STUDY ON THE PERFORMANCE OF CODING-ASSISTED SWARM COMMUNICATIONS

Anghel BOTOŞ Mihai P. ŞTEF Aurel VLAICU
Technical University of Cluj-Napoca
15 C. Daicoviciu str., 400020 Cluj-Napoca, Romania, Tel: +40264401906, Anghel.Botos@com.utcluj.ro

Abstract: Swarm communications represent some of the main solutions proposed for large content distribution in the current and future data communication networks. One of the possible solutions to improve the performance of swarm communication is represented by the use of Digital Fountain and Network Coding techniques integrated in the data exchange between swarm peers. The paper proposes a comparative study of the non-coded and coding assisted swarm communications in different settings related to the amount of employed signaling, supersymbol size, supersymbol number, and number of initial seeds. The study is based on computer simulations performed in some particular network topology and the goal is to identify the settings of coding assisted swarm communications which ensure significant decrease of its content distribution time.

Key words: *Content distribution, Digital Fountain, Network Coding, swarm communications.*

I. INTRODUCTION

Nowadays, large-scale content distribution mechanisms are a subject of great interest. Numerous applications require the distribution of large quantities of data to a very large number of users. The most common examples are critical software updates, content sharing applications for multimedia, etc. Recently a new paradigm for content distribution has emerged based on a fully distributed architecture where regular computers are used to share their resources and form a cooperative network [1]. Current research regarding such mechanisms tries to integrate coding techniques to improve performance and availability.

Integrating these techniques in protocols controlling swarm communications requires answers to several practical questions: What type of coding has to be used? Which is the amount of the required signaling? Which is the optimal supersymbol size to be used? In which situations does the use of coding significantly improve the overall transmission performance? Which are the effects of the sources leaving the swarm? etc.

Several publications consider these problems [1] and try to give answers to some of these questions by employing computer simulations:

- Can Digital Fountain (DF) [7] coding improve the quality of swarm communications? DF, being a source coding technique, allows an easier implementation than Network-Coding (NC) all processing being done in the source nodes only.
- Which are the optimal settings for the NC solution in the swarm context? The study considers the problem of finding the optimal number of coding groups and the optimal size of the supersymbols when NC is employed.

To study the above issues we performed computer simulations using various implementations of swarm content distribution mechanisms. Section II presents the general network topology model used in the simulations, Section III introduces our main performance indicators followed by Section IV which presents in detail all the scenarios together with the obtained results. Section V presents the conclusions of our study.

II. NETWORK TOPOLOGY MODEL

In our scenarios we considered a certain number of peers *N* taking part in the distribution of some content, usually large in size. The peers are connected through an arbitrary topology network, each peer having an access link to this network. The access links of the peers to the network are characterized by their uplink and downlink bit rates, being the only limitation imposed on the transfer of data between the peers. The uplink and downlink bit rates of the access links do not need to be equal, since in practice this asymmetrical case is the one most often encountered.

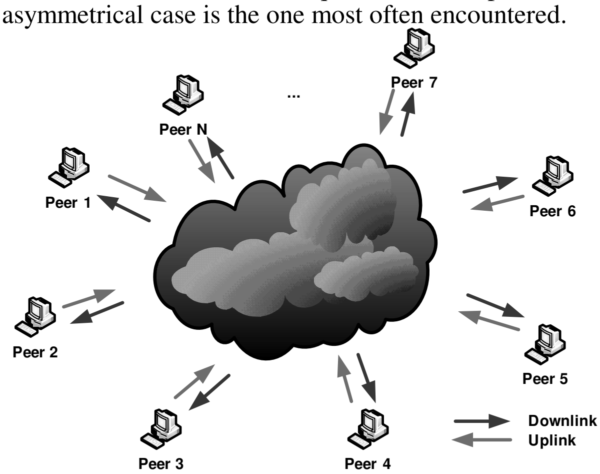

Figure 1. Network topology considered for simulations.

The network itself is capable of transferring any bit rate,

Manuscript received February 12, 2009; revised April 14, 2009

without errors, between any two peers, and without congestion. Thus, this is an ideal model for the network, but it is still suitable to assess the potential of the studied content distribution algorithms. This swarm is completely distributed, i.e. there is no central entity present in this topology. The operation of the swarm relies solely on the exchange of data/messages between the users present in the swarm.

All swarm intelligence is implemented at the application layer in the OSI model. The messages exchanged between the peers are considered to travel across a TCP/IP/Ethernet stack network. As a result the specific header sizes are used when computing the packet sizes in the simulations, and the time durations needed for their transfer.

We have also assumed that each peer knows all the time which peers are on-line in the swarm, thus it is no problem for any peer to contact anyone in the swarm. There are various ways to implement the necessary data exchange between the peers in order to make this information available, for example DHT (Distributed Hash Tables), but this is beyond the scope of this paper.

III. PERFORMANCE METRICS

At the start of all scenarios, one or more peers have the entire content that needs to be distributed, while the others have no part of this content. The purpose of the swarm is to distribute the content to all participants, therefore our main performance metric will be the time required to fulfill this purpose.

Besides this metric we will also observe how the availability of the content evolves in time. This is useful in order to draw conclusions upon the behavior of these techniques in the presence of churn (nodes entering/leaving the swarm during the download or shortly after they finished the download).

For some of the scenarios, where the initial sources leave the swarm after a certain amount of time, it is sometimes impossible for all users to download the whole content. These particular cases will not be taken into consideration in our performance estimations.

IV. SCENARIOS AND SIMULATION RESULTS

All our simulations compare scenarios where no coding is used against a scenario where some form of source coding is involved (either a Digital Fountain or a Network Coding type of coding). The specifics of the download strategy will be explained for each scenario in the following sub-sections, and the results obtained from each scenario will be presented as well. All scenarios consider content of a certain size that needs to be downloaded by all peers present in the swarm. The whole content is divided in supersymbols of data, which are codewords of the code employed, the transfer being performed one supersymbol at a time. A supersymbol denotes an elementary codeword which might be large compared to codewords from usual codes, having a size on the order of kilobytes or more, the used DF and NC codes being transparent in operation to its length. For the non-coded scenarios, a supersymbol is nothing more than a block of data from the original content.

IV.1. Non-coded vs. DF coding

For this scenario we compared a content delivery mechanism that uses no source coding to a mechanism that is based on a Digital Fountain type of coding.

DF coding refers to an erasure coding technique that comprises rate-less codes having the property that an encoder can produce from a finite amount of input data/symbols an almost infinite amount of encoded data/symbols. In order to be able to recover the initial data it is sufficient to correctly collect/receive *any* number of distinct encoded symbols that is only slightly higher than the initial number of symbols. Such codes are characterized by the average overhead required for the recovery of the initial data, which is the ratio between the number of additional symbols needed for complete recovery of the initial data and the number of initial data symbols. The operation of these codes is completely transparent to the symbol size (i.e. the symbols can be any length bit strings even hundreds of thousands), therefore the blocks into which the original content was divided are exactly the symbols that make up the original data. Different implementations of such codes exist, for example LT-Codes and Raptor Codes. For the DF-coded scenario we considered a code with a 5% overhead.

We tested two downloading strategies for this scenario. The two strategies are called *"No map"* and *"With map update"*.

The *"No map"* strategy, without coding works as follows: all non-source peers start by requesting a supersymbol of data from a random peer. If the contacted peer is one of the initial sources, it will respond by transmitting one random supersymbol of data from the original content back to the requesting peer. Otherwise it will send back one random supersymbol of data from the ones that it has already downloaded from other participants, or it will just ignore the request if it still does not have any yet.

For the same strategy, when DF coding is applied, sources respond to requests by transmitting back encoded supersymbols of data, each new encoded supersymbol of data being different than any other sent back by that source up to that moment of time. Any source can produce an almost infinite number of encoded supersymbols, all of them being unique. Peers which are not sources, respond to requests by sending back encoded supersymbols that they have downloaded from other peers. Once a peer becomes a source (by receiving a number of distinct encoded supersymbols that is 5% larger than the initial number) it will respond to future requests by producing new encoded supersymbols.

Figure 2 presents some simulation results for this scenario using the *"No map"* downloading strategy. The number of peers in the swarm was 20. Results for coded and non-coded are presented, for various numbers of initial sources. The time axis is normalized with respect to the highest download time needed in these simulations.

All scenarios using coding perform much better than the scenarios using the same parameters without coding. For the non-coded scenarios peers waste a considerable amount of time transferring duplicated supersymbols.

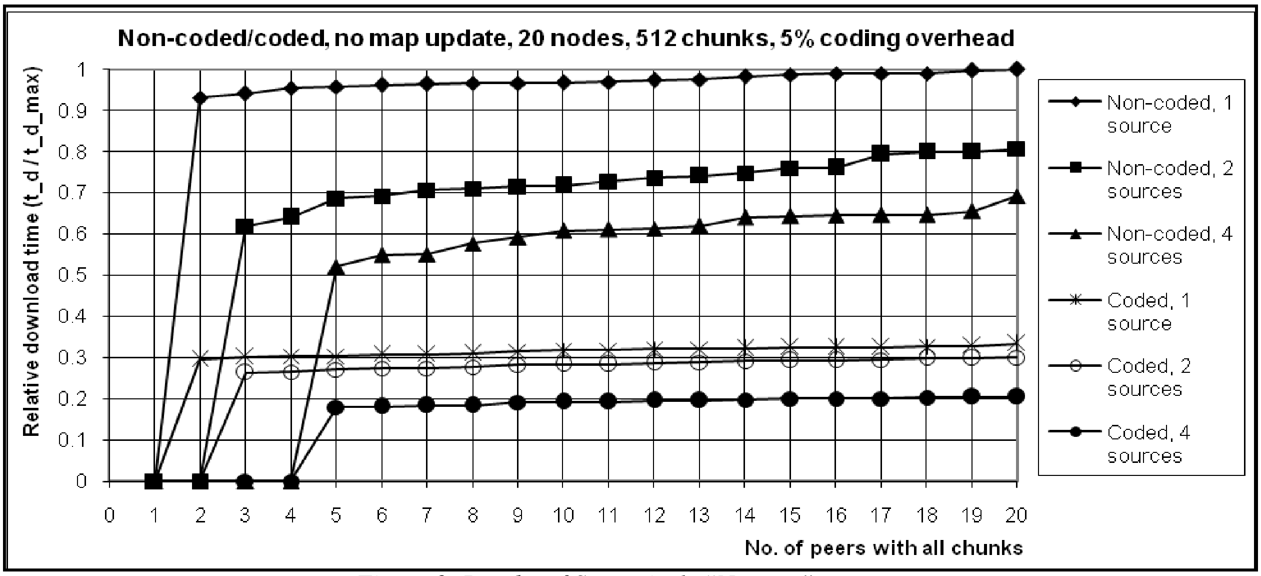

Figure 2. Results of Scenario 1, "No map" strategy

Duplicated supersymbols arise from the fact that peers respond always with a random supersymbol from the ones they possess. Since no coding is performed here, the total number of distinct supersymbols is constant (512 in our particular case). As a result a peer may respond to a request with a supersymbol that the requesting peer already has. This way time is wasted while the requesting peer does not progress with his download.

On the other hand, for the coded scenarios the total number of distinct supersymbols in the swarm is not constant, since sources respond always with a new encoded supersymbol. As a result the probability of a peer receiving a duplicate symbol as an answer to a request is lower; hence less time is wasted for the transfer of such duplicate supersymbols.

The effect of these duplicated supersymbols on the total download time is evaluated using the average number of duplicated supersymbols for each original one and for each participating peer. The formula is given in (1).

$$AvNDB = \frac{N_{duplicated}}{N_{uncoded}} \frac{1}{N_{peers}} \qquad (1)$$

For the non-coded case, in our scenarios, this performance indicator reached values even greater than 4, meaning that for every useful supersymbol that was transferred by each peer, another 4 supersymbols have been transferred uselessly through the network, consuming important bandwidth.

The *"With map-update"* strategy works similarly with the *"No map"* strategy, but it introduces additional signaling information between the peers. Whenever a peer makes a request to another peer, it includes in the request also a list of all the supersymbols that it currently owns.

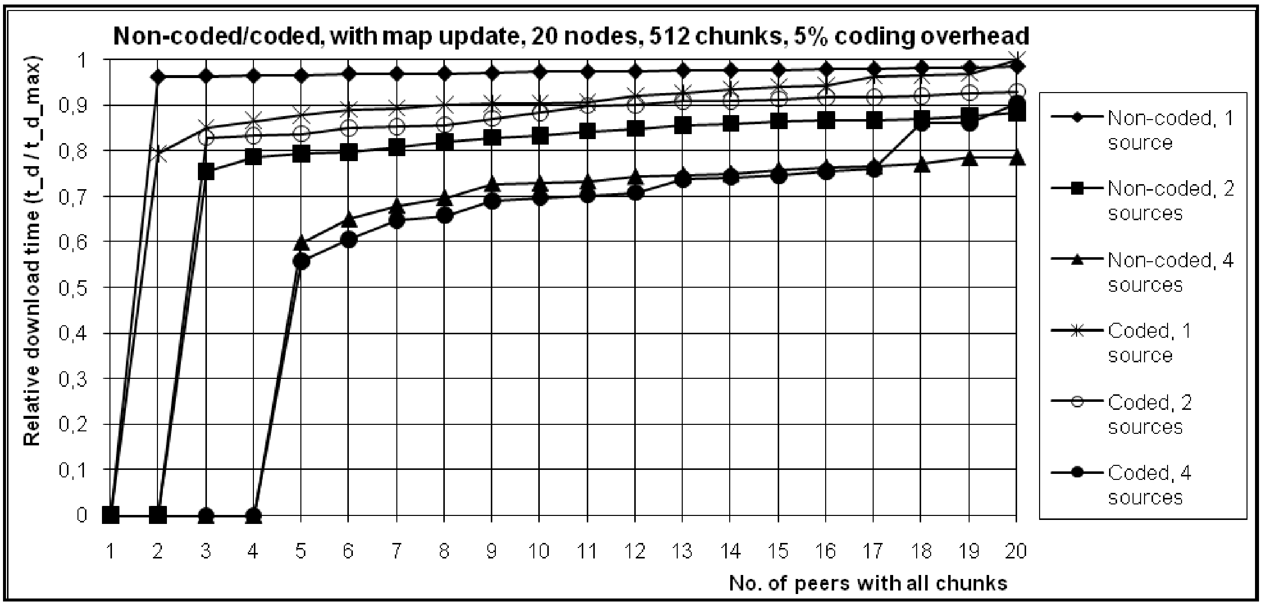

Figure 3. Results of Scenario 1, "With map-update" strategy

To minimize the amount of signaling traffic, a peer does not send the whole list of supersymbols it owns. It sends only the additional supersymbols it has collected since the last request issued to the same peer. The peer will respond to the request only if it has a symbol that the requesting peer does not own.

Using this strategy the number of duplicated supersymbols for the non-coded case will be drastically reduced, in this case *AvNDB* did not exceed a value of 0.5, meaning an 8-time improvement of this performance indicator.

For the coded case this strategy does not have an important impact. It is worth mentioning that duplicated supersymbols still appear, because these map-updates do not offer a real-time map of the supersymbol distribution in the swarm. By the time a peer responds to a request with a certain supersymbol, another peer might respond to another request originating from the same initial peer with the same supersymbol, thus wasting precious download time.

Results for this downloading strategy are presented in *Figure 3*. As one can observe, the increase of the number of initial sources leads to decrease of the download time needed, as it was expected. It is also worth mentioning that in this case the improvement brought by the use of DF type coding does not bring significant improvement any more to the total download time.

IV.2. Peers leaving the swarm

This scenario extends the previous one, using non-coded vs. DF coded transmission and the *"With map-update"* strategy, but considers the situation when sources leave the swarm before all peers have been able to complete their downloads. To evaluate this scenario, we have observed the minimum number of supersymbols that a source needs to upload before leaving the swarm, so that it is still possible for all peers to complete their downloads.

The results of these simulations are presented in *Table 1.*

Coding type/No. of peers	1 source	2 sources	4 sources	8 sources
Non-coded/20 peers	580	470	410	220
Coded/20 peers	538	270	136	68
Non-coded/200 peers	590	500	420	250
Coded/200 peers	538	270	136	68

Table 1. Minimum number of supersymbols which have to be transmitted by each source (seed)

The above results, together with others that are not presented here, show that DF coding allows a significant decrease of the number of supersymbols that need to be generated by each source in order for all peers to still be able to finish their download. This is important, since many peers which are initial sources participating in P2P networks, are not willing to allow their machines to take part for an extended amount of time in the swarm.

This improvement results from the fact that all generated supersymbols are distinct, whereas for the non-coded case, the total number of distinct supersymbols remains constant, no matter how many supersymbols are uploaded by each source.

IV.3. Non-coded vs. Network coded

The third and last scenario that we considered for our study compared the *"With map update"* strategy of the non-coded case to a network-coding based swarm operation.

In the case of the network coding based swarm all peers are allowed to perform coding on the data supersymbols that they own. Network coding consists in computing linear combinations of the supersymbols that a peer owns. The coefficients used in computing these linear combinations are elements of some GF. DF coding practically is also a particular type of network coding using coefficients from *GF(2)* and some specially crafted distributions for these coefficients. As we have seen from the previous scenarios DF coding cannot bring a significant improvement in the download time against a well designed download strategy for the non-coded case.

In the following we will describe in more detail the functioning of the network coding assisted swarm. Let us choose some Galois Field *GF(n)*. Divide the whole initial content in N supersymbols $\mathbf{x}_1 \ldots \mathbf{x}_N$ having their length in bits an integer number m times n. This choice leads to the supersymbols being m-dimensional vectors, having the magnitude in every dimension an element of *GF(n)*. When a source is requested a supersymbol from a peer, the source will produce an encoded supersymbol $\mathbf{y}$. The encoded supersymbol is produced by choosing $a_1 \ldots a_N$ random coefficients from *GF(n)* according to (2).

$$\mathbf{y} = \sum_{i=1}^{N} a_i \cdot \mathbf{x}_i \tag{2}$$

The multiplication of a coefficient a_i with $\mathbf{x}_i$ is the multiplication induced by the Galois Filed of each component of the vector $\mathbf{x}_i$ with a_i. The source will respond to the request with the resulting vector $\mathbf{y}$, and it will also include the coefficients a_i in the response.

In the following we will describe the operation of peers which are not sources. Suppose that such a peer owns l encoded supersymbols $\mathbf{y}_1 \ldots \mathbf{y}_l$. When receiving a request from another peer he will choose l random coefficients $a_1 \ldots a_l$ from *GF(n)*. The peer will then compute a new encoded supersymbol $\mathbf{y}$ using (3).

$$\mathbf{y} = \sum_{i=1}^{l} a_i \cdot \mathbf{y}_i \tag{3}$$

Taking into account that every encoded supersymbol owned by the peer, $\mathbf{y}_i$ is the result of a computation like the one presented in (2), we can write the following:

$$\mathbf{y} = \sum_{i=1}^{l} a_i \mathbf{y}_i = \sum_{i=1}^{l} a_i \sum_{j=1}^{N} a_j \mathbf{x}_j = \sum_{j=1}^{N} a_j^{'} \mathbf{x}_j \tag{4}$$

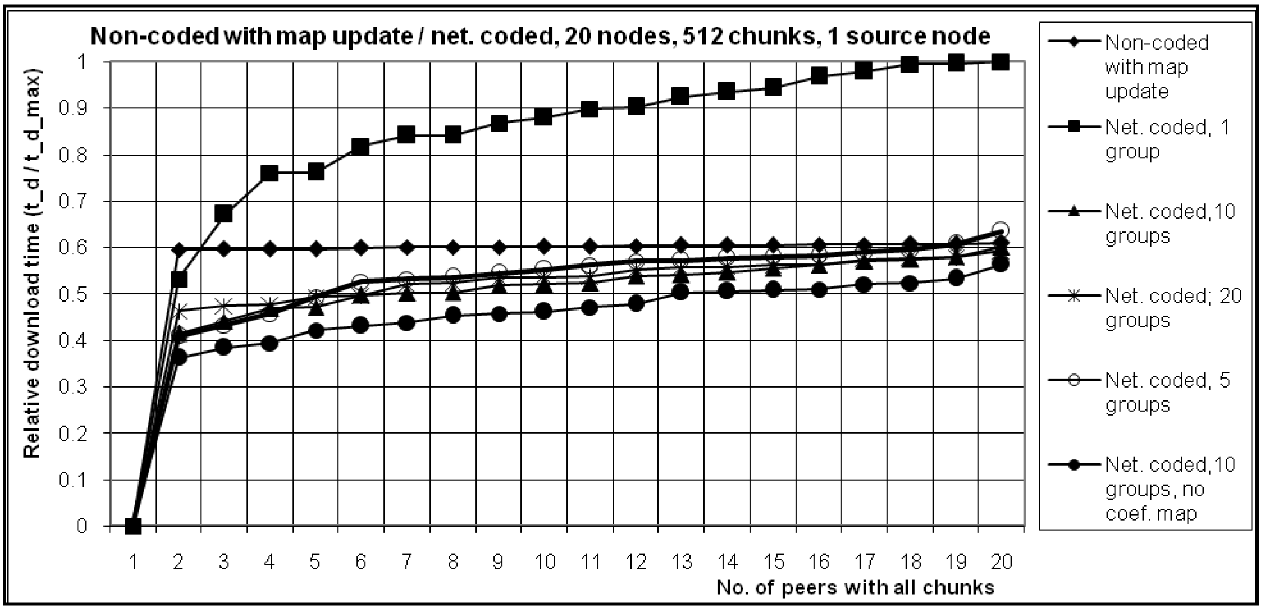

Figure 4. Results for Scenario 3 – Non-coded vs. Network coding assisted swarm

where

$$a_j' = a_j \sum_{i=1}^{l} a_l \qquad (5)$$

The peer will send back as a response the computed encoded supersymbol **y** and the list of coefficients a_j'.

To recover the original content a peer needs to have at least *N* linearly independent encoded supersymbols, from which by solving the linear system of equations it can obtain the original supersymbols.

One thing that needs to be mentioned here is the fact that the list of coefficients is sent together with the encoded supersymbol in the header of the packets. This generated a decrease of the efficiency of the transmission. One solution here is to split the set of supersymbols into smaller groups, and to perform the coding separately inside each group. This approach reduces the number of coefficients needed to be transmitted together with each encoded supersymbol.

Figure 4 presents some results from the simulations using this scenario. We used *GF(16)* and a varying number of groups into which the number of 512 supersymbols of data were divided.

If all the supersymbols of data are coded together in one group, the download time is greater than in the case of the non-coded operation. However, if several separate supersymbols groups are used, the download time can be decreased significantly. Also, for our specific values we can observe that the increase of the number of groups above a certain value does not bring any more significant increase in performance.

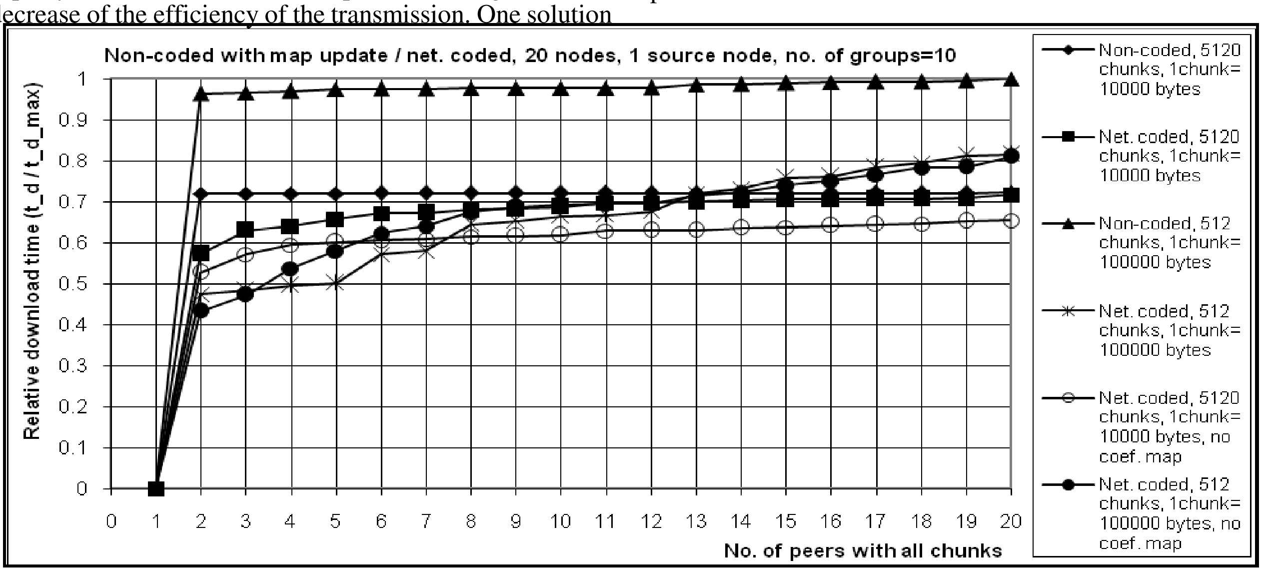

Figure 5. Results for scenario 3. Varying supersymbol size

Hence we can conclude that there is an optimum value for this parameter depending on all other parameters, like the total number of supersymbols, the size of the supersymbols, the order of the Galois Field in use, and so on.

The ratio defined by equation (6) gives an indicator of the average overhead induced by network coding for each exchanged supersymbol.

$$AvOB = \frac{No.of\ initial\ Supersymbols}{No.of\ Groups} \frac{GF\ order}{8 \cdot Supersymbol\ size[B]} \quad (6)$$

The total overhead induced by network coding must take into account also the probability of appearance of “duplicated” supersymbols (i.e. linearly dependent encoded supersymbols). This probability depends directly on the order of the employed Galois Field and on the distribution used for obtaining the coefficients from this field. The analytic expression for this probability is beyond the scope of this paper and we will not go further into the matter.

Some more results using the same scenario are presented in *Figure 5*. This time, the number of groups was fixed to 10 and we varied the size of the data supersymbols.

Figure 5 proves the fact that for certain values of the chosen parameters, network coding can bring significant improvements in the distribution of content to a large number of peers. For one of the simulated situations, the needed download time using network coding can be up to 35% faster than in the case when no coding is applied.

Another fact that can be observed from the previous simulation results is the evolution in time of the number of peers that have acquired the whole original content. For the non-coded approaches, it takes a significant amount of time for the first user to acquire the whole content, becoming himself a source, the rest of the peers following shortly after. On the other hand, for the network coding assisted swarm, the moment when the first peer is able to recover the whole initial content happens much earlier. This is an important fact, because this means that the resilience of the whole swarm to churn (peers leaving the swarm shortly after uploading content) is better. This behavior is easily explained when considering the whole encoding procedure, since it increases significantly the probability that for every new encoded supersymbol uploaded by a source, new information is introduced into the swarm.

V. CONCLUSIONS AND FUTURE WORK

The results obtained from our study show that the implementation of completely distributed content delivery mechanisms is feasible using different coding techniques. Some of the benefits of this type of content delivery mechanisms are the fact that by lacking a central entity (as it is the case with the tracker for BitTorrent) there is no single point of failure. Also the load on the initial source providing the content is greatly reduced, thus offering the possibility to be able to serve a much greater number of peers which request that content.

DF coding based swarms can improve the performance of the content distribution compared to non-coded based swarms where signaling between peers is reduced, but they also provide better performance when the initial sources leave the swarm early.

Network-coding based swarms offer significant improvements compared to non-coded swarms, even if complex signaling techniques are used, and can also be of great importance if the initial sources leave the swarm earlier. Of course these improvements are only visible if appropriate settings are ensured, e.g. the optimum division of the content in supersymbols and coding groups, the employed Galois field, etc.

Aspects presented in this paper need to be taken into account when designing viable solutions for swarm communications in real networks.

For future work we intend to continue our study by modeling network functionality in a more realistic manner. Another target of our future efforts is the aim to establish mathematical models and relationships that can express the achievable performance gains in such networks compared to classical approaches.

VI. ACKNOWLEDGEMENT

This work was partially funded by the European Union within the FP7-ICT-2007-1-216041-4WARD project.

REFERENCES

[1] C. Gkantsidis, P. Rodriguez Rodriguez, “Network Coding for Large Scale Content Distribution”, *IEEE Infocom*, Miami, 2005.
[2] J. Byers, J. Considine, M. Mitzenmacher, S. Rost, “Informed Content Delivery Across Adaptive Overlay Networks”, *IEEE/ACM Transactions on Networking*, vol. 12, pp. 767-780, 2004.
[3] M. Luby, “LT-Codes”, *Proceedings of the ACM Symposium on Foundations of Computer Science (FOCS)*, 2002
[4] A. Shokrollahi, “Raptor Codes”, *IEEE Transactions of InformationTheory*, vol. 52, pp. 2551-2567, 2006
[5] J. Byers, M. Luby, M. Mitzenmacher, A. Rege, “A Digital Fountain Approach to Reliable Distribution of Bulk Data”, *Proceedings of ACM SIGCOMM ’98*, pp. 56-67, Vancouver, 1998
[6] C. Fragouli, D. Katabi, A. Markopoulou, M. Médard, H. Rahul, “Wireless Network Coding: Opportunities & Challanges”, *IEEE Military Communications Conference MILCOMM*, 2007
[7] M. Luby, M. Mitzenmacher, A. Shokrollahi, D. Spielman, V. Stemann, „Practical Loss Resilient Codes“, 1997