RMT	V. Roca
Internet-Draft	A. Roumy
Intended status: Experimental Protocol	INRIA
Expires: April 26, 2012	B. Sayadi
	Alcatel-Lucent, Bell Labs
	October 24, 2011

The Generalized Object Encoding (GOE) LDPC-Staircase FEC Scheme

Abstract

This document describes a Generalized Object Encoding (GOE) FEC Scheme for the protection of one or multiple objects, in the context of a Content Delivery Protocol (CDP) like FLUTE/ALC, FCAST/ALC or FCAST/NORM. Unlike [RFC5052], the GOE approach [GOE] decouples the definition of Generalized Objects over which FEC encoding takes place homogeneously, from the natural source object boundaries. This separation enables either an Unequal Erasure Protection (UEP) of different portions of a given source object, or an efficient and global protection of a set of potentially small files, depending on the way the Generalized Objects are defined.

The present document defines the GOE LDPC-Staircase FEC Scheme, i.e., the GOE version of the FEC Encoding ID 3 (LDPC-Staircase) defined in [RFC5170] with the further restriction that the number of encoding symbols per group (i.e., the number of symbols sent in the same packet) MUST be equal to 1 (G=1). This document does not change the LDPC-Staircase code definition, and therefore it inherits most of [RFC5170]. It only modifies the FEC Payload ID and FEC OTI, i.e., it addresses the problem of UEP and efficient file bundle protection by means of pure signaling approach.

Table of Contents

1. Introduction

1.1. Traditional FEC Schemes, as per [RFC5052]

The use of Forward Error Correction (FEC) codes is a classic solution to improve the reliability of unicast, multicast and broadcast Content Delivery Protocols (CDP) and applications [RFC3453]. The [RFC5052] document describes a generic framework to use FEC schemes with objects (e.g., files) delivery applications based on the ALC [RFC5775] and NORM [RFC5740] reliable multicast transport protocols.
More specifically, the [RFC5053] (Raptor) and [RFC5170] (LDPC-Staircase) FEC schemes introduce erasure codes based on sparse parity check matrices for object delivery protocols like ALC and NORM.
Similarly, the [RFC5510] document introduces Reed-Solomon codes based on Vandermonde matrices for the same object delivery protocols.
The way these FEC schemes is used leads to two limitations. First of all, [RFC5052] defines an approach where the same FEC encoding is applied to all the blocks of a given object, i.e., the whole object is encoded using the same FEC scheme, with the same target code rate, resulting in an equivalent protection. This approach may not suit situations where some subsets of an object deserve a higher erasure protection than the others.
A second limitation is associated to the protection of a large set of small objects. [RFC5052] defines an approach where each object is protected individually. This feature limits the robustness of their delivery: since there is a small number of source and repair packets for a given small object, a significant number of these packets may be erased thereby preventing this object to be decoded at a receiver. For instance, if the source and repair packets of a given object are transmitted in sequence (which may not be the best strategy), a packet erasure burst will significantly impact transmission robustness. Other transmission ordering strategies (e.g., with long packet interleavings or random ordering strategies) can reduce the impacts of packet erasure bursts, but they do not solve the fundamental problem of the protection of small objects. On the opposite a global FEC protection of all the objects of this set, using a single FEC encoding (when possible),

provides optimal transmission robustness, since all the objects can be decoded as long as the erasure rate remains lower than the protection brought by the FEC code rate.

1.2. GOE FEC Scheme Principles

In order to mitigate the limitations of the traditional FEC Schemes, a better approach consists in decoupling FEC protection from the natural object boundaries. This is the goal of the Generalized Object Encoding (GOE) approach [GOE]. The set of source objects is first encoded using the No-Code FEC Scheme [RFC5445]. Each source symbol of each source object is therefore individually identified by its {TOI (i.e., ALC or NORM object identifier); SBN (source block identifier); ESI (symbol identifier)} tupple. Each Generalized Object is then defined as a sequence of consecutive No-Code encoding symbols, that starts at a given symbol, identified by its {TOI, SBN, ESI} tuple, and that is composed of a given number of such symbols. Each Generalized Object is then FEC encoded using an appropriate FEC code, with an appropriate code rate. Of course a Generalized Object may be a subset of a given source object or at the opposite may encompass several source objects. The key point when defining Generalized Objects is that all the corresponding source symbols require an equal erasure protection.
The GOE approach is independent of the nature of the FEC code, in the sense that the general mechanisms it defines is not restricted to a single type of FEC code. On the opposite, the GOE approach can be associated to any of the existing FEC schemes, re-using their code definition. However a new FEC Encoding ID value, a new FEC Object Transmission Information (FEC OTI) and a new FEC Payload ID (FPI) must be defined in order to accommodate the GOE specifics. This means that a dedicated FEC Scheme must be defined. For instance, [GOE] defines the GOE Reed-Solomon FEC Scheme for the particular case of Reed-Solomon codes over GF(2^^8) and no encoding symbol group, the GOE equivalent to FEC Encoding ID 5 defined in [RFC5510].
The present document defines the GOE LDPC-Staircase FEC Scheme, i.e., the GOE version of the FEC Encoding ID 3 (LDPC-Staircase) defined in [RFC5170], with the further restriction that the number of encoding symbols per group (i.e., the number of symbols sent in the same packet) MUST be equal to 1 (G=1).
Please refer to [GOE] for the details on the GOE procedures at a sender and at a receiver. An evaluation of GOE can also be found in [GOE.RR7699]. Finally [GOEatIETF81] provides a high level overview of GOE.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

2.1. Definitions, Notations and Abbreviations

2.1.1. Definitions

This document uses the following terms and definitions. Some of them are FEC scheme specific and are in line with [RFC5052]:

Source Packet: a data packet containing only source symbols, that is sent over the packet erasure channel. Most of the time a source packet will contain a single source symbol.

Repair Packet: a data packet containing only repair symbols, that is sent over the packet erasure channel. Most of the time a repair packet will contain a single repair symbol.

Packet Erasure Channel: a communication path where packets are either dropped (e.g., by a congested router, or because the number of transmission errors exceeds the correction capabilities of the physical layer codes) or received. When a packet is received, it is assumed that this packet is not corrupted.

Systematic code: FEC code in which the source symbols are part of the encoding symbols. The Reed-Solomon codes introduced in this document are systematic.

Code rate: the k/n ratio, i.e., the ratio between the number of source symbols and the number of encoding symbols. By definition, the code rate is such that: 0 < code rate ≤ 1. A code rate close to 1 indicates that a small number of repair symbols have been produced during the encoding process.

Object: the object (e.g., file) submitted to the CDP by the user.

Generalized Object: a group of consecutive source symbols, that belong to one or several objects (as defined above) and that are considered together for the purpose of a GOE scheme. Generalized objects may be a subset of a given object or at the opposite encompass several objects. The key point when defining generalized objects is that all the source symbols of a generalized object require an equal erasure protection.

Source symbol: unit of data used during the encoding process. In this specification, there is always one source symbol per ADU.

Encoding symbol: unit of data generated by the encoding process. With systematic codes, source symbols are part of the encoding symbols.

Repair symbol: encoding symbol that is not a source symbol.

Source block: a block of k source symbols that are considered together for the encoding.

2.1.2. Notations

This document uses the following notations:

k denotes the number of source symbols in a source block.

n denotes the number of encoding symbols generated for a source block.

E denotes the encoding symbol length in bytes.

NO denotes the number of source objects to be considered.

2.1.3. Abbreviations

This document uses the following abbreviations:

ADU stands for Application Data Unit.

TOI stands for Transmission Object Identifier.

SBN stands for Source Block Number, i.e., a block identifier.

ESI stands for Encoding Symbol ID.

FEC stands for Forward Error (or Erasure) Correction code.

LDPC stands for Low Density Parity Check.

MDS stands for Maximum Distance Separable code.

UEP stands for Unequal Erasure Protection.

FEC OTI stands for FEC Object Transmission Information.

3. Formats and Codes with FEC Encoding ID XXX for LDPC-Staircase Codes

This section introduces the formats and codes associated with the Fully-Specified FEC Scheme with FEC Encoding ID XXX, which focuses on LDPC-Staircase Codes. This GOE FEC Scheme is the GOE equivalent to FEC Encoding ID 3 defined in [RFC5170], with the further restriction that the number of encoding symbols per group (i.e., the number of symbols sent in the same packet) MUST be equal to 1 (G=1).

3.1. FEC Payload ID (for Repair Packets Only)

The FEC Payload ID, to be used only with repair packets, i.e., packets containing a repair symbol each, is composed of the Source Block Number (SBN) and the Encoding Symbol ID (ESI). There is no change in terms of format with respect to [RFC5170] but a restriction in terms of valid ESI as explained below:

* The Source Block Number (12-bit field) identifies from which source block of the object the encoding symbol in the payload is generated. There is a maximum of 2^^12 blocks per object.

* The Encoding Symbol ID (20-bit field) identifies which specific encoding symbol generated from the source block is carried in the packet payload. There is a maximum of 2^^20 encoding symbols per block. The first k values (0 to k - 1) identify source symbols; the remaining n-k values (k to n-k-1) identify repair symbols. Since only repair symbols are considered by this GOE FEC scheme, only the k to n-k-1 values, inclusive, MUST be used.

There MUST be exactly one FEC Payload ID per repair packet (since G=1). This FEC Payload ID refers to the one and only symbol of the packet.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Source Block Number  |      Encoding Symbol ID (20 bits)     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

3.2. FEC Object Transmission Information

3.2.1. Mandatory Elements

* FEC Encoding ID: the Fully-Specified FEC Scheme described in this section uses FEC Encoding ID XXX.

3.2.2. Common Elements

The Common elements are the same as those specified in [RFC5170] for FEC Encoding ID 3, namely: the Transfer-Length (L), the Encoding-

Symbol-Length (E), the Maximum-Source-Block-Length (B), and the Max-Number-of-Encoding-Symbols (max_n). These common elements refer to the Generalized Object for which LDPC-Staircase encoding is needed.

3.2.3. Scheme-Specific Elements

The following element MUST be defined with the present FEC scheme. It defines the composition of a generalized object:

* N1m3: an integer between 0 (default) and 7, inclusive. The target number of "1s" per column in the left side of the parity check matrix, N1, is then equal to N1m3 + 3. See [RFC5170] for guidelines on how to set N1m3.

* G: in this specification, G MUST be equal to 1.

* the Initial Source Symbol TOI (ISS_TOI) identifies the TOI of the first source symbol of this generalized object. The exact format of this field depends on the TOI format, which is CDP and use-case specific. For instance the TOI field of an ALC session is stored in a field of length 32*O+16*H bits, where O and H are the TOI flag and Half-word flag defined in LCT's header;

* the ISS TOI size (ISS_O) two bit field determines the TOI size, which is equal to 32*ISS_O + 30 bits. This flexibility is meant to be compatible with any NORM or ALC TOI format;

* the ISS Source Block Number (ISS_SBN) identifies the SBN of the first source symbol of this generalized object, within its original object. This is a 16 bit field, since this value results from the No-Code FEC encoding of the original object;

* the ISS Encoding Symbol ID (ISS_ESI) identifies the ESI of the first source symbol of this generalized object, within its original block. This is a 16 bit field, since this value results from the No-Code FEC encoding of the original object;

* the Generalized Object Size (GOS) identifies the size, in terms of number of source symbols that compose this generalized object;

3.2.4. Encoding Format

This section shows the two possible encoding formats of the above FEC OTI. The present document does not specify when one encoding format or the other should be used.

3.2.4.1. Using the General EXT_FTI Format

The FEC OTI binary format is the following, when the EXT_FTI mechanism is used (e.g., within the ALC [RFC5775] or NORM [RFC5740] protocols).

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   HET = 64    |      HEL      |                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               +
|                      Transfer-Length (L)                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Encoding Symbol Length (E)  | N1m3|  G = 1  |    B (MSB)    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        B (LSB)        |    Max Nb of Enc. Symbols  (max_n)    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           PRNG seed                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|*_O|                                                           |
+-+-+         ISS_TOI (length = 32*ISS_O + 30 bits)             +
|                             ...                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    ISS Source Block Number    |    ISS Encoding Symbol ID     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|               Generalized Object Size (GOS)                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

3.2.4.2. **Using the FDT Instance (FLUTE specific)**

When it is desired that the FEC OTI be carried in the FDT Instance of a FLUTE session [FLUTE], the following XML attributes must be described for the associated object:

    *FEC-OTI-FEC-Encoding-ID

    *FEC-OTI-Transfer-Length (L)

    *FEC-OTI-Encoding-Symbol-Length (E)

    *FEC-OTI-Maximum-Source-Block-Length (B)

    *FEC-OTI-Max-Number-of-Encoding-Symbols (max_n)

    *FEC-OTI-Scheme-Specific-Info

The FEC-OTI-Scheme-Specific-Info contains the string resulting from the Base64 encoding (in the XML Schema xs:base64Binary sense) of the following value:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         PRNG seed                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|*_O|                                                           |
+-+-+         ISS_TOI (length = 32*ISS_O + 30 bits)             +
|                           ...                                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    ISS Source Block Number    |    ISS Encoding Symbol ID     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  Generalized Object Size                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| N1m3|  G = 1  |
+-+-+-+-+-+-+-+-+
```

During Base64 encoding, the FEC OTI Scheme-Specific Information (of variable length) is transformed into a string of printable characters (in the 64-character alphabet) that is added to the FEC-OTI-Scheme-Specific-Info attribute.

4. Procedures with FEC Encoding ID XXX for LDPC-Staircase Codes

This section defines procedures that MUST be applied to FEC Encoding ID XXX. The block partitioning algorithm that is defined in Section 9.1 of [RFC5052] MUST be used. The procedure called "Determining the Maximum Source Block Length (B)" in [RFC5170] MUST be used. The procedure called "Determining the Maximum Number of Encoding Symbols Generated for Any Source Block (max_n)" in [RFC5170] MUST be used. The procedure called "Determining the Number of Encoding Symbols of a Block" in [RFC5170] MUST be used. The procedure called "Identifying the G Symbols of an Encoding Symbol Group" in [RFC5170] MUST NOT be used, since this specification requires that the number of encoding symbols per group MUST be equal to 1 (G=1). The procedure called "Pseudo-Random Number Generator" in [RFC5170] MUST be used.

4.1. Determining the Encoding Symbol Length (E)

The E parameter usually depends on the maximum transmission unit on the path Maximum Transmission Unit (PMTU) from the source to each receiver. This PMTU may be known, may be discovered, or may be estimated, depending on the target use case. In order to minimize the protocol header overhead (e.g., the Layered Coding Transport (LCT), UDP, IPv4, or IPv6 headers in the case of ALC), E MAY be chosen to be as large as possible. In that case, E is chosen so that the size of a packet composed of a single encoding symbol remains below but close to the PMTU (or by the minimum PMTU to each possible destinations in case of one-to-many sessions). This value E is also the source symbol size

(i.e., the source symbols, before FEC encoding, and the encoding symbols, after FEC encoding, are of equal size).
This size MUST be used to segment all of the NO source objects considered by the GOE FEC schemes for this CDP into source symbols. By doing so, a Generalized Object that straddles several objects (among the NO possibles) benefits from the same source symbol size across source object boundaries.

5. Security Considerations

TBD

6. Operational Considerations

LDPC-Staircase codes have excellent erasure recovery capabilities with large source blocks, close to ideal MDS codes. For instance, with a medium source block size k=1024, CR=2/3, N1=5, G=1, with a hybrid ITerative/Maximum Likelihood (IT/ML) decoding approach (see below) and when all symbols are sent in a random order (see below), the average overhead amounts to 0.64% (corresponding to 6.5 symbols in addition to k) and receiving 1043 symbols (corresponding to a 1.9% overhead) is sufficient to reduce the decoding failure probability to 5.1*10^^-5.
LDPC-Staircase codes are also a good solution whenever processing requirements at a software encoder or decoder must be kept to a minimum. This is true when the decoder uses an IT decoding algorithm, or an ML algorithm (we use a Gaussian Elimination as the ML algorithm) when this latter is carefully implemented and the source block size kept reasonable, or a mixture of both techniques which is the recommended solution. For instance an average decoding speed between 1.3 Gbps (corresponding to a very bad channel, close to the theoretical decoding limit and requiring an ML decoding) and 4.3 Gbps (corresponding to a medium quality channel where IT decoding is sufficient) are easily achieved with a source block size composed of k=1024 source symbols, a code rate CR=2/3 (i.e., 512 repair symbols), 1024 byte long symbols, G=1, and N1=5, on an Intel Xeon 5120/1.86GHz workstation running Linux/64 bits. Additionally, with a hybrid IT/ML approach, a receiver can decide if and when ML decoding is used, depending on local criteria (e.g., battery or CPU capabilities), independently from other receivers.
As the source block size decreases, the erasure recovery capabilities of LDPC codes in general also decrease. In the case of LDPC-Staircase codes, in order to compensate this phenomenon, it is recommended to increase the N1 parameter and to use a hybrid IT/ML decoding approach. For instance, with a small source block size k=256 symbols, CR=2/3, N1=7, and G=1, the average overhead amounts to 0.67% (corresponding to 1.7 symbols in addition to k), and receiving 267 symbols (corresponding to a 4.3% overhead) is sufficient to reduce the decoding failure probability to 1.4*10^^-5. Using N1=9 further improves these results if

need be, which also enables to use LDPC-Staircase codes with k=100 symbols for instance.
With very small source blocks (e.g., a few tens symbols), using for instance Reed-Solomon codes [RFC5510] or 2D parity check codes MAY be more appropriate.
The way the FEC Repair Packets are transmitted is of high importance. A good strategy, that works well for any kind of channel loss model, consists in sending FEC Repair Packets in random order (rather than in sequence) while FEC Source Packets are sent first and in sequence. Sending all packets in a random order is another possibility, but it requires that all repair symbols for a source block be produced first, which adds some extra delay at a sender.
For further information, the interested reader can refer for instance to [Cunche08][CunchePHD10].

7. IANA Considerations

Values of FEC Encoding IDs and FEC Instance IDs are subject to IANA registration. For general guidelines on IANA considerations as they apply to this document, see [RFC5052].
This document assigns the Fully-Specified FEC Encoding ID XXX under the "ietf:rmt:fec:encoding" name-space to "Generalized Object Encoding for LDPC-Staircase codes".

8. Acknowledgments

TBD

9. References

9.1. Normative References

[RFC2119]	Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119, .
[GOE]	Roca, V., Roumy, A. and S. Bessem, "The Generalized Object Encoding (GOE) Approach for the Forward Erasure Correction (FEC) Protection of Objects and its Application to Reed-Solomon Codes over GF(2^^8)", Work in progress draft-roca-rmt-goe-fec, July 2011.
[RFC5170]	Roca, V., Neumann, C. and D. Furodet, "Low Density Parity Check (LDPC) Forward Error Correction", RFC 5170, June 2008.

9.2. Informative References

[RFC3453]	Luby, M., Vicisano, L., Gemmell, J., Rizzo, L., Handley, M. and J. Crowcroft, "The Use of Forward Error Correction (FEC) in Reliable Multicast", RFC 3453, December 2002.

[RFC5445]	Watson, M., "Basic Forward Error Correction (FEC) Schemes", RFC 5445, March 2009.
[RFC5052]	Watson, M., Luby, M. and L. Vicisano, "Forward Error Correction (FEC) Building Block", RFC 5052, August 2007.
[GOE.RR7699]	Roumy, A., Roca, V., Bessem, S. and R. Imad, "Unequal Erasure Protection and Object Bundle Protection with the Generalized Object Encoding Approach", INRIA Research Report RR-7699, http://hal.inria.fr/inria-00612583_v1/en/, July 2011.
[GOEatIETF81]	Roca, V., Roumy, A. and S. Bessem, "The GOE FEC schemes (draft-roca-rmt-goe-fec-00) and UOD-RaptorQ versus GOE", Slides presented during the RMT meeting at IETF81, http://www.ietf.org/proceedings/81/slides/rmt-2.pdf, July 2011.
[RFC5510]	Lacan, J., Roca, V., Peltotalo, J. and S. Peltotalo, "Reed-Solomon Forward Error Correction (FEC) Schemes", RFC 5510, April 2009.
[RFC5053]	Luby, M., Shokrollahi, A, Watson, M and T Stockhammer, "Raptor Forward Error Correction Scheme", RFC 5053, June 2007.
[RFC5740]	Adamson, B., Bormann, C., Handley, M. and J. Macker, "NACK-Oriented Reliable Multicast (NORM) Transport Protocol", RFC 5740, November 2009.
[RFC5775]	Luby, M., Watson, M. and L. Vicisano, "Asynchronous Layered Coding (ALC) Protocol Instantiation", RFC 5775, April 2010.
[FLUTE]	Paila, T., Walsh, R., Luby, M., Roca, V. and R. Lehtonen, "FLUTE - File Delivery over Unidirectional Transport", Work in Progress, February 2011.
[Cunche08]	Cunche, M. and V. Roca, "Optimizing the Error Recovery Capabilities of LDPC-Staircase Codes Featuring a Gaussian Elimination Decoding Scheme", 10th IEEE International Workshop on Signal Processing for Space Communications (SPSC'08), October 2008.
[CunchePHD10]	Cunche, M., "High performances AL-FEC codes for the erasure channel : variation around LDPC codes", PhD dissertation (in French), http://tel.archives-ouvertes.fr/tel-00451336/en/, June 2010.

Authors' Addresses

Vincent Roca Roca INRIA 655, av. de l'Europe Inovallee; Montbonnot ST ISMIER cedex, 38334 France EMail: vincent.roca@inria.fr URI: http://planete.inrialpes.fr/people/roca/

Aline Roumy Roumy INRIA Campus Universitaire de Beaulieu RENNES Cedex, 35042 France EMail: aline.roumy@inria.fr URI: http://www.irisa.fr/prive/Aline.Roumy/

Bessem Sayadi Sayadi Alcatel-Lucent, Bell Labs France EMail: bessem.sayadi@alcatel-lucent.com