CodePlay: Live Multimedia Streaming in VANETs Using Symbol-Level Network Coding

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

***Abstract*—The fundamental challenges of providing live multimedia streaming (LMS) services in vehicular ad hoc networks (VANETs) come from achieving stable and high streaming rate (smooth playback) for all the interested vehicles while using minimal bandwidth resources, especially under the highly dynamic topology of VANETs and the lossy nature of vehicular wireless communications. Packet level network coding (PLNC) technique has been widely accepted as an effective approach to improve the network performance during the last decade. More recent *symbol-level network coding* (SLNC) could further improve the efficiency of bandwidth utilization by exploiting both wireless symbol-level diversity and the benefits of network coding. In this paper, we introduce CodePlay, a new LMS scheme in VANETs that fully takes advantage of SLNC through a coordinated local push mechanism. Streaming contents are actively disseminated from dedicated sources to interested vehicles via local coordination of distributively selected relays, each of which will ensure smooth playback for vehicles nearby. Extensive simulations show that simply replacing the SLNC with PLNC technique in previous LMS schemes can not provide satisfiable user experience, and special scheme design based on the unique characteristics of SLNC proposed in CodePlay is necessary for future LMS applications in VANET.**

***Index Terms*—Vehicular networks, symbol-level network coding, live multimedia streaming.**

I. Introduction

Live multimedia streaming (LMS) is promising in vehicular communications due to its more precise, comprehensive and user friendly merits compared with plain text based services. Typical scenarios for LMS applications could be illustrated as the following example. A roadside access point (AP) continuously broadcasts the streaming video of the current road traffic conditions to vehicles driving towards it for intelligent navigation, which is especially useful in inclement weathers. In order to provide the described services, we can utilize APs to disseminate the streaming content to vehicles passing by. However, due to the relatively high deployment cost of roadside APs and each AP's limited communication range, the entire road can not be fully covered merely by APs. Therefore, the vehicles have to form a vehicular ad hoc network (VANET) and cooperatively propagate the streaming content when they are out of coverage of APs.

Generally speaking, there are three primary requirements for LMS services in VANETs. Firstly, considering the large volume of each LMS content, all the receivers should achieve stable and high streaming rate for smooth playback. Note that the rate only needs to reach the requirements of related multimedia standards and higher rate is not necessary. Secondly, the service delivery delay should be short for all the receivers, and the delay variation should be small for neighboring receivers for possible coordinated actions between them, for example, bypassing a blocked road. Thirdly, LMS services should consume minimal amount of bandwidth resource for better coexistence with other competing services, since the bandwidth is a precious resource in VANETs. Essentially, this corresponds to improving bandwidth efficiency.

These requirements are conflicting with each other and are very challenging to be achieved simultaneously. In order to ensure smooth playback of LMS content, we have to combat with the lossy vehicular wireless links and highly mobile and dynamic topology of the underlying VANETs. In vehicular communications, packet loss is a frequent phenomenon due to channel fading. To ensure stable streaming reception within short time delay, a large number of (re)transmissions would be incurred, which severely decreases the bandwidth efficiency. In addition, smooth playback requires vehicles to make local optimal transmission decisions, such as which vehicle should transmit what content to which neighbors. This means vehicles need to learn precise and in-time neighbor information (such as reception status). However, under VANETs with ever-changing topology, this learning process may lead to high communication overhead. Thirdly, VANETs tend to experience frequent partitions [1], which increases the difficulty of determining the best relay nodes and proper transmission opportunities for them. This may result in major performance degradation without careful protocol design.

In this paper[1], we try to exploit symbol-level network coding (SLNC) [3] for designing a distributed live multimedia streaming scheme in VANETs. Compared with traditional packet-level network coding (PLNC), SLNC performs network coding on smaller symbols, which refers to a group of consecutive bits within a packet. SLNC not only enjoys the benefits of NC, but also gains from exploiting the symbol-level diversity in wireless transmissions [4]. By recovering correctly received symbols from erroneous packets, SLNC mitigates the impact of lossy links and packet collisions, improves the utility of each transmission and in turn reduces the total number of transmissions. However, how to provide satisfiable LMS services in VANETs with minimal bandwidth is not a trivial problem even with the help of SLNC. To this end, we make the following main contributions.

- We proposed CodePlay to fully exploit the benefits of SLNC in VANETs, the core of which is a coordinated local push mechanism. In order to disseminate

Manuscript received January 9, 2012; revised March 13, 2012; accepted April 17, 2012. The associate editor coordinating the review of this paper and approving it for publication was N. Kato.

Z. Yang is with Amazon.com, WA 98102 (e-mail: zhenyuy@amazon.com).

M. Li is with the Department of Computer Science, Utah State University (e-mail: ming.li@usu.edu).

W. Lou is with the Department of Computer Science, Virginia Tech (e-mail: and wjlou@vt.edu).

Digital Object Identifier 10.1109/TWC.2012.052512.120049

[1]The preliminary version of this paper appeared in [2].

the streaming content from sources to all the receivers timely and smoothly, a group of spatially separated relays are selected distributively, whose transmissions can bring most useful information to vehicles nearby. Each relay actively pushes coded information to cover its neighborhood. By taking advantage of SLNC's better tolerance for transmission interference, the concurrent transmissions of all relays could be optimally coordinated locally, which could provide continuous streaming coverage for the whole VANET efficiently.

- To enable CodePlay to perform well under various VANET densities, we also proposed an opportunistic transmission scheduling algorithm based on well-designed carrier sensing mechanism, where the network's spatial reusability can be adaptively enhanced with negligible overheads.
- We implemented CodePlay in NS-2 and carried out extensive simulations to evaluate its performance by various practical metrics. We showed both the potential and the constraints of providing LMS services in VANETs. Compared with traditional PLNC technique, the adoption of SLNC can provide more and better design choices for VANET designers. Also the particular topological characteristic of VANETs [1] (the vehicles running on the highway tend to form disjoint clusters rather than uniformly distributed) needs to be specifically considered into the scheme design. As far as we know, CodePlay made the first step towards this direction.

II. Related Work

A. NC-based streaming schemes

Streaming services are widely deployed on the Internet nowadays, such as PPLive, PPStream, etc. In particular, network coding (NC) [5], by allowing nodes to combine different packets received previously together to generate coded packets for transmitting, has been shown to be an effective technique that can improve the user experience of video streaming service for large scale systems. For example, Wang *et.al*. proposed R^2 [6], a random push-based P2P scheme using network coding[2]. Also, Liu *et.al*. deployed a NC-based on-demand streaming scheme in a large-scaled commercial system [7], which showed the benefits of NC for multimedia streaming in a real P2P network. In wireless mesh networks, Seferoglu *et.al*. proposed a video-aware opportunistic network coding scheme across different flows [8] and Yang *et.al*. proposed a reliable NC-based streaming broadcasting scheme that focuses on reducing transmission overheads [9]. However, all these schemes are for traditional wired or wireless networks and are not suitable for VANETs, due to VANETs' unique characteristics described previously.

B. Streaming schemes for VANETs

Bucciol *et.al.* carried out a series of experiments using two vehicles under different scenarios, which proved the possibility of video streaming between moving vehicles [10]. Maurizio *et.al*. proposed a real-time video transmission scheme in vehicular networks [11]. This scheme only considers unicast sessions and heavily relies on fast and reliable feedback from receiver side, which itself is hard to be guaranteed in VANETs. These works mainly showed the possibility of video streaming in VANETs and did not consider more practical issues such as dealing with dynamically changing network density, minimizing bandwidth cost, conforming to standards for wireless access in VANETs, etc., all of which are carefully considered in this paper. Park *et.al*. proposed NCDD for emergency related video streaming in VANETs using NC [12]. In this scheme, the transmission of each vehicle is triggered by a timer set upon the reception of every new packet. Since neighbors' current reception status is not considered, the broadcasted packets are not always useful for neighboring vehicles, which decreases the bandwidth efficiency. Also due to lack of coordination between concurrent transmitting vehicles, the scheme tends to suffer from severe collisions, especially under dense vehicular traffic. On the theoretical aspect, Ye *et.al.* provided useful analytical results which demonstrate the benefits of network coding in one dimensional infinite lattice network [13].

C. SLNC based schemes

SLNC was recently proposed by Katti *et.al*. [3] to improve unicast throughput in wireless mesh networks. It is motivated from the observation that in lossy wireless links, due to channel variation during the transmission of a packet, for an erroneously received packet some symbols of it are still likely to be received correctly. By performing network coding on the granularity of symbols (usually corresponds to several PHY symbols of a modulation scheme), SLNC can gain benefits from both network coding and symbol-level diversity. Kim *et.al*. [14] proposed a cooperative transmission scheme based on SLNC to explore its use in the physical layer of multi-channel wireless networks (such as WiMAX). In our recent work, CodeOn [15], it is shown that SLNC outperforms PLNC for content distribution in VANETs. However, in this paper, we study the benefits of SLNC for LMS services in VANETs. Compared with content distribution applications, which only pursue single primary goal of high downloading rate, LMS services need to achieve multiple objectives at the same time, which raise quite different challenges and necessitate a reconsideration of the whole spectrum of design choices.

III. Problem Formulation and Assumptions

In this paper, we consider the following LMS services in VANETs. Several dedicated sources actively broadcast LMS contents (e.g., local road traffic monitoring videos) with constant streaming rate to vehicles inside an *area of interest* (AoI), which can either be a segment of highway or an urban area. As a motivating scenario, we assume a highway with bidirectional traffics. At the left end of the road, an AP is deployed, which continuously broadcasts LMS contents about its local traffic condition to all the vehicles driving towards it for providing intelligent navigation[3]. The service architecture is illustrated

[2]Without explicit illustration, network coding refers to packet-level network coding in the rest of the paper.

[3]We can imagine that many such APs are deployed along the highway; here we show a typical part of the whole system. Also, for simplicity we only consider single streaming flow in this paper.

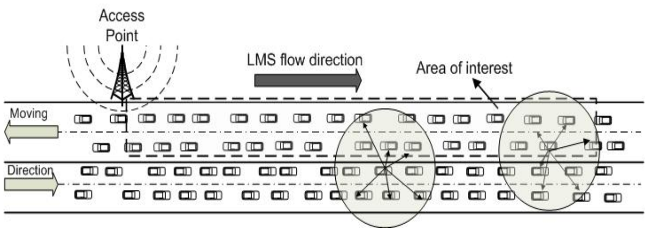

Fig. 1. The architecture for LMS.

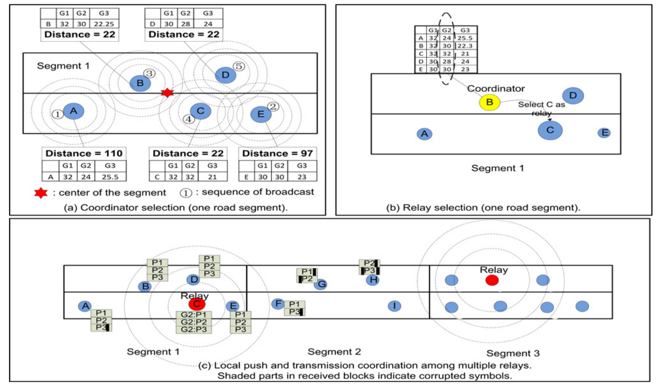

Fig. 2. The concept of coordinated local push.

in Fig. 1, where a live multimedia stream propagates against the moving direction of vehicles within AoI. We assume that the vehicles in the opposite road segment of AoI also assist the propagation of the multimedia stream, although they are not intended receivers.

According to IEEE 1609.4 standard [16], the frequency band allocated for vehicular networks is divided into multiple channels where one is reserved as control channel for safety messaging and others are used as service channels for commercial applications. However, except for a few works [15], [17], most previous schemes only assume single channel environment, either focusing on safety channel, such as [18] or commercial channel, such as [11], [12].

We assume that every vehicle is equipped with an on board unit with a wireless transceiver (single radio) which operates on multiple-channel mode. Without loss of generality, we only consider two representative channels, one control channel and one service channel, to model the coexistence of safety and commercial LMS services. According to [16], the time is divided into 100ms slots and all nodes (including vehicles and APs) are synchronized to switch simultaneously and alternatively between the control channel and service channel. How to adjust the time share of a time slot between two channels is out of the scope of this paper. For simplicity, we adopt the default allocation in [16], which splits the time slot equally.

In addition, we assume GPS device is equipped on each vehicle, given the prevalence of GPS nowadays, and also because precise time synchronization is required by IEEE 1609.4 standard for multi-channel operations. Each vehicle obtains real-time precise location (in the order of meters) information and synchronizes its clock (error smaller than 100ns). When vehicles are temporarily out of satellite coverage, they use auxiliary techniques to determine their location, and rely on their own hardware clocks.

IV. Design of CodePlay

A. Symbol Level Network Coding in CodePlay

SLNC is used throughout the design of CodePlay, and in this section we present the way SLNC actually operates in CodePlay. The source divides the original streaming content into equal-sized blocks or *generations* $G_1, G_2, G_3, G_4, ...$, each representing T seconds of playback. Every generation is again divided into K *pieces*, each of them consisting of M symbols and network coding is carried out within each single generation. K is generation size.

Each receiver v maintains a *playback buffer* for generations to be played in the immediate future, which buffers all the received useful coded symbols. Note that v also maintains a decoding matrix for each symbol position j of each generation, which consists of the coding vectors of all the j^{th} symbols it received currently. The rank of each matrix is called *symbol rank*. A coded symbol is called *useful* in CodePlay if: i) it is received correctly [3]; ii) it can increase the corresponding symbol rank (*innovative*); iii) it belongs to a generation that is after v's current playing point. When receiving enough useful symbols for a position, the receiver can decode the original symbols by performing Gaussian elimination on the corresponding matrix.

Each node plays the buffered generations sequentially and keeps eliminating older generations to make room for newer content. Those generations within α seconds after the current playback time is called *priority generations*. The piggybacked reception status which contains a priority generation with average rank less than K is considered as an implicit *urgent request*. Intuitively, urgent means the contents of those generations are mostly wanted by neighboring vehicles for the current time thus should be broadcasted with priority.

B. Overview

The primary goal of such a coordination scheme is to ensure bounded channel access delay for each vehicle, which renders most of the random based channel access schemes not appropriate. Essentially, this corresponds to the following design problem: which vehicles should transmit what content to whom at which service time slots?

The idea of CodePlay is that, *we introduce road segmentation during initialization so that the relay selection could be made locally within each segment and allow relays of adjacent segments to share the wireless channel resource in a round-robin fashion.*[4] For each time slot, a unique relay will be locally selected from all the vehicles within the same road segment based on the mechanism presented in the following section. The length of the segment, SL, is an important parameter that affects the utility of relay selection and propagation speed of the LMS flow. In general, we should ensure that for a pair of sender and receiver of distance SL, the symbol reception probability is sufficiently high. However, under realistic fading channel, it is hard to define such a range since symbol reception is probabilistic. For a

[4]We note that similar segmentation approach has been used for solving different problems in previous works [19], [20].

simpler alternative approach, we set $SL \approx CR$, an equivalent *data communication range* CR under free space propagation model(Friis) [15].

Specifically, several design choices need to be made.

1) Local Relay selection: We should ensure that only unique optimal relay could be selected within each segment for the purpose of avoiding heavy collisions. However, how to achieve this under error-prone wireless vehicle-to-vehicle communication is a challenging problem, because a node needs to know its neighbors' current reception statuses for the purpose of selecting optimal relay. Unfortunately, there is no efficient approach to frequently exchange such large amount of information reliably between nodes due to those characteristics of VANETs described in section I.

Our solution is to divide the relay selection into two steps: firstly, let vehicles within the same road segment achieve an agreement on the selection of a local "*coordinator*"; secondly, this "*coordinator*" selects the unique relay on behalf of other nodes. This is achieved by taking advantage of the obligated safety message service in the control channel required by the IEEE 802.11p standard, where every vehicle has to broadcast a safety message to inform its current location in each control time slot. CodePlay lets each vehicle piggyback a short piece of additional information in the safety message. This information contains the minimum Euclidean distance to the geographical center of the road segment that this vehicle currently knows, and also the vehicle's current LMS content reception and playback status (Fig. 2(a)). The piggybacked Euclidean distance could either be the vehicle's own distance to the center or the broadcasted distance overheard from another vehicle in the same segment. For example, in Fig. 2(a), vehicle A firstly broadcasts its safety message, thus it considers itself as the closest one to the segment center and piggybacks its distance 110 within the safety message. Vehicles E and B, which are the following ones to broadcast, will do the same as A. However, for vehicle C and D, since they overheard B's safety message and knew that B is closer to the segment center, they will piggyback B's distance in their safety messages. In this way, vehicle B, the closest to the center of the segment, will be selected as local coordinator with consensus by all the vehicles within the segment. Since the vehicle closest to the segment center will be repeatedly claimed as temporary coordinator by multiple safety messages, this accumulated consensus mechanism makes the probability of selecting multiple coordinators within one segment negligible, no matter there are lossy wireless links or sparse connections.

The coordinator selects real relay based on the reception and playback statuses of all nearby vehicles, i.e., what LMS contents each of them has received or are needed for playback in the immediate future. In particular, the coordinator computes the "*utility*" of each node in its segment as how many useful symbols can that node provides to its neighbors, and designates that node as relay via unicast. Due to the space limitations, we will not go into details about this algorithm and interested readers are referred to [2].

2) Local push and transmission coordination of relays: We have determined which vehicles should transmit what content to whom. In this section, we answer the last question: in which time slots should each relay actively push the coded LMS? This is addressed from both spatial and temporal aspects.

1. *Spatial Coordination*. Due to the use of SLNC, concurrent transmissions of more relays are encouraged to take advantage of spatial reusability [3]. But two transmitting relays that are too close will cause heavy collisions which in turn degrades the bandwidth efficiency. There exists an optimal average distance between two concurrent transmitting relays, D_{opt}, under which the relays can convey highest amount of useful information to their neighbors within unit time. However, under wireless propagation models with channel fading (such as Nakagami model), it is very hard to derive a closed form solution for Pr_{avg}. Alternatively, we get D_{opt} by Monte-Carlo simulations [15], where we find that not only SLNC have shorter D_{opt} than PLNC, but also it is quite close to *energy detection range* ER. The implications are that, by adopting SLNC, CodePlay can make the channel access decisions largely based on simple carrier sense mechanism. However, this is not the case for protocols adopting PLNC, which must deal with the well-known hidden terminal problem.

2. *Temporal Coordination*. To provide continuous streaming coverage and to satisfy the strict time constraint of LMS services, the traditional random medium access mechanisms are not appropriate since their channel access delays are not bounded. We propose to use local round-robin (LRR) scheduling to coordinate the transmissions of neighboring relays. Since it is impossible to know the inter-relay distance before those relays actually transmit, in practice, we convert D_{opt} into the number of separating segments (W_{opt}) between two adjacent concurrent transmitting relays. The observation is that, relays selected from one segment will tend to be uniformly distributed in it over time, and their average location is the segment center. As an approximation, we have $W_{opt} \times SL < D_{opt} < (W_{opt}+1) \times SL$, therefore $W_{opt} = \lfloor \frac{D_{opt}}{SL} \rfloor$. The round length R in LRR is exactly $W_{opt}+1$. For a relay in segment i, its scheduled slots T_i are determined as: $T_i = i \ mod \ (W_{opt}+1)$. For example, assume $W_{opt} = 2$, then segment 1 is scheduled to use time slots 1, 4, 7, 10, etc. Using this local round-robin schedule, LMS can flow from the source to receivers within the AoI smoothly. From a receiver's point of view, if the VANET is well-connected, it is always able to obtain new LMS content for playback within determined waiting time.

C. OLRR: Opportunistic LRR Scheduling for Sparse VANETs

Due to the highly dynamic nature of VANET, it tends to experience partitions frequently [12], especially when the traffic density is low. In sparse VANET, some road segments will be devoid of relays and the scheduled transmission opportunities would be wasted if the original LRR is adopted, which results in low bandwidth efficiency. This could be illustrated in Fig. 3(a), where the segments 4,7,10 contain no vehicles, and the scheduled transmission opportunities in this time slot for them are wasted. To solve this problem, we propose an opportunistic LRR (OLRR) scheduling algorithm by taking advantage of those available slots.

The OLRR operates in a way resembling cognitive radio, which leverages nodes' capability of carrier sensing. Essentially, during each service time slot, the coordinators in each segment will detect if there are relays in the nearby "*primary segments*", which are scheduled segments by LRR in that time slot, like segments 1, 4, 7 and 10 in time slot 1. (Fig. 3(a)). If not, certain *secondary segments* will gain channel access according to some priority assignment. Thanks to SLNC, each coordinator/relay does not need to consider the transmitters out of its energy detection capability, which greatly simplifies protocol design.

For enabling coordinators to sense the relay selection statuses of their neighboring road segments reliably and efficiently, we use a small period of time at the very begin of each service slot and divide it into $3 \times (W_{opt} + 1)$ subslots. Coordinator of road segment i that could find a relay will broadcast a short signal during the subslot $i \ mod \ 3 \times (W_{opt}+1)$ to notify other neighboring coordinators which will keep sensing the channel during those subslots. We note that the actual data transmissions start after those subslots. The reason we need $3 \times (W_{opt} + 1)$ subslots is to ensure that each coordinator will be able to determine a unique segment (w.h.p) that is transmitting in each subslot (cause for each subslot, there could be 2 possible notification signals broadcasted from both side). Since the sensing process is purely based on detecting the energy, the time overhead can be negligible. In CodePlay, we set the sensing signal length to be 50 bytes and the length of each sub-slot to be $100\mu s$, which takes preamble, SIFS, etc. into consideration. For $W_{opt} = 2$, the total extra time is $3 \times (2 + 1) \times 100 = 900\mu s$, which is less than 2% of a service time slot with length of $50ms$. The algorithm is described in Alg. 1. In line 3, there are two cases where a relay cannot be selected: C_i is the only node in i, or no node can provide innovative information to others. $ConflictSet(i)$ is the set of coordinators (also segments) that has higher transmission priority than i. The nearer a segment is to a primary segment (with lower ID), the higher its priority. If two secondary segments happen to have the same distance to their primary segments, they will both access the channel as is the case in LRR.

We use the examples in Fig. 3 to illustrate the basic idea of OLRR. Suppose $R = 3$ and C_1, C_4, C_7, C_{10} are scheduled to use the channel simultaneously in the current service time slot $T = 1$. In Fig. 3(a), if we apply original LRR, only C_1 will use this service time slot. If we apply OLRR, C_5 will decide to take this time slot since it senses that C_4 and C_7 do not exist. The same for C_8 and C_{11} and thus this service time slot will be consumed by C_1, C_5, C_8 and C_{11} simultaneously which obviously improves the spatial reusability. For VANET snapshot shown in Fig. 3(b), if OLRR is adopted, although C_8 will give up this opportunity, since otherwise it will incur unnecessary heavy interference to the transmission of C_{10}, C_5 still could use this service time slot along with C_1 and C_{10}. The operation of OLRR under the situation shown in Fig. 3(c) is a little more complicated. Now both *secondary segments* C_6 and C_8 will try to take the extra transmission opportunities left by empty segments 4 and 7 respectively. To avoid heavy collision between them, OLRR assigns each secondary segment a priority based on its distance to the primary segment with lower ID. In this case, C_6 is two segment away from the primary segment 4 and C_8 is only 1 segment away from the primary segment 7, thus C_8, which has higher priority, will take this transmitting opportunity and C_6 will keep silent during this service time slot.

Algorithm 1 Opportunistic LRR scheduling at each coordinator (at the beginning of a service channel slot)

1: **Input**: Segment ID i, coordinator C_i, round length $R = W_{opt} + 1$
2: **Output**: Whether to allow the relay access channel
3: If C_i is able to select a relay from i
4: Broadcast a short signal in the subslot $i' \leftarrow i \ mod \ 3R$ $ConflictSet(i) \leftarrow \emptyset$
5: For subslot j' from 0 to $3R-1$//determine which segments have relays
6: If sensed signal during j'
7: $ConflictSet(i) \leftarrow ConflictSet(i) \cup C_{j'}$, $C_{j'} \in Segment\ j$, where $Segment\ j$ is the nearest one to i between the two: $j' + i - i'$ and $j' + i - i' \pm 3R$ //the most probable segment
8: Prune from $ConflictSet(i)$ the segments that are more than R segments away from i //regarded as not conflicting
9: Prune from $ConflictSet(i)$ segments j with $j \ mod \ R > i \ mod \ R$ //the one nearer to a primary segment has higher priority
10: If $ConflictSet(i) \neq \emptyset$
11: C_i tells relay in i to abort transmission
12: Else, C_i tells relay in i to access the channel in current service time slot

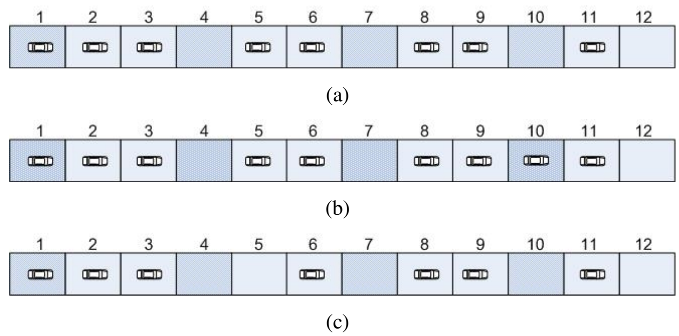

Fig. 3. Snapshots of 3 sparse VANETs (T=1, R=3). The road segment ID is illustrated above each road segment and the vehicles represent corresponding coordinators. Those dark shaded segments in each snapshot are designated to be scheduled in this time slot.

V. Performance Evaluation

We implemented and evaluated CodePlay by simulations using NS-2.34. The SLNC is implemented based on [3], with an enhanced run-length coding technique [15] which is more suitable for consecutively broadcasting a generation of coded pieces in CodePlay. To ensure unique coordinator selection within the same segment, at the beginning of service time slots use an additional broadcast round (shorter than 1ms) to resolve collisions between potential coordinators. The simulation scenario consists of a straight 4-lane highway with length of 3000 meters, and two LMS sources (e.g., access points) that are located at both ends of the highway, separately [5]. The upper part of the highway (west bound) is regarded as the AoI. We simulate both dense and sparse VANETs by using two

[5]Cause no existing schemes could support smooth video stream with single source due to significant throughput degradation after multi-hops in wireless networks, we do not show the evaluation result under this scenario and interested readers could refer to our conference paper [2] for more details.

TABLE I
PARAMETER SETTINGS

Data rates for LMS and safety msg.	12Mbps, 3Mbps
Data communication range	$CR = 250$m
Time per generation, piece size	2s, 1KB
Safety message length (with piggyback)	130B
Buffer capacity	15 generations
$PriValue$	32
# of generations in priority region	$\alpha = 1$
Default D_{opt} for CdePlay+SLNC	900m
Default D_{opt} for CdePlay+PLNC	1200m

traffic densities: 100 cars/km and 40 cars/km. The vehicular speeds are randomly selected from 20-30 m/s. The simulation parameters are shown in Table I.

The protocol for comparison is the PLNC version of CodePlay (CodePlay+PLNC) and the W_{opt} for PLNC is used. By default, the OLRR is applied for all the protocols. The most related state-of-the-art LMS scheme to ours is emergency video dissemination in VANETs using PLNC (NCDD, [12]). However NCDD was not designed to meet the practical application layer requirements defined in this paper, and it is hard to evaluate those metrics based on NCDD protocol. Thus we chose not to implement NCDD, but compare our results with the reported ones in [12]. Each point shown in the simulation results is averaged over 10 runs.

The performance of CodePlay is evaluated by multiple metrics: (1) Initial buffering delay, which is the user experienced service delay. In the simulation, we impose the same initial buffering delay for all receiving vehicles. (2) Source rate, which reflects the supported LMS generation rate from the application layer. (3) Skip ratio, the fraction of generations skipped due to incomplete reception before playback time over all the generations that are played. We note that this will make the skip ratio looks higher since any missing bit will render the whole generation useless. However, in practice, the LMS content usually does not need to achieve 100% reception ratio for playback. The playback video quality will gradually improved as the increase of the reception ratio by applying advanced video coding methods like multi-layer coding and multi-description coding. Since the adoption of those advanced video coding approaches is orthogonal to our work, we just use this all or nothing rule for simplicity in this paper. (4) Buffering level, the percentage of the buffered LMS contents between current playback time and physical world time. Both the skip ratio and buffering level could reflect the playback quality, i.e., smoothness [6].

A. Effect of D_{opt}

To see the impact of different D_{opt} on the performance, we run the protocols with varying optimal distances between adjacent concurrent transmitting relays under dense highway scenario. Specifically, 400m, 750m, 900m and 1200m are used. Since the segment length $SL = 250m$ in our simulations, these D_{opt} values represent the cases where the lengths of the round robin are 2, 3, 4 and 5 respectively. The result is shown in Fig. 4. We can see that the playback skip ratio for CodePlay+SLNC gradually decreases as the D_{opt} increased from 400m and achieves the minimum when the D_{opt} equals 900m, which also stands for the optimal playback

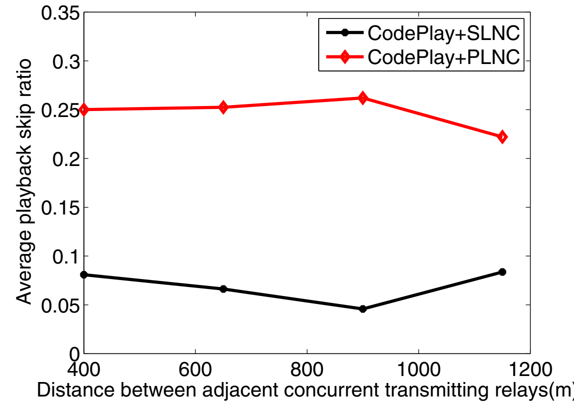

Fig. 4. Comparison between different distances of adjacent concurrent transmitting relays, dense highway, source rate=12KB/s, initial buffering delay=24Sec.

video quality. After that, the skip ratio goes up again as the D_{opt} increased. For CodePlay+PLNC, the average playback skip ratio decreases continuously as the increase of the $Dopt$. Besides, we also can see that under $D_{opt} = 1200m$, Although the skip ratio of CodePlay+SLNC increases from 5% to about 8%, it is still much better than the corresponding performance of CodePlay+PLNC, which again demonstrates the benefits of SLNC over PLNC.

B. Effect of Initial Buffering Delay

To illustrate the advantage of CodePlay in providing better LMS services under various VANET scenarios, we investigate the relationship between initial buffering delay, source rate and the metrics for smooth playback under a relatively sparse highway scenario. In the first simulation set, we fix initial buffering delay as 16 seconds, and increase the source rate from 24 KB/s to 36 KB/s, the result of which is shown in Fig. 5. We can see that the skip ratio for CodePlay+SLNC is much lower that its PLNC based opponent, where the former's skip ratio is 0 under 24 KB/s, 5.3% under 30 KB/s and 15% under 36 KB/s. This suggests that rate no greater than 30KB/s could be supported without affecting smooth playback. Also, for each rate CodePlay+PLNC's buffering level decreases faster over time, and is less stable compared with that of CodePlay+SLNC. This reflects that CodePlay+SLNC achieves a more stable flow of multimedia streaming, which shows the effectiveness of the integration of SLNC with the coordinated local push mechanism. We note that, the NCDD protocol only provided 10 KB/s source rate for video dissemination [12].

Another interesting observation when we look into details of the performance is the changing of the average reception ratio for those skipped generations, which is defined as the ratio between the average symbol rank over all symbol positions within the generation and the generation size. We find that the CodePlay+SLNC not only achieves lower skip ratio, but also achieves much higher average reception ratio compared with CodePlay+PLNC. For example, the average reception ratio for the former is always higher than 90% while that of the CodePlay+PLNC is no greater than 60% for both 30KB and 36KB/s cases (in the 24KB/s case, the reception ratio for CodePlay+SLNC is 0 due to no skipped generation). This

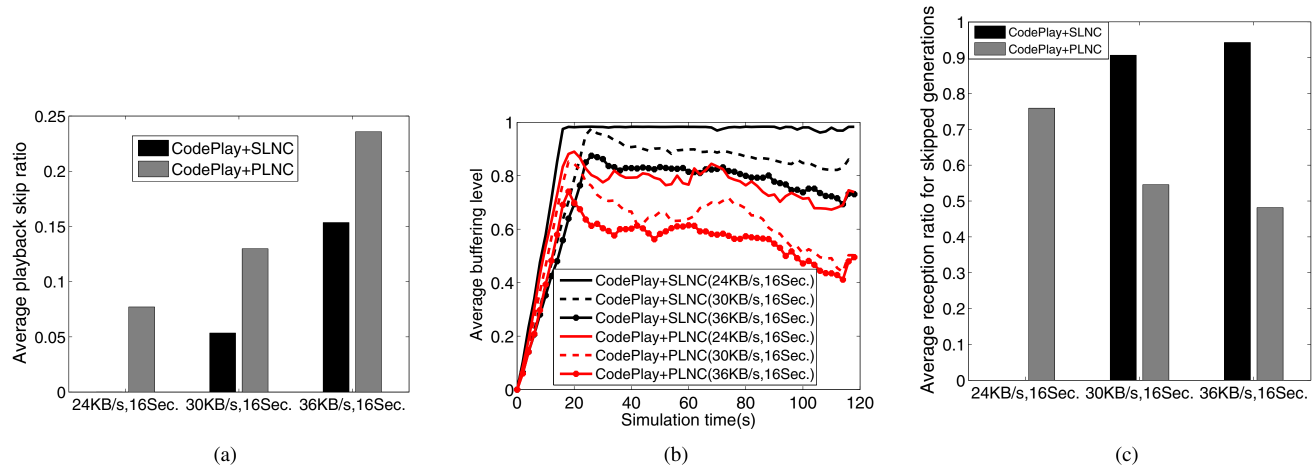

Fig. 5. Fixed initial buffering delay, varying source rates. Sparse highway.

means those skipped generations in CodePlay+SLNC contain more useful information and if advanced video quality criterion rather than the simple all-or-nothing policy is adopted, the advantage of SLNC based scheme will be enhanced. We also notice that as the increase of the source rate, the average reception ratio for CodePlay+PLNC gradually decreases while that of the CodePlay+SLNC is relatively stable. This is because that higher source rate brings more communication and thus incurs more transmitting contention into the network, which will significantly affect the probability of correct reception of the whole packet. On the other hand, the SLNC based scheme will accumulate all the correctly received symbols even for an error packet thus the impact of the higher contention is greatly alleviated.

The CodePlay+SLNC works well through all source rates no greater than 30 KB/s, and for buffering delays of 16s and 24s. We argue that those delays are acceptable in VANETs. For example, for delay equals to 16s and vehicular velocity of 30m/s, a car will travel about 500m after it enters the AoI to begin playing an emergency multimedia content. For $L =$ 3000m, the car will be at 2500m from the accident spot and may still have enough time to take actions.

C. Effect of Traffic Density

Next we study the performance of CodePlay under the dense traffic condition with various source rates. Though CodePlay+SLNC still outperforms CodePlay+PLNC, compared with the sparse case, the skip ratio of both protocols are higher and corresponding buffering levels are lower. Especially, only the skip ratio of CodePlay+SLNC under 12KB/s could be kept lower than than 5%, and the skip ratios of all the other cases are higher than 18% which could be unacceptable from receivers' point of view. The worse performance can be mainly ascribed to the fact that in a dense VANET, since there could be too many vehicles urgently demanding some LMS content, it is intrinsically hard to satisfy all their needs in a short time with wireless broadcasting which is error-prone in nature. Due to the time constraints of LMS applications, this leads to more frequent playback skips than in the sparse VANETs.

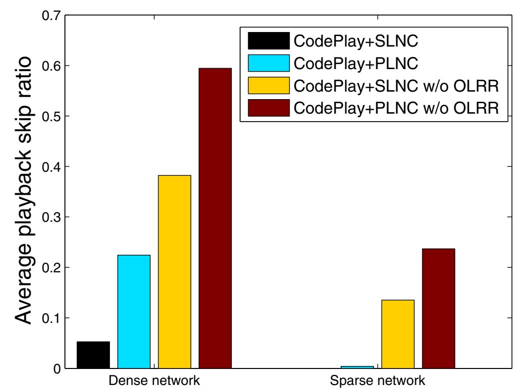

Fig. 6. Effect of opportunistic transmission scheduling.

D. Effect of Opportunistic Scheduling

In the previous simulations, we have the OLRR scheduling enabled by default. Yet it is interesting to see how the opportunistic scheduling affects the protocol performance. Thus, we present in Fig. 6 the results of enabling and disabling the OLRR algorithm (using LRR instead). All the protocols run with source rate of 12 KB/s under dense network and 24 KB/s under sparse network, the initial buffering delays of both are 16 Sec. We can see that the OLRR greatly improves the performance over the basic LRR algorithm for all the running cases. By opportunistically utilizing the idle scheduled transmission slots left by primary segments, the OLRR can adaptively "fill" the unnecessary gaps created during the propagation of the LMS flow. We note that OLRR could take effect not only under sparse network, but also under dense network. The reason is that according to [1], the vehicles running on the highway tends to form disjoint clusters rather than uniformly distributed even under relatively dense traffic.

E. Bandwidth efficiency

In the end of the performance evaluation, we would like to show the benefit of CodePlay in terms of bandwidth efficiency, which is always important in extreme network scenarios like VANETS. For easy comparison purpose, we borrow the definition of normalized packet overhead from [12], which refers to the total number of packets transmitted to the channel divided by the total number of data packets delivered. [12] shown that the normalized packet overhead for NCDD is about 0.2 under network setting where 200 vehicles scattered within 10km highway and source traffic rate is 10KB/s, compared with On Demand Multicast Routing Protocol (ODMRP) [21]'s much higher overhead of 0.7, which demonstrate the benefit of adopting PLNC technique. However, in a similar sparse highway setting with 120 vehicles scattered within 3km area and a little bit higher source rate (12KB/s), CodePlay could achieve even lower normalized packet overhead of 0.09. This advantage can be ascribed to both the finer granularity of SLNC technique which ensures the usefulness of each symbol of the transmitted packet, and also the protocol design of CodePlay which allows concurrent transmitters to share the bandwidth resource in the most efficient way.

VI. Conclusions

In this paper, we presented the design and performance evaluation of CodePlay for live multimedia streaming in the dynamic and lossy VANETs. Multiple objectives are pursued at the same time, including short buffering delay, smooth playback, and high source rate. The core of CodePlay is a coordinated local push mechanism with symbol level network coding, which establishes local and distributed coordination among vehicles to ensure stable and high streaming rates. Through the above mechanisms, the benefits of SLNC are fully exploited for better LMS performance in VANET. Our main conclusion in this paper is that symbol-level network coding is a good technique to support bandwidth consuming, delay constrained LMS applications in extreme environments like VAENTs. Even using SLNC, we may still need the help of few additional infrastructure (APs) along the road and well designed channel usage mechanisms to facilitate the dissemination of LMS content to end users.

Acknowledgment

This work was supported in part by the US National Science Foundation under grants CNS-1156311 and CNS-1156318.

References

[1] N. Wisitpongphan, F. Bai, P. Mudalige, V. Sadekar, and O. Tonguz, "Routing in sparse vehicular ad hoc wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 25, no. 8, pp. 1538–1556, Oct. 2007.

[2] Z. Yang, M. Li, and W. Lou, "Codeplay: live multimedia streaming in VANETs using symbol-level network coding," in *Proc. 2010 IEEE International Conference on Network Protocols*, pp. 223–232.

[3] S. Katti, D. Katabi, H. Balakrishnan, and M. Medard, "Symbol-level network coding for wireless mesh networks," in *Proc. 2008 SIGCOMM*, pp. 401–412.

[4] K. Jamieson and H. Balakrishnan, "PPR: partial packet recovery for wireless networks," in *Proc. 2007 SIGCOMM*, pp. 409–420.

[5] R. Ahlswede, N. Cai, S.-Y. Li, and R. Yeung, "Network information flow," *IEEE Trans. Inf. Theory*, vol. 46, no. 4, pp. 1204–1216, July 2000.

[6] M. Wang and B. Li, "R2: random push with random network coding in live peer-to-peer streaming," *IEEE J. Sel. Areas Commun.*, vol. 25, no. 9, pp. 1655–1666, Dec. 2007.

[7] Z. Liu, C. Wu, B. Li, and S. Zhao, "Uusee: large-scale operational on-demand streaming with random network coding," in *Proc. 2010 IEEE INFOCOM*.

[8] H. Seferoglu and A. Markopoulou, "Video-aware opportunistic network coding over wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 27, no. 5, pp. 713–728, June 2009.

[9] Z. Yang, M. Li, and W. Lou, "R-code: network coding-based reliable broadcast in wireless mesh networks," *Ad Hoc Networks*, 2010.

[10] P. Bucciol, E. Masala, N. Kawaguchi, K. Takeda, and J. De Martin, "Performance evaluation of h. 264 video streaming over inter-vehicular 802.11 ad hoc networks," in *Proc. 2005 PIMRC*, pp. 1936–1940.

[11] M. Bonuccelli, G. Giunta, F. Lonetti, and F. Martelli, "Real-time video transmission in vehicular networks," in *Proc. 2007 Mobile Networking for Vehicular Environments*, pp. 115–120.

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

[13] F. Ye, S. Roy, and H. Wang, "Efficient data dissemination in vehicular ad hoc networks," *IEEE J. Sel. Areas in Commun.*, vol. 30, no. 4, p. 769, 2012.

[14] R. Kim, J. Jin, and B. Li, "Drizzle: cooperative symbol-level network coding in multichannel wireless networks," *IEEE Trans. Veh. Technol.*, vol. 59, no. 3, pp. 1415–1432, Mar. 2010.

[15] M. Li, Z. Yang, and W. Lou, "Codeon: cooperative popular content distribution for vehicular networks using symbol level network coding," accepted by *IEEE J. Sel. Areas Commun.*, 2010.

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

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

[18] Q. Xu, T. Mak, J. Ko, and R. Sengupta, "Vehicle-to-vehicle safety messaging in dsrc," in *Proc. 2004 VANET*, pp. 19–28.

[19] M. Li, W. Lou, and K. Zeng, "Oppcast: opportunistic broadcast of warning messages in VANETs with unreliable links," in *Proc. 2009 IEEE MASS*.

[20] M. Johnson, L. D. Nardis, and K. Ramch, "Collaborative content distribution for vehicular ad hoc networks," in *2006 Allerton Conference on Communication, Control, and Computing*.

[21] S.-J. Lee, M. Gerla, and C.-C. Chiang, "On-demand multicast routing protocol," in *Proc. 1999 IEEE Wireless Communications and Networking Conference*, vol. 3, pp. 1298–1302.

Zhenyu Yang (S'08-M'11) earned a B.E and an M.E in Computer Science both at Xi'an Jiaotong University, China, in 2004 and 2007, respectively and a Ph.D. in Electrical and Computer Engineering department at Worcester Polytechnic Institute in 2011. He then joined Amazon.com and his current research interests are in the areas of cloud computing, wireless networks and network security, with emphases on network coding and protocol design.

Ming Li (S'08 - M'11) is an assistant professor in the Computer Science Department at Utah State Univ. He earned his B.E and M.E both in Electronic and Information Engineering from Beihang Univ. in China, and received his Ph.D. in Electrical and Computer Engineering from Worcester Polytechnic Institute in 2011. His current research interests are mainly in cyber security and privacy, with emphases on data security and privacy in cloud computing, security in wireless networks and cyber-physical systems. He is a member of IEEE and ACM.

Wenjing Lou received her Ph.D. in Electrical and Computer Engineering at the University of Florida. She joined the department of Electrical and Computer Engineering at Worcester Polytechnic Institute as an assistant professor in 2003, where she was promoted to associate professor with tenure in 2009. She then joined the Computer Science department at Virginia Polytechnic Institute and State University in 2011 as associate professor with tenure. Her current research interests are in cyber security, with emphases on wireless network security and date security and privacy in cloud computing. She was a recipient of the U.S. National Science Foundation Faculty Early Career Development (CAREER) award in 2008.