گزارش نهایی پروژه شبکه هوشمند دانشگاه

۱ .معرفی پروژه

هدف این پروژه طراحی و پیاده سازی یک شبکه دانشگاهی چند بخشی در Cisco Packet Tracer است شبکه از چهار ناحیه اصلی تشکیل شده است و مسیریابی بین ناحیه ها با OSPF انجام می شود سرویس های مرکزی در بخش Data Center قرار دارند و کاربران در شبکه های Academic و Admin و Library به این سرویس ها دسترسی دارند

در بخش تکمیلی پروژه یک سامانه مدیریت شبکه روی Management Console پیاده سازی شده است این سامانه وضعیت DNS و Web Serverو Mail Server و File Server و DHCP را بررسی می کند و نتیجه هر سرویس را به صورت Online و Offlineنمایش می دهد در بخش Bonus نیز یک سامانه Disaster Recovery مبتنی بر IoT ایجاد شده است که دما و دود را پایش می کند و در شرایط بحرانی تجهیزات ایمنی را به صورت خودکار کنترل می کند

۲ .اهداف پروژه

ایجاد چهار LAN مستقل برای بخش های Academic و Admin و Library وData Center

ایجاد چهار Router با مسیرهای افزونه و پروتکلOSPF

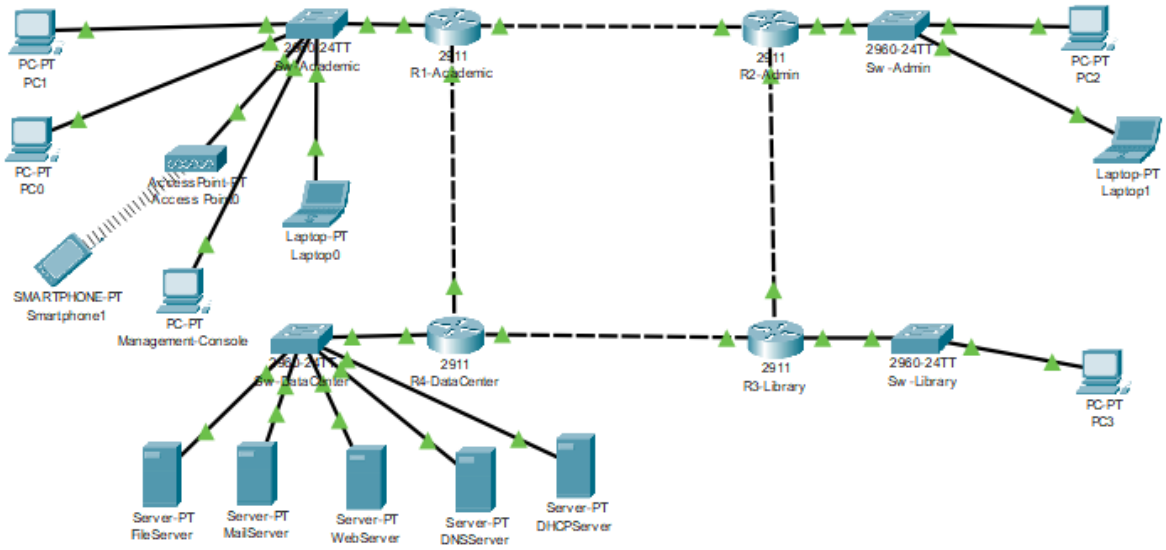
راه اندازی DHCP و DNS و Web و Mail وFTP

تست ارتباط بین کاربران و سرورها در چند LAN متفاوت

پیاده سازیUniversity Network Management Console

پیاده سازی سناریوی اضطراری و بازیابی خودکار با MCU و حسگرها و عملگرهایIoT

۳ .نمای کلی توپولوژی



۴ .طرح آدرس دهی شبکه

برای ساده شدن مدیریت و جداسازی منطقی بخش ها برای هر LAN یک شبکه خصوصی مستقل در نظر گرفته شده است لینک های بینRouter ها نیز با شبکه های کوچک نقطه به نقطه پیکربندی شده اند

Gateway	شبکه	بخش
192.168.10.1	192.168.10.0/24	Academic
192.168.20.1	192.168.20.0/24	Admin
192.168.30.1	192.168.30.0/24	Library
192.168.40.1	192.168.40.0/24	Data Center

۵ .آدرس لینک های بینRouter ها

سمت دوم	سمت اول	لینک
10.0.12.2/30	10.0.12.1/30	R2بهR1
10.0.23.2/30	10.0.23.1/30	R3بهR2
10.0.34.2/30	10.0.34.1/30	R4بهR3
10.0.41.2/30	10.0.41.1/30	R1بهR4

۶ .سرورهایData Center

کاربرد	IP	سرور
تخصیص خودکارIP	192.168.40.10	DHCP Server
نام دامنه و تبدیل نام بهIP	192.168.40.11	DNS Server
وب سایت دانشگاه	192.168.40.12	Web Server
SMTPوPOP3	192.168.40.13	Mail Server
FTPو انتقال فایل	192.168.40.14	File Server

۷. تنظیم DHCP و DNS

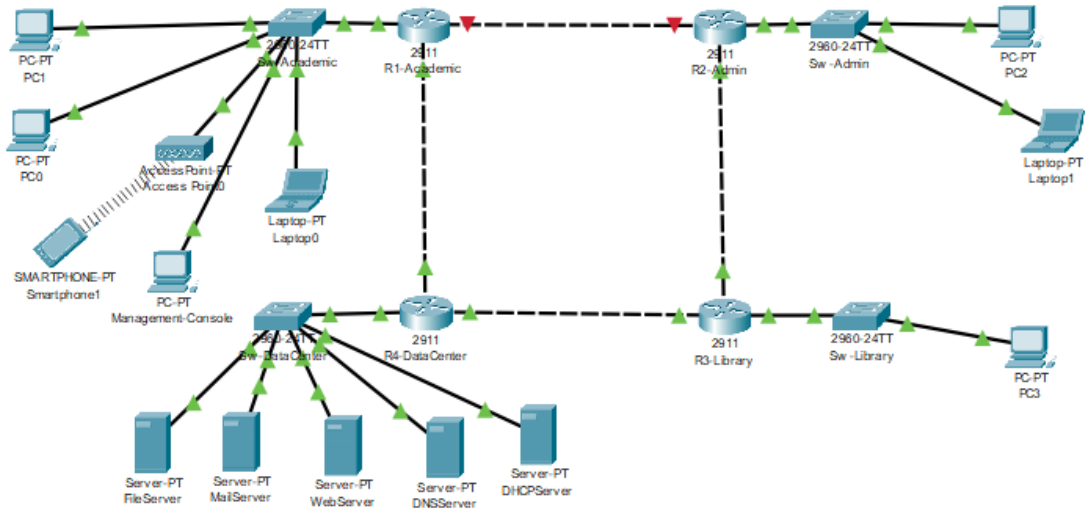
DHCP Serverبرای شبکه های Academic و Admin و Library Pool مستقل داردRouter های R1 و R2 و R3 از ip helper addressبرای انتقال درخواست DHCP به سرور مرکزی استفاده می کنند DNS Server نیز رکوردهای www.university.localو files.university.local را برای Web Server و File Server نگهداری می کند دامنه سرویس ایمیل university.comدر نظر گرفته شده است

Pool	Start IP	Gateway	DNS
Academic	192.168.10.100	192.168.10.1	192.168.40.11
Admin	192.168.20.100	192.168.20.1	192.168.40.11
Library	192.168.30.100	192.168.30.1	192.168.40.11

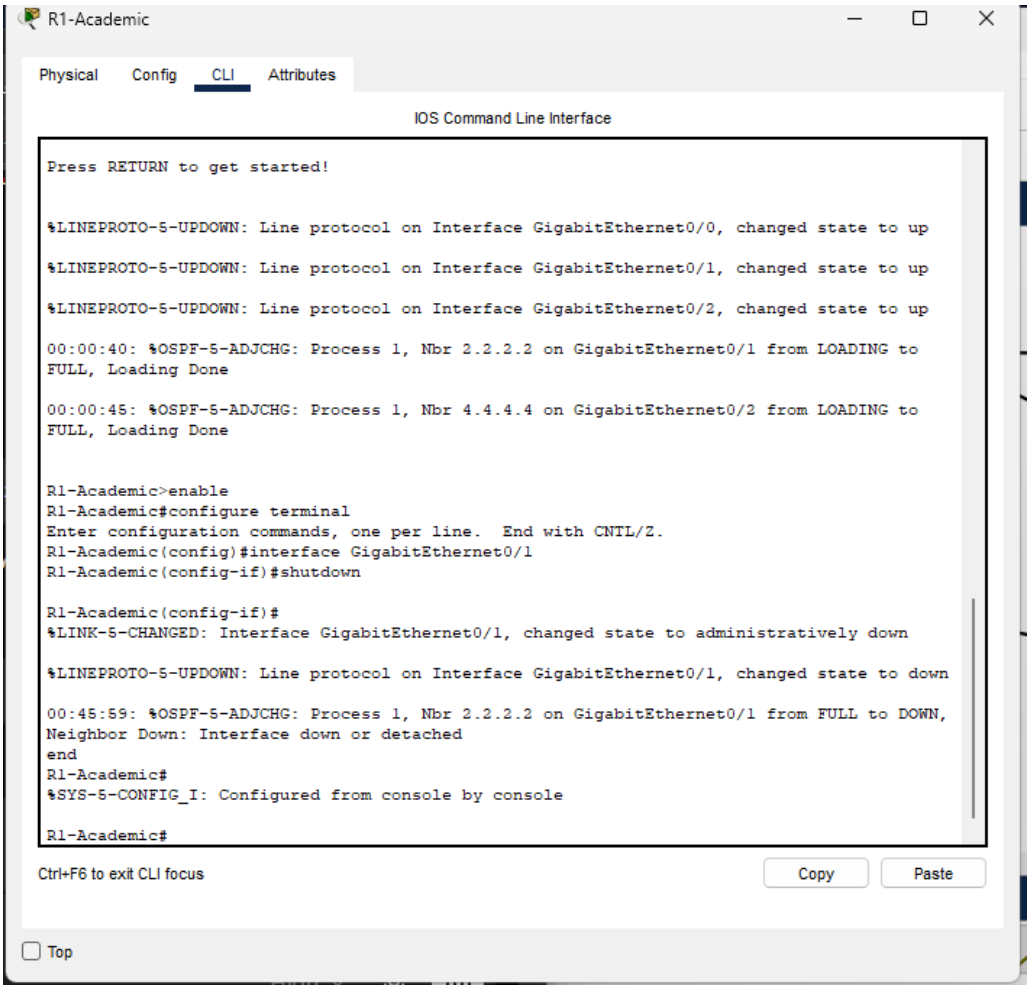
نام سرویس	رکورد	IP
Web	www.university.local	192.168.40.12
File	files.university.local	192.168.40.14
Mail	mail.university.com	192.168.40.13

۸ .مسیریابی OSPF و افزونگی

هر چهار Router در Area 0 قرار دارند ساختار حلقه ای بینRouter ها دو مسیر ممکن بین ناحیه ها ایجاد می کند در نتیجه در صورت قطع شدن یکی از لینک های بینRouter ها شبکه می تواند از مسیر جایگزین استفاده کند



تصویر ۲. قطع لینک مستقیم R1 و R2 برای تستFailover



تصویر ۳ .تغییر وضعیت همسایه OSPF پس از Shutdown شدن لینک R1 و R2

۹. نتیجه تستFailover

پس از قطع لینک مستقیم R1 و R2 ارتباط Management Console با شبکه Admin ادامه پیدا کرد Traceroute نشان داد ترافیک از مسیر R1 به R4 سپس R3 و در نهایت R2 عبور کرده است این آزمون نشان می دهد OSPF توانسته است مسیر جایگزین را به درستی انتخاب کند

آدرس	مرحله
192.168.10.1	Gatewayشبکه Academic
10.0.41.1	در مسیر جایگزین R4
10.0.34.1	در مسیر جایگزین R3
192.168.20.1	Gatewayشبکه Admin

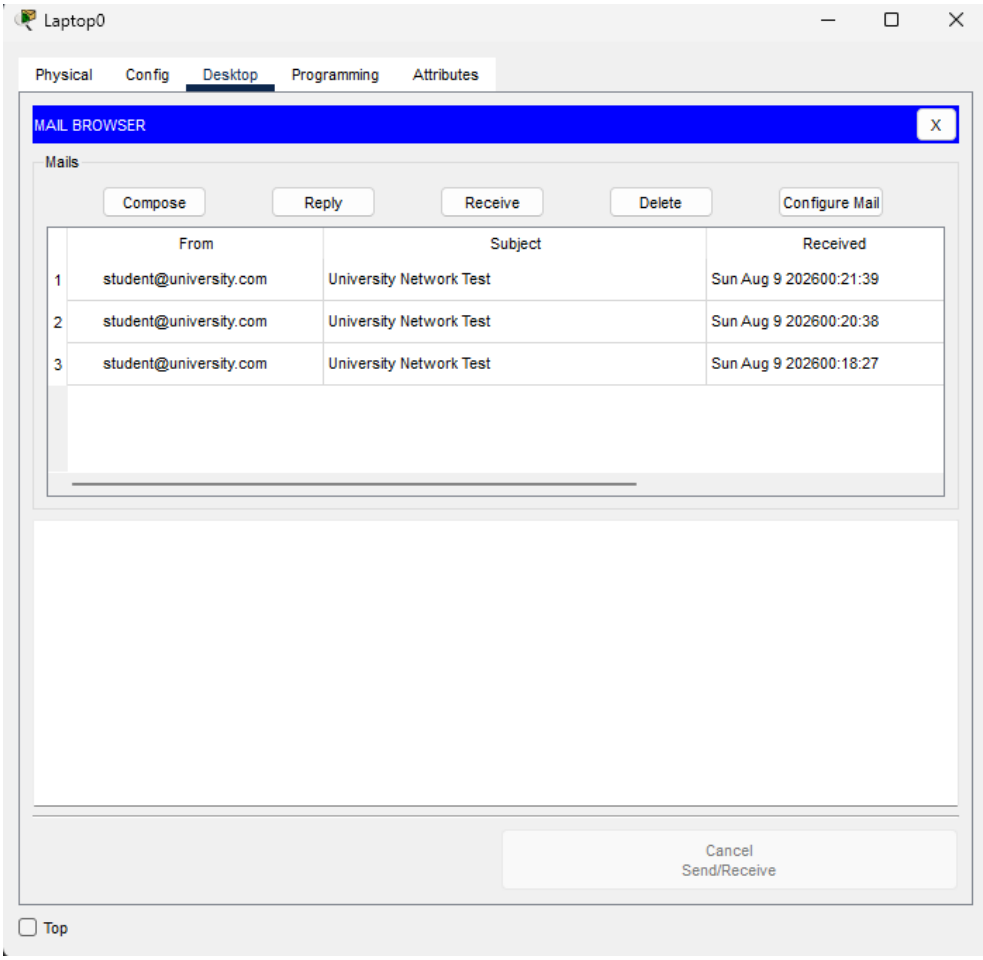
```
C:\>ping 192.168.20.1

Pinging 192.168.20.1 with 32 bytes of data:

Reply from 192.168.20.1: bytes=32 time=11ms TTL=252
Reply from 192.168.20.1: bytes=32 time<1ms TTL=252
Reply from 192.168.20.1: bytes=32 time=11ms TTL=252
Reply from 192.168.20.1: bytes=32 time=11ms TTL=252

Ping statistics for 192.168.20.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 11ms, Average = 8ms

C:\>tracert 192.168.20.1

Tracing route to 192.168.20.1 over a maximum of 30 hops:

  1   0 ms      2 ms      1 ms      192.168.10.1
  2   0 ms      0 ms      11 ms     10.0.41.1
  3   10 ms     0 ms      30 ms     10.0.34.1
  4   0 ms      17 ms     11 ms     192.168.20.1

Trace complete.

C:\>
```

تصویر ۴ Ping .موفق و Traceroute از مسیر جایگزین

پس از پایان تست لینک R1 و R2 دوباره فعال شد و همسایگی OSPF به وضعیت FULL بازگشت

۱۰. سرویس ایمیل

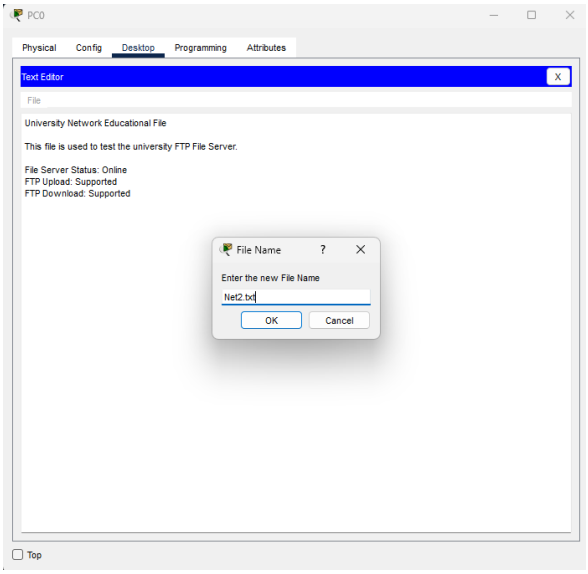
Mail Serverبا آدرس 192.168.40.13 راه اندازی شده است SMTP و POP3 فعال هستند دامنه ایمیل university.com تنظیم شده است و حساب های student و faculty برای آزمایش ارسال و دریافت ایمیل ساخته شده اند



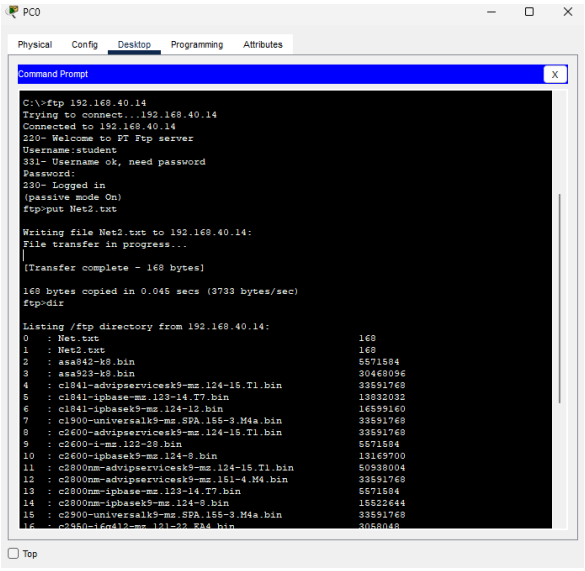
تصویر ۵. دریافت موفق ایمیل در کاربر Faculty

۱۱. سرویس FTP و File Server

File Serverبا آدرس 192.168.40.14 سرویس FTP را ارائه می کند کاربر student اجازه Read و Write و Delete و Rename و List دارد برای اثبات عملکرد سرویس یک فایل آموزشی روی PC0 ساخته شد و سپس از طریق FTP روی سرور بارگذاری شد



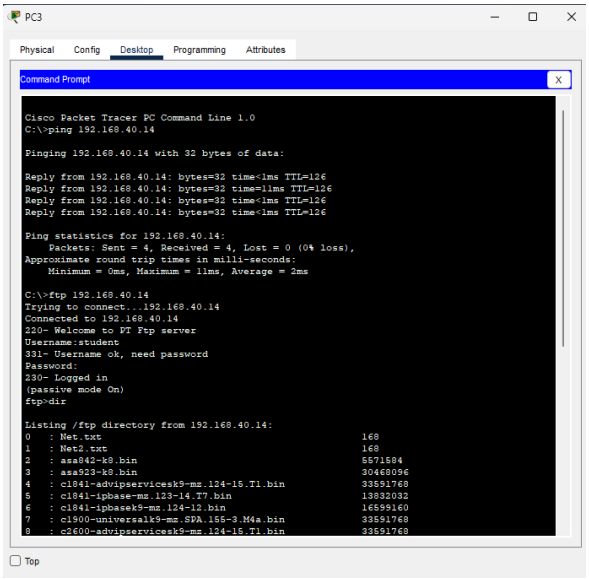
تصویر ۶ .ساخت فایل Net2.txt روی PC0



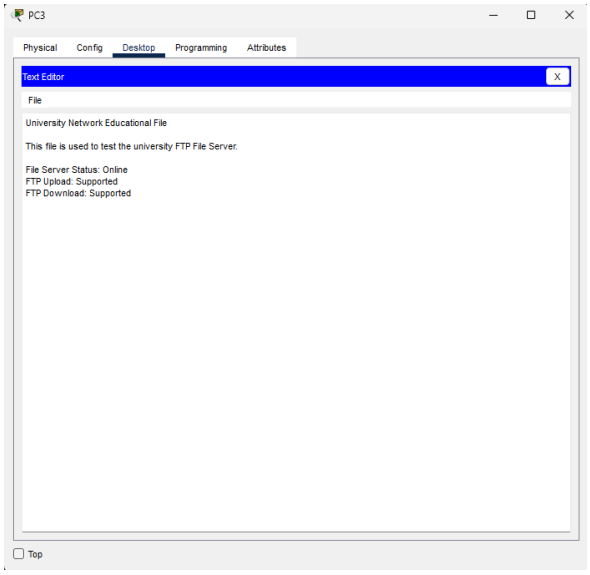
تصویر ۷ .ارسال فایل Net2.txt به FTP Server

۱۲ .آزمون FTP بین دو LAN

برای اثبات کارکرد سرویس در چند شبکه متفاوت PC3 در شبکه Library به File Server متصل شد Ping به 192.168.40.14 موفق بود و ورود به FTP Server نیز انجام شد سپس فایل متنی دریافت شد و محتوای آن در Text Editor مشاهده شد



تصویر ۸ .اتصال PC3 از شبکه Library به FTP Server

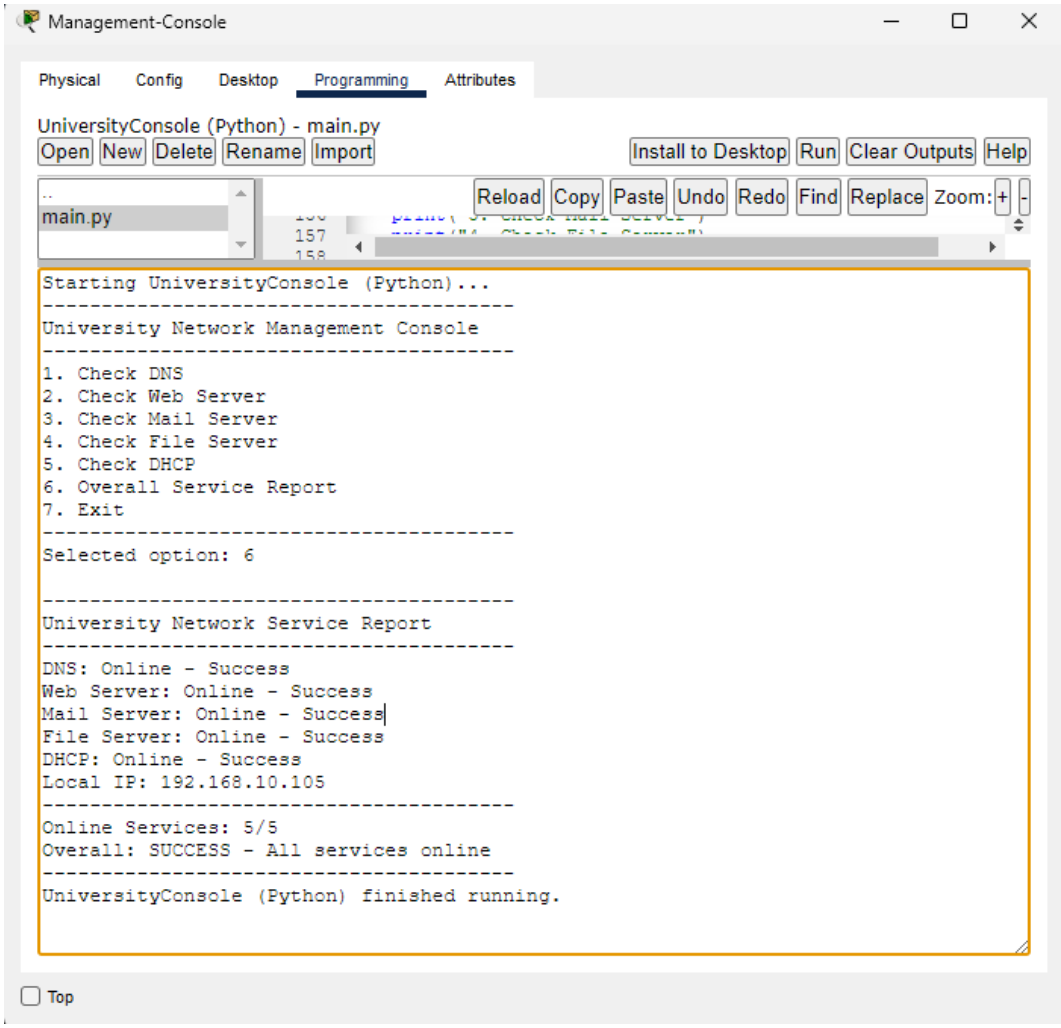


تصویر ۹ .مشاهده فایل دریافت شده درPC3

۱۳. University Network Management Console

Management Consoleدر شبکه Academic قرار دارد و از DHCP آدرس دریافت می کند برنامه Python روی این End Deviceوضعیت پنج سرویس اصلی را بررسی می کند برای Web و DNS از HTTP Test استفاده شده است برای Mail و File Serverاتصال TCP به پورت های سرویس بررسی می شود وضعیت DHCP نیز از روی آدرس محلی دستگاه کنترل می شود

منوی برنامه شامل بررسی جداگانه DNS و Web Server و Mail Server و File Server و DHCP است گزینه Overall Service Reportتمام سرویس ها را پشت سر هم بررسی می کند و در پایان یک گزارش کلی نمایش می دهد



تصویر ۱۰. خروجی Management Console با نتیجه 5 از 5 و وضعیت کلیSuccess

در آزمون نهایی هر پنج سرویس Online بودند آدرس Management Console برابر 192.168.10.105 بود و نتیجه Overall برابر SUCCESS ثبت شد

IoT .۱۴

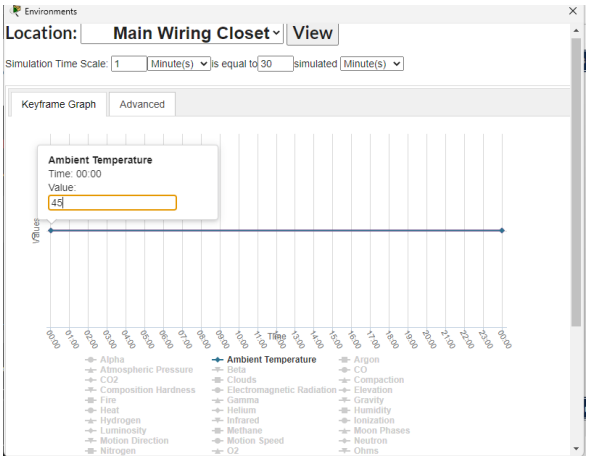
برای بخش Bonus از MCU PT استفاده شده است Temperature Sensorو Smoke Sensor ورودی های سامانه هستند Red LEDو Green LED و Alarm و Fan و Door و LCD به عنوان خروجی های ایمنی به MCU متصل شده اند

نقش	پینMCU	تجهیز
اندازه گیری دما	A0	Temperature Sensor
تشخیص دود	A1	Smoke Sensor
هشدار بصری بحران	D0	Red LED
نمایش حالت عادی	D1	Green LED
هشدار صوتی	D2	Alarm
تهویه اضطراری	D3	Fan
باز شدن خروج اضطراری	D4	Door
نمایش وضعیت سیستم	D5	LCD

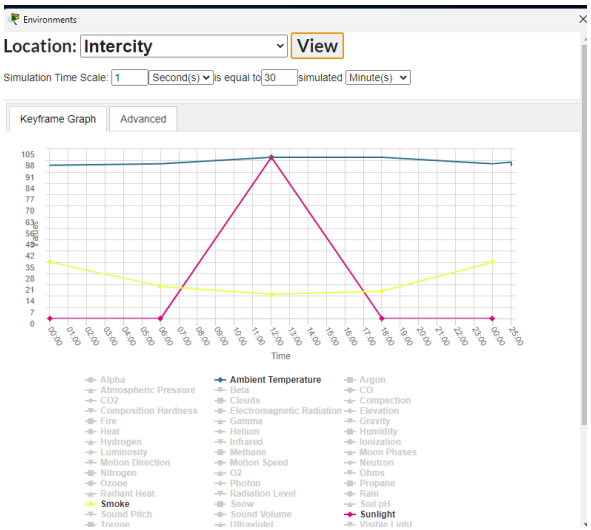
منطق سامانه به این صورت است که اگر دما بیشتر از 45 درجه باشد و همزمان دود تشخیص داده شود حالت Emergency فعال می شود در این حالت Alarm روشن می شود Red LED روشن می شود Green LED خاموش می شود Fan روشن می شود Door باز می شود و LCD پیام اضطراری را نمایش می دهد

۱۵ .تنظیم شرایط محیطی و تست حسگرها

مقادیر محیطی از بخش Environments در Physical View تنظیم شدند ابتدا مقدار Ambient Temperature روی 45 درجه تست شد سپس برای سریع تر شدن آزمون سناریوی بحرانی نمودار دما در سطح Intercity افزایش داده شد و Smoke نیز بالاتر از صفر قرار گرفت



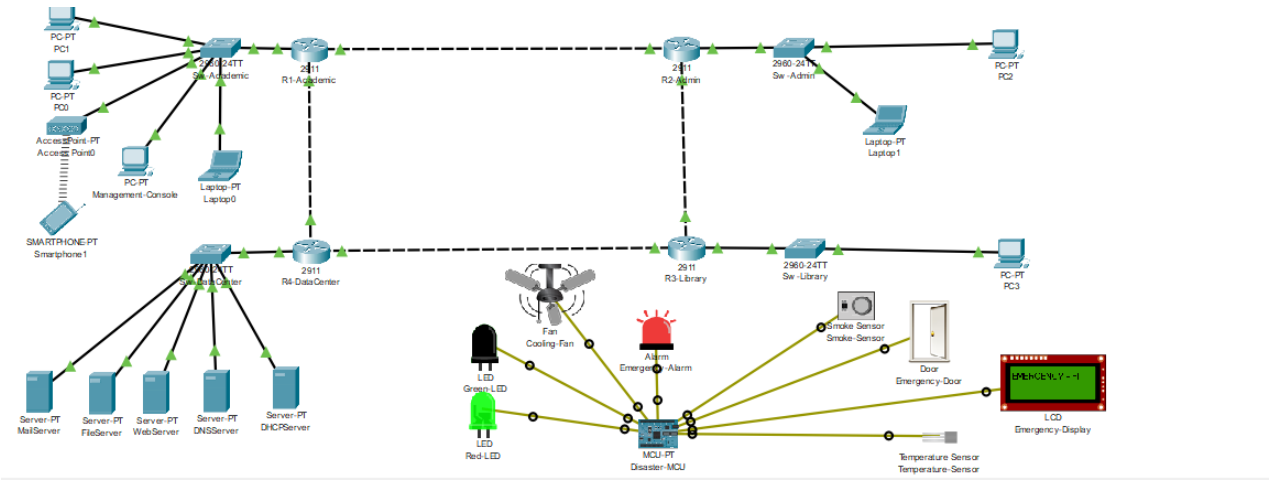
تصویر ۱۱ .تنظیم Ambient Temperature روی 45 درجه



تصویر ۱۲ .نمودار دما و Smoke در محیطIntercity

Temperature Sensorمقدار آنالوگ را روی A0 و Smoke Sensor مقدار آنالوگ را روی A1 تولید می کند با افزایش دما مقدار خام دما افزایش پیدا کرد و مقدار Smoke نیز از صفر خارج شد این تست صحت اتصال سنسورها به MCU را نشان داد

۱۶ .فعال شدن حالتEmergency



تصویر ۱۳ .وضعیت تجهیزات در حالت دمای بالا و وجودSmoke

```
--------------------------------
EMERGENCY DETECTED
Temperature: 100 C
Smoke Raw: 8
Alarm: ON
Red LED: ON
Green LED: OFF
Fan: ON
Door: OPEN
Display: EMERGENCY
--------------------------------
```

تصویر ۱۴ .خروجی برنامه در حالت Emergency با دمای بالا وSmoke

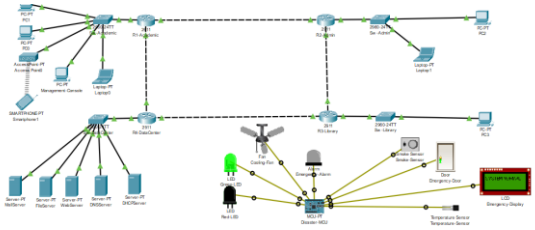
```
--------------------------------
--------------------------------
EMERGENCY DETECTED
Temperature: 48 C
Smoke Raw: 417
Alarm: ON
Red LED: ON
Green LED: OFF
Fan: ON
Door: OPEN
Display: EMERGENCY
--------------------------------
```

تصویر ۱۵ .نمونه خروجی Emergency با Smoke فعال

در حالت Emergency خروجی برنامه شامل EMERGENCY DETECTED است Alarm و Red LED و Fan فعال شدند Green LEDخاموش شد Door باز شد و Display در حالت EMERGENCY قرار گرفت

۱۷ .بازیابی خودکار پس از رفع خطر

برای تست Recovery مقدار Smoke به صفر تغییر داده شد در حالی که برنامه MCU همچنان در حال اجرا بود به دلیل AND بودن شرط Emergency با صفر شدن Smoke سامانه بدون Restart شدن برنامه به حالت Normal برگشت



تصویر ۱۷ .حالت Normal پس از Recovery خودکار

```
--------------------------------
SYSTEM RECOVERED
Temperature: 98 C
Smoke Raw: 0
Alarm: OFF
Red LED: OFF
Green LED: ON
Fan: OFF
Door: CLOSED
Display: NORMAL
--------------------------------
```

تصویر ۱۶ .خروجی SYSTEM RECOVERED پس از صفر شدنSmoke

در حالت Recovery پیام SYSTEM RECOVERED ثبت شد Alarm و Red LED و Fan خاموش شدند Green LED روشن شد Door بسته شد و Display وضعیت NORMAL را نشان داد

18 .چک نهایی پروژه

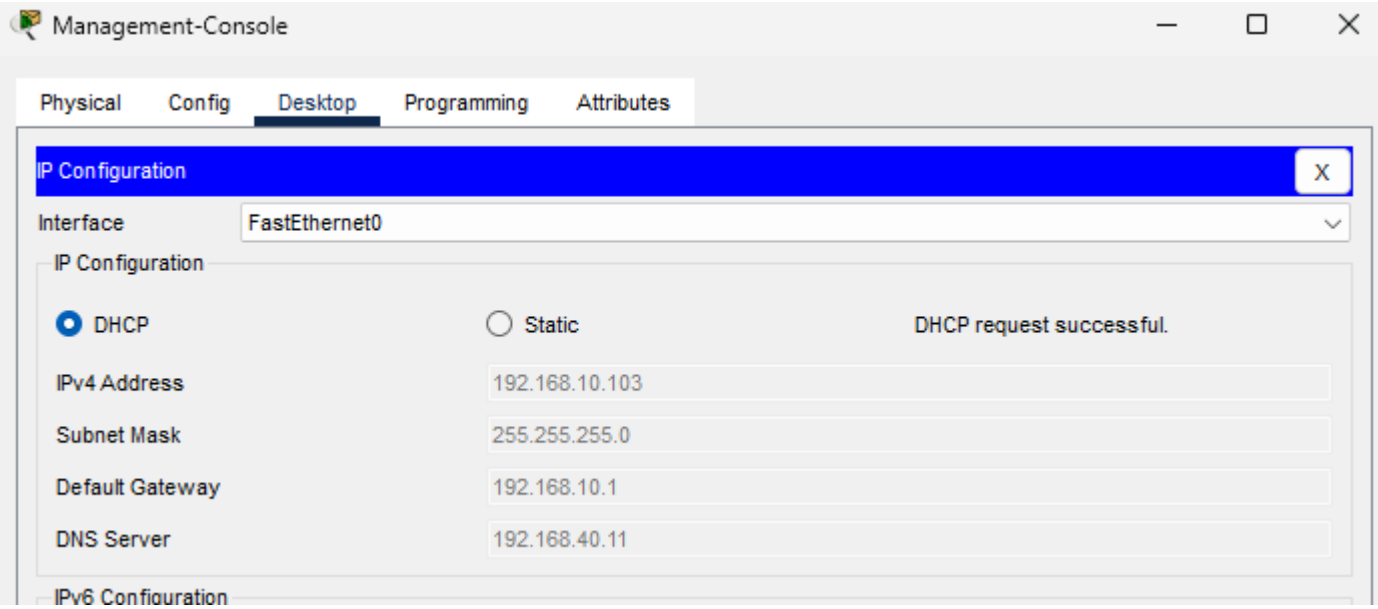
نتیجه	آیتم
انجام شد	چهار Router و چهارLAN
انجام شد	OSPFو مسیر جایگزین
انجام شد	DHCP
انجام شد	DNS
انجام شد	Web Server
انجام شد	Mail Server
انجام شد	File ServerوFTP
5از5	Management Consoleبا پنج سرویس
انچام شد	IoT Emergency
انجام شد	IoT Automatic Recovery

۲۰. مستندات تکمیلی کانفیگ و کد برنامه

این بخش به گزارش قبلی اضافه شده است تا مراحل عملی کانفیگ شبکه و وضعیت واقعی کلاینت ها و Router و OSPF و سرویس های DHCP و DNS و همچنین سورس کامل University Network Management Console به صورت مستند ارائه شود

۱.۲۰. دریافت آدرس در سه LAN اصلی

کلاینت های Academic و Admin و Library از DHCP Server مرکزی آدرس دریافت می کنند تصاویر Admin و Library دریافت صحیح IP و Gateway و DNS را نشان می دهند

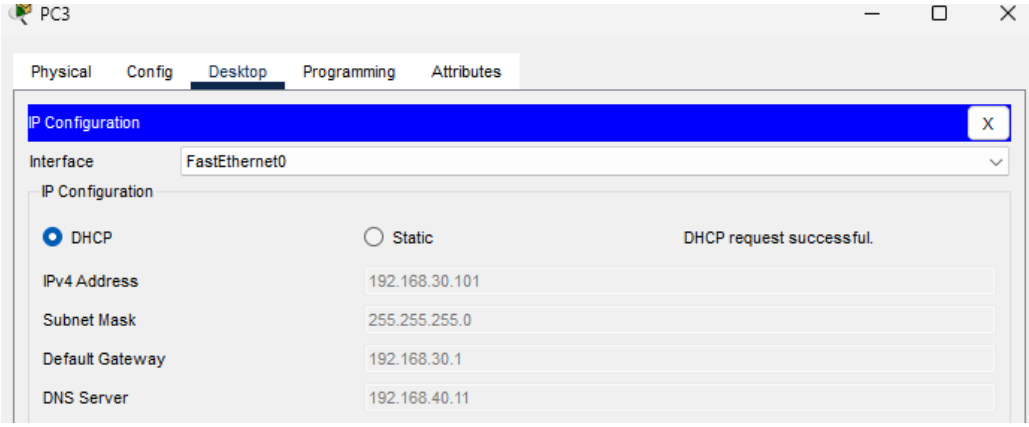


تصویر تکمیلی ۱. وضعیت DHCP در Management Console شبکه Academic

در این تصویر Management Console آدرس 192.168.10.103 و DNS Server برابر 192.168.40.11 را دریافت کرده است Default Gateway برابر 192.168.10.1 شد و تست های ارتباطی و سرویس ها با موفقیت انجام شدند

تصویر تکمیلی ۲. دریافت آدرس DHCP در Laptop1 شبکهAdmin

در شبکه Admin کلاینت آدرس 192.168.20.100 را دریافت کرده است Gateway این شبکه 192.168.20.1 و DNS Server همان DNS مرکزی با آدرس 192.168.40.11 است



تصویر تکمیلی ۳ .دریافت آدرس DHCP در PC3 شبکهLibrary

در شبکه Library کلاینت آدرس 192.168.30.101 را دریافت کرده است Gateway برابر 192.168.30.1 و DNS Server برابر 192.168.40.11است

۲.۲۰ .بررسی Interface ها و OSPF روی R1

پس از انجام کانفیگRouter ها وضعیتInterface های R1 بررسی شد سه Interface مورد استفاده شبکه در وضعیت up و پروتکل آن ها نیز up است این نتیجه نشان می دهد اتصال LAN و هر دو لینک بینRouter ها در وضعیت فعال قرار دارند

```
R1-Academic>enabl
R1-Academic#enable
R1-Academic#show ip interface brief
Interface              IP-Address      OK? Method Status                Protocol
GigabitEthernet0/0     192.168.10.1    YES manual up                    up
GigabitEthernet0/1     10.0.12.1       YES manual up                    up
GigabitEthernet0/2     10.0.41.2       YES manual up                    up
Vlan1                  unassigned      YES unset  administratively down down
R1-Academic#
```

Ctrl+F6 to exit CLI focus Copy Paste

تصویر تکمیلی ۴ .خروجی show ip interface brief روی R1

آدرس GigabitEthernet0/0 برابر 192.168.10.1 است لینک R1 به R2 روی GigabitEthernet0/1 با آدرس 10.0.12.1 قرار دارد لینک R1 به R4 روی GigabitEthernet0/2 با آدرس 10.0.41.2 قرار دارد

نمونه مراحل کانفیگ R1 در CLI به شکل زیر انجام شده است همین الگو با آدرس های مربوط به هر شبکه روی R2 و R3 و R4 اجرا شده است

```
enable
configure terminal
hostname R1-Academic
no ip domain-lookup
interface GigabitEthernet0/0
ip address 192.168.10.1 255.255.255.0
ip helper-address 192.168.40.10
no shutdown
exit
interface GigabitEthernet0/1
ip address 10.0.12.1 255.255.255.252
no shutdown
exit
interface GigabitEthernet0/2
ip address 10.0.41.2 255.255.255.252
no shutdown
exit
router ospf 1
```

```
router-id 1.1.1.1
passive-interface GigabitEthernet0/0
network 192.168.10.0 0.0.0.255 area 0
network 10.0.12.0 0.0.0.3 area 0
network 10.0.41.0 0.0.0.3 area 0
end
write memory
```

برای بررسی همسایگی OSPF دستور show ip ospf neighbor اجرا شد R2 با Router ID برابر 2.2.2.2 و R4 با Router ID برابر 4.4.4.4 هر دو در وضعیت FULL قرار دارند

```
R1-Academic#show ip ospf neighbor


Neighbor ID     Pri   State           Dead Time   Address         Interface
2.2.2.2           1   FULL/DR         00:00:30    10.0.12.2       GigabitEthernet0/1
4.4.4.4           1   FULL/DR         00:00:39    10.0.41.1       GigabitEthernet0/2
R1-Academic#
```

تصویر تکمیلی ۵ .همسایگی کامل OSPF بین R1 وRouter های مجاور

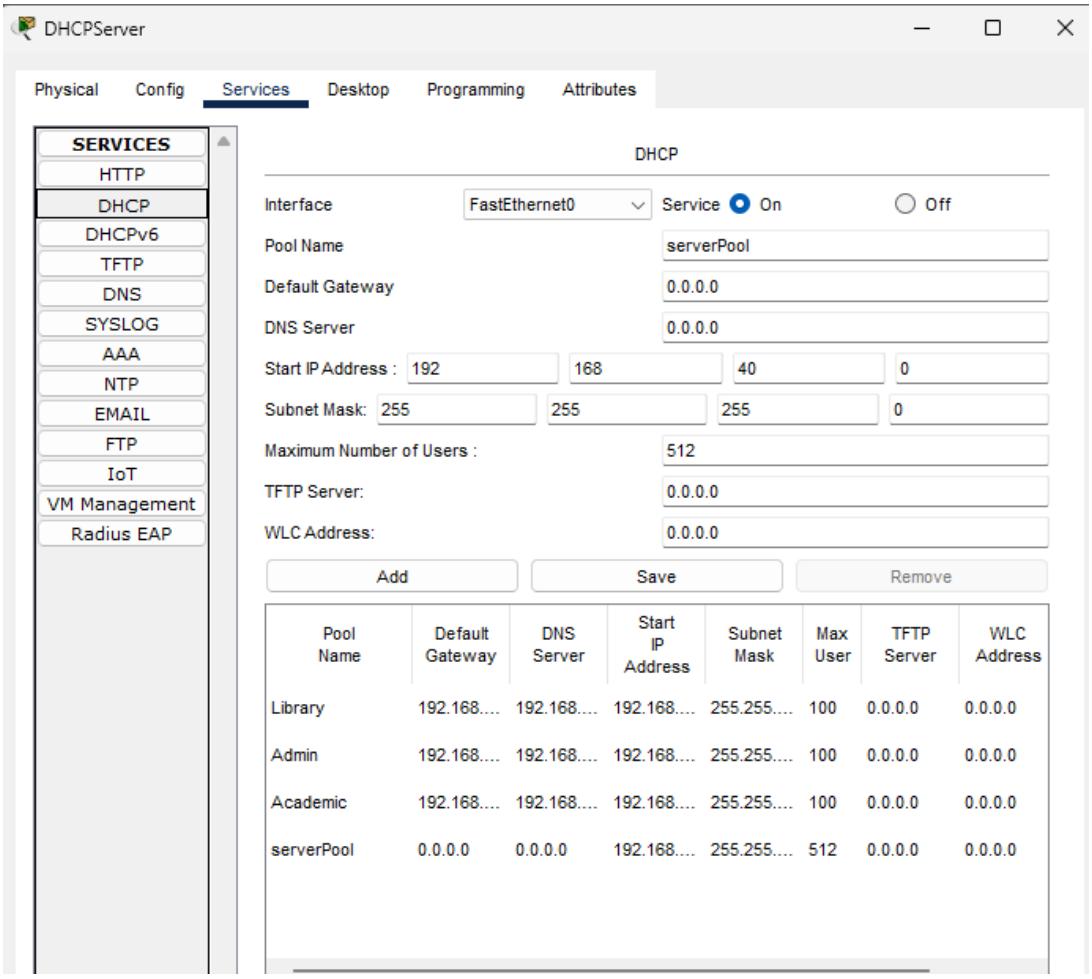
جدول Routing نیز وجودRoute های OSPF را برایLAN های دوردست نشان می دهد شبکه Admin و شبکه Library با کد O در جدول Route ثبت شده اند این وضعیت مسیریابی پویا بین بخش های دانشگاه را تایید می کند

```
R1-Academic#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/8 is variably subnetted, 6 subnets, 2 masks
C       10.0.12.0/30 is directly connected, GigabitEthernet0/1
L       10.0.12.1/32 is directly connected, GigabitEthernet0/1
O       10.0.23.0/30 [110/2] via 10.0.12.2, 01:21:54, GigabitEthernet0/1
O       10.0.34.0/30 [110/2] via 10.0.41.1, 01:21:54, GigabitEthernet0/2
C       10.0.41.0/30 is directly connected, GigabitEthernet0/2
L       10.0.41.2/32 is directly connected, GigabitEthernet0/2
     192.168.10.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.10.0/24 is directly connected, GigabitEthernet0/0
L       192.168.10.1/32 is directly connected, GigabitEthernet0/0
O    192.168.20.0/24 [110/2] via 10.0.12.2, 02:35:44, GigabitEthernet0/1
O    192.168.30.0/24 [110/3] via 10.0.41.1, 01:21:54, GigabitEthernet0/2
 --More--
```

تصویر تکمیلی ۶ .جدول مسیریابی R1 وRoute های یادگرفته شده توسطOSPF

۳.۲۰. کانفیگ DHCP Server و DNS Server

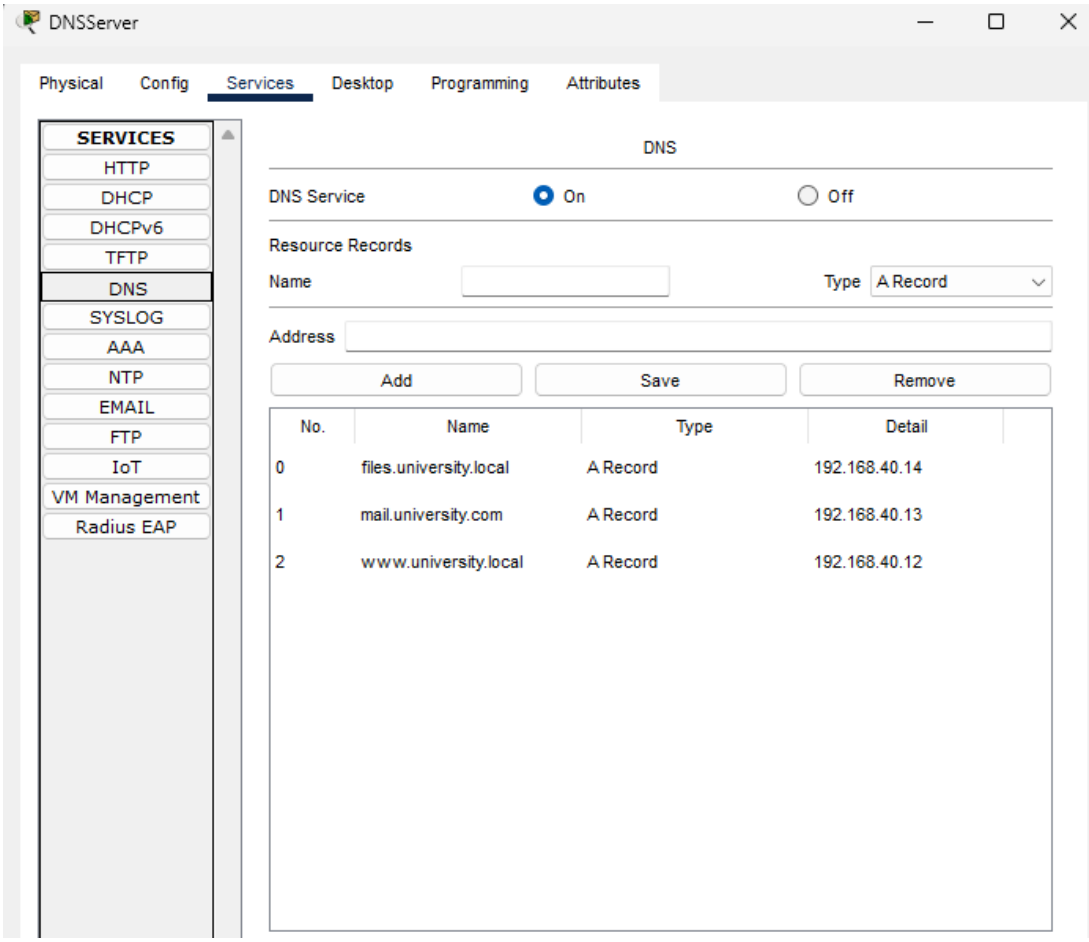
در DHCP Server سرویس DHCP فعال است و برای سه LAN اصلی Pool مستقل ایجاد شده استPool های Library و Adminو Academic در جدول سرویس قابل مشاهده هستند برای هر Pool آدرس Gateway همان Interface LAN روی Routerمربوطه است و DNS Server روی آدرس 192.168.40.11 قرار دارد



تصویر تکمیلی ۷. Pool های DHCP برای شبکه های Academic و Admin وLibrary

مراحل کانفیگ DHCP شامل فعال کردن Service و ساخت Pool برای هر LAN و تعیین Default Gateway و DNS Server و Start IPو Subnet Mask و Maximum Number of Users است فیلدهای بالای تصویر مربوط به serverPool پیش فرض Packet Tracerهستند وPool های اصلی پروژه در جدول پایین تصویر با نام های Library و Admin و Academic دیده می شوندRouter های R1 و R2 و R3 نیز با ip helper-address درخواست های DHCP را به سرور 192.168.40.10 منتقل می کنند

در DNS Server سرویس DNS فعال شده و سه A Record اصلی ثبت شده است آدرس www.university.local به Web Serverاشاره می کند آدرس mail.university.com به Mail Server اشاره می کند آدرس files.university.local نیز به File Serverاشاره می کند



تصویر تکمیلی ۸ .رکوردهای DNS مورد استفاده در پروژه

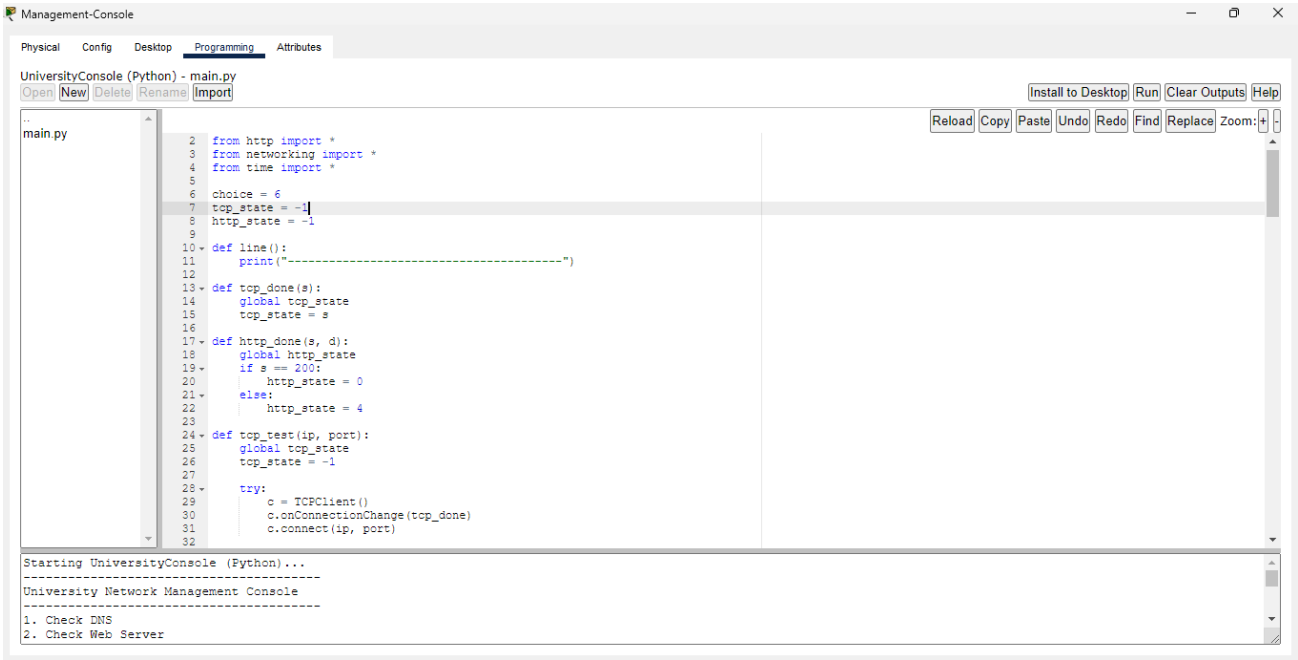
www.university.local برابر192.168.40.12

mail.university.com برابر192.168.40.13

files.university.local برابر192.168.40.14

۴.۲۰. مراحل ساخت University Network Management Console

برای ساخت Management Console یک PC در شبکه Academic قرار داده شد و با کابل Copper Straight Through به Sw-Academic متصل شد سپس از Desktop و IP Configuration حالت DHCP انتخاب شد تا PC به صورت خودکار IP و Gateway و DNS دریافت کند



تصویر تکمیلی ۹. محیط Programming و ابتدای سورس UniversityConsole در Packet Tracer

۵.۲۰. منطق برنامه Management Console

برنامه پنج سرویس اصلی را بررسی می کند تست DNS با باز کردن آدرس www.university.local انجام می شود Web Server با HTTP روی 192.168.40.12 بررسی می شود Mail Server با اتصال TCP به پورت 25 بررسی می شود File Server با اتصال TCP به پورت 21 بررسی می شود DHCP نیز بر اساس IP محلی Management Console و قرار گرفتن آن در بازه تعریف شده Academic ارزیابی می شود

برای جلوگیری از توقف برنامه در صورت خرابی یک سرویس از Timeout محدود و try و except استفاده شده است نتیجه هر تست به شکل Online - Success یا Offline - Failed نمایش داده می شود در پایان تعداد سرویس های Online محاسبه می شود و وضعیت کلی SUCCESS یا WARNING یا FAILED چاپ می شود

۶.۲۰ .سورس کاملUniversity Network Management Console

سورس زیر همان منطق نهایی اجرا شده در Management Console است

بخش 1 از سورس برنامه

```
from tcp import *
from http import *
from networking import *
from time import *

choice = 6
tcp_state = -1
http_state = -1

def line():
print("----------------------------------------")

def tcp_done(s):
global tcp_state
tcp_state = s

def http_done(s, d):
global http_state
if s == 200:
http_state = 0
else:
http_state = 4

def tcp_test(ip, port):
global tcp_state
tcp_state = -1

try:
c = TCPClient()
c.onConnectionChange(tcp_done)
```

```
c.connect(ip, port)

n = 0
while tcp_state == -1 and n < 50:
delay(100)
n += 1

c.close()
return tcp_state == 0
except:
return False

def http_test(url):
global http_state
http_state = -1

try:
h = HTTPClient()
h.onDone(http_done)
h.open(url)
```

بخش 2 از سورس برنامه

```
n = 0
while http_state == -1 and n < 50:
delay(100)
n += 1

h.stop()
return http_state == 0
except:
return False

def show(name, ok):
```

```
if ok:
print(name + ": Online - Success")
else:
print(name + ": Offline - Failed")

def check_dns():
ok = http_test("http://www.university.local")
show("DNS", ok)
return ok

def check_web():
ok = http_test("http://192.168.40.12")
show("Web Server", ok)
return ok

def check_mail():
ok = tcp_test("192.168.40.13", 25)
show("Mail Server", ok)
return ok

def check_file():
ok = tcp_test("192.168.40.14", 21)
show("File Server", ok)
return ok

def check_dhcp():
try:
ip = localIP()
p = ip.split(".")
ok = False

if len(p) == 4:
if p[0] == "192" and p[1] == "168" and p[2] == "10":
```

```
n = int(p[3])
if n >= 100 and n <= 199:
ok = True
```

بخش 3 از سورس برنامه

```
if ok:
print("DHCP: Online - Success")
print("Local IP: " + ip)
else:
print("DHCP: Offline - Failed")

return ok
except:
print("DHCP: Offline - Failed")
return False

def report():
good = 0

line()
print("University Network Service Report")
line()

if check_dns():
good += 1

if check_web():
good += 1

if check_mail():
good += 1

if check_file():
```

```
good += 1

if check_dhcp():
good += 1

line()
print("Online Services: " + str(good) + "/5")

if good == 5:
print("Overall: SUCCESS - All services online")
elif good == 0:
print("Overall: FAILED - All services offline")
else:
print("Overall: WARNING - Some services offline")

line()

def menu():
```

بخش 4 از سورس برنامه

```
line()
print("University Network Management Console")
line()
print("1 Check DNS")
print("2 Check Web Server")
print("3 Check Mail Server")
print("4 Check File Server")
print("5 Check DHCP")
print("6 Overall Service Report")
print("7 Exit")
line()
print("Selected option: " + str(choice))
print("")
```

```
if choice == 1:
check_dns()
elif choice == 2:
check_web()
elif choice == 3:
check_mail()
elif choice == 4:
check_file()
elif choice == 5:
check_dhcp()
elif choice == 6:
report()
elif choice == 7:
print("Console closed")
else:
print("Invalid option")

menu()
```

نکته درباره استفاده نکردن از input در برنامهManagement Console

در نسخه Python موجود در Cisco Packet Tracer محدودیت هایی در دریافت ورودی تعاملی از کاربر وجود دارد در آزمایش اولیه استفاده از تابع input باعث توقف برنامه و باقی ماندن آن در حالت انتظار شد و اجرای Console به درستی ادامه پیدا نکرد

به همین دلیل برای انتخاب گزینه منو از متغیر choice استفاده شد

بخش IOT

```
from gpio import *
from time import *

state = ""

def get_temperature(raw):
    return ((raw / 1024.0) * 201.0) - 100.0

def normal_mode():
    digitalWrite(0, LOW)
    digitalWrite(1, HIGH)
    digitalWrite(2, LOW)
    customWrite(3, "0")
    customWrite(4, "0")
    customWrite(5, "SYSTEM NORMAL")

def emergency_mode():
    digitalWrite(0, HIGH)
    digitalWrite(1, LOW)
    digitalWrite(2, HIGH)
    customWrite(3, "2")
```

```
    customWrite(4, "1")
    customWrite(5, "EMERGENCY - FIRE")

def main():
    global state

    pinMode(A0, IN)
    pinMode(A1, IN)

    pinMode(0, OUT)
    pinMode(1, OUT)
    pinMode(2, OUT)

    normal_mode()
    state = "normal"

    print("University Disaster Recovery System")
    print("System Started")
    print("Status: NORMAL")

    while True:
        temp_raw = analogRead(A0)
        smoke_raw = analogRead(A1)
```

```
temperature = get_temperature(temp_raw)

smoke = smoke_raw > 0


if temperature > 45 and smoke:

    if state != "emergency":

        emergency_mode()

        state = "emergency"


        print("--------------------------------")

        print("EMERGENCY DETECTED")

        print("Temperature: " + str(int(temperature)) + " C")

        print("Smoke Raw: " + str(smoke_raw))

        print("Alarm: ON")

        print("Red LED: ON")

        print("Green LED: OFF")

        print("Fan: ON")

        print("Door: OPEN")

        print("Display: EMERGENCY")

        print("--------------------------------")

else:

    if state != "normal":

        normal_mode()

        state = "normal"
```

```
            print("--------------------------------")
            print("SYSTEM RECOVERED")
            print("Temperature: " + str(int(temperature)) + " C")
            print("Smoke Raw: " + str(smoke_raw))
            print("Alarm: OFF")
            print("Red LED: OFF")
            print("Green LED: ON")
            print("Fan: OFF")
            print("Door: CLOSED")
            print("Display: NORMAL")
            print("--------------------------------")

        delay(500)

if __name__ == "__main__":
    main()
```

منطق برنامه

دو حسگر ورودی به MCU متصل شده اند `A0` برای Temperature Sensor و `A1`برای Smoke Sensor استفاده شده است مقدار خام حسگر دما توسط تابع `get_temperature` به درجه سانتی گراد تبدیل می شود

تابع `normal_mode` تجهیزات را در وضعیت عادی قرار می دهد در این وضعیت LED سبز روشن است LED قرمز و Alarm و Fanخاموش هستند Door بسته است و روی Display عبارت `SYSTEM NORMAL`نمایش داده می شود

تابع emergency_mode وضعیت بحران را فعال می کند LED قرمز و Alarm و Fan روشن می شوندLED سبز خاموش می شود Door باز می شود و روی Display عبارت EMERGENCY - FIREقرار می گیرد

۲۱ .نتیجه گیری

شبکه دانشگاهی طراحی شده تمام بخش های اصلی سناریو را پوشش می دهد OSPF امکان مسیریابی پویا و استفاده از مسیر جایگزین را فراهم کرده است سرویس های مرکزی ازLAN های مختلف قابل دسترسی هستند و Management Console وضعیت سرویس ها را به صورت یکپارچه گزارش می کند بخش Disaster Recovery نیز توانست شرایط دمای بالا و وجود دود را تشخیص دهد و واکنش اضطراری را به صورت خودکار اجرا کند پس از رفع شرط خطر سامانه بدون دخالت دستی به وضعیت Normal بازگشت

مجموع آزمون های انجام شده نشان می دهد توپولوژی و سرویس ها و سناریوی افزونگی و سامانه IoT به صورت عملی در Cisco Packet Tracerقابل اجرا هستند