Multi-flow Block Interleaving applied to Distributed Speech Recognition over IP networks

Angel M. Gómez, Juan J. Ramos-Muñoz, Antonio M. Peinado, Victoria Sánchez

Department of Signal Theory, Networking and Communications
University of Granada, Spain

amgg@ugr.es

Abstract

Interleaving has shown to be a useful technique to provide robust distributed speech recognition over IP networks. This is due to its ability to disperse consecutive losses. However, this ability is related to the delay introduced by the interleaver. In this work, we propose a novel multi-flow block interleaver which exploits the presence of several streams and allows to reduce the involved delay. Experimental results have shown that this interleaver approximates the performance of end-to-end interleavers but with a fraction of their delay. As disadvantage, this interleaver must be placed in a common node where more than one flow are available.
Index Terms: distributed speech recognition, IP networks, interleaving, active networks.

1. Introduction

Since its beginning, Internet has been growing in size, incorporating many new networks, as well as in functionality, adding new services. As many other features have been integrated into Internet, such as mailing, instant messaging, telephony and so on, speech enabled services (SES) are also being incorporated. These services provide ubiquitous speech recognition, allowing multiple users to remotely access and share high performance recognition engines.

A very attractive approach to speech recognition over IP networks is the distributed speech recognition (DSR) solution [1]. As many other services over Internet, it is based on a client-server architecture. On one hand, a simple and low power client (*front-end*) analyzes, quantizes and packetizes speech data and sends it over the communication channel. On the other hand, a remote server (*back-end*) receives the data and performs speech recognition. Only those parameters which are relevant to the recognition process are transmitted through the channel. Thus, the required bandwidth is significantly reduced.

Packet losses, generally due to congestion, characterize most packet switched networks and can introduce significant limitations on performing DSR. In order to improve the robustness against packet losses, some recovering techniques has been proposed [2, 3, 4]. However, packet losses tend to appear in bursts and, in DSR, this burst-like nature causes the most negative impact. Indeed, DSR has shown to be tolerant to high loss ratios ($\sim$50%) as long as the average burst length is reasonably short (one or two frames) [5].

Based on this fact, techniques which reduce the burst length observed at the receiver could be applied. Interleavers are used to achieve this objective. The main disadvantage is that interleavers cause an increase of the end-to-end delay [6]. Although an additional delay of a few hundred milliseconds may not be significant to the overall quality of service, excessive delays (more than 500 ms [7]) degrades the naturalness of the human-machine interaction.

Work supported by MEC/FEDER project TEC2004-03829/TCM.

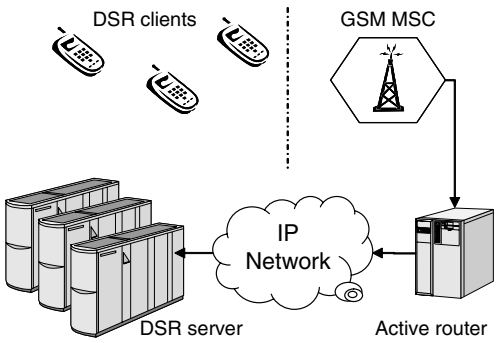

Figure 1: *Example architecture where multiple DSR clients use a common node at the Mobile Service Switching Center (*MSC*) that routes their streams toward a speech recognition server over an IP network.*

Interleaving is commonly applied at the sender, over just one user flow. However, it can be assumed an scenario where a group of local users requires SES services from a remote server. With this configuration, the interleaving process could be performed in some intermediate node, where more than one DSR flow would be available, reducing the interleaving delay. Examples of these scenarios are the users connected by local network that require SES services from a remote recognition engine, or the multiple users in a GSM cell performing DSR with a remote server through a mobile-IP connection (figure 1 illustrates this last example). It must be also considered the recently proposed *Active networks* [8], a novel approach to network architecture in which the switches of the network can perform customized computations on the packets flowing through them.

In this work we propose a block interleaver which exploits the presence of more than one DSR flow. This multi-flow interleaver [9] jointly reorders the sequence of frames from different streams, so that a reduction in the delay of the interleaving process is achieved. The proposed interleaver is evaluated in comparison with a reference scheme without interleaving, and with a single flow end-to-end interleaver. As we will show, our algorithm outperforms the reference system while approximates the results of the end-to-end interleaver but involving far less delay.

2. Experimental Framework

2.1. Front-end, Recognizer and Database

The experimental setup is based on the framework proposed by the ETSI STQ-Aurora working group [10]. On the client side, the Aurora DSR front-end segments the speech signal into overlapped frames of 25 ms every 10 ms. Every speech frame is represented by a 14-dimensional feature vector containing 13 MFCCs (including the 0th order one) plus log-Energy. These features are grouped into pairs and quantized by means of seven Split Vector Quantizers (SVQ). All codebooks have a 64-center size (6 bits), except the one for MFCC-0 and log-Energy, which has 256 centers (8 bits).

The recognizer is the one provided by Aurora [10] and uses eleven 16-state continuous HMM word models, (plus silence and pause, which have 3 and 1 states, respectively), with 3 gaussians per state (except silence, with 6 gaussians per state).

The speech data has been extracted from clean sentences of the Aurora-2 database (connected digits spoken by American English speakers). Training is performed from a set of 8440 utterances containing a total of 55 male and 55 female adult speakers, and test is carried out over the clean sentences of set *A*, with 4004 utterances distributed into 4 subsets.

2.2. Transmission and Channel Model

After the SVQ quantization, the bitstream is organized into a sequence of frame pairs encoded with 88 bits (44 bits per frame) followed by a 4-bit CRC. The first and second derivatives of the features are not transmitted, instead they are computed during recognition at the back-end. IP packets are generated according to the recommendations of the RTP payload format for DSR [11], where at least two frames (one frame pair) per packet must be transmitted in order to avoid too high a network overhead (due to headers). Following the RFC recommendations, only one frame pair per packet is sent.

A number of models of losses have been proposed in the literature but only a few of them model the burst of losses distribution and the interloss distance. In our transmission scheme, the channel burstiness exhibited by IP communications is modeled by a 6th order Markov chain. This Markov model has been trained with collected traces as described in [12]. The cumulative distribution functions (CDF) of burst length and interloss distance obtained from the trained model are shown in figure 2. The overall probability of loss is 8.2%. Since the transmission of multiple flows will be considered, the model does not indicate the loss of a packet. Instead it represents periods of losses or periods of receptions. All packets transmited during a period of losses will be assumed as lost.

The frame numbering included in the RTP header will be used to rearrange the packets received and to detect lost frames. At the receiver lost vectors are replaced by means of the Aurora standard mitigation algorithm. This can be summarized as follows: once a burst, containing $2B$ frames, is detected, the first B frames are substituted by the last correct frame before the burst and the last B ones by the first correct frame after the burst. In the case of a burst at the beginning of the utterance, the first correct frame after the burst is repeated backwards. A similar solution is applied for lost frames at the end of the utterance.

3. Interleaving applied to DSR

Interleavers can be used to permute the order in which speech feature vectors are transmitted. Thus, when they are put into their original order at the receiver, consecutive losses are distributed amongst many shorter bursts. Formally, an interleaver is a single input, single output finite-state device that takes sequences of symbols in a fixed alphabet and produces an output sequence over the same alphabet that is identical to the input sequence except for order. Given the input sequence $\ldots, a_{-2}, a_{-1}, a_0, a_1, a_2, \ldots$ and the output sequence $\ldots, b_{-2}, b_{-1}, b_0, b_1, b_2, \ldots$, interleaving can be expressed as a permutation $\pi : \mathcal{Z} \rightarrow \mathcal{Z}$ such that $a_i = b_{\pi(i)}$. In order to be realizable, this permutation must to be periodic, that is, $\pi(i + p) = \pi(i) + p$, where p is the period.

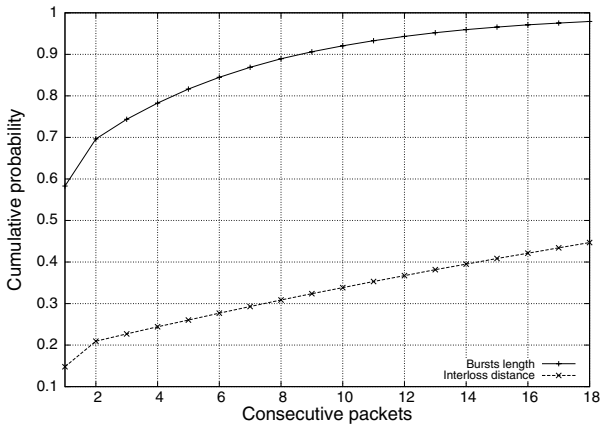

Figure 2: *Cumulative distribution functions burst length and interloss distance*

Interleaving causes a delay in the end-to-end transmission time. In order to be reordered, symbols need to be buffered prior to transmission. For an interlaver π, it can be defined δ, the maximum time delay between the arrival of an input symbol and the time it is produced as an output, as

$$\delta = max(\pi(i) - i). \tag{1}$$

Time restrictions are not so critical in speech recognition as in speech transmission. An immediate response from recognizer is not usually expected. However, interleaving delay should be kept small since responses excessively delayed are highly undesirable.

The ability of an interleaver to disperse consecutive errors is related to its *spread*. An interleaver π has spread s if:

$$|\pi(i) - \pi(j)| \geq s \qquad \text{whenever,} \qquad |i - j| < s. \tag{2}$$

That is, an interleaver with spread s reorders a sequence so that no contiguous sequence of s symbols in the reordered sequence contains any symbols that were separated by fewer than s symbols in the original ordering. As can be expected, large spreads are desirable, however, they entail longer delays.

3.1. Block Interleaving

A block interleaver of period N operates in blocks of N feature vectors, permuting these elements among themselves. A block interleaver successfully applied to DSR is the optimal delay block interleaver [5, 13]. The block interleaver of degree d operates by re-arranging the transmission order of a $d \times d$ matrix of input symbols. Two block interleavers are considered optimal in terms of maximising the spread of bursts for a given degree. They are given

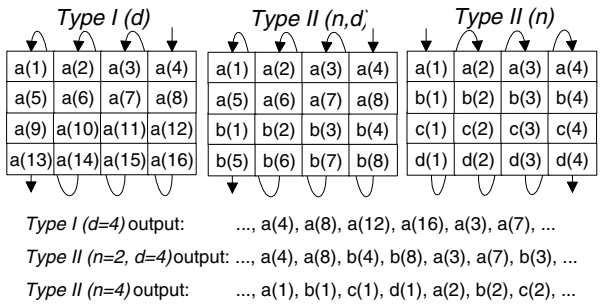

Figure 3: *Example outputs for the* Type I *and* II *interleavers.*

by,

$$\pi_1(id+j) = (d-1-j)d+i \qquad 0 \le i,j \le d-1, \tag{3}$$

$$\pi_2(id+j) = jd+(d-1-i) \qquad 0 \le i,j \le d-1. \tag{4}$$

These two interleavers form an invertible pair, that is, $\pi_1 = \pi_2^{-1}$ and $\pi_2 = \pi_1^{-1}$ and we will refer to both of them as *Type I* (d) interleaver (figure 3). This interleaver has an spread equal to its degree ($s = d$) and a corresponding delay of $\delta = d \cdot (d-1)$ frames.

A *Type I* (d) interleaver is limited in the sense that the introduced delay can make it potentially unfeasible to deal with large spreads and, therefore, with long bursts. Given an imposed end-to-end delay constraint, the degree of a *Type I* (d) interleaver is limited as

$$d \cdot (d-1) \cdot t_f < D_{max}, \tag{5}$$

where t_f is the generation period of symbols, and D_{max} is the maximum end-to-end delay that the system can tolerate. Assuming typical values $D_{max} = 250$ ms and $t_f = 10$ ms (frame rate), it can not be assured that bursts of length longer than 5 feature vectors will be completely scattered.

3.2. Multi-Flow Block interleaving algorithm

Assuming more than one DSR flow is available, we can interleave them jointly so that a reduction in the interleaver delay can be achieved. Based on this idea, let us suppose that $(f^1, f^2, \ldots, f^{n_f})$ are the n_f available DSR flows, and d is the degree of the block interleaver. To simplify, let us additionally define R_j^i, with $i = \{1, \ldots, n_f\}$ and $\{j = 1, \ldots, n_m\}$, as the number of consecutive rows that the flow f^i will be assigned for filling the interleaver matrix j, being n_m the number of matrices. Depending on n_f and d, we will consider two different cases.

1. Whenever $n_f \ge d$, the interleaver will be based on just one $(n_f \times 1)$ matrix ($n_m = 1$), in which $R_1^i = 1, \forall i = \{1, \ldots, n_f\}$. For this case, the interleaver output will be given by ..., f_i^1, f_j^2, ..., $f_k^{n_f}$, f_{i+1}^1, f_{j+1}^2, ..., $f_{k+1}^{n_f}$, ..., where the subscripts i, j, ..., k denote the sequence number for flows f^1, f^2, ..., f^{n_f}. As can be observed, no reordering is applied and frames are only time multiplexed ($\delta = 0$) toward the same output. For notation purposes, we will refer to this interleaver as *Type II* (n_f) interleaver (figure 3).
2. If $n_f < d$, we will refer to this interleaver as *Type II* (n_f, d) (figure 3). Under this condition, two different cases will be considered.
   - If $d = (n_f \cdot i), i \in \mathcal{N} \Rightarrow n_m = 1$. That is, only one interleaver with a $(d \times d)$ matrix will be used ($n_m = 1$);
   - Otherwise, n_f matrices of $(d \times d)$ will be required ($n_m = n_f$).

   Let us define $rem(x, y)$ as the remainder of the integer division x/y. Then, the n_m matrices will be filled as following:
   - For the first matrix, we will set $R_1^i = \left\lfloor \frac{d}{n_f} \right\rfloor$, for $i = \{1, 2, \ldots, (n_f - rem(d, n_f))\}$. Similarly, we will set $R_1^j = \left\lfloor \frac{d}{n_f} \right\rfloor + 1$, for $j = \{(n_f - rem(d, n_f) + 1), \ldots, (n_f - 1), n_f\}$.
   - If applicable, for the next $j = \{2, \ldots, n_f\}$ additional matrices, and for $i = \{2, \ldots, n_f\}$ flows, if $R_{(j-1)}^i = (\left\lfloor \frac{d}{n_f} \right\rfloor + 1)$ and $R_{(j-1)}^{(i-1)} = \left\lfloor \frac{d}{n_f} \right\rfloor$ then $R_j^i = \left\lfloor \frac{d}{n_f} \right\rfloor$ and $R_j^{(i-1)} = (\left\lfloor \frac{d}{n_f} \right\rfloor + 1)$.

3.3. Analysis of the Multi-Flow Interleaver

It can be demonstrated that the spread of the multi-flow interleaver is equal to its degree. As can be checked, bursts of length less or equal to d are scattered at the output of the interleaver by at least d feature vectors. Then, according to equation (2), $s = d$.

Attending to the maximum delay δ of the interleaver, let us define $r = rem(s, n_f)$ and $f = (s - r)/n_f$. Then, it can be shown that the delay caused by the multi-flow interleaver obey the following expressions,

- If $r \le (n_f - r) \Rightarrow$
  $\delta = d \cdot (r \cdot (f+1) - 1 - (r-1) \cdot f)$.
- If $r > (n_f - r) \Rightarrow$
  $\delta = d \cdot (r \cdot (f+1) - 1 - ((r-1) \cdot f + 2 \cdot r - n_f - 1))$

For a given interleaver degree d, the lowest maximum delay that can be obtained is achieved when either $n_f = (d-1)$ or $s/n_f = 2$ and $r = 0$. In this case, the delay corresponds to $\delta = d = s$. Thus, the maximum tolerated degree d, given a flow with a maximum allowed delay D_{max} and a period of t_f must satisfy,

$$d \cdot t_f < D_{max}. \tag{6}$$

which, as can be observed, is significantly less demanding in comparison with expression (5) corresponding to the *Type I* (d) interleaver.

Finally, the period of the proposed *Type II* (n_f, d) interleaver is given by,

$$p = \begin{cases} d^2/n_f & \text{if } d \equiv 0 \ (mod\, n_f) \\ d^2 & \text{otherwise} \end{cases} \tag{7}$$

4. Results

Although it has been shown that the *Type II* (n_f, d) interleaver provides the same spread than *Type I* (d) but involving considerably less delay, experimental results are provided to evaluate its performance in DSR. A simple scenario with n_f periodic flows is set. DSR frames from clients arrive to the active router with period equal to $t_f = 10$ ms (as the front-end segments the speech signal). For the *Type I* (s) case, just one flow ($n_f = 1$) is considered. We also assume no switching or any other routing delay.

Table 1 shows the word accuracy (Wacc) by applying a *Type II*(n_f, d) interleaver in comparison with a *Type I(d)* one and

	Type II(n_f)	*Type II(n_f, d)*			*Type I(d)*		
n_f	W_{acc}	d	W_{acc}	δ	d	W_{acc}	δ
2	94.25	*3*	95.03	3	*3*	96.78	6
		5	96.78	10	*5*	98.30	20
		6	97.29	12	*6*	98.56	30
		12	98.62	60	*12*	98.74	132
3	93.78	*7*	96.32	14	*7*	98.22	42
		9	97.09	18	*9*	98.48	72
4	93.65	*8*	95.75	8	*8*	97.98	56
5	93.50	*10*	95.49	10	*10*	97.31	90
6	93.39	*12*	95.20	12	*12*	96.75	132
7	93.02	*14*	94.78	14	*14*	96.09	182
8	93.01	*9*	93.36	9	*9*	95.61	72
9	92.67	*10*	93.07	10	*10*	94.91	90
10	92.45	*11*	92.87	11	*11*	94.24	110

Table 1: *Word accuracy (Wacc) obtained, and delay (δ) involved, by applying a* Type II(n_f, d) *interleaver in comparison with a* Type I(d) *one and no interleaving (*Type II(n_f)*).*

no interleaving (*Type II(n_f)*). Also in this table are given the delays (δ), expressed in number of frames, involved by each interleaver. Note that for the $TypeII(n_f)$ interleaver, δ is not shown since it is equal to 0.

As can be observed, $TypeII(n_f, s)$ interleaver provides better results than those obtained with no interleaving ($TypeII(n_f)$ interleaver). However, these are slightly worse than those obtained using $TypeI(s)$ interleaving. In this sense, $TypeI(s)$ interleaver outperforms both $TypeII$ interleavers. However, the delay imposed by this interleaver is prohibitive for degrees grater than 6. On the contrary, the delay imposed by $TypeII(n_f, s)$ interleaver is far less and greater degrees can be used.

5. Conclusions

This work is focused on the interleaving of multiple flows as a technique to improve the robustness of distributed speech recognition against packet losses on IP networks. Packet switched networks are characterized by packet losses which tend to appear in bursts. This burst-like nature has an important negative impact on the performance of DSR.

Interleaving can be especially useful in IP networks since it spreads consecutive losses into shorter bursts. In this work we have evaluated the optimal delay block interleaving and, as expected, improvements on the speech recognition accuracy has been obtained. However, these improvements have been achieved at the cost of increase the delay. Although delay is not as important in speech recognition as in speech transmission, it must be kept under reasonable limits.

In this paper we propose an interleaving algorithm which provides the same spread than optimal delay block interleavers but diminishing the delay per packet. This is possible by jointly interleaving packets from different flows so, to work properly, the interleaver must be placed in a common node before the path where losses are expected to occur. Traditionally, interleaving has been applied at the sender where only one flow is avaliable. However, multiple configurations can be described where interleaving could be applied in an intermediate node. In addition, the novel concept of active networks, where the switches of the network perform customized computations on the messages flowing through them, opens up a number of possibilities for this kind of interleaving.

The proposed multi-flow interleaving is tested with a realistic loss model trained with collected traces. As it has been shown, the optimal delay block interleaver provides slightly better results. However, the delay introduced by this interleaver can turn up prohibitive. In contrast, the multi-flow interleaver involves a more reasonable delay and, at the same time, provides significant improvements in comparison with a plain scheme without interleaving.

6. References

[1] D. Pearce: "Enabling New Speech Driven Services for Mobile Devices: An overview of the ETSI standard activities for Distributed Speech Recognition Front-ends". *AVIOS 2000: The Speech Aplications Conference*, San Jose (USA), May 2000.

[2] A.M. Gómez, A.M. Peinado, V. Sánchez, A.J. Rubio: "A Source Model Mitigation Technique for Distributed Speech Recognition over Lossy Packet Channels", in *Proc. Eurospeech*, 2003.

[3] A.M. Gómez, A.M. Peinado, V. Sánchez, A.J. Rubio: "Combining media-specific FEC and error concealment for robust distributed speech recognition over loss-prone packet channels", *IEEE Trans. on Multimedia* (In Press).

[4] A. M. Gómez, A. M. Peinado, V. Sánchez, B. P. Milner, A. J. Rubio: "Statistical-based Reconstruction Methods for Speech Recognition in IP Networks", in *Procs. Cost 278 and ITRW workshop*, 2004.

[5] B. Milner and A. James, "Robust Speech Recognition over Mobile and IP Networks in Burst-Like Packet Loss", *IEEE Trans. Speech and Audio Processing*, January 2006.

[6] K. Andrews, C. Heegard and D. Kozen: "A theory of interleavers", Technical report 97-1634, Computer Science Department, Cornell University, June 1997.

[7] C. Boulis, M. Ostendorf, E.A. Riskin, S. Otterson: "Graceful Degradation of Speech Recognition Performance Over Packet-Erasure Networks", *IEEE Transactions on Speech and Audio Processing*, Vol. 10, No. 8, November 2002.

[8] D.L. Tennenhouse, J.M. Smith, W.D. Sincoskie, D.J. Wetherall, G.J. Minden: "A survey of active network research", *IEEE Communications Magazine*, vol. 35, no. 1, 1997 Page(s): 80-86.

[9] J.J. Ramos-Muñoz, J.M. Lopez-Soler: "Low Delay Multiflow Block Interleavers for Real-Time Audio Streaming", *Lecture Notes in Computer Science*, vol. 3420, Jan 2005, Page(s): 909-916.

[10] D. Pearce, H. Hirsch: "The Aurora experimental framework for the performance evaluation of speech recognition systems under noisy conditions". in *Proc. ICSLP*, vol. 4, pp. 29-32, Beijing, China, October 2000.

[11] "RTP Payload Format for DSR ES 201 108", IETF Audio Video Transport WG, RFC3557, July 2003.

[12] M. Yajnik, J.Kurose, D. Towsley: "Packet loss correlation in the MBone multicast network experimental measurements and markov chain models." Tech. Rep. UM-CS-1995-115, 1995.

[13] B.P. Milner and A.B. James: "Analysis and Compensation of Packet Loss in Distributed Speech Recognition using Interleaving". in *Proc. Eurospeech*, 2003.