(a) Assume that using the described protocol, we get a set S that is not conflict free. Then there must be 2 processes P_i and P_j in the set S that both picked the value 1 and are going to want to share the same resource. But this contradicts the way our protocol was implemented, since we selected processes that picked the value 1 and whose set of conflicting processes all picked the value 0. Thus if P_i and P_j both picked the value 1, neither of them would be selected and so the resulting set S is conflict free. For each process P_i, the probability that it is selected depends on the fact that P_i picks the value 1 and all its d conflicting processes pick the value 0. Thus $P[P_i\text{selected}] = \frac{1}{2} * (\frac{1}{2})^d$. And since there are n processes that pick values independently, the expected size of the set S is $n * (\frac{1}{2})^{d+1}$

(b) Now a process P_i picks the value 1 with probability p and 0 with probability $1 - p$. So the probability that P_i is selected (i.e. P_i picks the value 1 and its d conflicting processes pick the value 0) is $p * (1 - p)^d$. Now we want to maximize the probability that a process is selected. Using calculus, we take the derivative of $p(1 - p)^d$ and set it equal to 0 to solve for the value of p that gives the objective it's maximum value. The derivative of $p(1-p)^d$ is $(1 - p)^d - dp(1 - p)^{d-1}$. Solving for p, we get $p = \frac{1}{d+1}$. Thus the probability that a process is selected is $\frac{d^d}{(d+1)^{d+1}}$ and the expected size of the set S is $n * \frac{d^d}{(d+1)^{d+1}}$. Note that this is $\frac{n}{d}$ times $(1 - \frac{1}{d+1})^{d+1}$ and this later term is $\frac{1}{e}$ in the limit and so by changing the probability, we got a fraction of $\frac{n}{d}$ nodes. Note that with $p = 0.5$, we got an exponentially small subset in terms of d.

[1] ex131.386.529