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

Zhenyu Yang, Ming Li and Wenjing Lou
Department of ECE, Worcester Polytechnic Institute, Worcester, MA 01609
{zyyang, mingli, wjlou}@wpi.edu

***Abstract*—Live multimedia streaming (LMS) services are important in vehicular ad hoc networks (VANETs) for their capability of providing comprehensive and user-friendly information. The fundamental challenges 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. A recent technique, *symbol-level network coding* (SLNC), has been shown to be an effective approach to 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. CodePlay is designed to simultaneously improve the performance of LMS service in terms of streaming rate, service delivery delay and bandwidth efficiency. We use extensive simulations to show that CodePlay is potentially suitable for future LMS applications in VANET.**

I. Introduction

A lot of research efforts have been focusing on the vehicular communications during the last decade for its capability of supporting many interesting applications, varying from safety warning messaging [1], content distribution [2]–[4] to Internet access on the road, etc. Among all these applications, live multimedia streaming (LMS) is particularly promising, because it consists of video, audio and can provide more precise, comprehensive and user friendly information than plain text based applications. Typical scenarios for LMS applications may include the following. 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. Also, when a police vehicle spots an accident, it disseminates emergency-related LMS content of the accident to vehicles following several miles behind for early warning. Then the paramedics can also make preparation more purposefully in advance based on the collected LMS content on their way to the accident scene.

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.

Different from non-streaming services, such as content distribution [2], where the main focus is on the average downloading rate, LMS services require not only high average streaming rate but also demand the streaming rate keep stable for the purpose of smooth playback. LMS services are also different from non-live streaming services such as video-on-demand, where various vehicles maybe interested in different contents and those contents are not closely related to real world's time. For LMS services, the streaming contents are usually generated as time progresses and only useful to vehicles within a short period of time, e.g., several to tens of seconds. However, these time constraints are usually not as tight as those of real-time services, like intelligent collision avoidance, which usually requires delay smaller than hundreds of milliseconds.

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 and it is very challenging to achieve them simultaneously, especially considering VANETs' specific characteristics. 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 VANET. 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 acquire precise and in-time neighbor information (such as reception status). However, under VANET with ever-changing topology, this learning process may lead to high communication overhead. Thirdly, VANETs tend

978-1-4244-8645-8/10/$26.00 ©2010 IEEE

to experience frequent partitions, 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 sum, all these factors make it hard to ensure smooth playback while keeping low bandwidth consumption.

Most existing works [5]–[8] on live multimedia streaming focused on traditional wired or wireless networks, where either the links are reliable or the topology is relatively stable over time. Many of these works adopted network coding (NC) [9] which has been shown to be an effective approach to improve the network bandwidth efficiency and simplify the protocol design for LMS service in those networks. On the other hand, only a few works [10] have applied NC to providing LMS services in VANETs. However, the gain of NC tends to be offset by severe packet collisions due to lack of proper transmission coordination mechanism among vehicles. To the best of our knowledge, none of existing works can well satisfy all the requirements simultaneously.

In this paper, we propose CodePlay, a distributed live multimedia streaming scheme in VANETs based on symbol-level network coding (SLNC) [11]. 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. 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 a coordinated local push mechanism to fully exploit the benefits of SLNC in VANETs. In order to disseminate the streaming content from sources to all the receivers smoothly and in a timely manner, 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. The concurrent transmissions of all relays are coordinated locally according to optimal schedules, to provide continuous streaming coverage for the whole VANET.
- To enhance the LMS performance for sparse VANETs, we proposed an opportunistic transmission scheduling algorithm, where the wasted transmission opportunities in the network can be adaptively utilized by the relays, merely based on carrier sensing.
- We implemented CodePlay in NS-2 and carry out extensive simulations to evaluate its performance. We showed both the feasibility and the constraints of providing LMS services in VANETs. Compared with using PLNC, CodePlay with SLNC can provide more and better design choices to network architects.

II. Related work

Currently, streaming services are widely deployed in the Internet, such as PPLive, PPStream, etc. In particular, network coding (NC) 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 [5], a random push-based P2P scheme using network coding[1]. 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 [6]. However, all these schemes are for traditional wired or wireless networks and are not suitable for VANETs, due to VANETs' unique characteristics described previously.

[10], [12]–[16] proposed several schemes on supporting various kinds of streaming services in VANETs, which can be divided into two categories:

(1) Schemes focusing on application layer. Maurizio *et.al.* proposed a real-time video transmission scheme in vehicular networks [14]. This scheme only considers unicast sessions and heavily relies on fast and reliable feedback from receiver side. Bucciol *et.al.* carried out a series of experiments using two vehicles under different scenarios, which proved the feasibility of video streaming between moving vehicles [16]. Qadri *et.al.* showed that by adopting error resilience coding, state-of-the-art routing protocols can support multicast video streaming in city VANETs when the network is not dense [15]. These works mainly showed the feasibility of video streaming in VANETs and have not considered more practical issues such as dealing with dynamically changing network density, minimizing bandwidth cost, etc., which are carefully considered in our paper.

(2) Schemes focusing on network and MAC layer. Park *et.al.* proposed NCDD for emergency related video streaming in VANETs using NC [10]. 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 nodes' neighbors, 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.

Soldo *et.al.* introduced SMUG, a TDMA-based scheme to support streaming media dissemination in city VANETs [13]. A tree structure is established for broadcasting streaming video content. However, it is hard to maintain a stable and up-to-date communication structure for dynamic VANETs, thus stable streaming rate is difficult to achieve. In [12], Guo *et.al.* proposed $V3$, a live video architecture for VANET, where directed broadcast is adopted for remote video request scenarios, which are different from the application in this paper.

SLNC was recently proposed by Katti *et.al.* [11] to improve unicast throughput in wireless mesh networks. It is motivated

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

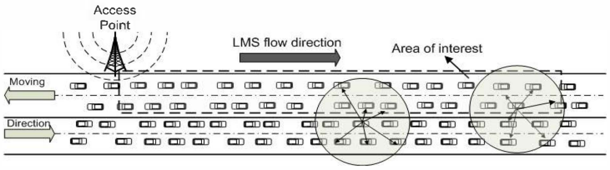


Fig. 1: The architecture for LMS.

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. In our recent work, CodeOn [2], it is shown that SLNC outperforms PLNC for content distribution in VANETs, in terms of downloading rate. However, in this paper, we study the benefits of SLNC for LMS services in VANETs. Compared with file downloading applications, which only pursue single primary goal, LMS service needs to achieve multiple objectives at the same time, which raises quite different challenges and necessitates a reconsideration of the whole spectrum of design choices.

III. Problem Formulation

A. Model 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[2]. The service architecture is illustrated 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 [17], the frequency bands 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 [2], [18], most previous schemes only assume single channel environment, either focusing on safety channel, such as [1] or commercial channel, such as [10], [12].

We assume that every vehicle is equipped with an on board unit with a wireless transceiver (single radio), and 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 [17], the time is divided into 100ms

[2]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.

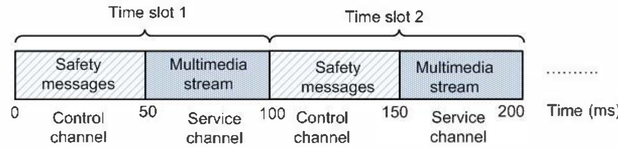


Fig. 2: The channel and time utilization for each AP and each vehicle.

slots and all nodes (including vehicles and APs) are synchronized to switch simultaneously and alternatively between the control channel and service channel. The allocation of channels and time slots is depicted in Fig. 2. How to adjust the time share between two channels is out of the scope of this paper. For simplicity, we adopt the default allocation in [17], which splits the time slot equally.

In addition, we assume GPS device is equipped on each vehicle, given the prevalence of GPS nowadays, and 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.

B. Objectives

The design of CodePlay pursues the following primary objectives.

- Smooth playback at all the interested vehicles, which refers to all vehicles inside AoI. This requirement can be translated into providing stable and high streaming rate.
- Prompt service delivery, which can be translated into short end-to-end delay for all the receivers. For a receiver, this delay is defined as the elapsed time from the generation of specific LMS content at the source to the start of playback of this content at the receiver. Meanwhile, it is desirable to achieve high degree of fairness, i.e., the service delays among neighboring receivers should be similar.
- Minimized bandwidth cost, which can be translated into incurring small protocol overhead and data traffic. This is for better coexistence with other possible services.

IV. The Design of CodePlay

A. Design Rationale of CodePlay

1) Push-based Network Coding is Good for LMS: Most LMS schemes adopting store-and-forward communication paradigm are pull-based, where each receiver sends explicit requests to other nodes for retrieving the missing contents. These schemes inclines to suffer from low bandwidth efficiency in VANETs due to high protocol overheads and dependence on TCP-based content retrieving [5], which is well-known for low efficiency in lossy wireless networks [19].

NC is a new communication paradigm originated from [9] and has gradually been accepted as a promising approach to improve the bandwidth efficiency during the last decade [5], [6], [10], [19], [20]. The core idea of NC is to give each node the flexibility of encoding different received packets, which breaks the store-and-forward routine. Besides improving the bandwidth efficiency, another important benefit of NC is that: any coded packet is as good as others, regardless of the node that generates them, which can greatly simplify the protocol design. By exploiting NC, several push-based multimedia streaming schemes are proposed recently, which showed better overall performance in various scenarios than pull-based schemes.

The design of CodePlay is partially inspired by the following two works. R^2 [5], a push-based peer-to-peer LMS scheme for Internet, where seeds actively push coded packets to downstream peers without the need of costly requests and collaboration. NCDD [10], a push-based scheme for emergency related video streaming in VANETs, which constraints the content retrieving process within one hop, while not suffering from scarcity of useful neighbors due to the use of NC. Both schemes apply UDP-based content retrieving, in sharp contrast to the traditional pull-based schemes adopting TCP-based communications.

2) SLNC Potentially Performs Better than PLNC in VANETs: Although NC benefits, the PLNC technique suffers from unnecessary performance degradation in VANETs. In PLNC, a small portion of the packet which is not received correctly will render the whole packet useless, and this happens frequently under the lossy wireless medium in VANETs. SLNC, however, by operating on smaller symbols and thus benefiting from both symbol-level diversity and NC, can potentially achieve higher bandwidth efficiency than PLNC. And this has been shown in content distribution in VANETs by [2].

However, to provide satisfiable LMC services using SLNC in a dynamic and lossy VANET, the biggest challenge is how to achieve multiple objectives (stable high streaming rate, small service delivery delay and minimal bandwidth consumption) simultaneously. Essentially, this corresponds to the following design problem of CodePlay: **which vehicles should transmit what content to whom at which service time slots?** In particular, since broadcast is adopted as the basic transmission paradigm, and multiple receivers may have different stream reception and playback statuses, how do we select proper relay nodes to ensure smooth playback of multiple vehicles? How to coordinate the transmission of multiple relays so that spatial reusability is maximized? How to efficiently achieve the above with small overhead? All these key issues imply that wholly new design considerations are needed.

3) Make All Ends Meet — Coordinated Local Push with SLNC: Corresponding to the above issues, our solution is a *coordinated local push* (CLP) mechanism based on SLNC, which mainly consists of two parts: distributed relay selection and transmission coordination of relays. The core idea is as follows. In each service time slot, a set of spatially separated relay nodes are dynamically and distributively selected. By actively pushing coded LMS contents in the locality, each relay node can provide most useful information to its neighbors so that their collective and individual needs for smooth playback can be both well satisfied. In space, the transmission of all the relays are coordinated in a way that maximizes the overall streaming rate, by exploiting the increased spatial reusability enabled by SLNC. In time, the relays belonging to consecutive road segments are scheduled in a round-robin fashion, so as to propagate the multimedia stream continuously throughout the network to reduce the end-to-end delay.

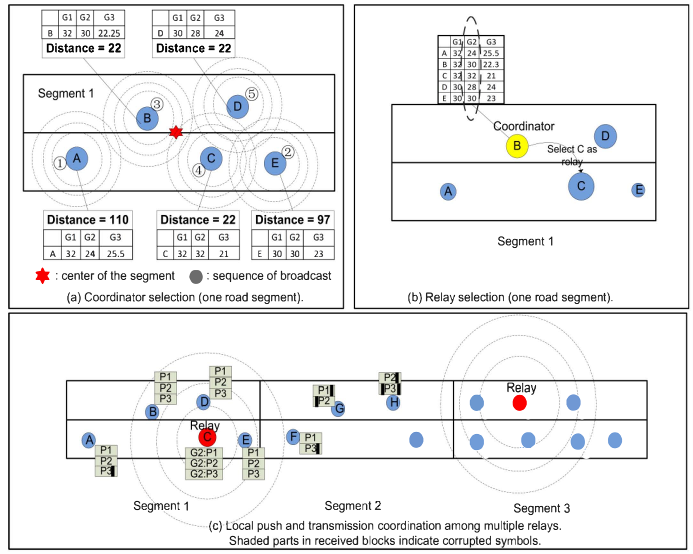


Fig. 3: The concept of coordinated local push.

B. Design overview

We illustrate the core design of CodePlay using a simple example. For proof-of-concept, in this paper, we describe CodePlay under a one-dimensional highway scenario (Fig. 3). However, CodePlay can be easily extended to the urban scenario, i.e., two-dimensional case.

1. *System initialization*. To ensure smooth playback in LMS, node coordination shall be facilitated in an efficient way, yet, in a dynamic VANET. The idea of CLP is that, *we introduce road segmentation during initialization so that coordination decisions can be made locally*[3]. Each road is divided into fixed segments of equal length (*SL*) and is uniquely numbered, which can be pre-configured and provided by the access points with the help of GPS. And every vehicle is assumed to possess this information before entering the AoI.

2. *Local coordinator selection*. In order to make the relay selection process reliable and efficient, the key method is to let vehicles agree on a local "*coordinator*" who selects the relay on behalf of other nodes (Fig. 3 (a)). 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 on the safety message. This information contains the minimum Euclidean distance to the geographical center of the road segment that this vehicle knows (either its own distance to the center or the broadcasted distance overheard from other vehicles in the same segment), and also the vehicle's current LMS content reception and playback statuses. We will introduce an efficient representation of this information later. The vehicle closest to the center of the segment is selected as local coordinator, like vehicle B in Fig. 3 (a).

[3]We note that similar segmentation approach has been used for solving different problems in previous works [21], [22].

3. *Distributed relay selection*. The coordinator selects real relay based on the reception and playback statuses of all nearby vehicles, i.e., what LMS contents each of them have 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 much useful information can that node provide to its neighbors, and designates that node as relay via unicast. This is shown in Fig. 3 (b), where coordinator B designates vehicle C as relay and the generation G_2 as the broadcasted content. One generation represents a short period of LMS content and the precise definition will be given in the following section.

4. *Local push and transmission coordination of relays*. In order to create a stable and continuous LMS flow, only relays in certain segments are allowed to transmit concurrently in each service time slot. Those relays actively "push" coded LMS blocks to their vicinity, which will be received by neighboring vehicles. To maximize spatial reusability, we exploit SLNC's symbol-level diversity by purposely reducing the distance between two concurrent transmitting relays (thus introducing a proper amount of signal interference). In the snapshot given in Fig. 3 (c), the two relays are separated by two road segments, which maybe too close to be allowed if packet level collision avoidance mechanism is adopted. Specifically, we address the following issues: i) what is the optimal number of segments between two adjacent transmitting relays? ii) how can we opportunistically schedule the relays' transmission if the density of the VANET is so sparse that some road segments are empty and no relay could be selected for them?

C. LMS Using Symbol Level Network Coding

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. K is also called generation size. SLNC is carried out within each generation to reduce the decoding complexity. At the source, the j^{th} symbol (at jth position) $\mathbb{s}'_j$ in a coded piece is a random linear combination of the j^{th} symbols of all the K original pieces within the generation:

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

where $\mathbb{s}_{ji}$ is the jth symbol in the ith original piece, and $\mathbb{v} = (v_1, ..., v_K)$ is called the *coding vector* of this coded symbol, each element of which is randomly chosen from a Galois field $\mathbb{F}_{2^q}$. The coding vector, which is shared by all the coded symbols, will be transmitted along with the coded piece for the purpose of decoding. The coding process at a relay node is a little different, since the correctly received symbols may be in positions not consecutive due to packet corruptions. To reduce the overhead incurred by potential multiple coding vectors, we adopt piece-division, run-length coding algorithm [2], where consecutive clean symbols share a coding vector. By using SLNC, the bandwidth efficiency of each coded transmission could be improved.

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. The rank of each matrix is called *symbol rank*. A coded symbol is called *useful* in CodePlay if: i) it is received correctly [11]; 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.

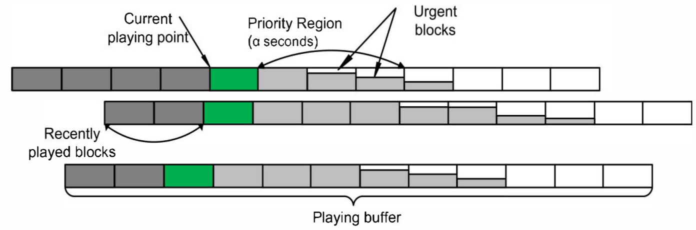


Fig. 4: playback buffer and priority generations.

For nodes to make decisions on transmission, coding and coordination, every node needs to disseminate its *reception status* to neighbors, i.e., symbol rank of each useful generation. Although this information is piggybacked on periodical safety messages which add no extra overhead to the service channel, we still should keep it minimal to reduce its impact on the reliability of safety messages. Here is a back-of-the-envelope calculation: for 10 generations with packet length of 30 symbols, the piggybacked information is 300 bytes, which is obviously too long for a safety message. To decrease the size of it, we use the fuzzy *average rank*. That is, for generation G_i, an average rank value $\lfloor \bar{r}_i \rfloor$ across all symbol positions is computed and transmitted. Now, the piggybacked information is only 10 bytes, which can be easily embedded in a safety message without affecting its reliability [1].

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*. The above definitions are depicted in Fig. 4. Note that vehicles on the opposite road of the AoI behaves exactly the same as described above, except that they do not need to playback the received LMS contents.

D. Coordinated and Distributed Relay Selection

The main purpose of the relay selection is to maximize the utility of each transmission to save the precious bandwidth resource in the VANET. The selected relays should best satisfy all neighbors' smooth playback needs, which can be inferred through vehicles' reception statuses. Here three components are needed: i) a local coordinator that serve as an arbitrator, with which a consensus on relay selection can be reliably and efficiently achieved; ii) the computation of nodes' "utilities" that represents their capability to satisfy others; iii) The selection of appropriate parameters(such as segment length, etc), for fast LMS propagation and continuous

Sequence number of current road segment	Temporary coordinator's ID	Temporary coordinator's distance	Flow ID	Sequence number of generation under playing	Sequence number of the first generation in buffer	Rank1	Rank2	......	RankN
1	4	4	1	2	2	N			

Fig. 5: The format of piggybacked information (in Byte), where N is the size of the playback buffer (in generation).

coverage.

1) Distributed Coordinator Selection: All vehicles in the same road segment agree on an unique local coordinator at the end of each control time slot, based on geographic information. For both reliability and efficiency considerations, we propose an accumulated consensus mechanism based on information piggybacked in the safety messages. We firstly define a *temporary coordinator* as the vehicle closest to the segment center that a vehicle currently knows. Each vehicle considers itself as the default temporary coordinator at the beginning of each control time slot. For each overheard safety message originated from a vehicle in the same segment, the receiver checks if the temporary coordinator piggybacked (Fig. 5) is closer to the segment center than the one known to itself presently. If yes, the receiver replaces its temporary coordinator with the overheard one. Since the vehicle closest to the segment center will be repeatedly claimed as temporary coordinator by multiple safety messages (like vehicle B in Fig. 3 (a)), 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.

2) Relay Selection: At the beginning of the following service time slot, each coordinator C firstly checks if its segment is scheduled to transmit in this slot or not, where the scheduling algorithm will be introduced in the next section. If yes, C will then calculate the *node utility* for each vehicle in $\mathcal{V}(C)$, the set of all the vehicles in the same segment as *C*, and designate the one with the highest utility as relay. If a tie appears, the vehicle located in the LMS propagation direction wins. The calculation of node utility consists of two steps:

i) Find the range of *interested generations* for all vehicles in $\mathcal{N}(C)$, which is the neighbor set of C and we require $\mathcal{N}(C) \supseteq \mathcal{V}(C)$. Only the generations representing streaming contents after the earliest playback time among vehicles in $\mathcal{N}(C)$ are regarded as interested ones. If there exists some urgent generations $UrgentGen$, C will give strict priority to the transmission of $UrgentGen$ during this time slot to ensure smooth playback at those vehicles. Otherwise, all the interested generations will be considered by *C*.

ii) Calculate node utility for each vehicle in $\mathcal{V}(C)$. If $UrgentGen \neq \emptyset$, only the generations in it will be considered in this calculation. With SLNC, the usefulness of a potential relay v's generation G_i is determined by the difference in the symbols' ranks of G_i between v and its neighbors. Due to wireless medium's broadcast nature, G_i's utility to others increases with both the average usefulness of G_i and the number of vehicles it can benefit. Thus, for $v \in \mathcal{V}(C)$, the *generation utility* of G_i is defined as:

$$U(G_i, v) = \sum_{v' \in \mathcal{N}(v)} Step(\lfloor \bar{r}_{v,i} \rfloor - \lfloor \bar{r}_{v',i} \rfloor) \times Urgent(G_i, v') \tag{2}$$

where $\lfloor \bar{r}_{v,i} \rfloor$ is the fussy average rank of node v's generation i. $Step(x) = x$, if $x > 0$; otherwise, $Step(x) = 0$. And $Urgent(G_i, v') = priValue$, if G_i is urgently requested by vehicle v', otherwise, $Urgent(G_i, v') = \frac{priValue}{2^{i-i_0}}$, where i_0 is the index of the urgent generation closest to the physical world's time. The $priValue$ is an adjustable system parameter which controls the relative importance of priority generations. Note that, since the coordinator does not know $\mathcal{N}(v)$ under the single-hop piggyback mechanism, we substitute $\mathcal{N}(v)$ by $\mathcal{N}(v) \cap \mathcal{N}(C)$. In fact, if we assume the safety messages are sent at the basic rate which can reach larger range (e.g. $2\times$) than normal data packets, then $\mathcal{N}(v)$ can be further reduced to nodes within v's data communication range ($\mathcal{N}'(v)$) (which will be explained later), which can be estimated by C.

This utility measures how much innovative information node v can give to other vehicles in $\mathcal{V}(c)$ in total if it broadcast coded packets generated from G_i. Currently we do not consider the link qualities between v and the receivers. The *node utility* $U(v)$ of vehicular node v is defined as $max_{G_i \in \text{interested generations}}\{U(G_i, v)\}$, which estimates the maximum amount of innovative information v can provide to other vehicles in $\mathcal{N}(v)$ for one generation. We do not look at the aggregate utility of multiple generations, since transmitting many generations takes a long time which may cross multiple time slots and the VANET topology has already changed.

The coordinator C designates R, the vehicle having the maximum $U(R)$, as the relay using a unicast message, which enables R to use the current service time slot. R then actively pushes coded packets generated from G_R with the maximum $U(G_R, R)$. Note that, the required number of coded pieces to send during one service time slot can be estimated based on $\frac{1}{|\mathcal{N}(R)|} \sum_{v' \in \mathcal{N}(R)} Step(\lfloor \bar{r}_{R,i} \rfloor - \lfloor \bar{r}_{v',i} \rfloor)$, which will not be elaborated here.

3) Determining the Segment Length: The length of the segment, SL, is an important parameter that affects the utility of relay selection and propagation speed of the LMS flow. On the one hand, if SL is too large, a relay at one end of a segment may not convey enough information to the neighboring segment in its scheduled time slot, and in the next slot the relay in the neighboring segment would have few innovative information to transmit, which affects smooth playback of LMS. On the other hand, if SL is too small, vehicles in adjacent segments tend to have similar reception statuses and their relays probably will transmit duplicate information. Both extremes could lead to low bandwidth efficiency and large service delivery delay.

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 simpler alternative approach, we define an equivalent *data communication range* CR under free space propagation

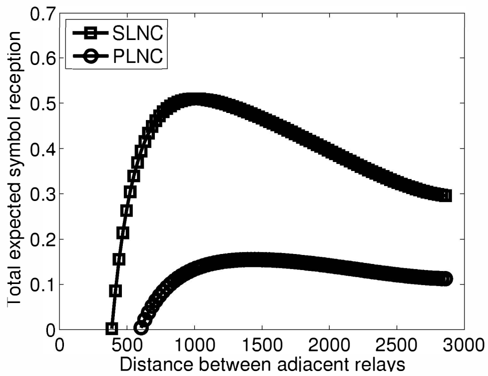


Fig. 6: The total expected symbol reception when CR=277m, ER=700m. Data rate is 12Mbps, and Nakagami fading model is used.

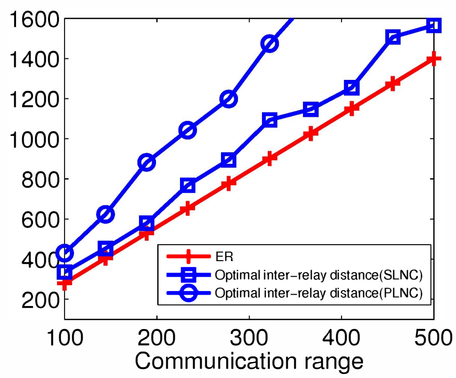


Fig. 7: The optimal distance between two adjacent relays under various data communication ranges.

model(Friis)[4], $CR = \sqrt{\frac{T_p G}{Th_{CR}}}$, where T_p is the transmission power, G is the antenna gain and Th_{CR} is the data reception threshold. Thus we set $SL \approx CR$ in this paper.

E. 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 [11]. 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. In other words, the bandwidth can be used most efficiently.

Next we discuss how to determine the D_{opt}. First we define "optimal inter relay distance". Consider a straight highway of length L, where vehicles are uniformly distributed. n relays, $v_1, v_2, ..., v_n$, lie on the highway with equal inter-distance. All the relays simultaneously and continuously transmit coded streaming content to other vehicles, and each symbol is assumed to be useful if it is correctly received. The *average symbol reception probability* Pr_{avg} for all the vehicles in the VANET is defined as: the average probability that each vehicle receives one symbol from any of the n relays during the period of one symbol's transmission. We assume a vehicle cannot receive more than one symbol at the same time. The inter-relay distance is considered as D_{opt} if the achieved Pr_{avg} is maximized.

Under wireless propagation models with channel fading (such as Nakagami model), it is very hard to derive a closed form solution for Pr_{avg}. Therefore, we approximate Pr_{avg} by Monte-Carlo simulations[5]:

$$Pr_{avg} = \frac{\text{Total \# of symbols correctly received by all vehicles}}{\text{Total \# of symbols sent by all relays} \times \text{total \# of receivers}} \quad (3)$$

where for each receiver, the received signal to noise ratio (SNR) is randomly sampled from the propagation model. We generate 10 random topologies on a highway of $10km$ with 1000 vehicles and all the relays transmit 100 pieces simultaneously, each containing 30 symbols. The value of n varies according to $\frac{L}{d}$. We also evaluate the performance of PLNC under the same setting, where a packet is considered as correctly received if all of its symbols are correctly received. The results are given in Fig. 6 and Fig. 7.

We can see that when $CR = 277m$, for SLNC, $D_{opt} \approx$ 800m, under which Pr_{avg} is above 0.5; if PLNC is applied, $D_{opt} \approx$ 1250m, under which $Pr_{avg} < 0.2$. This confirms that SLNC tolerates transmission errors better than PLNC, which allows more aggressive concurrent transmissions and achieves higher bandwidth efficiency. We also find similar conclusions under other CR values .

In addition, SLNC's shorter D_{opt} simplifies protocol design. In Fig.7, one can observe that SLNC's D_{opt} is quite close to *energy detection range* ER under all the communication ranges. ER is again an equivalent range defined under free space propagation model(Friis). The implications are that, with SLNC, we can make the channel access decisions largely based on carrier sense, which is not the case for PLNC (which must consider hidden terminals). We exploit this characteristic in the opportunistic scheduling algorithm in the next section.

2) Temporal Coordination: To provide continuous streaming coverage and to satisfy the strict time constraint of LMS service, 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. At first, we define the number of separating segments between two adjacent transmitting relays as W_{opt}, which can be calculated as $\lfloor \frac{D_{opt}}{SL} \rfloor$. 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: $T_i \equiv i \ mod \ (W_{opt} + 1)$. 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.

[4]Although this range is originally defined for packet reception in 802.11p standards, it is also a meaningful approximation for symbol reception.

[5]The details are omitted due to space limitations. Please refer to [2].

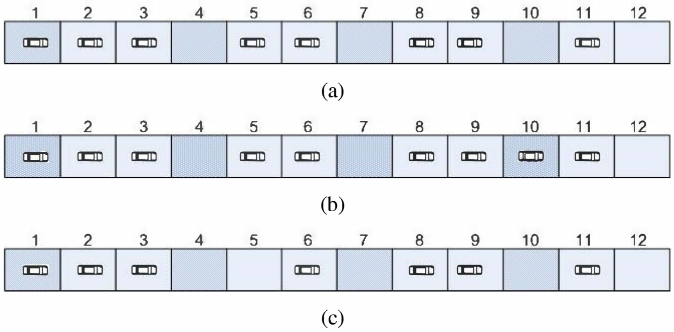


Fig. 8: Sparse VANETs (T=1, R=3). (a): LRR wastes transmission opportunities. (b)-(c): Using OLRR, secondary segments can take the unused transmission opportunities of primary segments (4,7,10).

F. OLRR: Opportunistic LRR Scheduling for Sparse VANETs

Due to the highly dynamic nature of VANET, it tends to experience partitions frequently [10], 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, which results in low bandwidth efficiency. This is illustrated in Fig. 8(a), where the segments 4,7,10 contain no vehicles, and their scheduled time slots 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. If not, certain *secondary segments* will gain channel access according to some priority assignment. In order to sense the channel, a few additional rounds ($3 \times (W_{opt} + 1)$ subslots) is allocated before data transmission. Thanks to SLNC, each coordinator/relay does not need to consider the transmitters out of its energy detection capability, which greatly simplifies protocol design.

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. 8 to illustrate the basic idea of OLRR. Suppose $W_{opt} = 2$ and C_1, C_4, C_7, C_{10} are scheduled to use the channel simultaneously in the current service time slot. In Fig. 8(a), 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}. For Fig. 8(b), C_8 will give up this opportunity, since otherwise it will incur unnecessary interference to the transmission of C_{10}. The situation in Fig. 8(c) is a little different. Now C_6 and C_8 will try to take 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_8 has higher priority and will take this transmitting opportunity.

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

1: **Input**: Coordinator C_i, segment ID 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

TABLE I: Parameter Settings

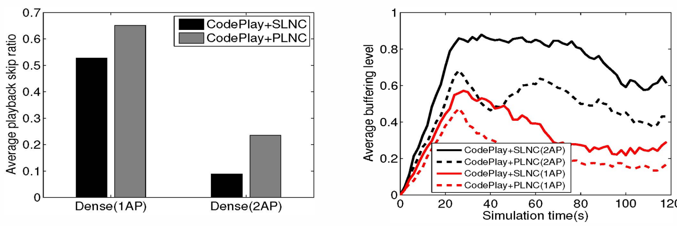
Data rates for LMS and safety msg.	12Mbps, 3Mbps
Data communication range	$CR = 250$m
Time per generation, piece size	2s, 1KB
Safety message length (with piggyback)	130B
Buffer capacity	15 generations
$PriValue$	32
# of generations in priority region	$\alpha = 1$

Finally, the reason we have $3 \times (W_{opt}+1)$ subslots is to ensure that each coordinator will be able to determine a unique segment (w.h.p) that is transmitting in each subslot. 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 μs, which takes preamble, SIFS, etc. into consideration. For $W_{opt} = 2$, the total extra time is $3 \times (2+1) \times 100 = 900$ μs, which is less than 2% of a service time slot with length of $50ms$.

V. Performance Evaluation

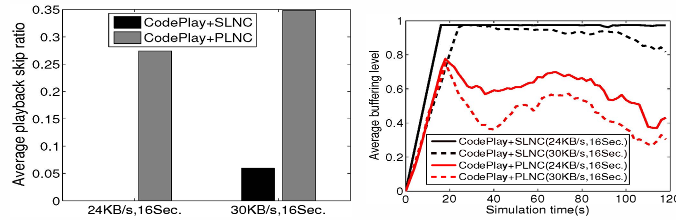
We implemented and evaluated CodePlay by simulations using NS-2.34. The SLNC is implemented based on [11], with an enhanced run-length coding technique which is more suitable for consecutively broadcasting a generation of coded pieces in CodePlay. To ensure unique coordinator selection within the same 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, and one or two LMS source(s) (e.g., access points) can be located at one or either ends of the highway. The upper part of the highway (west bound) is regarded as the AoI. We simulate both dense and sparse VANETs by using two traffic densities: 66.7 cars/km and 35.5 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. The closest state-of-the-art LMS scheme to ours is emergency video dissemination in VANETs using PLNC (NCDD, [10]). However



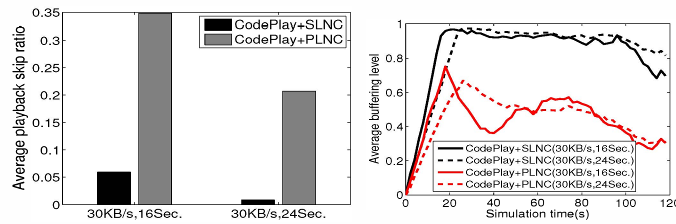
(a) (b)

Fig. 9: Comparison between using one and two APs, dense highway, source rate=12KB/s, initial buffering delay=16Sec.



(a) (b)

Fig. 10: Fixed initial buffering delay, varying source rates. Sparse highway.



(a) (b)

Fig. 11: Fixed rate, varying initial buffering delay. Sparse highway.

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 to not implement NCDD, but we have compared our results with the reported ones in [10].

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. Buffering level, the percentage of the buffered LMS contents between current playback time and physical world time. They both reflect the playback quality, i.e., smoothness [5].

A. Effect of Number of LMS Sources

We first consider how the LMS performance is affected by the number of sources (AP), i.e., only one AP which is placed on one end of the highway, or two at both ends of it. Our main finding is that, the two-source case significantly outperforms the single-source case. Fig. 9 shows the difference between using one and two APs under the dense highway with length $L = 2250$m. Both protocols, CodePlay+SLNC and CodePlay+PLNC perform much better under the two-AP case than the single-AP case. Another observation is that CodePlay+PLNC can not work well even in the two-AP case, the skip ratio of which is as high as 24%. However, the adoption of SLNC can reduce the skip ratio to less than 8%, which enables a much better playback experience.

This can be explained as follows. Because of lossy wireless links, a single flow is not able to sustain smooth playback of the LMS content after traversing a large number of hops in the VANET, which is also in line with the conclusions of routing throughput in multi-hop wireless networks. For two crossing flows with the same content, the packet losses are compensated by innovative symbols/packets from both directions. This can also be proved by the higher buffering levels in the two-AP case shown in Fig. 9(b). Therefore, in the following we evaluate CodePlay based on the two-AP case.

B. Initial Buffering Delay and Smooth Playback

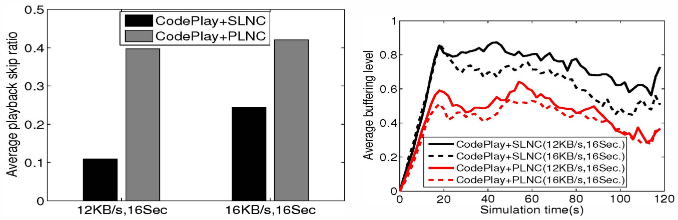
To further illustrate the advantage of CodePlay in providing better LMS services in VANET, we investigate the relationship between initial buffering delay, source rate and the metrics for smooth playback under a relatively sparse highway scenario. In the first simulation set, we fix initial buffering delay as 16 seconds, and increases the source rate from 24 KB/s to 30 KB/s. The results are presented in Fig. 10. We can see that the skip ratio for CodePlay+SLNC is much lower that its PLNC based component, where the former's skip ratio is 0 under 24 KB/s and 6% under 30 KB/s. This suggest that rate higher than 24 KB/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 [10].

In the second simulation set, we fix the source rate as 30 KB/s and increase the initial buffering delay from 16 to 24 seconds. From Fig. 11, we can see an obvious reduction in the skip ratio for the CodePlay+SLNC, from 6% to 0.8%, and an increase in the buffering level for both protocols. This result is consistent with intuitions, and implies that initial buffering delay plays an important role in VANET LMS services.

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 $L = 2250$m, the car will be at 1750m from the accident spot and may still have enough time to take actions.

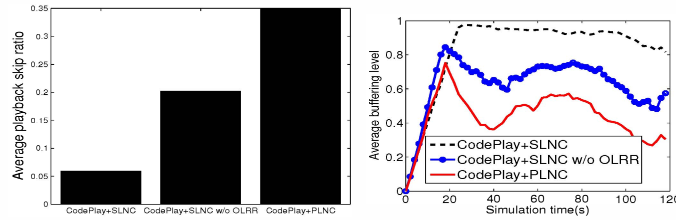
C. Effect of Traffic Density

Next we study the performance of CodePlay under the dense traffic condition. Fig. 12 shows the whole set of simulation



(a) (b)

Fig. 12: Impact of traffic density. Dense highway.



(a) (b)

Fig. 13: Effect of opportunistic transmission scheduling.

results with various source rates and buffering delays. Though CodePlay+SLNC still outperforms CodePlay+PLNC, compared with the sparse case, the skip ratio of both protocols are higher and buffering levels lower. Especially, the skip ratio reaches up to more than 10%, which could be unacceptable from application layer. We have observed (not shown) that the relay selection is almost always unique and is highly reliable, therefore the worse performance can be mainly ascribed to the limitations in the node utility functions, which is directly associated with how much innovative information a relay can deliver to all neighboring nodes. For broadcasting in a dense VANET, since there could be too many vehicles urgently demanding different portions of the LMS content, it is intrinsically hard to satisfy all their needs in a short time. Due to the time constraints of LMS applications, this leads to more frequent playback skips than in the sparse VANETs.

D. Effect of Opportunistic Scheduling

In the previous simulations for sparse scenario, we have the OLRR scheduling enabled by default. Yet it is interesting to see how the opportunistic scheduling affects the protocol performance. Thus, we presented in Fig. 13 the results of enabling and disabling the OLRR algorithm (using LRR instead). All the protocols run with source rate of 30 KB/s and initial buffering delay of 16 Sec. We can see that the OLRR much improves the performance over the basic LRR algorithm, which reduces the skip ratio from 20% to 6%. 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. And this mechanism works especially good for SLNC, since the transmission tends to be more reliable over larger distances.

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 is fully exploited for better LMS performance in VANET. Our main conclusions in this paper are: (1) LMS services in VANET with high source rates are hard, yet feasible to provide with satisfiable user experience. Even using SLNC, we may need the help of few additional infrastructure (APs) along the road to facilitate the dissemination of LMS. (2) Using CodePlay with SLNC, the playback smoothness can be greatly enhanced over traditional protocols for source rates up to 24 KB/s, and with acceptable buffering delay, especially in sparse VANETs.

Acknowledgement. This work was supported in part by the US National Science Foundation under grants CNS-0746977, CNS-0716306 and CNS-0831628.

References

[1] Q. Xu, T. Mak, J. Ko, and R. Sengupta, "Vehicle-to-vehicle safety messaging in dsrc," in *VANET '04*, 2004, pp. 19–28.
[2] M. Li, Z. Yang, and W. Lou, "Codeon: Cooperative popular content distribution for vehicular networks using symbol level network coding," *Technical Report, ECE, WPI, 2010.*
[3] 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.
[4] 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.
[5] M. Wang and B. Li, "R2: Random push with random network coding in live peer-to-peer streaming," *IEEE JSAC*, vol. 25, no. 9, pp. 1655 –1666, December 2007.
[6] H. Seferoglu and A. Markopoulou, "Video-aware opportunistic network coding over wireless networks," *IEEE JSAC*, vol. 27, no. 5, pp. 713 –728, June 2009.
[7] B. L. S. Z. Zimu Liu, Chuan Wu, "Uusee: Large-scale operational on-demand streaming with random network coding," in *INFOCOM '10*, 2010.
[8] J.-S. Park, M. Gerla, D. Lun, Y. Yi, and M. Medard, "Codecast: a network-coding-based ad hoc multicast protocol," *IEEE Wireless Communications*, vol. 13, no. 5, pp. 76 –81, Oct. 2006.
[9] R. Ahlswede, N. Cai, S.-Y. Li, and R. Yeung, "Network information flow," *IEEE Trans. Inf. Theory*, vol. 46, no. 4, pp. 1204–1216, Jul 2000.
[10] J.-S. Park, U. Lee, S. Y. Oh, M. Gerla, and D. S. Lun, "Emergency related video streaming in vanet using network coding," in *VANET '06*, 2006.
[11] S. Katti, D. Katabi, H. Balakrishnan, and M. Medard, "Symbol-level network coding for wireless mesh networks," in *SIGCOMM '08*, 2008, pp. 401–412.
[12] M. Guo, M. Ammar, and E. Zegura, "V3: a vehicle-to-vehicle live video streaming architecture," in *PerCom '05*, march 2005, pp. 171 – 180.
[13] F. Soldo, C. Casetti, C.-F. Chiasserini, and P. Chaparro, "Streaming media distribution in vanets," *GLOBECOM '08*, pp. 1–6, 2008.
[14] M. Bonuccelli, G. Giunta, F. Lonetti, and F. Martelli, "Real-time video transmission in vehicular networks," *MoVE*, May 2007.
[15] N. Qadri, M. Altaf, M. Fleury, M. Ghanbari, and H. Sammak, "Robust video streaming over an urban vanet," *IEEE Wireless and Mobile Computing, Networking and Communication*, pp. 429–434, 2009.
[16] 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 *PIMRC 2005*, Sept. 2005, pp. 1936 –1940.
[17] "Ieee trial-use standard for wireless access in vehicular environments (wave) - multi-channel operation," *IEEE Std 1609.4-2006*, pp. c1–74, 2006.
[18] T. Mak, K. Laberteaux, R. Sengupta, and M. Ergen, "Multichannel medium access control for dedicated short-range communications," *IEEE Trans. Vehi. Tech.*, vol. 58, no. 1, pp. 349 –366, Jan. 2009.
[19] 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.
[20] S. Chachulski, M. Jennings, S. Katti, and D. Katabi, "Trading structure for randomness in wireless opportunistic routing," *SIGCOMM CCR.*, vol. 37, no. 4, pp. 169–180, 2007.
[21] M. Li, W. Lou, and K. Zeng, "Oppcast: Opportunistic broadcast ofwarning messages in vanets with unreliable links," in *IEEE MASS'09*, Oct. 2009.
[22] M. Johnson, L. D. Nardis, and K. Ramch, "Collaborative content distribution for vehicular ad hoc networks," Sept. 2006.