We propose a self-stabilizing protocol for congestion control in distributed systems. Our protocol is designed to work on the application layer of a peer-to-peer system in order to reduce the message load for each single peer. The problem can be reduced to a scenario where n peers (called \emph{clients} in this paper) want to continuously send messages to a fixed common peer (called \emph{server} in this paper). The server has only limited information about the clients and needs only $O(W + \log n)$ bits of internal storage in legitimate states where W is the number of bits needed to store addresses of clients. Particularly the server's information can be corrupted in initial states. Once the server receives too many messages in a round, messages do not arrive at the server anymore. We want to optimize the probability for a message to successfully arrive at the server.Each client maintains a value indicating the probability to send a message to the server in a round. Our protocol is able adjust all probabilities of the clients using the average rule.Once this is done, the probability for a message to successfully arrive at the server is a very high constant that can also be adjusted through parameters $L,R \in \mathbb{N}$ for our protocol. Furthermore fairness among all clients is given, i.e., all clients eventually reach the same probability to contact the server. Our protocol lets the system converge from any initial state where initial client probabilities and server variables are not necessarily well-defined, i.e., they may have arbitrary values. Finally we state a lower bound on the convergence time for our model and compare it with the convergence time of our protocol, indicating that our protocol performs asymptotically optimal w.r.t. the lower bound in cases where the minimum client probability is less or equal to $1/n^2$ initially.