

WEEK 14

Write a program for congestion control using Leaky bucket algorithm.

CODE:

```
#include <stdio.h>
#include <stdlib.h> // Include this for the rand() function
int main()
{
    int buckets, outlets, k = 1, num, remaining;
    printf("Enter Bucket size and outstream size\n");
    scanf("%d %d", &buckets, &outlets);
    remaining = buckets;
    while (k)
    {
        num = rand() % 1000; // Generate a random number between 0 and
499
        if (num < remaining)
        {
            remaining = remaining - num;
            printf("Packet of %d bytes accepted\n", num); // Added missing
variable
        }
        else
        {
            printf("Packet of %d bytes is discarded\n", num);
        }
        if (buckets - remaining > outlets)
        {
            remaining += outlets; // Fixed the calculation
        }
        else
            remaining = buckets;
        printf("Remaining bytes: %d \n", remaining);
        printf("If you want to stop input, press 0, otherwise, press
1\n"); scanf("%d", &k);
    }
    while (remaining < buckets) // Fixed the condition
```

```

{
    if (buckets - remaining > outlets)
    {
        remaining += outlets; // Fixed the calculation
    }
    else
        remaining = buckets;
    printf("Remaining bytes: %d \n", remaining);
}
return 0; // Added a return statement to indicate successful completion
}

```

OUTPUT:

```

PS D:\WS Codes> cd "D:\WS Codes\Ques" ; IF ($?) { gcc bucket.c -o bucket } ; IF ($?) { ./bucket }

Enter bucket size and upstream size
2000
100
Packet of 43 bytes accepted
Remaining bytes: 1957
If you want to stop input, press 0, otherwise, press 1
1
Packet of 467 bytes accepted
Remaining bytes: 1490
If you want to stop input, press 0, otherwise, press 1
1
Packet of 334 bytes accepted
Remaining bytes: 1156
If you want to stop input, press 0, otherwise, press 1
1
Packet of 500 bytes accepted
Remaining bytes: 656
If you want to stop input, press 0, otherwise, press 1
1
Packet of 169 bytes accepted
Remaining bytes: 487
If you want to stop input, press 0, otherwise, press 1
1
Packet of 724 bytes accepted
Remaining bytes: 363
If you want to stop input, press 0, otherwise, press 1
1
Packet of 400 bytes is discarded
Remaining bytes: 363
If you want to stop input, press 0, otherwise, press 1
1
Packet of 358 bytes accepted
Remaining bytes: 245
If you want to stop input, press 0, otherwise, press 1
1
Packet of 962 bytes is discarded
Remaining bytes: 245
If you want to stop input, press 0, otherwise, press 1
0
Remaining bytes: 245
Remaining bytes: 448
Remaining bytes: 548
Remaining bytes: 648
Remaining bytes: 748

```

OBSERVATION:

```

1 #include <stdio.h>
# include <string.h>
# define N 8
# define Lgen Poly()
Char data [28]
Char check - value [20];
Char gen - poly [10];
int data - length i, j;
void XOR C) {
    for (j = 1; j < N; j++)
        Check - value [j] = (Check - value [j] ^ gen - poly [j]);
    S = '0'; '1');
}

3 void receives () {
    printf ("Enter the received data : ");
    scanf ("%10s", data);
    Print ("Data received : %s", data);
    calc();
    for (i = 0; (i < N - 1) && (check - value [i] != '1'); i++)
        if (i < N - 1)
            printf ("\n error detected \n");
        else
            printf ("In error not detected \n");
}
3 void calc () {
    for (i = 0; i < N; i++)
        Check - value [i] = data [i];
    do {
        if (check - value [0] == '1')
            XOR C);
        for (i = 0; i < N - 1; i++)
}

```

```

check_value[j] = check_value[j+1]
- check_value[j] = data[i++]
} while (i <= data_length + N - 1)
}
int main()
{
printf ("\n Enter data to be transmitted \n");
scanf ("%s", &data);
printf ("\n Enter generator polynomial \n");
scanf ("%s", &gen_poly);
data_length = strlen (data);
for (i = data_length; i < data_length + N - 1; i++)
    data[i] = 0;
printf ("\n Data Padded with n-1 zeros : %s", data);
crc();
printf ("\n CRC Value is : %s", check_value);
for (i = data_length; i < data_length + N - 1; i++)
    data[i] = check_value[i - data_length];
printf ("\n Check-value to be sent : %s", data);
printf ("\n ... \n");
Receive();
return 0;
}

```

Output

Enter the data word : 10001010111001001

Calculated CRC : 111010010111001