this set, which allows the manager to retrieve multiple variable values in a single request.

Today there are several standards for management information databases for the SNMP protocol. The main ones are the MIB-I and MIB-II standards, as well as the remote management version of the RMON MIB database. In addition, there are standards for specific devices MIB of a particular type (eg, MIB for hubs or MIB for modems), as well as private MIB of specific equipment manufacturers.

The original MIB-I specification defined only operations to read values of variables. Operations to change or set the values of the object are part of the specifications of MIB-II.

The MIB-I version (RFC 1156) defines 114 objects, which are divided into 8 groups.

System - general data about the device (e.g. vendor ID, time of the last system initialization).

Interfaces - parameters of the network interfaces of the device (for example, their number, types, exchange rates, maximum packet size).

Address Translation Table - description of the correspondence between network and physical addresses (for example, according to ARP protocol).

Internet Protocol - data related to IP protocol (IP gateway addresses, hosts, IP packets statistics).

ICMP - data related to ICMP control message exchange protocol.

TCP - data about the TCP protocol (for example, about TCP-connections).

UDP - data pertaining to the UDP protocol (the number of transmitted, received and erroneous UPD datagrams).

EGP - data related to the Exterior Gateway Protocol, used on the Internet (the number of messages received with and without errors).

From this list of groups of variables shows that the MIB-I standard was developed with a strict focus on the management of routers that support TCP / IP stack protocols.

In version MIB-II (RFC 1213), adopted in 1992, the set of standard objects was significantly (up to 185) extended, and the number of groups increased to 10. 7.6 shows an example of the tree structure of the MIB-II object base. It shows two of the 10 possible object groups, System (object names begin with the prefix Sys) and Interfaces (prefix if). The SysUpTime object contains the value of the system uptime since the last reboot, the SysObjectID object contains the identifier of the device (e.g. a router). Standard MIB-II tree representation is shown in Figure. 3.1.

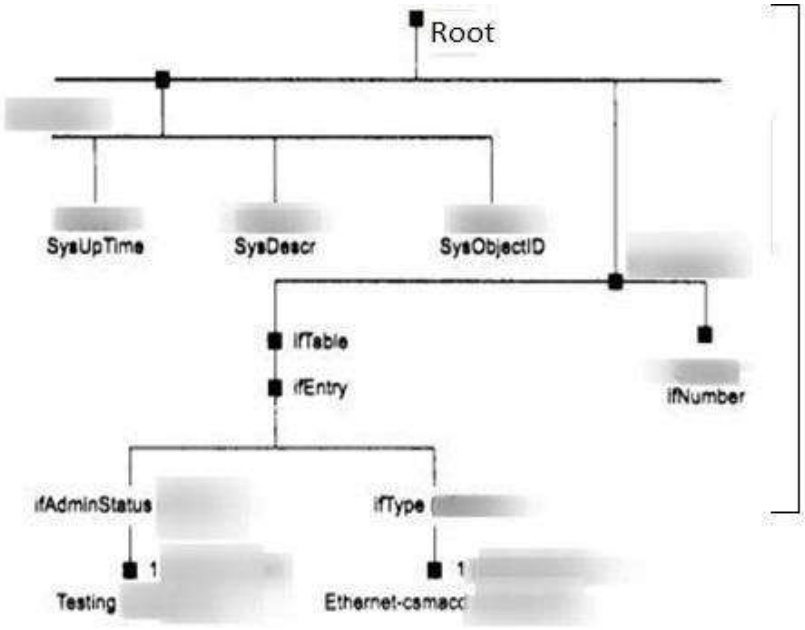

Fig. 3.1. Standard MIB-II tree

The ifNumber object defines the number of network interfaces of the device, and the ifEntry object is the node of the subtree describing one of the specific interfaces of the device. The ifType and ifAdminStatus objects included in this subtree define, respectively, the type and state of one of the interfaces, in this case the Ethernet interface.

The objects describing each specific interface of the device include the following.

- ifType is the type of protocol that the interface supports. This object accepts the values of all standard link layer protocols such as rfc877-x25, ethemet-csmacd, iso88023-csmacd, iso88024-tokenBus, iso88025-tokenRlng, etc.
- ifMtu is the maximum network layer packet size that can be sent through this interface.

● ifSpeed - interface bandwidth in bits per second (100 for Fast Ethernet).

● ifPhysAddress - physical address of the port, for Fast Ethernet this is MAC address.

● ifAdminStatus - desired status of the port.

● up - ready to transmit packets.

● down - not ready to transmit packets.

● testing - in test mode.

● ifOperStatus is the actual current status of the port and has the same values as ifAdminStatus.

● ifInOctets - total number of bytes received by this port including service bytes since the last initialization of the SNMP agent.

● iflnUcastPkts - number of packets with an individual interface address delivered to the top level protocol.

● IflnNUcastPkts - number of packets with a broadcast or multicast interface address, delivered to the top-level protocol.

● ifInDiscards - number of packets received by the interface, but not delivered to the upstream protocol, most likely due to packet buffer overflow or some other reason.

● ifin Errors - number of incoming packets that were not forwarded to up-layer protocol due to detection of errors in them.

Besides objects describing input packets statistics, there are analogous objects, but related to output packets.

As can be seen from the description of objects MIB-II, this database does not provide detailed statistics on the characteristics of errors Ethernet frames, in addition, it does not reflect the change in characteristics over time, which is often of interest to the network administrator.

These limitations were later removed by the new standard on MIB - RMON MIB, which is specifically focused on the collection of detailed statistics on the Ethernet protocol, moreover, with the support of such an important function, as the construction of statistical characteristics over time by the agent.

3.4. Formats and names of SNMP MIB objects

For the naming of MIB base variables and unambiguous definition of their formats an additional specification called SMI - Structure of Management Information is used. For example, the SMI specification includes the standard name IpAddress and defines its format as a string of 4 bytes. Another example is the name Counter, for which the format is defined as an integer between 0 and 232-1.

In describing MIB variables and SNMP formats, the SMI specification is based on the formal language ASN.1, adopted by ISO as a notation for describing communication protocol terms (although many communication protocols, such as IP, PPP, or Ethernet, do without this notation). The notation ASN. 1 serves to establish an unambiguous correspondence between terms taken from standards intended for human use and those data transmitted in communication protocols by hardware. The unambiguity achieved is very important for the heterogeneous environment characteristic of enterprise networks. So, instead of specifying that some protocol variable is an integer, a protocol developer using ASN.1 notation should specify exactly the format and the valid range of the variable. As a result, MIB documentation written with ASN.1 notation can be accurately and mechanically translated into the form of codes specific to protocol messages.

The ASN.1 notation is similar to other meta-languages, such as the normal Backus form used in describing programming languages, in particular Algol. ASN.1 notation supports a basic set of different data types, such as integer, string, etc., and also allows constructing composite data - arrays, enumerations, structures - from these basic types.

There are rules of translation of the data structures described in ASN.1, in the data structures of programming languages, such as C++. Accordingly, there are translators that do this work. Examples of data descriptions using the ASN.1 are given below, when describing the SNMP protocol data blocks.

The ASN.1 notation is widely used in describing many OSI standards, in particular managed object models and CMIP message structures.

MIB variable names can be written in both character and numeric formats. Symbolic format is used to represent variables in text documents and on the display screen,

and numeric names are used in SNMP messages. For example, the symbolic name SysDescr corresponds to the numeric name 1, and more precisely 1.3.6.1.2.1.1.1.

Composite numeric name of the SNMP MIB object corresponds to the full name of the object in the ISO standardization objects registration tree. Developers of the protocol SNMP protocol did not use the traditional for the Internet standards way of fixing the numerical parameters of the protocol in a special RFC, called "Assigned Numbers" (which describes, for example, the numerical values that can take the field Protocol IP packet, etc.). Instead, they registered SNMP MIB objects in the global ISO standards registration tree, shown in Fig. 7.7.

As with any complex system, the ISO object namespace has a tree-like hierarchical structure, with Fig. 7.7 shows only the top part of the tree. From the root of this tree, there are three branches corresponding to standards controlled by ISO, ITU and jointly ISO-ITU. In turn, ISO has created a branch for standards created by national and international organizations (the ogd branch). Internet standards were created under the auspices of the U.S. Department of Defense (Departament of Defense, DoD), so MIB standards fell into a subtree dod-internet, and then, of course, into the network management standards group - the mgmt branch. Objects of any standards created under the auspices of ISO are uniquely identified by composite character names beginning at the root of that tree. In protocol messages, character names are not used, but unambiguously corresponding composite numeric names are used. Each branch of the object name tree is numbered in the tree by integers from left to right, beginning with one, and these numbers replace the character names. ISO object namespace is represented in Figure 3.3.

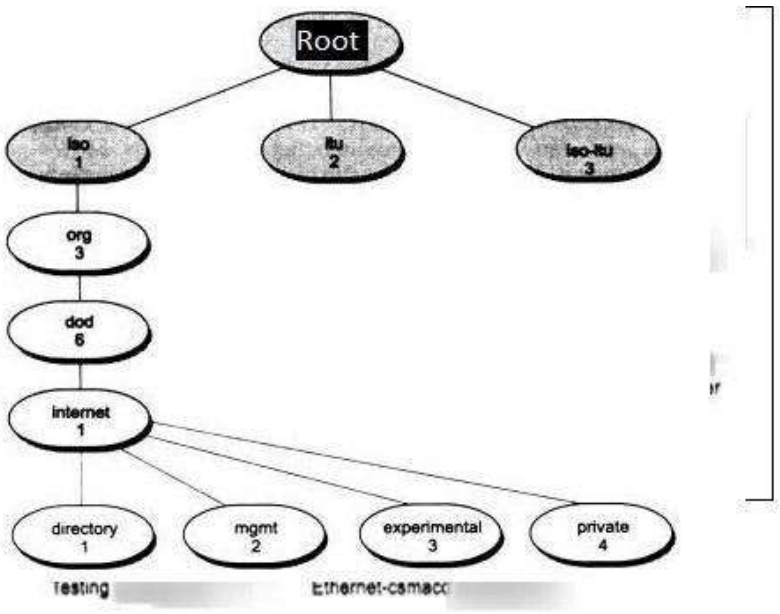

Figure 3.3. ISO object namespace

The group of private objects (4) is reserved for standards created by private companies, such as Cisco, Hewlett-Packard, etc. The same registration tree is used for naming classes of CMIP and TMN objects.

Accordingly, each group of MIB-I and MIB-II objects also has, in addition to the short character names given above, full character names and their corresponding numeric names. For example, the short character name of the System group has the full form iso.org.dod.internet.mgmt.mib.system, and its corresponding numeric name is 1.3.6.1.2.1. The part of the ISO name tree, which includes groups of MIB objects is shown in Figure. 3.4.

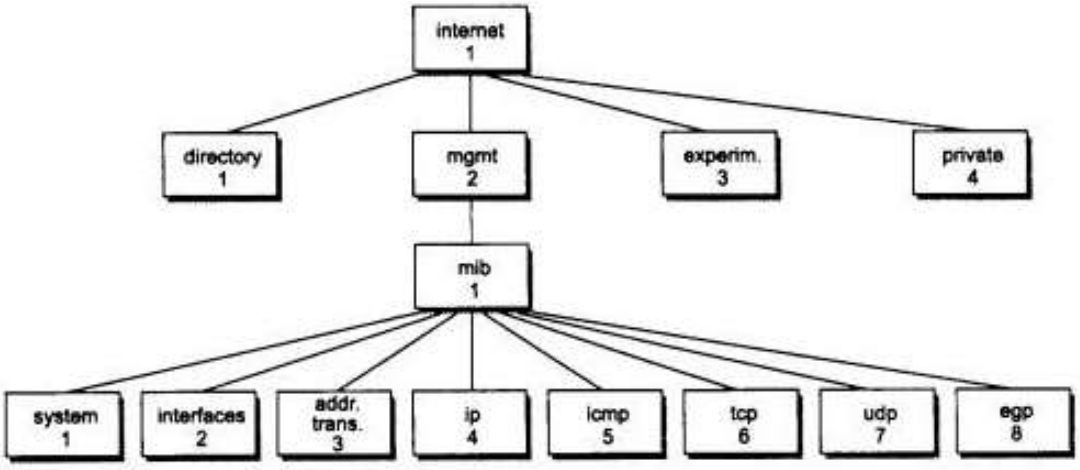

Figure 3.4. Part of the ISO name tree, including groups of MIB-I objects

3.5. SNMP message format

The SNMP protocol serves the transfer of data between agents and the station controlling the network. SNMP uses the UDP datagram transport protocol, which does not deliver messages reliably. The protocol, which organizes the reliable transmission of datagrams based on TCP connections, is very busy for the managed devices, which at the time of development of SNMP protocol were not very powerful, so they decided to abandon the services of TCP.

SNMP is often seen only as a management solution for TCP/IP networks. Although SNMP most often works on UDP (it can also work on TCP), it can also work on the transport network protocols of the OSI stack - TPO, TP4, CNLS, as well as the MAC layer protocols. Support for SNMP in other transport environments is growing. For example, Novell began to support SNMP protocol with NetWare 3.11, and some hardware manufacturers (Bay Networks, for example) implement in their devices transfer SNMP messages using both IP and IPX.

SNMP messages, unlike messages from many other communication protocols, do not have fixed field headers. In accordance with ASN.1 notation, an SNMP message is composed of an arbitrary number of fields and each field is preceded by a description of its type and size.

Any SNMP message consists of three main parts: the version of the protocol (version), the community identifier, used to group the devices managed by a particular manager, and the data area, which actually contains the above-described protocol commands, object names and their values. The data area is divided into Protocol Data Units (PDUs).

The general format of SNMP message in ASN.1 notation looks as follows:>

SNMP-Message ::=

SEQUENCE {

version INTEGER {

version-1 (0)

},

community

OCTET STRING,

SNMP-PDUs

ANY

}

The data area can contain five different types of PDUs corresponding to the five SNMP commands:

SNMP-PDUs :: =

CHOICE {

get-request

GetRequest-PDUs,

get-next-request

GetNextRequest-PDU,

get-response

GetResponse-PDU,

set-request

SetRequest-PDU,

trap

Trap-PDU,

}

Finally, for each type of PDU, there is a definition of its format. For example, the format of GetRequest-PDU is described as follows:

GetRequest-PDU ::=

IMPLICIT SEQUENCE {

request-id

RequestID,

error-status,

ErrorStatus,

error-index

Errorlndex,

variable-bindings

VarBindList

}

Next, the SNMP standard defines the format of GetRequest-PDU block variables accordingly. The variable Request ID is a 4-byte integer (used to match responses to requests), ErrorStatus and Errorlndex are single-byte integers that should be set to 0 in the request. VarBindList is a list of numeric object names whose values the manager is interested in. In ASN.1 notation, this list consists of name-value pairs. The variable value must be set to null when queried.

Here is an example of an SNMP protocol message, which is a request for the value of the SysDescr object (numeric name 1.3.6.1.2.1.1.1). Example of query protocol about the value of the SysDescr object is represented in Figure. 3.5.

30	29	02	01	00			
SEQUENCE	len = 41	INTEGER	len=1	vers = 0			
04	06	70	75	62	6C	69	63
string	len = 6	p	u	b	l	i	c
A0	1C	02	04	05	AE	56	02
getreq	len = 28	INTEGER	len = 4	--------	requested ID	--------	--------
02	01	00	02	01	00		
INTEGER	len = 1	status	INTEGER	len = 1	error	index	
30	0E	30	0C	06	08		
SEQUENCE	len = 14	SEQUENCE	len = 12	objectid	len = 8		
2B	06	01	02	01	01	01	00
1,3	6	1	2	1	1	1	0
05	00						
null	len = 0						

Fig. 3.5. Example of query protocol about the value of the SysDescr object

As you can see from the description, the message starts with code 30 (all codes are hexadecimal), which corresponds to the keyword SEQUENCE (sequence). The length of the sequence is specified in the next byte (41 bytes). This is followed by an integer of 1 byte length, which is the SNMP protocol version (in this case O, that is SNMP v.l, a 1 would mean SNMP v.2). The community field is of typestring (character string) 6 bytes long with

value public. The rest of the message is a GetRequest-PDU data block. That it is a Get-request operation, says the AO code (this value is defined in the SNMP protocol, not in the ASN.1 notation), and the total length of the data block is 28 bytes. According to the Getrequest-PDU block structure, it is followed by the request identifier (defined as a 4-byte integer). The block is followed by two single-byte integers status and error index, which are set to 0 in the request. Finally, the message ends with the list of objects, which consists of one pair: name 1.3.6.1.2.1.1.1.0 and the value null.

3.6. RMON MIB specification

The latest addition to the SNMP functionality is the RMON specification, which provides remote communication with the MIB base. Before RMON, the SNMP protocol could not be used remotely and allowed only local management of devices. The RMON MIB has an improved set of properties for remote management because it contains aggregated device information that does not require large amounts of information to be transmitted over the network. RMON MIB objects include additional packet error counters, more flexible trending and statistical analysis tools, more powerful filtering tools for capturing and analyzing individual packets, and more sophisticated alerting conditions. RMON MIB agents are more intelligent than MIB-I or MIB-II agents and do much of the device information processing previously done by managers. These agents can reside within various communication devices, and can also be made as separate software modules running on general-purpose personal computers and laptops.

The RMON object is assigned number 16 in the set of MIB objects, and the RMON object itself includes 10 groups of the following objects.

- Statistics - current accumulated statistical data on packet characteristics, number of collisions, etc.
- History - statistical data stored at certain intervals for subsequent analysis of trends in their changes.
- Alarms - threshold values of statistical indicators, above which RMON agent sends a message to the manager.

● Hosts - data about hosts on the network, including their MAC addresses.

● HostTopN - a table of the busiest hosts on the network.

● Traffic Matrix - statistics on traffic between each pair of hosts arranged as a matrix.

● Filter - packet filtering conditions.

● Packet Capture - conditions of packet capture.

● Event - conditions of events registration and generation.

These groups are numbered in that order, so for example the Hosts group has a numeric name of 1.3.6.1.2.1.16.4.

The tenth group consists of special objects of Token Ring protocol.

Total standard RMON MIB defines about 200 objects in 10 groups, which are fixed in two documents - RFC 1271 for Ethernet networks and RFC 1513 for Token Ring networks.

A distinctive feature of the RMON MIB standard is its independence from the network layer protocol (in contrast to MIB-I and MIB-II standards, oriented on TCP/IP protocols). Therefore, it is convenient for heterogeneous environments that use different protocols of the network layer.

Let's consider in more detail the Statistics group, which determines what information about Ethernet frames (called packets in the standard) the RMON agent can provide. History group is based on the objects of Statistics group, as its objects simply allow constructing a time series for Statistics group objects.

The Statistics group includes, along with some others, the following objects.

● etherStatsDropEvents - the total number of events in which packets were ignored by the agent due to lack of its resources. The packets themselves were not necessarily lost by the interface.

● etherStatsOrtets - total number of bytes (including erroneous packets) received from network (excluding preamble and including checksum bytes).

● etherStatsPkts - total number of received packets (including erroneous packets).

● etherStatsBroadcastPkts - total number of good packets that were sent to the broadcast address.

● etherStatsMulticastPkts - total number of good packets received by multicast address.

● etherStatsCRCAlign Errors - the total number of packets received which were between 64 and 1518 bytes long (excluding the preamble), did not contain an integer byte (alignment error) or had an invalid checksum (FCS error).

● etherStatsUndersizePkts is the total number of packets that were less than 64 bytes in length but were generated correctly.

● etherStatsOversizePkts - the total number of received packets that were longer than 1518 bytes but were formed correctly nevertheless.

● etherStatsFragments-The total number of packets received that did not consist of an integer number of bytes or had an invalid checksum and were also less than 64 bytes long.

● etherStatsJabbers - the total number of received packets that did not consist of a whole number of bytes or had an invalid checksum and were also longer than 1518 bytes.

● etherStatsCollisions - the best estimate of the number of collisions on a given Ethernet segment.

● etherStatsPkts640ctets - total number of received packets (including bad packets) of 64 bytes.

● etherStatsPkts65to1270ctets - total number of received packets (including bad packets) between 65 and 127 bytes.

● etherStatsPktsl28to2550ctets - total number of packets received (including bad packets) with size from 128 to 255 bytes.

● etherStatsPkts256to5110ctets - total number of packets received (including bad packets) between 256 and 511 bytes.

● etherStatsPkts512tol0230ctets - total number of packets received (including bad packets) between 512 and 1023 bytes.

● etherStatsPktsl024tol5180ctets - total number of packets received (including bad packets) with size from 1024 to 1518 bytes.

As you can see from the description of the objects, with the RMON agent built into the repeater or other communication device, you can perform a very detailed analysis of the

Ethernet or Fast Ethernet segment. First you can get the data on the types of frame errors occurring in the segment, and then it is reasonable to collect the time dependencies of the intensity of these errors (including time-dependent ones) with the help of the Histogram group. After the analysis of time dependencies it is often possible to make some preliminary conclusions about the source of erroneous frames and on this basis to formulate more subtle conditions for capturing frames with specific features (by setting conditions in theFilter group), corresponding to the put forward version. After that it is possible to carry out even more detailed analysis by examining the captured frames, extracting them from the objects of thePackage Capture group.

Later, the RMON 2 standard was adopted, which extends the ideas of the intelligent RMON MIB to upper layer protocols, doing some of the work of protocol analyzers.

3.7. Disadvantages of SNMP

SNMP protocol is the basis for many management systems, although it has several fundamental disadvantages, which are listed below.

The lack of means for mutual authentication of agents and managers. The only means, which could be classified as a means of authentication, is the use of the so-called "community string" in messages. This string is transmitted over the network in an open form in an SNMP message, and serves as the basis for dividing agents and managers into "communities," so that the agent communicates only with those managers who specify the same character string in the community string field as the string stored in the agent's memory. This is certainly not a way to authenticate, but a way to structure agents and managers. SNMP v.2 version was supposed to eliminate this drawback, but as a result of disagreement among the developers of the standard, new authentication features appeared in this version, but as optional.

Working through unreliable UDP protocol (which is how the vast majority of SNMP agent implementations work) leads to loss of alarm messages (trap messages) from agents to managers, which can lead to poor management. Correcting the situation by switching to a reliable connectivity transport protocol risks losing communication with the

huge number of built-in SNMP agents present in the equipment installed in the networks. (The CMIP protocol originally runs on top of a reliable OSI stack transport and does not suffer from this disadvantage). Management platform developers are trying to overcome these shortcomings. For example, the HP 0V Telecom DM TMN platform, a platform for developing multilevel management systems in accordance with TMN and ISO standards, operates a new implementation of SNMP that arranges reliable messaging between agents and managers by independently arranging SNMP message retransmissions in case of loss.

3.8. OSI Management Standards with SNMP

The OSI network management model, OSI Management Framework, is defined in ISO/IEC 7498-4: Basic Reference Model, Part 4, Management Framework, which is an evolution of the general seven-layer open systems communication model for when one system controls another.

- ISO/IEC 7498-4 consists of the following main sections.
- Terms and general concepts.
- The systems management model.
- The information model.
- Functional areas of systems management.
- Structure of systems management standards.

The functional areas of systems management have already been discussed in Section 7.1 as having general relevance for any management system.

The ISO management standards use terminology that is partly the same as, and partly different from, the SNMP management systems terminology.

As shown in Fig. 7.9, the exchange of management information using the Management Protocol takes place between Systems Management Application Entities (SMAEs). SMAE Entities are located at the application layer of the seven-layer OSI model and are elements of a management service. An SMAE in the OSI model is defined as a currently active protocol element of any layer that participates in the communication. Examples of SMAEs are agents and control system managers.

The exchange of management information using the Management Protocol takes place between Systems Management Application Entities (SMAEs). SMAE Entities are located at the application layer of the seven-layer OSI model and are elements of a management service. An SMAE in the OSI model is defined as a currently active protocol element of any layer that participates in the communication. Examples of SMAEs are agents and control system managers.

The definitions of agent and manager functions in OSI standards agree quite well with the definitions of SNMP systems, with some exceptions in terminology. Messages that an agent sends to a manager on its own initiative are called notifications.

For example, if some network element X fails, the manager needs to update its network configuration database. Element X, which is a managed object for the management system, can send a notification to the agent. Element X may be on the same managed system as the agent, or it may be on a different system. In turn, the agent sends a notification to the manager that element X has failed. In accordance with this notification, the manager updates the configuration database.

The manager not only collects and correlates data from agents, based on this data, he can also perform administrative functions, managing the operations of remote agents.

In OSI standards, the boundaries between managers and agents are not very clear. An SMAE entity that acts as a manager in one interaction may act as an agent in another interaction, and vice versa.

The OSI standards do not specify how an agent interacts with managed objects. The OSI standards also do not specify how the agent interacts with managed objects that are outside the managed system, i.e., objects that need to be communicated with via the network. In such cases, it may be necessary, for example, for one agent to request data about some object from another agent. The order of this kind of interaction is also not defined by OSI standards.

For a manager and an agent to be able to communicate, each must have some knowledge of the other. The OSI model calls this knowledge the Application Context (AC). AC describes the elements of the application layer of the OSI stack that are used by agents and managers.

The application layer of the OSI stack includes several general-purpose helper services that are used by application protocols and user applications (including management applications) to automate the most commonly performed actions. These are not complete application layer protocols like ftp, telnet, or NCP with which a network user can perform some useful action, but rather support system functions that help the developer of an application protocol or application write his program compactly and efficiently. At the application layer of the OSI stack there are the following helper services.

ACSE (Association Control Service Element). Is responsible for establishing connections between applications on different systems. A connection (session) in the OSI Application Layer is called an association. Associations can be individual or group (shared).

RTSE (Reliable Transfer Service Element). Supports restoration of the dialogue caused by a break in the underlying communication services within an association.

ROSE (Remote Operations Service Element). It organizes the execution of software functions on remote machines (analogous to the RPC remote procedure call service).

The CMIP protocol used in the OSI standards for communication between managers and agents, and software implementations of managers and agents make extensive use of these auxiliary services, especially the ROSE service for remote procedure calls.

The basic OSI management model includes: systems management, N-level management, and N-level operations. This division into three areas is made to account for all possible management situations.

Systems management deals with manageable objects at all seven OSI layers, including the application layer. It is based on the reliable transfer of control information between the end systems with the establishment of a connection. It must be stressed that the OSI management model does not permit the use of services without connectivity.

N-layer control is limited to managed objects of some particular layer of the seven-layer model. The control protocol uses the communication protocols of the layers below it. N-layer control is useful when not able to use all seven layers of the OSI. In this case, it is acceptable to use an N-layer control protocol which is strictly dedicated to that layer.

Examples of layered control protocols are the LAN control protocols developed by the IEEE (SMT of FDDI technology), which are limited to Layer 1 and Layer 2.

Finally, N-layer operations are reduced to monitoring and control based on the control information contained in the communication protocols of only this layer. For example, the network monitoring data contained in the STM-n frames of SDH technology refers to N-layer operations, namely the physical layer.

N-layer management standards and N-layer operations are not part of the OSI suite of management standards. The OSI standards consider only systems management using the full seven-layer stack.

The basic systems management model involves control operations and notification transfer between peer systems, which means that there is not necessarily a rigid allocation of roles to controlling and managed systems. This model facilitates the implementation of distributed aspects of management. On the other hand, peer systems are allowed to be implemented as managed and managed systems.

A managed object is OSI's representation of a resource for management purposes. A resource can be described as a managed object. A specific managed object is an instance of some class of managed objects. The OSI management model uses an object-oriented approach extensively. A class of managed objects is a set of properties that may be mandatory or conditional. By describing one class of managed objects, such as switches, you can create another class of managed objects, such as switches that support VLAN techniques, by inheriting all the properties of the switch class but adding new attributes.

To manage resources, the manager and agent must be aware of the details of those resources. The details of the representation of managed objects that are required to perform management functions are stored in a repository known as the Management Information Base (MIB). The OSI MIB not only stores descriptions of managed object classes, but also characteristics of the network and its elements. The MIB contains the characteristics of each piece of managed equipment and resources. The MIB also includes descriptions of actions that can be performed based on the data collected or that can be invoked by external commands. MIBs allow external systems to query, modify, create and delete managed

objects (while real network resources continue to operate, of course). The CMIP protocol and local management interfaces provide access to these capabilities.

The MIB is a conceptual model, and it has no relationship to the way data is physically or logically stored in the resource. The standards do not define aspects of data storage itself. OSI protocols define the syntax of the information stored in the MIB and the semantics of data exchange.

A large management system usually consists of a large number of agents and managers. In order to organize automatic interaction between managers and agents, it is necessary to somehow specify data containing the characteristics of agents and managers. A manager needs to know which agents are working in the management system, their names and network addresses, the classes of managed objects they support, etc. The agent also needs similar information about the managers, since it needs to send notifications on its own initiative and respond to managers' requests.

This data is called shared management knowledge between the manager and the agent in the OSI model. (In SNMP systems the organization of this data is not standardized and each particular management system stores this data in an individual form.)

The shared management knowledge must be known before the association between agent and manager is established. They are usually stored in some file or distributed database and are queried each time an association is established. During the establishment of an association, shared control knowledge is exchanged.>

Various aspects of control knowledge organization and access are standardized in OSI. Following the object-oriented approach has led to the use of special system objects to store this knowledge.

Standard ISO 10164-16.2 defines a model of control knowledge objects and classes of such objects. In addition, functions for working with controlling knowledge are defined.

There are three types of control knowledge and, accordingly, three types of objects that describe this knowledge.

Repertoire Knowledge describes the capabilities of the managed system, which include a list of supported classes of managed objects, supported control functions and

naming functions. Repertoire knowledge helps the manager identify the capabilities of the managed systems without accessing them.

Definition Knowledge includes formal descriptions of managed object classes, test categories, relationship classes, and definitions of control information understood by the managed system.

Instance Knowledge provides information about specific instances of managed objects present in the managed system.

Using tree-like databases to store control knowledge

The control system must store knowledge about supported object classes and spawned object instances in some form that is convenient for providing control system modules with access to this information. The OSI control architecture provides several database schemes about managed objects and their classes. These schemas are usually called trees because of the hierarchical organization of the information. There are the following trees.

Inheritance Tree, also called a registration tree. It describes the relationship between base and derived classes. A subclass inherits all the characteristics of a superclass and augments them with specific extensions (additional attributes, behaviors and actions). The OSI object classes are registered in the same tree as the Internet MIB objects. The inheritance tree may be global, with the root representing the whole world, or local, with the root corresponding to the top level objects of the organization or network. All managed OSI objects must be registered in a global ISO tree (in which MIB-I, MIB-II, RMON MIB objects of the SNMP standard are registered). Objects representing international standards are registered in the international branch of the tree, and private models developed by control system vendors are registered in tree branches beginning with the private branch.

Containment Tree. Describes the inclusion relationships of controlled objects of a real system.

The naming tree defines how objects are named in the control system. OSI objects can have several types of names: Relative Distinguished Name (RDN), Distinguished Name (DN), sometimes called Full Distinguished Name (FDN), and Local Distinguished Name (LDN). These names are related to the inclusion tree, because they define object names in

relation to the objects that include them. A relative name, RDN, corresponds to a short name that uniquely identifies an object among many other objects subordinate to the same parent object. For example, the name interface_a is an RDN that uniquely characterizes an object among the objects subordinate to the object node_a. A complete FDN distinguished name is a sequence of RDN names beginning at the top of a global name tree, that is, a tree describing some global network. Finally, the local distinctive name is a sequence of RDN names, but starting not at the global root, but at the root of the local control system name tree, which is responsible for part of the global name tree of the network.

The name tree is usually combined with an inclusion tree.

An example of an inclusion tree is shown in Fig. 7.10. An instance of a managed object of the sogr-sops class (corporate hub) has a name B1, and a max-slots attribute that describes the maximum number of slots of that hub class, equal in this case to 14. This object includes a number of other objects: objects of class repeater, switch and RAS, which in turn include objects of type interface, describing the ports of the concentrator modules. Example of an inclusion tree is represented in Figure 3.6.

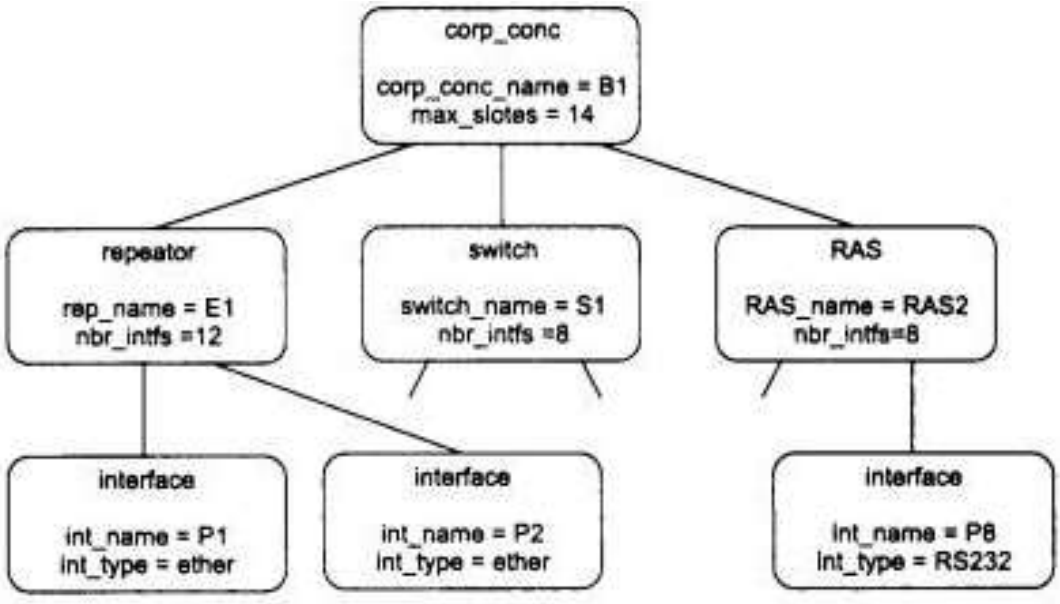

Fig. 3.6. Example of an inclusion tree

The object class name allows you to refer to the class description and find out the full list of attributes of that class or a reference to a parent class that inherits all or some of its attributes. An object instance name gives information about a specific module or interface belonging to a specific communication device, for example, the name B1.E1.P2 defines the second port of the E1 repeater module that is part of the B1 enterprise hub.

Conclusions on Part 3

In practice today, there are two families of standards for network management - Internet standards, based on SNMP (Simple Network Management Protocol), and international standards ISO/ITU-T, using as a protocol of interaction between agents and managers protocol CMIP (Common Management Information Protocol).

SNMP is used to obtain information from network devices about their status, performance, and other characteristics, which are stored in the Management Information Base (MIB).

An agent in the SNMP protocol is a processing element that provides access to the values of the MIB variables for the managers located at the network control stations, thus allowing them to implement control and monitoring functions for the device.

The main management operations are placed in the manager, and the SNMP agent plays mostly a passive role, transmitting to the manager at its request the values of the accumulated statistical variables.

SNMP is often seen only as a management solution for TCP/IP networks. Although SNMP most often works on UDP (it can also work on TCP), it can also work on the transport network protocols of the OSI stack - TPO, TP4, CNLS, as well as the MAC layer protocols.

PART 4 IMPLEMENTATION OF EFFICIENCY ANALYSIS OF WIRELESS NETWORKS

4.1. Overview

Network environments are not static. Far from it, they are probably one of the most volatile parts of our entire infrastructure. By definition, a network connects the different parts together, constantly exchanging forward and backward. There are many moving parts that can cause our network to shut down in the expected way: hardware failure, software bugs, personnel errors, regardless of their best intentions, and so on. I need a way to make sure that my network environment works as expected and that I am successfully notified if something goes wrong.

In the next chapters I will look at different ways to perform network monitoring tasks. Many of the tools I have considered so far can be linked together or directly managed from Python. Like everything I have considered so far, network monitoring must work with two parts. First, I need to know what information my hardware can transmit. Second, I need to determine what useful information I can interpret from it.

I will look at a number of tools that will allow us to effectively monitor my network environment:

- Simple Network Management Protocol (SNMP);
- Matplotlib and pygal visualization;
- MRTG and Cacti.

This list is not exhaustive and there is no doubt that there is no shortage of commercial vendors in this segment. However, the basics of network monitoring that I will cover are well served by both open source and commercial tools.

4.2. Installation of the laboratory

There is illustration of simple network architecture in Figure 4.1.

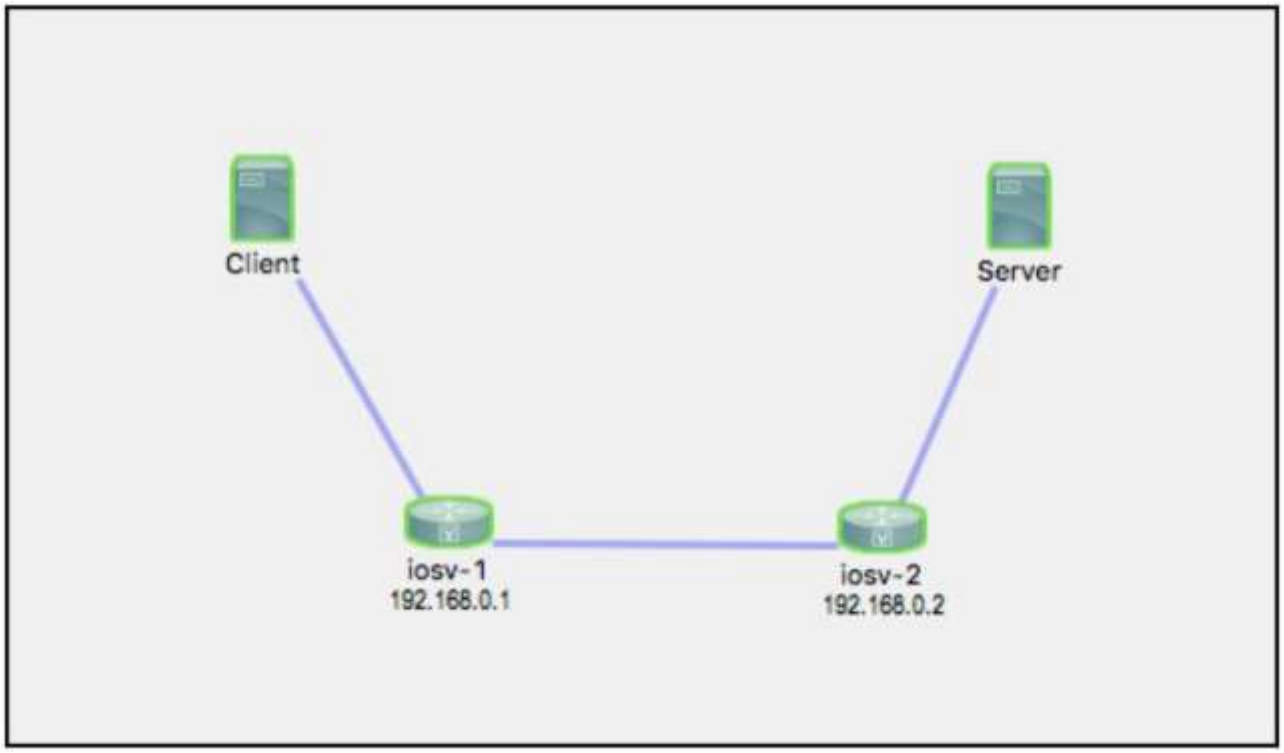

Fig. 4.1. Simulation explanation

To create the exchange in my network environment, two existing Ubuntu hosts will be used so that I can observe some non-zero counters.

SNMP is a standard protocol used for data collection and device management. Although this standard allows to use SNMP for device management, in my practice most network administrators prefer to leave SNMP only as a mechanism for collecting some information. Since SNMP works on top of UDP, which does not establish a connection and is considered to be a relatively weak mechanism in versions 1 and 2, making changes using SNMP usually complicates the work of network administrators. Version 3 of SNMP has added cryptographic security and new concepts and terminology to this protocol, but the way it adapts is different for network device manufacturers.

SNMP is widely used for monitoring network environments and was introduced in 1988 as part of RFC 1065. All operations are straightforward with the network dispatcher sending GET and SET requests to the device, and the device itself with an existing SNMP agent responds to certain information on the request. The most widely used standard is

SNMPv2c, which is defined in RFC 1901 - RFC 1908. It applies some simple community based security scheme for preservation. It has also introduced some new features such as the ability to obtain large information arrays. The scheme below shows the available top-level operations for SNMP. Available top-level operations example for SNMP is represented in Figure 4.2.

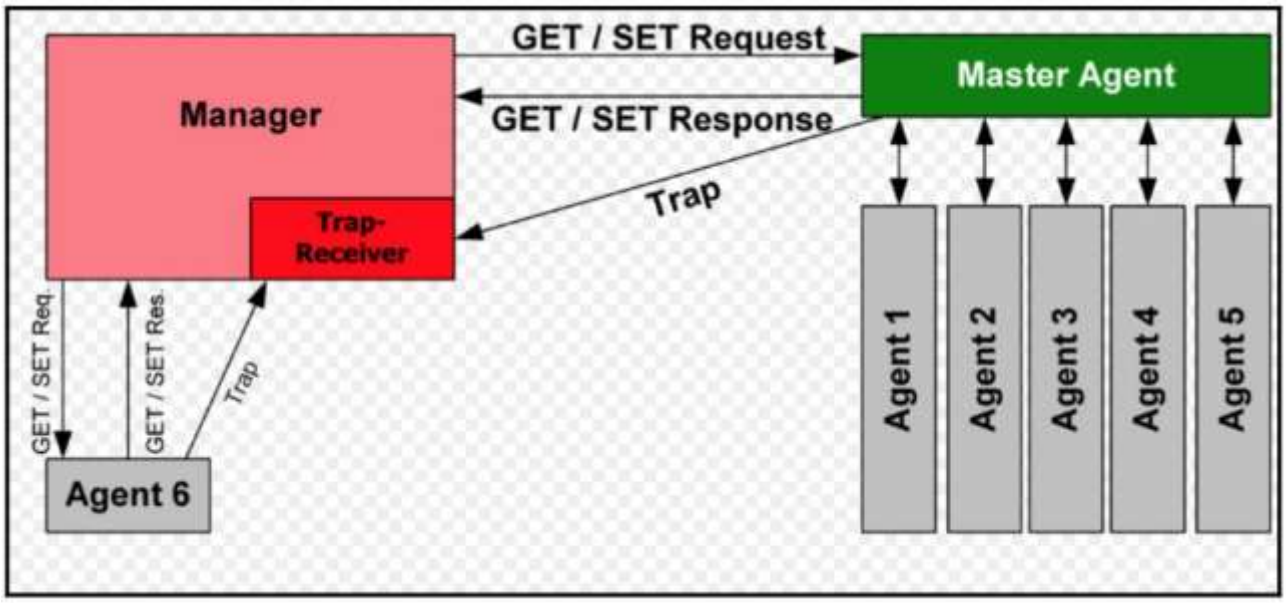

Fig. 4.2. Available top-level operations for SNMP

All information located in a certain device is structured as a Management Information Base (MIB). This MIB applies a hierarchical namespace containing an Object ID (OID, Object IDentifier) which represents information that can be read and returned to the requesting party. When I discuss the application of SNMP to request device information, I'm actually talking about using the management state for a specific OID that represents this information later. I will need to make some effort to combine basic general information into a single OID structure; however, the final outcome of all my efforts depends on how successfully these terms have been defined. At least based on my experience, I usually have to check with the vendor documentation how to define the OID I need.

Some of the fundamental things that will be able to do these things:

The implementation itself is largely dependent on the amount of information that the device agent itself can provide. This, in turn, depends on how the manufacturer interprets SNMP: as a main function or as an add-on function.

SNMP agents usually need CPU time from the control mechanism to return some value. Not only is this not efficient for devices with, say, large BGP tables, but it is also impossible to apply it to repeated data requests.

The user himself needs to know a certain OID to be defined in the data query.

4.3. Installation of SNMP

First let's make sure that the managed SNMP device and agent itself works in my installation. The SNMP packet itself can be installed either on an existing host in my example or on a managed device in my network management environment. As long as my dispatcher has IP reach to a particular device and the managed device itself allows connection to it, SNMP must work properly.

In my configuration I have installed SNMP on both the Ubuntu host and the managed network environment, as well as on a specific client host in my lab for security checks:

sudo apt-get install snmp

There are many optional parameters that I can configure in my network device, such as my contact, location, case ID and SNMP packet size. These options are device specific and I should check the available documentation for my device. For IOSv devices I will set up an access list to restrict only the host that is desirable to poll the device itself and link the access list to the SNMP community string. In my case I will apply the word secret to the community string and permit_snmp as an access name to the list:

!
ip access-list standard permit_snmp
permission 172.16.1.173 log
deny any log

!

!

snmp-server community secret RO permit_snmp

!

The SNMP community string acts as some kind of shared password for the dispatcher and the agent, it must be enabled whenever I want to poll the device.

I can use tools such as the available MIB locator (http://tools.cisco.com/ITDIT/MIBS/servlet/index) to find a specific OID to query it. Alternatively I just have to walk through the entire SNMP tree, starting with the main node of the Cisco corporate tree in .1.3.6.1.4.1.9:

$ snmpwalk -v2c -c secret 172.16.1.189 .1.3.6.1.4.1.9

iso.3.6.1.4.1.9.2.1.1.0 = STRING: "

Bootstrap program is IOSv

"

iso.3.6.1.4.1.9.2.1.2.0 = STRING: "reload"

iso.3.6.1.4.1.9.2.1.3.0 = STRING: "iosv-1"

iso.3.6.1.4.1.9.2.1.4.0 = STRING: "virl.info"

...

$ snmpwalk -v2c -c secret 172.16.1.189 .1.3.6.1.4.1.9.2.1.61.0

iso.3.6.1.4.1.9.2.1.61.0 = STRING: "cisco Systems, Inc.

170 West Tasman Dr.

San Jose, CA 95134-1706

U.S.A.

Ph +1-408-526-4000

Customer service 1-800-553-6387 or +1-408-526-7208

24HR Emergency 1-800-553-2447 or +1-408-526-7209

Email Address tac@cisco.com

World Wide Web http://www.cisco.com"

The last thing to be checked should be to make sure that the access list itself will deny unwanted SNMP requests. Since we have the keyword log for both permission and deny elements, only 172.16.1.173 is allowed to query this device:

**Mar 3 20:30:32.179: %SEC-6-IPACCESSLOGNP: list permit_snmp permitted 0 172.16.1.173 -> 0.0.0.0, 1 packet*

**Mar 3 20:30:33.991: %SEC-6-IPACCESSLOGNP: list permit_snmp denied 0 172.16.1.187 -> 0.0.0.0, 1 packet*

As I can see, the biggest challenge when installing SNMP is finding the correct OID. Some of the available OIDs are defined in the MIB-2 standard; others are located in a part of their corporate tree. However, the best hint is the manufacturer's documentation. There are many tools available that can help me, such as the MIB Browser; I can add MIBs (again provided by the manufacturers) to this browser and see the description of all OIDs based on corporate addressing. A tool like Cisco's SNMP Object Navigator turns out to be very valuable when I need to find the correct OID for the object I'm looking for.

4.4. PySNMP

PySNMP is a portable SNMP mechanism on pure Python implemented by Ilya Etingof development (https://github.com/etingof). It abstracts a lot of SNMP details for me as the greatest libraries do and supports both Python 2 and Python 3.

PySNMP requires a PyASN1 package. According to Wikipedia: "ASN.1 is a kind of standard and notation system that describes rules and structures for representing, encoding transmission and decoding data in telecommunications and computing networks. - https://asn1js.org/".

PyASN1 usually provides some kind of Python wrapper over ASN.1. Let's install this package first:

```
cd /tmp
git clone https://github.com/etingof/pyasn1.git
cd pyasn1/
sudo python3 setup.py install
```

Then we will install the PySNMP packet itself:

git clone https://github.com/etingof/pysnmp

cd pysnmp/

sudo python3 setup.py install

Let's take a look at how PySNMP requests the same contact information we used in the previous example, which was slightly modified from the PySNMP example at http://pysnmp.sourceforge.net/faq/response-values-mib-resolution.html. I import all the necessary modules and first I create a certain CommandGenerator object:

>>> from pysnmp.entity.rfc3413.oneliner import cmdgen

>>> cmdGen = cmdgen.CommandGenerator()

>>> cisco_contact_info_oid = "1.3.6.1.4.1.9.2.1.61.0".

I can execute SNMP using the getCmd command. The result is unpacked into various variables; I are primarily concerned with varBinds, which contains the query result itself:

>>> errorIndication, errorStatus, errorIndex, varBinds = cmdGen.getCmd(

... cmdgen.CommunityData('secret'),

... cmdgen.UdpTransportTarget(('172.16.1.189', 161)),

... cisco_contact_info_oid

...)

>>> for name, val in varBinds:

... print('%s = %s' % (name.prettyPrint(), str(val)).

...

SNMPv2-SMI::enterprises.9.2.1.61.0 = cisco Systems, Inc.

170 West Tasman Dr.

San Jose, CA 95134-1706

U.S.A.

Ph +1-408-526-4000

Customer service 1-800-553-6387 or +1-408-526-7208

24HR Emergency 1-800-553-2447 or +1-408-526-7209

Email Address tac@cisco.com

World Wide Web http://www.cisco.com

>>>

It should be noted that the obtained response values are PyASN1 objects. The existing prettyPrint() method converts some of these values into a human-readable format, but since in my case not all results can be converted, I will perform the conversion manually.

I can put some script applied in the previous interactive example in pysnmp_1.py with a check for corresponding errors or I will encounter problems. I can also include many OIDs in my getCmd() method:

```
system_up_time_oid = "1.3.6.1.2.1.1.3.0".
cisco_contact_info_oid = "1.3.6.1.4.1.9.2.1.61.0".
errorIndication, errorStatus, errorIndex, varBinds = cmdGen.getCmd(
 cmdgen.CommunityData('secret'),
 cmdgen.UdpTransportTarget(('172.16.1.189', 161)),
 system_up_time_oid,
 cisco_contact_info_oid
)
```

In the following example, I will save all values obtained from my query so that I can perform other functions, such as visualization, for the obtained data.

As a quick check, I can illustrate the correspondence of all interfaces to the devices:

```
$ snmpwalk -v2c -c secret 172.16.1.189 .1.3.6.1.2.1.2.2.1.2
iso.3.6.1.2.1.2.2.1.2.1 = STRING: "GigabitEthernet0/0"
iso.3.6.1.2.1.2.2.2 = STRING: "GigabitEthernet0/1"
iso.3.6.1.2.1.2.2.1.2.3 = STRING: "GigabitEthernet0/2"
iso.3.6.1.2.1.2.2.1.2.4 = STRING: "Null0"
iso.3.6.1.2.1.2.2.1.2.5 = STRING: "Loopback0"
```

According to the documentation I can match the values of ifOutOctets(10), ifInUcastPkts(11), ifOutOctets(16) and ifOutUcastPkts(17). By quickly checking GigabitEthernet0/0 in the existing OID 1.3.6.1.2.1.2.2.1.17.1 I can compare the command line and SNMP values. These values must be close, but not exactly equal as there can be some exchange in the wires:

 # Command line output

iosv-1#sh int gig 0/0 | i packets

5 minute input rate 0 bits/sec, 0 packets/sec

5 minute output rate 0 bits/sec, 0 packets/sec

38532 packets input, 3635282 bytes, 0 no buffer

53965 packets output, 4723884 bytes, 0 underruns

 # SNMP output

$ snmpwalk -v2c -c secret 172.16.1.189 .1.3.6.1.2.1.2.1.17.1

iso.3.6.1.2.1.2.1.17.1 = Counter32: 54070

When I put the solution into commercial operation, I will record all query results in a database.

In order to simplify the example, I will record the query values in a simple file. In pysnmp_3.py I have defined the different OIDs that I need to query:

 # Hostname OID

system_name = '1.3.6.1.2.1.1.5.0'.

 # Interface OID

gig0_0_in_oct = '1.3.6.1.2.1.2.1.10.1'.

gig0_0_in_uPackets = '1.3.6.1.2.1.2.1.11.1'.

gig0_0_out_oct = '1.3.6.1.2.1.2.1.16.1'.

gig0_0_out_uPackets = '1.3.6.1.2.1.2.1.17.1'.

I use these values in my snmp_query() function together with the host, community and OID names at the input:

```
def snmp_query(host, community, oid):
    errorIndication, errorStatus, errorIndex, varBinds = cmdGen.getCmd(
    cmdgen.CommunityData(community),
    cmdgen.UdpTransportTarget((host, 161)),
    oid
    )
```

All these values are placed in a certain dictionary with different keys and written to a file called results.txt:

result = {}

result['Time'] = datetime.datetime.utcnow().isoformat()

result['hostname'] = snmp_query(host, community, system_name)

result['Gig0-0_In_Octet'] = snmp_query(host, community, gig0_0_in_oct)

result['Gig0-0_In_uPackets'] = snmp_query(host, community, gig0_0_in_uPackets)

result['Gig0-0_Out_Octet'] = snmp_query(host, community, gig0_0_out_oct)

result['Gig0-0_Out_uPackets'] = snmp_query(host, community, gig0_0_out_uPackets)

with open('/home/echou/Master_Python_Networking/Chapter7/results.txt', 'a') as f:

f.write(str(result))

f.write('\n')

There will be a certain file with the results displaying all numbers presented at the moment of my request:

 # Example of output

$ cat results.txt

{'hostname': 'iosv-1.virl.info', 'Gig0-0_In_uPackets': '42005', 'Time':

'2017-03-06T02:11:54.989034', 'Gig0-0_Out_uPackets': '59733',

'Gig0-0_In_Octet': '3969762', 'Gig0-0_Out_Octet': '5199970'}.

I can make this script executable and connect cron to the schedule of jobs to be executed every 5 minutes:

$ chmod +x pysnmp_3.py

 # Crontab configuration

**/5 * * * /home/echou/Master_Python_Networking/Chapter7/pysnmp_3.py*

As already mentioned, in an industrial implementation environment, I would put all information into a certain database. In a NoSQL database I can use time as an index (or key) because it is always unique, followed by different key-value pairs.

I wait until the script is executed several times and move on to how I can use Python to visualize the data.

Python visualization

I collect network data for my own purposes to obtain insight into my network environment. One of the best ways to know what the available data means is to visualize it with charts. This is true for almost any data, but is especially true for time sequences of data in the context of network monitoring. How much data has been transmitted over my wires in the most recent week? What percentage was the TCP protocol in all traffic? These are the values I can carefully select with data acquisition mechanisms such as SNMP and visualize with some popular Python libraries.

In this section I will use the data I collected in the last section of SNMP and apply two popular Python libraries - Matplotlib and Pygal - for their graphical representation.

4.5. Matplotlib

Matplotlib (matplotlib.org) is some rendering library for the main Python library and its mathematical extension NumPy. It can provide typographic-quality drawings such as charts, histograms, and bar graphs in a few lines of code.

5. Installation

The whole installation can be done using a specific Linux package management system, depending on my distribution:

$ sudo apt-get install python-matplotlib

$ sudo apt-get install python3-matplotlib

First example Matplotlib

For the following default examples, all images are displayed as a standard output; it will be easier if I try to output them on the standard output. If I have followed this book with some virtual machine, it is recommended that I use a Windows VM instead of SSH. If I do not have access to the required standard output, I can save all the pictures and view them after they are uploaded. Note that I will need to set the required $DISPLAY variable in some of the charts below.

A line chart is simply defined by two lists of numbers that refer to values in the x axis and in the y axis:

>>> import matplotlib.pyplot as plt

>>> plt.plot([0,1,2,3,4], [0,10,20,30,40]).

[<matplotlib.lines.Line2D object at 0x7f932510df98>].

>>> plt.ylabel('Something on Y')

<matplotlib.text.Text object at 0x7f93251546a0>

>>> plt.xlabel('Something on X')

<matplotlib.text.Text object at 0x7f9325fdb9e8>

>>> plt.show()

This graph will display a certain line as Matplotlib line chart in Figure 4.3.

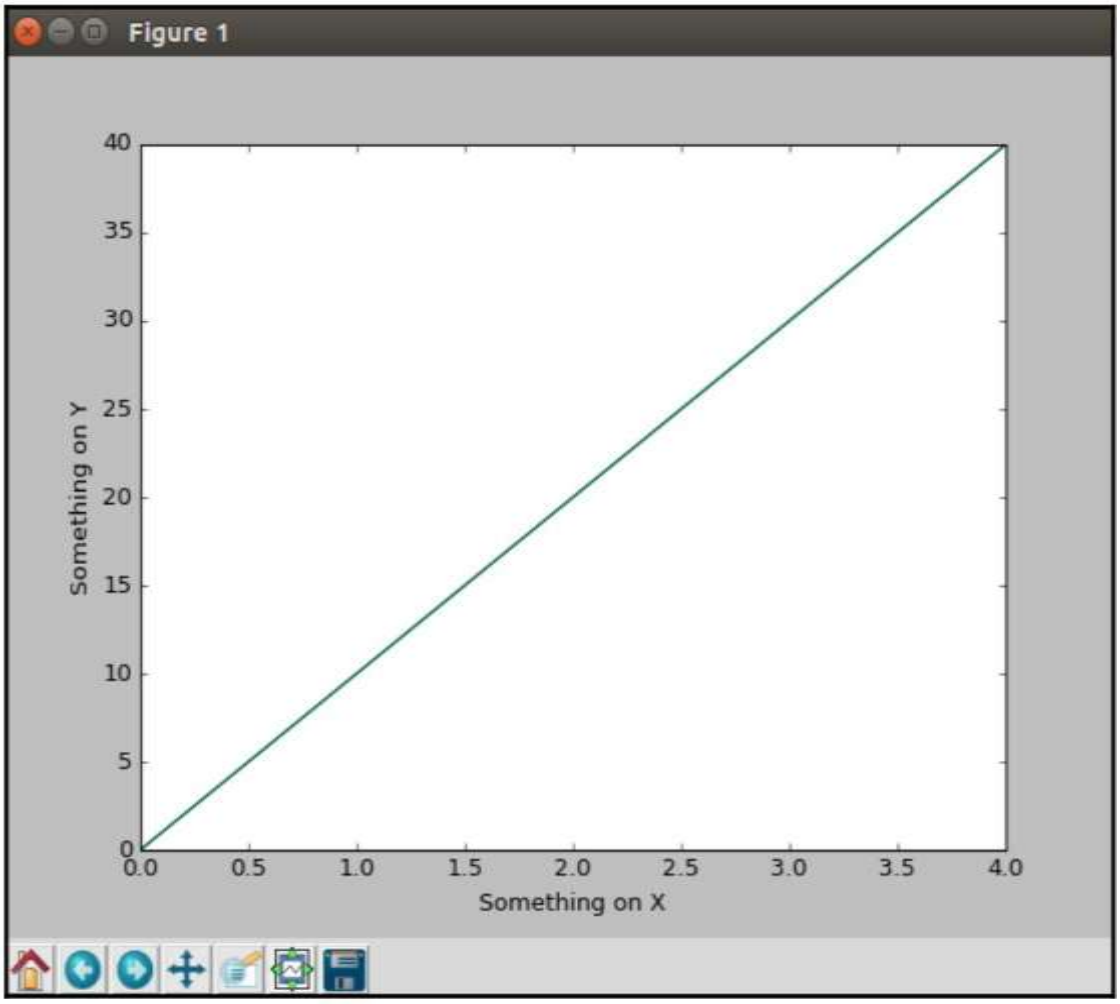

Figure 4.3. Matplotlib line chart

Alternatively, if I don't have access to the default output, or if I have to save this image first, I can use the savefig() method:

>>> plt.savefig('figure1.png').

or

>>> plt.savefig('figure1.pdf')

With this basic knowledge of drawing charts, I can now display the graphical results that I get from SNMP requests.

Matplotlib results for SNMP

In my first example, namely matplotlib_1.py, I import the required dates module inside the pyplot. I will use the matplotlib.dates module instead of the data module in the standard Python library because of the way I need it to be used, which is how Matplotlib will internally convert all data values into floating point values.

```
import matplotlib.pyplot as plt
import matplotlib.dates as dates
```

I will create two empty lists, each of which represents the values in the x axis and in the y axis. Note that on line 12 I use the built-in Python eval() function to read my input as a dictionary instead of some default line:

```
x_time = []
y_value = []
with open('results.txt', 'r') as f:
    for line in f.readlines():
        line = eval(line)
        x_time.append(dates.datestr2num(line['Time']))
        y_value.append(line['Gig0-0_Out_uPackets']).
```

To read the x-axis values in a human-readable data format, I need to use the plot_date() function instead of plot(). I will also change the existing drawing size slightly and expand all x-axis values so that I can read all values as a whole:

```
plt.subplots_adjust(bottom=0.3)
plt.xticks(rotation=80)
plt.plot_date(x_time, y_value)
plt.title('Router1 G0/0')
plt.xlabel('Time in UTC')
plt.ylabel('Output Unicast Packets')
plt.savefig('matplotlib_1_result.png')
```

plt.show()

My final result will be displayed in Router1 Gig0/0 Unicast Output Packet as is represented in Figure 4.4.

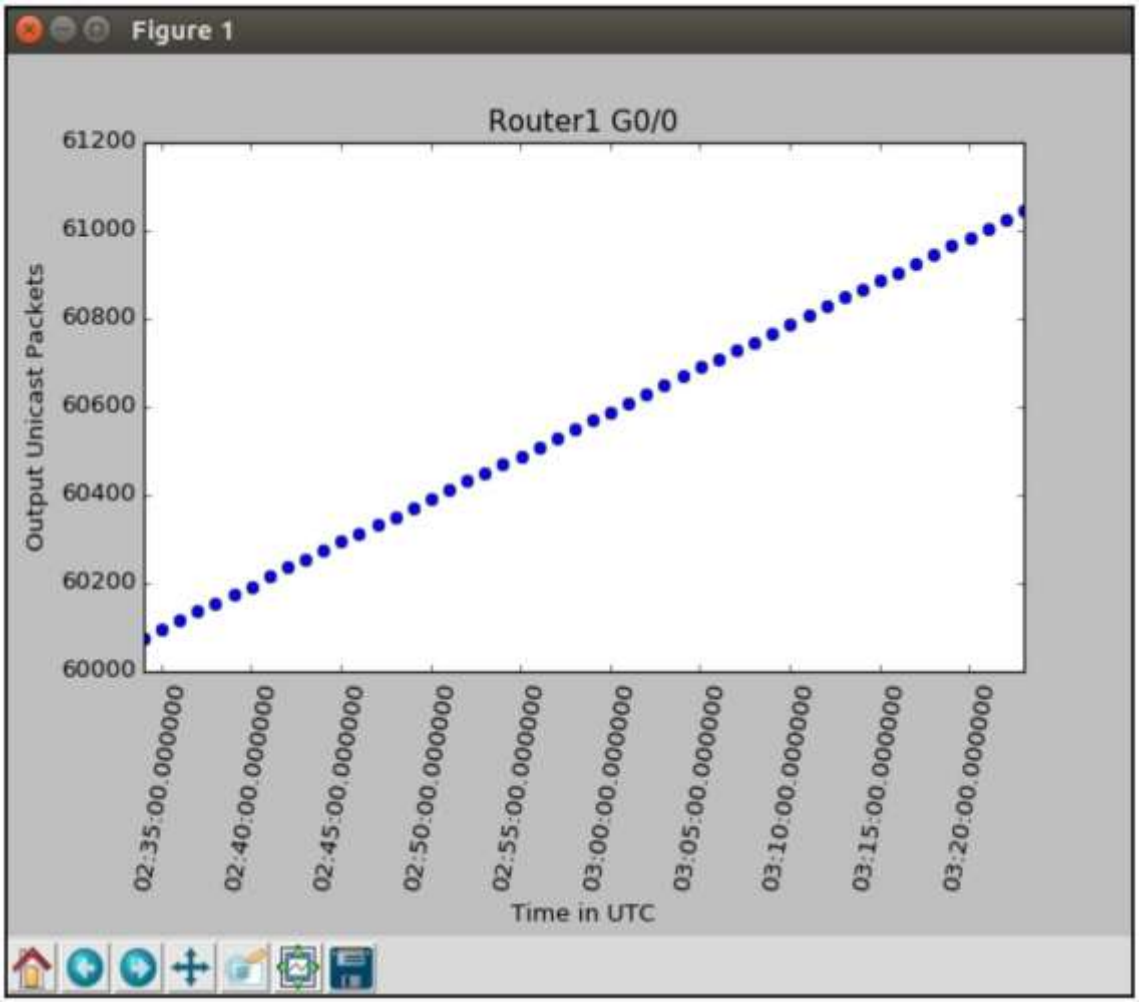

Fig. 4.4. Schedule Router1 Matplotlib

Note that if I prefer a solid line over points, I can use the third optional parameter in plot_date():

plt.plot_date(x_time, y_value, "-")

I can repeat all the steps for the remaining values to output octets, unidirectional incoming packets, and input as separate graphs. However, in my next example, which is matplotlib_2.py, I will show how to draw many graphical values within the same time range, in addition to the Matplotlib options.

In this case I will create additional lists and fill them with the corresponding values:

x_time = []

out_octets = []

```
out_packets = []
in_octets = []
in_packets = []
with open('results.txt', 'r') as f:
   for line in f.readlines():
...
      out_packets.append(line['Gig0-0_Out_uPackets']).
      out_octets.append(line['Gig0-0_Out_Octet']).
      in_packets.append(line['Gig0-0_In_uPackets']).
      in_octets.append(line['Gig0-0_In_Octet']).
```

Since I have identical x-axis values, I can simply add all the different y-axis values to the same chart:

```
# Apply plot_date to display data in the x-axis
plt.plot_date(x_time, out_packets, '-', label='Out Packets')
plt.plot_date(x_time, out_octets, '-', label='Out Octets')
plt.plot_date(x_time, in_packets, '-', label='In Packets')
plt.plot_date(x_time, in_octets, '-', label='In Octets')
```

In addition, let us add the grid and the legend for this graph:

```
plt.legend(loc='upper left')
plt.grid(True)
```

The final result will be a combination of all available values in some separate graph. Note that some of the values in the upper left corner overlap with the legend text. I can resize the image itself and/or use the pan/zoom option to move around the graph to see all values. Router1 Matplotlib graph with multiple lines is represented in Figure 4.5.

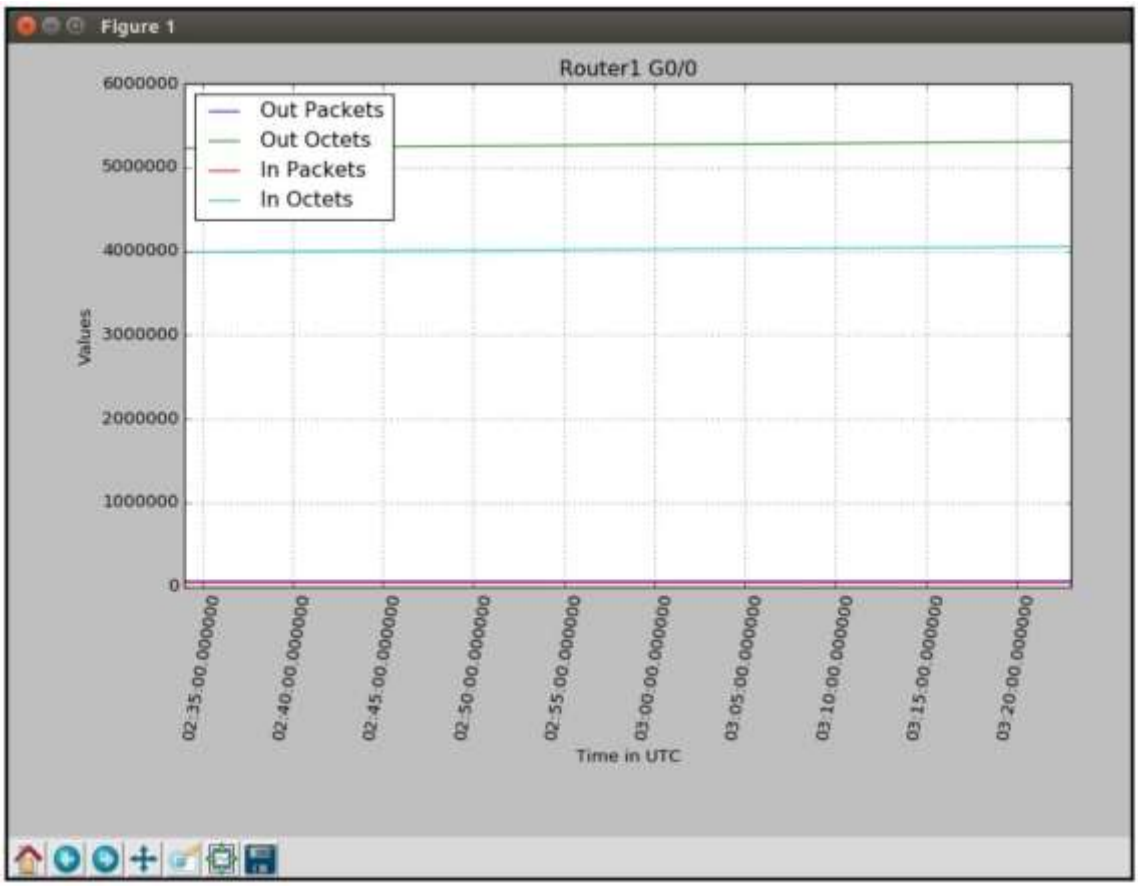

Fig. 4.5. Router1 Matplotlib graph with multiple lines

There are many additional graphical options available in Matplotlib; I are certainly not limited to just drawing charts. For example, I can use the following test data to draw the available percentage values of the different types of exchanges that I see in the cable:

#!/usr/bin/env python3

 # With measures from http://matplotlib.org/2.0.0/examples/pie_and_polar_charts/pie_demo_features.html

import matplotlib.pyplot as plt

 # The grain axis chart in which all shares will be arranged and displayed in a counterclockwise direction:

labels = 'TCP', 'UDP', 'ICMP', 'Others'

[15, 30, 45, 10]

explode = (0, 0.1, 0, 0) # Select UDP

fig1, ax1 = plt.subplots()

ax1.pie(sizes, explode=explode, labels=labels, autopct='%1.1f%', shadow=True, startangle=90)

ax1.axis('equal') # Equivalent aspect ratio ensures that the shares are drawn as a circle.

plt.show()

The above code results in the following fraction diagram from *plt.show()* that is represented in Figure 4.6.

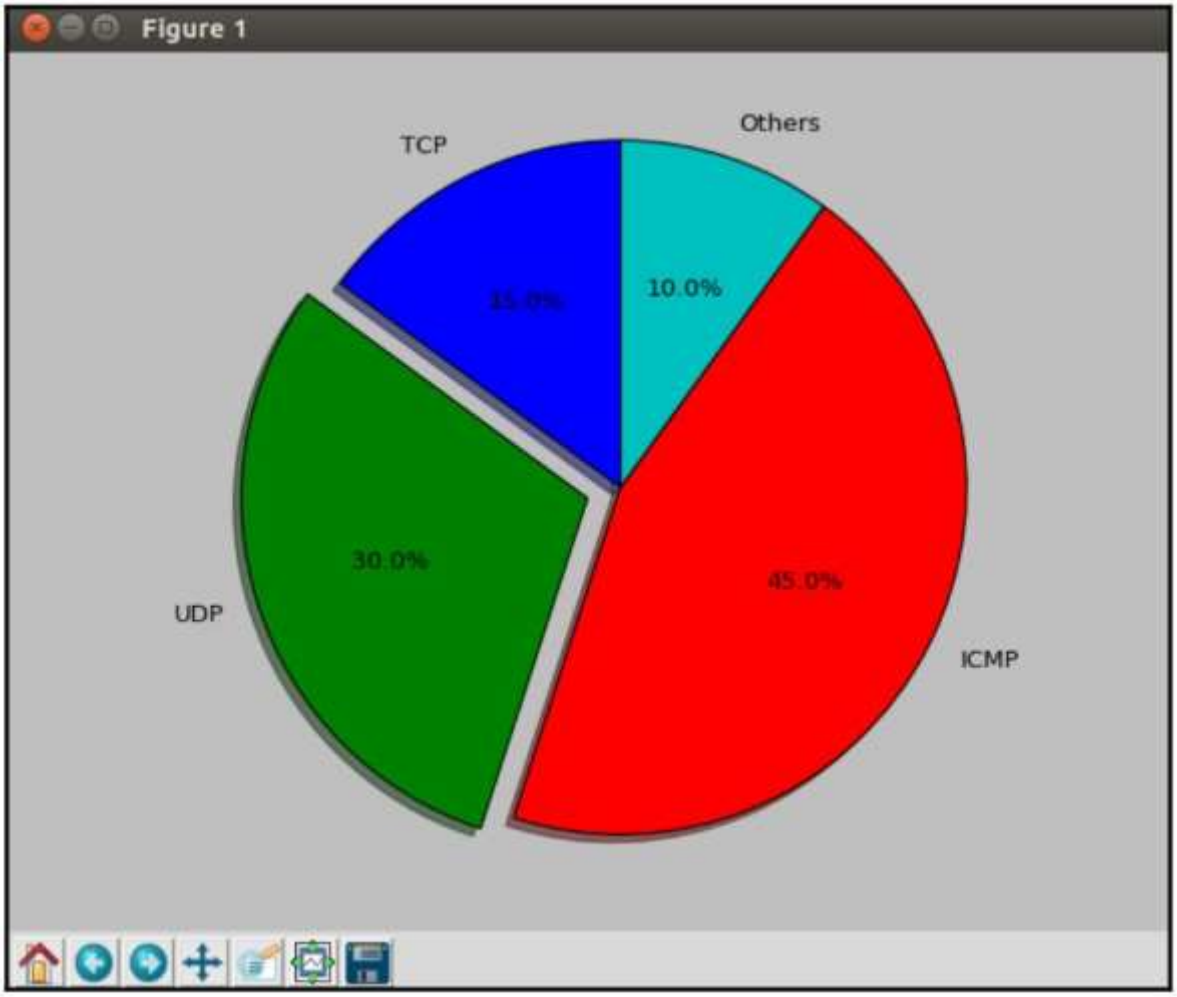

Fig. 4.6. Matplotlib fractional chart

Additional Matplotlib resources

Matplotlib is one of the best Python drawing libraries that provides typographic quality drawings. With over 4,800 stars on GitHub, it is, among other things, one of the most popular open source projects. It is translated directly into bug fixes, user community and universal application. It will take some time to study this package, but it will pay for itself a hundredfold.

In the next section, I'll look at another popular Python graphics library: Pygal.

4.7. Pygal

Pygal is some dynamic SVG graphics library written in Python. The most important advantage of Pygal, in my opinion, is that it creates the format of scalable vector graphics (SVG, Scalable Vector Graphics) in a simple and natural way. There are many advantages to SVG over other graphics formats, but two of the main advantages are that it is web browser friendly and that it provides scalability without compromising image quality. In other words, I can display all the resulting images in any modern web browser and resize it forward and backward without losing the details of this graphics.

Installation of

Installation is done with pip:

$ sudo pip install pygal

$ sudo pip3 install pygal

First example Pygal

Let's take a look at the example of line drawing shown in the Pygal documentation available at http://pygal.org/en/stable/documentation/types/line.html:

>>> import pygal

>>> line_chart = pygal.line()

>>> line_chart.title = 'Browser usage evolution (in %)'.

>>> line_chart.x_labels = map(str, range(2002, 2013))

>>> line_chart.add('Firefox', [None, None, 0, 16.6, 25, 31, 36.4, 45.5, 46.3, 42.8, 37.1]).

<pygal.graph.line.line object at 0x7fa0bb009c50>

>>> line_chart.add('Chrome', [None, None, None, 0, 3.9, 10.8, 23.8, 35.3]).

<pygal.graph.line.line object at 0x7fa0bb009c50>

>>> line_chart.add('IE', [85.8, 84.6, 84.7, 74.5, 66, 58.6, 54.7, 44.8, 36.2, 26.6, 20.1]).

<pygal.graph.line.line object at 0x7fa0bb009c50>

>>> line_chart.add('Others', [14.2, 15.4, 15.3, 8.9, 9, 10.4, 8.9, 5.8, 6.7, 6.8, 7.5]).

<pygal.graph.line.line object at 0x7fa0bb009c50>

>>> line_chart.render_to_file('pygal_example_1.svg')

View of the resulting chart in Firefox is represented in Figure 4.7.

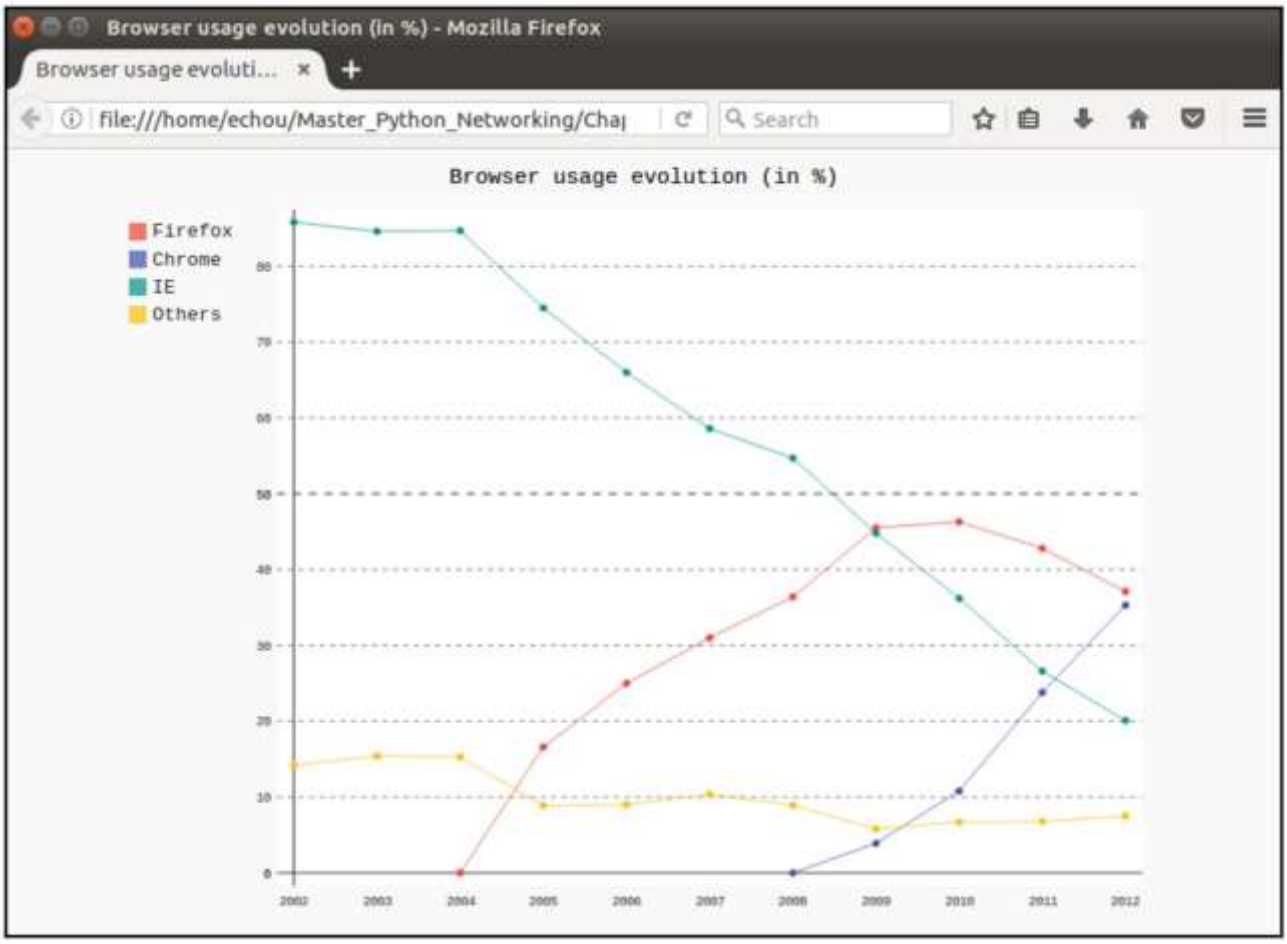

Fig. 4.7. Simple Pygal graph

I can use the same method to output SNMP graphical results that I have in my hands. I will do this in the next section.

Pygal results for SNMP

For Pygal line graphs, I can basically follow the same pattern as in my Matplotlib example, where I created a specific list of values by reading them from a specified file. I no longer need to convert my x-axis values to some internal format like I did for Matplotlib; however, I really do need to convert all the numbers for each value that I get as a floating point:

```
with open('results.txt', 'r') as f:
    for line in f.readlines():
        line = eval(line)
```

x_time.append(line['Time'])

out_packets.append(float(line['Gig0-0_Out_uPackets'])).

out_octets.append(float(line['Gig0-0_Out_Octet'])).

in_packets.append(float(line['Gig0-0_In_uPackets'])).

in_octets.append(float(line['Gig0-0_In_Octet'])).

I can use the same mechanism that I have seen when building my line chart:

line_chart = pygal.line()

line_chart.title = "Router 1 Gig0/0"

line_chart.x_labels = x_time

line_chart.add('out_octets', out_octets)

line_chart.add('out_packets', out_packets)

line_chart.add('in_octets', in_octets)

line_chart.add('in_packets', in_packets)

line_chart.render_to_file('pygal_example_2.svg')

The final conclusion is similar to what I have already seen, but this result is presented in SVG format, which is easier to display in some web page. It can be viewed from some browser as in example in Figure 4.8.

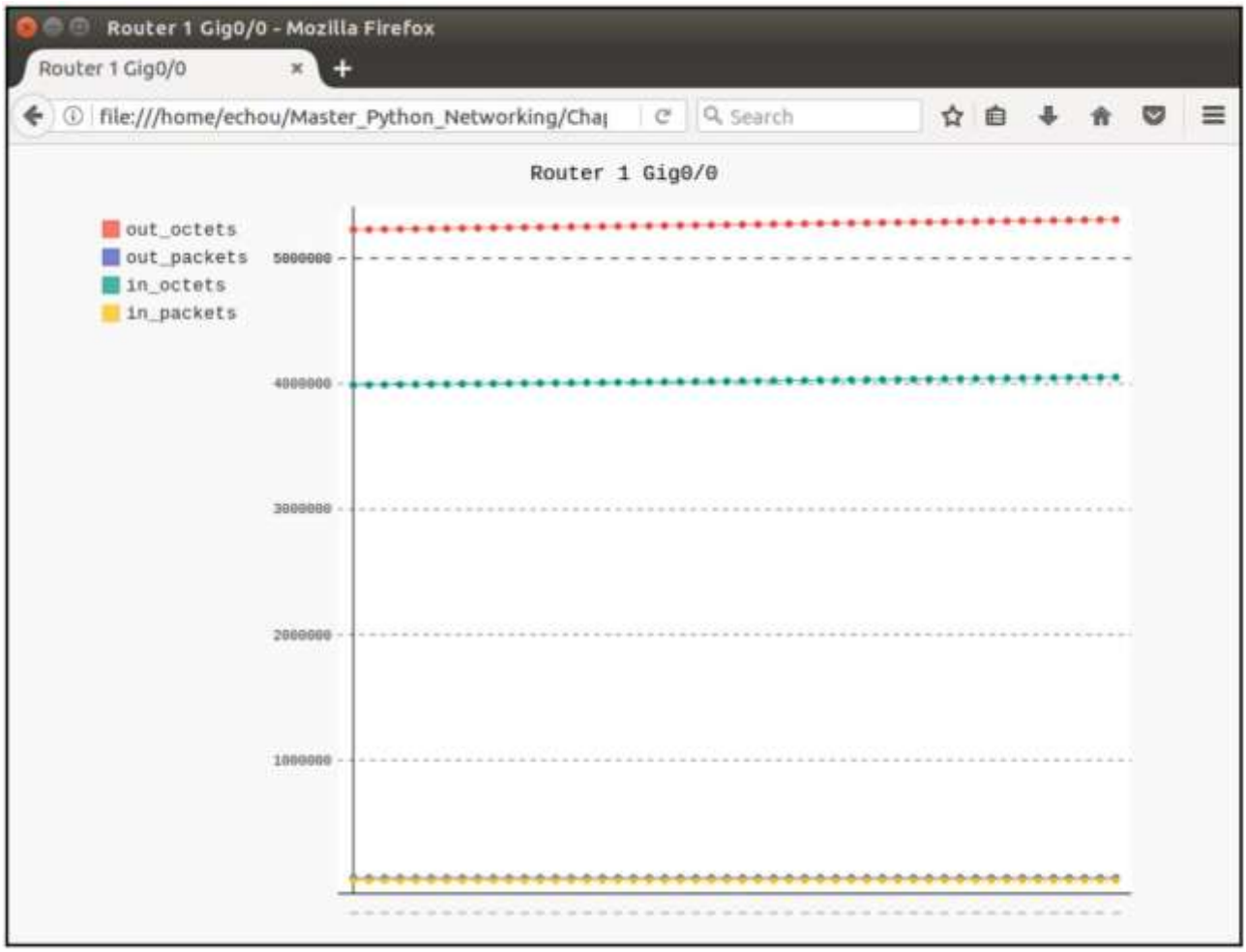

Fig. 4.8. Pygal Router 1 chart with multiple lines

Just like Matplotlib, Pygal provides a lot of additional features for drawing charts. For example, I can use the object pygal.Pie() to draw a grain axis chart in Pygal:

```
#!/usr/bin/env python3
import pygal
line_chart = pygal.Pie()
line_chart.title = "Protocol Breakdown"
line_chart.add('TCP', 15)
line_chart.add('UDP', 30)
line_chart.add('ICMP', 45)
line_chart.add('Others', 10)
line_chart.render_to_file('pygal_example_3.svg')
```

The resulting SVG file will be similar to the PNG that Matplotlib created that is represented in Figure 4.9.

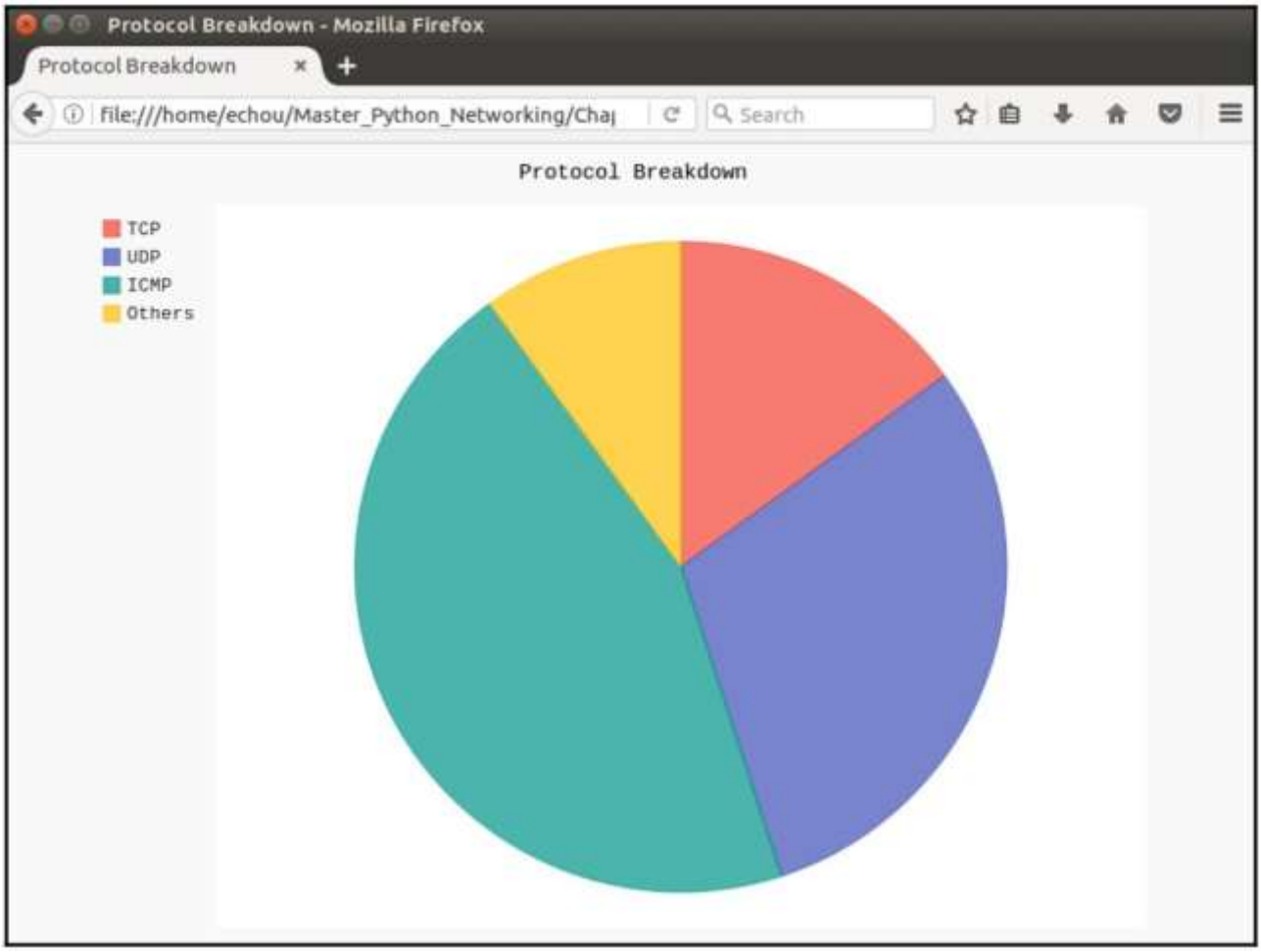

Fig. 4.9. Pygal Share Graph

Additional Pygal resources

Pygal provides many additional customizable properties and graphical features for those data I collect from more basic network monitoring tools such as SNMP. I have demonstrated the simplest line and fractional graphs here. I can find additional information about this project here:

- Pygal documentation
- GitHub Pygal project page

In my next section I will continue the topic of SNMP network monitoring, but for a complete network monitoring system called Cacti.

4.7. Python for Cacti

Released in 2001, Cacti is an open source network monitoring and graphical display tool designed as an improved interface for RRDtool. With the legacy of MRTG and RRDtool, I will find some familiar SNMP diagrams, templates and polls. Being a tool supplied in a package, its installation and application should remain within the framework of the tool itself. However, Cacti offers the functionality of personal data queries that I can use for Python. In this section, I will look at how I can apply Python as a technique for Cacti.

Installation in Ubuntu unpretentious; install Ubuntu under VM control:

$ sudo apt-get install cacti

This will include a sequence of installation and configuration steps, including actions for the MySQL database, web server (Apache or lighthttpd), and various configuration tasks. Once installed, go to *http://<ip>/cacti* to get started. The most recent step is to register with the default username and password (admin/admin); I will be prompted to change the set password.

After registering, I can add some device according to the documentation and match it with some template. There is a previously prepared Cisco template that I can continue with. Cacti has good documentation on adding devices and creating my own first graph, so I took a quick look at some screenshots in Figure 4.10.

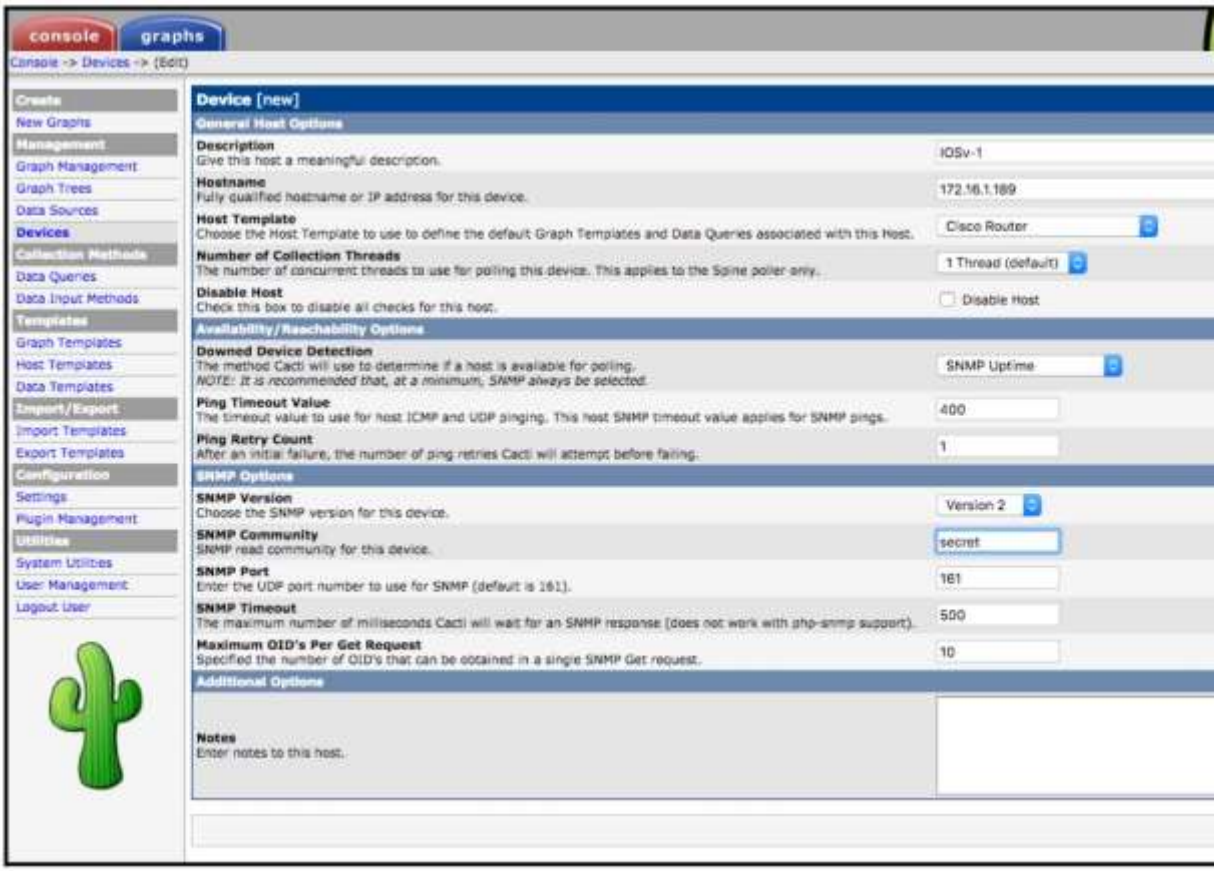

Fig. 4.10. Cacti screenshot usage example

An indication that myyour SNMP is working, so that I can observe the performance of this device, example of what is represented in Figure 4.11.

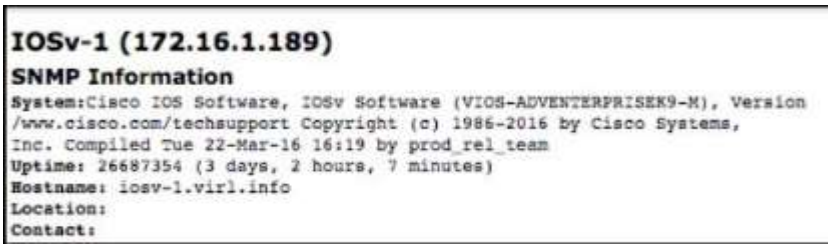

Fig. 4.11. Acknowledgement indication

For the exchange interface and other statistics I can add charts as in example of Figure 4.12.

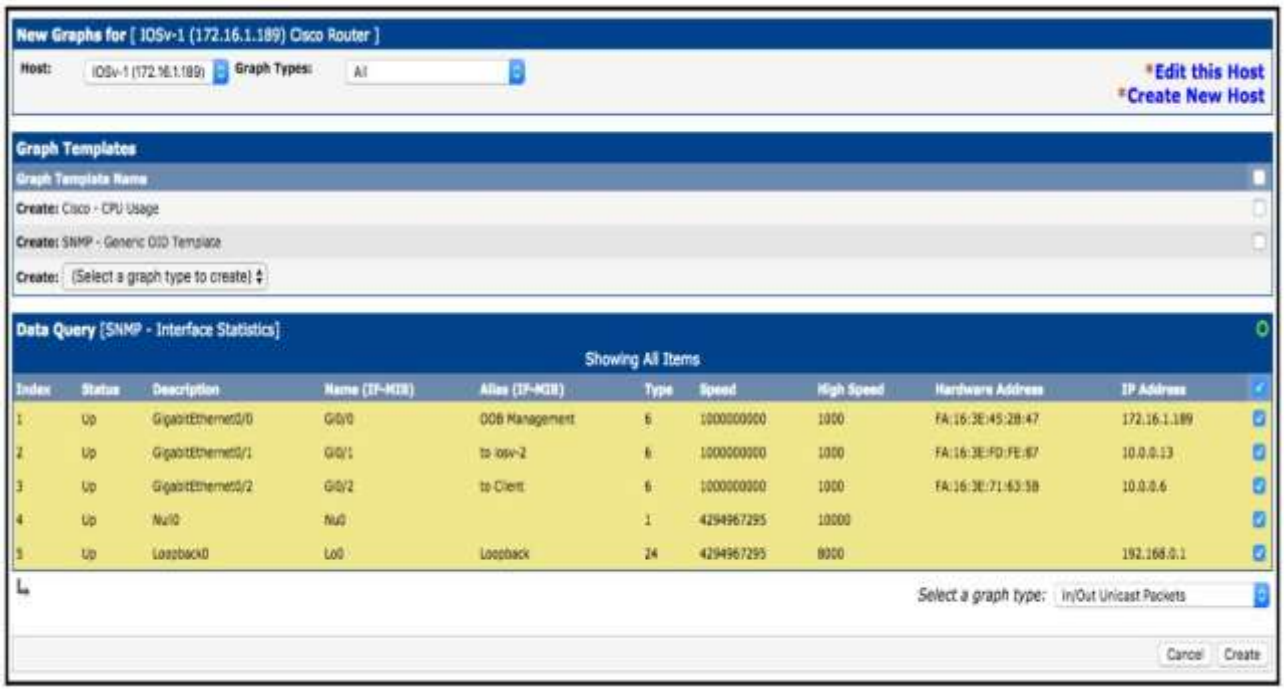

Fig. 4.12. Adding graphs of the exchange interface and other statistics

Over time, I will begin to track the exchange as it is displayed in Figure 4.13.

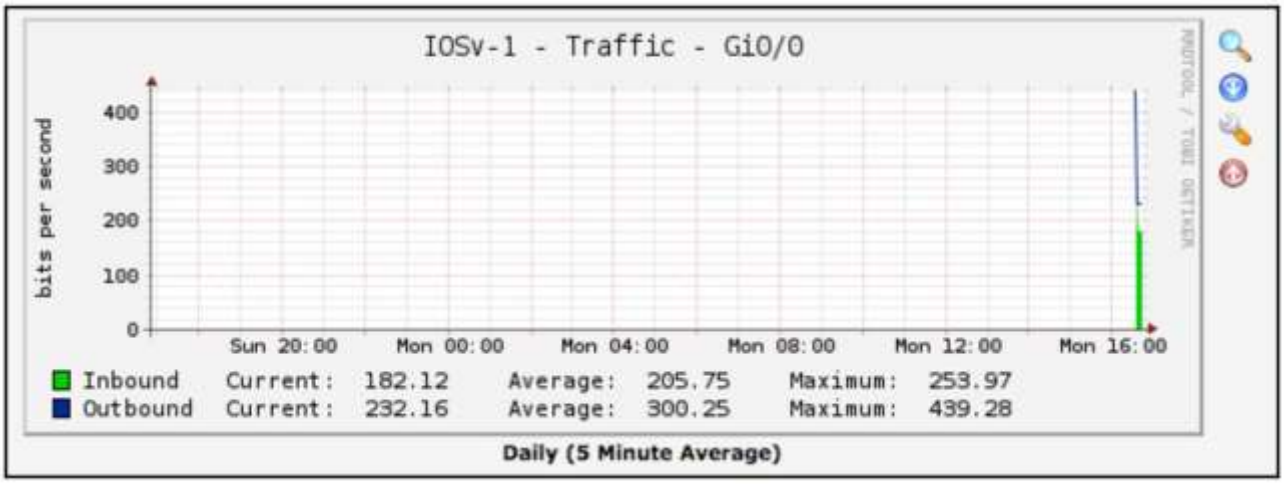

Fig. 4.13. Example result

Now I am ready to consider how to apply Python scripts to extend Cacti data collection functionality.

4.8. The Python script as a source of input data

There are two documents that I should read before applying my Python script as an input source:

- Methods of data input;
- How to make my scripts work with Cacti.

It will be interesting to know what variants of using Python script are used to extend data input? One of the usage options would be to monitor resources that do not have the corresponding OID. Let's say I would like to know how many times the available permit_snmp access list allowed a host to make some SNMP request. I know that I can see the total number of matches through the CLI:

iosv-1#sh ip access-lists permit_snmp | i 172.16.1.173

10 permit 172.16.1.173 log (6362 matches)

However, there is a possibility that the associated OID is missing (or I can pretend that it is not). This is where I can apply some kind of external script to implement some output that can be used by the Cacti host itself.

I can reuse the Pexpect script. I will rename it cacti_1.py. Everything should be familiar from the original script except that I will execute my CLI command and save its output:

```
for device in devices.keys():
...
    child.sendline('sh ip access-lists permit_snmp | i 172.16.1.173')
    child.expect(device_prompt)
    output = child.before
...
```

The following conclusion will appear in its raw form:

b'sh ip access-lists permit_snmp | i 172.16.1.173\r\n 10 permit 172.16.1.173 log (6428 matches)\r\n'.

I can use split() for this line to leave only the total number of matches and output it as a standard output in my script:

print(str(output).split('(')[1].split()[0])

To check, I can see a certain number of accumulated changes by running this script a certain number of times:

$ ./cacti_1.py

6428

$ ./cacti_1.py

6560

$ ./cacti_1.py

6758

I can make this script executable and place it at the default location of the Cacti scripts:

$ chmod a+x cacti_1.py

$ sudo cp cacti_1.py /usr/share/cacti/site/scripts/

Cacti documentation provides detailed steps on how to add script results to the overall output schedule. These steps include adding the script itself as a certain data input method, adding the required method to the data source and then creating a certain graph for viewing as in Figure 4.14.

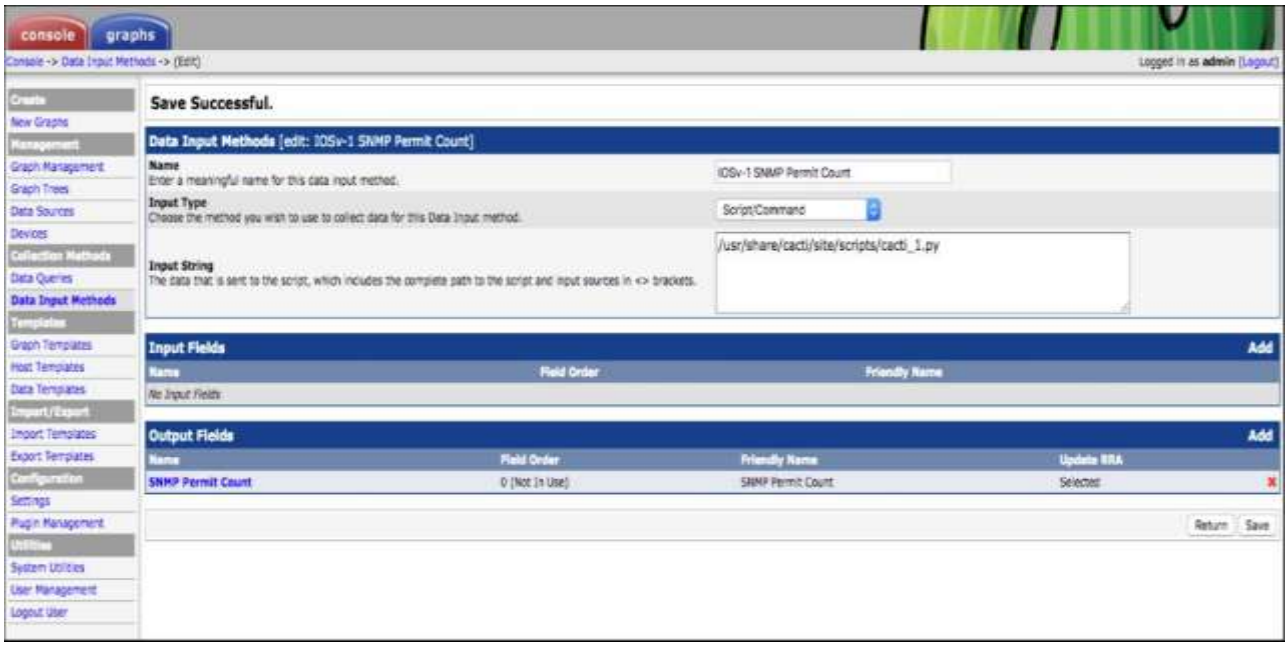

Fig. 4.14. Scenario of some data input method, adding the required method to the data source and then creating some graphics

Conclusions on Part 4

SNMP is a common method of providing network monitoring services for devices. RRDtool with Cacti as the interface provides a good platform to apply to all network devices via SNMP.

In this chapter It was learned how to perform network monitoring via SNMP. SNMP related commands were set up on my network devices and used my network management VM with SNMP query system to polling existing devices. PySNMP module was used to simplify the automation of SNMP queries. It was also learned how to save the query results in regular files for use in the following examples.

Later in this chapter was applied two different Python visualization packages, namely Matplotlib and Pygal, to graphically display the SNMP results. Each package has its own advantages. Matplotlib is a mature project with rich features, which is widely used in scientific data. Pygal produces graphics in SVG format, which are flexible and friendly to the web world.

Towards the end of this chapter I took a look at Cacti, some kind of SNMP-based network monitoring system, and how Python can be used to extend the monitoring capabilities of this platform.

CONCLUSIONS

The primary goal of the Master's Degree Thesis – "Efficiency analysis of wireless networks" – is the design of the network, built to solve the tasks of efficiency analysis and classification. The topicality of the master's degree thesis is the necessity to analyze the existing methods and techniques of network efficiency analysis, choose the most suitable for the given task and apply it with recommendation of further improvement. During the master's degree thesis preparation, a great volume of information was collected and analyzed.

Part 1 of the master's degree thesis contains the detailed description of the subject field, the overview of existing problem and actuality for network analysis.

According to the IEEE 802.15.4 specification, BSS is divided into three types of devices: terminal devices (OUs), routers and the only coordinator that manages and collects all information from the network. According to the standard, any router must take over the role of the coordinator in case of its failure.

All energy consumption values of the BCS node will be depend on the internal architecture of the node itself. BSS node consists of 5 main components. In this case microcontroller, transceiver and memory can to be made on one crystal that contributes as miniaturization of the BSS node itself, as well as reducing its power consumption.

Part 2 of the master's degree thesis t is that for the development of society, it is necessary to implement innovative systems. This is due to the fact that humanity is moving to a new level of communication and transmission of information.

The rapid development of telecommunications, based on the achievements of microelectronics, has dramatically increased the efficiency of transportation, distribution, processing, storage of information, as well as the throughput of systems and transmission media. The main feature of modern telecommunications is the transmission and processing of signals in digital form. Digitalization has allowed to build cost-effective digital communication systems with a wide range of services, compared to analog.

Part 3 of the master's degree thesis contains the description of the structure of SNMP, architecture of SNMP, it's advantages and disadvantages, principle of work and agent's description.

SNMP is used to obtain information from network devices about their status, performance, and other characteristics, which are stored in the Management Information Base (MIB).

An agent in the SNMP protocol is a processing element that provides access to the values of the MIB variables for the managers located at the network control stations, thus allowing them to implement control and monitoring functions for the device.

Part 4 contains different ways to perform network monitoring tasks' solutions. Many of the tools have been considered so far can be linked together or directly managed from Python. Like everything have been considered so far, network monitoring must work with two parts. First, It was needed to know what information hardware can transmit. Second, It was needed to determine what useful information It could be interpreted.

The biggest challenge when installing SNMP is finding the correct OID. Some of the available OIDs are defined in the MIB-2 standard; others are located in a part of their corporate tree. However, the best hint is the manufacturer's documentation. There are many tools available that can be helpful, such as the MIB Browser; I can add MIBs to this browser and see the description of all OIDs based on corporate addressing. A tool like Cisco's SNMP Object Navigator turns out to be very valuable.

In this chapter It was learned how to perform network monitoring via SNMP. SNMP related commands were set up on my network devices and used my network management VM with SNMP query system to polling existing devices. PySNMP module was used to simplify the automation of SNMP queries. It was also learned how to save the query results in regular files for use in the following examples.

Later in this chapter was applied two different Python visualization packages, namely Matplotlib and Pygal, to graphically display the SNMP results. Each package has its own advantages. Matplotlib is a mature project with rich features, which is widely used in scientific data. Pygal produces graphics in SVG format, which are flexible and friendly to the web world.

Towards the end of this chapter I took a look at Cacti, some kind of SNMP-based network monitoring system, and how Python can be used to extend the monitoring capabilities of this platform.

REFERENCE LIST

1. Колюбякин В. «Беспроводные мультисервисные сети»- М.: Теле-спутник. 2016.

2. IEEE Standard 802.11n-2009 – IEEE Standard for Information technology – Local and metropolitan area networks – Specific requirements. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 5: Enhancements for Higher Throughput.

3. Олифер В.Г., Олифер Н.А. Компьютерные сети. Принципы, технологии, протоколы. – СПб.: БХВ – Санкт-Петербург, 2010. – 916 с. 4 вид

4. Созикін А. Сети и системы телекоммуникаций WiFi [Електронний ресурс]. URL: http://www.slideshare.net/ AndreySozykin/09-networks1.

5. Джим Гейер. Беспроводные сети. Первый шаг: Пер. с англ. — М. : Издательский дом "Вильяме", 2005. — 192 с.: ил. — Парал.тит. англ.

6. «Оборудование беспроводных сетей передачи данных стандарта IEEE 802.11n» [Електронний ресурс]. URL: http://www.linkc.ru/article.php?id=223

7. «Группа стандартов WiFi IEEE 802.11» [Електронний ресурс]. URL: http://wi-life.ru/texnologii/wi-fi/wi-fi-standarty

8. «Wi-Fi и технология IEEE 802.11» [Електронний ресурс] . URL: http://www.bookasutp.ru/Chapter2_11_4.aspx

9. «Стандарт 802.11n — путь к новому поколению WLAN» [Електронний ресурс]. URL: http://compress.ru/article.aspx?id=10804. Увеличение эффективности передачи

10. «Эволюция скорости передачи данных в сетях Wi-Fi» [Електронний ресурс]. URL: https://habrahabr.ru/post/254559/

11. «Мультисервисные сети» [Електронний ресурс]. URL: http://www.intech-nsk.ru/solutions/5/

12. «Концепция построения мультисервисной сети связи» [Електронний ресурс].

13. Гургенидзе А. Т., Кореш В. И. Мультисервисные сети и услуги широкополосного доступа. Санкт-Петербург, Наука и техника, 2003. 400 с.