Campus Network Design and Implementation

Computer Network-1 Course Project

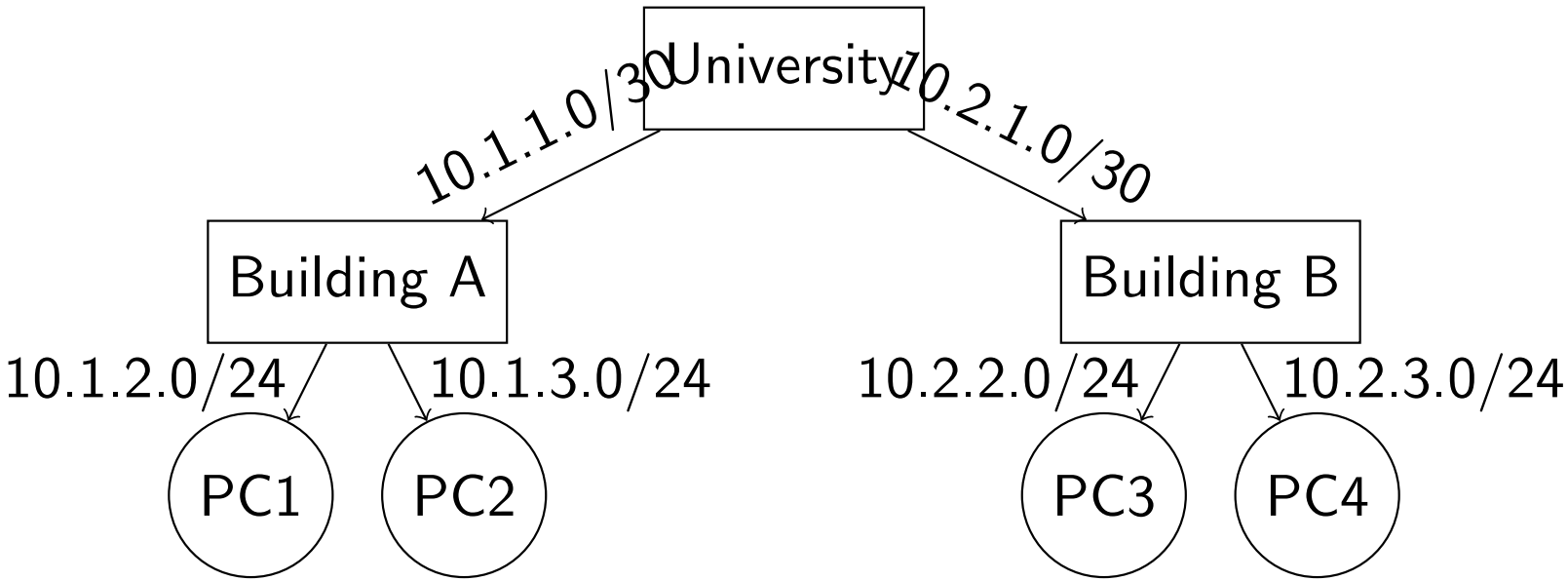
Theodoros

Contributors:

May 10, 2025

Contents

Network Diagram



Network Configuration

- MikroTik RouterOS is used for routers.
- Each router uses a unique identity and interface addressing.
- PCs have static IP addresses and default gateways.
- Subnetting: 10.1.x.x for Building A, 10.2.x.x for Building B.

Router Configuration: University Router (R1)

```
system identity set name="university"
ip address add address=10.1.1.1/30 interface=ether1
ip address add address=10.2.1.1/30 interface=ether2
```

- Connects to Building A and B routers.
- /30 subnets for point-to-point links.

Router Configuration: Building A Router (R2)

```
system identity set name="Building_A"
ip address add address=10.1.1.2/30 interface=ether1
ip address add address=10.1.2.1/24 interface=ether2
ip address add address=10.1.3.1/24 interface=ether3
```

- ether1: Uplink to University Router
- ether2: First LAN subnet for PC1
- ether3: Second LAN subnet for PC2

Router Configuration: Building B Router (R3)

```
system identity set name="Building_B"
ip address add address=10.2.1.2/30 interface=ether1
ip address add address=10.2.2.1/24 interface=ether2
ip address add address=10.2.3.1/24 interface=ether3
```

- ether1: Uplink to University Router
- ether2: First LAN subnet for PC3
- ether3: Second LAN subnet for PC4

PC Configuration: Building A

```
# PC1
PC1> ip 10.1.2.2/24 10.1.2.1
# PC2
PC2> ip 10.1.3.2/24 10.1.3.1
```

- Each PC has a static IP and gateway.
- Test: `ping 10.1.2.1` or `ping 10.1.3.1`

PC Configuration: Building B

```
# PC3
PC3> ip 10.2.2.2/24 10.2.2.1
# PC4
PC4> ip 10.2.3.2/24 10.2.3.1
```

- Each PC has a static IP and gateway.
- Test: `ping 10.2.2.1` or `ping 10.2.3.1`

RIP Configuration: University Router

```
1  # Enable RIP on relevant interfaces
2  routing rip interface add interface=ether1 send=v2 receive
       =v2
3  routing rip interface add interface=ether2 send=v2 receive
       =v2
4  # Tell RIP about directly connected subnets
5  routing rip network add network=10.1.1.0/30
6  routing rip network add network=10.2.1.0/30
7  # Set RIP settings
8  routing rip set redistribute-connected=yes
9  routing rip set update-timer=15s
10 routing rip set timeout-timer=30s
11 routing rip set garbage-timer=30s
```

RIP Configuration: Building A Router

```
1  routing rip interface add interface=ether1 send=v2 receive
       =v2
2  routing rip interface add interface=ether2 send=v2 receive
       =v2
3  routing rip interface add interface=ether3 send=v2 receive
       =v2
4  routing rip network add network=10.1.1.0/30
5  routing rip network add network=10.1.2.0/24
6  routing rip network add network=10.1.3.0/24
7  routing rip set redistribute-connected=yes
8  routing rip set update-timer=15s
9  routing rip set timeout-timer=30s
10 routing rip set garbage-timer=30s
```

RIP Configuration: Building B Router

```
1  routing rip interface add interface=ether1 send=v2 receive
       =v2
2  routing rip interface add interface=ether2 send=v2 receive
       =v2
3  routing rip interface add interface=ether3 send=v2 receive
       =v2
4  routing rip network add network=10.2.1.0/30
5  routing rip network add network=10.2.2.0/24
6  routing rip network add network=10.2.3.0/24
7  routing rip set redistribute-connected=yes
8  routing rip set update-timer=15s
9  routing rip set timeout-timer=30s
10 routing rip set garbage-timer=30s
```

RIP Command Explanations

- `routing rip interface add interface=etherX send=v2 receive=v2`: Enables RIP version 2 on the router's interface. Only interfaces used to connect other routers should run RIP.
- `routing rip network add network=10.X.X.0/YY`: Informs RIP which directly-connected networks to advertise to neighbors.
- `routing rip set redistribute-connected=yes`: Ensures directly-connected networks are included in RIP updates.
- `routing rip set update-timer=15s`: How often RIP sends updates (default = 30s; lower for lab/small network).
- `routing rip set timeout-timer=30s`: If no update is received in 30s, the route is considered invalid.
- `routing rip set garbage-timer=30s`: A route is removed 30s after being marked invalid.

Verification: End-to-End Connectivity with RIP

Ping from PC3 to PC2 (Building B to Building A):

```
PC3> ping 10.1.3.2

84 bytes from 10.1.3.2 icmp_seq=1 ttl=61 time=1.491 ms
84 bytes from 10.1.3.2 icmp_seq=2 ttl=61 time=1.291 ms
84 bytes from 10.1.3.2 icmp_seq=3 ttl=61 time=2.439 ms
```

Ping from PC1 to PC4 (Building A to Building B):

```
PC1> ping 10.2.3.2

84 bytes from 10.2.3.2 icmp_seq=1 ttl=61 time=1.831 ms
84 bytes from 10.2.3.2 icmp_seq=2 ttl=61 time=1.235 ms
84 bytes from 10.2.3.2 icmp_seq=3 ttl=61 time=2.860 ms
```

Successful replies indicate full network reachability via dynamic routing.

DHCP: University Router

```
ip pool add name=pool1 ranges=10.1.1.2-10.1.1.2
ip pool add name=pool2 ranges=10.2.1.2-10.2.1.2

ip dhcp-server add interface=ether1 address-pool=pool1 \
  lease-time=24h name=dhcp1 disabled=no
ip dhcp-server add interface=ether2 address-pool=pool2 \
  lease-time=24h name=dhcp2 disabled=no

ip dhcp-server network add address=10.1.1.0/30 \
  dns-server=8.8.8.8,8.8.4.4 gateway=10.1.1.1
ip dhcp-server network add address=10.2.1.0/30 \
  dns-server=8.8.8.8,8.8.4.4 gateway=10.2.1.1

ip dhcp-server enable 0
ip dhcp-server enable 1
```

DHCP University Router: Command Explanations (1/2)

- `ip pool add ...`
  Creates a pool (range) of IP addresses that the DHCP server can assign to clients, excluding addresses statically set on routers.
- `ip dhcp-server add ...`
  Enables a DHCP server on a specific interface, using the defined pool. `lease-time=24h` means each client keeps its IP for 24 hours.
- `name=dhcpX`
  Used to identify different DHCP servers per interface (e.g., `dhcp1` for ether1, `dhcp2` for ether2).

DHCP University Router: Command Explanations (2/2)

- `ip dhcp-server network add ...`
  Specifies the subnet, default gateway, and DNS servers (here, Google DNS) to inform clients.
- `ip dhcp-server enable X`
  Activates the DHCP server instance (replace `X` with the correct number).
- `ip dhcp-server print`
  Shows the current DHCP servers configured on the router.
- `ip dhcp-server lease print`
  Displays the list of IPs assigned to clients.

DHCP: Building A Router

```
1  ip address add address=10.1.2.1/24 interface=ether2
2  ip address add address=10.1.3.1/24 interface=ether3
3
4  ip pool add name=pool-buildingA1 ranges
       =10.1.2.100-10.1.2.200
5  ip pool add name=dhcp-pool-buildingA2 ranges
       =10.1.3.100-10.1.3.200
6
7  ip dhcp-server network add address=10.1.2.0/24 \
8    gateway=10.1.2.1 dns-server=8.8.8.8
9  ip dhcp-server network add address=10.1.3.0/24 \
10   gateway=10.1.3.1 dns-server=8.8.8.8
11
12 ip dhcp-server add interface=ether2 name=dhcp-buildingA1 \
13   address-pool=pool-buildingA1
14 ip dhcp-server add interface=ether3 name=dhcp-buildingA2 \
15   address-pool=dhcp-pool-buildingA2
16 ip dhcp-server enable dhcp-buildingA1
17 ip dhcp-server enable dhcp-buildingA2
```

DHCP: Building B Router

```
1  ip address add address=10.2.2.1/24 interface=ether2
2  ip address add address=10.2.3.1/24 interface=ether3
3
4  ip pool add name=pool-buildingB1 ranges
       =10.2.2.100-10.2.2.200
5  ip pool add name=dhcp-pool-buildingB2 ranges
       =10.2.3.100-10.2.3.200
6
7  ip dhcp-server network add address=10.2.2.0/24 \
8    gateway=10.2.2.1 dns-server=8.8.8.8
9  ip dhcp-server network add address=10.2.3.0/24 \
10   gateway=10.2.3.1 dns-server=8.8.8.8
11
12 ip dhcp-server add interface=ether2 name=dhcp-buildingB1 \
13   address-pool=pool-buildingB1
14 ip dhcp-server add interface=ether3 name=dhcp-buildingB2 \
15   address-pool=dhcp-pool-buildingB2
16 ip dhcp-server enable dhcp-buildingB1
17 ip dhcp-server enable dhcp-buildingB2
```

DHCP Test: PC Example

```
# On PC in Building A, subnet 10.1.3.0/24:
PC2> ip dhcp
# Expected: IP 10.1.3.x/24, GW 10.1.3.1

# On PC in Building B, subnet 10.2.2.0/24:
PC3> ip dhcp
# Expected: IP 10.2.2.x/24, GW 10.2.2.1
```

VLAN Configuration: Building A

```
1  interface vlan add name=vlan10 vlan-id=10 interface=ether2
       disabled=no
2  interface vlan add name=vlan20 vlan-id=20 interface=ether3
       disabled=no
3
4  ip address add address=10.1.2.1/24 interface=vlan10
5  ip address add address=10.1.3.1/24 interface=vlan20
6
7  ip address remove [find where interface=ether2]
8  ip address remove [find where interface=ether3]
9
10 interface vlan print
11 ip address print
```

- vlan10 on ether2: Subnet 10.1.2.0/24 (e.g., PC1, webterm, PC7)
- vlan20 on ether3: Subnet 10.1.3.0/24 (e.g., PC6)
- Only VLAN interfaces have subnet IPs; physical interfaces do not.

Switch1 Configuration: Building A VLANs

```
1  Example Switch1 configuration (for VLAN 10/20)
2   Port 0: dot1q trunk (to Router ether2/ether3)
3   Port 1: VLAN 10 access (to PC1)
4   Port 2: VLAN 10 access (to webterm)
5   Port 3: VLAN 10 access (to PC7)
6
7  Port | VLAN | Type
8  0 | 1 | dot1q # trunk to router ether2/ether3
9  1 | 10 | access # PC1
10 2 | 10 | access # webterm
11 3 | 10 | access # PC7
```

VLAN Configuration: Building B

```
1  interface vlan add name=vlan30 vlan-id=30 interface=ether2
       disabled=no
2  interface vlan add name=vlan40 vlan-id=40 interface=ether3
       disabled=no
3
4  ip address add address=10.2.2.1/24 interface=vlan30
5  ip address add address=10.2.3.1/24 interface=vlan40
6
7  ip address remove [find where interface=ether2]
8  ip address remove [find where interface=ether3]
9
10 interface vlan print
11 ip address print
```

- vlan30 on ether2: Subnet 10.2.2.0/24 (e.g., PC3)
- vlan40 on ether3: Subnet 10.2.3.0/24 (e.g., PC4, PC5)
- Only VLAN interfaces have subnet IPs; physical interfaces do not.

Switch2, Switch3, Switch4: Building B VLANs

```
# Switch2 configuration (Image 1)
Port | VLAN | Type
0 | 1 | dot1q
1 | 20 | access
2 | 20 | access
3-5 | 1 | access

# Switch3 configuration (Image 2)
Port | VLAN | Type
0 | 1 | dot1q
1 | 30 | access
2-5 | 1 | access

# Switch4 configuration (Image 3)
Port | VLAN | Type
0 | 1 | dot1q
1-2 | 40 | access
3-5 | 1 | access
```

VLAN Configuration Testing & Verification

- `interface vlan print`
  Confirms VLAN interfaces and their status.
- `ip address print`
  Shows all IP assignments, ensuring only VLANs have subnet addresses.
- Connect PCs to correct VLAN and confirm DHCP works.
- `ping <gateway>` from a PC to verify connectivity.

Network Testing: DHCP Assignment

```
PC7> ip dhcp
DORA IP 10.1.2.99/24 GW 10.1.2.1
PC6> ip dhcp
DORA IP 10.1.3.200/24 GW 10.1.3.1
PC3> ip dhcp
DORA IP 10.2.2.200/24 GW 10.2.2.1
PC5> ip dhcp
DORA IP 10.2.3.200/24 GW 10.2.3.1
PC4> ip dhcp
DORA IP 10.2.3.199/24 GW 10.2.3.1
```

- Each PC receives the correct IP address and gateway via DHCP.

Network Testing: Inter-VLAN/Inter-Building Connectivity

```
PC7> ping 10.1.2.1
84 bytes from 10.1.2.1 icmp_seq=1 ttl=64 time=0.327 ms

PC6> ping 10.1.2.99
84 bytes from 10.1.2.99 icmp_seq=1 ttl=63 time=1.741 ms

PC3> ping 10.1.2.99
84 bytes from 10.1.2.99 icmp_seq=1 ttl=61 time=1.478 ms

PC5> ping 10.1.3.200
84 bytes from 10.1.3.200 icmp_seq=1 ttl=61 time=2.051 ms

PC4> ping 10.1.2.99
84 bytes from 10.1.2.99 icmp_seq=1 ttl=61 time=2.036 ms
PC4> ping 10.1.2.3
84 bytes from 10.1.2.3 icmp_seq=1 ttl=61 time=1.390 ms
```

- All ping tests successful, showing inter-building and inter-VLAN routing with DHCP and VLANs.