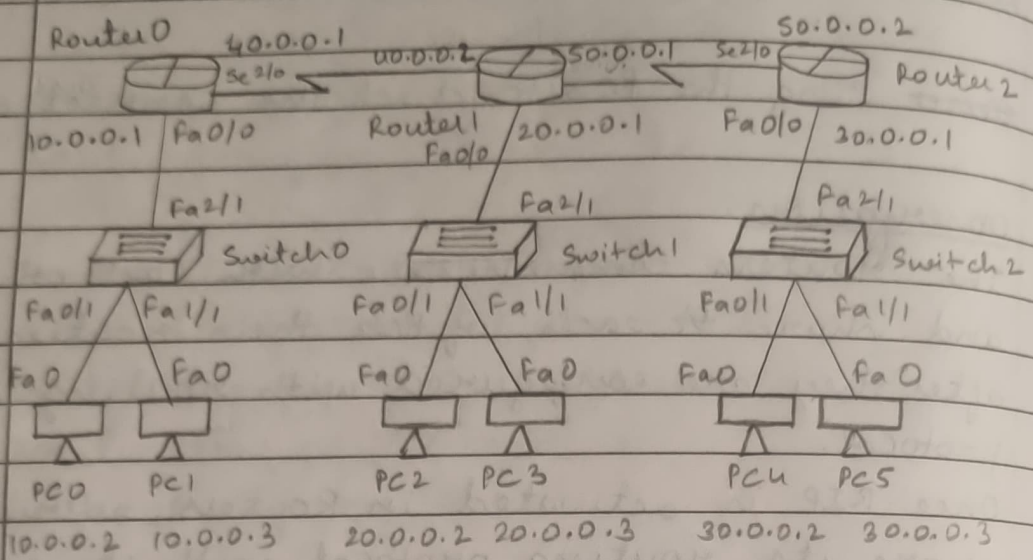
Date 20/11/24
Page

Experiment - 6

Q. Demonstrate the TTL or life of a packet

Aim: To determine the TTL or life of a packet



Topology:

1. Connect Router0 to Router1 & Router1 to Router2 using a serial-dct cable.

2. Connect Router0 to switch0 using copper straight cable. 111^{4} repeat for router 1 to switch1 & router2 to switch2.

3. Connect 6 PC's (2 each) to each of the switch using copper-straight cable & assign ip addresses

~~Observat~~ Procedure:

1. Open cisco packet tracer & establish the topology as shown above.

Date __/__/____
Page _____

2. Select the simple PDU icon & select source & destination

3. Include the PC0 and PC5 as part of the communication that is to take place, then switch to simulation mode.

4. Start the simulation by clicking on auto capture / play button & observe the TTL of a packet.

Observation:

The TTL field in a packet decrements by 1 at each router hop to prevent infinite loops. If the TTL reaches 0, the router discards the packet and sends an ICMP (Internet Control Message Protocol) "Time Exceeded" message back to the sender.

OSPF (Open Shortest Path First)

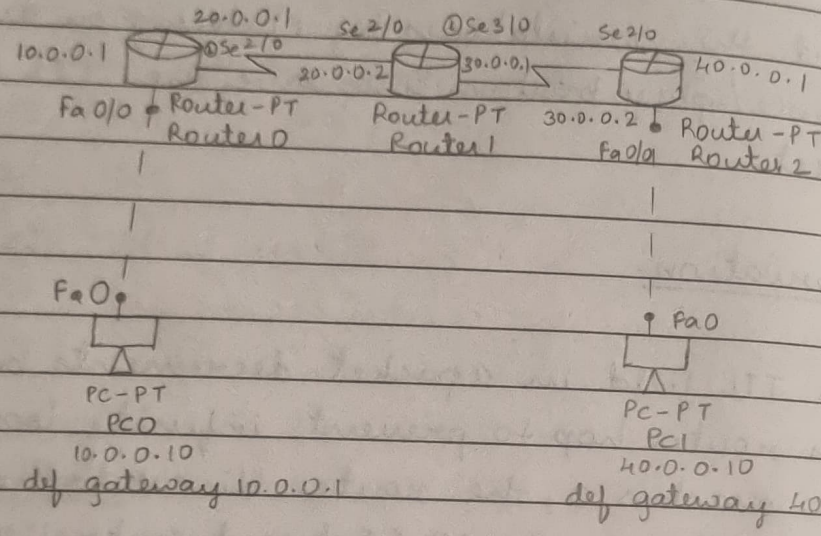
show ip route

Date 27/11/24
Page

Experiment-7

Q How to configure OSPF Routing Protocol & connect areas

Aim: To configure OSPF routing protocol & connect areas



Topology:

1. Router 1 connected to Router 0 & Router 2 through interfaces Se 2/0 & Se 3/0
2. PC 0 connected to Router 0 via copper cross cable through Fa 0/0 interface with ip address 10.0.0.10
3. PC 1 connected to Router 2 via cross cable through Fa 0/0 interface, with ip address 40.0.0.10

Procedure:

Open cisco packet tracer & drag the following:
Router: Place 3 routers in the middle.
PC: Place 2 PCs, each connected to Router 0

Date __/__/__
Page _____

& Router1 via Fa 0/0 interface

Configure all 3 routers:

- Router 0:

Router> enable
Router# config terminal
Router(config)> interface fastethernet 0/0
Router(config-if)> ip address 10.0.0.1 255.0.0.0
Router(config-if)> no shut

Router(config)> interface serial 2/0
Router(config-if)> ip address 20.0.0.1 255.0.0.0
Router(config-if)> encapsulation ppp
Router(config-if)> clock rate 64000
Router(config-if)> no shut

- Router 1:

Router(config)> interface serial 2/0
Router(config-if)> encapsulation ppp
Router(config-if)> ip address 20.0.0.2 255.0.0.0
Router(config)> interface serial 3/0
Router(config-if)> ip address 30.0.0.1 255.0.0.0
Router(config-if)> encapsulation ppp
Router(config-if)> clock rate 64000
Router(config-if)> no shut.

- Router 2:

Router(config)> interface serial 2/0
Router(config-if)> ip address 30.0.0.2 255.0.0.0
Router(config-if)> encapsulation ppp
Router(config-if)> clock rate 64000
Router(config-if)> no shut

Date __/__/__
Page

Router (config) > interface fastethernet 0/0
Router (config-if) > ip address 40.0.0.1 255.0.0.0
Router (config-if) > no shut
Router (config) > exit

- PC1

Set ip address = 10.0.0.10
Subnet Mask = 255.0.0.0
Gateway = 10.0.0.1 →

- PC2

Set ip address = 40.0.0.10
Subnet Mask = 255.0.0.0
Gateway = 40.0.0.1

→ Enable ip routing for configuring ospf routing protocol in all routers

Router 0:-

Router (config) # router ospf 1
Router (config)# router-id 1.1.1.1 →
Router (config)# network 10.0.0.0 0.255.255.255 area 3
Router (config)# network 20.0.0.0 0.255.255.255 area 1
Router (config)# exit

Router 1:

Router (config)# router ospf 1
Router (config)# router-id 2.2.2.2
Router (config)# network 20.0.0.0 0.255.255.255 area 1
Router (config)# network 30.0.0.0 0.255.255.255 area 0
Router (config) # exit.

Date / /
Page

Router 2:

```
Router (config) # router ospf 1
Router (config) # router-id 3.3.3
Router (config) # network 30.0.0.0 0.255.255.255 area 0
Router (config) # network 40.0.0.0 0.255.255.255 area 2
Router (config) # exit
```

→ Configure loopback address to routers

```
R0 (config) # interface loopback 0
R0 (config) # ip address 172.16.1.252 255.255.0.0
R0 (config) # no shut

R1 (config) # interface loopback 0
R1 (config) # ip address 172.16.1.253 255.255.0.0
R1 (config) # no shut

R2 (config) # interface loopback 0
R2 (config) # ip address 172.16.1.254 255.255.0.0
R2 (config) # no shut
```

→ Create virtual link b/w R0, R1

a 3
area 1

Router 0

```
R0 (config) # router ospf 1
R0 (config) # area 1 virtual-link 2.2.2.2
R0 (config) # exit
```

Router 1

area 1
area 0

```
R1 (config) # router ospf 1
R1 (config) # area 1 virtual-link 1.1.1.1
R1 (config) # exit.
```

P.T.O

Date __/__/__
Page

Observation:

The experiment demonstrates how OSPF dynamically learns & advertises routes, enabling efficient & scalable routing across multiple areas.
Routing tables on all routers must display networks from all areas with O IA indicating inter-area routes

Create virtual link b/w R1 & R2

R1:

router ospf 1
area 0 virtual-link 3.3.3.3
exit

R2:

router ospf 1
area 0 virtual-link 2.2.2.2
exit.

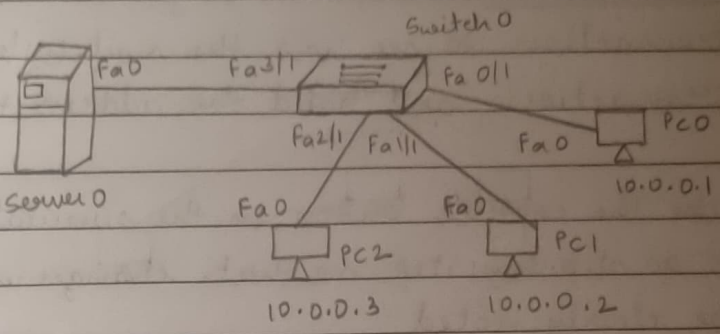
Date 19/12/24.
Page

(9)

Experiment - 8: To construct a simple LAN & understand concept & operation of ARP.

Aim: construct a simple LAN simulate operation of Address Resolution Protocol.

Topology:



1. Switch connected to ~~4~~ 3 PCs and a server via three fastethernet interfaces & one ethernet interface respectively

2. All connections made via copper straight-through cable.

Procedure:

1. Open cisco packet tracer & drag the following switch, PC: place 3 PC's, each connected to switch 0 and server: place 1 server and connect it to switch 0.

2. Assign an IP address & subnet mask to all the devices then connect them via a switch.

3. Use the inspect tool ('9'), click on a PC to view

Date __/__/__
Page

ARP table

3. Display the ARP table of all the devices.

4. Initially ARP is empty for all

5. Also in CLI of switch, the command = show mac address-table can be given on every transaction to see how the switch learns from transactions and build the address table.

6. Use the capture button in the simulation panel to go step by step so that changes in ARP can be clearly noted

7. Observe the switch as well as nodes update the ARP table as and when new communication starts.

Observation:

* As the message travels from one source host to its destination host the ARP table of all devices get updated.
ARP maps an IP address to a MAC address.
It ensures communication within a local network

ARP table for PC0 (source):

IP address	Hardware Address	Interface
10.0.0.3	0060.2F29.2CB8	FastEthernet0

Date __/__/__
Page _____

ARP table for PC 2 (destination)

IP Address	Hardware Address	Interface
10.0.0.1	00D0.D302.96DB	FastEthernet0

N
3/1/25

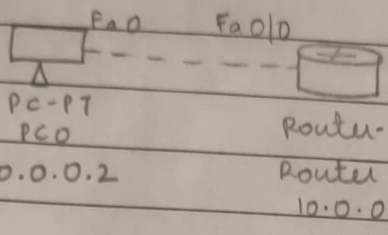
Date 19/12/24
Page

(10)

Experiment 9: To understand the operation of TELNET

Aim: To understand the operation of TELNET by accessing the router in server room from a PC in IT office.

Topology:



A router connected to a single PC via a fastethernet interface with copper cross-over cable.

Procedure:

1. Open cisco packet tracer and drag a PC & a router
2. Connect the PC to the router via fastethernet interface with a copper cross-over cable
3. Assign the IP address to the PC - 10.0.0.2 with gateway as 10.0.0.1

Configure the router:

```
Router > enable
Router # config
Router (config) # hostname r1
r1 (config) # enable secret p1
r1 (config) # ~~enable~~ interface fastethernet 0/0
r1 (config-if) # ip address 10.0.0.1 255.0.0.0
r1 (config-if) # no shut
```

Date __/__/__
Page _____

r1(config-if)# line vty 0 5
r1(config-if)# login
r1(config-line)# password p0
r1(config-line)# exit
r1(config)# exit
r1# write

In command prompt:
ping 10.0.0.1
Password for user authentication is p0
Password for enable is p1

Observations:

Telnet is a protocol for remote access to servers. It allows command-line communication over a network.
The PC is able to send the data to the router and indicates that the gateway is available and connected.

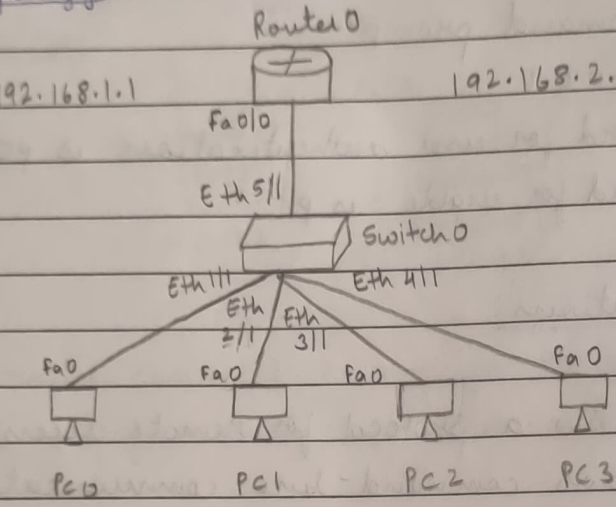
3/1/25

Date 18/12/24
Page

Experiment 10: (11) To construct VLAN and the PCs communicate along a VLAN

Aim: Construct a VLAN & enable communication between PCs among a VLAN.

Topology:



switch-PC - empty → add ports CPE
router 1841

Connect 4 PCs to the switch and a router as well to the switch. Assign the IP addresses to the PCs & set def gateway.

Procedure:

1. Choose the 1841 router & connect to a switch & 4 PC's via ethernet interface and fastethernet interface respectively

2. Set the IP addresses of the PCs & configure the router with IP address 192.168.1.1

Router> enable
Router # config terminal
Router (config) # interface Fa 0/0
Router (config-if) # ip addres 192.168.1.1
255.255.255.0

Date __/__/__
Page _____

Router (config-if) # no shut

3. In the switch, go to config tab & select VLAN Database
4. Set the VLAN number (2) & VLAN name (vi2e)
Select the * (which connects router) interface i.e., fastethernet 5/1 & make it the trunk. VLAN trunking allows switches to forward frame from different VLAN over a single link called trunk.
5. This is done by adding an additional header information called tag to the ethernet frame.
6. Look into the * (3/1 & 4/1 → change to 2) interfaces of the switches with 2 NEW VLAN systems which are 3/1 & 4/1

Config tab of router select VLAN DATABASE - enter ~~no~~ number & name of VLAN created

Router (vlan) # exit
Router # config t
Router (config) # interface fastethernet 0/0.1
Router (config-subif) # encapsulation dot1q 2
Router (config-subif) # ip address 192.168.2.1 255.255.255.0
Router (config-subif) # no shut
Router (config-subif) # exit
Router (config) # exit

Observations:

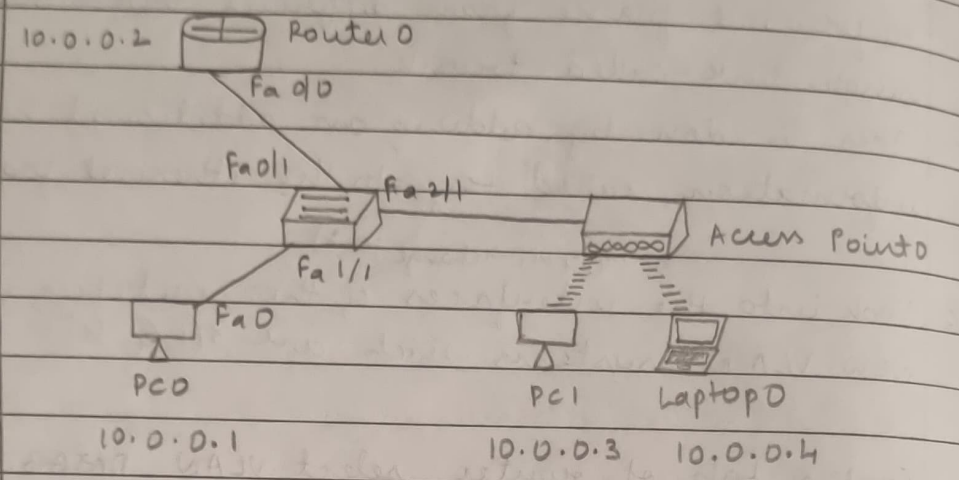
A VLAN segments a network into virtual groups It enhances security & reduces broadcast traffic. On pinging over the VLAN, the PCs are able to communicate

Date 18/12/24
Page

(12)
Experiment 11: Construct a WLAN and make the nodes communicate wirelessly

Aim: To construct WLAN and make nodes communicate wirelessly.

Topology:



Connect a router & access point to a switch through fast ethernet interface. Connect a PC and set its ip address. Take a PC & a laptop & set their ip addresses.

Procedure:

1. Drag a switch & connect it to a PC, router & an access point
2. Place a PC & laptop without any wired connection
3. Configure PC0 with ip address 10.0.0.1 & router 0.

4. Configure Access Point:
Port 1 → SSID Name → Enter any name → select

Date / /
Page

WEP & give any 10 digit hex key - 1234567890

5. Configure PC4 & laptop with wireless standards

6. Switch off the device. Drag the existing PT-HOST-NM-1AM to the component listed in the LHS. Drag WMP300N wireless interface to the empty port. Switch on the device

7. In the config tab, a new wireless interface would have been added. Now, configure SSID, WEP, WEP key, IP address & gateway to the device (select static)

8. Ping from every device to every other device & see the results

Observations:

WLAN enables wireless n/w comm. It uses radio waves for connectivity. WLAN connects devices wirelessly within a local area. It eliminates the need for physical cables.

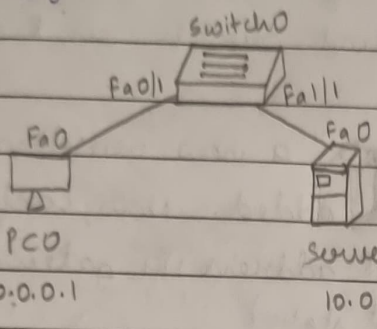
3/1/25

Date 18/12/24
Page

Experiment - 12: Configure web server, DNS within a LAN

Aim: To configure DNS server to demonstrate the mapping of IP addresses & domain names

Topology:



Connect a PC & a server to a switch, assign ip addresses as 10.0.0.1 & 10.0.0.2 resp.

Configuration:

Open cisco packet tracer & arrange as given in topology and configure the devices as given below

PC0:

IP address: 10.0.0.1

Server0:

IP address: 10.0.0.2

Connect PC0 & server0 via a switch PT

PC0 connects to switch on interface Fa 0 & switch on Fa 0/1

Server connects to switch on interface Fa0 & switch on Fa 1/1

Server 0:

Go to Server → Services → DNS

Enable on

Date __/__/__
Page _____

In the test fields add:
name: abc
address: 10.0.0.2
click add
go to HTTP
click edit for index.HTML [change if needed]
click save.

Procedure:

1. Go to PC0 → Desktop → Web browser
2. Search 'abc' in url bar (or)
3. Search 10.0.0.2 (10.0.0.2) in url bar

Output: for both 'abc' & 10.0.0.2

Cisco packet Tracer
Welcome to cisco packet tracer. Opening doors to new opportunities. Mind wide open.
Quick links:
A small page
Copyrights
Image Page
Image

Observation:

DNS translates domain names to ip addresses. It simplifies accessing websites by using human-readable names.
In this experiment, a web server & DNS were configured within a LAN to map domain names to ip addresses. The PC0 successfully accessed the server0 by both its ip address & the config-

Date __/__/__
Page

ured domain name 'abc'. The configuration was successful allowing the web page to be accessed via both methods.

N
3/1/25

Date 1/1/25
Page

Cycle 2

13 Write a program for error detecting code using CRC-CCITT (16-bits)

```
* def xor(a,b):
    result = []
    for i in range(1, len(b)):
        if a[i] == b[i]:
            result.append('0')
        else:
            result.append('1')
    return ''.join(result)
def mod2div(dividend, divisor):
    pick = len(divisor)
    tmp = dividend[0:pick]
    while pick < len(dividend):
        if tmp[0] == '1':
            tmp = xor(divisor, tmp) +
                  dividend[pick]
        else:
            tmp = xor('0'*pick, tmp) +
                  dividend[pick]
        pick += 1
    if tmp[0] == '1':
        tmp = xor(divisor, tmp)
    else:
        tmp = xor('0'*pick, tmp)
    checkword = tmp
    return checkword
def encode(data, key):
    l_key = len(key)
    appended_data = data + '0'*(l_key-1)
    remainder = mod2div(appended_data, key)
```

```
    codeword = data + remainder
    print("Remainder :", remainder)
    print("Encoded Data[Data + remainder]:",
          codeword)
    return codeword
def decode_data(encoded_data, key)
    remainder = mod2div(encoded_data,
                        key)
    print("Remainder after decoding:",
          remainder)
    if '1' not in remainder:
        print("No error detected in
              received data")
    else:
        print("Error detected in
              received data")
data = "10010010010100100"
key = "1101"
encoded_data = encode(data, key)
decoded_data = decode_data(encoded_data, key)
```

O/P:

Remainder = 11

encoded-data (data + remainder) =

10010010001001011

Remainder after decoding = 000

No error detected in received data.

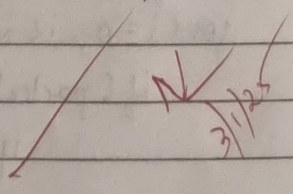
3/1/25

Date 1/1/25
Page

1. Write a program for congestion control using Leaky bucket algorithm.

```
#include<stdio.h>
#include<stdlib.h>
#include<unistd.h>
#define NOF_PACKETS 5
/* int rand(int a){
    int rn = (random()%10)%a;
    return rn == 0?1:rn;
}*/
/* #include<stdlib.h>
    long int random(void);
*/
int main(){
    int packet_sz[NOF_PACKETS],i,clk,b_size,o_rate,
    p_sz_rm=0,p_sz,p_time,op;
    for(i=0;i<NOF_PACKETS;i++)
        packet_sz[i]=random()%100;
    for(i=0;i<NOF_PACKETS;i++)
        printf("\n packet[%d]:%d bytes\t",i,
        packet_sz[i]);
    printf("\n Enter the output rate");
    scanf("%d",&o_rate);
    printf("Enter the Bucket size:");
    scanf("%d",&b_size);
    for(i=0;i<NOF_PACKETS;i++){
        if(packet_sz[i]+p_sz_rm>b_size)
            if(packet_sz[i]>b_size)
                printf("Incoming packet size
                (%d bytes) is greater
                than bucket capacity
                (%d bytes)-PACKET REJECTED
                packet_sz[i],b_size);
```

```
else
    printf("\n\n Bucket capacity exceeded-
                d-PACKETS REJECTED!!");
else {
    p-sz-rmt = packet_sz[i];
    printf("\n\n Incoming packet size: %d",
                packet_sz[i]);
    printf("\n Bytes remaining to transmit: %d",
                p-sz-rmt);
    // p-time = random() * 10;
    // printf("\n Time left for transmission: %d units",
                p-time);
    // for(clk=10; clk<=p-time; clk+=10)
    while(p-sz-rmt>0){
        sleep(1);
        if(p-sz-rmt){
            if(p-sz-rmt<=o-rate)
                op=p-sz-rmt, p-sz-rmt=0;
            else
                op=o-rate, p-sz-rmt-=o-r
                printf("\n Packet of size %d
                        transmitted", op);
                printf("--- Bytes Remaining
                        to transmit: %d", p-
    } else {
        printf("\n No packets to transmit!!");
}
}
}
}
}
```



Date __/__/__
Page _____

O/p of Leaky bucket

packet[0]: 83 bytes
packet[1]: 86 bytes
packet[2]: 77 bytes
packet[3]: 15 bytes
packet[4]: 93 bytes
Enter the o/p rate: 30
Enter the bucket size: 85

Incoming Packet Size: 83
Bytes remaining to Transmit: 83
Packet of size 30 Transmitted --- Bytes remaining to Transmit: 53
Packet of size 30 Transmitted --- " " "
" : 23
" " " 23 " --- " " "
" : 0

Incoming packet size (86 bytes) is greater than bucket capacity (85 bytes) - PACKET REJECTED

Incoming Packet size: 77
Bytes remaining to Transmit: 77
Packet of size 30 Transmitted --- Bytes remaining to Transmit: 47
" " " 30 " --- " "
" " " : 17
" " " 17 " --- " "
" " " : 0

3/11/25

Incoming Packet size: 15
Bytes remaining to Transmit: 15
Packet of size 15 Transmitted --- Bytes rema to transmit
: 0