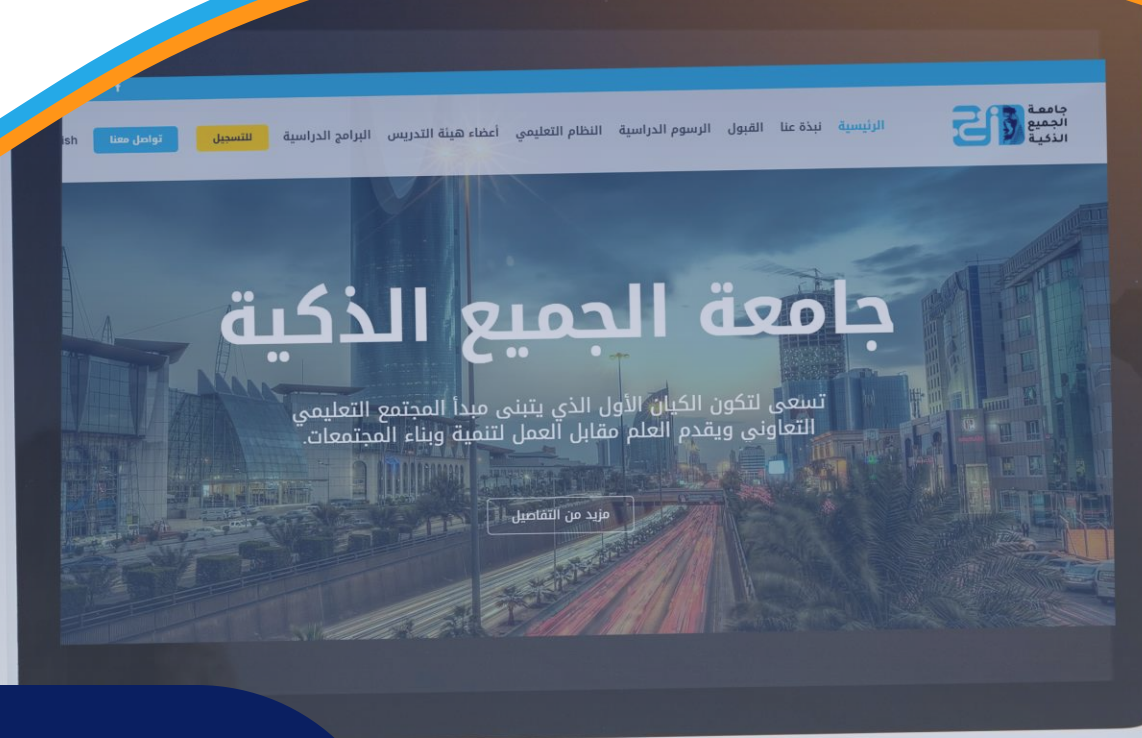




جامعة الجميع الذكية
EVERYONE'S SMART UNIVERSITY

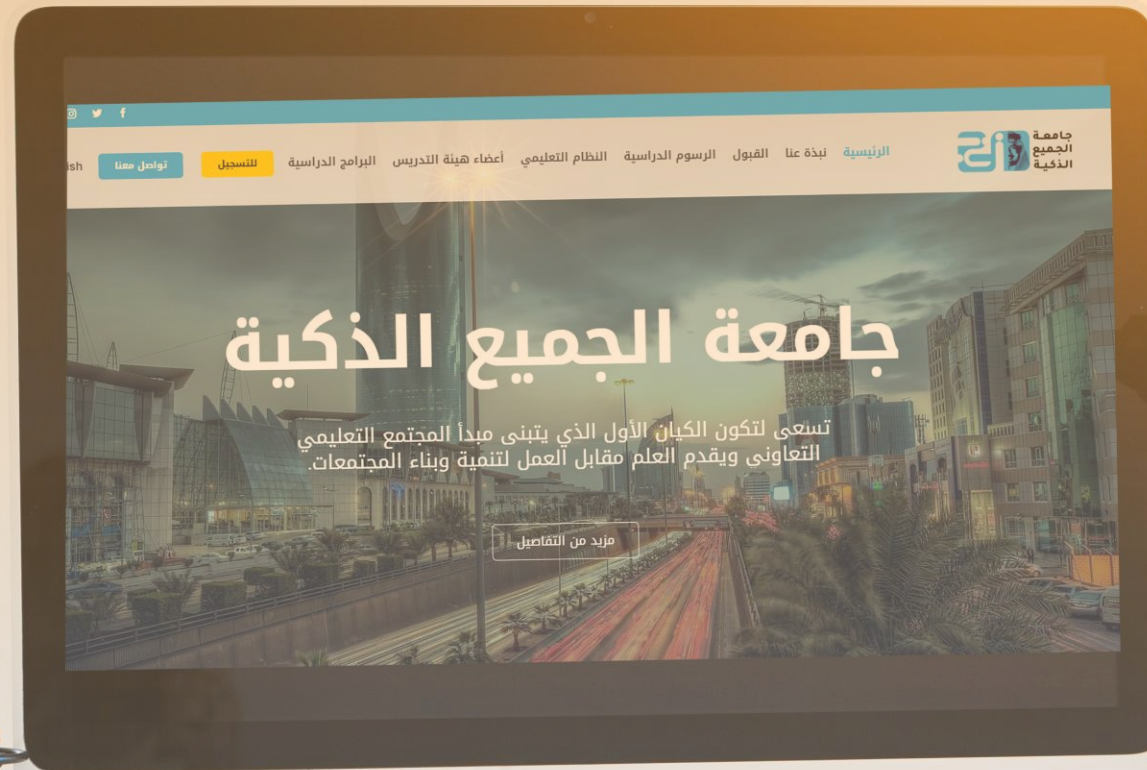


Prepared by
Dr. Hamada I. AbdulWakel
Lecture 7

CYB615
Network Programming



جامعة الجميع الذكية
EVERYONE'S SMART UNIVERSITY



Module5: Socket Names and Domain Name System

- Domain Name System (DNS).
- Client/Server Using Getaddrinfo() method.
- DNS module in Python.



- The Domain Name System (DNS) is the scheme by which millions of internet hosts cooperate to answer the question of what hostnames resolve to which IP addresses.
 - Names are easy to remember.
 - On Windows/Linux
`nslookup google.com`

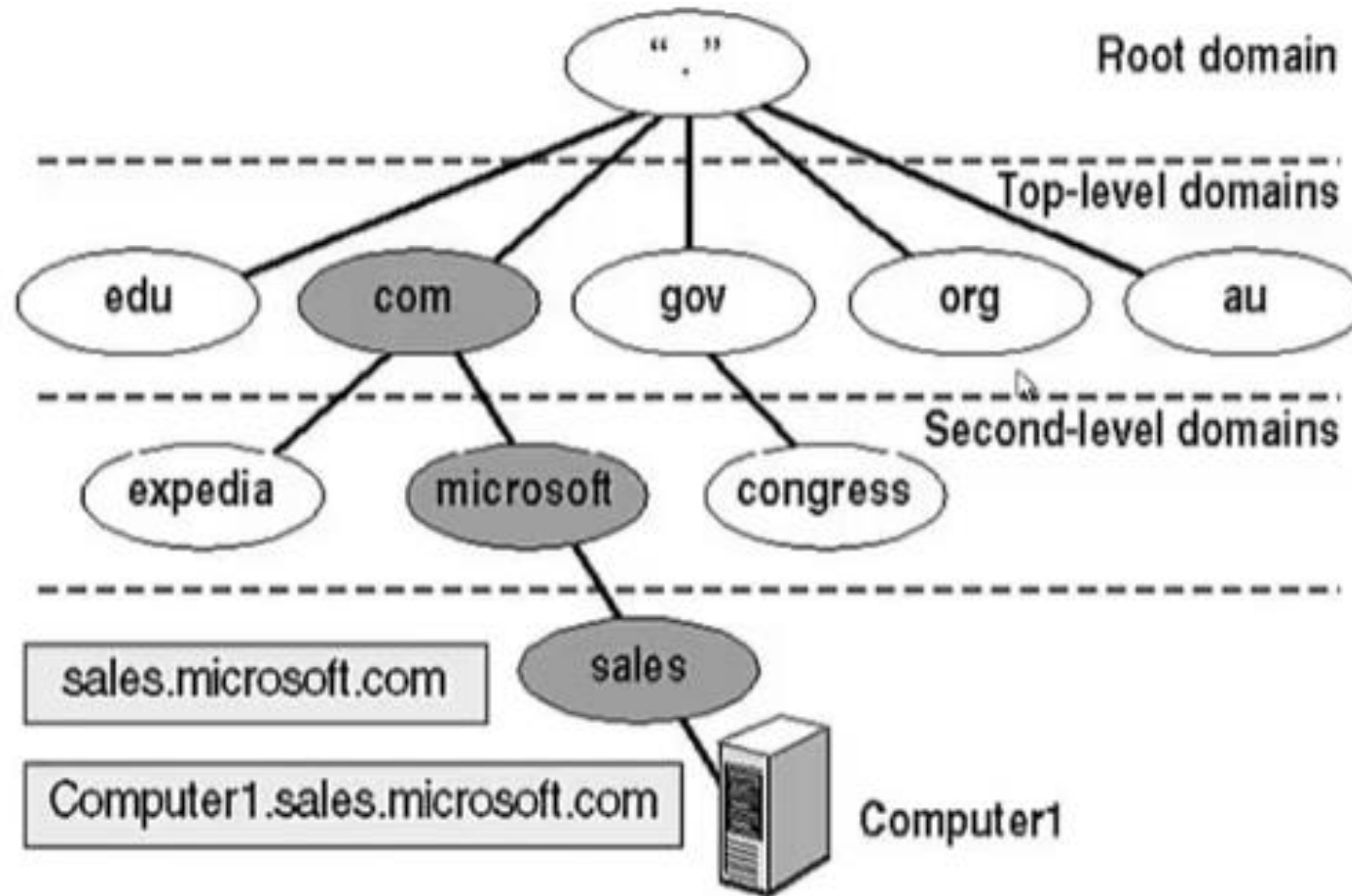
```
drhamada@drhamada-Latitude-E6520:~$ nslookup google.com
Server:          127.0.0.53
Address:         127.0.0.53#53

Non-authoritative answer:
Name:   google.com
Address: 142.250.200.206
Name:   google.com
Address: 2a00:1450:4006:80c::200e
```



Lecture 7

Internal Work of DNS



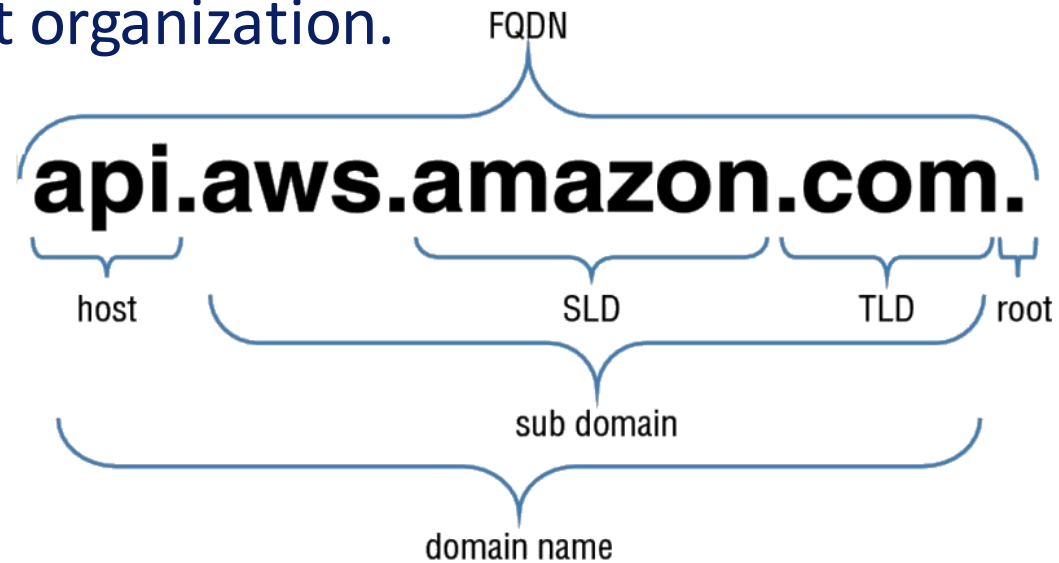
- Domain Names are hierarchical, and each part of a domain name is referred to as either the root, top level, second level or as a sub-domain.
- To allow computers to properly recognize a fully qualified domain name:
 - Dots are placed between each part of the name.
 - All resolvers start their lookups at the root.
 - The resolver navigates its way down the tree until it gets to the left-most part of the domain name.



- **Top-level domain (TLD):** the few hundred strings like **com**, **edu**, **net**, **org**, **gov** that together with country code like **eg** form the set of possible suffixes for valid domain names.
 - Each TLD has its own set of servers and organization that oversees granting ownership to domains.
- **Domain name:** the name that an organization appends as a suffix to its sites and hosts on the internet like **python.org** or **bbc.co.uk**.



- **Fully qualified domain name (FQDN):** names an internet site or host by appending its organization's full domain name to the name of a particular machine in that organization.



- **The address family:** AF_INET, AF_INET6.
- **Socket type:** SOCK_STREAM, SOCK_DGRAM.
- **The protocol:** has the value 0 to force it to be chosen automatically.
- **The fourth and fifth values:** used to make a connection are the IP address and port number.



- To able to use `AF_INET6` with socket object, you should first test if the python supports IPv6 by checking the `has_ipv6`.

```
import socket
```

```
print(socket.has_ipv6)
```



- `Getaddrinfo()` returns a list of tuples for a given network service with which socket objects can be created to connect to that service.

`socket.getaddrinfo(host, port, family=0, type=0, proto=0, flags=0)`

- Host: IP address or host name.
- Port: could be a number or service name like http and www.
- Family: IP version.
- Type: Stream or Datagram.
- Proto: TCP or UDP.
- Flags: used for filtering the addresses.



Lecture 7

getaddrinfo() Method Example

```
import socket
from pprint import pprint
infolist = socket.getaddrinfo('google.com', 'www')
pprint(infolist)
info = infolist[0]
print(*info[0:2])
sok = socket.socket(*info[0:2])
print(info[4])
sok.connect(info[4])
```



- If you want an address to provide to `bind()`, then you will call `getaddrinfo()` with `None` as the hostname.

```
Infolist = socket.getaddrinfo(None, 'www', socket.AF_INET)
```



Lecture 7

Server Creation Using getaddrinfo()

```
import socket
infolist = socket.getaddrinfo(None, 'www', socket.AF_INET)
info = infolist[0]
s= socket.socket(*info[0:2])
s.bind(info[4]); s.listen();
con, addr = s.accept()
con.send("Server: Hello".encode())
msg = con.recv(1024).decode()
print(msg)
```



Lecture 7

Client Creation Using getaddrinfo()

```
import socket
infolist = socket.getaddrinfo(None, 'www', socket.AF_INET)
info = infolist[0]
s= socket.socket(*info[0:2])
s.connect(info[4])
msg = s.recv(1024).decode()
print(msg)
s.send("client: I received your msg".encode())
```



- Call `getaddrinfo()` with `AI_ADDRCONFIG` flag, which filters out any addresses (IPv6 especially) that are impossible for your computer to reach.
- If the service that you are connecting supports only IPv4, you will also want to specify `AI_V4MAPPED` to return the IPv4 addresses



```
import socket
```

```
print(socket.getaddrinfo('fedoraproject.org', 'ftp'))
```

```
print(socket.getaddrinfo('fedoraproject.org',  
                          'ftp',  
                          socket.AF_INET,  
                          socket.SOCK_STREAM,  
                          0,  
                          socket.AI_ADDRCONFIG | socket.AI_V4MAPPED))
```



- `socket.gethostname()`: return local host name
- `socket.gethostbyname('cern.ch')`: return IP
- `socket.gethostbyaddr('127.0.0.1')`: return host name
- `socket.getprotobyname('UDP')`: return a number for this protocol
- `socket.getservbyname('www')`: return port number for this service
- `socket.getservbyport(80)`: return service name for this port



- Commonly used record types:
 - **A**: host address
 - **AAAA**: IPv6 host address
 - **CNAME**: (Canonical Name for an alias) is a type of record in the DNS used to map a domain name as an alias for another domain. CNAME records always point to another domain name and never directly to an IP address.



- Commonly used record types:
 - **MX**: also called mail exchanger record is a resource record in the DNS that specifies a mail server responsible for accepting email messages on behalf of a recipient's domain.
 - **NS** (Name Server): directs to name servers, where to ask if you want know about a subdomain.
 - **TXT** (Descriptive Text): lets the administrator insert into the DNS record.



- Install DNS corresponding module.

```
pip install dnspython3
```



```
import dns.resolver as dns

def lookup(name):
    NSrecord = dns.query(name, 'NS')
    For ns in NSrecord:
        print(ns)

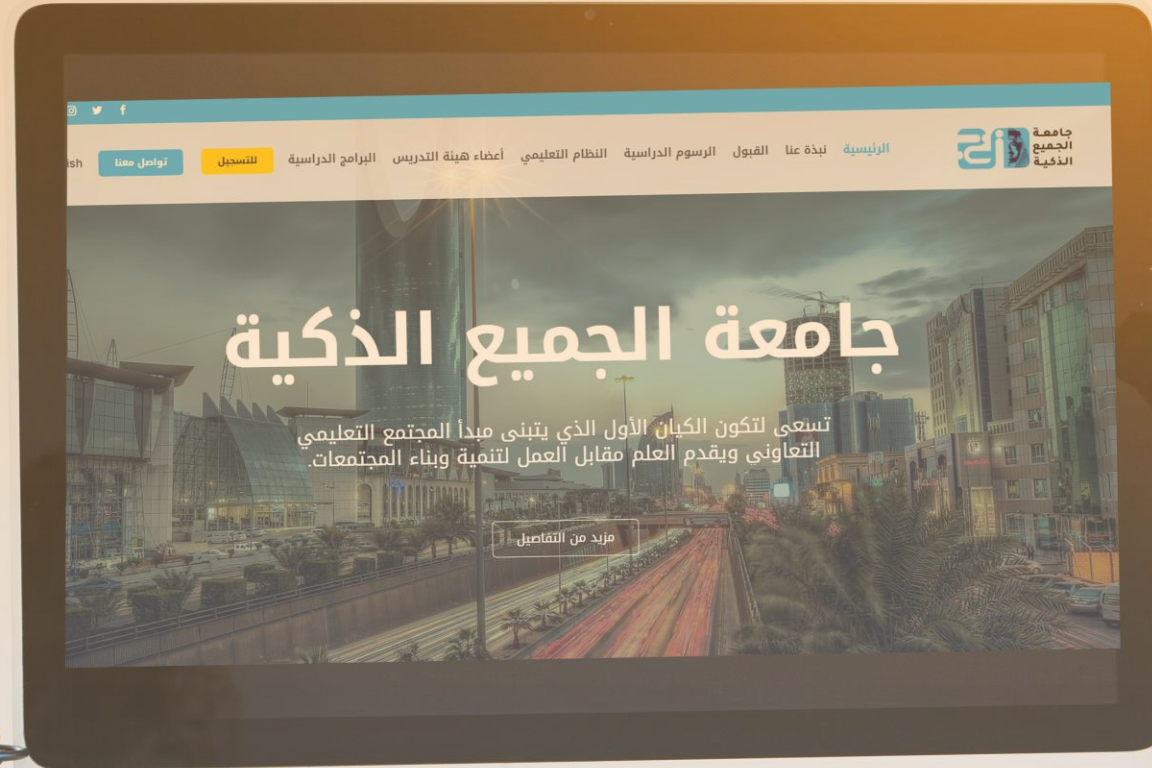
lookup('google.com')
lookup('matrix219.com')
```



```
import dns.resolver
def lookup(name):
    for qtype in 'A', 'AAAA', 'CNAME', 'MX', 'NS':
        answer = dns.resolver.query(name, qtype, raise_on_no_answer=False)
        for val in answer:
            print(val)

lookup('google.com')
```





جامعة الجميع الذكية
EVERYONE'S SMART UNIVERSITY

THANK YOU