Final Report

Problem motivation:

Nowadays, consensus problems became more and more essential in industry infrastructures. There were a plenty of consensus protocols to solve this kind of problems, such as Zookeeper protocol; however, Paxos algorithm almost represented the whole consensus problem area. So here is a question, what is Three-Army problem? This is the description I borrow from our project proposal:

In a battle, there are three blue army and one red army. The red army is too strong to be defeated by one blue army. Only way to defeat red army is that gathering over one blue army (the majority of all the blue army) to attack red army in the same time. These blue army need to negotiate with each other and get a specific time to attack red one together and they will follow the rules behind.

1. Every army have one staff officer (proposer) to provide a specific time; every army also

have one general (acceptor) to decide which time the blue army (learner) will attack red army; Before one staff officer notify the blue army to attack, this officer need get agreement from at least two generals among these blue armies.

1. Every staff officer will give proposal in two stage (prepare/commit) and the proposal will be given with a proposal ID.
2. If there is any confliction in preparation stage, the general will make decision according to the number id. The bigger, the winner.
3. If the staff officer receives general 's decided time, he or she must change his or her proposal time.

The reason why we choose the Three-Army problem as our topic is that the Three-Army problem can be simulated by the roles in Paxos, proposers, acceptors and learners. We could get further comprehension about consensus problem and Paxos mechanism from this project.

Design goals or performance questions:

In the final solution, we want to use Dos console to simulate different roles in Three-Army problem. And each army will be represented by an individual computer. On each computer, we will open three console windows to simulate proposer, learner and acceptor individually. After the programs finish their running, each learner will be received the same value by the proposers. And the majority of these proposers should get the same value. We use numbers to represent proposals in Paxos protocol. Furthermore, we want to simulate the Paxos solution with bad network issue. For example, when three computers are running the final solution, we will make one of these acceptors offline. We will shut down the console window of this acceptor. If our code is right and solution is appropriate,