CSI3344 Distributed Systems

Workshop 03

Q1. In many layered protocols, each layer has its own header. Surely it would be more efficient to have a single header at the front of each message with all the control in it than all these separate headers. Why is this not done?

Q2. Why are transport-level communication services often inappropriate for building distributed applications?

Q3. A reliable multicast service allows a sender to reliably pass messages to a collection of receivers. Does such a service belong to a middleware layer, or should it be part of a lower-level layer?

The following questions are *Optional* (for those who are good at C):

Q4. Consider a procedure *incr* with two integer parameters. The procedure adds one to each parameter. Now suppose that it is called with the same variable twice, for example, as *incr*(i, i). If i is initially 0, what value will it have afterward if call-by-reference is used? How about if copy/restore is used?

Q5. C has a construction called a union, in which a field of a record (called a struct in C) can hold any one of several alternatives. At run time, there is no sure-fire way to tell which one is in there. Does this feature of C have any implications for remote procedure call? Explain your answer.

END OF THE WORKSHOP