Network Coding and Dispersal Information with TCP for Content Delivery

Francisco de Asís López-Fuentes[1](✉), Raúl Antonio Ortega-Vallejo[1], and Ricardo Marcelín-Jiménez[2]

[1] Universidad Autónoma Metropolitana-Cuajimalpa, Av. Vasco de Quiroga 4871, Cuajimalpa, 05348 Mexico City, Mexico
flopez@cua.uam.mx, raulantonio@protonmail.com

[2] Universidad Autónoma Metropolitana-Iztapalapa, Av. Atlixco 186, Iztapalapa, 09340 Mexico City, Mexico
cal@xanum.uam.mx

Abstract. Dissemination information from many sources to many receivers can be fundamental in different systems. However, the components of these system may present some failure type both the software and hardware. In addition, problems related to the communication networks such as limited bandwidth or packet loss should be present. The information dispersal algorithm (IDA) has been used as a good solution to offer fault tolerance. On the other hand, network coding is a coding method mainly used to increase throughput of a communication channel, which is useful to face the limited bandwidth in the communication networks. In this paper, we integrate both methods into a content distribution scheme. We use a hybrid peer-to-peer (P2P) network based on TCP in order to evaluate the performance of IDA and network coding in a joint operation.

Keywords: Network coding · Information dispersal · P2P networks · TCP

1 Introduction

Content delivery is very popular today, users can exchange several content types such as video, text messages, PDF documents, music, and photos. Some content such as video demand for a large amount of resources from the Internet infrastructures and cooperation between nodes play an important role. New problems such as fault tolerance, limited performance or bandwidth limitations have emerged. To face these challenges some techniques such as the information dispersal algorithm (IDA) [1] and network coding [2] have been proposed. Using IDA in the communication systems we can reach redundancy of the information in different levels and make better use of the storage capacities of the devices. In this way, we can configure fault-tolerance storage systems more efficient. On the other hand, network coding allows that the intermediate nodes encode the received packets for immediately forwarding the encoded packets to the end nodes [3, 4] and [5]. In other words, the packets received in the intermediate nodes are combined before

J. A. Marmolejo-Saucedo et al. (Eds.): COMPSE 2021, LNICST 393, pp. 63–72, 2021.
https://doi.org/10.1007/978-3-030-87495-7_5

forwarding them to the following nodes. Network coding uses elements of a finite field for the linear operations during the packet manipulation. The communications systems can be benefited by using network coding because this technique helps to increase the throughput and reduce the latency. Although different benefits by using network coding and IDA are reported in the literature [18, 19], both techniques have been used separately. This paper proposes an architecture where the IDA and network coding are combined. Peers are used as a way to improve the content delivery. Our proposed architecture is implemented using TCP. We use specific characteristics of this protocol such as the retransmission as a way to reduce the number of descriptors to be sent to each intermediate node.

This paper has the following organization. Section 2 presents related work to IDA and network coding. Section 3 introduces basic and fundamental aspects of network coding and IDA. In this section, we also explain some concepts about P2P networks and TCP. Section 4 presents our proposed architecture, where network coding and the information dispersal algorithm are combined under TCP. Our implementation and an initial evaluation is described in Sect. 5. Our conclusions are presented in Sect. 6.

2 Related Work

Several studies to address the impact of redundancy on P2P storage systems have been reported in the scientific community. Many of these works have related with network coding and the information dispersal algorithm (IDA). The works of [10, 11, and 12] evaluate redundancy assuming a static network. In [10], there is a thorough analytical study of the mean-time to failure (MTTF), bandwidth and storage load, for either simple replication or error-codes, particularly based on Rabin's IDA [1]. Failures on the set of storage devices are regarded as independent and identically distributed (iid) random variables. The main result shows that, compared to simple replication, IDA provides by far a higher MTTF, under the same amount of redundancy. Also, IDA requires less bandwidth and storage. In turn, [11] is focused on the excess of information required to different strategies, in order to provide the same level of availability for a collection of files. This study is based on Monte-Carlo simulations. Each experiment consists of a number of files which are processed according to a given strategy, and require allocation on the devices that make up the system. Nevertheless, devices are available with a given probability. The main result here is that IDA needs less space to provide the same level of availability, compared to simple replication. The second part of [11], is an effort to consider the long term behavior of a P2P storage network. For this purpose, any peer connection is assumed to last an exponential random time. The findings show that the files' availability presents a faster degradation when stored under IDA replication. In [12], the study compares three strategies: *uncoded random storage, traditional erasure coding based storage, and random linear coding based storage.* The result proposes that network codes are a good option to maintain the best efficiency of the system. Two important studies [13, 14] developed models that evaluate the amount of redundant information delivered by either simple replication or IDA, according to the level of availability required by the set of allocated files. Also, both works addressed the problem of maintainability, i.e. the cost of recovering and reallocating the information stored in a peer which is presumed to be left permanently. They introduced the concept of membership expiration time, in order to estimate whether a disconnection should be regarded as temporal or permanent. Using this parameter they developed a formula to evaluate the amount of average peer bandwidth required to keep any file within a given level of availability. Apart from simple replication, IDA and the hybrid approach by [13], the work of [15] introduces network codes. The study is based on analytical models and trace-driven simulations. Their findings show that network codes provide a very efficient mechanism to support information maintenance. Authors in [18] use IDA to make an efficient content delivery on a P2P network. A solution for a secure PLC (Power Line Communication) communication among end nodes based on the Information Dispersal Algorithm (IDA) is presented in [20]. In this work, the authors propose an efficient scheme using the physical characteristics of PLC channels of a smart grid.

On the other hand, network coding have used in several works for content delivery, because this technique has already proven to provide solutions to a variety of networking problems. For example, authors in [16] presents a unified linear program formulation for optimal content delivery in content delivery networks (CDNs). In this work, different costs and constraints associated with content disseminations are considered from the source node to storage nodes. The end users can do an eventual fetching of content of storage nodes. In [17] network coding is deployed in the backbone network for an IPTV architecture. In this case, network coding helps to increase network capacity while improving robustness against network faults. Authors in [21] introduce ND-POR, which is a scheme based on network coding but it is observed that these can be sensitive to a certain type of small corruption attack on their integrity and, to turn it around, the dispersal coding is applied. This proposal has the main target the cloud storage systems. To the best of our knowledge, in most of the literature reviewed, both network coding and IDA have been used separately. In this paper is presented a collaborative architecture which combines network coding and IDA with TCP over a hybrid P2P network in order to evaluate the performance of both technique during content delivery. Peers help to improve the packets delivery, while TCP allows to reduce the number of descriptors to be sent to the intermediate nodes.

3 Background

Information redundancy is an important concern of any communication system. This important characteristic can be reached via replication or coding. However, both techniques have different approach. For example, using replication we can allocate 3 copies of a file F, on 3 different storage sites. In this way, a failure in 2 storage sites can be tolerated by the system. Nevertheless, the effective storage capacity is only one third of the overall capacity. Now, we review information redundancy using IDA [1]. In this case, file F is transformed into n dispersals, each one of size $|F|/m$ and F can be recovered provided that any m out of n dispersals remain available (see Fig. 1). We can see in Fig. 2 that it is only required $n/m \times 100\%$ of the original file size as redundant information using IDA. Let us suppose that we use a particular implementation of IDA with parameters $n = 5$, and $m = 3$. Compared to the previous case of replication, the file can be transformed into $n = 5$ dispersal, each of size $|F|/3$. In this case is produced an excess of information of (5/3) × 100%, which is less than 300% of information required by the replication strategy. Thus, any 2 dispersals can be loosed, but it is still possible to recover the original file. Based on this approach, IDA presents an effective capacity almost twice compared with the replication strategy. This scenario is shown in Fig. 2.

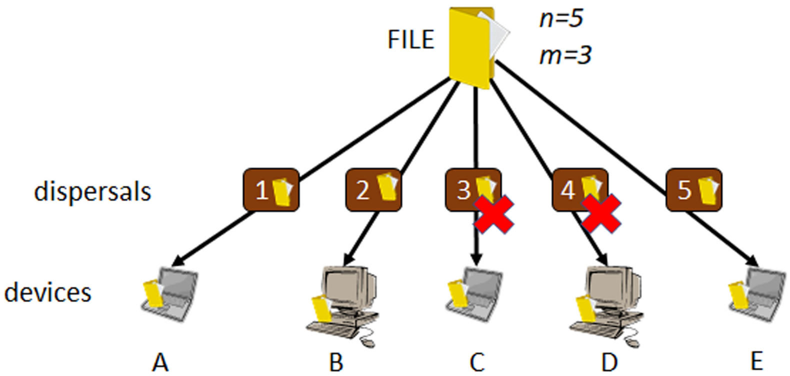

Fig. 1. IDA concept.

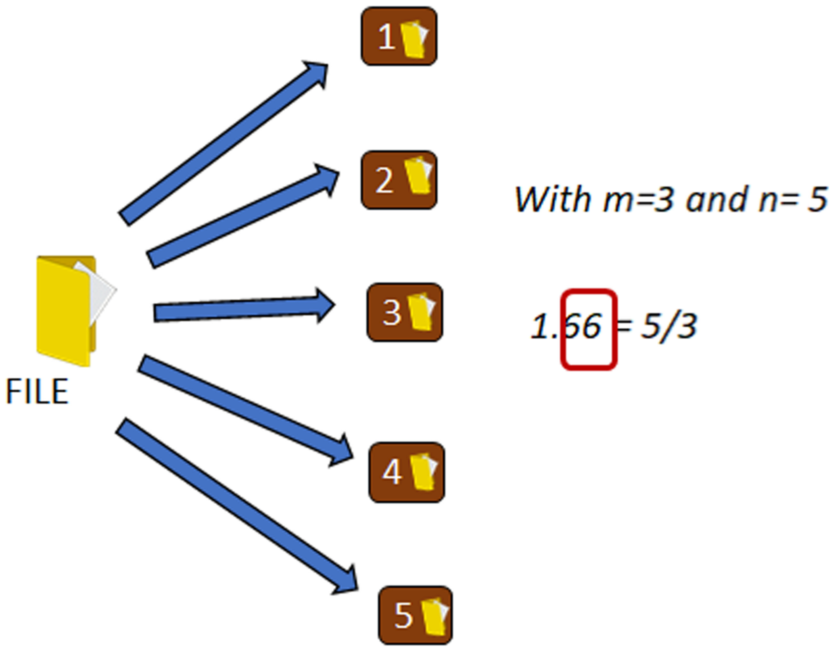

Fig. 2. An example of IDA for m = 3 and n = 5.

Network coding is an encoding technique used to increase the flow of packets without exceeding the link capacity [2]. To explain this scenario, we use a butterfly network (see

Fig. 3), which has a source node S, and two receiving nodes R1 and R2. Each edge has capacity of 1 as shown in Fig. 3a, and we can see that the maximum flow from the source S to any receiver (R1 or R2) has a value of 2. Simultaneously, source S sends bit b1 to receiver R1 and bit b2 to R2 (see Fig. 3b). Node 3 receives bits b1 and b2 from nodes 1 and 2, respectively, and it must send both bits to node 4. However, the link between nodes 3 and 4 requires two time units to send both bits. In contrast, using network coding (which is indicated with the operator $\oplus$ in Fig. 3c) the receiver R1 can recover both bits (b1 and b2), but bit b2 must be recovered from operation b1 $\oplus$ b2. Receiver R2 recovers both bits making a similar procedure as R1. In this scenario, node 3 is responsible to apply network coding. We can note that network coding allows to increase the multicast rate in the link from 1 to 2 bits/time unit.

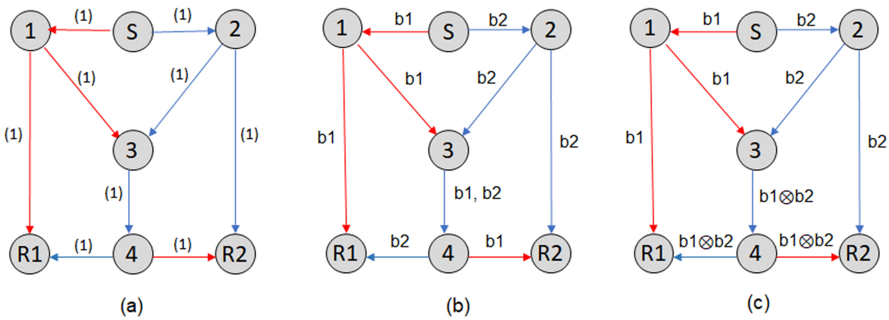

Fig. 3. Example of a communication network. a) Capacity of the edges, b) Traditional approach and c) approach with network coding.

P2P networks have become a popular paradigm for the next generation of distributed computing, and they are used to spread digital content to a large audience [6]. These types of networks are a research topic in several areas such as data communication networks, distributed systems, complexity theory and databases [7]. A P2P network is a virtual communication infrastructure deployed over a physical network. Nodes build a network abstraction on top of the physical network, and it is known as an overlay network. This overlay network is independent of the underlying physical network, and the connections between nodes are done using the Transmission Control Protocol (TCP). This protocol allows to abstract the physical connections in such a way that these are not reflected in the overlay network. The routing mechanisms use the logical tunnels implemented between nodes by the overlay network [8]. TCP is a transport protocol used to provide reliable delivery of data via a communication network [9]. Computers can exchange data with application programs in a way correct, secure and in order by using TCP. Communication networks can present packet loss during a transmission. To deal with this problem TCP uses retransmission to ensure the delivery of the packets. TCP can do n number of retransmissions, depending on the number of lost packets during the transmission. In this work, we combine IDA and network coding using TCP over a P2P infrastructure.

4 Proposed Model

This section describes our proposed model. Figure 4 shows our architecture, which is formed by 15 different nodes (peers and servers).

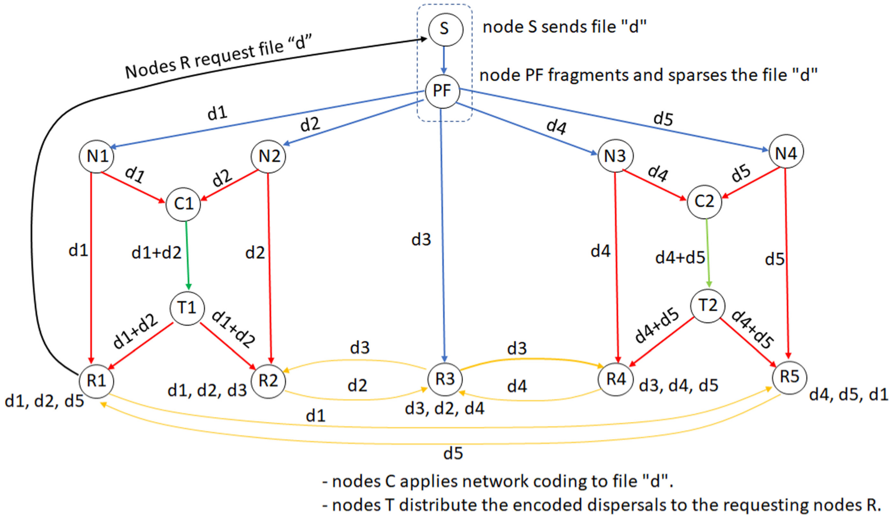

Fig. 4. Architecture combining network coding and IDA.

Node S is the source node, while node PF is the fragmenting peer, which is the responsible to fragment the file using the information dispersal algorithm (IDA). We use two butterfly schemes to deploy network coding. The nodes N1 and N2 are the source nodes in the butterfly 1, while N3 and N4 are the source nodes in the butterfly 2. Nodes C1 and C2 are the intermediate nodes which are responsible to do the network coding operation in the butterfly 1 and 2, respectively. Node T1 relays the encoded message received from the node C1 to the requesting nodes R1 and R2 in the butterfly 1, while node T2 relays the encoded message received from the node C2 to the requesting nodes R4 and R5 in the butterfly 2. The requesting peer R3 receives the descriptor from the node PF directly. All requesting nodes R are working as peers, which establish communication between them to share the descriptors and to have information about the contents that are shared within the network. For each delivery, a peer R creates a thread to distribute the received descriptor to other peers. The communication between a sending peer and a requesting peer is established through this thread. All nodes in the architecture have a specific role during the transmission, therefore they are renamed before that a file be sent to a requesting peer. All peers (nodes R) collaborate with each other to distribute a descriptor in our architecture. That is to say, the peers work as relay node too. This means that a peer receives a dispersal from the fragmenting node (PF) or node T and retransmits this dispersal to other peers in the architecture. For example, node R3 receives descriptor d3 from node PF, then d3 is relayed to nodes R2 and R4. On the other hand, node R1 receives descriptor d1 from node N1, then d1 is relayed to node R5. Nodes R2, R4 and

R5 have a similar behavior as node R1 because they only distribute a descriptor. Our proposed architecture works as a multi-source scheme because the nodes R receive three different dispersals from three different nodes.

Initially, the source sends the requested file by the requesting nodes R to the fragmenting node, which uses the IDA algorithm to fragment the file into five descriptors. Each receiver node R can recover the original file having only three descriptors. The fragmenting node requests the IP addresses of the nodes (N1, N2, R3, N3 and N4). Each of these IP addresses receives a descriptor from The PF node. After this, nodes N work as sources, and network coding is applied using two butterfly schemes. Nodes R1, R2, R4 and R5 recover two descriptors by decoding the message received from the nodes T. These receiving nodes with two descriptors establish communication with the other requesting nodes to delivery their descriptors. Because node R3 only receives a descriptor, this node should receive a descriptor from nodes R2 and R4. Thus, all requesting nodes R obtain the missing descriptor and they can assemble the original file using the IDA algorithm.

5 Implementation and Evaluation

Our work is in progress, and we have done an initial implementation of our architecture. This prototype has been developed in the C programming language for the Debian Linux operating system. Our prototype uses 14 containers and we performed 6 runs for each experiment. We transmitted different source vectors with the following dimensions: 1 MB, 5 MB, 10 MB, 15 MB, and 30 MB.

Our first experiment evaluates a content delivery using IDA without network coding. In this case, the butterflies are not done, and the nodes R receive the descriptors directly from the nodes N. Thus, the most intensive interaction occurs between the R nodes and the nodes N, which must retransmit the flow from the node PF. Nodes N work as relay nodes. Each vector is simulated as a broadcast, where because TCP is used. The nodes N require concurrent processes to emit a vector "d". The situation is similar for node PF when issuing all the vectors resulting from the IDA algorithm. In the second experiment, the topology shown in Fig. 4 is evaluated. In this case IDA and network coding are combined through the implementation of two butterfly schemes. Each butterfly applies network coding for the data streams coming from node PF. The nodes R work as peers and transmit the dispersal in a collaborative way. The network coding scheme used was the traditional network coding scheme, which is based on the use of arithmetic operations such as the binary XOR operation.

We measured the execution times for both experiments using two methods. The first method counts the time required by the processor, while the second method counts the real time of the simulator application, giving an expectation of the possible waiting time scenario for a user. In this way, the receiving nodes, which are the first nodes to be executed, accumulate the logs of the reception times. Results are shown in Figs. 5 and 6. We can see in Fig. 5 (IDA over TCP) the time costs for the execution of the simulator in simple transmissions of the scattered vectors. For a file of 1MB, we obtain a time of 0.632844 s for the processor, and a total time of 12.77619012 s (real time), which is the time it took for the simulator program to recover the complete source vector between each receiver. Figure 6 shows the results obtained from experiments using IDA with network coding. Here, we can observe that the transmission of a 1MB vector requires a processor time of 0.616914 s, while the total real time for the execution and termination of the simulator is 6.31214858 s.

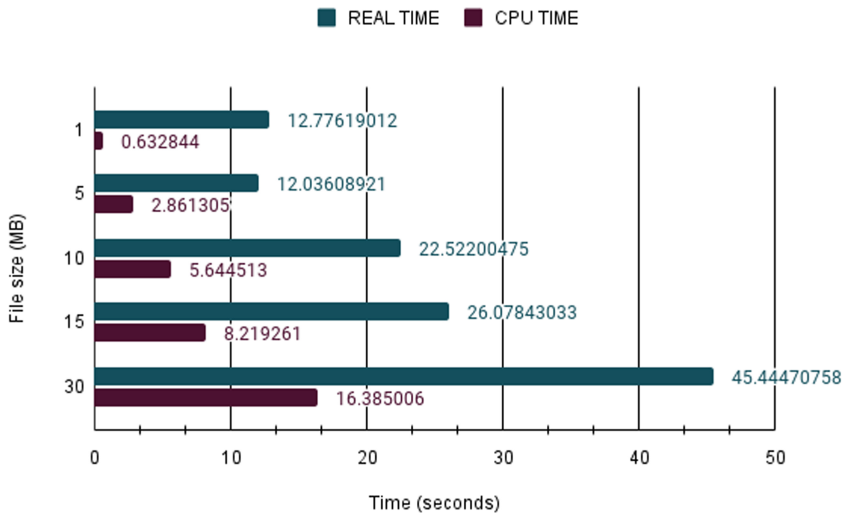

Fig. 5. Comparison of distribution time for architecture with IDA only.

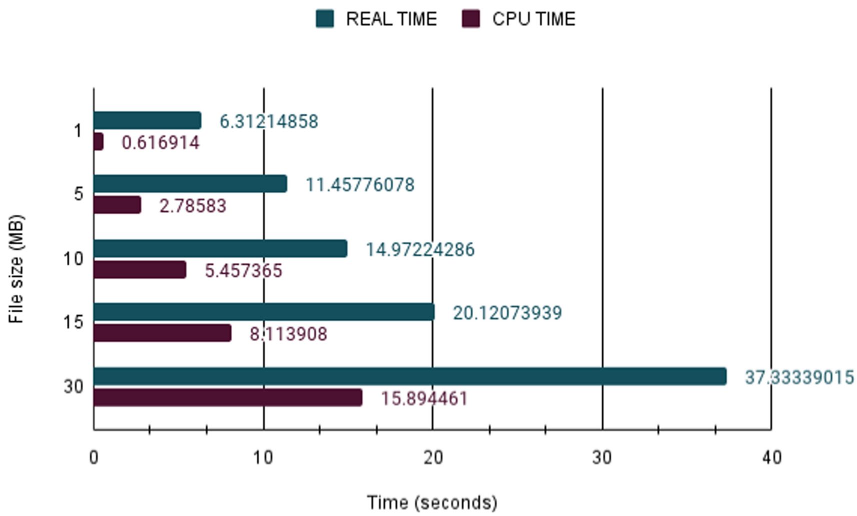

Fig. 6. Comparison of distribution time for architecture with IDA and network coding.

The used methods show that the CPU times could increase with the use of the network coding, because the intermediate nodes require more processing to perform the necessary operations of the schema. This could mean that the proposed architecture may generate a higher cost over time. However, the figures of real seconds, allow to visualize an expectation of the real time for the users (nodes R in Fig. 4) of the architecture. On the other hand, delivery time variations can be observed for different sizes of the source vector. We can also observe that for small vectors (e.g.: 1 MB and 10 MB) it might be convenient to use the proposed architecture. We are evaluating the performance of this architecture using larger vectors.

6 Conclusions

Content delivery plays an important role in the current communication networks. Because these services demand efficient distribution schemes. Network coding and the information dispersal algorithm (IDA) are techniques used to mainly improve the throughput of the communication networks and fault tolerance in the storage systems, respectively. Several applications using both techniques can be found in literature, but separately. In this work, we propose a distributed architecture which combines network coding with IDA to evaluate the impact of this strategy. Our work is in progress, and the preliminary results show that network coding combined with IDA can reduce the delivery time for files of small size. Our implementation has been done using TCP. Therefore, some specific characteristics of TCP such as retransmission of loss packets is very important. Requesting nodes working as peers play an important role because it allows these to work as relay nodes and not as simple leaves of a tree.

As future work we plan continue our experiments evaluating larger files to observe the behavior of our architecture under these scenarios. In addition, we are working in an implementation of our architecture using UDP in order to compare its performance with our implementation done in TCP. Data privacy can also be implemented in our architecture by using encryption techniques based on the AES algorithm.

References

1. Rabin, M.O.: Efficient dispersal of information for security, load balancing, and fault tolerance. J. ACM **36**, 335–3348 (1989)
2. Ahlswede, R., Cai, N., Li, S.-Y., Yeung, R.W.: Network information flow. IEEE Trans. Inf. Theory **46**, 1204–1216 (2000)
3. Chou, P., Wu, Y., Jain, K.: Practical network coding. In: 51st Allerton Conference on Communication, Control and Computation, Monticello, IL, USA (2003)
4. Gkantsidis, C., Rodriguez, P.R.: Network coding for large scale content distribution. In: IEEE INFOCOM 2005, Miami, FL, USA, pp. 2235–2245 (2005)
5. Sundararajan, J.K., Shah, D., Medard, M., Mitzenmache, M., Barros, J.: Network coding meets TCP. In: IEEE INFOCOMM 2009, Rio de Janeiro, Brazil, pp. 280–288 (2009)
6. Androutsellis-Theotokis, S., Spinellis, D.: A survey of peer-to-peer content distribution technologies. ACM Comput. Surv. **36**(4), 335–371 (2004)
7. Milojicic, D.S., et al.: Peer-to-peer computing. Technical report HPL-2002-57R1, HP Labs, Palo Alto (2002)
8. Ripeanu, M., Foster, I., Iamnitchi, A., Rogers, A.: In search for simplicity: a self organizing multi-source multicast overlay. In: 1st IEEE International Conference (SASO 2007), Boston, MA, USA, pp. 371–374 (2007)

9. Santifaller, M.: TCP/IP and NFS Internetworking in UNIX Environment. Addison-Wesley, Boston (1981)
10. Weatherspoon, H., Kubiatowicz, J.D.: Erasure coding vs. replication: a quantitative comparison. In: Druschel, P., Kaashoek, F., Rowstron, A. (eds.) IPTPS 2002. LNCS, vol. 2429, pp. 328–337. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-45748-8_31
11. Bhagwan, R., Moore, D., Savage, S., Voelker, G.M.: Replication strategies for highly available peer-to-peer storage. In: Schiper, A., Shvartsman, A.A., Weatherspoon, H., Zhao, B.Y. (eds.) Future Directions in Distributed Computing. LNCS, vol. 2584, pp. 153–158. Springer, Heidelberg (2003). https://doi.org/10.1007/3-540-37795-6_28
12. Acedanski, S., Deb, S., Médard, M., Koetter, R.: How good is random linear coding based distributed networked storage? In: 1st Workshop on Network Coding, Theory, and Applications (NetCod) (2005)
13. Rodrigues, R., Liskov, B.: High availability in DHTs: erasure coding vs. replication. In: Castro, M., van Renesse, R. (eds.) IPTPS 2005. LNCS, vol. 3640, pp. 226–239. Springer, Heidelberg (2005). https://doi.org/10.1007/11558989_21
14. Blake, C., Rodrigues, R.: High availability, scalable storage, dynamic peer networks: pick two. In: HOTOS 2003: Proceedings of the 9th conference on Hot Topics in Operating Systems, pp. 1–6 (2003)
15. Dimakis, A.: Network coding for distributed storage systems. In: INFOCOM 2007. 26th IEEE International Conference on Computer Communications, pp. 2000–2008 (2007)
16. Derek, L., Ho, T., Cathey, R.: Optimal content delivery with network coding. In: 43rd Annual Conference on Information Sciences and Systems, CISS 2009, pp. 414–419 (2009)
17. Kwon, M., Kwon, J., Park, B., Park, H.: An architecture of IPTV networks based on network coding. In: International Conference on Ubiquitous Future Networks ICUFN, pp. 462–464 (2017)
18. López Fuentes, F.A., Mendoza Almanza, J., Marcelin-Jiménez, R., Velázquez-Méndez, B.: Efficient content distribution and storage P2P system based on information dispersal. In: 6th International Conference on Control, Decision and Information Technologies (CoDIT) (2019)
19. Mendoza-Almanza, J., López-Fuentes, F.A.: Optimal network coding based on machine learning methods for collaborative networks. In: 6th International Conference on Control, Decision and Information Technologies (CoDIT) (2019)
20. Noura, H.N., Melki, R., Chehab, A., Hernandez-Fernandez, J.: Efficient and robust data availability solution for hybrid PLC/RF systems. In: Computer Networks (2021)
21. Omote, K., Tran, P.T.: ND-POR: a POR based on network coding and dispersal coding. IEICE Trans. Syst. **E98-D**(8), 1465–1476 (2015)