If you want to keep this title, you should demonstrate the differences (in terms of ~~performance~~ network performance) between regular UDP and so reliable UDP.

O.K.

UDP: The Cost of Reliability

An Exploratory Implementation of UDP with Attempt at Reliability

Please donot use reliable ~~CBP~~ UDP Java program which is available from the Internet. !

[redacted]

***Abstract*—Indicates an experimental, Java-based simulation of the UDP transport protocol to determine the effects of reliability enhancement at the protocol level. Various application conditions are to be modeled and an evaluations are to be produced of the pre-and-post-enhanced UDP protocol models showing the effects of reliability-enhancement in the different scenarios to justify the use or lack of use of such enhancement.**

Keywords—transport protocols; UDP; simulation; reliability

I. INTRODUCTION

Among the layers of of networking in Internet operations, it is the transport layer that is most-closely connected to the valuable data that utilizes all of the value provided by the underlying layers. Transport protocols are responsible for the end-to-end coordination and communication of data-level information from application-to-application across the ocean of hardware and software that is the Internet. Among these protocols are the connection-oriented Transmission Control Protocol (TCP) and the connectionless User Datagram Protocol (UDP). Because UDP is connectionless, it is able to bypass the handshaking procedures and overhead of its contemporary TCP. This allows UDP to operate much more efficiently in terms of data delivery. However, this efficiency comes at the cost of reliability. UDP is considered a "stateless" protocol, which means that once a message is sent, the protocol effectively forgets about the message, not retaining any state information for that message. It is up to the application to deal with this shortcoming and provide any necessary end-to-end reliability necessary. UDP is not without extensive potential uses, however. The simplicity and unidirectional nature of UDP make it an ideal candidate for broadcast-oriented applications and those applications where time-sensitivity of data reception would make retransmission essentially useless. Gaming applications have long made use of UDP because of its low overhead and allowance of relatively-large data chunk transmissions. The transaction-oriented nature of UDP makes it an optimal choice for applications that can be modeled around such a protocol. Given all of these factors, is it really necessary for UDP to be so unreliable? It is the aim and scope of this project to explore this question through experimental implementation of the UDP protocol. Using a "control" implementation of a basic UDP simulation, system properties including transmission delays, ~~end-to-end communications~~ times, and data quantity scale will be measured to provide quantifiable justifications for drawn analytical conclusions.

?

II. A MORE-RELIABLE UDP?

After gaining a stronger understanding of the workings, structure, and function of the UDP protocol, it is my intention to simulate and model the operation of the UDP protocol with enough detail in the internal structure of the implementation in the form of system parameters and variables to extract statistical data and operational effects. It is a clear fact that increased data size increases transmission and reception time. It is my belief that data size will also produce a multiplicative effect on the degradation of transmission efficiency in the case of retransmission. This is perhaps the reason for the choice of error-corrective measures being included natively in the UDP protocol and yet even these measures require application support. Through exploration of the operations of UDP under various design-modifications intended to enhance UDP reliability, a more thorough and effective knowledge of UDP

How you can do it? Are you talking about "error detection"?