CS 2204-01 Communications and Networking

Instructor: Professor Sharon Sisitzky

Name: Ryohei Hayashi

Learning Journal Unit 6

**1. In the TFTP protocol:**

* **If the client changes its port number on a subsequent connection, but the server does not, what prevents an old-duplicate data packet sent by the server from being accepted by the new client?**
* **If the server changes its port number on a subsequent connection, but the client does not, what prevents an old-duplicate data packet sent by the server from being accepted by the new client?**

In the TFTP protocol, each data packet or ACK is given a unique identifier (e.g. sequence number) due to the nature of the communication. This system allows the following mechanisms to work even if the client or server changes its port number:

If the client changes the port number but the server does not, the reason why the old duplicate data packets are not accepted by the new client connection is that the client uses a different transaction ID (or sequence number) for the new connection and therefore packets from the old connection from the old connection are ignored.

If the server changes the port number but the client does not, likewise, the old data packets will not be accepted by the new connection because the new transaction ID is used on the server side. The new connection will expect the new transaction ID and the old packet will be ignored because the transaction IDs do not match.

**2. In an RPC-like protocol in which multiple requests can be outstanding, and replies can be sent in any order: Assume that requests are numbered and that ACK[N] acknowledges reply[N]. Should ACKs be cumulative? If not, what should happen if an ACK is lost?**

In protocols such as RPC, ACKs need not be cumulative, as replies can be sent in any order with multiple requests pending. With cumulative ACKs, if some ACKs are lost, it becomes necessary to re-acknowledge all messages received before the lost ACK. However, with non-cumulative, it is possible to design a system that sends a separate ACK for each reply and only requires retransmission for that particular message if the ACK is lost. This allows efficient network traffic management and increases reliability while saving resources. If an ACK is lost, the sender detects a timeout and responds by retransmitting the corresponding request or requesting a specific ACK again.

Word-Count: 266