Distributed Storage Codes Reduce Latency in Vehicular Networks

Maheswaran Sathiamoorthy
University of Southern California, Los Angeles, CA, USA
Email: msathiam@usc.edu

Alexandros G. Dimakis
University of Southern California, Los Angeles, CA, USA
Email: dimakis@usc.edu

Bhaskar Krishnamachari
University of Southern California, Los Angeles, CA, USA
Email: bkrishna@usc.edu

Fan Bai
General Motors Global R&D, ECI Lab, Warren, MI, USA
Email: fan.bai@gm.com

***Abstract*—We investigate the benefits of distributed storage using erasure codes for file sharing in vehicular networks through realistic trace-based simulations. We find that coding offers substantial benefits over simple replication when the file sizes are large compared to the average download bandwidth available per encounter. Our simulations, based on a large real vehicle trace from Beijing combined with a realistic radio link quality model for a IEEE 802.11p dedicated short range communication (DSRC) radio, demonstrate that coding provides significant cost reduction in vehicular networks.**

I. INTRODUCTION

The recent development of the IEEE 802.11p WAVE (Wireless Access in Vehicular Environment) protocol [1] and the allocation of Dedicated Short Range Communications (DSRC) spectrum have increased interest in vehicular networking. This protocol enables vehicle-to-vehicle (V2V) as well as vehicle-to-infrastructure (V2I) communication (and vice versa), and capabilities like these open up a number of possibilities. Most applications focus on safety, such as avoiding rear-end collisions; extended braking [1], [2]; and detecting and disseminating information about potholes, bumps and other anomalous road conditions [3]. Recently, applications that concern entertainment and file sharing are also receiving attention and involve different challenges (e.g., AdTorrent [4], CarTorrent [5], FleaNet [6], C2P2 [7]).

Content access and vehicle file sharing would enable users to access movies, music, videos, and other relevant content. In this paper, using realistic trace based simulations, we investigate the possibility of exploiting inter-vehicular communication to enable P2P file sharing without the use of access points (APs). One possibility for content access is to use the cellular infrastructure, but recent reports suggest that with the increasing use of smart phones, cellular data bandwidth is likely to remain limited and expensive [8]. Another option is to use APs, but they may be hard to deploy in high densities. In addition, due to the latency of content access from the Internet, a vehicle quickly passing by an AP might not have sufficient time to download its desired data. In contrast, the WAVE/WAVE BSS modes of IEEE 802.11p allow for rapid V2V file transfers [1] over potentially longer contact durations (e.g., if the vehicles are traveling in the same direction). Nevertheless, we note that the coded storage techniques we explore in this paper could also be used in a heterogeneous network architecture which integrates V2V communication with V2I communication.

We identify two basic dissemination schemes that can be employed for data transfers in vehicular networks - one is the well studied ([4], [5], [9]–[13]) *push-based* mechanism and the other is a *pull-based* retrieval scheme which we study in this paper. It may be noted that push-based mechanisms work well for small sized data transfers such as traffic updates, pothole monitoring and other content that might be of interest to all users, but would fail to perform well for large file transfers of interest to only a few users (e.g., movies, long videos). This can be seen easily because traditional push-based schemes involve replicating the same file over multiple relays (e.g., epidemic routing [9], spray and wait [10], file swarming [4], [5]), which can be quite inefficient for large files. Other methods have been proposed such as the use of erasure coding [11], [12] and network coding [13], and even though they help reduce the delay and enhance the reliability, they fundamentally work by pushing more data than the file size into the network.

Hence in this work, we consider a P2P file sharing application where the files are stored in the nodes as a distributed repository and interested users retrieve these files on-demand. The amount of data downloaded by a node is no more than the file size (excluding control data), and thus such a pull-based scheme is not only efficient but will also scale well with the number of nodes, file size etc. But in order to improve the latency and reliability of content access, intuitively the files should be stored with some redundancy. Thus, in order to reduce the latency of file access, we shift the burden from the expensive bandwidth to the relatively inexpensive storage, thereby enabling additional applications to run. Previously Kapadia *et al.* [7] have suggested a similar scheme, where the content is stored using simple uncoded replication. The novel contribution of our work is that we recommend the use of erasure codes, especially for large files, and we run a series of simulations to show the performance improvement of coded storage compared to uncoded replication.

This research was sponsored in part by General Motors Research and Development; the U.S. Army Research Laboratory under the Network Science Collaborative Technology Alliance, Agreement Number W911NF-09-2-0053; NSF Career Grant 1055099 and a Research Gift by Microsoft Research.

We consider out of scope of this paper the orthogonal problem of the process by which the file repository is initially created and maintained. For this purpose other previously proposed schemes such as coded dissemination [14] or direct infrastructure download could be used.

In this work, we present a comprehensive performance analysis using a real vehicle trace consisting of 1,000 taxis in Beijing, combined with a realistic 802.11p DSRC Packet Delivery Ratio (PDR) model. These simulations demonstrate that coded storage substantially improves the timeliness of file downloads particularly when the bandwidth is limited as compared to the file size.

A. Related Work

The fundamental goal of this paper is to simulate a pull-based P2P vehicular content storage system and study the effect of coding on file download latencies.

In contrast to prior work such as [7] which assumes that the content in such a system is stored using uncoded replication, we advocate the use of erasure codes for storing content. Erasure coding consists of separating each file into k chunks of size $\mathcal{M}/k$ each and from these generating $n > k$ chunks of the same $\mathcal{M}/k$ size each. If a Maximum Distance Separable (MDS) erasure code [15] is used, *any k out of the n encoded chunks suffice to reconstruct the original file*. One specific family of codes that are almost-MDS and are suitable for our application are digital fountain codes. Initially proposed by Byers *et al.* [16] and later developed by Luby [17] and Shokrollahi [18], digital fountain codes are binary near-MDS (almost all sets of $k(1+\epsilon)$ chunks suffice to reconstruct the file with high probability, but we neglect ϵ for simplicity) and have very fast and simple encoding and decoding algorithms.

Erasure coding and uncoded replication have previously been compared in other contexts. For instance, [19] compares coding and uncoded replication for distributed storage in a wired system, and argues that coding is a clear winner as it provides mean times to failure that are magnitudes higher than that provided by replication. Our work on vehicular networks has a different focus, not on ensuring availability in the face of failures, but rather on reducing latency in the face of sparse and short-duration vehicular encounters. In this setting, we argue that coding is indispensable, particularly for large files.

Because they involve intermittent encounters, sparse vehicular networks can be considered examples of Delay/Disruption Tolerant Networks (DTNs). Closest in spirit to our work are two previous studies with an overlapping set of authors, who have examined the use of erasure codes in DTNs for reliably routing information between a particular source-destination pair [11], [12]. These studies provide a comparative analysis showing that the use of erasure coding can provide significant robustness to en-route path/node failures (the focus of [11]), as well as reduced latency (the focus of [12]), for push-based networks. In contrast, our emphasis in this work is on evaluating the latency and reliability of erasure coding for a pull-based network, specifically suitable for large files.

Also, seemingly related to our work are those that adopt network coding to handle content distribution in vehicular networks, e.g., CodeTorrent [13], VANETCODE [20], CodeOn [21], and VCD [14]. Again, an essential distinction is that these works focus primarily on pushing files and messages to other nodes, whereas our focus is on pull-based retrieval for large files.

II. MODEL AND PROBLEM SETUP

We assume there are N identical participating vehicles (or nodes) in a closed system of a vehicular network, each with a storage capacity of $\mathcal{C}$ bits allocated for the file sharing application. The total number of different files stored in the system is denoted by m; for simplicity, we assume that all the files have the same size of $\mathcal{M}$ bits (assume $\mathcal{C} \geq \mathcal{M}$) and are equally likely to be requested.

It is desired to distribute these m files among the nodes. It is assumed that the total available storage exceeds the total size of all files: i.e., that $N\mathcal{C} \geq m\mathcal{M}$. Denote $\alpha = \frac{N\mathcal{C}}{m\mathcal{M}}$ and note that we can store each file $\alpha \geq 1$ times throughout the system and saturate the available capacity in the system. We refer to α as the **system redundancy**, since it is the number of times each bit is stored in the system. In this paper, we consider and analyze the expected delay in downloading files when uncoded replication scheme and the coded storage scheme are used. For the uncoded replication scheme (also called uncoded storage), we simply store each file α times in the nodes ensuring that a node doesn't store the same file multiple times (maximal spreading). On the other hand, for the coded storage scheme, an (n, k) MDS code is used and each file is split into k chunks and encoded into n chunks of the same size. We set $n/k = \alpha$, equal to the total system redundancy. This is because, as the effective size of each file after coding is $n\mathcal{M}/k$, in order to saturate the system capacity, we need $N\mathcal{C} = m(n\mathcal{M}/k)$, yielding $\alpha = n/k$.

We focus on the latency experienced by a given sink vehicle that is trying to download one of the m files. The amount of data that a node can download from another upon an encounter is called the download bandwidth. Note that it is not a constant but is rather a random variable that depends on the contact duration and the link quality model used. We refer to d as the average **bandwidth constraint**.

Given all other parameters, we would like to determine the optimal values of n and k for coding. In order to do so, we note that each chunk has size $\mathcal{M}/k$ and so we want to choose k such that the chunk is downloadable within the average bandwidth constraint (d). Thus we want $k \geq \lceil \mathcal{M}/d \rceil$, while prefering lower values of k for lower coding complexity. Note that $k = 1$ in fact corresponds to not using any coding at all. Now, once k is fixed, choose $n = \alpha k$.

III. TRACE BASED EXPERIMENTS

We now turn to an empirical evaluation of the benefits of coded storage, using a real vehicular trace. We use GPS traces of 1,000 (randomly chosen out of the available 2,927) taxis in Beijing collected from 00:00hrs to 23:59hrs on Jan 5, 2009 local time, recorded every minute. We assume that the nodes continue to run their application throughout the day. For inter-vehicular communication, we used a realistic model of IEEE 802.11p from [22], the details of which are given in section III-B below.

In order to characterize the performance of the system, we cannot simply use the average delay in downloading a file as a figure of merit. This is because, since the traces are time limited, there could be files that may not get fully reconstructed by the end of the duration of the trace, and so it is hard to quantify the delay of such incompletely downloaded files. Thus, we rely primarily on two metrics: one is the *full-recovery probability*, which measures the probability that a file can be fully recovered by a sink by a given time and the other is the *average file download percentage*, which measures, on average, how much of a file is downloaded by a given time. Thus, for example, a file-recovery probability of 0.9 means that the nodes were able to successfully download full files 90% of the time and an average file download percentage of, say, 95 means that the nodes were able to download 95% of the file on the average.

A. Simulation Setup

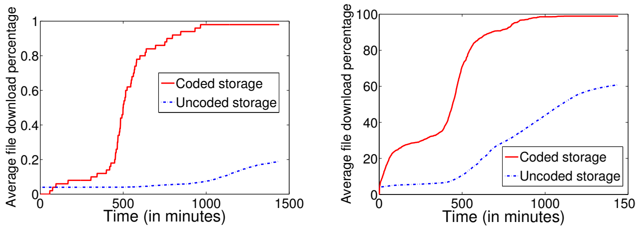
Both in the uncoded and the coded storage schemes, the chunks and files are stored by ensuring maximal spreading, so that, in the case of uncoded storage, a file is not stored in the same node twice and in the case of coded storage, multiple chunks of the same file are not stored in the same node, unless all other nodes have been used. In fact, we found that by randomly storing files/chunks, coding still performed virtually the same whereas the performance of uncoded storage scheme decreased slightly. Thus, we decided to use maximal spreading so as not to worry about the performance degradation introduced by randomization, even though random storage may be more realistic.

Next, the day-long trace is divided into intervals of length one minute each and thus at each time step, all we need to do is to determine the distance between the given sink and every other node, and apply the radio model (described below) to find out the number of packets transferred, if any.

Since the end goal is to deploy a file sharing system in a vehicular network, we try to make reasonable choices of various parameters involved. A capacity of 100GB per node is assumed as a default, unless specified otherwise. Similarly, by default, files are assumed to be of size 1GB, typical of movie clips and we consider a default of 2,500 files in the system, so that each file can be replicated $\alpha = 40$ times when there are 1,000 nodes.

B. Realistic Radio Link Model

The IEEE 802.11p standard specifies the data rate to range from 1.5Mbps to 27Mbps with the default being 3Mbps, which we use in our simulations. For inter-vehicular communication, we use an empirical model of packet delivery characteristics obtained from [22]. The authors characterize the packet delivery ratio (PDR) against various parameters such as the separation between two nodes, their relative velocity etc., in a number of different environments and the overall experiments lasted for about 30 hours. Of the various environments in which their experiments were conducted, the closest match to our dataset is the Suburban Road (SR) environment. Thus we use their PDR vs separation distance data (Fig 3(a) in [22]) to carry out our simulations. It may also be emphasized that the authors found that the relative velocity between two nodes does not significantly affect the PDR, the way inter-vehicular distance does. We choose packet sizes of 380 bytes with payload 300 bytes. Additionally a protocol set up time of about 1ms is considered.



(a) Full-recovery probability (b) Average file download percentage

Fig. 1. Evaluating the performance of distributed storage codes in the default setting consisting of 2,500 files each of size 1GB stored in nodes each having 100GB storage and there are 1,000 nodes in total.

C. Experimental Methodology

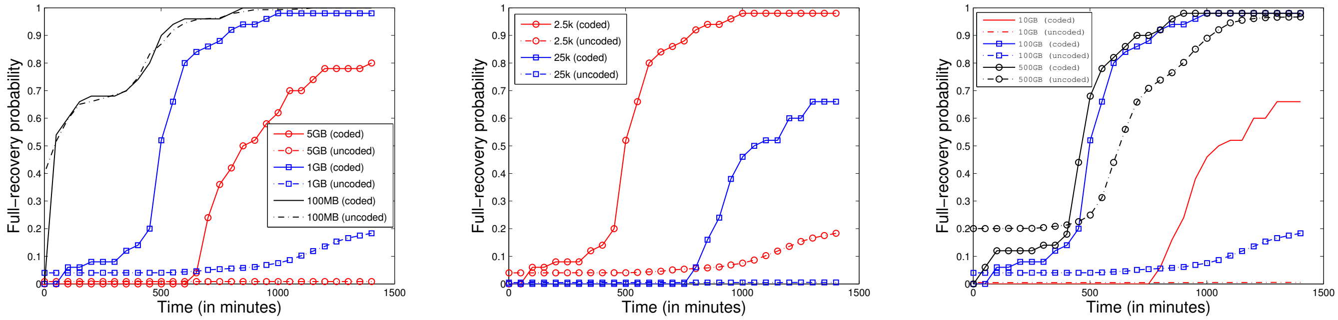
As explained before, the two primary metrics of performance are the full-recovery probability and the average file download percentage, both characterized as functions of time. Once the files or the chunks (depending on the scheme to evaluate) are stored in all the nodes, in order to simulate the file sharing application, a random node is selected to be the sink and it tries to collect a random file. For each sink-file pair, we keep track of the percentage of the file downloaded and whether the file download is complete or not at each time step. When presenting the results, we average over 50 random sinks, and for each sink, we run the entire simulation 100 times choosing a different file each time.

D. Choice of the coding parameter k

From the dataset, we observed an average contact duration of 55.6s (assuming a radio range of 500m) leading to an average data transfer of 21MB (at 3Mbps under ideal conditions). Since its desirable to be able to transfer multiple chunks per encounter, we choose a safe chunk size of 1MB.

E. Discussion of the Results

Our most important results are shown in Fig 1, in which we consider a typical file sharing scenario with 2,500 files each of size 1GB; and each node having about 100GB storage. Such a system is implemented atop the dataset, and both the full-recovery probability and the average file download percentage are measured for each time step. We note that coding offers significant benefits compared to uncoded replication. For example, at the end of 24 hours, files are reconstructed fully

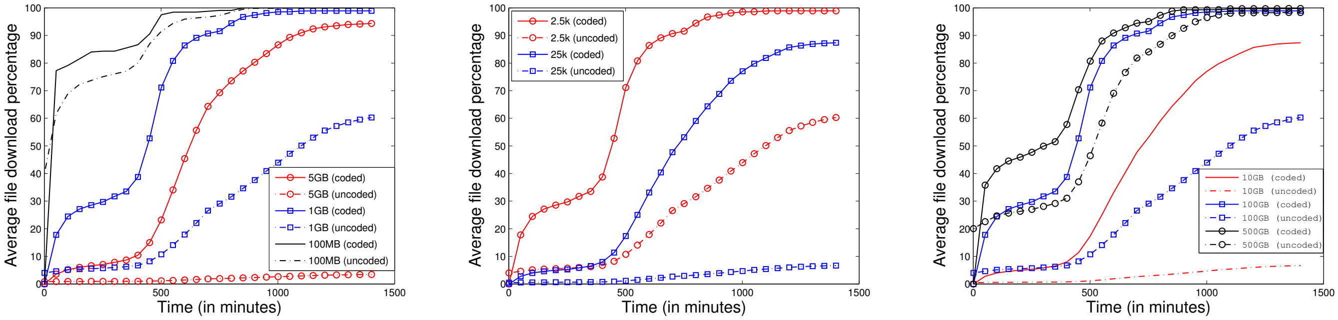


(a) File size varied

(b) Number of files varied

(c) Capacity per node varied

Fig. 2. Plots showing how various parameters affect the full-recovery probability. In each of the cases, one parameter is varied while keeping the others constant. Typical values used are a storage capacity of 100GB, 2,500 files and file size 1GB.



(a) File size varied

(b) Number of files varied

(c) Capacity per node varied

Fig. 3. Plots showing the impact of different parameters on the average file download percentage. The parameters are same as in Fig 2.

98% of the time by using coding, whereas without coding, only 19% of the files are reconstructed fully (see Fig 1(a)). The corresponding values for average file download percentage are 99% and 61% respectively. If we were to consider the instant when 80% of nodes are able to complete their downloads, this corresponds to about 600 minutes in the trace when coding is used, but only 4.4% of nodes are successful in full downloads by 600 minutes if coding is not used. An interesting observation to make is that since the Beijing trace begins in the middle of the night with relatively little traffic, one can see from Fig 1 that the rate at which files are completed starts to slow down around 60 minutes (1 a.m.) and then picks up again at 400 minute (7 a.m.). Another factor affecting the rate towards the end is the scarcity of new chunks (similar to the coupon collector problem).

Further, we performed a number of experiments to thoroughly understand the effect of various parameters on the performance of the system, by systematically varying the parameters $\mathcal{M}$, $\mathcal{C}$ and m. In our evaluations, we keep two parameters constant and vary the third.

1) Effect of file size: As file size increases, since system storage remains constant, we are effectively decreasing the system redundancy, which should adversely impact latency. This is observed for both coded and uncoded storage, but there are clear differences in relative performance. We notice from Fig 2(a) and Fig 3(a) that when the file size is very small (100MB in the figures), coding offers no benefit at all. But as the file size is increased to 1GB, coding offers tremendous improvements by being able to fully download full files most of the time (98% of the time in Fig 2(a)), whereas only about a fifth of the time (Fig 2(a)) without coding. When the file size is increased further to 5GB, the performance of coding suffers, but not drastically, whereas in the absence of coding, the probability of full recovery drops almost to zero (from Fig 3(a), and we note that many sinks have been able to download about a tenth of the file on the average, but not a complete file).

2) Effect of the number of files and the capacity: Figs 2(b) and 3(b) show the impact of the number of files on the system performance. As the number of files increases, the system redundancy decreases and hence the full-recovery probabilities and the file download percentages both start to decrease. And, as the capacity increases from 10GB to 100GB to 500GB, files can be replicated many more times and hence the full-recovery probabilities and the file download percentages both start to get better (Fig 2(c) and Fig 3(c)). An interesting observation to make is that the curve corresponding to the case when there are 25,000 files with 100GB storage per car in Fig 2(b) and the curve corresponding to 2,500 files with 10GB storage per car in Fig 2(c) (or Fig 3(b) and Fig 3(c)) are both identical (if we choose the same set of sink file pairs). This is because having 25,000 files on nodes with 100GB has the same system redundancy as having 2,500 in 10GB nodes. Also note that some of the probabilities or percentages for the uncoded replication start non-zero, since some of the sinks already contain the files they are interested in, whereas when coding is used, no node can contain a full file by itself and so all the probabilities and percentages are 0 to begin with.

F. Absolute File Download Latency

A cautionary note is in order in interpreting our results in this section in terms of the absolute numbers, which suggest that downloading a large 1GB-sized file in a vehicular network is likely to take six to ten hours even with coding. We note that our trace, though it involves 1,000 nodes, is still relatively quite sparse in terms of encounters as it involves a large area in Beijing. Further, it is important to note that the simulations start around midnight, which also skews the latencies observed, as there is not much encounter activity till many hours later. Thus the latency values presented in our study in terms of absolute numbers may not be representative of what might be possible with much denser vehicular network deployments (say 100,000+ vehicles in a large city) during high-traffic hours. But the dramatic gaps observed between the performance of coded and uncoded storage in these simulations indicate strongly that the use of coding is essential for speeding up large file downloads in encounter-based vehicular networks, regardless of vehicular density.

IV. CONCLUSION

We have demonstrated the benefits of coded storage on the latency of on-demand, pull-based content access in an intermittently connected vehicular network via realistic simulations based on a large-scale vehicular trace involving taxis in Beijing. There are still many unanswered questions. Open questions include how to re-distribute content when there is node churn, and the possibility of learning patterns in vehicular encounters to further optimize the content storage. Another interesting problem is to determine the optimal storage strategy if the popularities of various files are known beforehand. We are currently working on developing an analytical model for this problem, which we hope will give us further insigts into vehicular content dissemination problem. We also note that our traces, albeit involving 1,000 cars, are still relatively sparse given that they involve city-scale mobility. It is important to investigate content access latency in larger deployments, to understand the performance of file sharing in large-scale vehicular networks.

REFERENCES

[1] D. Jiang and L. Delgrossi, "IEEE 802.11 p: Towards an international standard for wireless access in vehicular environments," in *IEEE VTC Spring 2008.*

[2] L. Armstrong, "Classes of Applications," *Presentation, http://tinyurl.com/vanetapps*.

[3] J. Eriksson, L. Girod, B. Hull, R. Newton, S. Madden, and H. Balakrishnan, "The pothole patrol: Using a mobile sensor network for road surface monitoring," in *Proc. of ACM MobiSys*, 2008.

[4] A. Nandan, S. Tewari, S. Das, M. Gerla, and L. Kleinrock, "AdTorrent: Delivering location cognizant advertisements to car networks," in *Proc. IEEE/IFIP WONS*, 2006.

[5] K. Lee, S.-H. Lee, R. Cheung, U. Lee, and M. Gerla, "First experience with cartorrent in a real vehicular ad hoc network testbed," in *IEEE MOVE*, May 2007.

[6] U. Lee, J. Lee, J. Park, and M. Gerla, "Fleanet: A virtual market place on vehicular networks," *Vehicular Technology, IEEE Transactions on*, vol. 59, no. 1, pp. 344–355, 2010.

[7] S. Kapadia, B. Krishnamachari, and S. Ghandeharizadeh, "Static Replication Strategies for Content Availability in Vehicular Ad-hoc Networks," *Mobile Networks and Applications*, vol. 14, no. 5, pp. 590–610, 2009.

[8] "Excited About the Cloud? Get Ready for Capped Data Plans," *http://pogue.blogs.nytimes.com/2011/06/16/excited-about-the-cloud-get-ready-for-capped-data-plans/*.

[9] A. Vahdat and D. Becker, "Epidemic routing for partially connected ad hoc networks," CS-2000-06, Duke University, Tech. Rep., 2000.

[10] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and wait: an efficient routing scheme for intermittently connected mobile networks," in *WDTN*, 2005.

[11] S. Jain, M. Demmer, R. Patra, and K. Fall, "Using redundancy to cope with failures in a delay tolerant network," *ACM SIGCOMM Computer Communication Review*, vol. 35, no. 4, pp. 109–120, 2005.

[12] Y. Wang, S. Jain, M. Martonosi, and K. Fall, "Erasure-coding based routing for opportunistic networks," in *Proc. of ACM WDTN*, 2005.

[13] U. Lee, J. Park, J. Yeh, G. Pau, and M. Gerla, "CodeTorrent: Content Distribution using Network Coding in VANET," in *Proc. of ACM MobiShare*, 2006.

[14] U. Shevade, Y.-C. Chen, L. Qiu, Y. Zhang, V. Chandar, M. K. Han, H. H. Song, and Y. Seung, "Enabling high-bandwidth vehicular content distribution," in *Proc. of ACM CoNEXT*, 2010.

[15] T. Richardson and R. Urbanke, *Modern Coding Theory*. Cambridge University Press, 2008.

[16] J. Byers, M. Luby, M. Mitzenmacher, and A. Rege, "A digital fountain approach to reliable distribution of bulk data," in *Proc. of ACM SIGCOMM*, 1998.

[17] M. Luby, "LT Codes," *Proc. of IEEE Foundations of Computer Science*, 2002.

[18] A. Shokrollahi, "Raptor codes," *IEEE Trans. on Information Theory*, June 2006.

[19] H. Weatherspoon and J. D. Kubiatowicz, "Erasure coding vs. replication: a quantitiative comparison," in *Proc. of IPTPS*, 2002.

[20] S. Ahmed and S. Kanhere, "VANETCODE: Network Coding to enhance cooperative downloading in Vehicular Ad-hoc Networks," in *Proc. of ACM ICWCMC*, 2006.

[21] M. Li, Z. Yang, and W. Lou, "CodeOn: Cooperative popular content distribution for vehicular networks using symbol level network coding," *Selected Areas in Communications, IEEE Journal on*, vol. 29, no. 1, pp. 223–235, 2011.

[22] F. Bai, D. Stancil, and H. Krishnan, "Toward understanding characteristics of dedicated short range communications (DSRC) from a perspective of vehicular network engineers," in *Proc. of ACM MobiCom*, 2010.