Programming Assignments

COEN 233 Computer Networks - Winter Quarter 2017

General Guidelines

- Programming projects are individual programs each student should have his/her own code.
- Each project requires a demo, during which the student should explain how the code works.
- Demos are part of the grade. The student only will receive full credit if demo has correct results.
- Each student should submit the source code in addition to the demo.
- The program should be turned in latest by deadline; demo is performed at the time the program has been turned in.

Program assignment 1:

Client using customized protocol on the top of UDP protocol for sending information to the server.
One client connects to one server.

Design a protocol with the following primitives:

Start of Packet identifier ….. 0XFFFF
End of Packet identifier ….. 0XFFFF
Client Id……….Maximum 0XFF (255 Decimal)
Length ……….. Maximum 0XFF (255 Decimal)

Packet Types:

DATA…. ………..……..0XFFF1
ACK (Acknowledge)……….0XFFF2
REJECT…………………….0XFFF3

Reject sub codes:

REJECT out of sequence……………0XFFF4
REJECT length mismatch…………...0XFFF5
REJECT End of packet missing……..0XFFF6
REJECT Duplicate packet…………...0XFFF7

Data Packet Format:

Bytes:	2	1	2	1	1	255	2
	Start of Packet id	Client ID	DATA	Segment No	Length	Payload	End of Packet id

ACK (Acknowledge) Packet Format:

Bytes: 2	1	2	1	2
Start of Packet id	Client ID	ACK	Received Segment No	End of Packet id

REJECT Packet Format:

Bytes: 2	1	2	2	1	2
Start of Packet id	Client ID	REJECT	Reject sub code	Received Segment No	End of Packet id

Procedure:

Client sends five packets (Packet 1, 2, 3, 4, 5) to the server, the server acknowledges with ACK each packet of client.
The client will start an **ack_timer** at the time the packet is sent to server, if the ACK (Acknowledge) for each packet has not been received during **ack_timer** period by client before expiration of timer then client should retransmit the packet that was sent before.
The timer can be set at 3 seconds (recommended) and a retry counter should be used for resending the packet if the ACK for the packet has not been arrived this counter will be started every time the packet is retransmitted, and the counter should be set to 3 times.
After 3 times the client has resent the packet and no ACK has been received from server the client should generate the following message and display on screen:
"Server does not respond".

Error handling:

NOTE: All four error handling messages below should be simulated and displayed on the screen the error response messages should be included in a file and turned in with the source code of program.

ACK error handling:
If the ACK timer expires and the ACK from server has not been received by client then an error message should be displayed on the screen by client prior to resending the packet.

Reject error handling with sub code:

Case-1: An error message should be displayed on the screen when the received packet at server is not in sequence with expected packet from client; an error message should be generated by server and sent to client.

Case-2: The server receives a packet which its length field does not match the length of data in the payload field then an error message should be generated by server and sent to the client.

Case-3: The server receives a packet which does not have the End of Packet Identifier an error message should be generated by server and sent to the client

Case-4: The server receives a duplicated packet (sequence number) an error message should be generated by server and sent to the client.

Program assignment 2:

Client using customized protocol on the top of UDP protocol for requesting identification from server for access permission to network.

One client connects to one server.
The client requests specific information from server; the server will verify the validity of request and then will respond accordingly.
The communication between client and server will use the ACK timer which was described in the first program assignment.

For the program assignment 2 you can imagine client's software module is acting in behalf of a cell phone.
The client's software module sends the request for identification of their devices in a packet to server; the packet has the following information:

Access permission request packet format:

Bytes: 2	1	2	1	1	1	4	2
Start of Packet id	Client Id	Acc_Per	Segment No	Length	Technology	Source Subscriber No	End of Packet id

Payload

Start of Packet identifier ….. 0XFFFF
End of Packet identifier. ….. 0XFFFF
Client Id…………………………...Maximum 0XFF (255 Decimal)
Acc_Per (Access Permission)….……0XFFF8
Length ……………………………….Maximum 0XFF (255 Decimal)
Source Subscriber No………………..Maximum 0XFFFFFFFF (4294967295 Decimal)

Technology:

2 G…………...02
3 G…………...03
4 G…………...04
5 G…………...05

Server has the file V**erification_Database** saved on the server which contains the Subscriber's Number, Technology, and payment's status (paid = 1, not paid = 0).

Verification_Database Format:

Subscriber Number	**Technology**	**Paid**	
408-554-6805	04	1	(1 paid)
408-666-8821	03	0	(0 not paid)
408-680-8821	02	1	(1 paid)

After verification of the content of **Identification request packet** with the content of the V**erification_Database** file following messages can be generated by server:

Subscriber has not paid message:

Bytes:	2	1	2	1	1	1	4	2
	Start of Packet id	Client Id	Not paid	Segment No	Length	Technology	Source Subscriber No	End of Packet id

Payload

Start of Packet identifier ….. 0XFFFF
End of Packet identifier. ….. 0XFFFF
Client Id…………………….Maximum 0XFF (255 Decimal)
Not paid…. ………….….….0XFFF9
Length ………………………Maximum 0XFF (255 Decimal)

Technologies:

2 G…………...02
3 G…………...03
4 G…………...04
5 G…………...05

Source Subscriber No. Maximum 0XFFFFFFFF (4294967295 Decimal)

Subscriber does not exist on database message:

Bytes:	2	1	2	1	1	1	4	2
	Start of Packet id	Client Id	Not exist	Segment No	Length	Technology	Source Subscriber No	End of Packet id

Payload

Start of Packet identifier ….. 0XFFFF
End of Packet identifier …... 0XFFFF
Client Id…………………….Maximum 0XFF (255 Decimal)
Not exist…. ………….….….0XFFFA

Length …………………….. Maximum 0XFF (255 Decimal)

Technologies:

2 G…………...02
3 G…………...03
4 G…………...04
5 G…………...05

Source Subscriber No. Maximum 0XFFFFFFFF (4294967295 Decimal)

Subscriber permitted to access the network message:

Bytes:								
	2	1	2	1	1	1	4	2
	Start of Packet id	Client Id	Acc_OK	Segment No	Length	Technology	Source Subscriber No	End of Packet id

Payload

Start of Packet identifier ….. 0XFFFF
End of Packet identifier ….. 0XFFFF
Client Id…………………….Maximum 0XFF (255 Decimal)
Access_OK ……………..….0XFFFB
Length …………………….. Maximum 0XFF (255 Decimal)

Technologies:

2 G…………...02
3 G…………...03
4 G…………...04
5 G…………...05

Source Subscriber No. Maximum 0XFFFFFFFF (4294967295 Decimal)