



Aplicație de tip Zero-config bazată pe mDNS și DNS-SD

Găină Andrei

Melinte Alexandru-Gicu

Grupa 1306A, Facultatea de Automatică și Calculatoare
Iași, Universitatea Tehnică „Gheorghe Asachi” Iași





Cuprins

1.INTRODUCERE.....	3
1.1 ZERO CONFIGURATION NETWORKING...	3
1.2 DNS(DOMAIN NAME SYSTEM).....	3
1.3 mDNS(MULTICAST DNS)	4
1.4 DNS-SD(DNS SERVICE DISCOVERY) ..	6
2.IMPLEMENTARE	8
2.1 CERINȚE DE IMPLEMENTARE:	8
2.2 DESCRIEREA IMPLEMENTĂRII	9
2.2.1 <i>Interfață</i>	9
2.2.2 <i>Funcții</i>	13
2.2.3 <i>Clase</i>	14
2.2.4 <i>Biblioteci utilizate</i>	15



1. Introducere

1.1 Zero configuration networking

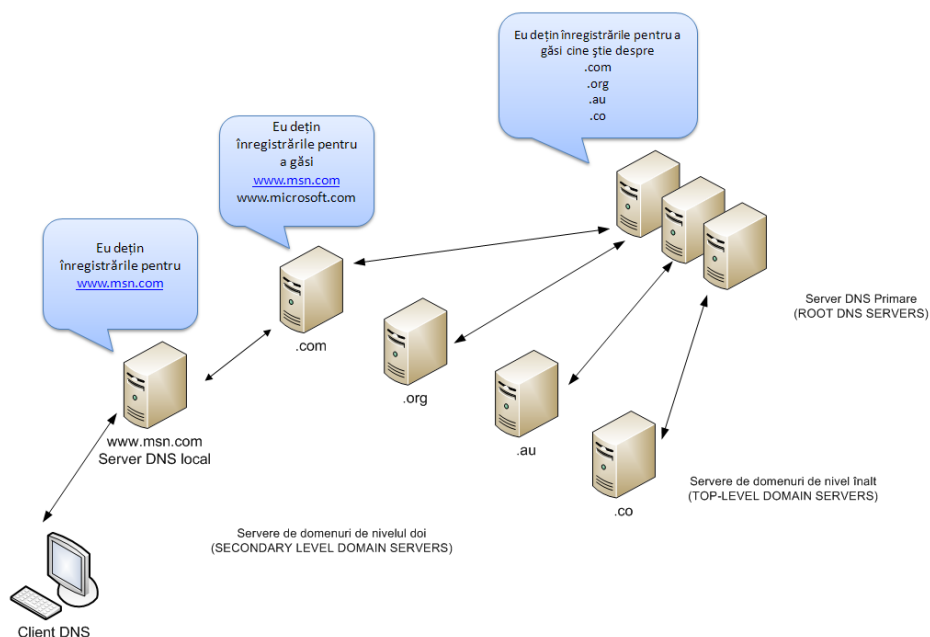
Zero-config este un set de tehnologii care creează automat o rețea de calcul bazată pe Internet Protocol Suite atunci când computerele sau perifericele rețelei sunt interconectate. Nu necesită intervenții manuale ale operatorilor și servere speciale de configurare. Fără această tehnologie, un administrator de rețea este obligat să seteze serviciile de rețea (DHCP, DNS, etc.) sau să configureze manual rețeaua fiecărui computer în parte. **Zero-config** este alcătuită din 3 tehnologii-nucleu:

- ☺ alocarea automată a adreselor numerice de rețea pentru dispozitivele din rețea;
- ☺ distribuirea și rezoluția automată a hostname-urilor computerului;
- ☺ localizarea automată a serviciilor de rețea.

1.2 DNS(Domain Name System)

DNS este un sistem distribuit de păstrare și interogare a unor date arbitrare într-o structură ierarhică. Cea mai cunoscută aplicație DNS este gestionarea domeniilor în Internet. DNS are responsabilitatea de a atribui nume de domenii și de a mapa aceste nume resurselor Internet prin desemnarea serverelor de nume autorizate pentru fiecare domeniu. Caracteristicile **DNS-ului** sunt:

- ☺ folosește o structură ierarhizată;
- ☺ delegă autoritatea pentru nume;
- ☺ baza de date cu numele și adresele IP este distribuită.





Încă un exemplu/o analogie, este faptul că DNS este ca și o agendă cu numere de telefon pentru Internet, doar că în loc de numere de telefon există o adresă IP. DNS-ul poate fi actualizat rapid și transparent, permițând schimbarea locației unui serviciu în rețea fără a afecta utilizatorii finali, care continuă să utilizeze același hostname.

1.3 mDNS(multicast DNS)

mDNS oferă abilitatea de a face operații DNS într-un domeniu local în absența oricărui alt DNS unicast convențional[RFC 6762]. Protocolul mDNS utilizează IP multicast UDP(User Datagram Protocol) packets și este implementat de Apple Bonjour și open source Avahi software packages, fiind inclus în mai toate distribuțiile Linux. **mDNS** este de asemenea implementat și în Windows 10, inițial pentru descoperirea imprimantelor conectate în rețea, iar mai apoi capabil să rezolve și hostname-uri.

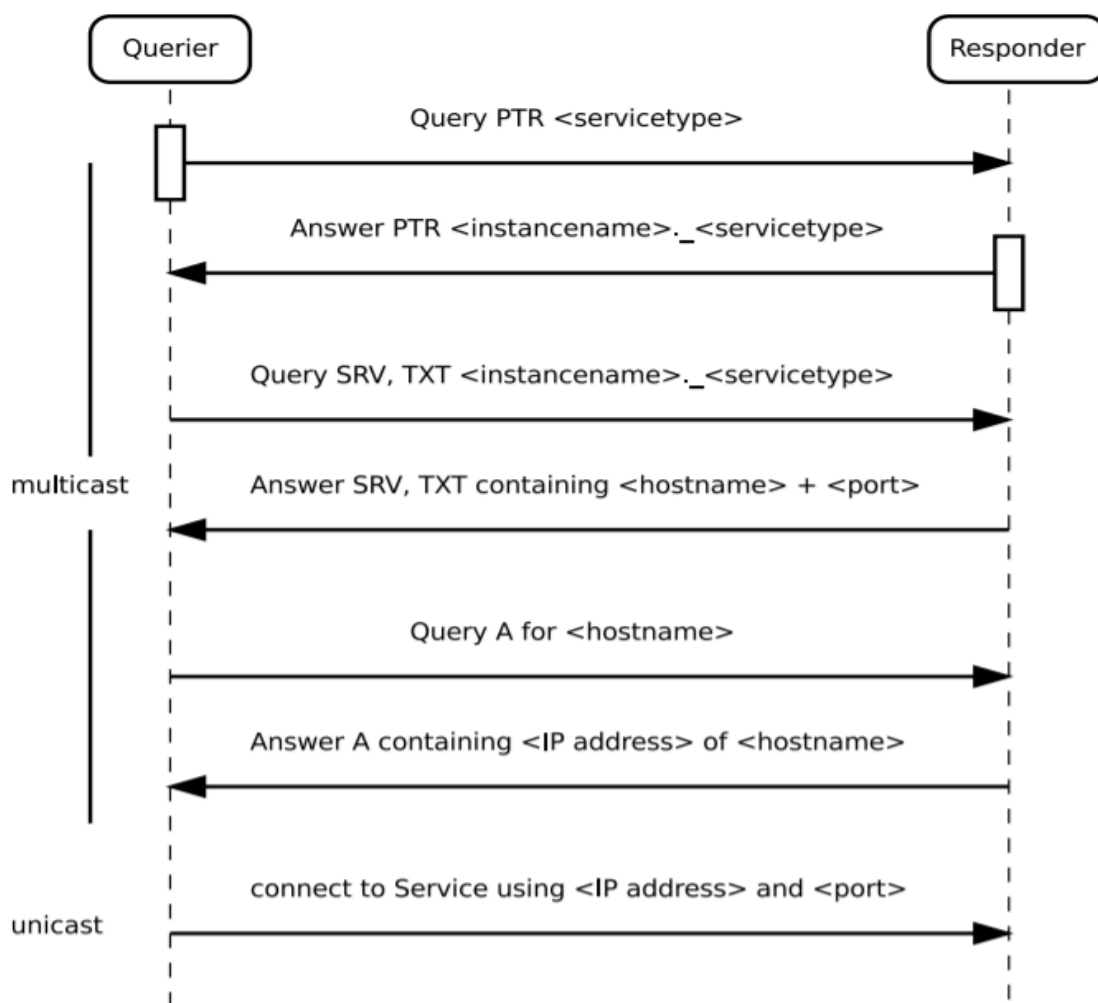


Fig. 1. Queries and responses sent using mDNS-SD.



Când un client mDNS trebuie să rezolve un hostname, acesta trimite un mesaj de interogare IP multicast care cere gazdei să se identifice. Acea mașină țintă transmite apoi un mesaj care include adresa sa IP. Toate mașinile din acea subrețea pot utiliza apoi aceste informații pentru a-și actualiza cache-urile mDNS. Orice gazdă poate renunța la revendicarea unui nume trimițând un pachet de răspuns cu **TTL egal cu zero**. În mod implicit, mDNS rezolvă exclusiv hostname-ul care se termină cu domeniul **.local top-level domain**. Acest lucru poate cauza probleme dacă **.local** include gazde care nu implementează mDNS, dar care pot fi găsite printr-un server DNS unicat convențional. Rezolvarea unor astfel de conflicte necesită modificări de configurare a rețelei pe care mDNS a fost conceput să le evite. Când este utilizat cu mDNS, DNS-SD poate oferi zero-config, spre exemplu doar conectați un dispozitiv DNS-SD / mDNS și serviciile sale sunt publicate pe link-ul local fără alte interacțiuni cu utilizatorul.

Un mesaj mDNS este un **pachet UDP** multicast trimis folosind următoarea adresare:

- ☺ **IPv4 address** 224.0.0.251 sau **IPv6 address** ff02::fb; **????????**
- ☺ **UDP port** 5353;
- ☺ când se utilizează frame-uri Ethernet, **adresa MAC multicast IP standard** 01: 00: 5E: 00: 00: FB (pentru **IPv4**) sau 33: 33: 00: 00: 00: FB (pentru **IPv6**) **????????**
<http://www.dns-sd.org/ServiceTypes.html>

Discovery of available services

- PTR | ANY queries for **_services._dns-sd._udp.<Domain>**
- A feature specified for "**problem diagnosis**".

```
*****RESULTS*****
00:08:9b: 192.168.  QUERY Answer: _services._dns-sd._udp.local. PTR Class:IN "workstation_tcp."
00:11:32: 192.168.  QUERY Answer: _services._dns-sd._udp.local. PTR Class:IN "adisk_tcp."
00:11:32: 192.168.  QUERY Answer: _services._dns-sd._udp.local. PTR Class:IN "smb."
00:11:32: 192.168.  QUERY Answer: _services._dns-sd._udp.local. PTR Class:IN "device-info."
00:11:32: 192.168.  QUERY Answer: _services._dns-sd._udp.local. PTR Class:IN "afpovertcp."
00:11:32: 192.168.  QUERY Answer: _services._dns-sd._udp.local. PTR Class:IN "http."
08:00:27:45:a7:7f 192.168.56.105 QUERY Answer: _services._dns-sd._udp.local. PTR Class:IN "workstation_tcp."
08:00:27:45:a7:7f 192.168.56.105 QUERY Answer: _services._dns-sd._udp.local. PTR Class:IN "ssh."
```

- A list of registered DNS SRV Service Types can be found in [IANA, 2017].



1.4 DNS-SD(DNS Service Discovery)

DNS-SD permite clienților să descopere o listă denumită de instanțe de serviciu și să rezolve aceste servicii către hostname-uri utilizând interogări DNS standard. Specificația este compatibilă cu serverul unicast DNS existent și software-ul client, dar funcționează la fel de bine cu mDNS într-un mediu de zero-config. Fiecare instanță de serviciu este descrisă folosind DNS SRV și DNS TXT record. Clientul descoperă lista cu instanțele disponibile pentru un serviciu interogând DNS PTR. Serverul returnează 0 sau mai multe nume de forma **Service Instance Name** = **<Instance>.<Service>.<Domain>** fiecare corespunzând unei înregistrări **TXT/SRV**. (TXT – parametrii de configurare, SRV – oferă numărul portului și numele gazdei țintă unde poate fi găsit serviciul). **<Instance>** este un username format din arbitrary Net-Unicode text, **<Service>** constă dintr-o pereche de etichete DNS, în conformitate cu convenția deja stabilită pentru SRV și **<Domain>** specifică DNS-ului subdomeniul în care sunt înregistrate aceste nume. Poate fi „local” adică „link local mDNS[RFC6762], sau poate fi un nume de domeniu DNS unicast convențional, cum ar fi „abc.org”.

DNS-SD Example

PTR record with the list of all LPR printers:

```
_printer._tcp.example.com.
```

DNS name of a service instance:

```
Apple LaserWriter 8500._printer._tcp.example.com.
```

SRV record:

Port 515

TXT record name/value pairs (excerpt):

```
txtvers=1
```

```
rp=auto
```

```
ty=Apple LaserWriter 8500
```

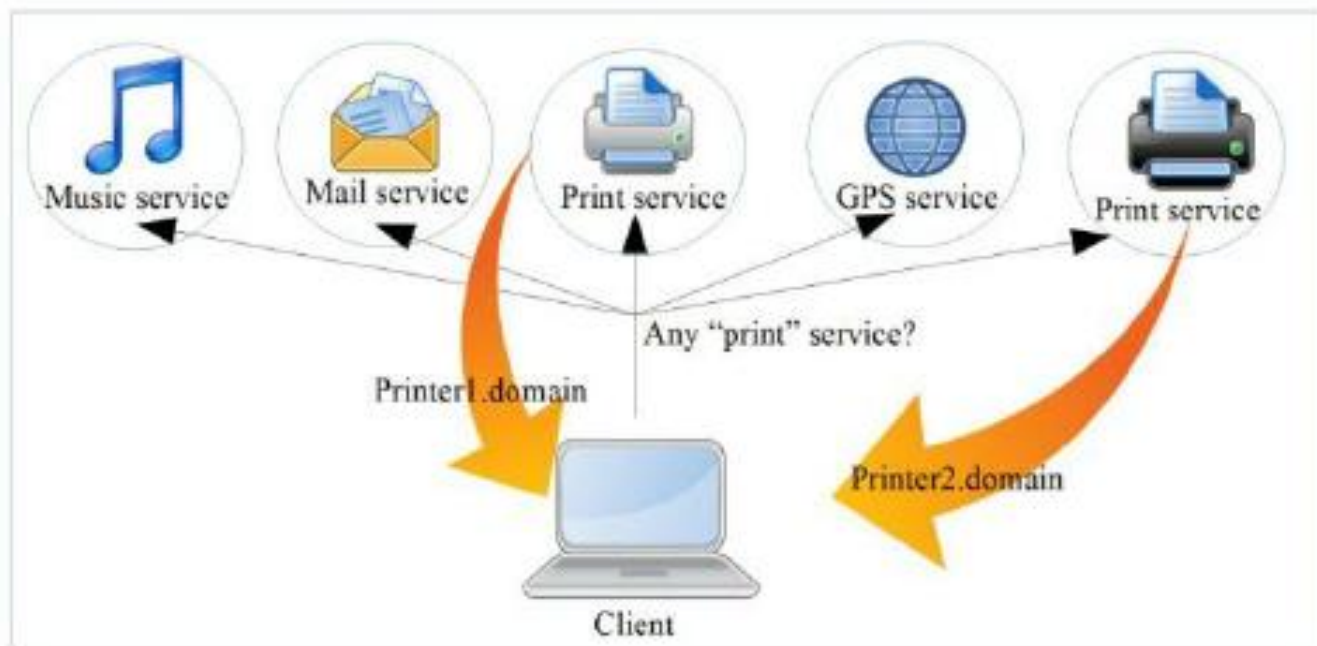
```
Copies=T
```

```
Duplex=F
```

```
PaperCustom=T
```

DNS SRV este de ajutor în localizarea instanțelor de un anumit tip de serviciu, când toate instanțele nu se pot distinge și oferă același serviciu clientului. De exemplu, înregistrarea SRV cu numele „_http._tcp.example.com.” ar permite unui client să descopere serverele de implementare a serviciului „_http._tcp” pentru domeniul „exemplu.com”. Presupunerea nedeclarată este că toate acestea serverele oferă un set identic de pagini web și nu contează clientul pe care dintre servere îl folosește, atâta timp cât îl selectează pe unul aleatoriu în conformitate cu regulile de greutate și prioritate stabilite în DNS SRV.

DNS-SD este utilizat de produsele Apple, majoritatea imprimantelor de rețea, multe distribuții Linux, inclusiv Debian și Ubuntu. Windows 10 include suport pentru DNS-SD pentru aplicații scrise folosind JavaScript. Delegând subdomeniile „_tcp” și „_udp”, tot volumul de muncă legat la DNS-SD poate fi descărcat pe o altă mașină. Această flexibilitate, de a gestiona DNS-SD pe serverul DNS principal sau nu, la decizia administratorului de rețea, este unul dintre avantajele utilizării DNS. Incertitudinea este că toate aceste servere oferă un set identic de pagini web și nu contează opțiunea clientului, adică pe care dintre servere îl folosește, atâta timp cât îl selectează pe unul aleatoriu în conformitate cu regulile stabilite în DNS SVR.





2. Implementare

2.1 Cerințe de implementare:

Aplicație de tip Zero-config bazată pe mDNS și DNS-SD.

☺ Aplicația va include două moduri de lucru: resolver și responder

☺ Mod responder :

- Prin intermediul interfeței se va putea configura numele echipamentului (folosind domeniul .local)
- Prin intermediul interfeței vor putea fi adăugate/șterse diferite servicii oferite (intrări de tip SRV)

☺ Mod resolver

- Va putea determina adresa IP a unui echipament pe baza numelui din domeniul .local
- Va putea determina toate serviciile disponibile pe un echipament
- Va putea determina toate echipamentele din rețeaua locală care oferă un anumit serviciu



2.2 Descrierea implementării

2.2.1 Interfață

Aplicatie de tip Zero-config bazata pe mDNS si DNS-SD

About Exit

Service Types

ALL TYPES
File Transfer (ftp)
Web Server (http)
Printer (lpr)
Remote Login (ssh)

Search

Host Name

epson.local

GET IP ADDRESS

Service

_ipp._udp.local 5353

ADD REMOVE

OLD Host Name

homespeaker.local

NEW Host Name

boespeaker.local

EDIT

Output

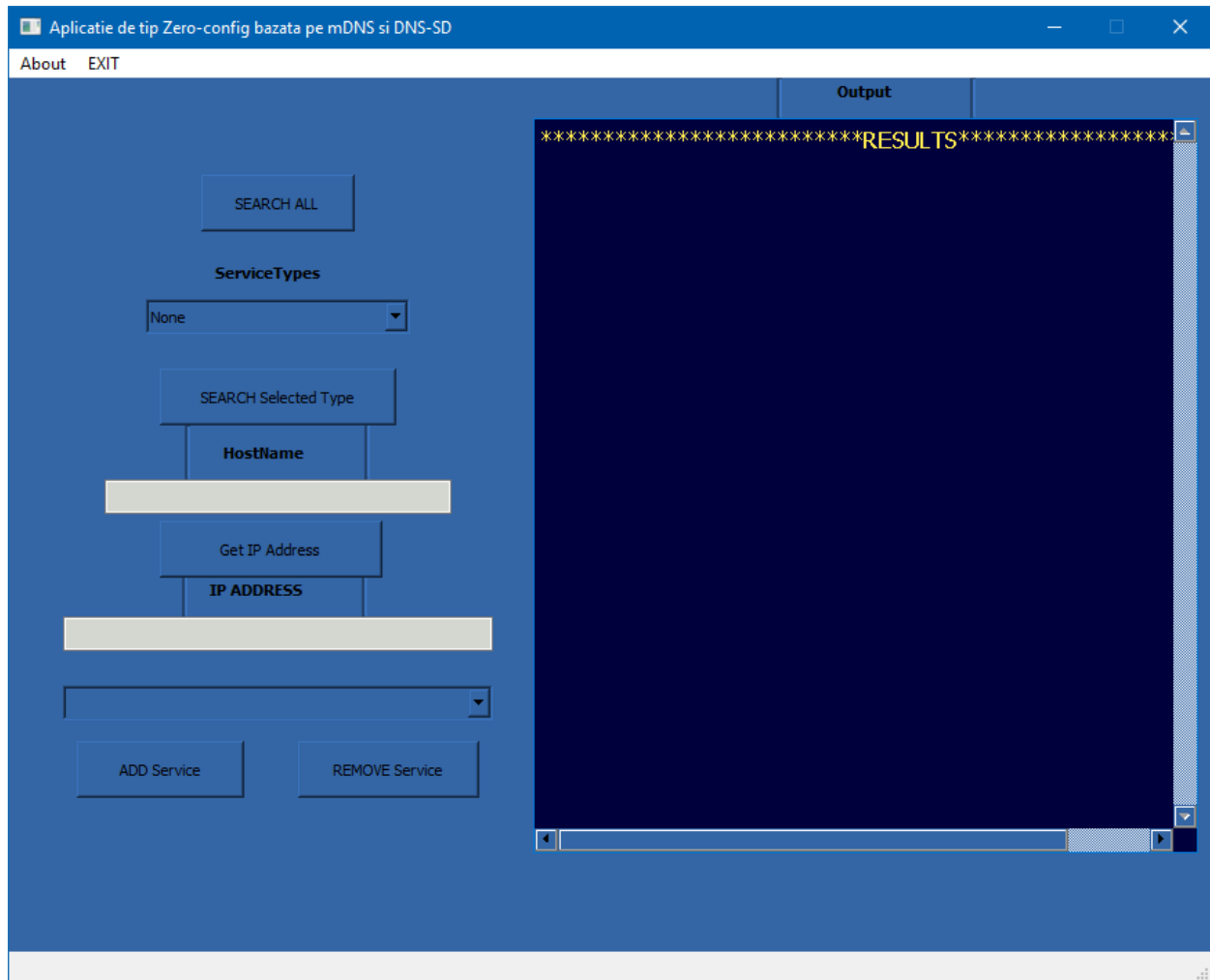
*****RESULTS*****

RECORDS:	FQDN
_ftp._udp.local	smartftp.local
_ipp._udp.local	hp.local
_printer._udp.local	epson.local
_soundtouch._udp.local	homespeaker.local
_freemhand._udp.local	sheetmusic.local
_ssh._tcp.local	linux.local

Searched IP ADDRESS

192.168.56.2

Prototipul interfeței



Interfața finală

Descriere Interfață:

Interfața este compusă dintr-o casetă de output cu toate tipurile de servicii, care se regăsesc în DNS SRV (RFC 2782). Generarea output-ului se va face prin butonul **SEARCH ALL** ("Image 1"). Selectând tipul de serviciu, în urma butonului de **SEARCH Selected Type**, în output vor apărea mai multe informații despre acel serviciu ("Image 2"). Pe lângă caseta **ServiceTypes**, pe interfață se mai regăsește o casetă cu **HostName-ul** a cărui IP ADDRESS dorim să-l aflăm. În urma procesării, se va regăsi o casetă **IP ADDRESS** ce conține adresa IP generată în urma click-ului pe butonul **Get IP Address** corespunzătoare HostName-ului ("Image 4"). De asemenea, cu ajutorul interfeței putem adăuga/elimina diferite servicii și edita numele echipamentelor ("Image 3"/ "Image 5").



Generarea interfeței se va face prin utilizarea limbajului de programare Python(PyQT-software).

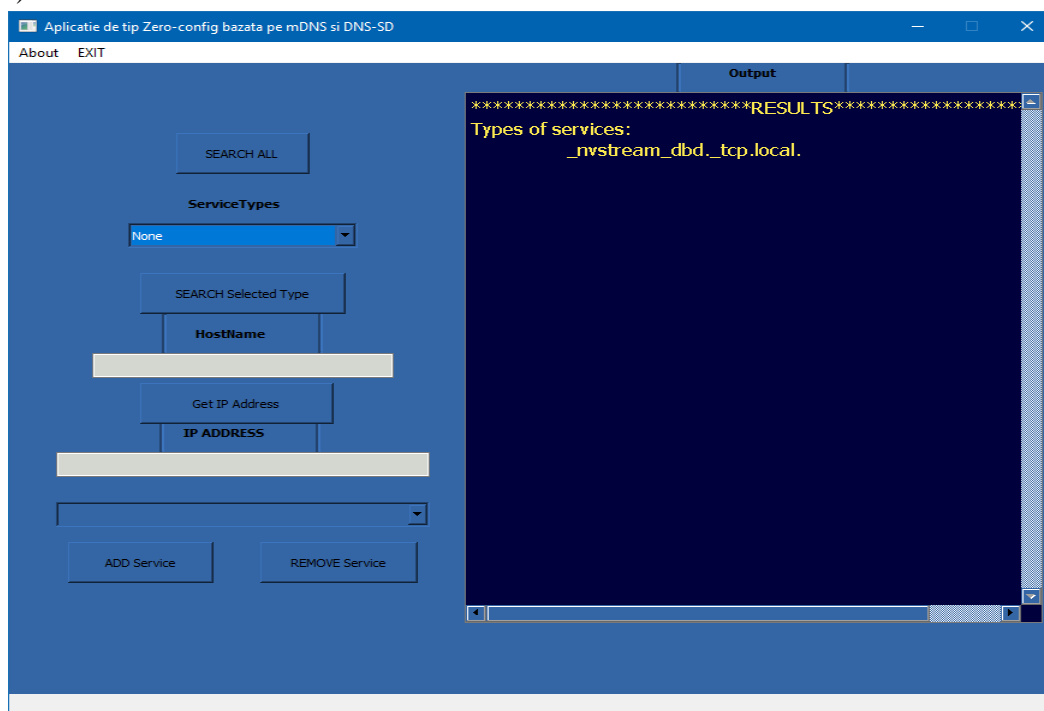


Image 1

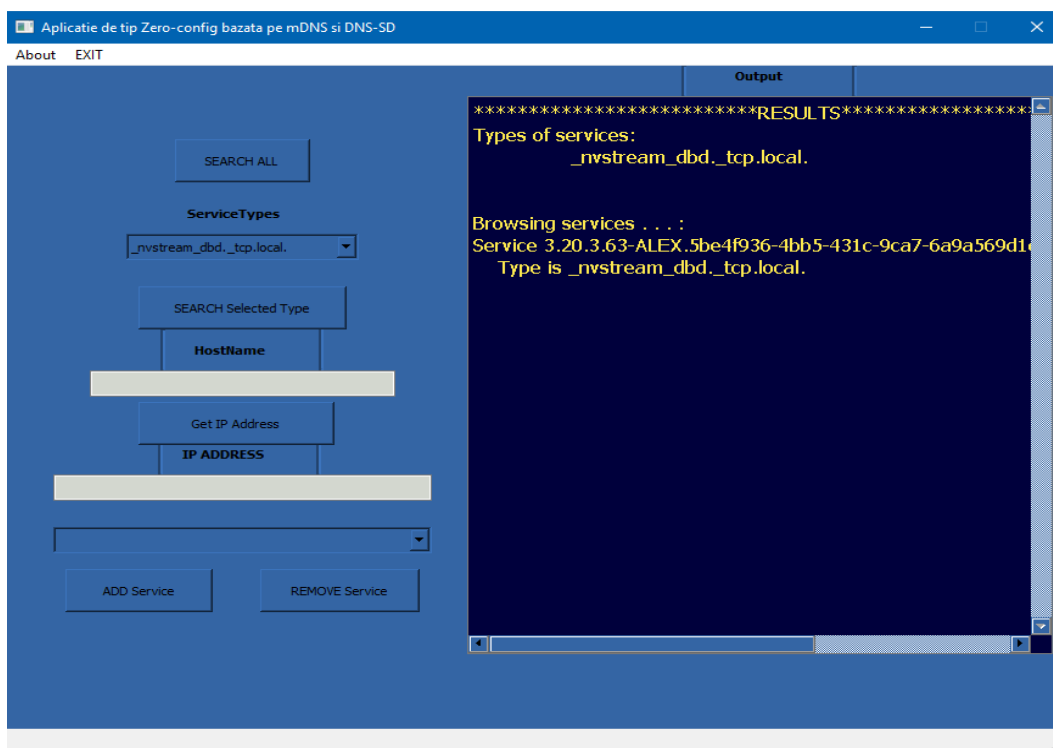


Image 2

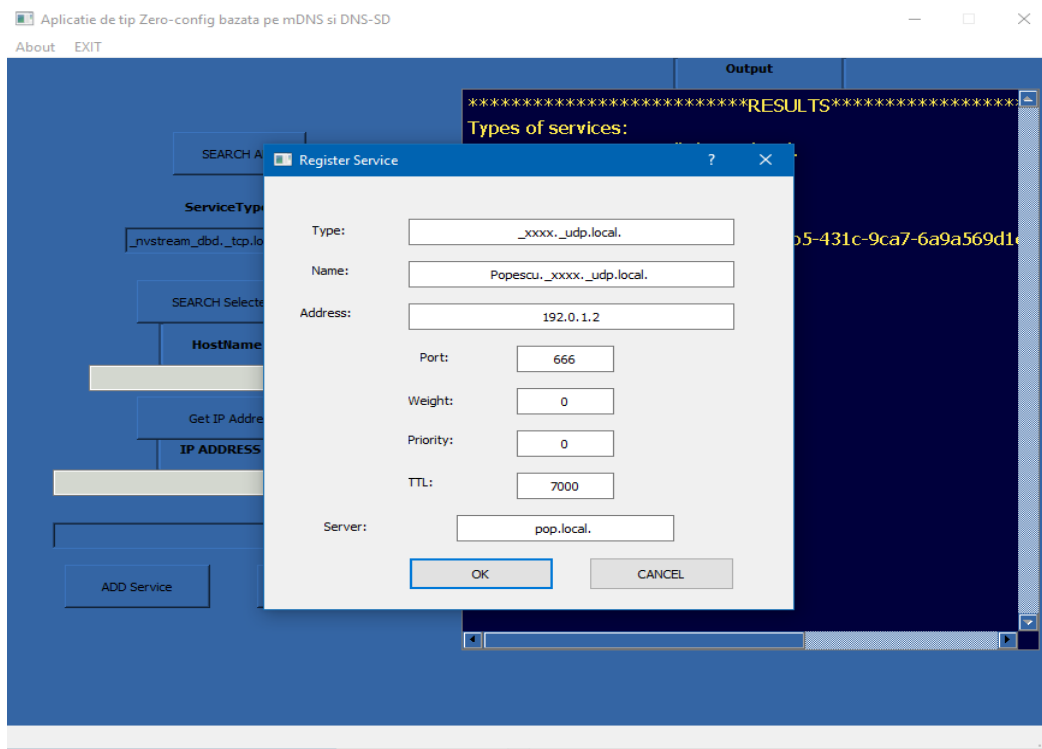


Image 3

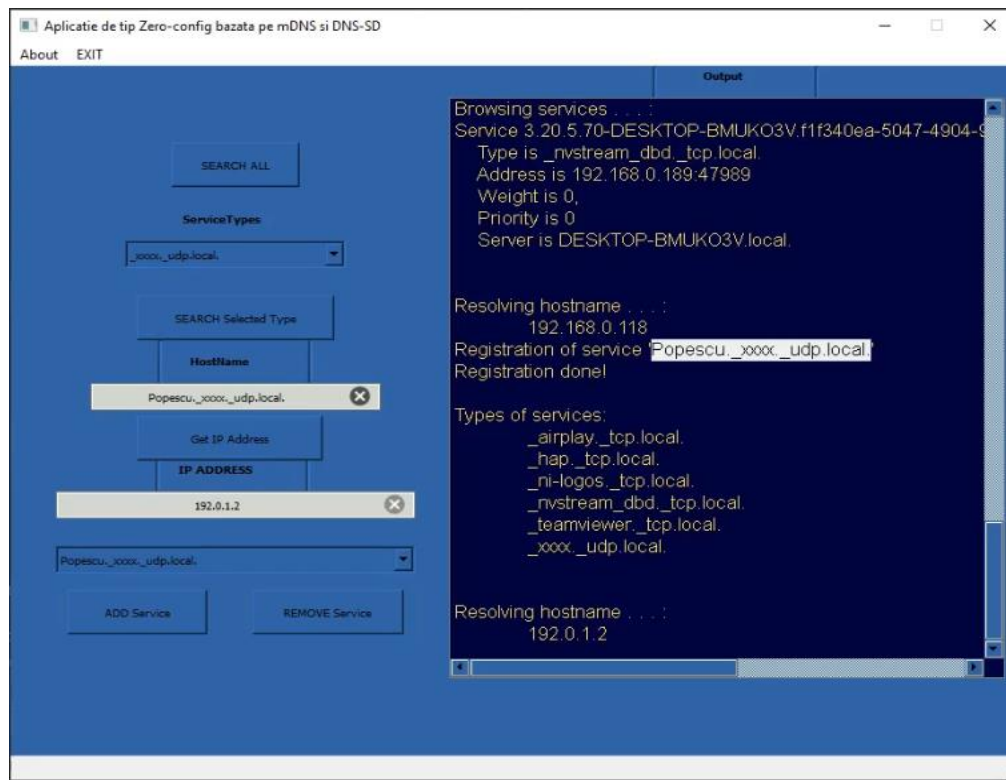


Image 4

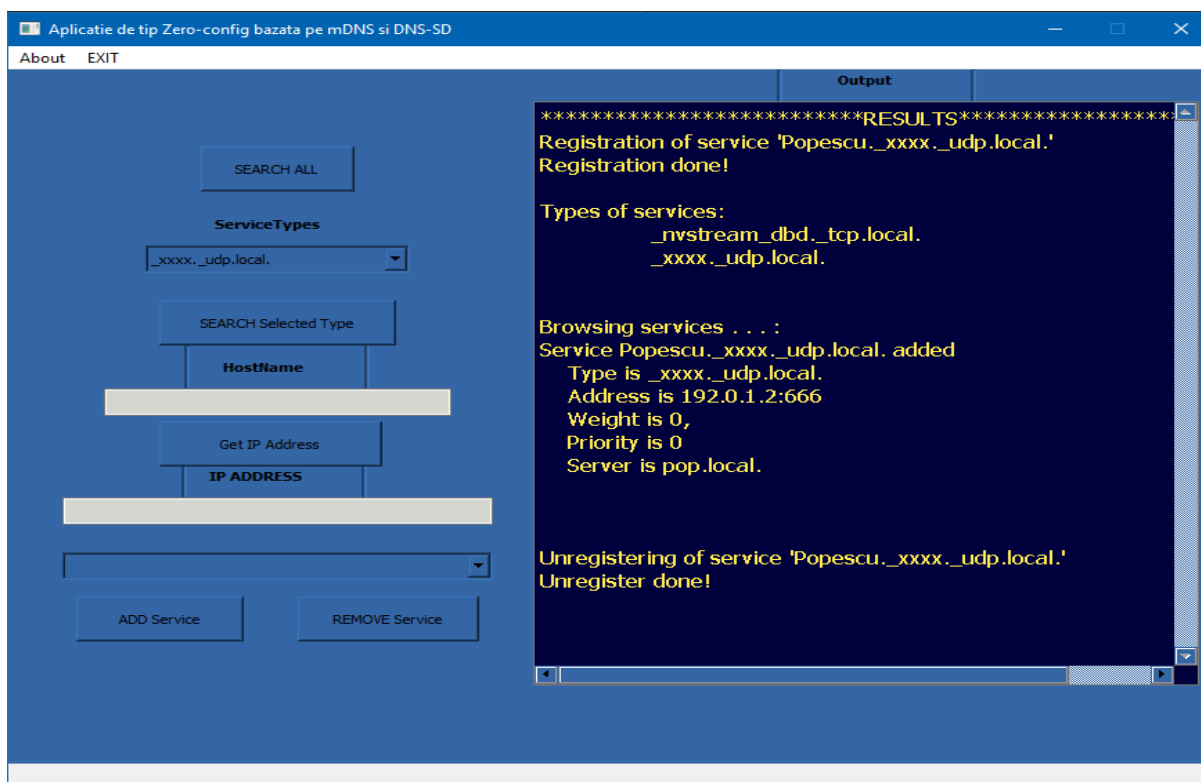


Image 5

2.2.2 Funcții

- **find_ServicesTypes()** – funcție creată într-o clasă Connections, cu ajutorul căreia se face filtrarea serviciilor din sistem.
- **search_SelectedType()** – funcție ce are rolul de a genera informații suplimentare în legătură cu serviciul selectat. În caz de neselectare, funcția returnează un pop-up cu mesajul "Nu ati selectat un tip de serviciu."
- **get_IPaddress()** – funcție ce returnează adresa HostName-ului introdus în câmpul aferent.
- **verify_register()** – verifică sintaxa HostName-ului, în caz de nerespectare a constrângerilor, funcția returnează un pop-up de eroare, iar în caz contrar, va afișa un mesaj pe output de confirmare.
- **remove_Service()** – șterge serviciul selectat. În caz de neselectare, funcția va genera un pop-up de eroare.
- **new_socket()** – crează un nou socket.
- **send()** – funcție de transmitere a datelor pe socket.



2.2.3 Clase

- **DNSEntry**
- **DNSQuestion**
- **DNSRecord** – înregistrare cu TTL.
- **DNSAddres(DNSRecord)** - DNSRecord de tip A.
- **DNSPointer(DNSRecord)** - DNSRecord de tip PTR.
- **DNSText(DNSRecord)** - DNSRecord de tip TXT.
- **DNSService(DNSRecord)** – DNSRecord de tip SRV.
- **DNSIncoming** – pachet de intrare(RESPONSE) cu funcții de **unpack**(se extrage o anumită informație din pachet), **read_header**(se citește header-ul pentru a afla informațiile necesare continuării despachetării), **read_int/unsignedshort/string_characterstring**(citire tip de date), **is_query**(returnează true dacă pachetul este de tip QUERY), **is_response**(returnează true dacă pachetul este de tip RESPONSE), **read_utf8**(se citește un string de o anumită lungime și de la un anumit offset din pachet), **read_domain_name**(se citește numele domeniului), **read_questions**(se citesc întrebările din pachet), **read_other_data**(se citesc alte date din pachet precum răspunsurile).
- **DNSOutgoing** – pachet de ieșire(QUERY) cu funcții de **add_question**(se adaugă o întrebare), **add_answer_at_time**(se pune în pachet un răspuns dacă nu expiră pentru o anumită perioadă de timp), **add_answer/authoritative/additional**(se pune în pachet un răspuns), **pack**(se adaugă un câmp în pachet), **write_byte/short/int/string/utf8**(se scrie un tip de date în pachet BE), **write_domain_name**(se scrie numele domeniului în pachet), **write_question**, **write_record**, **packet**(se împachetează informațiile).
- **DNSCache** – cache-ul pachetelor de intrare cu funcții de **add**, **remove**, **get**.
- **Engine** – restricționează accesul de citire pe socket, permițând obiectelor ce urmează a primi date de pe socket să fie refolosite.
- **Listener** – recepționează ce tip de mesaj DNS sunt trimise, permițând intrarea în cache a informațiilor.
- **Reaper** – șterge intrările expirate din cache.
- **ServiceBrowser** – generează informațiile pentru un serviciu de un anumit tip, cu funcții de **update**, **run** și **cancel**.
- **ServiceInfo** – creează o descriere de serviciu, cu funcții de **set**, **get**, **update** și **request**.



- **Zeroconf** – implementează zero-configuration mDNS-SD.
- **ZeroconfServiceTypes** - generează toate serviciile din domeniul local.

2.2.4 Biblioteci utilizate

- **functools** – este un modul pentru funcții high-order.
- **six** - oferă utilități între Python 2 și Python 3.
- **os** – modul de interacțiune cu sistemul de operare.
- **sys** – modul ce oferă funcții și variabile care sunt utilizate la Runtime.
- **re** – modul ce oferă posibilitatea de a face match pe string-uri.
- **PyQt5** – set cuprinzător de legături Python Qt v5, ajută la crearea legăturilor dintre partea de back-end și front-end.
- **abc** – utilizarea claselor abstracte.
- **logging** -funcții și clase ce implementează un eveniment de logging in sistem.
- **threading** – utilizarea firelor de execuție.
- **typing** – modul ce oferă suport la runtime pentru diferite tipuri de date.
- **socket** – accesul la interfața BSD socket.
- **ipaddress** – modul ce oferă posibilitatea de a crea, manipula și prelucra adrese IPv4 și IPv6.