Emergency Related Video Streaming in VANET using Network Coding

Joon-Sang Park, Uichin Lee,
Soon Y. Oh, Mario Gerla
UCLA Computer Science Department
{jspark|uclee|soonoh|gerla}@cs.ucla.edu

Desmond S. Lun
Coordinated Science Laboratory
University of Illinois at Urbana-Champaign
dslun@uiuc.edu

ABSTRACT

Multimedia (e.g., video) information exchange in VANET, if feasible, will help enhance vehicle navigation safety. We show that *network coding* allows very reliable and efficient data dissemination and thus is suitable for multimedia safety information dissemination. If the vehicle column has gaps, network coding jointly with "data muling" on vehicles in the opposite direction can deliver the multimedia files to disconnected components faster than other known schemes.

Categories and Subject Descriptors: C.2.1 [Network Architecture and Design]: Wireless communication

General Terms: Design, Reliability

Keywords: Multimedia Streaming, Network Coding, Data Mule

1. INTRODUCTION

Vehicular ad hoc networks (VANET) are becoming a reality driven mainly by navigation safety applications. Multimedia data (e.g., video), if feasible, will help enhance navigation safety. For example, videos clips of an accident or dangerous situation ahead will provide drivers with precise information. This will allow them to make a more informed decision (whether to proceed or turn back) based on personal priorities and/or on vehicle capabilities. Suppose that a critical traffic/safety situation occurs on a highway, say, major traffic congestion, natural disaster, fire, or terrorist attack. In such cases, video streaming could be triggered on one or more lead cars and propagated to vehicles following several miles behind - to *visually* inform the drivers of the problem and allow them to decide if they should turn around. Besides private vehicles, also first responders and rescue operations can greatly benefit from more prompt and precise situation awareness delivered by such video streams.

For the above warning systems to work, however, it is crucial that safety related messages be *reliably* delivered to all the *impacted* vehicles in the vehicular network so that they can cooperatively coordinate evasive actions. Thus far, a vast literature has been only focused on reliable data dissemination of short message, yet none on multimedia data. Multimedia files are innately large and thus, without proper controls, dissemination of such files can cause severe congestion and shut down other safety traffic.

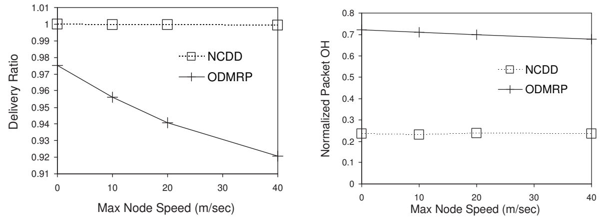
In vehicular networks, packets may be corrupted and lost due to channel errors and collisions. These type of packet losses tend to be random and diverse locally and thus can be countered efficiently with a local recovery strategy. To fight losses due to congestion and broken links, path diversity, i.e., the use of multiple, disjoint paths, paves an effective way. Path diversity is abundant in urban vehicular networks when they are densely packed, during peak hour say. We seeks to achieve reliable multimedia data dissemination exploiting localized neighbor recovery and path diversity. The main ingredient in our approach is *random network coding* [2] which transparently implements both localized recovery and path diversity with remarkably low overhead.

Cars may become separated on the highway, forming platoons. To bridge disconnected platoons, we propose to use the vehicles coming in the opposing directions as "data mules" (assuming that the highway has multiple lanes in both directions). This strategy speeds up the delivery of alarms with obvious navigation safety benefits. We show that network coding can be an efficient solution for reliable multimedia data delivery across multiple disconnected platoons as well as within a contiguous platoon.

VANET'06, September 29, 2006, Los Angeles, California, USA.
ACM 1-59593-540-1/06/0009.

2. RELIABLE MULTIMEDIA DELIVERY

Let us assume for simplicity that there is one single multimedia data source and it generates a stream of equal size frames $\mathbf{p}_1$, $\mathbf{p}_2$, $\mathbf{p}_3$, $\cdots$ where subscripts denote unique and consecutive sequence numbers. We abuse lowercase boldface letters to denote vectors or frames/packets and italics to denote variables or fields in the packet header. A *coded packet* $\mathbf{c}_{(bid,\, bsize)}$ is a linear combination of frames with sequence numbers in $[bid,\, bid + bsize)$ and can be represented as $\mathbf{c}_{(bid,\, bsize)} = \sum_{k=1}^{bsize} e_k \mathbf{p}_{(k-1+bid)}$. When generating such a coded packet, each e_k is drawn randomly from a finite field $\mathbb{F}$. Application frame $\mathbf{p}$'s and coded packet $\mathbf{c}$'s are also regarded as vectors over the field. In the header of a coded packet $\mathbf{c}_{(bid,\, bsize)}$, the encoding vector $\mathbf{e} = [e_1 \cdots e_{bsize}]$ is stored along with bid and $bsize$. Sending the encoding vector along with a coded packet was originally proposed in [1]. We say a application frame $\mathbf{p}_k$ belongs to $(bid,\, bsize)$ if its sequence number k is in $[bid,\, bid + bsize)$. When all the frames belonging to $(bid,\, bsize)$ are collected, the source generates and transmits to the neighborhood a coded packet $\mathbf{c}_{(bid,\, bsize)}$ $bsize$ times. On reception of a coded packet $\mathbf{c}_{(bid,\, bsize)}$, every node (or vehicle) stores the packet in its local memory. To recover $bsize$ original frames belongs to $(bid,\, bsize)$, a node should collect $bsize$ coded packets labeled $(bid,\, bsize)$ and encoding vectors that are linearly independent of each other. When a node receives a coded packet with a new tuple $(bid,\, bsize)$, it sets up a timer $\mathrm{T}_{(bid,bsize)}$ expiring in $btout$, say 0.2, seconds. When the timer expires, it broadcasts to its neighbors one *locally* encoded packet $\acute{\mathbf{c}}_{(bid,bsize)} = \sum_{k=1}^{rnk} \acute{e}_k \mathbf{c}_k$ carrying an encoding vector $\acute{\mathbf{e}} = \sum_{k=1}^{rnk} \acute{e}_k \mathbf{e}_k$ where $\mathbf{c}_k$ is a coded packet labeled $(bid, bsize)$ in local memory and $\mathbf{e}_k$ is the encoding vector prefixed to $\mathbf{c}_k$. rnk is the number of $\mathbf{c}_{(bid,bsize)}$'s found in local memory. When encoding, each $\acute{e}_k$ is again drawn uniformly from $\mathbb{F}$. $\mathrm{T}_{(bid,bsize)}$ is reset on expiration unless decodable



(a) Packet Delivery Ratio (b) Normalized Packet Overhead

Figure 1: Comparison of NCDD and ODMRP

set of coded packets labeled (bid, $bsize$) is collected. The number of frames/packets that are combined to yield a coded packet is recorded as rnk in the header of the packet. A coded packet $\mathbf{c}_{(bid,\, bsize)}$ with rnk smaller than $bsize$ indicates that the sender of $\mathbf{c}_{(bid,\, bsize)}$ is in need of more coded packets labeled (bid, $bsize$) for decoding. If a node receives such a packet **c**, it transmits another coded packet to help the sender of **c** collecting more coded packets. Every node broadcasts periodically (with a very long interval) the biggest bid of any coded packet in local memory. On reception of the biggest bid advertisement, a node compares its biggest bid to see if it has any missing blocks. If necessary, a node recovers the missing blocks from the advertiser (or other neighbors) by sending out a header only packet with rnk = 0 for a specific block. If the block organization is unknown, a node broadcast a header only packet with bid set to the biggest bid advertisement received, say M, received and $bsize$ set to the difference between its current biggest bid and the biggest bid advertisement received, say N, and neighbors respond if possible with coded packets corresponding to any application frame belonging to (M, N). Owing to the periodic advertisement and the recovery process multimedia data is delivered reliably across partitions, i.e., platoons.

3. PERFORMANCE EVALUATION

First, we study via simulation the performance of our Network Coding based Data Dissemination (NCDD). We use QualNet with the default 802.11b setting. One data source generates a constant $10Kbytes/sec$ traffic. Every node is the receiver. 200 vehicles are moving either in one direction or the opposite with different speeds along the $10km$ long $50m$ wide track. In NCDD, $bsize$ is set to 8. To simulate a lossy channel, nodes are forced to drop successfully received packets with probability 0.1. We contrast NCDD to ODMRP [3]. NCDD demonstrates near 100% data delivery regardless of mobility types and packet drop probability whereas ODMRP's packet delivery ratio degrades from 98% to 92% as mobility increase. More importantly, as shown in Fig 1(b), NCDD incurs less overhead than ODMRP. The reduction in overhead is as much as 70%. The normalized packet overhead in Fig 1(b) is defined as the total number of packets transmitted to the channel divided by the total number of data packets delivered.

Second, We study via analysis the highway "data muling" scenario where platoons of vehicles moving in the opposite direction on the highway act as "data mules." We compare the three following schemes.

- *Relay without coding (R-WC)* - A platoon passing by the accident site randomly picks up uncoded application frames and data-mules them to the disconnected target platoon.
- *Relay with erasure coding (R-EC)* - A source encodes application frames using a erasure coding scheme.
- *Relay with network coding (R-NC)* - A source encodes application frames using a random linear (network) coding similar to described in the previous section.

Vehicles arrive at independently distributed random intervals. In the traffic theory [4], this type of random arrival is often modeled using an exponential distribution with parameter λ (vehicles/second). Without loss of generality this can be extended also to a platoon arriving at the scene of the accident on the freeway. 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 are purely determined by the underlying Poisson arrival process.

Vehicles within a platoon may leave the highway. This depends on the density of ramps along the highway as well as the probability of defecting from a given platoon. For this reason, we introduce p_e denoting the effectiveness of a random platoon. For ease of analysis, we assume that a platoon can pick up on average $\overline{N}_p$ packets. The effective number of packets delivered to the target platoon is simply given as $p_e\overline{N}_p$. Let N_d denote the total number of packets for a given multimedia file that must be delivered. We assume that multiple number of platoons is required 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 $\overline{T}_h = 1/\lambda$. 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.

Our main result on average delivery delay of each scheme is given below. The detailed derivations can be found in the full version of this paper [5].

$$D_{\text{R-WC}} = \left\lceil \frac{N_d \ln N_d}{p_e \overline{N}_p} \right\rceil \times \frac{\overline{T}_h}{2} \quad (1)$$

$$D_{\text{R-EC}} = \left\lceil \frac{N_d (1+r) \ln(1+1/r)}{p_e \overline{N}_p} \right\rceil \times \frac{\overline{T}_h}{2} \quad (2)$$

$$D_{\text{R-NC}} = \left\lceil \frac{N_d}{p_e \overline{N}_p} \right\rceil \times \frac{\overline{T}_h}{2} \quad (3)$$

where r denotes the redundancy factor.

From the equations, we see that R-NC, the network coding approach, yields the lowest delivery delay among the three schemes. The key property of erasure coding is that the original data can be reconstructed from any of N_d packets out of $N_d(1+r)$ coded packets. Both R-WC and R-EC are analogous to a coupon collection problem where 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 help us to attain the lowest bound.

4. CONCLUSIONS

We have considered the problem of disseminating emergency video streams to vehicles following an accident. Simulation results showed that the video dissemination to vehicles connected to the source could benefit from network coding especially in fast mobility and degraded radio channel scenarios. Also, our mathematical analysis showed that network coding reduced delay in delivery across platoons via "data muling."

5. REFERENCES

[1] P. Chou, Y. Wu, and K. Jain. Practical network coding. In *Proc. Allerton Conf. Communication, Control and Computing '03*.

[2] T. Ho, R. Koetter, M. Médard, D. R. Karger, M. Effros, The Benefits of Coding over Routing in a Randomized Setting. In *Proc. IEEE ISIT '03*

[3] S.-J. Lee, M. Gerla, C.-C. Chiang. On-Demand Multicast Routing Protocol. In *Proc. of IEEE WCNC '99*.

[4] H. Lieu. *Revised Monograph on Traffic Flow Theory*. U.S. Department of Transportation Federal Highway Administration, 2003.

[5] J.-S. Park, U. Lee, S. Oh, D. Lun, M. Gerla. Emergency Related Video Streaming in VANETs using Network Coding. UCLA CSD Technical report, 2006.