Damian Mirizzi

Question 1:

The protocol I will be using is TCPesque. When a user logs in to the terminal the act of opening up the ATM terminal a Session will be started and the user will be prompted to login via his debit card and pin all in 256 ascii bytes. The credit card and pin will be sent in the first packet the last byte in the stream will be an end of message ascii symbol and the server will ack by sending back banking info to the ATM the atm will then Ack by sending ack statements for all the banking packets. The server will keep sending the banking info packets until the all ack statements are received. The atm once it has all of its stream data(its been looking for) will display the banking info and allow for the user to withdraw money. The packets requesting the withdraw will be sent to the server, it will validate that it received the packets and return Y or N, if the atm receives a Y it will ack the server an despence the cash to the person. The EOM message will allow for regular expression parsing

|  |  |  |
| --- | --- | --- |
| Sequence # starts at a random # | Message using RSA encryption ascii with EOM symbol at end.  Node on message: Hello+EOM  Log in: Card # + Pin+EOM  User Data Request: Account + EOM  Withdraw request: Amount of money requested +eom  Deposit Request Cash/Check: input Cash value or check routing and value+eom | Source Port |
| Server Reply:  Node on: User please log in+EOM  Log in reply: y or n  User Data: Account balance, account name+ EOM  withdraw: y or n to deposit+ error if it occurs+eom  deposit: add Cash to account, or look for check routing and transfer funds+ return Account | Destination Port | CheckSum |

Question 2:

The Stateless protocol will be UDPesque. It will be carried along the network and the end node will receive the the data via message the messages. This form of information transfer is only scary due to the lack of checksums in the messages, the bank could send the atm incorrect data making the account statement incorrect.

|  |  |  |
| --- | --- | --- |
| SorcePort | Message packet- this will be 256 bytes of data in ASCII. If the ATM does not receive a reply it will reattempt after 20 ms.  Node on message: Hello+EOM  Log in: Card # + Pin+EOM  User Data Request: Account + EOM  Withdraw request: Amount of money requested +eom  Deposit Request Cash/Check: input Cash value or check routing and value+eom  ServerReplys:  Node on: User please log in+EOM  Log in reply: y or n  User Data: Account balance, account name+ EOM  withdraw: y or n to deposit+ error if it occurs+eom  deposit: add Cash to account, or look for check routing and transfer funds+ return Account | Destination Port |
| Length of message-  Will tell the receiver the total length of the message it will be receive. |  |  |

Question 3:

A: <a,a,1> <b,b,1><d,d,1>

B: <b,b,1> <c,c,1><a,a,1>

C: <c,c,1> <b,b,1><f,f,1>

D: <d,d,1> <a,a,1><e,e,1>

E: <e,e,1> <f,f,1><d,d,1>

d reports to a:

A: <a,a,1> <b,b,1><d,d,1><e,a,2>//a ignored

B: <b,b,1> <c,c,1><a,a,1>

C: <c,c,1> <b,b,1><f,f,1>

D: <d,d,1> <a,a,1><e,e,1>

E: <e,e,1> <f,f,1><d,d,1>

F: <e,e,1> <f,f,1><c,c,1>

b reports to a:

A: <a,a,1> <b,b,1><d,d,1><e,d,2><c,b,2>

B: <b,b,1> <c,c,1><a,a,1>

C: <c,c,1> <b,b,1><f,f,1>

D: <d,d,1> <a,a,1><e,e,1>

E: <e,e,1> <f,f,1><d,d,1>

F: <e,e,1> <f,f,1><c,c,1>

…

….

All reports done

A: <a,a,1> <b,b,1><d,d,1><e,d,2><c,b,2><f,e,3>

B: <b,b,1> <c,c,1><a,a,1><d,a,2><e,d,3><f,c,2>

C: <c,c,1> <b,b,1><f,f,1><d,a,3><e,f,2><a,b,2>

D: <d,d,1> <a,a,1><e,e,1><c,f,3><b,a,2><f,e,2>

E: <e,e,1> <f,f,1><d,d,1><a,d,2><b,a,3><c,f,2>

F: <e,e,1> <f,f,1><c,c,1><a,d,3><b,c,2><d,e,2>

Question 4:

|  |  |  |  |
| --- | --- | --- | --- |
| Destination | Cost | Next hop | Reason |
| A | 5 | R1 | No Change |
| b | 8 | R1 | Cost increases because the cost from r1 to node b increased |
| c | 7 | R2 | Cost Stays the same so no change |
| d | 7 | R1 | The old path had the cost 8 we were redirected to r1 to have a shorter path of 7 |
| e | 9 | R3 | No change |
| f | 9 | R1 | Node added to table at R1 cost +1 |