Peer-to-peer networks – (due till March 18, 2009)

Exercise 3.1: Random Linear Fountain Codes (part 1/2)

In the last exercise, the receiver gathered packets until a large enough set of random vectors (and data packets) was obtained that allowed to generate the original file.

So far we did not obtain linearly dependent vectors from the random number source.

1) Write a program* which draws n bits from a random number generator and combines them into a vector. This is repeated for n vectors. A simple routine should try to invert the n x n matrix (according to the modulo-2 arithmetic used in the last exercise). Your program need not be optimal. Keep it as simple as you like.

2) In the peer-to-peer context, we hope to reconstruct a file split into n packets after receiving n packets. How often are n packets sufficient, how often do you need 1, 2, 3, ... additional packets? (test for n=8, 16, 64)

3) What is the average number of additional packets for different values for n? (again, simulate for n=8, 16, 64)

*(In the next exercise, you'll need your program for a small extension)