# Error Resilient Real-Time Multimedia Streaming over Vehicular Networks

**Marco Pasin, Matteo Petracca, Paolo Bucciol†, Antonio Servetti, Juan Carlos De Martin**

**Dipartimento di Automatica e Informatica, †National Research Council of Italy**
**Politecnico di Torino – Corso Duca degli Abruzzi 24, I-10129 Torino, Italy**
**E-mail: *name.surname@polito.it***

**Abstract** This paper focuses on the optimized transmission of multimedia sequences over vehicular ad-hoc networks (VANETs). Among the main drawbacks of the high node mobility of the vehicular scenario are the high channel loss rate, burst length and intermittent connectivity. To allow real-time streaming of media sequences in such a scenario, we propose a technique based on forward error correction (FEC) and interleaving algorithms able to recover most of packet losses and reduce the length of the error bursts.
The proposed technique is based on the optimization of the FEC/Interleaving parameters under real-time constraints. It is implemented at packet level, to allow a straightforward adoption in the existing wireless devices. By resorting to standard compliant, real-time RTCP reports, we also develop an adaptive technique able to adapt to fast channel variations and to optimize both the overhead required by the proactive error recovery scheme and the additional delay introduced by the interleaver. Simulations based on real world experiments show that the proposed technique is able to gain over 2 dB in terms of video PSNR with respect to the standard transmission, while in the adaptive case the gain is over 2.5 dB PSNR, with a total overhead of about 50%.

**Keywords** Multimedia streaming, Interleaving, Forward Error Correction (FEC), Real-time Transport Protocol (RTP), Vehicular Ad hoc Network (VANET)

## 1 Introduction

The strong evolution of inter-vehicle communications in the Intelligent Transport System (ITS) sector, along with the widespread adoption of portable devices equipped with IEEE 802.11 wireless interfaces, fosters the deployment of innovative wireless communication services based on the real-time streaming of multimedia flows. Inter-vehicle multimedia streaming has countless applications, ranging from safety services, to collaborative driving and generic value added services such as advertising and infotainment.
However, the high variability of inter-vehicle communication channels based on the IEEE 802.11 standard makes the transmission of real-time multimedia information a very challenging problem. Among the main drawbacks of streaming applications over VANETs there is the high percentage of packet losses which can be experienced over the wireless channel. Even if multimedia information is tolerant to some packet losses, high losses do not actually allow the faithful reconstruction of the media with respect to its original version, thus not guaranteeing the quality necessary for object and speech recognition algorithms.
In this paper we address the problem of protecting real-time multimedia communications over inter-vehicle networks to guarantee the quality necessary for sophisticated multimedia signal processing techniques.
Let us consider the following scenario, where a video-communication software is installed in two cars that are going over the same path. The front car is transmitting real-time video information to the back car. As cars move along the path, the wireless channel experiences noise due to environmental elements, thus suffering from multipath fading. It causes variable bit error rate, depending on several parameters such as the distance between the two cars, the presence of objects between the cars and the relative speed. Certain combinations of these parameters can also generate very long bursts of packet losses resulting in intermittent connectivity. When it happens, the real-time transmission of the considered video flow is unfeasible, unless appropriate counteractions are taken.
The packet-level Forward Error Correction (FEC) technique is able to recover packet losses without resorting to packet retransmission requests (which would generate too high delays in this real-time constrained scenario). Packets to be transmitted are grouped in blocks, and their loss can be recovered until the packet loss rate of a given block exceeds the percentage of redundant packets inserted. If made aware of the channel conditions, the sender can then adapt the percentage of FEC to match the actual channel conditions.
This mechanism works well with the assumption of uniformly distributed losses. However, VANET transmissions heavily suffer from burst losses that strongly impact on the possibility to recover data packets belonging to such bursts. To overcome this

problem, in this paper we resort to packet-level interleaving. With adequate constraints, we show that a joint FEC/Interleaving technique is able to consistently improve the transmission quality, while respecting real-time constraints. We then modify the proposed algorithms to dynamically adapt to channel variations. Finally, the proposed techniques are validated with simulations based on real-world experiments, showing consistent gains of up to more than 2.5 dB in PSNR.
The remainder of the paper is organized as follows. In Section 2 the principles of real-time multimedia streaming are presented. In Section 3 and 4 the building blocks of our solution are described, namely the FEC and interleaving techniques, and non-adaptive optimal parameters are obtained. In Section 5 the proposed solution to the adaptive case is derived, and the performance of both the adaptive and the non-adaptive technique is evaluated. Finally, Section 6 concludes the paper with some considerations and remarks on future work.

## 2 Real-Time Multimedia Streaming

The requirements for real-time data transport mechanisms are distinctively different from those for traditional data communications. For example, real-time delivery requirements restrict the use of retransmissions to recover from packet losses so that the Transmission Control Protocol (TCP) is not suitable for this scenario. Instead, the real-time transport protocol, specified in RFC 3550 [1], is the *de facto* standard for delivering data with real-time content over IP networks.
To enable real-time transmission and playout at the receiver, the RTP packet header carries sensitive information such as the sequence number and the timestamp. An RTP packet may contain one or more codec frames, with the sequence number incrementing by one for each packet sent and the timestamp increasing at the rate of the sampling clock. An RTP receiver uses the sequence number to detect lost packets and the timestamp field to determine when to playout received data.
The RTP Control Protocol (RTCP) is used to monitor the quality of service and to convey information about the participants in an on-going session. Basically, RTCP carries long-term statistic information (e.g., mean packet loss rate (PLR), round trip time, jitter, etc.) related to the RTP session participants.

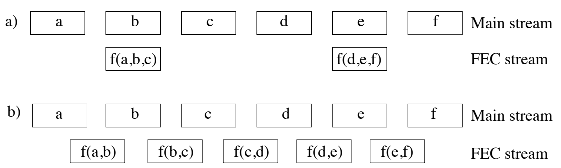


**Fig.1. Two basic sample schemes using generic FEC as defined in RFC 5109.**

In this work we discuss how RTCP reports can support RTP transmission to track frequent variations of the wireless channel in order to provide the streaming server with regular feedbacks from the receiver on the suffered packet loss rate. Timely feedback is used, at the sender, to adapt the transmission policy to the channel characteristics in order to achieve the best video quality as perceived by the end user. Error control techniques are introduced to improve the communication reliability against time-varying and bursty packet losses. In fact, IEEE 802.11 link-layer retransmissions are efficient only on a shorter time-scale and in the face of short-term fluctuations (fast fading); more persistent fluctuations (slow fading) in an high-mobility scenario render these mechanisms inefficient. Application-level error control techniques may provide additional reliability on a longer timescale and, as described in the next sections, cross-layer integration can be exploited to regulate the trade-off between error control aggressiveness and transmission overhead according to the channel loss trends reported by the RTCP protocol.

## 3 Forward Error Correction

Generic FEC is a codec independent method of protecting RTP payloads against packet erasures by adding redundant data to the transport stream. In this work we use a common method for generating FEC data that takes a set of packet payloads and applies the binary exclusive or (XOR) operation across the payloads. This scheme allows the recovery of missing data in the case where *one* of the original packets is lost, but the FEC packet is received correctly. The RTP payload format for using generic FEC based on XOR operations has been published in RFC 5109 [2].
In recent years, several proposals have been made to use well-known error correcting codes, such as Reed-Solomon codes, for packet loss recovery as well. However, the weakness of the more complex schemes is the computational complexity, which may cause performance problems with long packets and a large number of parity packets. This is why we limit the scope of this paper to XOR-based FEC codes only. Nevertheless, the basic principles discussed are easily convertible for other kinds of linear codes.
Figure 1 shows two basic schemes using the generic FEC defined in RFC 5109. In this paper, we adopt the definition of function $f(x, y, ...)$ to denote the resulting FEC packet when the XOR operation is applied to the packets $x, y, ...$ . In example (a) a single packet loss every three packets (in the original media stream) can be recovered, and in example (b) every packet loss can be recovered, assuming that the FEC stream is received correctly in both cases.

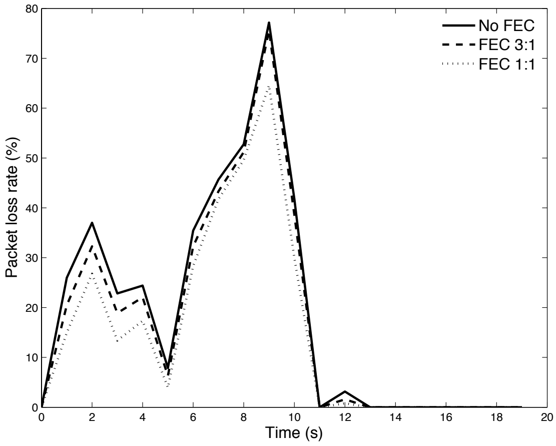


**Fig.2. Application level packet loss rate as a function of time for two generic XOR FEC schemes compared to the case of no FEC. FEC overhead is 100% for FEC 1:1 and 33% for FEC 3:1.**

Clearly both schemes require more network bandwidth because of the redundancy overhead. Example (a), that is denoted FEC 3:1, introduces an overhead of 33% since a FEC packet is sent every three data packets, while example (b), that is denoted FEC 1:1, introduces an overhead of 100%. In general, a FEC i:1 introduces a FEC packet every i data packets, causing an overhead of *(100/i)*%.

In practice, the media stream and the FEC stream are usually transmitted using the same transport medium. This is why we cannot expect packet losses to occur only in the media stream as both streams are likely to suffer from similar error characteristics. In the network perspective, it is realistic to assume the media stream and the FEC stream to form a single stream containing both media and FEC packets. Given a sequence of media and FEC packets, we can easily see the variation in error recovery rates when we examine the residual media data loss rate after applying different kinds of FEC patterns to the sequence. In Fig. 2 we plot the packet loss rate at the network level for a wireless intervehicle transmission trace together with the application level data loss rate for FEC examples (a) and (b). Clearly the more overhead is introduced, the more media data loss rate decreases. Nevertheless, the loss rate reduction is lower than expected. This is because the high packet loss rates of wireless transmission usually occur through correlated (adjacent) packet losses. In this case, the loss distribution (i.e., loss pattern) is a key parameter that determines the FEC performance. Clustered losses considerably reduce the efficiency of FEC and decrease the decoding quality. It is clear that the packet loss rate at the application level does not depend only on the packet loss rate, but also on *which* packets are lost.

A method that can be used to tackle this problem is to use interleaving to spread adjacent frames in different packets [3] as described in the next section.

## 4 Packet Interleaving

We explore a simple packet interleaving scheme to convert burst losses into an equivalent number of isolated losses which are easier to recover from using forward error control. Compared to other types of error-resilience techniques, packet interleaving provides the advantages of (1) being simple and (2) not requiring any increase in bitrate. Furthermore, packet interleaving can easily be coupled with FEC techniques.

A potential drawback of packet interleaving is that it requires additional delay. Interleaving delay is of particular concern in high interactive applications, such as Internet telephony, that cannot tolerate a delay above 400 ms [4]. However, the required delay, which depends on the channel burst length characteristics, can generally be bound to relatively short values, so even in this kind of applications, the end-to-end delay introduced by this technique is usually acceptable. Since many approaches for interleaving exist, we introduce the specific packet interleaving strategy used in this paper.

A simple packet interleaver that permutes the packet transmission order is represented in Fig. 3. At the sender, packets are first written into the interleaver in rows, with each row corresponding to a block of $n$ packets, among them $k = n - 1$ are data packets and the last one is a XOR-based FEC packet. Then the packets are transmitted by columns as soon as $m$ rows of packets fill up. At the receiver, when packets are reordered using their timestamp and sequence number, loss bursts are converted into separated losses. Let us consider, for example, the case of a transmission channel afflicted by a burst loss of length three. Using the *(n,m)* interleaver shown in Fig. 3 the burst loss affects separated packets 1, 4 and 7 instead of successive packets 1, 2 and 3.

The effectiveness of the interleaver depends on the block size and the interleaving depth of the interleaver, and on the loss characteristics of the channel. With an interleaving depth of $m$, a burst loss of length $B$ can be converted into a shorter burst with a maximum length of $\lceil B/m \rceil$, where $\lceil x \rceil$ denotes the smallest integer not smaller than $x$. In an ideal case, when $m \geq B$, the burst loss is converted into isolated losses.

| | | | | |
|---|---|---|---|---|
| | 1 [1] | 2 [4] | 3 [7] | f(1,2,3) [10] |
| m | 4 [2] | 5 [5] | 6 [8] | f(4,5,6) [11] |
| | 7 [3] | 8 [6] | 9 [9] | f(7,8,9) [12] |
| | | n | | |

**Fig.2. Packet interleaver with block size $n$ = 4 and interleaving depth $m$ = 3. Packets are transmitted by columns following the sequence numbers in brackets.**

In this case, the separation between any two losses is either $n$ or $n - 1$. A larger interleaver is more effective in that it can convert a longer burst loss into isolated losses or increase the separation of the converted isolated losses. However, this is at the cost of higher latency. At the client, an interleaved packet received cannot be used until all the packets it depends on are received. For a *(n,m)* interleaver, the n-th packet in the original order suffers from the highest delay, as it has to be transmitted in the $((n-1) \times m) + 1$-th place. Hence, the decoding delay corresponding to a *(n,m)* interleaver is

$$(n - 1) \times m \,, \tag{1}$$

and a trade-off exists between the effectiveness in permuting the packets and the latency. It should be noted that the total delay here is not the typical $n \times m$ which arises in channel coding situations, since we do not have the delay of applying FEC across the entire interleaved data [5].

Figure 4 illustrates the advantage of using different interleaving lengths for the same FEC scheme in the real wireless scenario considered in this paper and shown in Fig. 2. It is observed that the interleaver leads to lower packet loss rate by converting the burst losses into isolated losses so that the XOR FEC scheme can effectively recover the missing data packets. Note that at the network level the total number of losses is the same in both cases, the difference is only on the pattern of the losses. In addition we clearly see that after a certain interleaving depth there is nearly no gain in increasing the interleaver depth. This is because the interleaving depth is equal or greater than the mean burst length of the network channel and that this value is large enough to benefit from burst loss spreading.

In the next section we determine the optimal combination of FEC redundancy and interleaver length *(n,m)* under certain application-related delay constraints.

## 5 Experimental Procedures

Researchers have been working for long time to improve FEC based error control mechanism. The major research interest is still how to make the FEC code size adaptive instead of using a fixed FEC code under all communication environments.

Several works proposed an adaptive FEC schemes that adjust the code size according to an optimization model based on the assumption that the packet loss in a network follows a Bernulli process [6], a Gilbert-Elliot model [7], etc. However, the method of employing fixed models to determine the characteristics of wireless channels works reasonably well for an environment where the end nodes are fixed or have low mobility. For an environment that changes dynamically in time and speed, finding an appropriate model is still a major research issue.

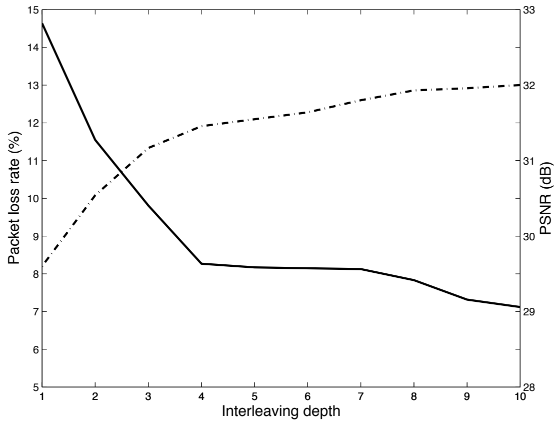


**Fig. 4. Application level packet loss rate and PSNR (dotted line) as a function of the interleaver depth with FEC 1:1 for the *Foreman* sequence and network trace of Fig. 2.**

So we propose an alternative solution, that is to use a feedback loop to determine the changing channel conditions and consequently to adjust the strength of FEC code depending on the notification of corrupted packets at the receiver end.

In the following we compare a fixed FEC scheme with our feedback-based adaptive FEC scheme through trace-based simulations. Loss patterns have been collected from real transmission experiments between two vehicles at the Nagoya University thanks to the equipment of the Center for Integrated Acoustic Information Research (CIAIR). An example trace has been presented in Fig. 2. Each transmitted RTP packet contains a sequence number that is intended for inter-media synchronization. These values are used at the receiver to calculate the loss pattern at the network level that is regularly sent back to the receiver by means of the RTCP reports. Based on the receiver feedback, the streaming server adaptively forges and transmits the FEC data along with the video stream.

### 5.1 FEC and interleaving depth adaptation

The proposed XOR-based adaptive FEC scheme uses the averaged loss rate $p$ reported periodically by RTCP to adjust the amount of redundancy (FEC) to be transmitted. XOR-based FEC protocol produces an additional redundancy packet from $k$ media packets and it has the capacity to overcome a single packet loss over the $n = k + 1$ consecutive packets. This provides resiliency against a maximum packet loss rate of $p = 1/n$ when considering that even FEC packets may be affected by loss. Thus, based on the averaged packet loss rate measurements such as that provided by the RTCP feedback, it is possible to constantly adjust the redundancy amount by changing the number of media packets ($k$) covered by the FEC packet as follows:

$$k = \left\lfloor \frac{1}{p} \right\rfloor - 1 \qquad (2)$$

The maximum acceptable loss rate threshold beyond which the streaming server triggers FEC adaptation may differ depending on the nature of the audiovisual content and its loss resiliency characteristics. In these simulations the threshold has been set to 10% so the maximum value of *k* has been set to 9, based on Eq. 2 .

The other dimension of the interleaving matrix, i.e., the number of rows (*m*), depends on the overall delay that can be tolerated by the real-time application. The total end-to-end delay consists of three components: the codec delay, the network delay and the playout delay. The latter is set according to the jitter introduced by the network transmission and, when interleaving is used, it should be increased so that it can accommodate also the interleaving delay. Playout buffer size is set by the receiver at the beginning of the transmission, before media decoding, and, in the simplest scenarios, it is usually kept constant. So, if we denote it by $d_{po} = d_j + d_i$ , where $d_j$ corresponds to the jitter component and $d_i$ to the interleaving component, the value of *m* can be dynamically calculated from Eq. 1 as a function of $d_i$ and *n* as:

$$m = \left\lfloor \frac{d_i}{n-1} \right\rfloor \qquad (2)$$

An additional issue that must be considered is that the FEC adaptation model poses a problem when dealing with channels that exhibit varying packet loss rates over time. The frequency of the receiver reports, which give to the sender an estimate about the network loss rate and other parameters, may reduce the responsiveness of the FEC scheme, leading to suboptimal FEC efficiency. A high frequency would enhance the responsiveness at the sender, while causing high variations between successive measurements and possibly leading to instability, not to mention excessive feedback traffic overhead. On the other hand, a low frequency would have good stability and low overhead but poor responsiveness.

In this work, RTCP report granularity has been assessed with a set of simulations to identify the best tradeoff between feedback overhead and server responsiveness and it has been set to 1s (that roughly corresponds to 127 data packets). Every second, the sender receives an RTCP packet with a report of the current PLR and it calculates the PLR estimate *p* for the subsequent time interval $\left(\hat{p}(i)\right)$ using the reported PLR value $\left(p(i-1)\right)$ and the previous PLR estimate $\left(\hat{p}(i-1)\right)$ according to:

$$\hat{p}(i) = \hat{p}(i-1) \times \alpha + p(i-1) \times \left(1-\alpha\right) , (4)$$

where the value of $\alpha$ has been experimentally chosen as the optimal first-order estimator so that it gives a good noise reduction ratio while maintaining a reasonable rate of convergence.

**5.2 Performance Evaluation**

The proposed system is evaluated using the Peak Signal to Noise Ratio (PSNR) to measure the objective quality of a reconstructed video stream as it is perceived by the end user. The PSNR value, in the following experiments, represents the distortion between the original video stream and the reconstructed video sequence at the receiver. It is shown that the PSNR is influenced not only by the PLR, but also by the loss pattern.

We used Common Intermediate Format (CIF – 352x288 pixels) H.264-coded [8] sequences with the GOP size set to 12 frames. In each GOP the first frame is intra-coded and all subsequent frames are coded as P-frames. In total four standard test sequences in CIF format are used, Foreman, Mother-Daughter, Salesman and Claire. Each sequence has 300 frames at 30 fps, and each is coded with a constant bitrate of 500 Kb/s which produces an average PSNR of 35.58 dB. The distortions are obtained by doubling the length of each sequence and averaging the results for 3 loss realizations shifted across the whole loss pattern, for a total of 600x3 loss realizations.

Two FEC schemes are considered. An *adaptive* scheme that is able to dynamically adapt the amount of redundancy and the interleaving delay according to the wireless channel conditions, and a *fixed* scheme that introduces the same overall redundancy and delay of the adaptive scheme, but constant over the whole length of the experiment. The value of α is set to 0.1. Both techniques reduce the average length of the error bursts from 4.5 packets (in the plain transmission case) to about 2.5 packets, supporting the capacity of the interleaver to convert error bursts into isolated losses. In Fig. 5 we plot the PSNR values of the received frames with respect to the original sequence (Foreman) for the network trace represented in Fig. 2.

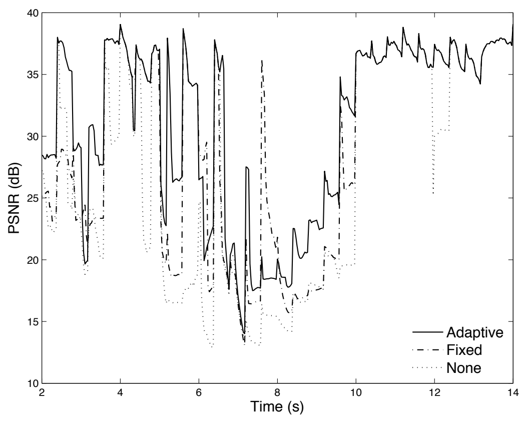


**Fig. 5. PSNR performance on the Foreman sequence for the network trace of Fig. 2 as a function of time for all the techniques.**

**Table 1. Performance comparison in terms of overhead, application level packet loss rate and PSNR for two joint FEC/interleaving schemes with respect to plain transmission (Foreman sequence).**

| Scheme | Overhead (%) | PLR(Appl.) (%) | PSNR (dB) |
|---|---|---|---|
| None | 0.00 | 18.82 | 28.24 |
| Fixed | 50.00 | 11.78 | 30.40 |
| Adaptive | 49.76 | 9.53 | 30.91 |

Channel loss characteristic clearly influences the quality perceived by the user so that the PSNR frequently drops of more than 5 dB in case of heavy losses. However, it is shown that the proposed feedback-based adaptive FEC scheme with interleaving is able to frequently avoid such quality drops. In particular, whenever the mean network PLR is roughly below 50% (see between 2 and 5 seconds and around 12 s), the perceived quality is noticeably increased. On the other side, when the PLR exceeds that threshold, the gain of few dB, e.g., from 15 to 20 dB, is practically hidden by the high level of distortion.

From Table 1 it is observed that the adaptive FEC scheme plotted in Fig. 5, that introduces an overhead of about 50% using a maximum value of $k$ equal to 9 and a delay bound of 33 ms for $d_i$, achieves, on a network trace with a PLR of 18.82%, a PLR reduction to 9.53% and a gain in PSNR of 2.67 dB The same scheme, compared to the fixed FEC scheme for the same network conditions with an equal amount of FEC overhead and delay, shows an appreciable PSNR gain of 0.51 dB.

## 6 Conclusions and future work

This paper presented an adaptive technique based on forward error correction and interleaving algorithms for real-time video streaming over VANETs. RTCP feedbacks are used to optimize the transmission parameters as a function of the fast channel variations in a typical vehicular scenario. Simulations based on real world experiments showed that this technique, with a total overhead of about 50%, can gain over 2.5 dB in terms of application PSNR with respect to standard transmission. Further improvements to the proposed technique can be implemented if the streaming server is made known of an accurate characterization of the loss process (i.e., with the information about the burstiness of losses). It is however necessary to use a richer feedback than what is supplied by conventional RTCP. Basically, a feedback mechanism is needed that provides information as to whether each packet is received or lost. Given this information, the FEC adaptation algorithm can capture the channel loss characteristics (including burst loss) and then it can more efficiently face them by adapting also the number of rows used in the interleaver to the channel state [9]. Future work will consider a comparison of the present work with a fine grained adaptation of the interleaver to the channel burstiness. In the literature has been reported in fact that, if the interleaver delay becomes too large, isolated losses of one burst loss may come closer to isolated losses corresponding to the next burst loss in the sequence thus leading to a reduction in the achievable gain. Real experiments will be set up as a part of the Vehicle-to-Vehicle-to-Infrastructure Communication for Sustainable Urban Mobility (VICSUM) project, led by Politecnico di Torino's Department of Electronics, to prove the effectiveness of such schemes in a real streaming implementation for an intervehicle scenario.

## 7 Acknowledgments

Authors would like to thank the NEDO project [10] and in particular prof. Takeda, prof. Abut and prof. Hansen, who placed their equipment at our disposal for the intervehicle trace collection phase at the Nagoya University.

## References

[1] Schulzrinne, H., Casner, S., Frederick, R., Jacobson, V., "RTP: A Transport Protocol for Real-Time Applications," RFC 3550(Standard),Jul.2003.

[2] Li, A., "RTP Payload Format for Generic Forward Error Correction," RFC 5109 (Proposed Standard), Dec. 2007.

[3] Perkins, C., Crowcroft, J., "Effects of interleaving on RTP header compression," in *Proc. of IEEE INFOCOM*, 2000, pp. 111–117.

[4] ITU-T Rec. G.114, "One way transmission time," Tech. Rep., ITU-T, 1993.

[5] Liang, Y.J., Apostolopoulos, J.G., Girod, B., "Model-based delay-distortion optimization for video streaming using packet interleaving," in *Proc. of Asilomar Conf. on Signals, Systems and Computers*, Nov 2002, vol. 2, pp. 1315–1319.

[6] Bolot, J.-C., Fosse-Parisis, S., Towsley, D., "Adaptive fec-based error control for internet telephony," in *Proc. IEEE INFOCOM*, Mar 1999.

[7] Yao, J., Huang, W.-F., Chen, M.-S., "DFEC: Dynamic forward error control for DVB-H," in *Proc. IEEE Intl. Conf. SUTC*, June 2006.

[8] ITU-T Rec. H.264 & ISO/IEC 14496-10 AVC, "Advanced video coding for generic audiovisual services," Tech. Rep., ITU-T, 2008.

[9] Nafaa, A., Taleb, T., Murphy, L., "Forward error correction strategies for media streaming over wireless networks," *Communications Magazine, IEEE*, vol. 46, no. 1, pp. 72–79, January 2008.

[10] Project NEDO, "Driving behaviour signal processing based on large scale real world database,": http://www.sp.m.is.nagoya-u.ac.jp/NEDO.