Alexandria University

Ziyad Elbanna -26

Lab -2 -Networks

Project Packages

Client

Important points:

1. Info for the client is input from a file (IP, port,initial window size ,...)
2. Connect to Server is done as the previous project
3. Checksum is calculated also using the function calculateChecksum
4. Ack is sent to the Server using SendAck method
5. For each of the methods, I implemented recieve GBN, recieve selective repeat, and recieve stop and wait

Server

Imp. points:

1. Again info about the Server is entered from a file
2. Points 3-4 are the same as above

Both

Packet

```
struct ack_packet
{
    uint16_t cksum; /* Optional bonus part */
    uint16_t len;
    uint32_t ackno;
};
/* Data-only packets */
struct packet
{
    /* Header */
    uint16_t cksum; /* Optional bonus part */
    uint16_t len;
    uint32_t seqno;
    /* Data */
    char data[500]; /* Not always 500 bytes, can be less */
};
```

Main steps are:

1. The client sends a datagram to the server to get a file giving the its filename. This sendneeds to be backed up by a timeout in case the datagram is lost.

2. The server forks off a child process to handle the client.

3. The server (child) creates a UDP socket to handle file transfer to the client.

4. Server sends its first datagram, the server uses some random number generator random()function to decide with propability p if the datagram would be passed to the methodsend() or just ignore sending it

5. Whenever a datagram arrives, an ACK is sent out by the client to the server.

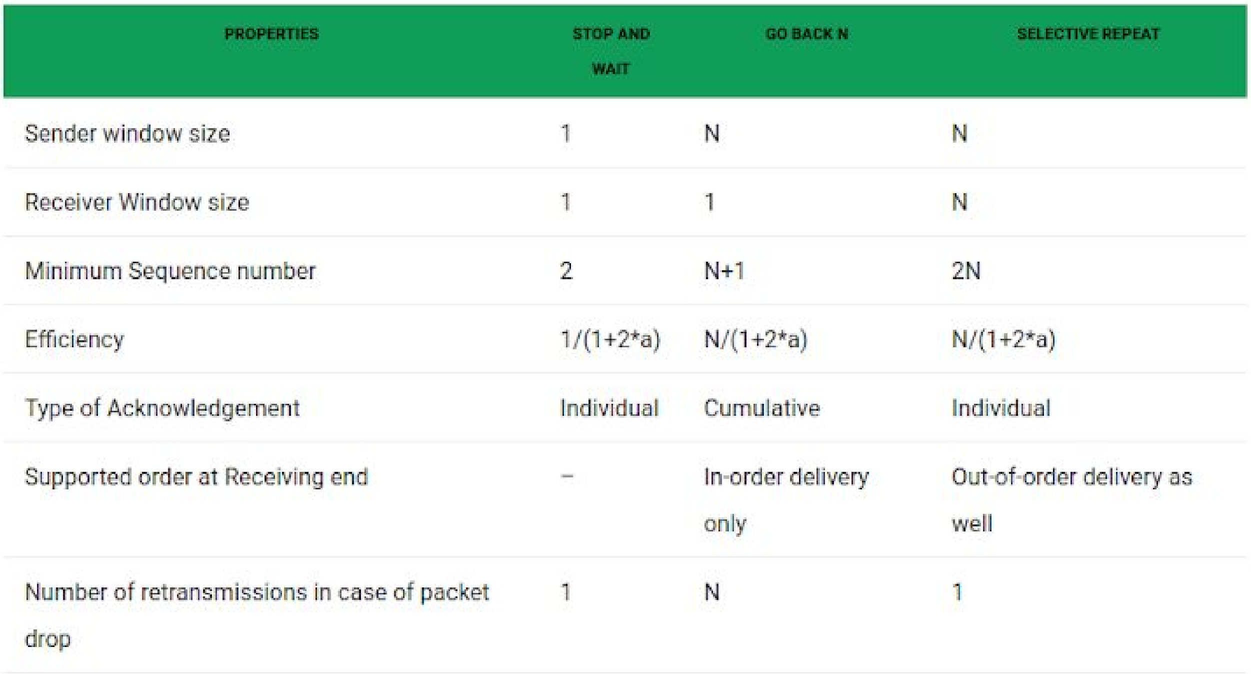
6. If you choose to discard the package and not to send it from the server the timer willexpire at the server waiting for the ACK that it will never come from the client (sincethe packet wasn’t sent to it) and the packet will be resent again from the server.

7. Update the window, and make sure to order the datagrams at the client side.

8. repeat those steps till the whole file is sent and no other datagrams remains.

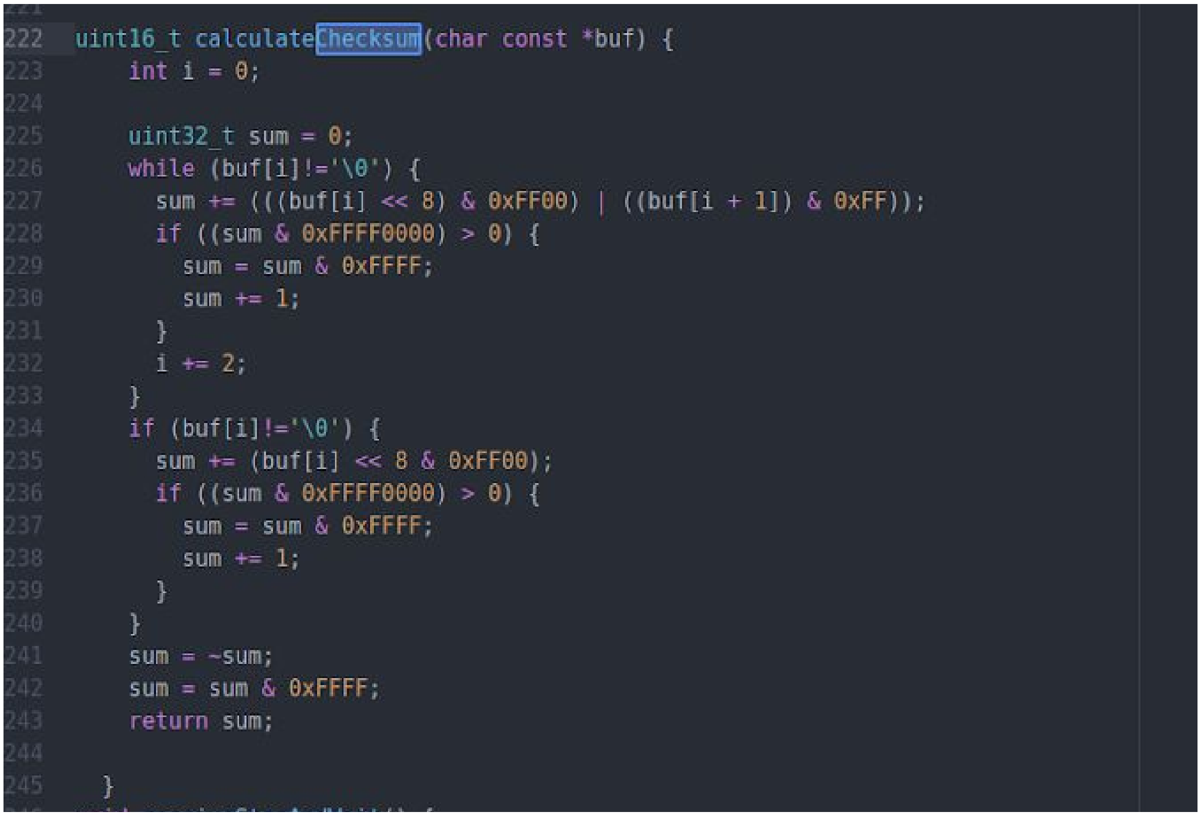
9. close the connection.

Differences:



PROPERTIES	STOP AND WAIT	GO BACK N	SELECTIVE REPEAT
Sender window size	1	N	N
Receiver Window size	1	1	N
Minimum Sequence number	2	N+1	2N
Efficiency	1/(1+2*a)	N/(1+2*a)	N/(1+2*a)
Type of Acknowledgement	Individual	Cumulative	Individual
Supported order at Receiving end	–	In-order delivery only	Out-of-order delivery as well
Number of retransmissions in case of packet drop	1	N	1

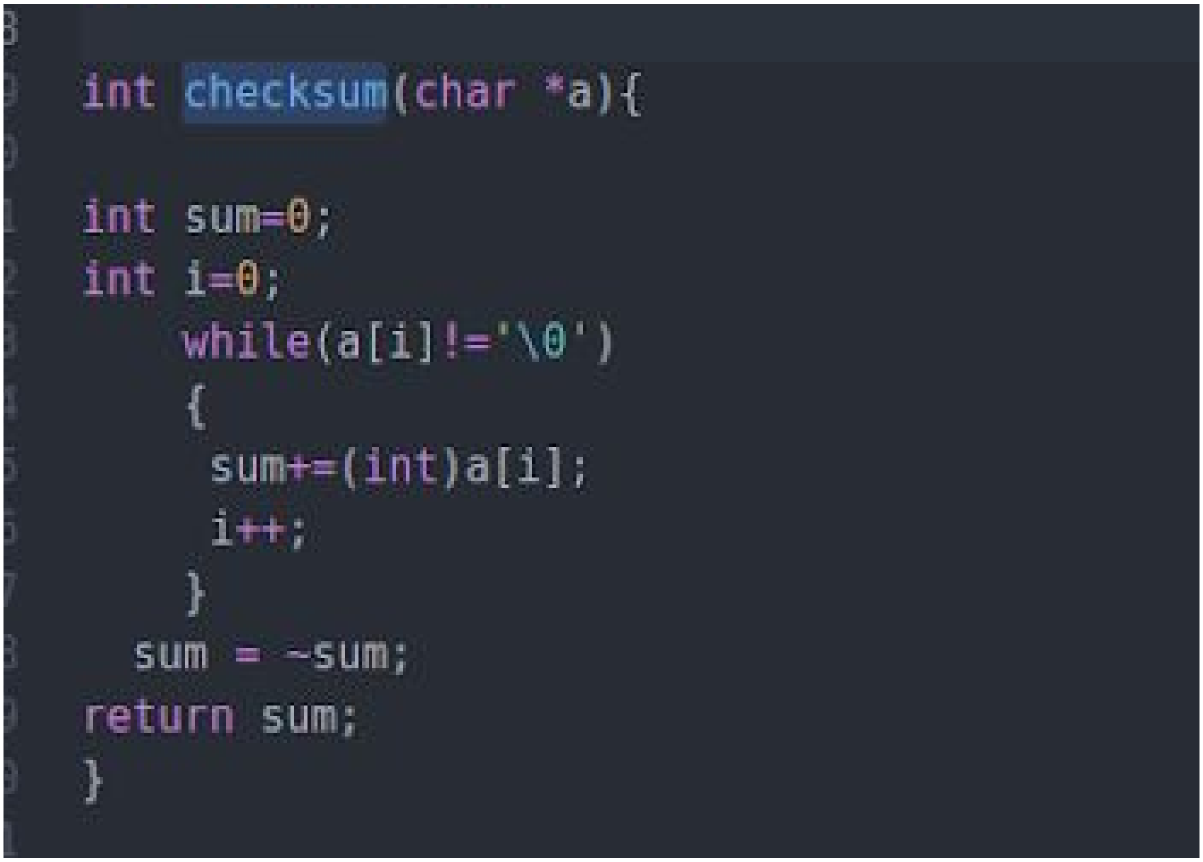
Bonus part:



```
uint16_t calculateChecksum(char const *buf) {
    int i = 0;

    uint32_t sum = 0;
    while (buf[i]!='\0') {
      sum += (((buf[i] << 8) & 0xFF00) | ((buf[i + 1]) & 0xFF));
      if ((sum & 0xFFFF0000) > 0) {
        sum = sum & 0xFFFF;
        sum += 1;
      }
      i += 2;
    }
    if (buf[i]!='\0') {
      sum += (buf[i] << 8 & 0xFF00);
      if ((sum & 0xFFFF0000) > 0) {
        sum = sum & 0xFFFF;
        sum += 1;
      }
    }
    sum = ~sum;
    sum = sum & 0xFFFF;
    return sum;

  }
```



```
int checksum(char *a){

int sum=0;
int i=0;
    while(a[i]!='\0')
    {
     sum+=(int)a[i];
     i++;
    }
  sum = ~sum;
return sum;
}
```

This is to calculate whether the result will be all zeros to see if there is an error