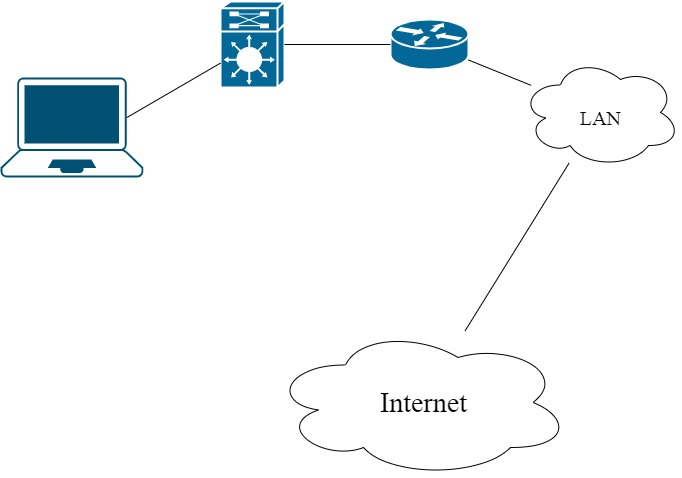
Sistem praćenja mrežnog saobraćaja pomoću SNMP protokola

**Predmet: Internet protokoli i tehnologije**

**Student: Stefan Muharem RT-50/19**

# UVOD

U datom radu prikazaćemo servis napisan u programskom jeziku Python koji pomoću SNMP (engl. Simple Network Managment Protocol) protokola prati stanje mrežnih uređaja, konkretno prikuplja informacije o ulaznom i izlaznom saobraćaju, iskorišćenosti procesora itd. te iste skladišti u bazu podataka RRD (Round Robin Database). Zatim pomoću RRD Tool-a i podataka iz baze generiše grafikone koji se krajnjem korisniku prikazuju putem web stranica generisanih pomoću jinja2 šablona.



# SNMP

SNMP (engl. Simple Network Managment Protocol) je standard za nadgledanje i upravljanje uređaja povezanih na mrežu, kao što su ruteri, svičevi, računari, štampači, video kamere itd.

Osim mogućnosti čitanja statistike sa samih uređaja, SNMP može slati i kontrolne poruke uređaju da izvrši određeni zadatak npr. koristeći SNMP komande moguće je restartovati ruter na daljinu.

Generalno sistemi nadgledani sa SNMP protokolom se sastoje se od tri glavne komponente:

* **Sistem za nadzor**. Zadužen za nadgledanje svih uređaja.
* **Nadgledani uređaji**. Tu spadaju svi uređaji nadgledani sistemom za nadzor.
* **SNMP agent**. Aplikacija koja se pokreće na svakom od nadgledanih uređaja i sarađuje sa sistemom za nadzor.

Protokol ima 7 osnovnih komandi, od kojih su nam najzanimljivije: ***get***, ***get* *bulk*** i ***response***. Kao što se može naslutiti, prve dvije komande su komande koje sistem za nadzor šalje agentu a response je odgovor od agent softvera.

## Razmjena informacija na osnovu upita između sistema za nadzor i SNMP agenta.

## SNMP protokol nema definisan način razmjene informacija, stoga sistem za nadzor nema način da zatraži od agenata listu dostupnih promjenjivih. Ovaj problem riješen je upotrebom MIB baze (Managment Information Base), svaki uređaj obično ima dodjeljen MIB, koji opisuje strukturu nadgledanih podataka na tom sistemu. Takav MIB bi ispisao listu u hierarhijskom poretku svih OIDs(object identifiers) koji su dostupni na nadgledanom uređaju. Oid predstavlja čvor u stablu objekata.

U projektu ćemo koristiti 2 OID-a:

* OID String: 1.3.6.1.2.1.2.2.1.10
  + OID Ime: IfInOctets
  + Opis: Ukupan broj bajtova (okteta) primljenih na interfejs
* OID String: 1.3.6.1.2.1.2.2.1.16
  + OID Ime: IfOutOctets
  + Opis: Broj bajtova (okteta) proslijeđen sa interfejsa ka mreži

Autentifikacija na SNMP:

U ranijim verzijama SNMP autentifikacija je bila primitivna i podložna napadima. SNMP agent defniše 2 community stringa:

Jedan za read-only pristup (sam za čitanje), drugi za read/write pristup ( i pisanje i čitanje). Kad se ostvari veza sistem za nadzor i agenta, sistem se mora autentifikovati sa jednim od ova 2 stringa. Agent će prihvatati komande samo od nadzornog sistema koji se autentifikovao sa ispravnim community stringom.

Prije nego krenemo pisati aplikaciju, bacimo pogled na SNMP upit s komandne linije. Ovo je praktično korisno ako želimo provjeriti da li sve radi kako treba.

Najkorinsija komanda iz Net-SNMP-Utils-a je **snmpwalk**, koja uzima OID čvor kao argument i pokušava otkriti sve čvorove ispod njega (child oids).

Ova komanda koristi SNMP operaciju **getnext** koja vraća sljedeći čvor u stablu i dozvoljava put kroz podstabla datog čvora.

Primjer primjene snmpwalk komande na TP-LINK ruteru:

$ snmpwalk -v2c -c public -On 192.168.10.1 1.3.6.1.2.1.1

.1.3.6.1.2.1.1.1.0 = STRING: "TD-8840T"

.1.3.6.1.2.1.1.2.0 = OID: .1.3.6.1.4.1.1.2.3.4.5

.1.3.6.1.2.1.1.3.0 = Timeticks: (44089000) 5 days, 2:28:10.00

.1.3.6.1.2.1.1.4.0 = ""

.1.3.6.1.2.1.1.5.0 = STRING: "TP-LINK"

.1.3.6.1.2.1.1.6.0 = ""

.1.3.6.1.2.1.1.7.0 = INTEGER: 14

Ako ne navedemo OID , snmpwalk će koristiti default-ni SNMP OID (1.3.6.1.2.1) kao početnu tačku.

Takav ispis bi zauzeo nekoliko strana pa ga stoga odlučio za gore pomenuti primjer.

# RRDTool

## Smještanje podataka upotrebom RRDTool-a

RDDTool je aplikacija kreirana od strane Tobias Oetiker-a, te kasnije postaje de fakto (de facto) standard za grafičko nadgledanje podataka. Grafovi kreirani RRDTool-om su koristeni u raznim različitim alatkama za nadzor (monitoring), kao što su: Nagios, Cacti itd..

Ovdje ću napraviti osvrt na samo strukturu RRDTool baze podataka a i aplikacije kao cjeline.

Pomenuću neke specificnosti ove baze, kako dodati nove podatke u nju te kako ih opet koristiti kasnije. Takođe komande i tehinke na osnovu kojih ovaj alat od podataka kreira graf.

Samo integrisanje RRDTool baze podataka sa aplikacijom prikazaću kroz sami kod kasnije, koji će biti ispraćen komentarima.

## Upoznavanje sa samim RRD alatom

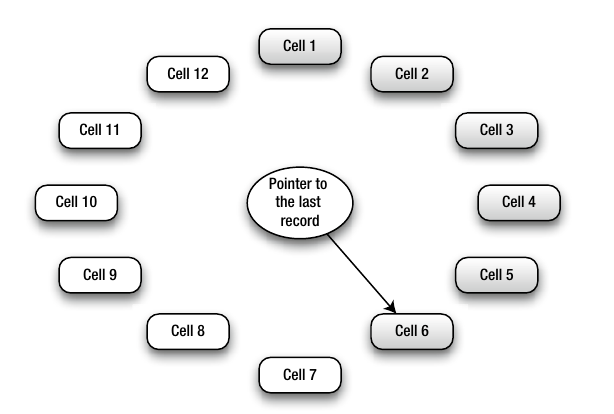
RRDTool nudi tri različite mogućnosti. Prva, koistimo ga kao bazu podataka koja nam omogućava ti vraćanje vrijednosti u sklopu samog alata.

Takođe izvršava kompleksne zadatke manipulisanja (obrade) podataka, kao što su:

data resampling i rate calcultions (rate calculations bi bilo izracunavanje prirasta?).

Konačno treća mogućnost je kreiranje sofisticiranih grafova koristeći podatke iz različitih izvora naše baze.

Prva osobina koja čini RRD drugačijim od konvencionalnih baza je da baza ima limitiranu veličinu. Što znači da smo svjesni veličine baze jos od trenutka inicijalizacije i veličina se nikad ne mjenja. Novi podaci se pišu na mjesto zastarjelih podataka, i tako se proces ponavlja iznova i iznova.



Ubacio sam sliku radi lakše vizuelne reprezentacije, zamislimo da je naša baza u stanju čuvati 12 zapisa, svaki u svojoj ćeliji. U slučaju kada je baza prazna, pišemo podatke u ćeliju 1. Takođe svakim korakom mjenjamo i vrijednost pokazivača sa ID-om posljednje ćelije u kojoj smo pisali. U trenutku kada dođemo do posljednje ćelije (ćelija 12), proces počinje iznova, od ćelije broj jedan.

Jedina svrha RRD-og čuvanja podataka je čuvanje performansnih podataka, što ne iziskuje održavanje kompleksnih veza između tabela s različitim podacima. U suštini u RRD bazi ne postoje tabele već je ona više nalik skupu individualnih podataka(DSs).

Još jedna unikatna stvar kod RRD je što je baza podataka dizajnirana da čuva informacije u serijama vremena, stoga svaki zapis treba biti označen sa vremenom(timestamp).

Prilikom kreiranja nove baze potrebno je dodjeliti vremenski ciklus, na osnovu koga će se dodavati novi zapisi u bazu. Default-na vrijednost tog vremena je 300 sekundi ili ti 5 minuta, što u nekim slučajevima može biti zanemareno ako je to potrebno. Podaci u RDD bazi čuvaju se u Round Robin Archive(RRA). RRA je baš ono što čini RRD tako korisnim, u poređenju sa klasičnom bazom RRA bi bila nešto nalik tabele. Možemo se odlučiti za jedno od 4 agregacione funkcije (CF) : average , min, max i last; koji će biti primjenjeni na neki broj/grupaciju podataka(grupisanih na osnovu vremena - timestamp) koju mi odredimo. Rezultat se čuva u round robin "tabelu". Možemo čuvati više RRA-ova u našoj bazi, i one mogu biti neovisne jedne od drugih. Na primjer, jedna RRA čuva prosjek(average) za vrijednosti posljednjih 10 zapisa dok druga čuva prosjek za posljednjih 100 zapisa.

# Zahtjevi

Ideja je da kreiramo sistem koju će pratiti mrežne uređaje u sklopu naše mreže i količinu saobraćaja koji prolazi kroz iste. Stoga prvi zahtjev koji sistem mora ispuniti je slanje upita ka mrežnim uređajima u sklopu mreže pomoću SNMP protokola. Te informacije prikupljenje od strane uređaja mora negde i čuvati određeni vremenski period radi dalje analize, zato je RRD baza podataka idealan kandidat. Na samom kraju informacije prikupljene u bazi treba grafički prikazati i omogućiti im laku dostupnost krajnjem korisniku, putem web stranica.

# 

# Izrada

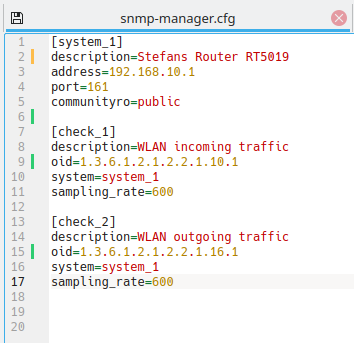
Pošto je prvi zahtev praćenje uređaja pomoću SNMP protokola, potrebno je pronaći odgovarajuću Python biblioteku za rad nad SNMP servisom. SNMP moduo nije sastavni dio standardne Python instalacije, te moramo instalirati eksterni moduo *PySNMP*.

Za smještanje podataka koristićemo RRD Tool. Pošto je RRD Tool baza izgrađena tako da svaka “tabela” ima limit u vidu količine podataka koji se mogu skladištiti tako da se najstariji brišu, o tome nećemo morati voditi računa u samom Python servisu, još jedna odlična stvar je i činjenica da RRD Tool možemo koristiti i za generisanje grafika. Isto kao i SNMP, RRDTool ima svoj Python moduo koji je takođe eksterni. Na kraju za samo generisanje web stranica koristićemo jinja2 Python moduo.

Informacije o nadgledanim uređajima držaćemo u jednostavnom *Windows INI* konifugracionom fajlu.

Za konfiguraciju ćemo trebati sljedeće informacije, dostupne za svaku provjeru:

* IP adesa ili razriješivo ime domena za sistem koji pokreće SNMP agenta.
* Community string (read only) koji će biti korišten za autentifikaciju sa agentom.
* The OID nodes numerical representation.



Aplikacija se sastoji iz dva dijela:

* Prvi dio: alat za prikupljanje informacija, koji šalje upite (queries) svim konfigurisanim uređajima i smješta podatke u RRDTool bazu, nazvaćemo ga snmp-manager.py
* Drugi dio: generator izvještaja, koji će generisati strukturu web stranice zajedno sa svim neophodnim slikama (grafova koje je kreirao rrdtool), nazvaćemo ga snmp’pages.py

Obe komponente će se pokretati sa standardne UNIX zakazivač aplikacije - **cron**, na svakih pet minuta.

**\*/5 \* \* \* \* cd /home/m00hz/Desktop/rt5019\_snmp/; echo "StefanRT5019" >> log.txt; python3 snmp-manager.py >> log.tx; python3 snmp-pages.py**

# 

# Objašnjenje python3 koda

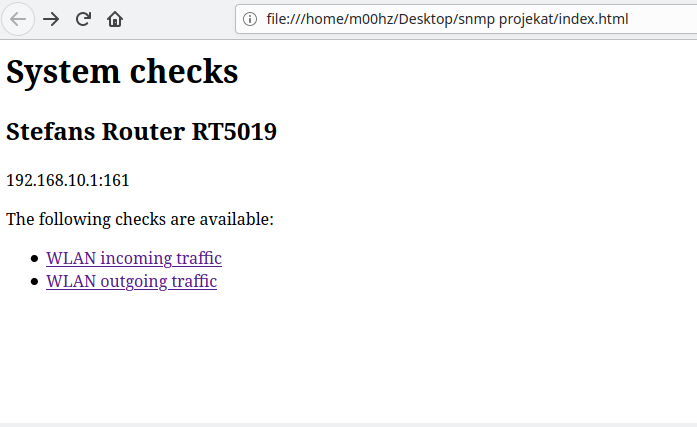
Ostavljao sam komentare važnijih dijelova koda, da sada ne kopiram sve u dokumentaciju, piložiću kod na uvid i poslati ga zajedno sa dokumentacijom.

# Konačan ishod

Izgled početne stranice kreiranje na osnovu index.tpl templejta. Zove se System checks jer linkovi sa ispte upucuju na zasebne provjere (engl. checks). Ispod imena tu je ime uređaja na kojem/kojima se izvršavaju snmp upiti. Ispod imena uredjaja nalazi se tuple u vidu ip adrese uređaja i njegovog porta za snmp protokol.

Obzirom da se u konfigurcionom fajlu nalaze samo 2 provjere, srazmjerno tome imamo i samo 2 nove stranice sa po sopstvenim grafom za svaku.

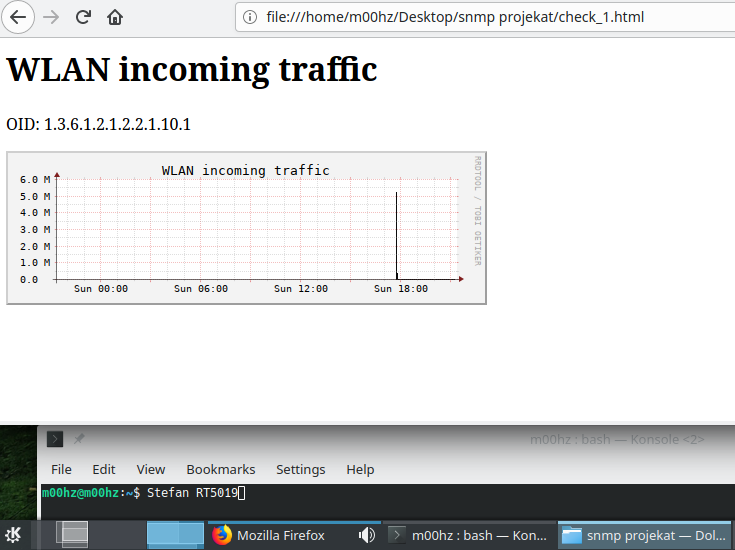
Cron zakazivač na svakih 5 minuta pokreće snmp-manager.py i snmp-pages.py te na osnovu istih, na svakih 5 min dobijamo novi izgled našeg RRDTool-a grafikona, bilo to za prikaz ulaznog ili izlaznog wlan saobraćaja.



Sa glavne stranice došli smo na narednu koja je vezana za ulazni wlan saobraćaj (WLAN incoming traffic). Kreirana je na osnovu details.tpl.

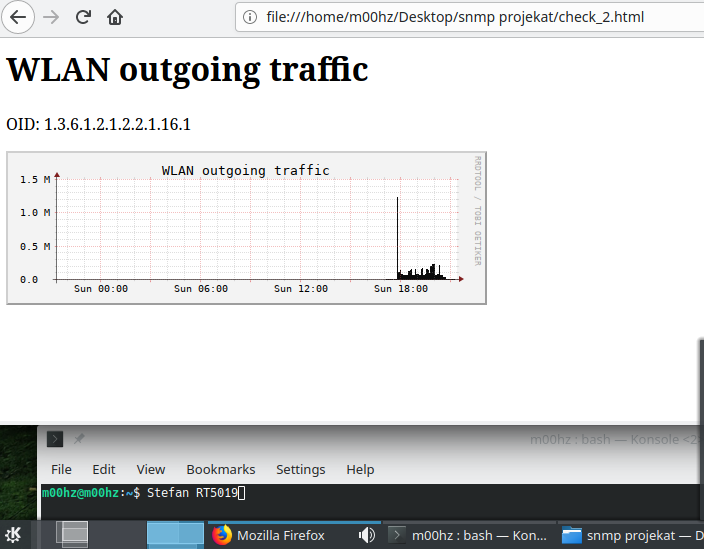
Ispod samog imena vidimo OID(Objects ID) string koji predstavlja string zadužen za mjerenje ulaznog saobraćaja.

Ukupan broj bajtova (okteta) primljenih sa mreže na interfejs.



Drugi podprozorčić je izlazni wlan saobraćaj (WLAN outgoing traffic). Identična priča kao i kod ulaznog saobraćaja samo što ovdje OID predstavlja mjerač izlaznog saobraćaja.

Ukupan broj bajtova (okteta) proslijeđenih sa interfejsa ka mreži.



# Zaključak:

U ovom projektu napravili smo jednostavan sistem za nadzor uređaja. Radeći na tome učio sam o SNMP protkolu, takođe o bibliotekama RRDTool za prikupljanje i prikazivanje (u obliku grafa) podataka te Jinja2 šablon na osnovu koga su kreirane web stranice.

Neke važne teze:

* Većina uređaja povezanih na mrežu otkriva svoje interne brojače koristeći SNMP protokol
* Svaki takav brojač ima dodjeljen, posvećen ID objekat
* OIDs su organizovani u strukturu nalik stablu, gdje su grane dodjeljene raznim organizacijama te daljim namjenama.
* RRDTool je biblioteka koja omogućava čuvanje, vraćanje i prikazivanje(obično u obliku grafa) statističkih podatka o mreži.