

# Vehicular communications: emergency video streams and network coding

Joon-Sang Park · Uichin Lee · Mario Gerla

Received: 11 January 2010 / Accepted: 23 February 2010 / Published online: 10 March 2010  
© The Author(s) 2010. This article is published with open access at Springerlink.com

**Abstract** Vehicular communications are becoming a reality driven by various applications. Among those applications safe navigation support is of most significance. In designing such navigation safety applications, reliable dissemination of data, i.e., *every affected vehicle* receives data, is the key issue. Past research focused on the reliable dissemination problem of plain media type (e.g., text) safety messages, whereas we look at the problem of reliable and efficient dissemination of multimedia type (e.g., video, audio) safety information. Considering the potential volume of multimedia traffic in a large metropolis and the unpredictability of vehicular networks (e.g., high speed, partitions, obstacles, radio propagation anomalies, radio interference, etc.), reliable and efficient multimedia dissemination is non-trivial. By using a recently developed technique, *network coding*, we describe a method for reliable dissemination of video streams in case of emergencies. Simulation results show that in a typical setting, with representative channel errors/losses, our approach yields near 100% delivery ratio as compared to 92% delivery ratio by traditional multicasting. More importantly, the overhead is reduced by as much as 60%. Another important benefit is robustness to temporary disconnections. If the column of vehicles on the road has gaps, network coding jointly

with “data muling” using vehicles in the opposite directions can deliver the multimedia files even to intermittently connected components.

**Keywords** Vehicular networks · Network coding · Data mules

## 1 Introduction

In the near future, car-to-car communications and networking will be driven mostly by navigation safety support applications. In the simplest example, vehicular communications can be used to exchange *virtual* tail lamp signals between vehicles more reliably than the conventional visual tail lamp system. Moreover, human reaction times can be considerably reduced. To implement such a safety warning system, a finite set of signals augmented with additional information such as vehicle location and speed can be represented as plain media type (e.g., text) data and packed into a very small number of short packets, resulting in rather modest data rates. Emerging vehicular networks will be equipped with advanced radios and wireless LAN grade bandwidth. This will allow aggressive data rates and applications well beyond text-only alarms. In the longer term, multimedia data will also be introduced to enhance navigation safety. In fact, multimedia data such as images and even short video clips of an accident or dangerous situation ahead (e.g., flood, fire, earthquake damage, terrorist attack damage, etc.) will provide drivers with more precise information than simple text messages. It will allow them to make a more informed decision (whether to proceed or turn back) based on personal priorities and/or on vehicle capabilities.

For instance, suppose that a critical traffic/safety situation occurs on a highway, say, a hurricane caused flood followed

---

J.-S. Park (✉)  
Hongik University, 72-1 Sangsoo-dong, Mapo-gu, Seoul,  
South Korea  
e-mail: [jsp@hongik.ac.kr](mailto:jsp@hongik.ac.kr)

U. Lee  
Bell Labs, Alcatel-Lucent, 791 Holmdel-Keyport Rd., Holmdel,  
NJ 07733, USA  
e-mail: [uichin.lee@bell-labs.com](mailto:uichin.lee@bell-labs.com)

M. Gerla  
University of California, Los Angeles, CA 90095, USA  
e-mail: [gerla@cs.ucla.edu](mailto:gerla@cs.ucla.edu)

by major traffic congestion. In such cases, multimedia content transmission such as image and video streaming could be triggered on one or more lead cars and propagated to vehicles several miles behind—to *visually* inform drivers of the problem and allow them to decide if they should proceed or turn around. In more serious emergencies, this visual information will help first responders to best prepare for the assistance of the victims, and it will help the traffic agents to design vehicle evacuation strategies. This information dissemination must take place within minutes from the accident, much before police and news helicopters can come to the scene and broadcast video feeds to the networks.

For the above warning systems to work, however, it is crucial that the broadcast be efficient and lightweight so as not to congest the network. Also, the safety-related messages must be *reliably* delivered to all the *impacted* nodes in a vehicular network so that they can cooperatively coordinate evasive actions. Thus reliable data dissemination is a key requirement for multimedia navigation safety applications. The application must be protected against unreliable wireless channel, e.g., due to random obstacles on the path and packet drops. It must also be robust to the unpredictable nature of the vehicular network, e.g., variable network density, frequent partitions and gaps in the column of cars driving in the same direction, etc. Finally, the delivery of emergency images and video streams is most critical following major disasters that have destroyed the entire infrastructure. For example, a hurricane or earthquake will wipe out all cellular repeaters and Wi-Fi access points. Thus, the emergency communications cannot rely on infrastructure support but must be totally contained within the vehicular network.

Cars may become separated on the highway, forming platoons. If an accident occurs, a platoon that follows by 30 seconds, for example, will automatically incur a 30-second delay. There are two approaches to deal with this problem. First, the applications discussed in this paper are intrinsically delay-tolerant, in the sense that reactive action to the accident alert (e.g., stopping and turning around) is generally required only when the second platoon establishes radio contact with the first one. Thus, it is important that the vehicles in the second platoon learn of the accident all at the same time (albeit with a 30-second latency). The second approach exploits vehicles coming in the opposite direction (assuming that the highway has multiple lanes in both directions). The opposing traffic may actually consist of vehicles that have already turned around after the accident. This second approach uses the vehicles in the opposing directions as “data mules.” It can be used in conjunction with the first approach and can speed up the delivery of alarms with obvious navigation safety benefits.

In this study we address both the reliable delivery within a single, connected convoy of cars, and the “delayed” delivery to multiple convoys separated by random gaps. The

goal of our reliable dissemination service design within a single convoy is to use loss recovery efficiently, while at the same time keeping low overhead in check. Packet losses cause quality degradation in the multimedia data but 100% recovery is not the right answer since full recovery may incur too much overhead and most of multimedia data streams tolerate losses in the order of 1–2%. Delay must also be kept in check when considering recovery. A few seconds of discrepancy between neighbor vehicles can be tolerated, yet an end-to-end 100% recovery scheme (for example, end-to-end recovery of individual packets using TCP) that leads to delays in the order of several seconds would be unacceptable in vehicular safety video streaming. In this paper, we show that network coding can be a very effective solution for reliable, low-overhead, limited delay delivery within a contiguous convoy.

In the delayed delivery study we address the “data muling” of a video streams to disconnected platoons using the traffic in the opposite direction and again we show that network coding provides an efficient solution, leading to lower completion delays or conversely better accuracy than other techniques.

The rest of this paper is organized as follows. Section 2 presents the problem statement. Section 3 illustrates the network coding based data dissemination protocol, Section 4 evaluates the protocol via simulations, and an analysis of the protocol is presented in Sect. 5. Section 6 discusses the related work. Finally, Sect. 7 concludes this paper.

## 2 Problem statement

There is a vast literature on reliable data dissemination in vehicular networks (see Sect. 6). The main focus however has been on short alert messages broadcast to neighbor vehicle to prevent crashes. The short message application can be supported by conventional solutions such as repetition of messages. The reliable data dissemination problem becomes much more complex with multimedia data that must be disseminated over multiple hops to remote vehicles well beyond the visual range. Multimedia files (e.g., images) are inherently large and multimedia streams require high bandwidth. When disseminated without proper controls, they can cause severe congestion and shut down other safety traffic. In data dissemination over vehicular networks, packets may be corrupted and lost because of many reasons. Fading, environment interference, and mobility can produce random-like losses. Another cause of loss is packet collision. In particular, collisions among hidden terminals are quite frequent in broadcast where the RTS/CTS feature of IEEE 802.11 MAC [27] is disabled. We assume in this study that the DSRC MAC protocol or the emerging 802.11p MAC protocol are used. Because of broadcast, packet losses tend

to be random and non-uniform even in the same locality. That is, each node in a neighborhood has dissimilar, rapidly varying packet reception characteristics. Therefore, upon packet loss, a “local diversity” recovery strategy, i.e., neighbors helping each other, can be very effective. To fight locally correlated losses which are experienced by multiple receivers in the same neighborhood due to common upstream congestion, broken link, etc., path diversity paves the way to effective recovery. Path diversity, i.e., opportunities to use multiple, disjoint paths, is abundant in urban vehicular networks when vehicles are densely packed. The main issue is to utilize path diversity efficiently. Also, the overhead must be low, so that the scheme does not introduce extra inefficiency when the vehicle network is “sparse.” Our approach seeks to achieve reliable multimedia data dissemination using localized neighbor recovery and multipath diversity with very low overhead. The key ingredient of our approach is a *random network coding* scheme which transparently implements both localized neighbor recovery and path diversity with remarkably low overhead.

In addition to delivery ratio and traffic overhead, a critical measure for multimedia video delivery is end-to-end delay. We must distinguish between real-time multimedia (e.g., video conference) and non-real-time multimedia (e.g., one-way video streaming). For real time, the delay average and jitter are subject to tight constraints. For non-real time, the average delay and jitter are subject to much more relaxed constraints (in the order of seconds). Basically, the receiver reassembles packets and orderly streams them to the application using a reconstruction buffer. The attention then shifts to buffer requirements and user buffer limitations. In this study, we address only *non-interactive real-time streaming*. Note that the delay requirements are more strict than in conventional video streaming (e.g., video on demand). Drivers in the proximity of each other must take coordinated actions based on what they are shown. So, discrepancies of more than a few seconds are not tolerated. Regarding buffering requirements, this constraint will be ignored since the vehicle (as opposed to hand-held or sensor-based platforms) has practically unlimited buffer capacity.

### 3 Reliable multimedia delivery

In this section, we describe a network coding based reliable multimedia delivery service.

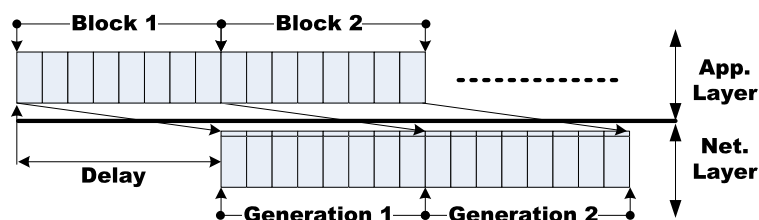
Suppose a multimedia data source generates a stream of frames  $\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \dots$ , where subscripts denote unique and consecutive sequence numbers. We assume that streams can be uniquely distinguished by the source address and port number pair or a globally unique identification number assigned to each stream. We use a tuple  $(blockid, blocksize)$  where  $blocksize > 0$  to indicate a block of frames with sequence numbers greater than or equal to  $blockid$  and smaller than  $(blockid + blocksize)$  (i.e.,  $\mathbf{p}_{blockid}, \dots, \mathbf{p}_{blockid+blocksize-1}$ ). A *coded packet*  $\mathbf{c}_{(blockid, blocksize)}$  is a linear combination of frames in  $(blockid, blocksize)$ . That is,

$$\mathbf{c}_{(blockid, blocksize)} = \sum_{k=1}^{blocksize} e_k \mathbf{p}_{(blockid+k-1)}$$

where  $e_k$  is an element in a finite field  $\mathbb{F}$ . Data frames  $\mathbf{p}$ 's and coded packets  $\mathbf{c}$ 's are also regarded as vectors over the field. In the header of a coded packet, the coefficient vector  $\mathbf{e} = [e_1 \dots e_{blocksize}]$  is stored along with  $blockid$  and  $blocksize$  for the purpose of *decoding* at the receivers. When generating a  $\mathbf{c}$ , each  $e_k$  is chosen randomly from  $\mathbb{F}$ , which is in general referred to as the random linear coding.  $\mathbf{c}$ 's with the same label  $(blockid, blocksize)$  are said to be in the same *generation* (Fig. 1). Throughout this paper we use lowercase boldface letters to denote vectors, encoding vector, frames, or packets, uppercase letters to denote matrices or constant numbers, and italics to denote variables or fields in packet headers.

The reliable delivery service agent (or layer) residing on the video source generates and transmits coded packets to the receivers. Since a block of frames is required to generate a coded packet, the agent residing on the video source collects frames generated by the application and buffers them. For simplicity, we assume that data storage on a vehicle is large enough to store all the data for a limited amount of time. Under the general framework of random linear coding, many variant strategies regarding encoding and forwarding can exist. Here we present two encoding schemes: *progressive coding* scheme and *block coding* scheme. In the progressive coding scheme, whenever a new frame  $\mathbf{p}_k$  becomes

**Fig. 1** Multimedia streaming with  $blocksize = 8$



available, a coded packet  $\mathbf{c}_{(k-blocksize+1,blocksize)}$  is generated by combining the new frame with the preceding  $blocksize - 1$  frames and is transmitted instantly. In the block coding scheme, the agent implements a delayed transmission strategy: instead of transmitting a packet instantly when a frame becomes ready, a series of coded packets,  $\mathbf{c}_{(blockid,blocksize)}$ 's, are transmitted when all the frames in  $(blockid, blocksize)$  are collected and blocks are organized such that no original data frame appears in two different blocks. In fact,  $blocksize$  needs not be a fixed value. If the agent receives application frames the format of which is known a priori,  $blocksize$  can be determined on the fly, according to the delay constraints of the packets. The objective is to make the size of each block big enough to gain efficiency while minimizing the possibility of delivering packets with delay constraint violations. (In general, the bigger the block size, the greater the efficiency gain is, and also the delay.) Otherwise, the agent uses a predefined number to limit maximum wait time in the buffer. Once the whole block is amassed, the agent generates  $(blocksize)$  coded packets and broadcasts them to the neighborhood. Due to the page limit and for the ease of explanation, we hereafter assume that we only use the block coding scheme.

On reception of a coded packet  $\mathbf{c}_{(blockid,blocksize)}$ , every node stores the packet in its local memory for later decoding and forwarding. To recover  $blocksize$  original frames belonging to  $(blockid, blocksize)$ , a node should collect a  $blocksize$  number of coded packets tagged with  $(blockid, blocksize)$  whose encoding vectors are linearly independent of each other. Once collected, the reliable delivery service agent recovers the  $blocksize$  original data frames and delivers them to the upper layer. Let  $\mathbf{c}_k$  be a coded packet labeled  $(blockid, blocksize)$  in a node's local memory,  $\mathbf{e}_k$  be the encoding vector prefixed to  $\mathbf{c}_k$ , and  $\mathbf{p}_{blockid+k-1}$  be an original data frame to be recovered, where  $k = 1, \dots, blocksize$ . Further, let  $\mathbf{E}^T = [\mathbf{e}_1^T \dots \mathbf{e}_{blocksize}^T]$ ,  $\mathbf{C}^T = [\mathbf{c}_1^T \dots \mathbf{c}_{blocksize}^T]$ , and  $\mathbf{P}^T = [\mathbf{p}_{blockid}^T \dots \mathbf{p}_{blockid+blocksize-1}^T]$ , then conceptually  $\mathbf{P} = \mathbf{E}^{-1}\mathbf{C}$ , which corresponds to the original data frames where superscript  $T$  denotes the transpose operation. Note that all  $\mathbf{e}_k$ 's must be linearly independent.

When a node receives a coded packet with a new tuple  $(blockid, blocksize)$ , it sets up a timer for the tuple  $(blockid, blocksize)$  expiring in  $blocktimeout$  seconds. When the timer expires, it broadcasts one coded packet  $\hat{\mathbf{c}}_{(blockid,blocksize)}$ , after local re-encoding to its neighbors. The local re-encoding goes through the same process that the data source has undergone to generate a coded packet, i.e., a random linear combination of packets with the same  $(blockid, blocksize)$  available in local memory. Note that though the packets in memory are coded ones thus the re-encoded packet  $\hat{\mathbf{c}}_{(blockid,blocksize)} = \sum_{k=1}^{blocksize} \acute{e}_k \mathbf{c}_k$  is tagged with the encoding vector  $\acute{\mathbf{e}} = \sum_{k=1}^{blocksize} \acute{e}_k \mathbf{e}_k$  where each  $\acute{e}_k$  is drawn uniformly from  $\mathbb{F}$  and  $\mathbf{c}_k$  and  $\mathbf{e}_k$  are again a coded packet la-

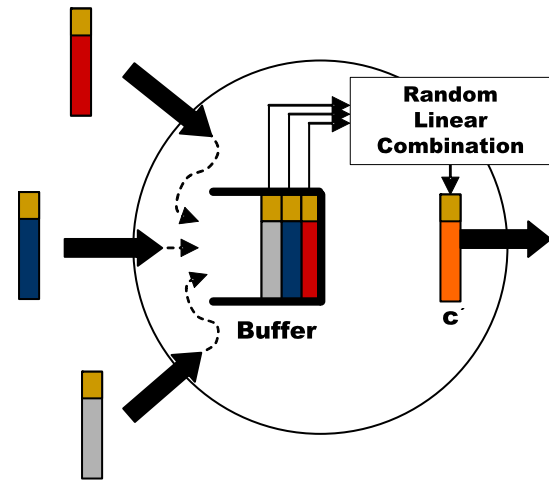


Fig. 2 Re-encoding at an intermediate node

beled  $(blockid, blocksize)$  in memory and the encoding vector prefixed to  $\mathbf{c}_k$  respectively (see Fig. 2). The timer for  $(blockid, blocksize)$  is reset on expiration unless a decodable set of packets is collected for the tuple  $(blockid, blocksize)$ .

On the expiration of the timer for  $(blockid, blocksize)$ , even though there are less than  $blocksize$  number of packets of  $(blockid, blocksize)$  in the local memory, a node has to generate and transmit a coded packet using packets available in memory. The number of frames/packets that are combined to yield a coded packet is recorded in the field  $rank$  in the header of the coded packet. A coded packet  $\mathbf{c}_{(blockid,blocksize)}$  with  $rank$  smaller than  $blocksize$  indicates that the sender of the coded packet is in need of more coded packets tagged with  $(blockid, blocksize)$ . On reception of such packets, nodes transmit another coded packet to help the original sender of  $\mathbf{c}_{(blockid,blocksize)}$  collecting more coded packets. Owing to this recovery process in combination of buffering of packets, our protocol can deliver packets efficiently and reliably across partitions. Suppose that a vehicle encounters a platoon of other vehicles carrying data that the vehicle does not have. The vehicle runs the recovery process and it collects data from the platoon. By recovering process we mean a vehicle sending out a coded data packet tagged with  $blockid, blocksize$ , and  $rank$  where  $rank < blocksize$  and in response to the  $help\ request$  packet, neighbors of the vehicles sending appropriate coded data packets. If a vehicle in seek of help has no data, then it just sends out header-only packets with  $rank = 0$ . In fact, the help request and responses' handshaking are not necessarily to be done block by block (or generation by generation) if a vehicle wants to collect consecutive blocks of frames. There also can be a case where  $blocksize$  is unknown. Suppose a vehicle wants to collect frames with consecutive sequence numbers from  $M$  to  $N$  where  $N - M$  is larger than  $blocksize$  or  $blocksize$  is unknown. Then the vehicle broadcasts a header-only packets with  $blockid = M, blocksize =$

$(N - M)$  to its neighborhood and neighbors respond with coded packets  $\mathbf{c}(blockid, blocksize)$ 's, where  $blockid \geq M$  and  $(blockid + blocksize) \leq N$ .

Since every transmission is MAC/link layer broadcasting, a small random amount of wait time called broadcast jitter is applied to every transmission. Without broadcast jitter, MAC/link layer broadcasting suffers severely from the hidden terminal problem.

#### 4 Simulation results: delivery within a simple convoy

In this section, we study via simulation the performance of our proposed Network Coding based Data Dissemination (NCDD) scheme when applied to a single connected vehicular network—say, a convoy of cars—and compare it with conventional multicast. To this end, we implemented NCDD in QualNet [18], a packet-level network simulator, and conducted a set of simulations using the following settings: 802.11 DCF MAC; two-ray ground path-loss propagation model; 376 m of transmission range and 3 Mbits/sec of bandwidth (which is the minimum data rate of DSRC); one data source generates a constant bit-rate 10 Kbytes/sec traffic using fixed 512 bytes packet size and every node receives the traffic. Nodes move according to the Real-Track (RT) mobility model [17]. RT permits to model vehicle movements in an urban environment more realistically than other simpler and more widely used mobility models, such as Random Way Point (RWP), by restricting the areas where nodes can appear (e.g., roads). Also, in the RT model vehicles tend to aggregate and move in groups because of traffic signals and because directions can change only at road intersections. The minimum node velocity is fixed to zero and the maximum is varied from run to run. Results are averaged over 10 runs with various random seeds. For NCDD, we set 40 ms for *blocktimeout* and *blocksize* = 8.

We contrast NCDD to the plain UDP service running on top of a conventional multicast protocol in mobile settings. In the comparison, UDP assumes underlying On Demand Multicast Routing Protocol (ODMRP) [11]. We restrict our attention to ODMRP since as shown in [11] it is one of the best performing multicast protocols for one-to-many communications especially in mobile and lossy channel settings in which we are specially interested. The challenge of vehicular networks is, in fact, that of maintaining network operations in the face of nodes' mobility and lossy wireless channel. To simulate a lossy channel, nodes are forced to drop successfully received packets randomly with some probability. In figures, NCDD-dp $\beta$  denotes NCDD with packet drop probability  $\beta\%$  and similarly UDP-dp $\beta$  denotes UDP for the packet drop probability  $\beta\%$  case.

First, we look at a *street scenario*. Two hundred vehicles are moving along the streets in a confined 2000 m  $\times$  2000 m

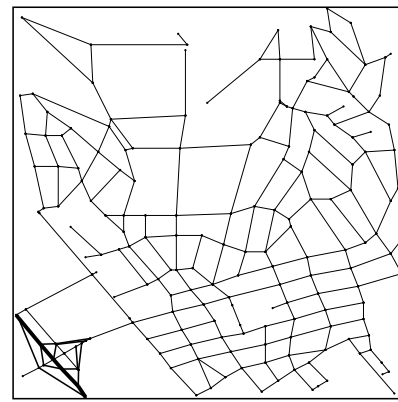


Fig. 3 Street map in the vicinity of UCLA

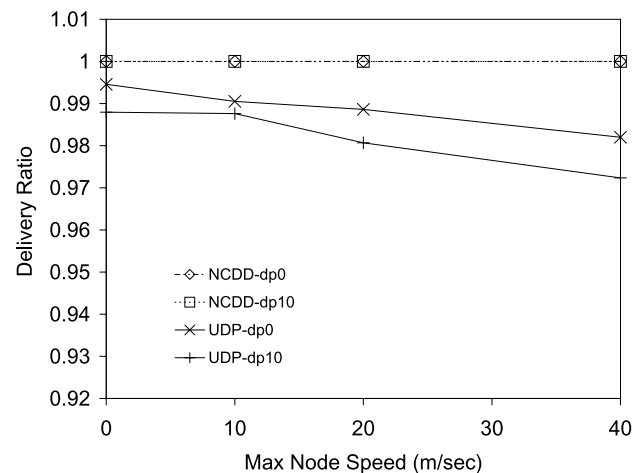
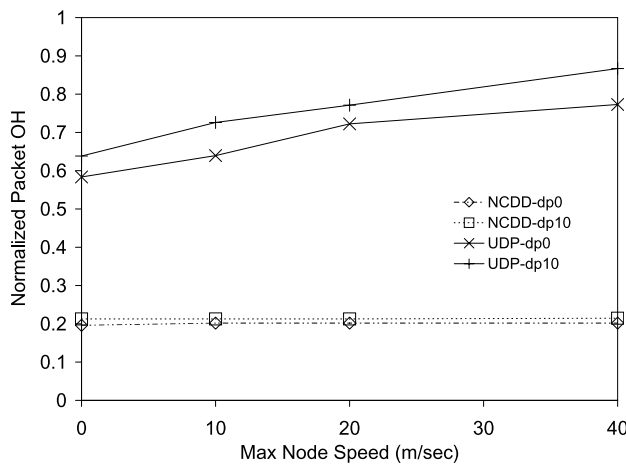


Fig. 4 Packet delivery ratio (street)

area as shown in Fig. 3. In Fig. 4, NCDD demonstrates near 100% data delivery regardless of mobility types and packet drop probability. To vary mobility types, we used four different maximum-node velocities represented by the x-axis in the figure. On the other hand, the packet delivery ratio of the conventional multicasting represented by ODMRP degrades from 99% to 97% as mobility and packet drop probability increase. The packet delivery ratio is defined as the ratio of data packets received by all receivers over total data packets sent. More importantly, as shown in Fig. 5, NCDD incurs less overhead than ODMRP. When the maximum node speed is 40 m/s, the reduction in overhead is as much as 70%. To measure protocol overhead, we use a common metric, the normalized packet overhead defined as the total number of packets transmitted to the wireless channel by any node in the network divided by the total number of data packets delivered to any receiver. As maximum node velocity increases, ODMRP's overhead also increases. The reason is that ODMRP is designed to use more and more nodes as forwarding nodes to effectively counter frequent route breakages due to the increased mobility, which



**Fig. 5** Normalized packet overhead (street)

is equivalent to trading off overhead for high packet delivery ratio to cope with mobility.

Second, we look at the *highway scenario*. Two hundred vehicles are moving either one direction or the opposite with different speeds along the 10 km long 50 m wide track. Similarly to the previous street scenario, Fig. 6 shows that NCDD demonstrates near 100% data delivery regardless of mobility types and packet drop probability, whereas ODMRP's packet delivery ratio degrades from 99% to 92% as mobility and packet drop probability increase. The overhead of ODMRP rather remains static, meaning that the number of forwarders remains constant. In a narrow highway, ODMRP fails to increase the forwarding group size and thus becomes prone to route breakage which in turn translates into low packet delivery ratio. Figure 6(c) shows the major drawback of NCDD, namely, end-to-end delay. The end-to-end delay is the difference between packet generation time at a source and packet delivery to the application at the receiver. In NCDD, a certain level of increase in end-to-end delay is inevitable since at the source it takes time to collect a block of packets such that coding over the block is possible. In our simulations, the application generates packets at a rate of 20 packets/sec so if the block size is 8 packets, each packet spends on average around 0.2 seconds waiting in the buffer at the source. Note that the progressive coding scheme does not necessarily incur this delay, which is well-suited to real-time applications. Propagation delay (end-to-end path can be as long as 10 km in our example), as well as packet recovery delay (which does not exist in the ODMRP case), also contributes to the NCDD's high end-to-end delay.

## 5 Delayed delivery across platoons

In this section we study the model with opposite direction vehicles that act as “data mules.” We are interested in evalu-

ating different schemes and their delayed delivery performance. Part of these results is reported in an earlier paper [19]. Figure 7 shows the overall diagram of short segments of the freeway. Given that specialized sensors and wireless communication devices are installed in vehicles, vehicles can *cooperate* to relieve traffic congestion and thus maximize traffic throughput. To realize this, vehicles continually exchange information about speed, acceleration, braking, obstacles, and so forth. In the future they may even travel with help of computer control at closely spaced intervals in small platoons [24]. Let  $P_f-k$  and  $P_r-k$  denote the  $k$ th platoon in a forward and reverse direction respectively. In our specific scenario, the vehicles exchange information about an accident. Namely, vehicles located just in front of the accident site multicast video streams to the platoon  $P_f-1$ . Obviously the data cannot propagate immediately to other disconnected platoons such as  $P_f-2$ . Our freeway relay model uses platoons in the opposite directions to pick up, carry, and forward the data to the other platoons in the forward direction, e.g.,  $P_f-2$ .

In this paper we are interested in calculating the delay of delivering the complete data file to the other platoons. This can be simply expressed as follows:

$$\text{Delay} = \min(\text{overtake delay}, \text{relay delay}).$$

Overtake delay is defined as the time for a random platoon to catch up with (reach within its communication range) the source platoon (or other platoons that have merged into the source platoon) that is driving in the *same* direction. Relay delay is defined as the time for a target platoon to receive the whole data from data mule platoons driving in the opposite directions. For example, let us say that we have a target platoon  $P_f-2$  as shown Fig. 7. Overtaking delay is the time for  $P_f-2$  to catch up with the source platoon  $P_f-1$ , and relay delay is the time for  $P_r-1$  to encounter  $P_f-2$ . Given the limited data transfer rate between two crossing platoons, the target platoon may have to receive different parts of the data file from different platoons, i.e., data mules. This is the main focus of our analysis.

When relaying data, we could image the following strategies:

- *Relay without coding (R-WC)*: A platoon passing by the accident site randomly picks up a number of packets and “data mules” them to the disconnected target platoon.
- *Relay with erasure coding (R-EC)*: A source encodes the data using erasure coding. Erasure coding protects from packet loss caused by relaying platoons that may exit the freeway.
- *Relay with network coding (R-NC)*: The data is distributed using random linear network coding.

Vehicles arrive at independently distributed random intervals. In the traffic theory [12], this type of random arrival

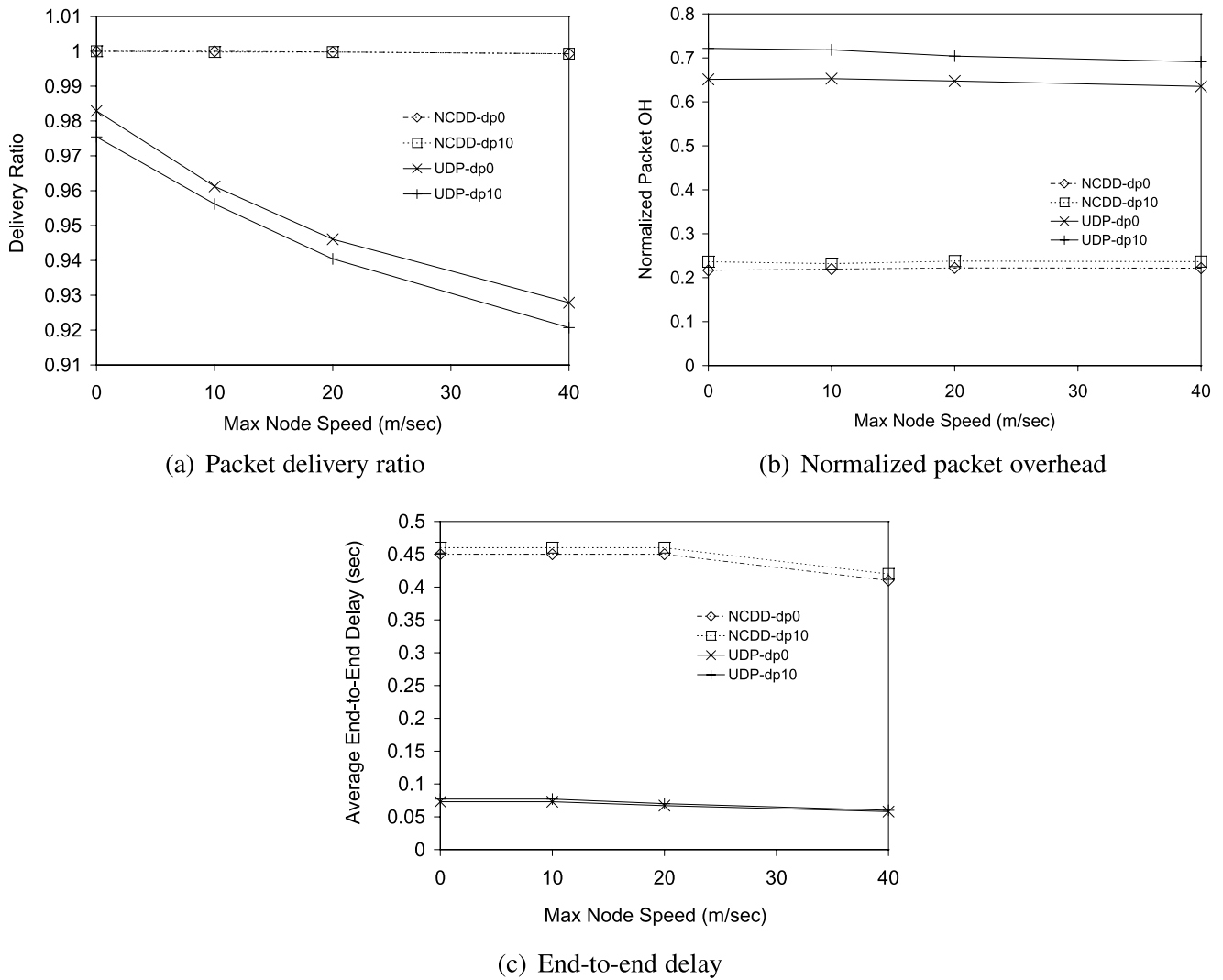
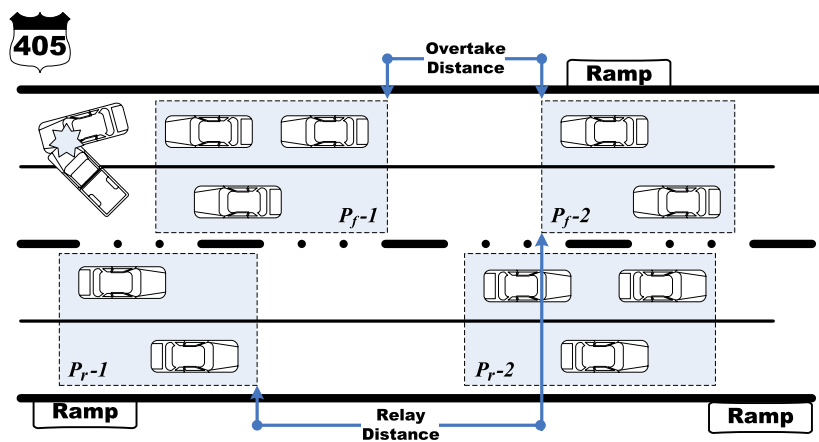


Fig. 6 Performance of NCDD and ODMRP in the highway scenario

Fig. 7 Freeway relay model scenario



is often modeled using an exponential distribution with parameter  $\lambda$ , where  $\lambda$  is the flow in vehicles/second. Without loss of generality this can be extended also to a platoon ar-

riving at the scene of the accident on the freeway. The size of a random platoon could be represented as a geometric distribution with mean  $s_p$ . Then, the overall arriving process is

simply modeled as a compound Poisson process. Since our focus is calculating the relay delay, we simply assume that the speeds of platoons are constant with  $v_0$ . The distance between two platoons is purely determined by the underlying Poisson arrival process.

Let us define the process formally. A compound Poisson process with rate  $\lambda$  and jump size distribution  $G$  is a continuous-time stochastic process  $\{Y(t) : t \geq 0\}$  given by

$$Y(t) = \sum_{i=0}^{N(t)} D_i \tag{1}$$

where  $\{N(t) : t \geq 0\}$  is a Poisson process with rate  $\lambda$  and  $\{D_i : i \geq 0\}$  are independent and identically distributed random variables, with distribution function  $G$ , which are also independent of  $\{N(t) : t \geq 0\}$ . Here, the distribution function  $G$  follows a geometric distribution with mean  $s_p$ . Let us assume that the average length of a vehicle is  $\ell_v$ . The length of a platoon is simply given as  $G \cdot \ell_v$ .

From (1) we can derive the expected number of vehicles  $\mathbb{E}[Y(t)]$ , or a *random sum* with Poisson process  $\{N(t) : t \geq 0\}$  as follows:

$$\mathbb{E}[Y(t)] = \lambda \mathbb{E}[G]. \tag{2}$$

Given that we know the expected number of platoons to relay message, this allows us to derive the expected number of vehicles.

Vehicles within a platoon may leave the freeway. This depends on the density of ramps along the freeway as well as the probability of defecting from a given platoon. Let  $d_r$  denote the density of ramps and  $\bar{p}_l$  denote the average defecting probability from a given platoon.<sup>1</sup> The ineffectiveness of a random platoon in terms of delivering packets can be expressed by  $p_{ie} \propto d_r \cdot \bar{p}_l$ , thus  $p_{ie} = \alpha \cdot d_r \cdot \bar{p}_l$  where  $\alpha$  is a constant. Let  $p_e$  denote the effectiveness of a random platoon, which is given as  $p_e = 1 - p_{ie}$ . Let  $N_p(t)$  denote the number of packets that a random platoon can pick up at time  $t$ ; for example, in the figure how many packets  $P_r-1$  can pick up from  $P_f-1$ .<sup>2</sup> For ease of analysis, we assume that a platoon can pick up on average  $\bar{N}_p$  packets. The effective number of packets delivered to the target platoon is simply given as  $p_e \bar{N}_p$ .

Let  $N_d$  denote the total number of packets for a given data file that must be delivered. We assume that  $N_d > p_e \bar{N}_p$ , or we need multiple number of platoons to get the whole data. Since a random platoon arrives at the highway with a Poisson process, the average delay between two platoons is simply given as  $\bar{T}_h = 1/\lambda$ . From this, we see that the average

distance is simply  $\bar{D}_h = \bar{T}_h \times v_0$ . We model a low traffic flow scenario such that the average distance is larger than the communication range, i.e., two consecutive platoons cannot directly communicate. Now we are ready to analyze the proposed schemes.

*Relay without coding* Each “mule” platoon randomly picks up  $\bar{N}_p$  packets and delivers on average  $p_e \bar{N}_p$  packets to the target platoon. This problem is analogous to the *coupon collecting problem*. The number of coupons, i.e., the number of data packets that we need to deliver, is given as  $N_d$ . At each step, we draw a coupon whose type is uniformly distributed among all  $N_d$  types. Thus, the average number of trials required at each step to get a new coupon increases as more coupons have been collected. Let  $T_{N_d}$  be the time by which we have collected coupons belonging to all  $N_d$  distinct types. Then, the expected number of trials that we need to make is simply given as

$$\mathbb{E}[T_{N_d}^{R-WC}] = 1 + \frac{N_d}{N_d - 1} + \dots + \frac{N_d}{1} \tag{3}$$

$$= N_d \sum_{k=1}^{N_d} \frac{1}{k} \tag{4}$$

$$\simeq N_d \ln N_d. \tag{5}$$

Recall now that each platoon can carry  $p_e \bar{N}_p$  number of packets, and thus, the expected number of platoons that a target platoon must see (i.e., passes the data mule can make in order to transfer the file) is given as  $\lceil \mathbb{E}[T_{N_d}^{R-WC}] / p_e \bar{N}_p \rceil$ . It is interesting to note that since the target platoon travels in an opposite direction, the expected delay for the target platoon to meet a random platoon is simply given as  $\bar{T}_h/2$ . Therefore, the overall delay  $D_{R-WC}$  is given as

$$D_{R-WC} = \left\lceil \frac{\mathbb{E}[T_{N_d}^{R-WC}]}{p_e \bar{N}_p} \right\rceil \times \frac{\bar{T}_h}{2} \tag{6}$$

$$= \left\lceil \frac{N_d \ln N_d}{p_e \bar{N}_p} \right\rceil \times \frac{\bar{T}_h}{2}. \tag{7}$$

Thus, the time grows with  $N_d \ln N_d$ , where  $N_d$  is the data file size.

*Relay with erasure coding* Given that redundancy factor is  $r (>0)$ , erasure coding produces  $N_d(1+r)$  coded packets where  $N_d$  is the number of the original data packets. The key property of erasure coding is that the original data can be reconstructed from any of  $N_d$  packets. R-EC is quite similar to R-WC but we now only collect any subset with size  $N_d$  of  $N_d(1+r)$  coded packets; then, we may want to ask what the relative advantage of using erasure coding in terms of the average delay. Similarly to R-WC, the expected number

<sup>1</sup>The defecting probability mainly depends on the size of a platoon.  
<sup>2</sup> $N_p(t)$  is directly related to the length of a platoon as well as the speed of a platoon.



of trials for R-EC is given as:

$$\mathbb{E}[T_{N_d}^{R-EC}] = 1 + \frac{N_d(1+r)}{N_d(1+r)-1} + \dots + \frac{N_d(1+r)}{N_d r} \tag{8}$$

$$= N_d(1+r) \sum_{k=N_d r}^{N_d(1+r)} \frac{1}{k} \tag{9}$$

$$\simeq N_d(1+r) \ln\left(1 + \frac{1}{r}\right). \tag{10}$$

This approximation is based on the following lemma.

**Lemma 1** *As n goes to infinity, we have*

$$\lim_{n \rightarrow \infty} \sum_{k=nr}^{n(1+r)} \frac{1}{k} = \ln\left(1 + \frac{1}{r}\right).$$

*Proof* Since the summation is a monotonically decreasing function, we can approximate it by integrals:

$$\int_{nr}^{n(1+r)+1} \frac{1}{x} dx \leq \sum_{k=nr}^{n(1+r)} \frac{1}{k} \leq \int_{nr-1}^{n(1+r)} \frac{1}{x} dx, \tag{11}$$

$$\ln \frac{n(1+r)+1}{nr} \leq \sum_{k=nr}^{n(1+r)} \frac{1}{k} \leq \ln \frac{n(1+r)+1}{nr-1}. \tag{12}$$

As n goes to infinity, both upper and lower bound become tight, so we have:

$$\lim_{n \rightarrow \infty} \sum_{k=nr}^{n(1+r)} \frac{1}{k} = \lim_{n \rightarrow \infty} \ln \frac{n(1+r)}{nr} \tag{13}$$

$$= \ln\left(1 + \frac{1}{r}\right). \tag{14}$$

□

The overall delay  $D_{R-EC}$  is given as

$$D_{R-EC} = \left\lceil \frac{N_d(1+r) \ln(1+1/r)}{p e \bar{N}_p} \right\rceil \times \frac{\bar{T}_h}{2}. \tag{15}$$

Note that the constant logarithm factor  $\ln(1+1/r)$  plays a key role determining  $\mathbb{E}[T_{N_d}^{R-EC}]$ . If  $r \geq 0.5$ , we have  $\mathbb{E}[T_{N_d}^{R-EC}] \simeq N_d(1+r)$ . It is important to note that the delay improvement of R-EC comes at the cost of increased redundancy factor  $r$ . The video source has to generate and broadcast redundant packets proportional to the total number of packets ( $r \times N_d$ ); thus, we are utilizing the channel less efficiently (also potentially causing more collisions).

*Relay with network coding* Both R-WC and R-EC are analogous to coupon collection. The intrinsic problem of coupon collection is that once we have collected half of the

coupons, it takes a progressively longer and longer time to collect the rest of coupons. On the other hand, “algebraic mixing” of the original data in random network coding helps us to attain near optimal bound of data dissemination. The key idea of random network coding is that no matter how many coded packets you have collected so far, any new random coded packet is “helpful” with high probability. This claim has been proven in [3], and for the sake of completeness, we include the result in Lemma 2. Suppose that we use random linear coding with a finite field of size  $q$ . As mentioned earlier, a coded packet includes a code-vector as well as coded data. In order to decode packets, one must collect  $N_d$  independent code-vectors (i.e., rank  $N_d$ ).

**Lemma 2** *Suppose node v transmits a coded packet to node u. Let  $S_u^-$  and  $S_v^-$  denote the subspaces spanned by the code-vectors with u and v respectively at the beginning. Let  $S_u^+$  denote the subspaces spanned by the code-vectors by u after receiving a coded packet from v. Then,*

$$Pr(\dim(S_u^+) > \dim(S_u^-) | S_v^- \not\subseteq S_u^-) \geq 1 - \frac{1}{q},$$

where  $q$  is the size of the field.

Let  $\tilde{T}_{N_d}^{R-NC}$  denote the number of IID Bernoulli trials with success probability  $p = 1 - 1/q$ . Trials with probability  $p$  are being continued until the  $N_d$ -th success. The distribution gives the probability that  $z$  experiments are needed to reach  $N_d$  successes, which is known as a negative binomial distribution as shown below.

$$Pr(\tilde{T}_{N_d}^{R-NC} = z) = \binom{z-1}{N_d-1} p^{N_d} (1-p)^{z-N_d}, \tag{16}$$

where  $z = N_d, N_d + 1, \dots$ . Thus, the expected number of trials is given as

$$\mathbb{E}[\tilde{T}_{N_d}^{R-NC}] = \frac{N_d}{p} = N_d \frac{q}{q-1}. \tag{17}$$

From Lemma 2, the actual success probability is greater than  $p$ , and thus,  $\mathbb{E}[\tilde{T}_{N_d}^{R-NC}]$  will be the upper bound of the mean number of trials:

$$N_d < \mathbb{E}[T_{N_d}^{R-NC}] \leq \mathbb{E}[\tilde{T}_{N_d}^{R-NC}]. \tag{18}$$

As the field size increases, (17) shows that the upper bound approaches the lower bound. For a large field size, we conclude that

$$\mathbb{E}[T_{N_d}^{R-NC}] \simeq N_d. \tag{19}$$

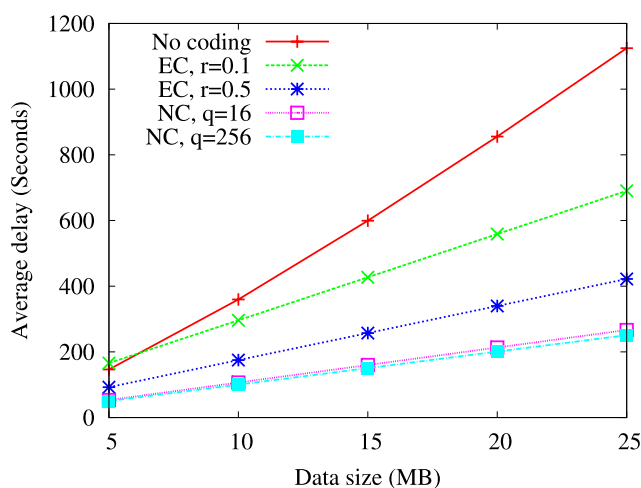
Therefore, the overall delay  $D_{R-NC}$  is given as

$$D_{R-NC} = \left\lceil \frac{N_d}{p e \bar{N}_p} \right\rceil \times \frac{\bar{T}_h}{2}. \tag{20}$$

Note that compared to R-WC where the delivery time is proportional to  $N_d \ln N_d$ , that of network coding is proportional to  $N_d$ . R-EC can achieve the same delay, but again with great cost. In conclusion, network coding provides a considerable improvement in the time required to propagate reliably via “data muling” critical video streams to disconnected platoons.

*Example 1* To give an intuitive idea of our approach we present a simple example. Let us say that vehicles drive with constant speed of 110 km/h (= 30 m/s). A vehicle sends packets over a 11 Mb/s radio channel with maximum throughput of 7.74 Mb/s (considering MAC/PHY overhead) and the communication range is 250 m. We assume that the freeway has a single lane (10 m wide) in both directions. The connection duration of two mobile vehicles in both directions is then approximately  $\frac{500 \text{ m}}{60 \text{ m/s}} = 8.3$  seconds. Assuming that the channel is fairly shared by 15 competing vehicles within the communication range, the effective capacity of the contact will be  $\frac{8.3 \text{ s/contact} \times 11 \text{ Mb/s} \times 0.7}{15 \times 8 \text{ bits}} \simeq 0.5$  MB/contact. Assuming that  $\lambda = 1/10$ , the average delay between two vehicles is 10 seconds; thus, for every 10 seconds, 512 KB of data is delivered to the destination. Let us also assume that  $p_e = 1$  such that no vehicle is exiting.

Let us then compare R-WC, R-EC, and R-NC in terms of delay. For R-EC we use  $r = 0.1, 0.5$  and for R-NC we use field sizes of  $q = 16, 256$  (4 and 8 bits respectively). We vary the size of video data to see the impact of delay. Figure 8 shows the delay. As expected, R-NC performs best of all the cases and moreover, even with a small field size of 16, we can obviously see the benefits of network coding. In the case of R-EC, we see that erasure coding improves the average delay as its redundancy factor increases from 0.1 to 0.5, but it is still inferior to network coding.



**Fig. 8** Comparison of three methods

## 6 Related work

Urban Multi-hop Broadcast (UMB) [10] supports directional broadcast in vehicular networks. UMB tries to improve reliability of broadcast by alleviating a hidden terminal problem through an RTS/CTS-style handshake, and broadcast storms through black-burst signals to select a forwarding node that is farthest from the sender using location information. Unlike UMB, Broadcast Medium Window (BMW) [23] and Batch Mode Multicast MAC (BMMM) [22] require all the receiving nodes to send back ACK to the sender in order to achieve reliability. BMMM has also adapted to directional MAC in vehicular networks [28]. However, all the previous work basically requires considerable amount of contention resolution time for each transmission, and thus, it is not suitable for real-time streaming which could potentially generate a large number of packets for a relatively short period of time. Our scheme cannot only handle this, but also efficiently deal with broadcast storm and reliability issues using random linear network coding. Moreover, our scheme can handle intermittent connectivity using the carry-and-forward “data muling” method.

At the applications level, several cooperative peer-to-peer type schemes have been proposed for vehicular environments. TrafficView [16] disseminates, or pushes (through flooding), information about the vehicles on the road, thus providing real-time road traffic information to drivers such as speed of vehicles. To alleviate broadcast storms, this work focuses on data aggregation/fusion based on the distance from the source. Vehicular Information Transfer Protocol (VITP) [4] provides on-demand, location-based, traffic-oriented services to drivers using information retrieved from vehicular sensors. A user can pull information from virtual ad hoc servers (i.e., collection of private vehicles) at the target location by sending a location-aware query. Similarly, V3 [5] supports a video request query (i.e., video trigger message) to the target location. Multiple vehicles at the destination could forward the video data to the query originator, and intermittent connectivity is handled by the carry-and-forward method. Our work is different from V3 in the following aspects. First, our protocol “pushes” urgent video streams regarding emergency situations such as natural disaster, traffic accidents, terrorist attacks, etc., which cannot be “pulled” by remote customers a priori. These emergency streams must be disseminated to other surrounding vehicles in order to help drivers effectively avert the danger. Second, our protocol exploits random linear network coding to provide reliable streaming. Finally, we address the carry-and-forward approach to deliver delayed streams to disconnected platoons. In the vehicle environment, the limited bandwidth allows only piecemeal transfer between vehicles traveling at high speeds in opposite directions, thus creating the video

data packets' collection problem (i.e., the coupon collecting problem [21]). However, random linear network coding elegantly solves this problem.

The effect of node mobility on vehicle communications was addressed in [29]. This work confirms the idea that as mobility increases, the number of encounters increases, whereas the duration of an encounter decreases. In this respect, Wu et al. [26] proposed an analytical model for information propagation delay of a single packet when the carry-and-forward approach is used. As shown in [29], it is better to employ multiple vehicles in parallel (when available) to deliver a relatively large video file using the carry-and-forward approach. Based on [29], we have developed a simple analytical model to estimate the file delivery delay using multiple, rather than a single, packet carrier. We show that network coding minimizes the delay.

By network coding, we refer to the notion of performing coding operations on the contents of packets throughout a network. This notion is generally attributed to Ahlswede et al. [1], who showed the utility of network coding for multicast. The work of Ahlswede et al. was followed by other work by Koetter and Médard [9] that showed that codes with a simple, *linear structure* were sufficient to achieve the capacity of multicast connections in lossless, wireline networks. This result was augmented by Ho et al. [6], who showed that, in fact, a *random construction* of the linear codes was sufficient.

The utility of such *random linear codes* for reliable communication over lossy packet networks—such as vehicular networks—was soon realized [14]. In [13], a prescription for the efficient operation of vehicular networks is given, which proposes using the random linear coding scheme of [14] coupled with optimization methods for selecting the times and locations for injecting coded packets into the network. This problem of selecting the times and locations for injecting packets is called *subgraph selection*. The prescription given in [13] allows potentially to find the optimal way of setting up a single connection. The optimal solution however may be complex, especially under the unpredictable characteristics and constraints imposed by vehicular networks. Previous works related to network coding's application to wireless networks include [7, 8] both of which focus on protocols solely for unicast and [25] which focuses on energy efficiency under a collision-free ideal MAC. Also in [20], the authors proposed a network coding based multicast protocol for ad hoc networks and showed how it can improve efficiency and reliability in such networks. The coding schemes and/or packet header formats used in [20, 25] are similar to the ones we use, which are variations of the standard techniques originally proposed in [2]. We show here how the techniques can be used in vehicular ad hoc networks for reliable multimedia type data dissemination and how they can be seamlessly extended to delay-tolerant operations. One of

the problems of random network coding is increased packet size. There are coding techniques that limit the impact of the overhead of network coding to the performance such as the one proposed by Maymounkov et al. [15].

## 7 Conclusions

In this paper we have considered the problem of disseminating emergency video streams to oncoming vehicles after an accident or a major disaster. To this end, we have proposed network coding and have evaluated its efficacy and suitability for this task. We have found via simulation that the video dissemination to vehicles connected to the source (i.e., single convoy) can benefit from network coding especially in fast mobility and when the radio channel is degraded by errors and interference. For example, for average speeds in the order of 40 m/s and channel drop rates of 10%, network coding offers 100% delivery ratio while one of the most robust multicast protocols, ODMRP, yields only 92% delivery ratio. The overhead also tends to be lower in network coding.

We also explored the problem of “delayed” delivery of the video file to disconnected convoys using the vehicles traveling in the opposite directions as “data mules.” Considering the fact that the opposite vehicles can randomly pick up (by “osmosis”) only a limited number of the packets at each pass, this problem was reformulated as a “coupon collector” problem. Using simple analytic models, we have shown that network coding outperforms two previously proposed schemes by featuring a delivery time linear with file size  $N$ , as opposed to  $N \log N$  as random picking without coding, or random picking with erasure coding. This result is important as it will enable much prompter situation awareness in both urban grid and intercity highways.

Future work in this specific area will be focused on the development of more accurate simulation models for single convoy as well as for multiple disconnected convoy dissemination. In the latter case, we will seek validation of our analytic approximations. We will also investigate the usefulness of network coding for a broader range of vehicle applications and scenarios beyond safe navigation.

On a broader scope, one can expect that in the future there will be increased interest in delivering videos using inter-vehicular communications because of the popularity of broadcasting user authored videos to other drivers (e.g., my experience in a museum down the road); and also distributing content downloaded from the Internet (e.g., soccer games). In these situations a critical issue will be the interplay between Internet and inter-vehicle since it is still challenging to provide high bandwidth Internet access to every vehicle in urban vehicular environments; e.g., for LTE, less than 10% of users can access videos at the rate of 500 Kbps, assuming a 10 MHz spectrum, 12 Mbps downlink, and 300

vehicles per sector. Given that the content generated on the roadside may be so “local” in terms of scope and interests, it may have limited monetary value, i.e., little opportunity to profit from associated advertising, which hinders mobile operators from additional investments (e.g., reserving extra spectrum or deploying temporary cell towers). Thus it will be better to keep the content entirely on vehicles. In the case of distributing popular content, it would be more economical to disseminate it via peer-to-peer schemes rather than to further stress the bandwidth limited infrastructure or wireless Internet. Vehicles could interact with pedestrians (e.g., wireless enabled hand-held devices), and the latter may become important video consumers; not to mention the fact that drivers eventually leave their cars and want to continue watching the soccer game. When hand-held devices interact with vehicles, naturally we face a new problem—limited energy. Thus, energy-efficient content distribution among pedestrians needs to be investigated. In all of these applications the issues of disruptive urban communications will be critical. We posit that the basic network coding and delay-tolerant networking concepts introduced in this paper will serve as a useful foundation for such expeditions.

**Open Access** This article is distributed under the terms of the Creative Commons Attribution Noncommercial License which permits any noncommercial use, distribution, and reproduction in any medium, provided the original author(s) and source are credited.

## References

- Ahlswede R, Cai N, Li S-YR, Yeung RW (2000) Network information flow. *IEEE Trans Inform Theory* 46(4):1204–1216
- Chou P, Wu Y, Jain K (2003) Practical network coding. In: Proceedings of the 51st Allerton conference on communication, control and computing, October 2003
- Deb S, Médard M, Chout C (2004) Algebraic gossip: a network coding approach to optimal multiple rumor mongering. In: Proceedings of the Allerton'04, Allerton, IL, September 2004
- Dikaiakos MD, Iqbal S, Nadeem T, Iftode L (2005) Vitp: an information transfer protocol for vehicular computing. In: VANET'05, September 2005
- Guo M, Ammar MH, Zegura EW (2005) V3: a vehicle-to-vehicle live video streaming architecture. In: PerCom'05, March 2005
- Ho T, Médard M, Koetter R, Karger DR, Effros M, Shi J, Leong B (2006) A random linear network coding approach to multicast. October 2006
- Katti S, Katabi D, Hu W, Rahul H, Médard M (2005) The importance of being opportunistic: practical network coding for wireless environments. In: Proceedings of the 43rd annual Allerton conference on communication, control, and computing, September 2005
- Khalili R, Salamatian K (2005) A new relaying scheme for cheap wireless relay nodes. In: Proceedings of the third international symposium on modeling and optimization in mobile, ad hoc, and wireless networks (WIOPT 2005), April 2005, pp 197–206
- Koetter R, Médard M (2003) An algebraic approach to network coding. *IEEE/ACM Trans Netw* 11(5):782–795
- Korkmaz G, Ekici E, Özgüner F, Özgüner U (2004) Urban multi-hop broadcast protocols for inter-vehicle communication systems. In: VANET'04, Philadelphia, PA, USA, October 2004
- Lee S-J, Su W, Gerla M (2002) On-demand multicast routing protocol in multihop wireless mobile networks. *ACM/Kluwer mobile networks and applications, special issue on Multipoint Commun Wirel Mob Netw*
- Lieu H (2003) Revised monograph on traffic flow theory. US Department of Transportation Federal Highway Administration
- Lun DS, Médard M, Koetter R (2005) Efficient operation of wireless packet networks using network coding. In: Proceedings of the international workshop on convergent technologies (IWCT), June 2005. Invited paper
- Lun DS, Médard M, Koetter R, Effros M (2008) On coding for reliable communication over packet networks. *Phys Commun* 1(1):3–20
- Maymounkov P, Harvey NJA, Lun DS (2006) Methods for efficient network coding. In: Proceedings of the 44th annual Allerton conference on communication, control, and computing
- Nadeem T, Dashtinezhad S, Liao C, Iftode L (2003) TrafficView: traffic data dissemination using car-to-car communication. *ACM Mob Comput Commun Rev* 8(3):6–19
- Nandan A, Tewari S, Das S, Gerla M, Kleinrock L (2006) Adtorrent: delivering location cognizant advertisements to car networks. In: Proceedings of the third annual conference on wireless on demand network systems and services WONS'06
- Scalable networks. <http://www.scalable-networks.com>
- Park J-S, Lee U, Oh SY, Gerla M, Lun DS, Ro WW, Park J (2008) Delay analysis of car-to-car reliable data delivery strategies based on data muling with network coding. *IEICE Trans Inf Syst* (10)
- Park J-S, Lun DS, Yi Y, Gerla M, Médard M (2006) Codecst: a network-coding-based ad hoc multicast protocol. *IEEE Wirel Commun* 13(5)
- Ross SM Introduction to Probability Models, 6th edn. Academic Press, New York (1997)
- Sun MT, Huang L, Arora A, Lai TH (2002) Reliable MAC layer multicast in IEEE 802.11 wireless networks. In: ICCP'02, Vancouver, August 2002
- Tang K, Gerla M (2001) MAC Reliable Broadcast in Ad Hoc Networks. In: Proceedings of the IEEE MILCOM'01, Washington, DC, October 2001
- Tsugawa S (2000) An introduction to Demo 2000: the cooperative driving scenario. *IEEE Intell Syst* 15(4):78–79
- Widmer J, Fragouli C, Le Boudec J-Y (2005) Low-complexity energy-efficient broadcasting in wireless ad hoc networks using network coding. In: Proceedings of the WINMEE, RAWNET and NETCOD 2005 workshops, April 2005
- Wu H, Fujimoto R, Riley G (2004) Analytical models for data dissemination in vehicle-to-vehicle networks. In: Proceedings of the VTC'04, September 2004
- Xu K, Gerla M, Bae S (2002) How effective is the IEEE 802.11 RTS/CTS handshake in ad hoc networks? In: Proceedings of the IEEE GLOBECOM'02
- Yadumurthy RM, Sadashivaiah ACHM, Makanaboyina R (2005) Reliable MAC broadcast protocol in directional and omnidirectional transmissions for vehicular ad hoc networks. In: Proceedings of the VANET'05, September 2005
- Yuen W, Yates R, Sung C (2003) Effect of node mobility on highway mobile infostation networks. In: Proceedings of the MSWiM'03, September 2003