CodeOn: Cooperative Popular Content Distribution for Vehicular Networks using Symbol Level Network Coding

Ming Li, *Student Member, IEEE,* Zhenyu Yang, *Student Member, IEEE,* and Wenjing Lou, *Senior member, IEEE*

***Abstract*—Driven by both safety concerns and commercial interests, one of the key services offered by vehicular networks is popular content distribution (PCD). The fundamental challenges to achieve high speed content downloading come from the highly dynamic topology of vehicular ad hoc network (VANET) and the lossy nature of the vehicular wireless communications. In this paper, we introduce CodeOn, a novel push-based PCD scheme where contents are actively broadcasted to vehicles from road side access points and further distributed among vehicles using a cooperative VANET. In CodeOn, we employ a recent technique, *symbol level network coding* (SLNC) to combat the lossy wireless transmissions. Through exploiting symbol level diversity, SLNC is robust to transmission errors and encourages more aggressive concurrent transmissions. In order to fully enjoy the benefits of SLNC, we propose a suite of techniques to maximize the downloading rate, including a prioritized and localized relay selection mechanism where the selection criteria is based on the usefulness of vehicles' possessed contents, and a lightweight medium access protocol that naturally exploits the abundant concurrent transmission opportunities. We also propose additional mechanisms to reduce the protocol overhead without sacrificing the performance. Extensive simulation results show that, under a wide range of scenarios, CodeOn significantly outperforms a representative state-of-the-art scheme.**

I. INTRODUCTION

Vehicular communications have attracted lots of attentions recently. Since the advent of dedicated short range communications (DSRC) [1], [2], and IEEE 802.11p and IEEE 1609 standards [3], people have envisioned and designed numerous tempting applications of vehicular networks, ranging from safety warning [4], intelligent navigation to mobile infotainment [5]. A particularly promising type of application is related to both safety-related and commercial services. That is, the distribution of "popular" multimedia contents to vehicles inside a geographical *area of interest* (AoI) by road side infrastructure (e.g. access points (APs)), which is referred to as *popular content distribution* (PCD) in this paper. Examples of PCD may include: an ads company periodically broadcasts multimedia advertisements of local businesses in a city to vehicles driving through a segment of suburban highway passing by that city (like a digital billboard); a traffic authority delivers real-time traffic and accident information about the roads in an urban area for intelligent navigation or emergency warning purposes, or disseminates an accurate update of the GPS map about a city or a scenic area.

Different from the usual "content downloading" services where various vehicles are interested in downloading different files from the Internet [6], [7], the popular contents in PCD are often commonly "interested" by most of the vehicles driving through an AoI, and sometimes may even be disseminated mandatorily such as emergency videos [8]. An important aspect in common about popular contents is their potentially large file sizes, because multimedia files including video and audio are more vivid and effective, thus are always preferred over text-only files. For example, an advertisement video may be as large as 100 MB. Indeed, disseminating such large contents is possible in vehicular networks, given that four sub-channels in DSRC are allocated as service channels, while the IEEE 802.11p supports data rates up to 27 Mbps.

The primary requirement of PCD in vehicular networks is to achieve short *downloading delay*, or equivalently, high *downloading rate*. The former is the average time required for end-vehicles to receive a file completely. From a driver's point of view, fast reception of a video about an accident or traffic condition may help the driver to plan his/her route in advance to avoid possible traffic jams or accidents. From the content provider's viewpoint, shorter downloading delay improves the ratio of vehicles that can receive the content. Thus, a short delay is essential for both commercial and non-commercial contents. In addition, it is also critical for PCD to maintain a high degree of efficiency, i.e., to introduce low protocol overhead and reasonable amount of data traffic, so that PCD is readily compatible with other potential services running under the same channel.

Due to the relatively high cost of deploying APs, the access to wireless Internet is quite limited in vehicular networks. In the initial deployment phase APs may be rare, which could be placed in highway service areas, gas stations or road intersections. Since it takes usually less than 1 minute for moving vehicles to pass by an AP, vehicles may not finish downloading a large file within such a short time period. When the vehicles are out of the coverage of the APs, the vehicles form a vehicular ad hoc network (VANET) and cooperative distribution of the popular content is thus necessary.

However, it is non-trivial to design a high-rate and efficient cooperative PCD scheme. The main challenges come from the lossy wireless medium under vehicular environments, and the highly mobile and dynamical nature of VANETs. First, the lossy wireless links cause frequent packet losses and collisions, leading to prolonged downloading delay and decreased efficiency, and negatively affects the protocol per-

The authors are with the Department of Electrical and Computer Engineering, Worcester Polytechnic Institute. E-mail: {mingli,zyyang,wjlou}@ece.wpi.edu

formance. In addition, the ever-changing VANET topology prevents real-time acquisition of precise neighbor information (such as reception status) which forms the basis of optimized, distributed transmission decision making. If there lacks a well-devised coordination mechanism among the transmitting vehicles, duplicate transmissions may fill up the channel and waste the precious VANET bandwidth. Also, a PCD scheme could potentially incur large protocol overhead spent in collecting those information needed to achieve high performance.

Towards solving these problems, many existing works [5], [8]–[11] have adopted *network coding* (NC) [12] for content downloading in VANETs, because NC effectively reduces duplicate transmissions and simplifies the transmission scheduling. Most of these protocols employ a pull-based cooperative content downloading approach [5], [10], where vehicles transmit passively upon others' downloading requests, which suffers from low efficiency. When downloading popular files, many vehicles make requests for the same content and many vehicles respond to their requests. Due to the lack of coordination, these protocols cannot avoid severe packet losses and collisions, especially under a dense VANET. This could lead to extremely low efficiency and large downloading delay. Thus the performance gain obtained from network coding is under-exploited and even offset by unrefined protocol design.

In this paper, we put forward CodeOn, a high-rate cooperative PCD scheme for vehicular networks. We explore *symbol level network coding* (SLNC) [13] for cooperative PCD. In contrast with traditional packet level network coding, SLNC performs network coding on finer granularity of physical layer symbols. Since the error rate of a symbol is smaller than that of a packet's, SLNC has better error tolerance, enhances reception reliability and thus the downloading rate. Fully exploiting the advantage of SLNC for PCD necessitates non-trivial protocol design, whereas we make the following main contributions.

(1) CodeOn provides a whole new set of push-based content distribution protocol design for VANETs. The popular contents are actively broadcasted from a few APs to all vehicles within an AoI, through the cooperation of a set of dynamically selected relay nodes. In order to maximize the usefulness of every piece of content broadcasted by those relays, we propose a prioritized relay selection mechanism to coordinate the transmissions of vehicles, in which every vehicle's transmission priority is proportional to how much additional useful content it can provide to its neighbors.

(2) In order to fully take advantage of the increased transmission concurrency enabled by SLNC, we present a lightweight medium access control (MAC) protocol where a candidate relay node cancels its broadcast whenever it senses a busy channel. Surprisingly, we find that this design, although simply based on carrier sense, can actually result in overall downloading rate that is close to maximum. This result stems from the fact that, the impact of hidden terminals can be greatly reduced due to SLNC's better error tolerance, which is not the same case for the traditional network coding methods.

(3) To reduce the protocol overhead without degrading the performance, we propose a scalable and efficient average-rank method for vehicles to represent and exchange their content reception status under SLNC. By taking advantage of the multi-channel property of VANET, vehicles piggyback this tiny information in their safety messages sent in control channel, which incurs *zero* overhead for content downloading.

(4) We implement CodeOn in NS-2 and evaluate its performance by extensive simulations. We compare CodeOn with an enhanced version of CodeTorrent, which is a pull-based, network coding based content distribution protocol and represents the current state-of-the-art. Simulation results show that CodeOn performs significantly better than CodeTorrent, in terms of average downloading delay, protocol efficiency and fairness. Significant improvements in average downloading rate are obtained for both highway and urban scenarios. To the best of our knowledge, this is the first time that cooperative PCD has been studied under lossy VANET environments.

The rest of the paper is organized as follows. Sec. II formulates the PCD problem in vehicular networks and discusses related works, Sec. III introduces symbol level network coding and its benefits for content downloading. The main design of CodeOn is presented in Sec. IV. Sec. V contains the performance evaluation and results. Finally, Sec. VI concludes the paper.

II. PROBLEM FORMULATION AND RELATED WORK

A. Problem Formulation

1) Model and assumptions: In this paper, we consider the following PCD service architecture for vehicular networks. The content provider (e.g. a city wide traffic administration bureau) wants to distribute some popular files to all vehicles inside an *area of interest* (AoI), which can be either a highway segment or an urban area. There are multiple APs (or road side units) deployed in an AoI, and APs are connected together through a wired backhaul. APs are controlled by the service provider to actively disseminate popular contents to the vehicles within the AoI. APs can be placed either deterministically or randomly and optimal placement is outside the scope of this paper. The service architecture is illustrated in Fig. 1.

Each vehicle is equipped with an on board unit including a wireless transceiver (single radio). The wireless interface operates on multiple channels [1], [2]. To model the coexistence of safety and commercial applications, we consider two representative channels. The control channel is used to broadcast safety messages, which may contain vehicles' locations, speeds etc.; one service channel is dedicated for PCD. In order to guarantee the quality of service of safety messages (the interval between two consecutive safety messages should be smaller than 100ms [14]), time is divided into periodical, 100ms slots and all vehicles and APs are synchronized to switch simultaneously between the control channel and service channel. The utilization of time and channels is depicted in Fig. 2. Although there are advanced MAC protocols that dynamically adjust the time shares of control channel and service channel for better service [14], we fix it to $1/2 : 1/2$ for simplicity.

In the control channel, each AP and each vehicle broadcasts one beacon message in each slot. When a vehicle is in the range of an AP, it merely listens to the AP's content

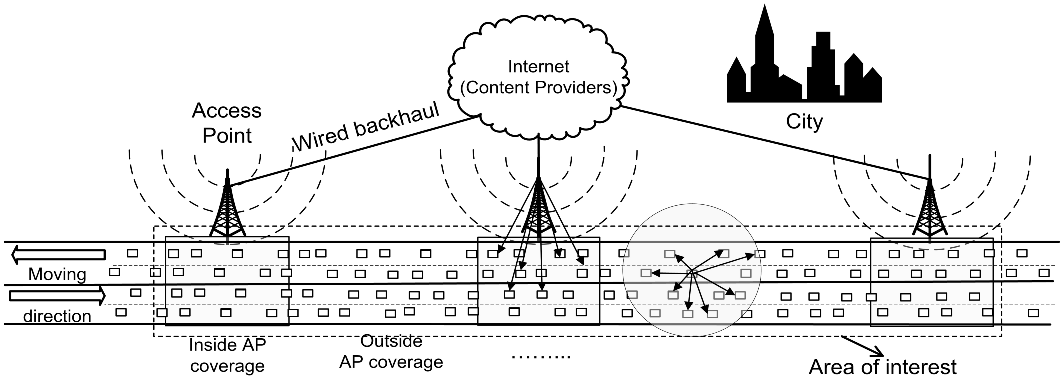


Fig. 1: The architecture for PCD. Inside the AP coverage, AP broadcasts and vehicles receive; outside the AP coverage, vehicles distribute their received contents cooperatively.

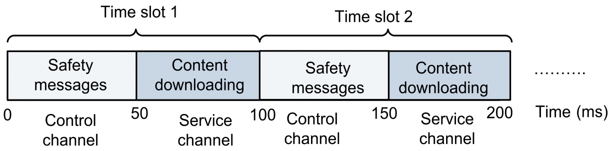


Fig. 2: The time and channel utilization of each vehicle and each AP.

broadcast in the service channel; otherwise, it may share its received content with neighboring vehicles cooperatively. Vehicles outside the AoI do not involve in content distribution.

In addition, we assume all vehicles are equipped with Global Positioning System (GPS) devices, from which vehicles obtain their real-time locations and synchronize their clocks (error smaller than 100ns). GPS devices are low-cost and are available to most of the drivers nowadays. When vehicles are temporarily out of satellite coverage, they can use auxiliary techniques to determine their location, and rely on their own hardware clocks. Note that, GPS time synchronization is required by the IEEE 1609.4 standard for multi-channel operations [3].

2) Objectives: For any content distributed by the PCD service, the primary objective is to achieve low average downloading delay, which is equivalent to high average downloading rate. For each vehicle in an AoI, its *downloading delay* is defined as the elapsed time from downloading start to 100% completion. Meanwhile, it is desirable to achieve a high degree of fairness, i.e., the variation of downloading delays among different vehicles should be small. Finally, high-rate content distribution cannot come at the cost of incurring too much protocol overhead and data traffic, otherwise the PCD service would be less compatible with other possible services in the service channel. Thus it is also important to maintain high protocol efficiency.

B. Related work and our contributions

In [6], Nandan *et.al.* first studied cooperative downloading in VANETs. They proposed SPAWN, a pull-based, peer-to-peer content downloading protocol for VANETs that extends BitTorrent. Later, they proposed "AdTorrent" [15], which is a semi push-based peer-to-peer protocol for vehicles to download advertisements they are interested in. In both SPAWN and AdTorrent, the peer and content selection mechanisms have high overhead and are not scalable, especially when most of the vehicles are interested in downloading popular contents. Also, they suffer from the "coupon collector problem" which enlarges downloading delay. Moreover, they use TCP for content delivery, which performs poorly over multi-hop lossy wireless links in highly mobile VANETs.

1) Network coding for content downloading: To avoid such problems, many researchers resort to network coding (NC) [12], [16]. The NC mixes the packets by coding them together at every intermediate node and exploits the broadcast nature of wireless medium, so that the usefulness of each coded packet is increased. Lee *et.al.* proposed CodeTorrent [5], a pull-based content distribution scheme using NC, where vehicles need to explicitly initiate requests to download a piece of content. CodeTorrent restricts the peer selection and content delivery to the one-hop neighborhood of a vehicle, thus eliminating the need of multi-hop routing. Also, the use of NC mitigates the peer and content selection problems.

Later, Lee *et.al.* further studied the practical effects of content distribution in VANETs using NC [10] based on a variation of CodeTorrent. It is shown that the resource constraints such as disk access, computation and buffer have significant impacts on the performance. They discussed approaches to reduce the communication and computation overhead of NC while maintaining the gain of it. Since our paper focuses on dealing with the lossy wireless links in content downloading for VANETs, our work is orthogonal to [10].

The above schemes are all pull-based in essence. They could suffer from large downloading delay, since nodes passively respond to their neighbors' requests and the bandwidth is wasted (i.e., being idle much of the time). For example, in CodeTorrent it takes 200 seconds to download a 1 MB file in an urban scenario [5]. If a node wants to receive new information continuously, it must send out requests frequently. The transmissions from multiple responders tend to collide with each other, leading to low-efficiency in turn. Park *et. al.* proposed a push-based content delivery scheme for emergency related video streaming using NC [8]. However their "push" protocol design essentially reduces to controlled flooding, which tends to be inefficient.

In fact, with packet level network coding (PLNC), it is difficult to achieve high downloading performance especially under lossy wireless links in VANETs, whether or not push based protocol design is adopted. The wireless medium in VANET has been shown to be lossy by empirical analysis [17]–[19]. In practice, network coding for a large file is usually done within each block of the file, namely a *generation* [5], [9], [10]. In order to maintain reasonable coding/decoding complexity while reducing the protocol overhead, the basic coding unit (coded piece) shall be larger than a usual packet. During the transmission of such a coded piece, any error to the coding vector or message body will render the whole piece useless, leading to degraded downloading performance.

In this paper, we put forward CodeOn, a whole new set of push-based protocol design that can well solve those problems. Instead of using PLNC, we take advantage of symbol level network coding (SLNC) [13] which has much better resiliency to transmission errors due to symbol-level diversity.

2) Transmission coordination in content downloading: Transmission coordination is an important issue for content distribution in VANETs. Bad coordination could result in severe packet collisions that affects the downloading performance. However, this issue has not been well addressed in previous works. In [8], a simple time out mechanism is used for each vehicle to decide when to transmit a coded packet. However, this mechanism does not take into account vehicles' content reception status, which leads to a non-negligible chance of duplicate information. Also, packet collisions are severe when the network is dense.

In [20], Zhang *et. al.* studied this problem from link layer, and proposed VC-MAC, a cooperative medium access control (MAC) protocol for gateway downloading scenarios in vehicular networks. In order to avoid possible interference among multiple transmissions, and to maximize the "broadcast throughput", a heuristic relay selection algorithm with a backoff mechanism is proposed. However, the "broadcast throughput" is purely based on link quality, which is not content-aware. The relay chosen by VC-MAC may have nothing innovative to transmit to its neighbors.

In CodeOn in this paper, we explicitly consider the content usefulness of nodes for higher rate content downloading. A dynamic set of relay nodes which are selected based on their content availability and usefulness, actively broadcast (push) useful contents to neighboring nodes, and make medium access decisions based on both their content usefulness and local channel status.

3) Multi-channel compatibility: Few existing work considered the compatibility of content downloading with other channels. In [14], the authors propose mechanisms to adjust the time share of the service channel to enhance the performance of content downloading while guaranteeing the QoS of safety messages. Our paper considers the coexistence of a service channel with the control channel, with the difference that we design a better PCD protocol given a fixed time share of service channel. Also we novelly utilize the control channel for better content downloading.

4) Other related works: In [21], Zhao *et. al.* proposed data pouring, a push-based data dissemination protocol for VANETs. They focus on broadcasting small data items to all vehicles inside an area, while we aim at disseminating large popular files. In [22], Zhao *et. al.* also studied the problem of drive-thru access to roadside APs, and proposed a vehicle-to-vehicle relay strategy to extend the coverage of APs. In [23], Yang *et. al.* proposed a push-based, reliable broadcast protocol for wireless mesh networks using network coding.

In addition, Fiore *et. al.* focused on cooperative downloading in urban VANETs [7]. The Roadcast [24] is a popularity-aware content sharing protocol in VANETs. These protocols are mainly suitable for applications where each vehicle may be interested in downloading different files, while we consider the popular content distribution.

III. SYMBOL-LEVEL NETWORK CODING

In this section, we first describe the symbol-level network coding technique. Then, we give a motivating example to show the potential advantage of exploiting symbol-level diversity in content distribution in VANETs.

A. A Brief Review of Symbol-level Network Coding

SLNC was recently introduced by Katti *et. al.* [13] to improve the unicast throughput in wireless mesh networks. SLNC arises from the observation that in wireless networks, even if a packet is received erroneously, some small groups of bits ("symbols") within that packet are likely to be received correctly. SLNC gathers these correctly received (i.e., "*clean*") symbols aggressively, and performs network coding on the granularity of symbols. In contrast to PLNC, SLNC gains from both symbol-level diversity and network coding. In addition, since more bit errors are tolerated than PLNC, SLNC can also gain higher throughput by encouraging more aggressive concurrent transmissions.

In general, SLNC works as follows. A symbol is defined as a group of consecutive bits in a packet, which may correspond to multiple PHY symbols of a modulation scheme. Assume the source has K packets to send, each of them expressed as a vector with elements from a Galois field $\mathbb{F}_{2^q}$. The jth symbol $\mathbb{s}'_j$ in a coded packet at the source is a random linear combination of the jth symbol in all K source packets:

$$\mathbb{s}'_j = \sum_{i=1}^{K} v_i \mathbb{s}_{ji}. \tag{1}$$

where $\mathbb{s}_{ji}$ is the jth symbol (at jth position) in the ith original packet, coefficient v_i is randomly chosen from $\mathbb{F}_{2^q}$, and $\mathbb{v} = (v_1, ..., v_K)$ is the coding vector of the coded packet, which is also the coding vector for each symbol. Each receiver node v maintains a decoding matrix for every symbol position. A newly received coded symbol for position j is called *innovative* to v, if that symbol increases the *rank* of the decoding matrix of the jth symbol position, referred to as *symbol rank*. Only innovative clean symbols are buffered.

Each coded packet transmitted by a relay node consists of random linear combinations of buffered clean symbols. For a source, every symbol in a packet is clean and shares the same coding vector. However, at a relay node, coding vectors may

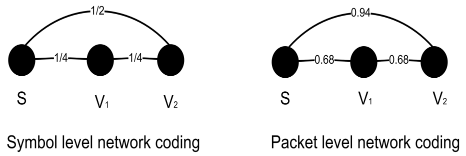


Fig. 3: The topology for the example in Fig. 4. Left: numbers on the edges (links) show the symbol error probabilities; right: corresponding packet error probabilities.

be different across symbols. For a coded packet to be sent by relay u, the jth coded symbol is expressed as

$$\mathbb{s}_j'' = \sum_{i=1}^{R} v_i' \mathbb{s}_{ji}' = \sum_{i=1}^{R} (v_i' \sum_{l=1}^{K} v_{li} \mathbb{s}_{jl}) = \sum_{l=1}^{K} (\sum_{i=1}^{R} v_i' v_{li}) \mathbb{s}_{jl}, \quad (2)$$

where R is the number of buffered clean symbols at position j, $\mathbb{s}_{ji}'$ is the ith buffered clean symbol (row) at position j (column), and $\mathbb{v}_i = \{v_{1i}, ..., v_{Ki}\}$is the coding vector for that symbol. $\mathbb{s}_{jl}$ is the jth symbol of the lth source packet. From Eq. (2), $\mathbb{s}_j''$ is still a random linear combination of source symbols, and its new coding vectors are $\mathbb{v}' = (\sum_{i=1}^{R} v_i' v_{1i}, ..., \sum_{i=1}^{R} v_i' v_{Ki})$.

In the extreme case, every symbol's coding vector is different and needs to be sent along with a packet, which incurs high overhead. To minimize this overhead, optimized run-length coding method can be adopted [13], where consecutive clean symbols are combined into a "run".

B. How VANET content distribution benefits from SLNC

To illustrate how SLNC works and see the potential performance gain of SLNC over PLNC, we give a 3-node simple example for content distribution in VANET (Fig. 3 and Fig. 4). The corresponding topology is shown in Fig. 3. Assume source S has two original packets X and Y to broadcast. Assume a simple scheduling: S broadcasts coded packets until V_1 can decode the original packets, and then V_1 broadcasts until V_2 decodes all original packets.

Suppose S generates and broadcasts three coded packets A, B and C, each of them divided into 4 symbols. Let the symbol error probability from S to V_1 be $P_{se}(S, V_1) = \frac{1}{4}$, and it happens that each packet received by V_1 contains an erroneous symbol (Fig. 4). Luckily, for each symbol position at least two clean symbols are received. Since any two coding vectors among $\mathbb{v}, \mathbb{v}', \mathbb{v}''$ of A, B and C are independent[1], V_1 can decode X and Y by solving 4 linear equations. When V_1 broadcasts two packets (say, D and E), it generates two new coded symbols at each position, and packs the 8 coded symbols into D and E. Each new coded symbol is also a random linear combination of original symbols. Thus, V_2 can recover all original symbols after collecting 2 innovative coded symbols at each position, which may come from both S and V_1.

Now we compute the expected downloading delay of node V_1. Without loss of generality, we assume S has K source packets to broadcast, and each packet is divided into M symbols. We are interested in when V_1 is able to decode all

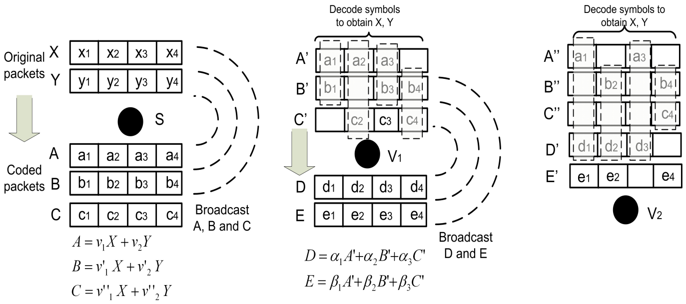


Fig. 4: Symbol level network coding in VANET content distribution. S: source node; V_1 and V_2: downloading vehicles & relays.

the source symbols from S (receive at least M correct and innovative symbols in all the positions). Assume all symbols in the same position are independently received according to an i.i.d. bernoulli process[2], where the probability of receiving a symbol correctly in one trial is $1 - P_{se}$, $(P_{se} = P_{se}(S, V_1))$. Let Z_i denote the number of packets sent (trials) for V_1 to receive exactly K correct symbols in the ith position. Then,

$$P(Z_i = k) = \begin{cases} \binom{k-1}{K-1} P_{se}^{k-K} (1 - P_{se})^K & \text{if } k \geq K \\ 0 & 0 \leq k < K, \end{cases} \quad (3)$$

and we have $P(Z_i \leq k) = \sum_{m=1}^{k} P(Z_i = m)$. Define r.v. Z as the smallest number of packets sent for V_1 to receive at least K correct symbols at all positions, then

$$Z = \max\{Z_i\}, i = 1, ..., M.$$

We have

$$P(Z \leq k) = [P(Z_i \leq k)]^M. \quad (4)$$

Therefore, the expected number of packets transmitted by S for V_i to decode is:

$$\mathbb{E}[Z] = \sum_{k=0}^{\infty} P(Z > k) = \sum_{k=0}^{\infty} [1 - P(Z \leq k)]. \quad (5)$$

Compute the above numerically by plugging in $K = 2, M = 4, P_{se} = 1/4$, we obtain $\mathbb{E}[Z] = 3.67$. That is, 3.67 coded packets should be sent by S on average for V_1 to decode X and Y. Thus, V_1's downloading delay is proportional to $\mathbb{E}[Z]$.

Next we compare SLNC to using PLNC for the same case. We compute the expected number of packets $\mathbb{E}[Z']$ sent by S for V_1 to receive K source packets. Since PLNC discards a packet with any erroneous symbol in it, the error probability from the packet level could be much larger than that of symbol level. For simplicity, we assume independent symbol error in one packet[3], so

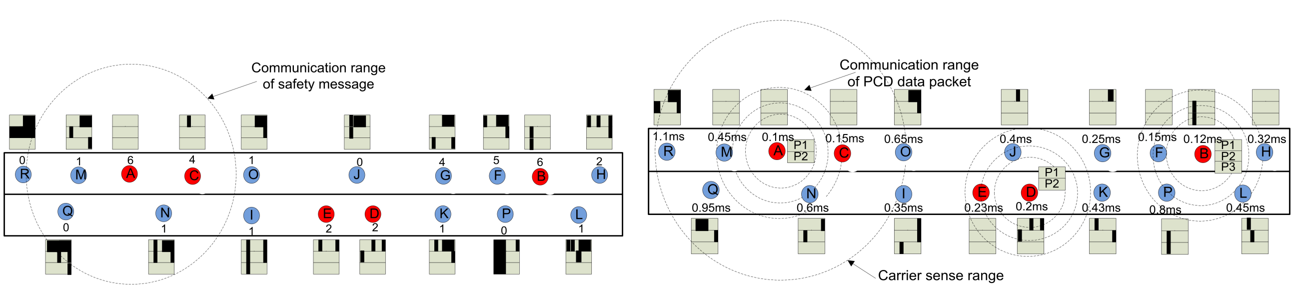
$$P_{pe} = 1 - (1 - P_{se})^M. \quad (6)$$

---

[1] This happens with high probability when the size of $\mathbb{F}_{2^q}$ is large.

[2] This assumption is valid in VANETs. The channel coherence time: $T_c \approx \frac{0.42}{\Delta f}$, where $\Delta f = \frac{v f_0}{c}$ is the doppler spread. With average relative speed $v = 30$m/s, central frequency f_0=5.9GHz, $T_c = 0.72$ms. Using the data rate 12Mbps in IEEE 802.11p, the time to send a 1KB packet is T_{tx}=0.68ms. Since $T_c \approx T_{tx}$, consecutive received packets can be regarded as independent, so are the symbols in the same positions.

[3] Albeit there exist error correction coding (ECC) techniques to enhance the error-resiliency of packet transmission, they do not change the nature of the following derivation since they are still limited in error-correcting capabilities. On the other hand, ECC can also be added to SLNC [13].



(a) Exchange of neighbor information and utility calculation based on nodes' reception status. All nodes' reception status are depicted. Black parts in a piece indicate corrupted symbols in a node's buffer.

(b) Transmission coordination among potential relays, based on both node priority and carrier sense. Backoff delays are inversely related to nodes' utilities (nodes A-F have the least delays, but only A, B, D become relays).

Fig. 5: Overview of cooperative content distribution in CodeOn.

The resulting error rates are shown in the right of Fig. 3 where $P_{pe}(S, V_1) = 0.68$. Assuming independent packet reception, $\mathbb{E}[Z'] = K/(1 - P_{pe})$. For the simple example, S must transmit $\frac{2}{1-P_{pe}(S,V_1)} = 6.26$ packets on average for V_1 to decode. Thus, the downloading delay of V_1 has been reduced by $\frac{6.26-3.67}{6.26} = 41\%$ due to the use of SLNC.

For node V_2, although it can overhear useful information from both S and V_1, as we can see from Fig. 3, $P_{pe}(S, V_2) = 0.94 \approx 1$, while $P_{se}(S, V_2) = 1/2$. In this case, the (S, V_2) link can almost be neglected for PLNC, and SLNC is expected to achieve higher gain for V_2 than V_1.

Note that, in reality the symbol errors may be correlated, which is related to the channel coherence time T_c. Then the actual difference between P_{pe} and P_{se} is smaller. Therefore, the gain we derived can be regarded as an upper bound. However, independent assumption still holds when the size of a "symbol" is on the order of a packet's, which is true for the piece division run-length SLNC used in CodeOn (see Sec. V-B), where the length of a run is usually on the order of a packet. In this case, the above model can exactly characterize the gain of piece-division run-length SLNC over traditional piece-division PLNC, where each generation is divided into pieces and a piece is either received or not received as a whole.

IV. THE DESIGN OF CODEON

We first give the main notations used in this paper in Table. I.

TABLE I: Frequently used notations

Notation	Definition
F	The file to be distributed
N	Data packet size (bytes)
L	File length (number of generations)
K	Generation size (number of pieces)
J	Piece size (bytes)
M	Number of symbols in a packet
G_i	Generation i
$\mathbb{F}_{2^q}$	The Galois field used in network coding
$U(v)$	The utility of a node v
$\mathcal{N}(u)$	The neighbor set of node u
$\bar{r}_{v,i}$	Average symbol rank of G_i in vehicle v
γ	Average received SNR or SINR for a symbol

A. Overview

CodeOn is a push-based cooperative content distribution protocol, where a large file F is actively distributed from the APs to the vehicles inside the AoI through the help of a dynamic set of relay nodes. Each AP is a source for F, and F is divided into equal-sized generations (chunks), and the SLNC is performed within each generation. In Fig. 5, we illustrate the general process of content distribution in CodeOn, assuming F has only one generation consisting of 3 pieces.

Each AP/source broadcasts the source file to vehicles in its range based on vehicles' reception status, which is not shown in Fig. 5. Outside the ranges of APs, vehicles distribute the file cooperatively by agreeing on a set of relay nodes. This is the core to CodeOn, which consists of three steps.

(1) Exchange of neighbor information. This is done in each control time slot, where every vehicle broadcasts a safety message that piggybacks a sketch of its content reception status, which will be used as an implicit content request for step (2). In this way, zero overhead is incurred in the service time slots. To limit the impact of piggyback overhead on control time slots, we will introduce a fuzzy representation of nodes' reception status later.

(2) Node utility calculation. This is the first step of distributed relay selection. In the beginning of each service time slot, every node computes its own utility based on neighbors' reception status information collected from step (1). The utility reflects each node's priority in relay selection, i.e., the total amount of useful content that this node can provide to all of its neighbors. Under such a priority assignment, the usefulness of each relay's transmission will be maximized, which enhances both the downloading rate and protocol efficiency. The utility of every node is shown in Fig. 5 (a).

(3) Transmission coordination among potential relays. As the last step of relay selection, we need to determine which nodes should actually access the channel, based on both node priority and the channel status. Each node computes a backoff delay that is inversely related to its utility, and upon the expiration of the delay it will sense the channel. If it cannot detect signal energy, it will broadcast coded contents without delay. Otherwise, it remains silent throughout the time slot. This process is captured by Fig. 5 (b). Thanks to SLNC's better error tolerance, this aggressive way of channel access, although simple, will be shown to achieve close to maximum

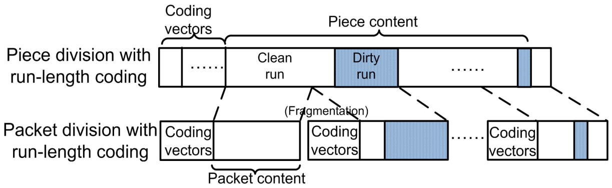


Fig. 6: Comparison between the overhead of piece division and packet division, when both uses run-length SLNC.

overall downloading rate in the following.

B. Network Coding Method

Symbol level network coding (SLNC) is used throughout the design of CodeOn. We describe the way that SLNC actually operates in CodeOn. Assume F with size $|F|$ is divided into L generations $G_1, G_2, ..., G_L$, where each generation contains K pieces. A piece has size J and contains $\lceil J/N \rceil$ packets. Then, $|F| = L \cdot K \cdot J$. In order to reduce the overhead brought by SLNC, we adopt "*piece-division, run-length SLNC*".

The reasons are two fold. On the one hand, if a generation is divided into packets (packet-division), in order to keep small computational overhead we must use relatively small K (the computation complexity of decoding is usually $O(K^3)$), thus a large number of generations is required for large F. This reduces the gain of NC due to the "coupon collector's problem" [10], and increases the communication overhead for exchanging the content availability. On the other hand, using multi-packet pieces (piece-division), K can be maintained at a reasonable value by scaling the piece length linearly with file size. However, the number of symbols in a piece ($\frac{J \cdot M}{N}$) increases with the piece length. In the extreme case if every symbol in a piece has a different coding vector, the communication overhead is at least $\frac{J \cdot M \cdot K \cdot q}{N}$ bits, which equals to 10KB if $J = 20$KB, $N = 1$KB, $K = 32, M = 32, q = 8$. This is clearly unacceptable. Fortunately, run-length coding method [13] can be used to reduce the communication overhead of SLNC, in which one coding vector is used for each sequence of consecutive clean symbols (*run*). Dynamic programming is used to choose appropriate combination of runs to minimize the overhead [13]. Therefore, in CodeOn, we combine run-length SLNC with the piece division to achieve higher network coding gain and reduce the communication overhead, which we call *piece-division run-length SLNC*. When a coded piece is transmitted, it is separated into several packets; only the header of the first packet contains the coding vectors of runs that composing the piece, while subsequent packets only have normal small headers. Thus, a piece can be regarded as a "big packet".

Compared with PLNC, the gain from symbol-level diversity can be easily seen from the analysis in Sec. III. Meanwhile, the overhead of our method is always smaller than run-length SLNC combined with packet division. Generally, the number of coding vectors in a piece equals to the number of runs. However, using packet division a run may be fragmented into more than one runs, which needs more coding vectors in total. In the worst case, each symbol is a run and the overheads are equal. This is illustrated in Fig. 6. In reality, since the symbols errors are often bursty (due to packet collisions), the number of runs is usually much smaller compared with the number of symbols. For example, if there are 20 runs in a 20KB piece the overhead is about 640B, which is 3.2% of piece size.

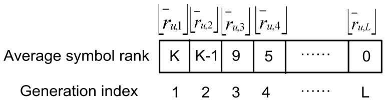


Fig. 7: The average rank representation of a file's reception status at node u.

In order to balance the gain and overhead of SLNC in CodeOn, we fix the number of pieces in a generation (K) and the number of generations (L) (e.g. 32 and 50, respectively). Although the piece size J scales linearly with the file size, since SLNC tolerates symbol errors, the size of a piece has small impact on the protocol performance.

C. Efficient Exchange of Content Reception Status

An important piece of information exchanged in CodeOn is every node's *content reception status* (i.e., how much content is downloaded for each generation), which is essential to enabling optimized, distributed transmission decisions. It could be obtained by sending gossip messages in each service time slot, but this consumes a large portion of a service time slot. In CodeOn, we choose to piggyback the reception status in safety messages, thus adding zero overhead in the service channel.

However, for SLNC, it will incur large overhead to represent the exact reception status of each generation. The decoding matrix can be represented by a single null-space vector [5]. However, the size of the reception status information adds up to $\frac{L \cdot J \cdot M \cdot K \cdot q}{N}$ bits, where Kq is the maximum size of one null-space vector. For $L = 50, J/N = 20, M = 32, K = 32, q = 8$, this amounts to 1MB which is too large.

Therefore, in CodeOn we propose a fuzzy *average rank* method to represent the reception status in an efficient way. An important property of network coding is that the rank of the decoding matrix determines the amount of received information. For two nodes u and v with symbol ranks $r_{u,i,j}$ and $r_{v,i,j}$ for position j in G_i, respectively, if $r_{u,i,j} > r_{v,i,j}$, then a recoded symbol $\mathbb{s}'_j$ sent from u is innovative to v with high probability [12]. Otherwise, this does not hold[4]. Therefore, we can substitute each null-space vector with a rank, which has $log_2 K$ bits. For a generation G_i received by node u, there are many symbol positions with different rank values. But since the size of a piece is relatively small (e.g., $J = 20$KB) compared to what can be transmitted in a 50ms slot using DSRC (55KB when data rate is 11MBps), the ranks of various symbol positions are expected to increase at similar rates thus are similar to each other.

Therefore, we use the average rank $\lfloor \bar{r}_i \rfloor$ across all symbol positions in G_i to represent how much information is received for G_i. It is rounded to an integer, because it is more meaningful to interpret the average rank as how many "pieces" are

---

[4] The property was original proved under random linear packet level NC, assuming $|\mathbb{F}_{2^q}|$ is large. The same applies to SLNC, which is also based on random linear coding.

received. It does not make much difference when the variation of $\bar{r}_i$ is smaller than 1. The range of the rank is in $[0, K]$; if $\lfloor \bar{r}_{u,i} \rfloor < K$, this means "some information in G_i is received"; and $\lfloor \bar{r}_{u,i} \rfloor = K$ means "G_i is received completely". Therefore, the total overhead becomes $L \cdot (log_2 K)$ bits, which equals 31B when $L = 50, K = 32$. Note that, this is independent of the piece size and also the file size. Now, the overhead takes an acceptable percentage ($\approx 10\%$) of the typical size of a safety message (300B) and is small enough to be piggybacked without affecting the QoS of safety applications [25]. The average rank representation is illustrated in Fig. 7.

D. Distributed Relay Selection in Cooperative PCD

Once vehicles are out of the range of an AP, they begin distributing the content cooperatively through the VANET. Due to the mobile nature of the VANET, the very notion of "cooperative" is captured in that vehicles distributively agree on a set of relay nodes, based only on local information.

1) Node utility calculation: In order to determine a set of relay nodes that can bring the largest useful amount of content to the others, each node needs to calculate its own "*utility*" based on neighbors' content reception status collected from the safety messages in the control channel. The *utility of a generation* at node u is defined as:

$$U(G_i, u) = \sum_{v \in \mathcal{N}(u)} Step(\lfloor \bar{r}_{u,i} \rfloor - \lfloor \bar{r}_{v,i} \rfloor), \tag{7}$$

where $Step(x) = x$, if $x > 0$, otherwise, $Step(x) = 0$. This quantity measures how much innovative information G_i of node u can provide to its neighbors in total.

The *utility* $U(v)$ of node v is defined as the maximum value among all generations' utilities of v. This estimates the maximum additional amount of innovative information v can provide to all neighbors, and reflects v's priority in accessing the wireless medium. We do not look at the aggregate utility of multiple generations, because to transmit many generations takes a long time while the VANET topology could change dramatically.

2) Transmission coordination: After nodes' priorities are determined, only a subset of the high-priority nodes (relays) will become the ones who actually broadcast their contents, in order to achieve high downloading rate and prevent from severe interference. Those relays are decided via a contention process, in a local and opportunistic way. In particular, the vehicles with the highest priorities in their locality should access the channel first, and suppress the others to avoid unnecessary packet collisions.

To this end, at the beginning of each service time slot, each vehicle v sets a backoff delay Δt which is inversely proportional to its utility before it makes channel access decision. When the timer expires, v senses the channel; if it is clear v will broadcast a short control message which is sent immediately by the MAC layer, even without additional random backoff in 802.11 [5]. Note that, an AP always has the highest utility, so it will be a relay every time if there are vehicles still in need of the file in its local range.

Backoff delay function. A straightforward one is as follows:

$$\Delta t(v) = \Big(1 - \frac{U(v)}{K \cdot |\mathcal{N}(v)|}\Big) \cdot \Delta t_{max}, \tag{8}$$

where parameter Δt_{max} is the maximum allowable backoff delay (e.g., 2ms). However, Eq. (8) suffers from a major problem. That is, each node v has different neighborhood and $\mathcal{N}(v)$. If v merely has one neighbor but its generation utility for G_i is K, it will have the highest priority and $\Delta t(v) = 0$. However, compared with another node w who has 10 neighbors and utility $5K$, v is obviously not as beneficial to the whole network as w. Ideally, the $|\mathcal{N}(v)|$ should be a maximum possible neighborhood size ($|\mathcal{N}|_{max}$) and be the same for all vehicles, so that they have a common basis of priority comparison. However, setting it to be a fixed value is undesirable since the vehicle density will change.

Therefore, we estimate the maximum local neighborhood size. To do so, each node broadcasts its neighborhood size to others, and propagates its own estimation about the maximum neighborhood size. After several rounds, all nodes can obtain the same $|\mathcal{N}|_{max}$. Although the VANET topology may change every tens of time slots so that $|\mathcal{N}|_{max}$ varies over that time, we actually need not to maintain the same $|\mathcal{N}|_{max}$ for all nodes in the network. Rather, it is sufficient for vehicles in a local 1-hop range to agree on the same estimated $\widehat{|\mathcal{N}|_{max}}$, while the local propagation requires only very few rounds to converge. To achieve this, each vehicle will attach its local estimate of $\widehat{|\mathcal{N}|_{max}}$ in the safety message, and update it in a way similar to distance updates in distance vector routing.

In addition, to resolve ties, a random jitter is added to the backoff delay of each vehicle. Thus, in CodeOn, each vehicle sets its backoff delay according to the following:

$$\Delta t(v) = \Big(1 - \frac{U(v)}{K \cdot \widehat{|\mathcal{N}|_{max}}}\Big) \cdot \Delta t_{max} + Rand(0, T_J). \tag{9}$$

where T_J is the maximum jitter.

Discussion of parameter selection. First, Δt_{max} must be large enough to distinguish two vehicles with adjacent utility rankings. For a common neighbor v_c of two vehicles v_1 and v_2, the minimum difference between $U(v_1)$ and $U(v_2)$ is 1. Therefore, the minimum difference between v_1 and v_2's backoff delays is $\min\{|\Delta t(v_1) - \Delta t(v_2)|\} = \frac{1}{K \widehat{|\mathcal{N}|_{max}}} \cdot \Delta t_{max}$, which should be larger than the signal propagation delay. When their distance $d(v_1, v_2) = 300$m the propagation delay is $\frac{300}{3 \times 10^8} = 1\mu$s. Therefore, we can choose $\Delta t_{max} > 2$ms, i.e., when $\widehat{|\mathcal{N}|_{max}} = 50$, $K = 32$, $\min\{|\Delta t(v_1) - \Delta t(v_2)|\} > 1.2\mu$s. On the other hand, Δt_{max} shall not be too large since it will waste bandwidth. For $\Delta t_{max} = 2$ms, if transmission of one generation spans 10 service time slots (500ms), the percentage of wasted time can be as low as $2/500 = 0.4\%$.

Second, T_J should be both large enough to distinguish two contending nodes v_1 and v_2 with the same utility, and small enough to preserve the priorities between nodes with different utilities. Assume all the contending nodes have the same neighbor set. Since node utility is an integer, for node

---

[5] If the random backoff delay in IEEE 802.11 broadcast superimposes on that of CodeOn's, the total backoff delays will be incorrect. This matters since the differences in CodeOn's backoff delays are on the order of that of 802.11's (100μs) when the reception status of all vehicles are similar and approaches completion.

v_1, the utility of the node v_3 with priority next to v_1 is at most $U(v_1) - |\mathcal{N}(v_1)|$ (since $\lfloor \bar{r}(v_3,i) \rfloor = \lfloor \bar{r}(v_1,j) \rfloor - 1$ for some G_i, G_j and every neighbor is counted once). Thus, the utility difference is at least $|\mathcal{N}(v_1)|$. Therefore, we have $T_J \approx \frac{\Delta t_{max}}{K}$ (e.g. 0.1ms). Note that, we do not consider $U(v_1) - U(v_3) \ll |\mathcal{N}(v_1)|$ since this is rare in reality, i.e., contending nodes always share a large portion of neighbors.

E. The merit of carrier sense under SLNC

We have used *carrier sense* in the contention process for transmission opportunities by potential relay nodes. That is, *a node quits the contention for channel access whenever it detect the energy of an ongoing transmission, otherwise it is allowed to transmit concurrently with others*. Traditionally for packet-level broadcast (with/without NC), this leads to the well-known "hidden terminal" problem, since such concurrent transmissions may cause interference at their neighbors[6]. Various mechanisms have been proposed to solve this problem, such as clearing the channel within a range larger than carrier sensing range [4]. However, due to SLNC's better tolerance in transmission errors and interference, more aggressive concurrent transmission may be possible. In the following we show that the simple carrier sensing rule actually provides near-optimal performance in terms of average downloading rate, as the impact of hidden terminals is greatly alleviated by SLNC.

The fundamental question is, with SLNC, what is the optimal distance between two nearby relay nodes so that the concurrent transmissions of all relays achieve highest average downloading rate? That is, what is the maximum spatial reusability that can be achieved? Intuitively, if the relays are faraway from each other, there is no interference but the space is not fully utilized; but if they are too close, severe interference will in turn decrease the downloading rate. First we define a quantity that reflects the average downloading rate in the network (assume all contents are useful for simplicity):

Definition 1 (average symbol reception probability): ($\chi(v_1, v_2, ..., v_n)$) For n relay nodes $v_1, v_2, ..., v_n$ in the network, the average probability that every vehicle receives one symbol from any of them during unit time (e.g., the period of one symbol's transmission).

A simple case. To derive χ, we first consider a scenario where there are only two relays v_1, v_2 in the network. We are interested in the relationship of average symbol reception probability with the inter-relay distance, and when can concurrent transmission gain advantage over non-concurrent transmission (i.e., two relays transmit separately and alternatively). The following characterizes the condition when concurrent transmission is better than separate transmission:

$$\alpha_c = \chi(v_1, v_2) > [\chi(v_1) + \chi(v_2)]/2, \tag{10}$$

where α_c is denoted as "concurrency gain". χ can be derived from symbol error probability (P_{se}) at each receiving node. However, it is hard to obtain the closed form solution of P_{se} under concurrent transmissions (see Appendix. A). Therefore, we approximate χ by the *average symbol reception ratio*:

$$\chi \approx \frac{\text{Total \# of symbols correctly received by all nodes in unit time}}{\text{Total number of receivers in the network}}, \tag{11}$$

which is obtained through Monte Carlo simulations. Given relays v_1 and v_2, for each of their neighbors w, the received signal to noise ratios (SNRs) are randomly sampled from Nakagami fading model (Eq. (15)), while successful symbol reception follows the probability $1 - P_{se|\gamma}$ (Eq. (16)). Under concurrent transmissions, the SINR is computed from Eq. (17). To simulate packet capture effect in reality[7], we let w receive the clean symbols in a packet from v_1 if its average SINR $\overline{\gamma_1} > \overline{\gamma_2}$, and vice versa. Note that, this estimation is done in the worst case, i.e., the relays transmit simultaneously so that every symbol is possibly interfered.

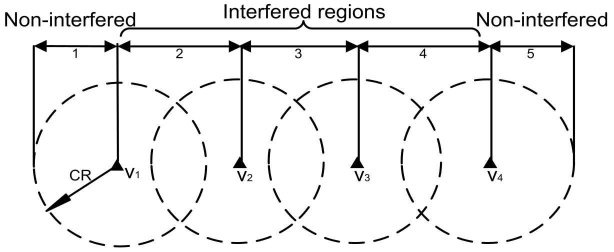


Fig. 9: Conceptual illustration of the n-relay concurrent transmission and calculation of average symbol reception probability.

For convenience of illustration, we define the following ranges under free space propagation model (Friis): (1) "energy detection range" (ER) (or carrier sensing range) for a transmitter, within which nearby nodes can detect its signal energy. We have $ER = \sqrt{\frac{T_p G}{Th_{ER}}}$, where Th_{ER} is the carrier sensing threshold, T_p is the transmission power and G is the antenna gain. (2) The "data communication range" (CR), in which nearby nodes can receive a data packet correctly. $CR = \sqrt{\frac{T_p G}{Th_{CR}}}$, where Th_{CR} is the data reception threshold. Normally $ER \geq CR$. These ranges imply statistical transition points across which nodes have different reception results.

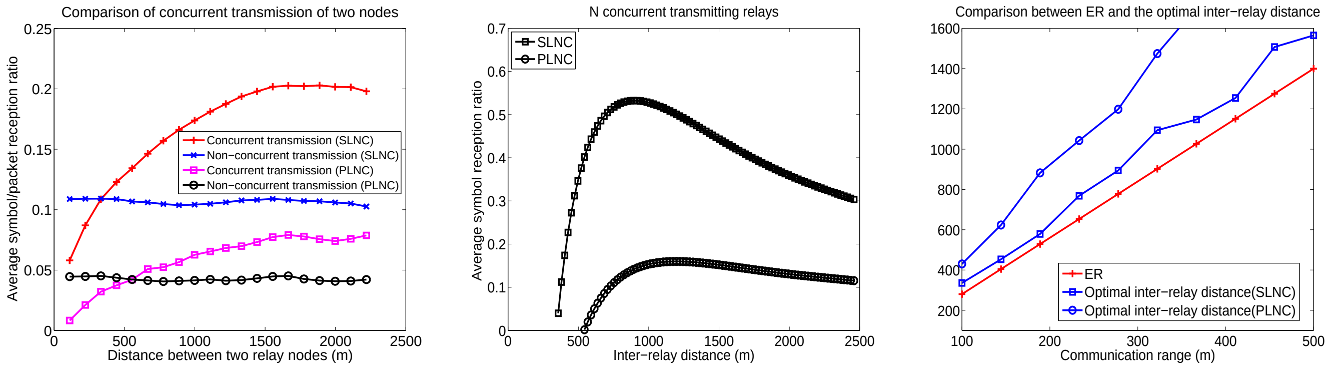
We generate 10 random topologies for VANET on a highway with traffic density 100/km, fix relay v_1 and change v_2's position. Each of v_1 and v_2 transmits 10 packets (each having 30 symbols). The number of received symbols is recorded for every node in the network. We also compare with PLNC under the same setting. $\chi(v_1, v_2)$ under the concurrent case is compared with $[\chi(v_1) + \chi(v_1)]/2$ under the non-concurrent case, against a changing inter-relay distance.

The results are given in Fig. 8 (a). It can be seen that, for SLNC when $d(v_1, v_2) = 2250$m, the concurrency gain $\alpha_c \approx 2$; when $d(v_1, v_2)$ decreases, α_c monotonically decreases until it becomes smaller than 1, at a small cross-distance d_c. While for PLNC, the average packet reception ratio is much smaller, and its cross distance is larger, which shows PLNC's inferior tolerance with concurrent transmission than SLNC.

---

[6] With packet-level broadcast, carrier sense is shown to work well under a two transmitter setting in [26]. Here we focus on a multi-transmitter setting instead, using SLNC.

[7] We assume no node can receive more than one symbol or packet from different transmitters at the same time.



(a) The effect of concurrent transmission of two relays. CR =250m, ER=700m, data rate: 12Mbps.

(b) The average symbol reception ratio under n concurrent transmitting relays. CR =250m, ER=700m.

(c) Comparison between ER and the optimal inter-relay distance for different CR.

Fig. 8: Optimal inter-relay distance for transmission coordination in CodeOn.

The n-relay case. The results of multiple ($n > 2$) relay nodes transmitting concurrently are based on that of the simple case. Without loss of generality, we assume that vehicles are uniformly distributed and are not too sparse so that neighbor conditions are similar. And n relays are assumed to lie on a straight highway (of length $\mathscr{L}$) with equal inter-distance d.

Now we derive the relationship of $\chi(v_1, v_2, ..., v_n)$ with $\chi(v_1, v_2)$. Let $d_c = d(v_1, v_2)$ when $\alpha_c = 1$ in the simple case. This point can be interpreted equivalently as half of the nodes around each relay are heavily interfered while the rest are not. Assuming uniform vehicle distribution, since d_c is larger than CR (Fig. 8 (a)), most of these interfered nodes locate in the region between v_1 and v_2. We are interested in $d(v_1, v_2) > d_c$, when approximately only the nodes within that region experience a decrease in their symbol reception probabilities. Therefore, a third relay v_3 adds little interference to the region between v_1 and v_2, which is illustrated in Fig. 9.

By assumption, $\chi(v_i, v_{i+1}) \approx \chi(v_j, v_{j+1}), \forall i, j$, so

$$\begin{aligned}\chi(v_1, v_2, ..., v_n) &= \chi(v_1, v_2)[1/2 + (n-1)(\alpha_c - 1) + 1/2] \\ &= \chi(v_1, v_2)[(n-1)(\alpha_c - 1) + 1], \qquad (12)\end{aligned}$$

where $n = \lfloor \frac{\mathscr{L}}{d} \rfloor$, and α_c is a function of d and CR. For each CR, α_c is obtained by simulation and curve fitting. Since α_c is increasing w.r.t. d, $\chi(v_1, v_2, ..., v_n)$ has a maximal point.

To see the gain of $\chi(v_1, v_2, ..., v_n)$ under SLNC versus PLNC, the results are shown in Fig. 8 (b) for $\mathscr{L}$ =10km and CR =250m. Two observations follow: (1) the maximum average symbol reception probability with SLNC is higher than PLNC's average packet reception probability defined likewise, indicating SLNC achieves higher average downloading rate. (2) the optimal point of d with SLNC is smaller than that of PLNC, meaning SLNC encourages higher spatial reusability in concurrent transmission due to its better error-tolerance.

In Fig. 8 (c) we present the relationship of optimal d with ER for various CRs. Surprisingly, the optimal d is close to ER under SLNC for a wide range of CR, while it is not the case for PLNC. Due to SLNC's high interference-tolerance, only when d approaches d_c does the interference level increase rapidly. For multiple concurrent relays, the contradicting interplay between increased spatial reuse and decreased reception probability results in the optimal inter-distance to be slightly larger than ER.

From the above results, we can conclude that the simple carrier sense medium access rule in CodeOn can achieve close to maximum average downloading rate for PCD in the VANET. The advantage is that, there is no need to employ additional mechanisms to avoid hidden terminals as in traditional broadcast schemes; the only information needed for each node to make distributed decision is its local channel status and its backoff timer, which greatly simplifies the protocol design.

F. *Broadcast Content Scheduling*

Finally, we briefly highlight the way that broadcast content scheduling is dealt with in CodeOn.

1) Content scheduling at APs: In CodeOn, the APs broadcast the contents in a round-robin way to maintain the "information difference" between vehicles moving out of the AP range at different times. In order to make more efficient use of the VANET bandwidth, the content scheduling should also be aware of local vehicles' reception status. Therefore an AP will sort its file generations according to their utilities; in addition to round-robin, it transmits the one with both larger ID and the highest utility that hasn't been transmitted in the last "batch".

2) Content scheduling at vehicles: After a vehicle becomes a relay node, it broadcasts the generation with the maximum utility. To avoid from transmitting duplicate information, it is important for vehicles to decide when to stop the transmission.

To this end, we estimate the number of pieces that each relay should send in one batch. The intended number of (innovative) pieces that v sends to a neighbor w for G_i is estimated as $K_{v,w} = Step(\lfloor \bar{r}_{v,i} \rfloor - \lfloor \bar{r}_{w,i} \rfloor)$. Then, the number of pieces that v should send to all neighbors for G_i is computed as

$$Z_v(G_i) = \lceil \frac{1}{|\mathcal{N}(v)|} \sum_{w \in \mathcal{N}(v)} K_{v,w} \rceil, \qquad (13)$$

which is also the size of a batch. When the average rank $\bar{r}_{v,i}$ and those of all of its neighbors are equal to K (full rank), we set $Z_v(G_i) = 0$. Note that, the above is a conservative estimation, which treats the link qualities as perfect.

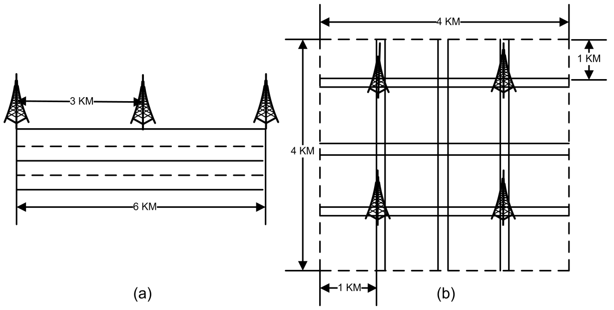


Fig. 10: (a) Highway scenario. (b) Urban scenario.

In addition, we need to deal with two situations. (1) If a batch spans multiple service time slots, relay v accesses the channel deterministically by setting its $\Delta t(v) = 0$ during the following time slots in order to finish transmitting its batch. (2) If the transmission of a batch terminates before the end of some service time slot k, to avoid waste of VANET bandwidth, v will fill the rest of the channel by transmitting additional coded pieces from the same G_i until time slot k is used up.

V. PERFORMANCE EVALUATION

A. Methodology

In this section, we evaluate the performance of CodeOn by simulations. We compare CodeOn with an enhanced version of CodeTorrent [10], which is pull-based and uses PLNC. The AP is treated as a normal node. Each node periodically broadcasts a gossip message to tell others about its content availability. Based on this, a node v periodically broadcasts a downloading request, asking for the index of the rarest generation G_i among its neighbors, and attaches a null-space vector of G_i computed from v's corresponding decoding matrix. Each neighbor w, upon receiving the request, checks if it has G_i. If yes, and if the null-space vector is not orthogonal to the subspace spanned by w's coding vectors of G_i, w responds v with one coded piece from G_i via unicast, after waiting for a random backoff delay to reduce collisions. Only the first packet in a piece contains the coding vector; if that packet is lost then the whole piece is lost. Upon successful reception of a piece, node v continues sending another downloading request. Otherwise, v waits till the next period to broadcast its request. Nodes other than v exploit opportunistic overhearing, i.e., buffer a piece sent to v if that piece is useful and received correctly.

We made the following additional modifications to CodeTorrent. We equip it with multi-channel capability as in CodeOn. To ensure a fair comparison, we apply the same channel switching mechanism in CodeTorrent, which results in a 1/2 reduction in the downloading rate. Also, in order to increase the success probability of overhearing, each node is allowed to receive multiple different pieces during the reception of one piece. Moreover, the packets in a piece do not have to arrive in order; a node flushes an incomplete piece after a certain time from its first reception, say 0.5s.

In addition, we introduce a variation of CodeOn, *CodeOn-Basic*, which is also push-based, piece-division but based on PLNC. A piece is used as a whole for encoding and decoding. A node buffers any overheard piece as long as it receives the coding vector in the first packet of that piece, and the same buffer flushing mechanism as in CodeTorrent is adopted. Moreover, in content scheduling a relay node pads a service slot with whole pieces. If the remaining service slot time is not enough for sending a whole piece, it terminates the current batch, rather than filling with individual packets. Other than that, CodeOnBasic is the same with CodeOn.

TABLE II: Simulation parameter settings

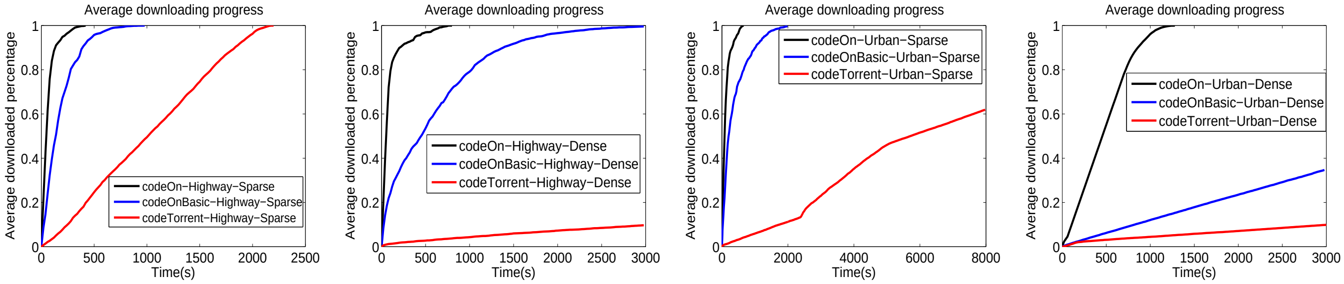
CodeOn/CodeOnBasic		CodeTorrent	
Δt_{max}	2ms	Maximum random backoff delay	5ms
T_J	100μs	Gossip interval	0.5s
		Periodic Request interval	0.5s
		Unicast retry limit	7
Common parameters			
$\lvert F\rvert$		16MB	
L		50	
K		32	
M		16	
q		8	
J/N		10 (J = 10KB)	
CR, ER		250m, 700m	
Data rate, base rate		12Mbps (16QAM), 3Mbps (BPSK)	
SNR thresholds		15dB, 4dB	
Data capture threshold		20dB	
Data/safety message sizes		1KB, 256B (without header)	
Propagation model		Nakagami $m = 3$	

We implemented CodeOn, CodeOnBasic and CodeTorrent in NS-2.34 [27]. For CodeOn, we implemented the run-length coding with dynamic programming algorithm to minimize the communication overhead in sending each coded piece [13]. In simulation the number of runs seldom exceeds 20 for 10KB pieces. We simulate independent symbol errors in a packet by first computing packet error probability using the propagation model, and then derive P_{se} use Eq. (6). Packet capture effect is enabled; and when two packets collide, if no packet can be captured, the symbols from the point of collision are all discarded. Otherwise, the captured packet is received as usual. We do not consider vehicular buffer constraints.

We have a few notes on broadcast data rate selection. First, the safety message's communication range shall be larger than that of PCD data packets, so that the neighbor set used in relay selection can cover the set of nodes that can receive a data packet. Otherwise, the utility cannot truthfully reflect a node's total content usefulness. Considering the reliability of safety messages, we chose the base rate (3Mbps) for broadcasting safety messages. Second, we want to achieve high downloading rate for PCD. For SLNC, choosing a higher data rate is beneficial because it has better error-tolerance. Since a too high rate is also undesirable due to very small communication range, the data rate of PCD packets is set to be 12Mbps throughout the paper. The determining of optimal data rates is out of the scope of this paper.

B. Simulation Settings

We consider both highway and urban scenarios (Fig. 10). We use a VANET mobility generator [28] to generate the movement patterns. Vehicles are placed uniformly at random in the road area; when a vehicle hits the boundary it ran-

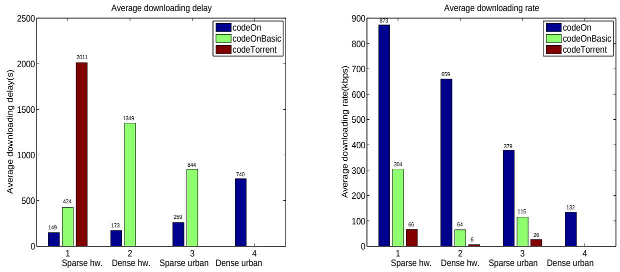


(a) Sparse highway.
(b) Dense highway.
(c) Sparse urban.
(d) Dense urban.

Fig. 11: Downloading progresses.

domly selects another entry point of the map. This removes the boundary effect; equivalently, the AoI is infinitely large. Table. II is a list of parameters.

The highway scenario consists of a bi-direction, four lane highway with length 6km. Vehicles' speeds are randomly drawn from $[20, 30]$m/s with a maximum acceleration of $0.5m/s^2$. The urban scenario is 4km×4km as shown in Fig. 10. In order to evaluate the impact of topology and traffic density, we simulate sparse and dense traffic for both scenarios. The sparse settings simulate delay-tolerant network (DTN), where the total number of vehicles is 100 for highway and 160 for urban. The dense highway setting has 300 vehicles while the dense urban has 400 vehicles.



(a) Average downloading delay.
(b) Average downloading rate.

Fig. 12: Downloading delays and rates.

C. Results

1) Downloading performance: We evaluate the downloading performance from three aspects: (1) downloading progress, which is the change of average downloaded percentage of the file with the elapsed time (averaged upon each vehicle); (2) average downloading delay: the average elapsed time from downloading start to 100% completion; (3) average downloading rate, where the downloading rate for each vehicle is the file size divided by its downloading delay.

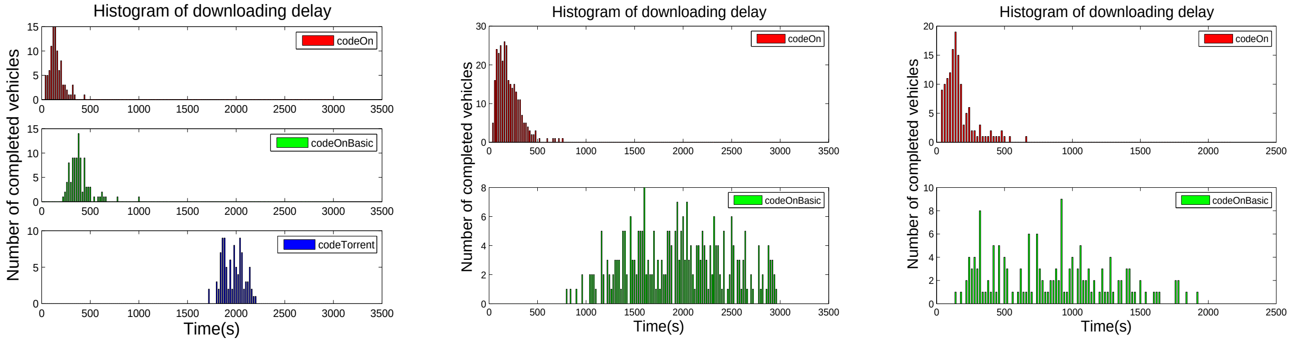
We present the downloading progresses in Fig. 11 for all three scenarios. It can be seen that CodeOn significantly outperforms both CodeOnBasic and CodeTorrent. The downloading progress of CodeOn is the fastest (Figs. 11 (a)–(d)), especially when the average downloaded file percentage is below 90%. The comparison between CodeOnBasic over CodeTorrent demonstrates the effectiveness of our new set of push-based protocol design, while the comparison between CodeOn and CodeOnBasic shows the advantage of the use of SLNC, which we will discuss later.

Next, we evaluate the average downloading delays and rates in Fig. 12. Some of the average delays are not shown since their downloading progresses cannot reach 100% within the given simulation period. There are two key observations. First, the average downloading rates of CodeOn are much higher than both CodeOnBasic and CodeTorrent, for both highway and urban scenarios and both sparse and dense traffic. Second, CodeOn maintains high downloading rate in all cases shown, especially for the two extremes, i.e., sparse urban scenario and dense highway cases which represent the lowest and highest traffic density, respectively. This means CodeOn is the most robust to variations in topology and vehicle density.

The first phenomenon above is attributed to the push-based protocol design combined with SLNC. In CodeOn, using a prioritized relay selection mechanism with the transmission coordination that avoids heavy packet collisions, the contents can be distributed proactively to the vehicles in the AoI so that the VANET bandwidth is fully utilized. Moreover, each piece of transmitted content brings the maximum usefulness to a relay's whole neighborhood. In addition, with SLNC, the symbols in content pieces are received with higher-speed from APs and relays, which results in higher downloading rate.

The robustness of CodeOn under low traffic density is mainly attributed to the enhanced reception reliability brought by SLNC. Compared with PLNC, SLNC actually enables vehicles in a larger range to receive some useful information in a piece. In the sparse urban setting, although the vehicular contact opportunities are much less, CodeOn is able to mitigate the impact of low traffic density.

On the other hand, CodeOn is less affected under dense VANET. For the dense scenarios, the differences between CodeOn's downloading rates and those of CodeOnBasic and CodeTorrent are both larger than the sparse scenarios (Fig. 12 (b)). For CodeTorrent, the performance degradation is due to lack of coordination and using of PLNC for a large file. (1) Under dense VANET, the number of requesting vehicles in a node's neighborhood increases. Since there may be more than one responder for each requester, the chance of packet collisions also increases. The unicast-with-overhearing mechanism retransmits packets after they are collided, which aggravates



(a) Sparse highway.

(b) Dense highway.

(c) Sparse urban.

Fig. 13: The distributions of downloading delays.

the problem. (2) For both CodeTorrent and CodeOnBasic, the use of PLNC prevents a requester from receiving a whole piece under frequent packet collisions. However, through prioritized relay selection and the use of SLNC, CodeOn alleviates the above problems dramatically.

2) Fairness: The fairness is embodied in the distribution of downloading delays of all vehicles, shown in Fig. 13. We show the distributions for all three cases. The most fair situation has variance 0, i.e, all the delays equal the average value. From Figs. 13 (a)-(c), one can see that the distributions of CodeOn are more concentrated (more fair) than those of CodeOnBasic and CodeTorrent. Few vehicles need very long time to receive the whole file. Again, the same robustness of CodeOn to variations in traffic density can be observed.

The superiority of CodeOn in fairness is still attributed to the use of SLNC. SLNC enables more reliable reception of the coded symbols, since an overhearing node will buffer any innovative clean symbol it received. In CodeOn, since the granularity of information reception is smaller, and vehicles have similar opportunities to contact with APs and other vehicles within a time period of order 1000s, their reception progresses have small variance. However in CodeOnBasic and CodeTorrent, a vehicle either receives a whole piece or receive nothing, so the variance among reception progresses is larger. Again, the results on fairness demonstrate the benefit of using SLNC and the effectiveness of CodeOn's protocol design.

3) Protocol efficiency: One may wonder if CodeOn achieves fast push-based downloading by sacrificing protocol efficiency. To further investigate this issue, we present the results on protocol efficiency in Table. III.

As we have shown in Sec. IV-C, the protocol overhead of CodeOn is small. To evaluate the amount of incurred data traffic, we show the average number of pieces sent by a vehicle and an AP during the whole simulation time (all nodes stop transmitting if all of their neighbors receive 100% of the file) (see Table. III). CodeOn has the fewest number among the three protocols. Its high protocol efficiency comes from both the high symbol reception probability due to SLNC, and the high usefulness of the transmitted symbols due to relay selection. As CodeOnBasic adopts the same relay selection mechanism, it enjoys similar high protocol efficiency to CodeOn. However, CodeTorrent sends many pieces due to a large number of failed overhears explained in the following. Note that, the APs are always the most advantageous nodes so they transmit a lot in all three protocols.

To further study the role of relay selection, we compute the percentage of total number of non-innovative pieces out of the total number of received pieces, which reflects the usefulness of the received content. Also, we calculate the average number of failed overheard pieces (in which the coding vectors are received but not all the subsequent packets) per received piece. For the former, CodeOnBasic is slightly higher than CodeTorrent; but for the latter, CodeOnBasic is much lower than CodeTorrent. This is because in CodeTorrent a responder uses the requester's null-space vector to decide whether to transmit a coded piece, which is definitely innovative to the requester. However, in CodeTorrent a responder's transmission mainly benefits the requester itself but few others due to uncoordinated transmissions. On the other hand, in CodeOnBasic the selected relays can benefit their whole neighborhood, while the broadcasted contents are still highly useful. As a result, both the downloading rate and efficiency are high.

4) Discussion: Finally, we give some insights that can be obtained from our results.

Push v.s. pull. First we compare the *push versus pull* based content distribution in VANETs. CodeOnBasic and CodeTorrent are both based on PLNC, but the former performs much better than the latter for all scenarios in Figs. 11 and 12. An obvious reason is the difference on the bandwidth utilization. CodeOnBasic let the APs and relays broadcast proactively (push), so that the service time slots are almost fully utilized. However, in CodeTorrent each node make requests (pull) periodically and responders transmit passively. Whenever received a piece in error, a requester will wait until the next period to make subsequent requests. Due to the lossy property of the wireless channel in VANETs, this happens frequently so that the service channel is under-utilized.

However, a more fundamental reason that the push method in CodeOn and CodeOnBasic is better, goes to the relay selection mechanism. If there was no transmission coordination between vehicles, the push-based content distribution could easily lead to frequent packet collisions. For CodeTorrent which is pull-based, its high chance of packet collisions is already evident from the large number of failed overheard

TABLE III: Protocol efficiency (Total number of pieces in the file: 1600).

Protocols	Percentage of non-innovative received pieces	Average # of failed overheard pieces per received piece	Average # of pieces sent by a vehicle	Average # of pieces sent by an AP
Sparse highway scenario				
CodeOn	N/A	N/A	2202.12	26023.00
CodeOnBasic	0.476	4.26	4054.87	51578.00
CodeTorrent	0.325	27.27	32889.87	53665.00
Sparse urban scenario				
CodeOn	N/A	N/A	1031.14	43445.25
CodeOnBasic	0.228	3.47	3525.31	143905.00
CodeTorrent	0.167	80.74	52465.69	222287.50

pieces of CodeTorrent in Table III. One can imagine that this situation will be aggravated if CodeTorrent is changed to push-based where nodes transmit more aggressively.

Apart from transmission coordination, in designing a push-based protocol, it is always critical to maximize the usefulness of the broadcasted content from each relay nodes. Since nodes do not make explicit downloading requests, and since "push" uses broadcast transmission in nature, it is basically impossible to ensure the usefulness of broadcast content of a relay for all its neighbors. In CodeOn and CodeOnBasic, our approach is to select a relay to be the one that can bring maximum amount of useful contents to all its neighbors, by implicitly calculating node utilities based on fuzzy average rank differences. In contrast, in CodeTorrent each responder will only ensure the content to be 100% innovative for one requestor, using accurate null-space indicators. Interestingly, as one can see from the number of non-innovative pieces in Table III, the number of CodeOnBasic is quite close to that of CodeTorrent, which can be regarded as a lower-bound. This proves the effectiveness of our relay selection approach.

SLNC v.s. PLNC. The advantage of using SLNC is evident by comparing CodeOn with CodeOnBasic in Fig. 11, which are only different in the network coding method. With PLNC, in CodeOnBasic a coded packet is discarded whenever it is received in error, which leads to unsuccessful reception of the whole piece. However, with SLNC, CodeOn records every innovative received symbol in a piece, and then combines innovative symbols to decode the piece.

As previously mentioned, SLNC is superior in tolerating transmission error. This is a direct reason of why CodeOn has the best robustness under dense traffic scenarios. By both coding and receiving according to a small granularity of symbols (yielding higher content diversity), vehicles have higher chances of receiving some useful information, even when packet collisions are frequent due to dense traffic, or when there are few vehicles or APs around. However, with PLNC, the content diversity is lower. Although our push-based protocol design is able to choose the best relay nodes and alleviate collision, without SLNC, small downloading delays and a high level of fairness are very hard to achieve for all topologies and traffic densities.

VI. CONCLUDING REMARKS

In this paper, we have presented CodeOn, a novel push-based popular content distribution scheme in vehicular networks, where large files are broadcasted proactively from a few APs to vehicles inside an interested area. CodeOn is designed to primarily achieve high downloading rate and high protocol efficiency. To combat the lossy wireless transmissions in VANETs, we leverage symbol level network coding (SLNC), which enjoys the benefits of both network coding and symbol-level diversity. The use of SLNC contributes as a key factor for the superior and robust performance of PCD across VANETs with different traffic densities and topologies. In addition, to allow "push" efficiently without broadcasting useless information and to avoid from incurring frequent packet collisions, we designed a prioritized relay selection algorithm along with a lightweight transmission coordination mechanism, which are shown to improve greatly upon a previous pull-based protocol, CodeTorrent. Compared with CodeTorrent, CodeOn achieves a significant gain in terms of average downloading rate, where one important part of it comes from the use of SLNC, and the other is attributed to the new push-based protocol design. Our work demonstrates the strong potential to achieve fast PCD in realistic vehicular networks.

APPENDIX A
DERIVATION OF THE SYMBOL ERROR PROBABILITY

Assuming the underlying modulation scheme is M-QAM[8]. We want to obtain the symbol error probability at a receiver (P_{se}) as a function of the distance to the transmitter. Denote the SNR for a symbol as γ, then

$$P_{se} = \int_0^\infty P_{se|\gamma} f(\gamma), \tag{14}$$

where $f(\gamma)$ is the PDF of the SNR. For Nakagami,

$$f(\gamma, m, \Omega) = \frac{m^m}{\Gamma(m)\Omega^m} \gamma^{m-1} e^{-(m\gamma/\Omega)}, \tag{15}$$

where $\Omega = \mathbb{E}[\gamma]$ is the average SNR and m is the fading parameter. From basic communication theory [29],

$$P_{se|\gamma} = 1 - \Big(1 - 2\Big(1 - \frac{1}{\sqrt{M}}\Big) Q\Big(\sqrt{\frac{3\gamma}{M-1}}\Big)^2\Big). \tag{16}$$

Without concurrent transmission, the closed form formula of P_{se} can be found in [30] (Eq. (8.109)). Under concurrent transmissions, the SINR at node v_1 is:

$$\gamma_1' = \frac{N_0\gamma_1}{N_0 + N0\gamma_2} = \frac{\gamma_1}{1+\gamma_2}, \tag{17}$$

where N_0 is the noise power, γ_1 and γ_1 are independent r.v., with distributions $f(\gamma_1, m, \Omega_1)$ and $f(\gamma_2, m, \Omega_2)$ respectively.

---

[8] This is specified by DSRC.

Theoretically, the PDF of γ_1' can be obtained as a function of γ_1', m, Ω_1 and Ω_2. But it is very hard to compute the closed form solution of Eq. (14). Although Eq. (16) can be approximated by an exponential function [31], the approximation error is large.

ACKNOWLEDGEMENTS

This work was supported in part by the US National Science Foundation under grants CNS-0746977, CNS-0716306, and CNS-0831628. We thank Uichin Lee for providing codes of network coding in CodeTorrent and his useful discussions. We also thank the anonymous reviewers for their helpful comments.

REFERENCES

[1] "Dedicated Short Range Communications." [Online]. Available: http://www.standards.its.dot.gov/Documents/advisories/dsrc_advisory.htm

[2] D. Jiang, V. Taliwal, A. Meier, W. Holfelder, and R. Herrtwich, "Design of 5.9 ghz dsrc-based vehicular safety communication," *IEEE Wireless Communications*, vol. 13, no. 5, pp. 36–43, October 2006.

[3] "Ieee trial-use standard for wireless access in vehicular environments (wave) - multi-channel operation," *IEEE Std 1609.4-2006*, pp. c1–74, 2006.

[4] M. Li, K. Zeng, and W. Lou, "Oppcast: Opportunistic broadcast of warning messages in vanets with unreliable links," in *IEEE MASS '09*, Oct. 2009.

[5] U. Lee, J.-S. Park, J. Yeh, G. Pau, and M. Gerla, "Code torrent: content distribution using network coding in vanet," in *MobiShare '06*, 2006, pp. 1–5.

[6] A. Nandan, S. Das, G. Pau, M. Gerla, and M. Y. Sanadidi, "Co-operative downloading in vehicular ad-hoc wireless networks," in *WONS '05*, 2005, pp. 32–41.

[7] M. Fiore and J. M. Barcelo-Ordinas, "Cooperative download in urban vehicular networks," in *IEEE MASS '09*, Oct. 2009.

[8] J.-S. Park, U. Lee, S. Y. Oh, M. Gerla, and D. S. Lun, "Emergency related video streaming in vanet using network coding," in *ACM VANET '06*, 2006, pp. 102–103.

[9] S. Ahmed and S. S. Kanhere, "Vanetcode: network coding to enhance cooperative downloading in vehicular ad-hoc networks," in *IWCMC '06*, 2006, pp. 527–532.

[10] S.-H. Lee, U. Lee, K.-W. Lee, and M. Gerla, "Content distribution in vanets using network coding: The effect of disk i/o and processing o/h," in *IEEE SECON '08*, June 2008, pp. 117–125.

[11] M. Johnson, L. D. Nardis, and K. Ramchandran, "Collaborative content distribution for vehicular ad hoc networks," in *Proceedings of 44th Allerton Conference on Communication, Control, and Computing*, 2006.

[12] R. Ahlswede, N. Cai, S.-Y. Li, and R. Yeung, "Network information flow," *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1204–1216, Jul 2000.

[13] S. Katti, D. Katabi, H. Balakrishnan, and M. Medard, "Symbol-level network coding for wireless mesh networks," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 4, pp. 401–412, 2008.

[14] T. Mak, K. Laberteaux, R. Sengupta, and M. Ergen, "Multichannel medium access control for dedicated short-range communications," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 1, pp. 349–366, Jan. 2009.

[15] A. Nandan, S. Tewari, S. Das, M. Gerla, and L. Kleinrock, "AdTorrent: Delivering Location Cognizant Advertisements to Car Networks," in *WONS '06*, 2006, pp. 203–212.

[16] T. Ho, M. Medard, R. Koetter, D. Karger, M. Effros, J. Shi, and B. Leong, "A random linear network coding approach to multicast," *IEEE Transactions on Information Theory*, vol. 52, no. 10, pp. 4413–4430, Oct. 2006.

[17] K. Ramachandran, M. Gruteser, R. Onishi, and T. Hikita, "Experimental analysis of broadcast reliability in dense vehicular networks," *IEEE Vehicular Technology Magazine*, vol. 2, no. 4, pp. 26–32, Dec. 2007.

[18] M. Torrent-Moreno, F. Schmidt-Eisenlohr, H. Fussler, and H. Hartenstein, "Effects of a realistic channel model on packet forwarding in vehicular ad hoc networks," in *IEEE WCNC*, 2006, pp. 385–391.

[19] V. Taliwal, D. Jiang, H. Mangold, C. Chen, and R. Sengupta, "Empirical determination of channel characteristics for dsrc vehicle-to-vehicle communication," in *ACM VANET '04*. ACM, 2004.

[20] J. Zhang, Q. Zhang, and W. Jia, "Vc-mac: A cooperative mac protocol in vehicular networks," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 3, pp. 1561–1571, March 2009.

[21] J. Zhao, Y. Zhang, and G. Cao, "Data pouring and buffering on the road: A new data dissemination paradigm for vehicular ad hoc networks," *IEEE Transactions on Vehicular Technology*, vol. 56, no. 6, pp. 3266 –3277, Nov. 2007.

[22] J. Zhao, T. Arnold, Y. Zhang, and G. Cao, "Extending drive-thru data access by vehicle-to-vehicle relay," in *ACM VANET '08*, 2008, pp. 66–75.

[23] Z. Yang, M. Li, and W. Lou, "R-code: network coding based reliable broadcast in wireless mesh networks with unreliable links," in *IEEE GLOBECOM'09*, 2009, pp. 2168–2173.

[24] Y. Zhang, J. Zhao, and G. Cao, "Roadcast: A popularity aware content sharing scheme in vanets," *IEEE ICDCS '09*, pp. 223–230, 2009.

[25] Q. Xu, T. Mak, J. Ko, and R. Sengupta, "Vehicle-to-vehicle safety messaging in dsrc," in *ACM VANET '04*, 2004, pp. 19–28.

[26] M. Z. Brodsky and R. T. Morris, "In defense of wireless carrier sense," in *ACM SIGCOMM '09*, 2009, pp. 147–158.

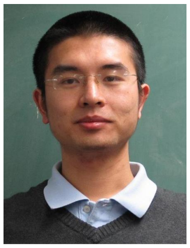
[27] "Ns2," *http://www.isi.edu/nsnam/ns*.

[28] "The usc mobility generator tool." [Online]. Available: http://nile.cise.ufl.edu/important/software.htm

[29] J. G. Proakis, *Digital communications*. McGraw-Hill, New York, 1995.

[30] M. K. Simon and M.-S. Alouini, *Digital Communication over Fading Channels*. Wiley-IEEE Press, 2004.

[31] F. F. Digham and M.-S. Alouini, "Average probability of packet error with diversity reception over arbitrarily correlated fading channels," *Wireless Communications and Mobile Computing*, pp. 155–173, 2004.



Ming Li (S'08) received his B.E and M.E degrees in Electronic and Information Engineering both from Beihang University, China, in 2005 and 2008, respectively. He is currently a Ph.D. student in the Electrical and Computer Engineering department at Worcester Polytechnic Institute. His current research interests include: wireless ad hoc and sensor networks, with emphases on protocol design and network security; and security & privacy in E-healthcare systems.

Zhenyu Yang (S'08) received his B.E and M.E degrees in Computer Science both from Xi'an Jiaotong University, China, in 2004 and 2007, respectively. He is currently a Ph.D. student in the Electrical and Computer Engineering department at Worcester Polytechnic Institute. His current research interests are in the area of wireless networks and network security, with emphases on network coding and protocol design.

Wenjing Lou (S'01 - M'03 - SM'08) earned a BE and an ME in Computer Science and Engineering at Xi'an Jiaotong University in China, an MASc in Computer Communications at the Nanyang Technological University in Singapore, and a PhD in Electrical and Computer Engineering at the University of Florida. She joined the Electrical and Computer Engineering department at Worcester Polytechnic Institute as an assistant professor in 2003, where she is now an associate professor. Her current research interests are in the areas of ad hoc, sensor, and mesh networks, with emphases on network security and routing issues. She was a recipient of the U.S. National Science Foundation Faculty Early Career Development (CAREER) award in 2008.