



Extension and analysis of hybrid ARQ schemes in the context of cooperative relaying

Anna Vanyan

► To cite this version:

Anna Vanyan. Extension and analysis of hybrid ARQ schemes in the context of cooperative relaying. Other [cond-mat.other]. Université Paris Sud - Paris XI, 2014. English. NNT : 2014PA112105 . tel-01058051

HAL Id: tel-01058051

<https://theses.hal.science/tel-01058051>

Submitted on 26 Aug 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

UNIVERSITE PARIS-SUD

ÉCOLE DOCTORALE
Sciences et Technologie de l'Information, des Télécommunications
et des Systèmes
Laboratoire des Signaux et Systèmes

DISCIPLINE : PHYSIQUE

THÈSE DE DOCTORAT

Soutenue le 10/06/2014

par

Anna VANYAN

EXTENSION AND ANALYSIS OF HYBRID ARQ SCHEMES IN THE CONTEXT OF COOPERATIVE RELAYING
--

Composition du jury :

Directeur de thèse :
Co-encadrante de thèse :
Rapporteurs :

Président du jury :
Membres invités :

Pierre DUHAMEL
Francesca BASSI
Jean-Francois HÉLARD
Charly POUILLAT
Philippe CIBLAT
Aude HERRY

Directeur de la recherche CNRS, Fellow IEEE
Professeur, ESME Sudria
Directeur de la recherche, INSA Rennes
Professeur, INP ENSEEIHT Toulouse
Professeur, TELECOM Paris Tech
Tagattitude

Acknowledgement

I would like to thank the reviewers of my thesis for their constructive remarks and comments, which were extremely helpful in editing this work. Many thanks to the jury members for accepting to be the part of the thesis committee.

I would like to express my gratitude to the directors and supervisors of my thesis for their time, availability and guidance throughout this work.

I would like to deeply thank all the academic and administrative staff at Supelec and ESME universities, for welcoming me nicely, and always being available for discussions and helping me with everything.

Many thanks to all the PhD student peers both at Supelec and ESME, for always cheering up the atmosphere and making the life in lab bright and funny.

Of course, my greatest thanks go to my family and friends for everything.

Contents

I	SYNTHÈSE EN FRANÇAIS	5
II	HARQ analysis in the context of cooperative relaying	21
1	Introduction	23
1.1	Introduction	23
1.2	Performance Metrics Definitions	35
1.3	Thesis Motivation	37
2	Energetic fair criterion derivation	39
2.1	Considered setting	40
2.1.1	HARQ transmission system	40
2.1.2	STBC-ARQ transmission system	40
2.2	Theoretical performance analysis	41
2.3	Numerical results	42
2.3.1	Efficiency η	43
2.3.2	Packet Error Rate	45
2.3.3	Network layer delay	48
3	Deterministic Protocol Design and Analysis via FSM	51
3.1	Background	51
3.1.1	Finite state machines	53
3.2	One Source, One Destination, no combining	56
3.2.1	Protocol description	56
3.2.2	System model	57
3.2.3	Transmit FSM	57
3.2.4	Performance evaluation with FSMC	57
3.2.5	Simulation Validation	60
3.3	One Source, One Relay, One Destination Scheme, no combining	60
3.3.1	Protocol Description	61
3.3.2	System Model	63
3.3.3	Performance Analysis using Combinatorial Approach	64
3.4	Analysis of the One Source, One Relay, One Destination, no combining Scheme using FSMs	67
3.4.1	Transmit FSM, DCF relay mode	67

3.4.2	Performance evaluation with FSMC	68
3.4.3	Performance evaluation with FSMC, DMF protocol	69
3.4.4	Performance evaluation with FSMC, DCF protocol	72
3.5	Numerical Results	78
3.5.1	Combining versus no combining	80
3.5.2	Efficiency	82
3.5.3	FER	84
3.5.4	FER in an energetic fair context	86
3.5.5	Delay	88
3.5.6	Delay in an energetic fair context	88
3.5.7	DCF versus DMF	90
3.6	Conclusions	96
4	Multuser cooperative schemes	101
4.1	Introduction	101
4.1.1	Protocol design choices	101
4.1.2	Representation of the Protocol using a FSMC	102
4.2	Two Source, One Destination Scheme	103
4.2.1	Protocol definition	103
4.2.2	System model and decoding error probabilities	104
4.2.3	Functional description of the protocol	104
4.2.4	Finite state machine representation	105
4.3	Two Source, One Relay, One Destination Scheme, DCF	108
4.3.1	Definition of the communication protocol	108
4.3.2	System model and decoder design	111
4.3.3	Functional description of the protocol	111
4.3.4	FSMC description	111
4.3.5	Stochastic Transition Matrix	119
4.3.6	Transmitting FSM for DCF Protocol	119
4.4	Two sources, one relay, one destination scheme, DMF	122
4.5	Communication protocol definition	122
4.6	System model and decoder design	123
4.7	Functional representation of the protocol	125
4.8	FSMC description	125
4.8.1	On the number of states in the FSM	130
4.9	Numerical Results	130
4.9.1	Efficiency	131
4.9.2	FER	131
4.9.3	FER in an energetic fair context	133
4.9.4	Delay	134
4.9.5	Delay in an energetic fair context	136
4.9.6	2-S-1-R-1-D versus 1-S-1-R-1-D	136
4.10	Conclusions	138

5	Reducing the number of states in the FSM	143
5.1	One source, one destination, no combining	144
5.1.1	Equivalent Probabilistic Protocol	145
5.1.2	Performance evaluations	147
5.1.3	Simulation validation	148
5.2	One source, one relay, one destination, no combining	150
5.2.1	Equivalent Probabilistic Protocol	150
5.2.2	Performance Evaluations	152
5.3	Performance optimization	153
5.3.1	Physical layer setup	153
5.3.2	Optimization of the deterministic protocol	155
5.3.3	Optimization of the probabilistic protocol	158
5.3.4	Conclusions	161
6	Conclusions and Future Work	165
	Appendices	168
A	Evaluation of the elementary fragment error rate	171
A.1	No Combining	171
A.2	Demodulate-and-Forward with Combining	172
A.3	Decode-and-Forward with Combining	174
B	Evaluation of the elementary fragment error rate, multisource case	175
B.1	DCF protocol	175
B.2	DCF protocol	177

List of Figures

1	Butterfly network graph representation	7
2	Réseau coopératif multi-utilisateur multi-relais	8
3	Machine à états finis, protocole équivalent 1s-1d	17
1.1	An example of a directed graph	24
1.2	An example of a cut in a directed graph	25
1.3	Min-cut in a directed graph	25
1.4	Single-level diversity coding system graph representation	26
1.5	Butterfly network graph representation	27
1.6	Example of $\mathcal{C}_{4,2}$ Combination Network Topology	28
1.7	Example of $\mathcal{C}_{3,2}$ Combination Network Topology	29
1.8	Representation of $\mathcal{C}_{3,2}$ Combination as Butterfly Network	29
1.9	Multi-user multi-relay cooperative network	29
1.10	Cooperative relay network 1-s-m-d	34
2.1	Efficiency of the FEC, HARQ, STBC, STBC-ARQ systems	43
2.2	FEC, HARQ, STBC, STBC-ARQ PER as function of $\frac{E_s}{N_0}$	45
2.3	FEC, HARQ, STBC, STBC-ARQ PER as function of $\frac{E_s}{N_0}$	46
2.4	FEC, HARQ, STBC, STBC-ARQ network layer delay as function of $\frac{E_s}{N_0}$	48
2.5	FEC, HARQ, STBC, STBC-ARQ network layer delay as function of $\frac{E_s}{N_0}$	49
3.1	Finite State Machine Example	54
3.2	FSM describing the protocol behavior, 1s-1d	57
3.3	FER at the MAC layer, 1s-1d	61
3.4	Cooperative relay network 1s-1r-1d	61
3.5	FSM describing the protocol behavior, 1-s-1-r-1-d	68
3.6	Finite state machine DCF protocol of the 1-s-1-r-1-d scheme	75
3.7	Comparison of theoretical and simulated results of DCF protocol FER, 1s-1r-1d	79
3.8	FER of one and two FRAG transmissions in Rayleigh channel	80
3.9	DCF FER with and without combining at D; E_s/N_0 , 1-s-1-r-1-d	81
3.10	DCF FER with and without combining at D; true E_s/N_0 , 1-s-1-r-1-d	81
3.11	DCF FER with and without combining at D; E_s/N_0 , 1-s-1-r-1-d	82
3.12	DCF FER with and without combining at D; true E_s/N_0 , 1-s-1-r-1-d	83
3.13	DCF η_{ARQ} at the MAC layer as a function of E_s/N_0 , 1-s-1-r-1-d	83
3.14	DCF η_{gen} at the MAC layer as a function of E_b/N_0 , 1-s-1-r-1-d	84

3.15	DCF FER at the MAC layer as a function of E_b/N_0 , 1-s-1-r-1-d	85
3.16	DCF FER at the MAC layer as a function of E_s/N_0 , 1-s-1-r-1-d	86
3.17	DCF FER at the MAC layer vs true E_b/N_0 , 1-s-1-r-1-d	87
3.18	DCF FER at the MAC layer vs true E_s/N_0 , 1-s-1-r-1-d	88
3.19	DCF delay at the MAC layer as a function of E_b/N_0 , 1-s-1-r-1-d	89
3.20	DCF delay at the MAC layer as a function of E_s/N_0 , 1-s-1-r-1-d	89
3.21	DCF delay at the MAC layer vs true E_b/N_0 , 1-s-1-r-1-d	90
3.22	DCF delay at the MAC layer vs true E_s/N_0 , 1-s-1-r-1-d	91
3.23	DCF vs DMF η_{ARQ} at the MAC layer as a function of E_b/N_0 , 1-s-1-r-1-d . .	92
3.24	DCF vs DMF η_{gen} at the MAC layer as a function of E_b/N_0 , 1-s-1-r-1-d . .	92
3.25	DCF vs DMF FER at the MAC layer as a function of E_b/N_0 , 1-s-1-r-1-d . .	93
3.26	DCF vs DMF FER at the MAC layer as a function of E_s/N_0 , 1-s-1-r-1-d . .	94
3.27	DCF vs DMF FER at the MAC layer vs true E_b/N_0 , 1-s-1-r-1-d	94
3.28	DCF vs DMF FER at the MAC layer vs true E_s/N_0 , 1-s-1-r-1-d	95
3.29	DCF vs DMF delay at the MAC layer as a function of E_b/N_0 , 1-s-1-r-1-d . .	96
3.30	DCF vs DMF delay at the MAC layer as a function of E_s/N_0 , 1-s-1-r-1-d . .	97
3.31	DCF vs DMF delay at the MAC layer vs true E_b/N_0 , 1-s-1-r-1-d	97
3.32	DCF vs DMF delay at the MAC layer vs true E_s/N_0 , 1-s-1-r-1-d	98
4.1	Example: block-diagram for two source one destination scheme	106
4.2	Example: block-diagram for two source one destination scheme compact . .	106
4.3	Cooperative relay network 2s-1r-1d	108
4.4	Block-diagram DCF protocol of the 2-s-1-r-1-d scheme	112
4.5	Block-diagram DCF protocol of the 2-s-1-r-1-d scheme: compact form . . .	113
4.6	Block-diagram DCF protocol of the 2-s-1-r-1-d scheme: states classification	115
4.7	Transmitting FSM for the deterministic DCF Protocol	121
4.8	Algorithm of the 2 Source 1 Relay 1 Destination Scheme DMF Protocol . .	126
4.9	DCF vs DMF η_{ARQ} at the MAC layer as a function of E_b/N_0 , 2-s-1-r-1-d . .	131
4.10	DCF vs DMF η_{gen} at the MAC layer as a function of E_b/N_0 , 2-s-1-r-1-d . .	132
4.11	DCF vs DMF FER at the MAC layer as a function of E_b/N_0 , 2-s-1-r-1-d . .	132
4.12	DCF vs DMF FER at the MAC layer as a function of E_s/N_0 , 2-s-1-r-1-d . .	133
4.13	DCF vs DMF FER at the MAC layer vs true E_b/N_0 , 2-s-1-r-1-d	134
4.14	DCF vs DMF FER at the MAC layer vs true E_s/N_0 , 2-s-1-r-1-d	135
4.15	DCF vs DMF delay at the MAC layer as a function of E_b/N_0 , 2-s-1-r-1-d . .	135
4.16	DCF vs DMF delay at the MAC layer as a function of E_s/N_0 , 2-s-1-r-1-d . .	136
4.17	DCF vs DMF delay at the MAC layer vs true E_b/N_0 , 2-s-1-r-1-d	137
4.18	DCF vs DMF delay at the MAC layer vs true E_s/N_0 , 2-s-1-r-1-d	137
4.19	DCF 2-S-1-R-1-D VS 1-S-1-R-1-D FER at the MAC layer	138
4.20	DCF 2-S-1-R-1-D VS 1-S-1-R-1-D delay at the MAC layer	139
4.21	DCF 2-S-1-R-1-D VS 1-S-1-R-1-D FER at the MAC layer, true E_s/N_0 . .	139
5.1	FSM, 1s-1d ARQ protocol	145
5.2	FSM, 1s-1r-1d, equivalent	151
5.3	Simulated values of π_{SD} for various code rates.	155
5.4	Simulated values of π_{SD} and π_{RD} for various code rates. The normalized distance between relay and	

- 5.5 Optimized goodput for the 1s-1r-1d deterministic protocol, DMF mode, no combining, as a function of N_S
- 5.6 Optimum values of N_S (left) and N_R (right) for the 1s-1r-1d protocol, DMF mode, no combining, as a function of N_T
- 5.7 FER for the optimized 1s-1r-1d deterministic protocol, DMF mode, no combining, as a function of N_T
- 5.8 \bar{S} (left) and \bar{T} (right) for the optimized 1s-1r-1d protocol, DMF mode, no combining, as a function of N_T
- 5.9 Optimized goodput for the 1s-1r-1d probabilistic protocol, DMF mode, no combining, as a function of N_S
- 5.10 FER for the optimized 1s-1r-1d probabilistic protocol, DMF mode, no combining, as a function of N_T
- 5.11 \bar{S} (left) and \bar{T} (right) for the optimized probabilistic 1s-1r-1d protocol, DMF mode, no combining, as a function of N_T

List of Tables

1	La numérotation d'états, 1s-1d	14
1.1	Codebook for Scenario 1	32
1.2	Codebook for Scenario 2	32
1.3	Codebook for Scenario 3	32
1.4	Codebook for Scenario 4	32
2.1	Linear region extreme points for the all considered setups.	44
2.2	<i>Maximum efficiency</i> points for the all considered setups.	45
3.1	Numbering of original FSM states for one source, one destination scheme	59
3.2	DCF and DMF protocols in a nutshell	62
3.3	DMF FSM Original States	70
3.4	Stochastic transition probability matrix, 1-s-1-r-1-d, DCF protocol	76
3.5	Simulation configurations, 1-s-1-r-1-d scheme	79
3.6	Improvement achieved by increased transmissions, DCF, 1-s-1-r-1-d	87
5.1	Theoretical prediction of the FER and simulation results, $N = 5$	149
5.2	Theoretical \bar{T} prediction and simulation results, $N = 5$	149
5.3	DMF Equivalent FSM States	151
5.4	Considered puncturing matrices and corresponding code rates.	154

Résumé

Dans le canal sans fil, la communication coopérative permet à un ou plusieurs relais d'aider la transmission entre la source et la destination. L'objectif de cette thèse est de développer les outils pour analyser les systèmes coopératifs avec le déploiement de techniques HARQ, afin de pourvoir la protection du data cross-layer.

Le premier chapitre de cette thèse donne des informations sur le codage de réseau dans les réseaux de relais de coopération, et introduit la motivation pour ce travail.

L'objectif du deuxième chapitre de cette thèse est d'analyser et d'évaluer les performances de la qualité de service (QoS) des schémas ARQ-STBC, HARQ et FEC, dans un contexte de l'efficacité énergétique, aux niveaux des couches MAC et IP. Afin d'atteindre ce but, un nouveau cadre est dérivé et appliqué à un scénario de réseau point à point (P2P). Ceci nous permet d'établir les comparaisons entre les schémas avec et sans retransmissions, dans une manière appropriée à la consommation d'énergie de chaque système considéré. Cela nous permet de déterminer dans quelles conditions il est énergétiquement plus efficace d'utiliser la protection cross-layer, que simplement le codage de canal.

Dans le troisième chapitre de cette thèse, les protocoles coopératifs déterministes sont étudiés. Les protocoles que nous considérons sont différents en termes du comportement de relais(s) de source(s), et de la destination. Nous considérons deux types de protocoles coopératifs: decode-and-forward (DCF), et demodulate-and-forward (DMF). Les performances de chacun de ces protocoles, sont analysés avec et sans combinaison sur la destination. Les mécanismes de décodage de destination sont dérivés pour chaque protocole considéré, afin qu'on puisse comparer les performances de ces protocoles au niveau de la couche MAC. Les évaluations de QoS qui sont examinées sont: le taux d'erreur de trame, le délai, l'efficacité, et le goodput.

Les dérivations analytiques sont effectuées à l'aide de la machine à états finis de Markov, ainsi que grâce à l'approche combinatoire. Cependant, il est démontré, que la complexité de ces dérivations augmente au moment où le crédit de retransmissions et/ou des noeuds dans le réseau est augmentée. Donc cette approche devient non-traitable pour des grands schémas coopératifs.

Le quatrième chapitre présente une classe de protocoles de communication probabilistes, où les nuds retransmettent avec une certaine probabilité. Il est démontré et prouvée, qu'un protocole équivalent, qui est montré d'obtenir les mmes performances que le protocole déterministe. En utilisant la preuve de concept, nous démontrons que le protocole probabiliste permet d'effectuer les évaluations analytiques de réseaux multi-nuds, pour un certain nombre de transmissions. Basée sur cela, nous déduisons les paramètres QoS, et les évaluons également par des simulations Monte-Carlo.

Ensuite, les paramètres d'évaluation de performances dérivées sont optimisés en limitant le taux d'erreur de trame, et en essayant de trouver le nombre de transmissions le plus optimale et le code rate qui maximisent le goodput. Il est également démontré que le protocole équivalent obtient une région d'optimalité plus grande que celui du protocole déterministe.

Cette thèse donne le cadre pour l'analyse des réseaux sans fil coopératifs pour n'importe

quel nombre de nuds (relais, sources) avec des mécanismes HARQ. Les principaux objectifs de ce travail sont obtenus par l'analyse et l'optimisation des performances dans un contexte de l'efficacité énergétique. Il est démontré, que les métriques de QoS peuvent être analysés en utilisant les chaînes de Markov à états finis. Cela sert en tant qu'une base pour poursuivre les recherches sur d'autres protocoles de coopération (DCF, AF), avec plusieurs nœuds sur le côté de l'émetteur, du relais et de la destination. En outre, les zones de goodput réalisables sont calculées pour différents code rate, et sont optimisées. Il est ainsi possible de déterminer la meilleure combinaison de transmissions à crête et les taux de codage à la source et au relais. Ces résultats peuvent être utilisés afin de comparer les performances obtenues à ceux du protocole DCF, et puis peuvent être étendus aux réseaux coopératif multi-relais et multi-sources. Cette thèse permet aussi de poursuivre les études des gains de diversité, réalisables avec l'utilisation de plusieurs antennes sur le côté de l'émetteur et du relais, des dérivations de régime qui améliore le QoS, etc.

Part I

SYNTHÈSE EN FRANÇAIS

Introduction

Récemment l'intérêt dans la recherche a considérablement augmenté vers les techniques du codage des réseaux, qui permettent d'utiliser efficacement la bande passante, ce qui a été démontré par Ahlswede [1]. Dans ce travail les auteurs décrivent une technique simple, qui rassemble le flux de données provenant de différents noeuds en un seul paquet, ce qui conduit à la réduction de la bande passante utilisée par le système. La technique utilisée dans l'article mentionné ci-dessus consiste en une simple opération binaire qui s'appelle *exclusive-OR* (XOR) [2]. Cette exemple a été démontré sur un réseau *butterfly*, Figure 1.5. Comme il est représenté par la figure, le système *butterfly* se compose d'une source qui émet deux bits différents (b_1 et b_2) à deux récepteurs: t_1 et t_2 . Nous pouvons voir qu'en effectuant XOR entre les deux flux de données au noeud de relais 3, on peut réduire le nombre d'intervalles de temps nécessaires pour envoyer les deux messages à leur destinataires.

Tandis que la plupart des articles dans la littérature est axée sur l'atteinte d'un meilleur *débit de multidiffusion* et/ou d'une meilleure capacité [3–6], une autre direction a été prise par d'autres auteurs, qui consiste à augmenter le gain de diversité [7–12]. Une approche pratique a été exploitée dans [5, 6] en proposant de nouvelles architectures pour les réseaux sans fil (par exemple, COPE, MIXIT, ANC [6]), en présentant des algorithmes polynomiaux pour le codage et le décodage, [13]. Dans [13] Koetter et Médard ont montré que les codes avec une simple structure linéaire sont suffisantes pour atteindre la capacité dans le problème de multidiffusion. D'autres types de codage de réseau proposent de mélanger les données sur le niveau de symbole [14], ou d'utiliser la modulation analogique sans synchronisation entre les signaux parasites et en l'appliquant à des software radios [15], ou en utilisant les codes de réseau aléatoires [16, 17].

Les auteurs de [7–9] ont souligné, qu'afin d'obtenir des gains de diversité dans les réseaux coopératifs, l'opération binaire XOR n'est pas optimale, et que les codes de blocs

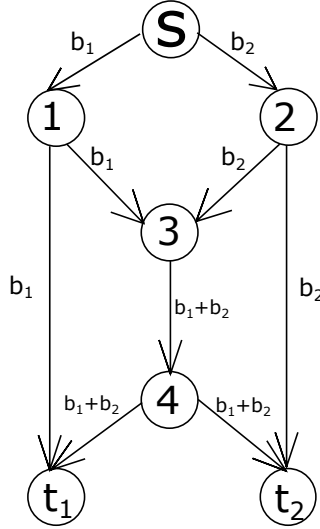


Figure 1: Butterfly network graph representation

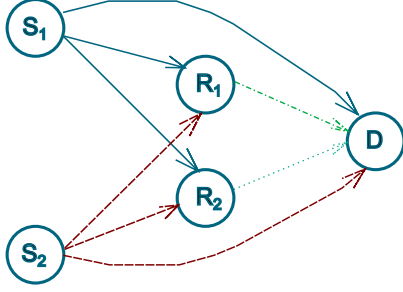


Figure 2: Réseau coopératif multi-utilisateur multi-relais. Les différents styles de ligne (et couleurs) indiquent les intervalles de temps. Dans les premiers 2 intervalles de temps S_1 transmet le data à R_1 , R_2 et D . Dans la deuxième intervalle de temps S_2 transmet le data à R_2 , R_1 et D . Dans le troisième intervalle de temps R_1 transmet D , et au quatrième R_2 D .

linéaires non-binaires peuvent aider à obtenir les gains de diversité souhaités.

Dans [9] les auteurs considèrent que l'on appelle le **codage de réseau généralisé et dynamiques**, (Generalized Dynamic Network Codes, GDNC). Leur travail a été motivé par [8], où les gains de diversité ont été atteints pour un programme de coopération multi-utilisateur à l'aide de **des codes de réseau dynamiques**, (Dynamic Network Codes, DNC). Dans [8] le système se compose de M utilisateurs qui agissent comme relais. Chacun des utilisateurs dispose de données indépendantes à transmettre à la destination, et dans le premier intervalle de temps les données de tous les noeuds sont envoyées en mode broadcast à l'autre et à la station de base. Du deuxième jusqu'au m -ième intervalle de temps les utilisateurs transmettent $M - 1$ combinaisons linéaires non binaires des données qui sont reçues à la station de base. Le travail [8] montre que l'utilisation des codes de réseau linéaires non binaires atteint mieux la diversité que celle des codes de réseau linéaire binaires. Dans [9] les auteurs étendent l'idée de DNC à celle de GDNC, en permettant les utilisateurs d'envoyer plusieurs paquets originaux au lieu d'un; et plusieurs paquets codés au lieu d'un. En faisant cela, ils ont de plus d'observations, et sont capables d'atteindre un meilleur taux et une meilleure diversité, que ceux de DNC.

Cependant, les auteurs de [10–12] ont prouvé qu'il est en effet possible d'atteindre des gains de diversité en utilisant l'opération XOR simple défini au corps de Galois, F_2 . Le choix du corps F_2 permet de maintenir la complexité de calcul du système très basse et peut être appliquée à la protection d'erreur inégale dans les réseaux de relais coopératif multi-relais multi-utilisateurs, [10, 11, 18].

Dans [10–12] les auteurs ont étudié les effets de codage de réseau binaire dans les réseaux de relais coopératifs, et ont obtenu la probabilité moyenne d'erreur de bit (ABEP) au niveau de la couche physique (PHY), en utilisant des approximations afin d'obtenir des estimations asymptotiquement serrées.

Le modèle de réseau qui est considéré dans [10–12], est consisté en deux sources, deux relais, et une destination comme illustré sur la figure (1.9).

Les auteurs font l'assumption que le relais ne vérifie pas si les bits qu'il reçoit des sources sont correctes ou non. Dans [10] les auteurs obtiennent les ABEPs pour les

canaux idéaux entre les sources et les relais, alors que dans [11] ils dérivent les ABEPs pour des canaux réalistes entre les sources et les relais. En outre, dans tous ces travaux, les auteurs considèrent quatre scénarios, qui représentent les cas suivants :

1. les relais envoient les observations directs des bits d'information qu'ils ont reçus de la part des sources S_1 et S_2
2. les deux relais effectuent une opération XOR sur les bits d'information qu'ils ont reçus de la part des sources S_1 et S_2 , avant de les envoyer à la destination
3. l'un des relais envoie la version des bits d'information codé avec un XOR, et l'autre relai envoie la version non- codée. D'une telle manière les auteurs munissent une protection d'erreur inégale pour l'une des sources.

Ces approches diffèrent en fonction des données que le relais envoie. Une brève description de chacun de ces scénarios est résumé ci-dessous.

Scénario 1: R_1 et R_2 envoient les données des sources d'information S_1 et S_2 la destination dans des intervalles de temps orthogonaux. En d'autres termes, R_1 envoie l'observation de b_1 ; et R_2 envoie l'observation de b_2 .

Scénario 2: R_1 et R_2 envoient les bits d'information qui viennent des sources S_1 et S_2 codés avec le codage de réseau à la destination dans des intervalles de temps orthogonaux. L'opération XOR binaire est utilisé pour introduire un codage de réseau. En d'autres termes, le relais envoie $b_1 \oplus b_2$.

Scénario 3: R_1 envoie la version codée des observations de b_1 et b_2 ; et R_2 envoie l'observation directe de b_1 . Dans ce cas l'information de la source S_2 est plus protégée, que celle de la source S_1 .

Scénario 4: R_1 envoie l'observation directe de b_1 ; et R_2 envoie la version codée des observations de b_1 et b_2 . Dans ce cas l'information de la source S_1 est plus protégée, que celle de la source S_2 .

Tous ces scénarios sont évalués en fonction de la probabilité moyenne d'erreur de bit d'information pour chaque source. De plus, ils considèrent les trois types de décodeurs à la destination : a) MDD, b) H-MLD, et c) S-MLD [10–12].

Motivation de la thèse et contributions

Inspiré par ces travaux, cette thèse a pour l'objectif d'obtenir une technique simple pour analyser les réseaux de relais coopératifs avec plusieurs noeuds, et avec des retransmissions. L'innovation de ce travail consiste à développer une analyse qui sera utile pour des protocoles de relais coopératifs avec les schémas HARQ et multiples noeuds dans le réseau. Nous avons réuni les techniques du codage convolutif (FEC), les techniques de retransmissions (ARQ), du codage de réseau (le cas échéant) dans un contexte des réseaux coopératifs. Nous avons développé un outil basé sur la machine à états finis afin d'analyser les schémas avec plusieurs noeuds. Nous démontrons dans cette thèse que l'approche conventionnelle n'est plus efficace, quand le nombre des noeuds dans le réseau augmente :

même pour les schémas coopératifs les plus simples, qui consistent en trois noeuds, les expressions sont assez compliquées, alors que la machine à états finis, si les états sont bien définis, nous permet d'obtenir les performances dans une manière simple et algorithmique. Puis nous démontrons que pour le schéma avec deux sources, l'analyse devient trop compliqué même si on utilise la machine à l'états finis. Par conséquent, nous prouvons qu'il est possible de réduire le nombre de la machine à l'états finis en définissant un protocole équivalent, basé sur le principe probabiliste.

En outre nous effectuons l'optimisation du protocole probabiliste pour le schéma non-coopératif et coopératif, et nous démontrons que le protocole probabiliste obtient des meilleures performances que le protocole déterministe.

De plus, notre recherche est faite dans un contexte énergétiquement juste, dans le sens que toutes les comparaisons et évaluations des performances tiennent en compte la consommation d'énergie par un bit d'information reçu avec succès.

Avant de passer à la section suivante, nous allons introduire les définitions des métriques qui seront dérivées et considérées pour chaque système considéré dans cette thèse. Les définitions sont génériques et peuvent être appliquées aux schémas multi-source multi-relais, et aux schémas 1-source-1-destination.

Le taux d'erreur de trame ou Frame Error Rate (FER)

$$\text{PER} = \frac{\text{nombre de trames erronées appartenant à la source } i}{\text{nombre total de trames transmises pour la source } i \text{ de la destination et du relais (le cas échéant)}} \quad (1)$$

Le taux d'erreur de paquet ou Packet Error Rate (PER)

$$\text{PER} = \frac{\text{nombre de paquets erronés appartenant à la source } i}{\text{nombre total de paquets transmises pour la source } i \text{ de la destination et du relais (le cas échéant)}} \quad (2)$$

Le délai par un paquet/fragment

$$\bar{T} = \frac{\text{nombre de trames (ou paquets) transmises appartenant à la source } i}{\text{nombre de trames (ou paquets) générés à la source } i} \quad (3)$$

Le délai par un paquet/fragment reçu avec succès

$$\bar{S} = \frac{\text{nombre de trames (ou paquets) transmises appartenant à la source } i}{\text{nombre de trames (ou paquets) correctement reçues et appartenant à la source } i} \quad (4)$$

L'efficacité qui tient en compte la redondance introduite par les retransmissions et par le taux du codage au niveau de la couche PHY, η_{gen}

$$\eta_{\text{gen}} = \frac{\text{nombre de bits d'information appartenant à la source } i \text{ correctement décodés}}{\text{nombre totale de bits codés transmises de la destination et du relais (le cas échéant)}} \quad (5)$$

L'efficacité qui tient en compte seulement la redondance introduite par les retransmissions, η_{ARQ}

$$\eta_{\text{ARQ}} = \frac{\text{nombre de bits d'information appartenant à la source } i \text{ correctement décodés}}{\text{nombre totale de bits d'information transmises de la destination et du relais (le cas échéant)}} \quad (6)$$

Les dérivations d'une métrique pour une analyse énergétiquement juste

Dans la littérature les comparaisons entre les schémas sans et avec retransmissions sont effectuées en utilisant le ratio entre l'énergie par symbole transmis et la variance du bruit (SNR) en tant que l'unité de mesure. Cependant, dans les schémas (H)ARQ, la consommation d'énergie pour la réception correcte d'un symbole est amplifiée grâce aux retransmissions redondantes du même symbole. Cela signifie que la méthode conventionnelle des évaluations en fonction de SNR ne donne pas des comparaisons justes entre ces deux types de schémas. Pour cette raison dans le deuxième chapitre de cette thèse nous dérivons une métrique qui tient en compte ce fait. Nous appelons cette métrique le *vrai SNR*. Afin d'analyser les résultats obtenus en fonction du vrai SNR, nous effectuons des évaluations de performances des schémas suivants: HARQ, STBC-ARQ et ARQ. Les schémas STBC consistent en la transmission des données à partir de multiples antennes en utilisant le codage *Space-Time Block Codes*. Dans cette thèse nous considérons deux antennes au niveau de l'émetteur et une antenne au niveau du récepteur. La destination reçoit la somme des signaux des toutes les antennes d'émission. Afin d'être en mesure de décoder les symboles originaux, il faut que le canal reste constant pendant les deux intervalles de temps consécutifs. La matrice de codage STBC utilisée dans cette thèse correspond à celle d'Alamouti [19] et est représentée ci-dessous

$$G = \begin{pmatrix} x_1 & x_2 \\ -x_2^* & x_1^* \end{pmatrix}, \quad (7)$$

où $(\cdot)^*$ est l'opérateur de la conjugaison complexe, et les termes x_i sont les symboles envoyés des deux antennes.

L'analyse théorique des performances QoS des systèmes de transmission HARQ et STBC-ARQ est basée sur les résultats dans [2, 20, 21]. Nous sommes intéressés par la dérivation des performances QoS au niveau de la couche IP : le taux d'erreur de paquet (PER), le délai, et l'efficacité. Afin d'évaluer ces métriques, nous aurons besoin de calculer le taux d'erreur de bit d'information (BER) au niveau de la couche physique (PHY), car elles sont exprimées en fonction de ce dernier. Pour le système STBC-ARQ le BER peut être calculé en utilisant l'expression bien connue dans la littérature [19]. Néanmoins, pour les schémas dans lesquels les expressions théoriques n'existent pas, nous utiliserons les résultats de simulation. L'expression de BER pour STBC-ARQ au niveau de la couche PHY est donc

$$\text{BER}_{\text{STBC}} = p^2 (1 + 2(1 - p)), \quad (8)$$

où $p = \frac{1}{2} - \frac{1}{2} \left(1 + \frac{2}{\frac{E_s}{N_0}} \right)^{-1/2}$, et $\frac{E_s}{N_0}$ est l'énergie dépensée par le symbole transmis. Le BER_{HARQ} doit être évalué par simulation, pour chaque taux de codage, R_c . La proba-

bilité d'une erreur dans un FRAG est notée par le terme π , et défini comme la probabilité d'avoir incorrectement reçu un FRAG au niveau de la destination après une transmission.

$$\pi = 1 - (1 - \text{BER})^{L_{\text{FRAG}}} . \quad (9)$$

Le terme π^{N_S} donc définit le taux d'erreur de trame au niveau de la couche MAC, soit le fragment n'a pas pu être décodé après l'expiration du nombre maximale de transmissions.

Le taux d'erreur de paquet au niveau de la couche IP (Packet Error Rate, PER) est défini comme le nombre moyen des paquets non décodables sur le récepteur au niveau de la couche IP. Un paquet IP est considéré comme non décodé dès que l'un de fragments appartenant à ce paquet est rejetée au niveau de la couche MAC. Le PER est donc exprimée par

$$\text{PER} = 1 - (1 - \pi^{N_S})^{N_{\text{FRAG}}} . \quad (10)$$

Le nombre moyen de transmissions d'un fragment, \bar{T} , est évaluée par

$$\bar{T}_{\text{MAC}} = \sum_{k=1}^{N_S} k (1 - \pi) \pi^{k-1} + N_S \pi^{N_S} \quad (11)$$

Le premier terme dans Eq. ((2.6)) exprime le nombre moyen de transmissions d'un *fragment reçu avec succès*. Le délai au niveau de la couche IP, n est définie comme la moyenne nombre de transmissions de fragments associés à un paquet IP décodée avec succès. Son expression est définie par

$$\bar{S}_{\text{IP}} = N_{\text{FRAG}} \left(\sum_{k=1}^{N_S} k \frac{(1 - \pi) \pi^{k-1}}{1 - \pi^{N_S}} \right) . \quad (12)$$

L'efficacité η est défini comme le ratio entre le goodput au niveau de la couche MAC et le débit au niveau de la couche PHY ¹. Elle exprime le nombre moyen d'utilisations de canal, nécessaires pour une réception réussie d'un bit d'information à la destination, mesurant ainsi le coût de la stratégie de protection d'erreur inter-couche. Une petite valeur de η indique que le système introduit une grande quantité de redondance (le codage d'information *et* les retransmissions des fragments) afin de protéger l'information. Il est facile de vérifier que l'efficacité generale η_{gen} prend la forme suivante

$$\eta_{\text{gen}} = \rho R_c \frac{(1 - \pi^{N_S})}{\bar{T}} , \quad (13)$$

où ρ est un coefficient qui tient en compte la redondance introduite par les en-têtes et par le CRC dans une trame, et R_c est le taux de codage convolutive ($R_c = 1$ pour le schéma STBC-ARQ).

Les performances des deux systèmes sont évaluées par les comparaisons de la qualité de service: PER, \bar{S} , qui sont habituellement exprimés en fonction d'énergie moyenne dépensée

¹ *Le goodput au niveau de la couche MAC* signifie le nombre moyen de bits d'information (*ie* appartenant à des fragments au niveau de la couche MAC) correctement reçus par l'unité de temps. Le *débit* au niveau de la couche PHY signifie le nombre moyen de bits d'information transmis sur le canal par l'unité de temps.

par symbole, $\frac{E_s}{N_0}$. Cette quantité, cependant, n'est pas une métrique exhaustive pour analyser les systèmes considérés, étant donné qu'elle ne permet pas de calculer l'énergie moyenne dépensée par symbole pour les schémas avec retransmissions. Par conséquent, nous définissons une nouvelle métrique, $\bar{\frac{E_s}{N_0}}$, comme l'énergie moyenne dépensée par le système afin de transmettre un symbole correctement à la destination. Elle est exprimée par :

$$\bar{\frac{E_s}{N_0}} = \frac{E_s}{N_0} \frac{1}{\eta_{gen}}, \quad (14)$$

Conception et analyse de protocole avec une machine à états fini

Dans le troisième chapitre de cette thèse nous développons des outils d'analyse du système pour les réseaux de relais coopératifs. Nous utilisons la machine à états finis afin d'obtenir les performances QoS du système. Nous avons choisi d'effectuer l'analyse du système en utilisant cette méthode, parce qu'elle est plus efficace que l'approche combinatoire. Cela devient crucial dans le cas avec plusieurs noeuds. D'abord nous appliquons cet outil au réseau le plus simple qui se compose de trois noeuds: une source, un relais et une destination. Puis nous compliquons le schéma en rajoutant plusieurs noeuds et nous démontrons l'utilisation de la machine à états finis afin d'analyser les performances de protocole. Nous utilisons également le *vrai SNR* dérivé plus tôt, pour les évaluations dans le contexte énergétiquement juste.

Rappelons qu'une machine à états finis est un modèle de calcul mathématique, utilisé pour la description d'un système. Les systèmes à états finis peuvent être modélisés par les *machines de Mealy* ou par les *machines de Moore* qui sont des automates finis avec sortie. Dans les machines de Mealy, les actions (sorties) sont liées aux transitions, tandis que dans les machines de Moore, les actions sont liées aux états. En d'autres termes, une machine à états finis se compose d'états, d'entrées et de sorties.

Nous commençons par le développement d'une machine à états finis pour le schéma le plus simple, qui consiste en une source et une destination. Nous démontrons qu'il est possible d'obtenir toutes les métriques d'évaluation des performances du système en utilisant cette technique. Ces résultats sont validés par des simulations Monté-Carlo et par des expressions théoriques décrites plus tôt, dans le deuxième chapitre de cette thèse.

Afin d'obtenir les résultats analytiques nous représentons la machine à états finis avec une matrice de transition d'état, ce qui nous permet de calculer les probabilités stationnaires. Grâce à ce calcul, nous évaluons le taux d'erreur de trame, le délai, et l'efficacité.

Description de protocole avec une source et une destination

Pour le système avec une source et une destination, le protocole est assez simple: la source transmet son message et attend un accusé de réception de la part de la destination. Si le message a été reçu avec succès, la source transmet un nouveau message. Sinon, la source retransmet le message qui n'a pas été reçu, jusqu'à la réception réussie ou jusqu'à l'expiration du crédit des retransmissions.

Nous considérons un canal de Rayleigh avec le bruit Gaussien. Le signal est modulé selon la constellation BPSK. Chaque trame au niveau de la couche MAC consiste en L_{FRAG} bits d'information. La source possède un compteur A , qui est incrémenté chaque fois que la source (re)transmet le message actuel. Au début de la procédure de transmission d'un

Le nombre de l'état	La definition de l'état
S_1	$W_{t-1} = \text{ACK}, A_t = 1$
S_2	$W_{t-1} = \text{NACK}, A_t = 2$
S_3	$W_{t-1} = \text{NACK}, A_t = 3$
\dots	\dots
S_N	$W_{t-1} = \text{NACK}, A_t = N$
S_{N+1}	$W_{t-1} = \text{NACK}, A_t = 1$

Table 1: La numérotation d'états de la machine à états finis pour le schéma avec une source et une destination

nouveau message ce compteur a une valeur égal à 1. Le compteur peut donc prendre l'une des valeurs suivantes: $\{1, \dots, N\}$. Le succès ou l'échec d'une transmission est dénoté par la variable K , qui prend la valeur $K = \text{ACK}$ si la transmission a été réussie, et $K = \text{NACK}$ dans le cas opposé.

La machine à états finis qui décrit le schéma de transmission de ce protocole est représentée par la figure 3.2.

0.0.0.1 Définition des états

Chaque état dans la machine à états finis est associé à une probabilité de transition vers un autre état. La matrice stochastique (aussi appelée matrice de Markov) est une matrice où chaque élément P_{ij} représente la transition de l'état i vers l'état j . La somme des éléments de chaque ligne est égale à 1.

Notre définition d'un état est la suivante : le pair (le nombre de la transmission actuelle, le dernier accusé de réception) = (A_t, W_{t-1}) . Les indices t et $t-1$ signifient l'intervalle de temps actuel et l'intervalle de temps précédent, respectivement. On remarque, que grâce à l'organisation du protocole, l'état avec un numéro de transmissions $1 < A_t \leq N$ peut être visité seulement si la dernière transmission a échoué, c'est-à-dire $W_{t-1} = \text{NACK}$. Par conséquent, les états $(A_t = a, W_{t-1} = \text{ACK})$ pour $a \in \{2, \dots, N\}$ n'existent pas.

- L'état $(A_t = 1, W_{t-1} = \text{ACK})$ représente la première transmission d'un nouveau FRAG, qui commence après avoir reçu avec succès le FRAG précédent.
- L'état $(A_t = 1, W_{t-1} = \text{NACK})$ représente la première transmission d'un nouveau FRAG, qui commence après ne pas avoir reçu le FRAG précédent.

Le nombre totale d'états est égale à $N+1$. Leur numérotation est faite selon le tableau 3.1.

0.0.0.2 L'évaluation des probabilités de transition

Les transitions dépendent de l'état actuel et du résultat W_t (c'est-à-dire le résultat de la transmission actuelle A_t).

De l'état $(A_t = 1, W_{t-1} = *)$

Les états S_1 et S_{N+1} peuvent aller à S_1 en cas d'une transmission réussie et à S_2 en cas d'une transmission échoué.

$$\begin{aligned} P_{1,1} &= P_{2,1} = P(W_t = 1) = (1 - \pi), \\ P_{1,2} &= P_{2,3} = P(W_t = 0) = \pi \end{aligned} \quad (15)$$

De l'état $(A_t = a, W_{t-1} = \text{NACK}), 1 < a < N$

N'importe quel état S_i ($A_t = a, W_{t-1} = \text{NACK}$) avec $1 \leq a \leq N$ peut aller à l'état S_1 en cas d'une transmission réussie $W_t = \text{ACK}$, et à l'état $i + 1$ ($A_t = a + 1, W_{t-1} = \text{NACK}$) si la transmission précédente a échoué.

$$\begin{aligned} P_{a,1} &= P(W_t = 1) = (1 - \pi), \\ P_{a,a+1} &= P(W_t = 0) = \pi \end{aligned} \quad (16)$$

De l'état $(A_t = N, W_{t-1} = \text{NACK})$

L'état S_N peut aller ou à l'état S_1 (si la dernière transmission du FRAG a été réussie) ou à l'état S_{N+1} (si la dernière transmission du FRAG a échoué).

$$\begin{aligned} P_{N,1} &= P(W_{t-1} = \text{ACK}) = (1 - \pi), \\ P_{N,N+1} &= P(W_{t-1} = \text{NACK}) = \pi \end{aligned} \quad (17)$$

La matrice de probabilités de transition prend deon la forme suivante:

$$P = \begin{pmatrix} 1 - \pi & \pi & 0 & 0 & 0 & \dots & 0 \\ 1 - \pi & 0 & \pi & 0 & 0 & \dots & 0 \\ 1 - \pi & 0 & 0 & \pi & 0 & \dots & 0 \\ 1 - \pi & 0 & 0 & 0 & \pi & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 1 - \pi & \pi & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (18)$$

Afin d'évaluer les performances QoS du système nous allons utiliser le vecteur propre de la matrice de probabilités de transition avec la valeur propre 1.

Taux d'erreur de trame Afin de trouver le taux d'erreur de trame nous allons utiliser les états suivants de la machine à l'états finis: l'état où le système transmet un nouveau FRAG après avoir reçu avec succès le FRAG précédent, et l'état où le système transmet un nouveau FRAG après ne pas avoir reçu le dernier FRAG. Le taux d'erreur de trame est égal à la moyenne de la probabilité de transmettre un nouveau FRAG après ne pas avoir reçu le dernier FRAG. On dénote la probabilité stationnaire d'être dans l'état i par p_i . Le FER est donc représenté par

$$\text{FER} = \frac{p_{N+1}}{p_1 + p_{N+1}} \quad (19)$$

Average number of transmitted FRAGs per successful FRAG The average number of transmitted FRAGs per successful FRAG for this cooperative scenario is defined as the average number of transmitted FRAGs from the source and the relay, in order to successfully receive one FRAG:

$$\begin{aligned}
\bar{S} &= \sum_{i=1}^N i \cdot P(i \text{ transmissions} | \text{FRAG est réussie}) \\
&= \sum_{i=1}^N i \cdot \frac{P(\text{d'être en état 1} | i\text{-th transmission})}{P(\text{d'être en état 1})} \\
&= \sum_{i=2}^N i \cdot \frac{(1-\pi)p_i}{p_1} + (1-\pi) \frac{p_1 + p_{N+1}}{p_1}
\end{aligned} \tag{20}$$

Nombre moyen de trames transmises L'expression du nombre moyen de trames transmises est représentée par

$$\bar{T} = \bar{S} \frac{p_1}{p_1 + p_{N+1}} + N \frac{p_{N+1}}{p_1 + p_{N+1}} \tag{21}$$

Nombre moyen de trames transmises par une trame réussie Le nombre moyen de trames transmises par une trame réussie est égal au nombre moyen de trames transmises afin de recevoir correctement une trame

$$\begin{aligned}
\bar{S} &= \sum_{i=1}^N i \cdot P(i \text{ transmissions} | \text{FRAG est réussie}) \\
&= \sum_{i=1}^N i \cdot \frac{P(\text{d'être en état 1} | i\text{-ème transmission})}{P(\text{d'être en état 1})} \\
&= \sum_{i=2}^N i \cdot \frac{(1-\pi)p_i}{p_1} + (1-\pi) \frac{p_1 + p_{N+1}}{p_1}
\end{aligned} \tag{22}$$

Nous démontrons que les expressions obtenues sont en parfait accord avec les expressions combinatoires, et les simulations de Monté-Carlo.

Dans une manière assez similaire, nous avons étudié le schéma avec une source, un relais et une destination; le schéma avec deux sources, un relais et une destination. Pour chacun de ces schémas nous avons démontré que l'analyse combinatoire devient très compliquée et il faut la refaire chaque fois que le protocole change. En outre, nous avons remarqué que avec l'augmentation du nombre des noeuds dans le réseau, et avec l'augmentation du nombre des retransmissions, l'analyse devient de plus en plus compliqué même si on utilise cette technique basée sur les machines à l'états finis.

De plus, pour chacun de ces schémas de relais coopératifs, nous avons défini et étudié deux protocoles différents. Ces protocoles diffèrent en termes du comportement du relais et de la distance source-relais : un protocole est assez simple à réaliser, mais il y a de

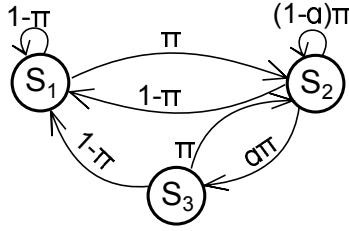


Figure 3: Machine à états finis pour le schéma avec une source et une destination $S_1 = (W_{t-1} = 0)$, $S_2 = (W_{t-1} = 0, J_{t-1} = 1)$, $S_3 = (W_{t-1} = 0, J_{t-1} = 0)$

compromis dans les performances, et l'autre protocole est plus difficile à réaliser, mais il donne des meilleures performances.

Réduction du nombre des états de la machine à états finis

Dans la quatrième chapitre de cette thèse nous avons réduit le nombre des états dans la machine à états finis, et nous démontrons que cette approche simplifie l'analyse pour les schémas plus grandes. Pour cette raison-là nous avons redéfini le protocole, en retrouvant une protocole équivalent.

Dans ce résumé nous allons démontrer sur le schéma avec une source et une destination, qu'il existe un protocole équivalent à celui de la chapitre trois. Ce protocole est basé sur un principe probabiliste, ce qui nous permet d'obtenir des meilleures performances que celles du protocole déterministe. Cela est expliqué par le fait que le protocole probabiliste est plus optimale.

Le protocole probabiliste pour le schéma le plus simple est défini comme suite : si la première transmission de la trame est échouée, une deuxième transmission sera attribué dans une manière déterministe. Si cette deuxième transmission échoue, la t -ème transmission est attribué avec une probabilité $(1 - \alpha)$. Afin de concevoir cette règle, nous définissons une variable aléatoire J_t^S qui prend les valeurs dans l'intervalle $\{0, 1\}$, et $P(J_t^S = 1) = (1 - \alpha)$. En d'autres termes, cette variable détermine si la source est autorisée de transmettre pour la t -ème fois.

La machine à l'états finis pour ce protocole est représentée par trois états décrits ci-dessous :

État S_1 : $S_1 = (W_{t-1} = 1)$ est associé à la première transmission d'un nouveau FRAG après avoir recu le FRAG précédent avec succès.

État S_2 : $S_2 = (W_{t-1} = 0, J_{t-1} = 0)$ est associé à la retransmission du FRAG actuel après qu'il n'a pas été recu dans la première et deuxième transmission

État S_3 : $S_3 = (W_{t-1} = 0, J_{t-1} = 1)$ est associé à la première transmission d'un nouveau FRAG après ne pas avoir recu le FRAG précédent.

La machine à états finis est représentée par la Fig. 5.1.

Les transitions entre les états dépendent du résultat de la transmission, W_t , et de la réalisation de J_t . Les probabilités de transition pour cette machine sont évaluées par :

$$P_{i \rightarrow 1} = P(W = 1) = (1 - \pi), \quad i \in \{1, 2, 3\} \quad (23)$$

où le terme π dénote la probabilité d'une erreur dans la transmission d'un FRAG sur le canal source-destination, et il est évalué en utilisant Eq. (2.4).

Les états S_1 et S_3 peuvent aller à l'état S_2 après un échec d'une transmission:

$$P_{i \rightarrow 2} = P(W = 0) = \pi, \quad i \in \{1, 3\} \quad (24)$$

L'état S_2 peut aller à S_3 , si $W_t = 0$ et $J_t = 0$; et il peut rester en lui-même si $W_t = 0$ et $J_t = 1$.

$$\begin{aligned} P_{2 \rightarrow 2} &= P(W_t = 0, J_t = 1) = \pi(1 - \alpha), \\ P_{2 \rightarrow 3} &= P(W_t = 0, J_t = 0) = \pi\alpha \end{aligned} \quad (25)$$

Ainsi, la matrice prend la forme suivante :

$$P = \begin{pmatrix} 1 - \pi & \pi & 0 \\ 1 - \pi & (1 - \alpha)\pi & \alpha\pi \\ 1 - \pi & \pi & 0 \end{pmatrix} \quad (26)$$

En dénotant la la probabilité stationnaire d'être en état i par p_i , nous obtenons le FER :

$$\text{FER} = \frac{p_3}{p_1 + p_3} \quad (27)$$

Nous avons également évalué le terme α , qui prend la forme suivante:

$$\begin{aligned} \alpha &= \frac{P(A_t = N | 1 \leq A_{t-1} \leq N-1)}{P(A_t = N)} \\ &= \frac{P(A_t = N, 1 \leq A_{t-1} \leq N-1)}{P(A_t = N | 1 \leq A_{t-1} \leq N-1)} \\ &= \frac{P(A_t = N)}{\sum_{i=2}^N P(A_t = i)} \\ &= \frac{\pi^{(N-1)}}{\sum_{i=2}^N \pi^{(i-1)}} \end{aligned} \quad (28)$$

Dans une manière similaire toutes les autres métriques ont été obtenues, et comparées à celles du protocole déterministe.

Nous avons également optimisée les performances du protocole probabiliste. Nous avons effectué cette procédure pour le schéma du relais coopératif, et les résultats que nous avons obtenu ont montré que le protocole probabiliste est meilleur que le protocole déterministe en fonction des performances QoS, et en fonction de la complexité de la machine à l'états finis. Les résultats que nous avons obtenu permettent de les appliquer pour des réseaux du relais coopératifs, et d'avoir un outil d'analyse assez simple.

Conclusions et Perspectives

Cette thèse a pour but d'obtenir un outil pour analyser les réseaux coopératifs sans fil et d'étudier les performances QoS en utilisant les techniques HARQ et codage de réseau. Nous avons commencé par les études des réseaux simple dans le but de démontrer qu'il est possible d'obtenir les métriques de QoS en utilisant la machine à l'états finis. Puis nous avons défini deux protocoles différents pour les réseaux coopératifs avec trois noeuds. Nous démontrons que le relais peut être utile seulement dans le cas si le protocole est bien défini, et le décodeur du côté de la destination est adapté aux conditions du canal.

Néanmoins, on peut voir que même pour le schéma coopératif le plus simple, la machine à l'états finis devient compliquée. Puis on montre que la complexité augmente pour le schéma avec deux sources, en effectuant l'étude des deux protocoles déterministes pour ce schéma. La machine à l'états finis immense, et rend la complexité de calcul très élevée. On arrive à classifier les états en utilisant un algorithme, mais il est pratiquement difficile de calculer les métriques QoS.

Afin de surmonter cette difficulté, nous développons une nouvelle stratégie probabiliste, qui nous permet d'avoir une machine à l'états finis très petite. D'abord nous faisons une démonstration sur les schémas non-coopératifs. Appart d'être simple à implémenter, cette approche est plus efficace. Sur cet exemple nous avons montré que le protocole probabiliste obtient les meilleures performances que le protocole déterministe.

Puis nous avons montré que ces conclusions sont vraies pour le schéma coopératif le plus simple. Nous effectuons l'optimisation des performances, en prouvant que le protocole est meilleur et dans le sens des métriques, et dans le sens de la complexité de calcul.

De plus, toutes les comparaisons entre les schémas avec et sans retransmissions sont faites dans un contexte énergétiquement juste.

En outre, les zones de goodput réalisables sont calculées pour différents code rate, et sont optimisées. Il est ainsi possible de déterminer la meilleure combinaison de transmissions crdit et les taux de codage à la source et au relais.

On peut approfondir les études dans ce domaine, en commençant par d'autres protocoles pour les réseaux avec trois noeuds, afin de trouver quel protocole probabiliste sera plus efficace. Puis on peut poursuivre le chemin en rajoutant plusieurs noeuds, pour trouver le nombre de relais le plus optimale, qui permettra d'obtenir les meilleures performances, en gardant la dépense d'énergie au niveau le plus bas possible.

De plus, les études peuvent être poursuivi en examinant l'effet de plusieurs antennes au niveau de l'émetteur. Normalement, on doit obtenir des gains de diversité. Cependant, il est possible d'obtenir ces derniers en utilisant seulement une antenne sur chaque côté du système, grâce au relais.

Une autre direction peut être prise, afin d'étudier l'effet de la distance entre la source, le relais et la destination.

Enfin, il est possible d'examiner l'effet du taux de code convolutif sur les performances, pour les différents types de modulation pour les canaux réalistes, avec le canal d'acquittement non-idéal.

Part II

HARQ analysis in the context of cooperative relaying

Chapter 1

Introduction

1.1 Introduction

The evolution of wireless technologies is driven by the need of high data rate and more and more heterogeneous services, which imply that the network operators provide more and more bandwidth and power in order to satisfy the user expectations.

Over the past decade the competition between the mobile operators and equipment manufacturers resulted in the emergence of 2.5G technologies, such as GPRS, EDGE (Enhanced Data for GSM), standards for mobile communications. With the increasing demand of data transmission rate, in the mid 1990s Universal Mobile Technology System (UMTS) was introduced, permitting multimedia data transmission through wireless high-speed links with 2000 Kbytes/s.

Then, 3G technologies have been introduced, which were still enhancing the 2G networks evolution. The primary technologies are CDMA2000 1X/EVDO, UMTS-HSPA+.

Moving beyond 3G, the LTE (Long-Term Evolution) and LTE-Advanced offered the next generation capabilities. It was released by 3rd Generation Partnership Project (3GPP), with the latest release in 2008. It is a combination of various techniques, such as: Orthogonal Frequency Division Multiple Access (OFDMA), high order modulations (64-QAM), large bandwidth (up to 20 MHz) MIMO transmission in the downlink (up to 4×4), and Hybrid Automatic Repeat reQuest (HARQ), etc., in order to achieve high data rates.

4G, on the other hand, is an ultra-high-speed broadband wireless network. The deployment of cooperative relay-based technology is an emerging and promising approach allowed in 4G networks. Cooperative networks consist of the transmitter, receiver, and some intermediate nodes, or *relays*, which can provide spatial diversity, and along with the decoder techniques attain achievable diversity gains. The *relay station* is either a moving mobile node that is used to aid in the data exchange between the base and the mobile, for example in Vehicular Ad-hoc NETWORK (VANET) or in Mobile Ad-hoc NETWORK (MANET), or a fixed station, e.g. in a Wireless Local Area Network (WLAN). Depending on the various relay operation modes, (e.g. Amplify-and-Forward, Decode-and-Forward, etc. [22, 23]), the Quality of Service (QoS) performance of such networks can be quite different.

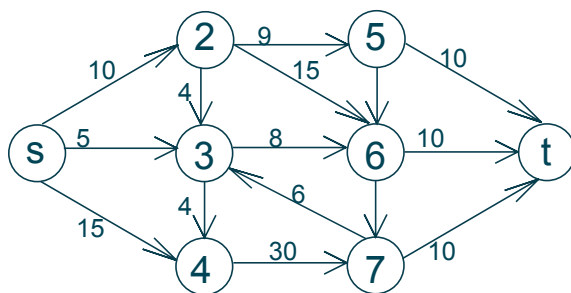


Figure 1.1: An example of a directed graph

The first prototype of relaying technology has been introduced by Edward C. Van Der Meulen in 1971 [24], where one terminal helps in transmitting data to the destination terminal, by decoding and re-encoding it using a binary XOR operation.

Recently the interest in the research community has greatly increased towards the network coding (NC) techniques, due to the spectrally efficient bandwidth utilization, that can be achieved. The breakthrough in this field is considered to be done by Ahlswede in [1], which motivated network coding due to the increased multicast capacity. The authors show that treating network nodes as encoders is more useful than limiting their function to that of a switch. The model developed in [1] can be considered as the generalized case of multilevel diversity coding system [25] without distortion.

Diversity coding systems with distortion have been studied by [25, 26]. Typically, in a diversity coding system an information source is encoded by a set of encoders. There is a set of decoders, each of which can access only a subset of encoders. Then each decoder reconstructs the source. The success and the quality of the decoding depends on the distortion of the system. The diversity coding systems are used in fault tolerant applications, i.e. instead of storing the data in the same location, it is distributed and saved at different locations [25]. However the diversity coding is out of scope of this work, and for a deeper study on this subject, we kindly refer our reader to [25, 26].

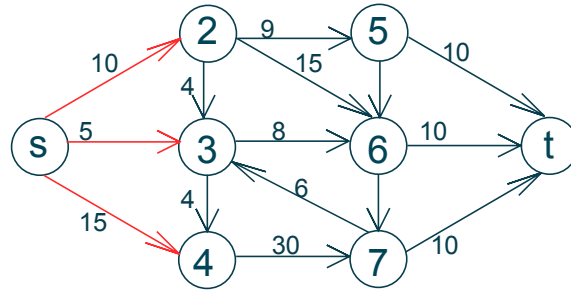
The main goal of the authors in [1] is the derivation of admissible code rates for relay networks. Furthermore, in their work the improvement of the throughput is shown with the network codes. They also provide a class of network codes that are referred to as α -codes.

The main result of their work can be viewed as max-flow min-cut theorem in the information-theoretic context. Before going in further details, let us introduce several useful notions that will be used in what follows. Assume a directed weighted graph G defined by the set of edges E and vertices V , i.e. $G = (E, V)$. An example of such a graph is given in Figure (1.1). As it can be observed from the figure, each edge has a weight. The set of vertices in the graph in Figure (1.1) is $V \in \{s, t, 2, 3, 4, 5, 6, 7\}$ and the set of edges is, respectively

$$E \in \{(s, 2), (s, 3), (s, 4), (2, 3), (2, 5), (2, 6), (3, 4), (3, 6), (4, 7), (7, 3), (5, 6), (5, t), (6, 7), (6, t), (7, t)\}.$$

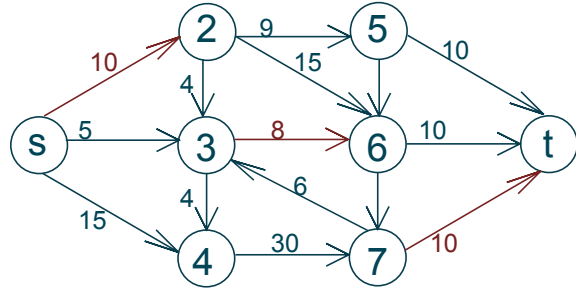
The edges of the graph are weighted.

Definition 1. A cut in a graph is a partition (S, T) , such that $s \in S$ and $t \in T$.



Capacity of (S,T) = sum of weights of edges leaving s
(S,T) cut capacity = 30

Figure 1.2: An example of a cut in a directed graph



Capacity = 28

Figure 1.3: Min-cut in a directed graph. Capacity of min-cut is equal to 28

Definition 2. The capacity of a cut (S,T) in a graph is equal to the sum of weights of the edges that leave S .

For the example given in Figure (1.2) the capacity of the (S,T) cut is 30.

Definition 3. The min cut of a graph corresponds to an (S,T) cut with minimal capacity.

In order to find the minimal cut of a graph, one would have to remove the **best** set of edges so that no flow can pass in this graph. For the example in Figure (1.1), the min-cut of the graph is illustrated in Figure (1.3).

A network can be abstractly represented by a directed finite weighted graph. A flow in a network is a real function $f : V \times V \rightarrow \mathcal{R}$.

Definition 4. A flow $\{(u,v)$ in a network is the assignment of weights to the edges of the graph, with the following two constraints:

1. conservation of flows: $\sum_{u:(u,v) \in E} f(u,v) = \sum_{u:(v,u) \in E} f(v,u)$, i.e. the inflow to a vertex v has to be equal to the outflow from the vertex v
2. capacity constraint: $f(u,v) \leq c(u,v)$ the flow of an edge has to be less than or equal to the capacity $c(u,v)$ of the edge

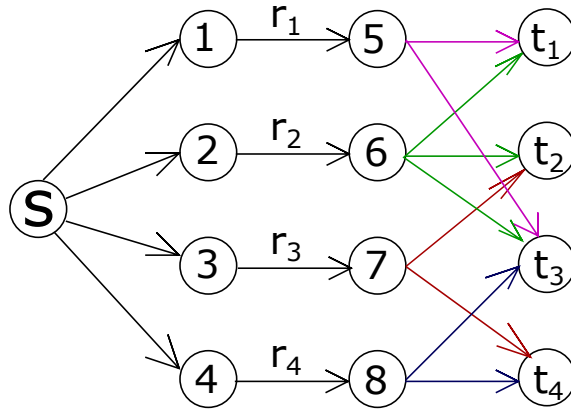


Figure 1.4: Single-level diversity coding system graph representation

Definition 5. A max flow problem consists in finding the flow that maximizes the net flow into the sink.

Theorem 1. Max-flow of a weighed directed finite graph is equal to the min-cut of this graph.

The authors represent a network in a form of a directed graph $G = (V, E)$ with set of edges E and set of nodes V of a point-to-point communications network. The information is assumed to be sent noiselessly from any i to j , $(i, j) \in E$.

They address a one-level diversity coding system (or the so-called single-source problem), where they give the necessary conditions for achieving the coding rate $r = [r_1, \dots, r_m]$. The graph representation of the single-source diversity coding system is given in Figure(1.4).

So, in [1] the authors address a specific case of a diversity coding system with only one level. The authors give a graph-theoretic interpretation of the above described problem. In the second part of their work a class of α -codes is constructed.

The main result of the authors in [1] is that in the information-theoretic context, the nodes should not be exploited only as switches, but they can be used to also encode the incoming data flows.

Various network coding techniques in wireless cooperative relay networks have been examined in [1, 10–12, 27]. The idea of network coding is usually illustrated using the well-known butterfly network example [1]. A butterfly network consists of two sources, two sinks and two relays as shown in Figure 1.5. Each source has to transmit data that to a specific destination. We can see from the figure, that instead of sending b_1 and b_2 separately (e.g. in different timeslots, or in different frequencies), the nodes 3 and 4 send the network-coded version of these two bits. The very intuitive example of a network code is the binary XOR operation, defined on Galois field $GF(2)$. As we can notice, the sinks t_1 and t_2 are capable of perfectly recovering (in [1] noiseless channels are assumed) both bitstreams, since they also have a direct copy. In such a way, one timeslot is saved, which results in a more efficient utilization of the resources.

Depending on the implementation considerations, there are different possible ways to realize the network coding. The most straightforward way is to use an XOR operation, which is defined on a binary Galois field ($GF(2)$). This approach has been used in [10–12, 28]. In [28] the authors use the XOR operation since the complexity of the codes is linear and depends on the length of the message. The goal of their work is to achieve min-cut capacity for a very special type of networks: combination networks, [28].

Another type of network coding, the combination network codes (CNC) is studied in [29], where the authors derive the maximal achievable throughput improvement and routing cost reduction via an upper-bound. The CNC is a type of network codes which are applied to combination networks. Combination networks have a $C_{n,k}$ structure, which is given by an **undirected** graph, that has a three-layer topology: 1) the first layer is the source; 2) the second layer are the n relays; and finally 3) $\binom{n}{k}$ receivers. Each relay is connected to the source, and each receiver is connected to a different **set** of k relays. An illustration of combination network topology is given in Figure (1.6). Note that the link directions are there only to denote the multicast flow of the combination network code, i.e. the underlying network topology is **undirected**. As we can see from the figure, the terms x and y represent the uncoded data, whereas the $x + y$ and $2x + y$ are the network-coded data flows over a finite field. All the decoders are able to recover the original data based on what they receive.

The authors of [29] derive the **coding advantage** (the throughput improvement of network codes), and the **cost advantage** (the minimum routing cost necessary to achieve the desired throughput), with the help of analyzing the cost advantage instead of the coding advantage. This is done because the analysis of the coding advantage would require **multicast tree packing**, which in turn becomes too complicated for the undirected networks.

Definition 6. *Multicast tree packing problem is stated as follows: for a given set of*

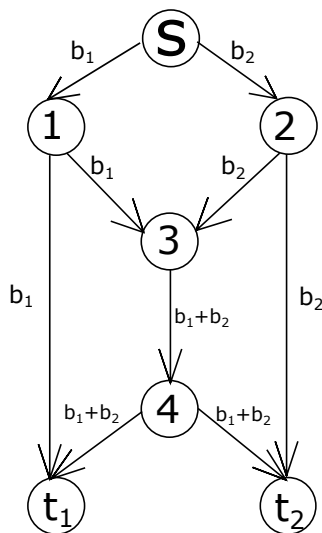


Figure 1.5: Butterfly network graph representation

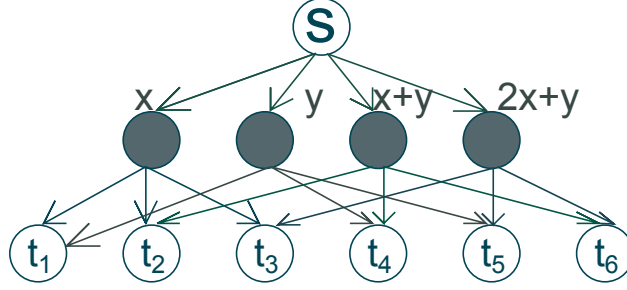


Figure 1.6: Example of $C_{4,2}$ Combination Network Topology. The terms x and y denote the uncoded information flows; $x+y$ and $2x+y$ the coded information flows. Each receiver t_i can recover x and y from the received data

multicast group M in the graph $G = (V, E)$, $M \subset V$, find a subgraph of G , such that it spans M and has minimum total cost. The subgraph is required to be a tree, and the cost is measured as the sum of weights of the edges in the solution.

Furthermore, the authors in [29] assume that the considered network topologies are **uniform-cost**, which means that the weight of the edges are equal. The authors explain that the undirected network topology though does not represent computer networks, yet it is more interesting for multitude of reasons, such as in contrast to directed networks (where the coding advantage is unbounded), in undirected networks the coding advantage is bounded; undirected network model is simple for a theoretical study, etc.

They derive and prove the **minimum multicast cost with network coding** and the **minimum cost of a multicast tree** in uniform-cost $C_{n,k}$ networks, and compute the **cost advantage** of the network coding as the ratio between two of them in a form of an upper-bound.

Interestingly, the authors remark that the traditional butterfly network from the work of Ahlswede [1], Figure 1.5, is isomorphic to a $C_{3,2}$ network (Figure 1.7) with only one receiver shifted to a source (Figure 1.8). As it is highlighted in [29], in an undirected multicast network with given network topology, link capacities, and terminal node set, the maximal achievable multicast throughput is independent of the choice of the source from the receiver set.

While most of the literature focuses on attaining better multicast throughput and/or capacity [3–6], another direction that has been started off with the network codes is obtaining diversity gains [7–12]. Practical approach has been exploited in [5, 6] by proposing new architectures for wireless networks (e.g. COPE, MIXIT, ANC [6]), together with polynomial time algorithms for encoding and decoding, [13]. In [13] Koetter and Medard showed that codes with a simple linear structure are sufficient to achieve capacity in the multicast problem. Other types of network coding propose mixing data on the symbol level [14], or by using analog modulation with lack of synchronization between the interfering signals and applying this to software radios [15], or using random network codes [16, 17].

The works [7–9] pointed out that in order to obtain diversity gains cooperative networks, the binary XOR operation is not optimal, and that non-binary linear block codes help in achieving the desired diversity gains.

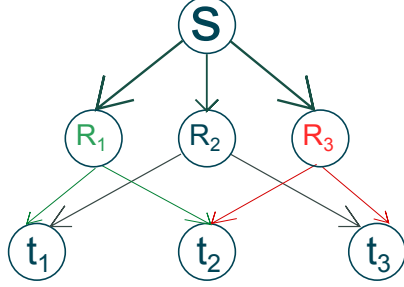


Figure 1.7: Example of $\mathcal{C}_{3,2}$ undirected combination network topology. The link directions only denote the multicast flow, but the underlying topology is undirected

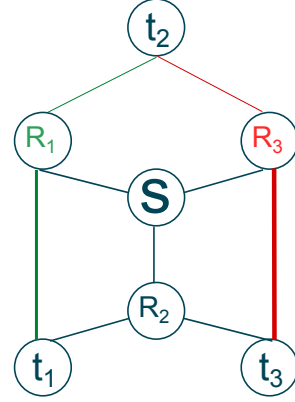


Figure 1.8: Representation of $\mathcal{C}_{3,2}$ Combination as Butterfly Network. The multicast throughput is not affected by the change of source and the receiver

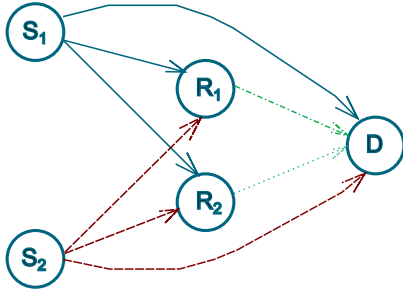


Figure 1.9: Multi-user multi-relay cooperative network. The different line styles (and colors) denote timeslots. In the first timeslot S_1 broadcasts to R_1 , R_2 and D . In the second timeslot S_2 broadcasts to R_1 , R_2 and D . In the third timeslot R_1 transmits to D , and in the fourth R_2 to D .

In [9] the authors consider the so-called **Generalized Dynamic Network Codes**, (GDNC). Their work was motivated by [8], where diversity gains were attained for a multiuser cooperative scheme with the help of **Dynamic Network Codes**, (DNC). In [8] the system consists of M users that act as relays. Each of the users has independent data to be transmitted to the destination, and in the first timeslot the data from all nodes is sent in broadcast mode to each other and to the base station. From second until the M -th timeslots the users transmit $M - 1$ nonbinary linear combinations of the data that it has received to the base station. The work in [8] shows that the usage of nonbinary linear network codes attains better diversity than that of the binary linear network codes. In [9] the authors extend the idea of DNC to the GDNC, by allowing the users to send several original packets instead of one; and several encoded packets instead of one. By doing this, they have longer codewords, and are able to achieve better rate and diversity, than that of the DNC.

However, the authors [10–12] proved that it is indeed possible to attain diversity gains by using the simple XOR operation defined at the GF(2), which keeps the computational complexity of the system quite low and can be applied to unequal error protection in multi-user multi-relay cooperative relay networks, [10, 11, 18].

In [10–12] the authors study the effects of binary network coding in cooperative relay networks, and derive the average bit error probability (ABEP) at the PHY layer, using approximations in order to obtain asymptotically-tight estimates.

The network model that is considered in [10–12], is based on two sources, two relays, and one destination as illustrated in Figure (1.9).

The authors consider that the relay does not check if the bits it receives from the sources are correct or not. In [10] the authors derive the ABEPs for ideal source-to-relay channels, whereas in [11] they derive the ABEPs for realistic (i.e. non-ideal) source-to-relay channels. Furthermore, in all these works the authors consider four scenarios, which represent the a) relay-only, b) XOR-only, and c) UEP based approaches. These approaches differ depending on the data that the relay sends, and their brief description is given below.

Scenario 1: R_1 and R_2 send the data of respectively S_1 and S_2 to the destination in orthogonal timeslots. In other words, R_1 sends b_1 ; and R_2 sends b_2 observation.

Scenario 2: R_1 and R_2 send the network-coded version of S_1 and S_2 data to the destination in orthogonal timeslots. The binary XOR operation is used to introduce network coding. In other words, the relay sends $b_1 \oplus b_2$.

Scenario 3: R_1 sends the network-coded version of b_1 and b_2 observations; and R_2 send the direct observation of b_1 .

Scenario 4: R_1 sends the direct observation of b_1 ; and R_2 send the network-coded version of b_1 and b_2 observations.

All these scenarios are evaluated in terms of the end-to-end average bit error probability of each source, for the three types of decoders at the destination: a) MDD, b) H-MLD, and c) S-MLD. The work principle of these decoders is given below, however more detailed explanations can be found in the related literature, [10–12].

1.1.0.3 Received Signal Model

Before describing the decoding techniques used in [10–12], let us introduce shortly the system model that is considered in all these works.

The authors consider BPSK modulation as: $x_{S_i} = \sqrt{E_s}(1 - 2b_{S_i})$, where $\sqrt{E_s}$ is the energy per modulated symbol and b_{S_i} is the original information bit generated at source i . The data received on all the nodes after the first two timeslots is given by:

$$\begin{aligned} y_{S_i D} &= h_{S_i D} x_{S_i} + n_{S_i D}, \\ y_{S_i R_1} &= h_{S_i R_1} x_{S_i} + n_{S_i R_1}, \\ y_{S_i R_2} &= h_{S_i R_2} x_{S_i} + n_{S_i R_2}, \end{aligned} \tag{1.1}$$

where h_{XY} is the fading coefficient from node X to Y , given by a circularly symmetric Gaussian R. V. with zero mean and $\sigma_{fading,XY}$ variance per dimension; n_{XY} are the complex AWGN coefficients from node X to node Y ; and finally y_{XY} are the received symbols at node Y from node X .

The relay performs coherent demodulation of the received symbols as:

$$\begin{aligned}\hat{b}_{S_i R_1} &= \underset{\tilde{x}_{S_i} \in \{-\sqrt{E_s}, \sqrt{E_s}\}}{\operatorname{argmin}} \{ |y_{S_i R_1} - h_{S_i R_1} \tilde{x}_{S_i}|^2 \}, \\ \hat{b}_{S_i R_2} &= \underset{\tilde{x}_{S_i} \in \{-\sqrt{E_s}, \sqrt{E_s}\}}{\operatorname{argmin}} \{ |y_{S_i R_2} - h_{S_i R_2} \tilde{x}_{S_i}|^2 \},\end{aligned}\tag{1.2}$$

where $\tilde{\cdot}$ represents the symbol in BPSK constellation (i.e. trial symbol used in the detection); and the terms $\hat{\cdot}$ the detected symbol.

1.1.0.4 Considered Decoders

a) MDD, which is a two-level decoder, based on

a.I: coherent demodulation of the sequence,

$$\begin{aligned}\hat{b}_{S_i D} &= \underset{\tilde{x}_{S_i} \in \{-\sqrt{E_s}, \sqrt{E_s}\}}{\operatorname{argmin}} \{ |y_{S_i D} - h_{S_i D} \tilde{x}_{S_i}|^2 \}, \\ \hat{b}_{R_j D} &= \underset{\tilde{x}_{R_j} \in \{-\sqrt{E_s}, \sqrt{E_s}\}}{\operatorname{argmin}} \{ |y_{R_j D} - h_{R_j D} \tilde{x}_{R_j}|^2 \},\end{aligned}\tag{1.3}$$

By the end of this phase, independently of the scenario, the destination always has a sequence of four demodulated bits: $[\hat{b}_{S_1 D}, \hat{b}_{S_2 D}, \hat{b}_{R_1 D}, \hat{b}_{R_2 D}] = [\hat{c}_1, \hat{c}_2, \hat{c}_3, \hat{c}_4]$.

a.II: minimum distance decoding of the received network coded bits, which is done by feeding the hard-decision estimates obtained in the first phase, to the minimum distance decoder:

$$[\hat{b}_{S_1}, \hat{b}_{S_2}] = \underset{\tilde{c}_i^k, k \in \{1,4\}}{\operatorname{argmin}} \sum_{i=1}^4 |\hat{c}_i - \tilde{c}_i^k|\tag{1.4}$$

where \tilde{c}_i^k represents \tilde{k} -th bit in the i -th codeword of the codebook. The codebook is based on which scenario is used and contains all possible bit sequences that could have been transmitted to the destination. The codebooks for Scenarios 1 – 4 are given by the corresponding matrix in Table (1.1-1.4).

b) H-MLD, which is also a two-level decoder, based on:

b.I: coherent demodulation of the sequence, given by Eq. (1.5)

Table 1.1: Codebook for Scenario 1

c_i	c_i^1	c_i^2	c_i^3	c_i^4
$c_1 =$	0	0	0	0
$c_2 =$	0	1	0	1
$c_3 =$	1	0	1	0
$c_4 =$	1	1	1	1

Table 1.2: Codebook for Scenario 2

c_i	c_i^1	c_i^2	c_i^3	c_i^4
$c_1 =$	0	0	0	0
$c_2 =$	0	1	1	1
$c_3 =$	1	0	1	1
$c_4 =$	1	1	0	0

Table 1.3: Codebook for Scenario 3

c_i	c_i^1	c_i^2	c_i^3	c_i^4
$c_1 =$	0	0	0	0
$c_2 =$	0	1	1	1
$c_3 =$	1	0	1	0
$c_4 =$	1	1	0	1

Table 1.4: Codebook for Scenario 4

c_i	c_i^1	c_i^2	c_i^3	c_i^4
$c_1 =$	0	0	0	0
$c_2 =$	0	1	0	1
$c_3 =$	1	0	1	1
$c_4 =$	1	1	1	0

b.II: MLSE detection, which is done by feeding the hard-decision estimates obtained in the first phase to the MLSE decoder:

$$[\hat{b}_{S_1}, \hat{b}_{S_2}] = \underset{\tilde{c}_i^k, k \in \{1;4\}}{\operatorname{argmin}} \sum_{i=1}^4 w_i |\hat{c}_i - \tilde{c}_i^k| \quad (1.5)$$

where w_i are coefficients that are derived based on the error probabilities between the source to relay and relay to destination links, and depend on the scenario.

c) S-MLD, which is a one-level decoder, where the demodulation and network decoding are done in one step.

More detailed explanations on the ABEP derivations and approximation [12].

In [10] the authors derived the ABEPs for the ideal source-to-relay channel-aware decoders, which they call benchmark-MDD, benchmark-H-MLD, and benchmark-S-MLD. Obviously, the best performance is achieved by the S-MLD decoder (both for the benchmark scenario and for the non-ideal source-to-relay channels), which is explained by the fact that there are no hard inputs to the decoder in contrast to the MDD and H-MLD cases. In [10] the authors theoretically compute attainable diversity gains, and show practically that these values are attained: for UEP scenarios (3 and 4) they are able to obtain diversity of order 3 for the source with higher level of protection.

In [11] the authors derived the ABEPs for non-ideal source-to-relay channels, for the decoders MDD, H-MLD and S-MLD. They also conclude, that as long as the $S \rightarrow R$ channels are ideal, there is no difference between the scenarios 1 and 2, i.e. network coding on both relays does not provide any benefits and is equivalent to simply relaying data. Furthermore, when the channels between the source and the relay are realistic, then the authors observe that scenario 2 outperforms scenario 1, which means that even the same operation on both relays increases the system robustness. Another observation the authors had is, that when UEP schemes are applied (i.e. scenarios 3 and 4), the improvement in the performance of the source i with highest protection (i.e. diversify

of order 3) comes with the expense of ABEP degradation of source j if compared to the scenario 2 ABEPs.

The analysis of cooperative relay networks in terms of PHY layer BER and diversity in low SNR regime has been carried out in [30, 31]. Outage probability in a closed-form has been derived in [32] for networks without FEC. The authors in [30] consider three terminal network with one source, one relay, and one destination. The relay helps in the communication process by forwarding only a subset of symbols that it received from the source, thus keeping the communication procedure spectrally efficient. Moreover, as opposed to the literature works, the relay always sends the set of symbols (and does not measure the quality of the received data). After the reception of data from the source and from the relay, the destination performs cooperative maximal ratio combining (C-MRC) estimation, and feeds the LLR of C-MRC estimation to the BCJR decoder. An upper bound on the BER is provided in the form of an approximation, using the pairwise error probability computations.

Optimal joint network and channel decoding (O-JNCD) for cooperative networks has been studied in [33] where the authors derive optimal decoders, and it is shown that better bit error rates are obtained compared to the traditional JNCD [34] in Rayleigh fast and block fading channels. JNCD has been also studied in [35], where the authors obtain better throughput in Rayleigh block fading channels for multi-user HARQ schemes.

In all above mentioned works the stress is made on the cooperative system performance at the PHY layer. However, in order to have a full understanding of the system performance, one would need also to estimate other QoS parameters, e.g. the frame error rate at the MAC layer, packet error rate at the IP layer, delay, etc. Unfortunately, these works do not provide us with this information. Furthermore, we are interested in what would happen if the source and the relays are able to receive feedback from the destination, i.e. when (H)ARQ is involved. Would it be useful to encode the FRAGs with NC? Would it still be possible to attain diversity gains if we use channel-aware detectors? In order to do this, we derive channel-aware detectors for various C-(H)ARQ protocols that we define in the following chapters. Furthermore, since our comparisons include schemes with retransmissions, and without, we also perform energetic fair comparisons: the energy to convey one information bit is computed both for C-(H)ARQ and C-FEC schemes.

An emerging topic in the field of NC is the combination of (H)ARQ and NC in order to benefit from the advantages of both techniques. Such approach has been used in [27, 36, 37].

Ref. [36] considers a wireless broadcast network with one source and multiple receivers, as illustrated in Figure 1.10. They propose a) to network code several lost packets of individual receivers into one using XOR operation, b) network code these resulting packets that belong to different receivers into one and forward to the destination, along with the indexes of the receivers. In such a way each receiver knows how to decode the packets.

This scheme proposed by [36], is an improved version of a so-called NC-HARQ protocol given in [38], where the authors were simply network coding one lost packet of each receiver and forwarding to the destination. In the first m timeslots the source broadcasts the data. By assuming that the ACK/NACK messages are perfectly received by the source, and that it knows to which destination they belong, the source XORs all the lost packets and sends them as one, along with the indexes. As we might notice, in each retransmission

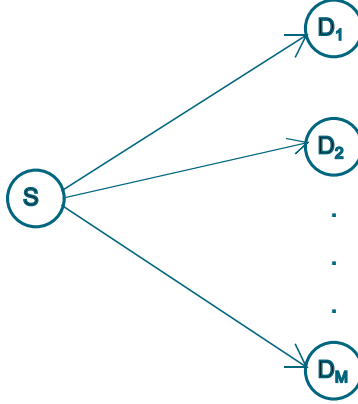


Figure 1.10: Cooperative relay network with one source and m destinations

the XOR contains only one lost packet per individual receiver. The authors in [36] also propose instead of always sending the XOR of several lost packets, sending a) XOR in the first retransmission, b) original packet in the second transmission. In such a way they are able to obtain diversity gain, and obtain better throughput than that of [38].

In [27] the authors propose distributed HARQ protocols for DF (Decode and Forward) links with network coding, and succeed in reducing the latency and improving the throughput.

In [37] the authors propose a protocol for a two-source, one relay and one destination scheme by exploring the limited feedback from the destination and relay. Instead of using XOR, the authors provide network coding via signal superposition modulation. When the data is not correctly received at the destination (i.e. source and relay receive NACK from the destination), the relay regenerates incremental redundancy frame in case if it has correctly decoded it; otherwise the source sends the original fragment. The relay sends incremental redundancy to the destination using the signal superposition modulation. The maximal number of retransmissions is set to one, which implies that if after retransmission from the relay the data is still not successfully received, the respective fragment is dropped. They consider two case-studies: a) the source to relay channels are perfect, b) source to relay channels are fixed to have $\text{SNR} = 16$ dB. In both cases they compare the cooperative HARQ scheme to the transmission without cooperation, and demonstrate that their method provides better results in terms of frame error rate. However in this work, the benefits of XOR operation are not explored, thus leaving the reader with the same question: "What will happen if we use XOR and HARQ together?"

Despite the large number of the works in this emerging field, they are usually focused at the performance evaluations of the combination between ARQ and network codes. Naturally, a question can be raised: "is it really useful to have the data network coded when the retransmissions are already present in the system?". Unfortunately in the literature this question has not been addressed, and this will be the main and most important goal of this work. Other contribution of this work is that we perform energetic fair comparisons between the schemes that are based on ARQ and that are not. Usually in the literature these comparisons are done without accounting for the fact that retransmitting results

in extra energy cost. We derive a metric that considers the total energy expenditure per successful information bit.

1.2 Performance Metrics Definitions

In this section we define the quality of service performance evaluation metrics that will be evaluated in this thesis. Note that the definitions are given in a generic manner, and can be applied both to single-source single-relay networks, and multi-source multi-relay networks.

We are focusing on the following performance evaluation metrics:

- frame error rate (FER) at the MAC layer and IP layers
- efficiency computed as a function of correctly received FRAGs per total number of transmitted FRAGs (i.e. not taking into account the code rate at PHY layer), which is called η_{ARQ}
- efficiency computed as a function of correctly received information bits per total number of transmitted bits (i.e. by taking into account the code rate at PHY layer), which is called η_{gen}
- average delay per successful FRAG (or packet)
- average delay per FRAG

The frame error rate at the MAC layer is defined as the probability that a FRAG, belonging to a source is not received at the destination after the maximal number of transmissions associated with that FRAG is expired. In the simulated model this entity is computed as follows:

$$\text{FER} = \frac{\text{number of erroneous FRAGs received from source } i}{\text{total number of FRAGs transmitted (from source } i \text{ and relay)}} \quad (1.6)$$

Please note that this notion requires the knowledge of the source to which the FRAG belongs, since in this work the main focus is on schemes with several sources and a single destination. In case if the number of receivers (aka destinations) is more than 1, then the FER has to be defined accordingly.

The packet error rate at the IP layer is defined as the probability that a FRAG, belonging to a source is not received at the destination after the maximal number of transmissions associated with that FRAG is expired. In the simulated model this entity is computed as follows:

$$\text{PER} = \frac{\text{number of erroneous packets of source } i}{\text{total number of packets transmitted for source } i} \quad (1.7)$$

The efficiencies η_{ARQ} and η_{gen} are defined below:

- η_{ARQ} , or the so-called ARQ-efficiency, which is defined as the ratio between the number of correctly received information bits belonging to source i and the total transmitted information bits (related to the source i). This efficiency accounts only for the retransmissions but not for the redundancy introduced by the channel codes.
- η_{gen} , the general efficiency that is defined as the ratio between the correctly received information bits belonging to source i and the total transmitted coded bits (related to the source i). This efficiency accounts both for the retransmissions and the code rate at the PHY layer.

As it will be clear later, we need these two definitions in order to derive the new performance metric, which we call **true** E_b/N_0 and which is used in order to provide energetic fair comparisons between schemes that have retransmissions and the ones that do not. Below we give the expressions allowing to evaluate these two efficiencies:

$$\begin{aligned}
\eta_{\text{ARQ}} &= \frac{\text{number of correctly decoded information bits belonging source } i}{\text{number of total transmitted information bits (from source } i \text{ and relay)}} \\
\eta_{\text{gen}} &= \frac{\text{number of correctly decoded information bits belonging source } i}{\text{number of total transmitted coded bits (from source } i \text{ and relay)}}
\end{aligned} \tag{1.8}$$

Please note that since these above mentioned quantities are expressed in terms of the bits, their definitions are universal both for IP and MAC layer. So, in the next chapters, when we speak about MAC (IP) layer metrics, the efficiencies are also given at the MAC (IP) layer.

Another important quality of service metric in a communications network, is the average delay. In this work, we define two different delays at the MAC and IP layers:

- \bar{S} , the average number of FRAGs transmitted per successful MAC FRAG (or IP packet), for source i , or the so-called delay per successful FRAG (these terms will be used interchangeably)
- \bar{T} , the average number of FRAGs transmitted per MAC FRAG (or IP packet), for source i , or the so-called delay per FRAG (these terms are used interchangeably in further)

These metrics are evaluated for i -th source as shown below:

$$\begin{aligned}
\bar{S} &= \frac{\text{number of transmitted FRAGs (or packets) belonging to source } i}{\text{number of correctly received FRAGs (or packets) that belong to source } i} \\
\bar{T} &= \frac{\text{number of transmitted FRAGs (or packets) belonging to source } i}{\text{number of FRAGs (or packets) generated at source } i}
\end{aligned} \tag{1.9}$$

In this work we use cross-layer optimized approach, which has been introduced in [20, 21, 39–41]. This approach is based on the following idea [20, 21]: the IP layer packet is dropped when even one FRAG associated with that packet is not successfully received at the destination. This optimizes the bandwidth utilization of the system, in the sense that the IP packet cannot be recovered if even one FRAG is missing, so there is no use in sending additional FRAGs that belong to the same IP packet.

Please note, that in order keep the notation simple, we drop the index which denotes at which layer is the metric evaluated, but it is always stated whether it is at IP or MAC layer.

1.3 Thesis Motivation

The extensive advance in the relaying has driven the research to another technique, namely the Cooperative ARQ. As the name suggests, it is made up of retransmitting the incorrectly received chunks of data when needed in the cooperative relay networks. The majority of the works in this domain focuses on optimal deployment of the relay to obtain better error rate performance usually at the MAC layer.

Most of the works in the literature are focused at the performance study and evaluation of cooperative ARQ without FEC. In this work we provide extensive analysis and performance evaluation of cooperative HARQ in the following context:

- no cooperation with FEC and without ARQ, which is referred to as FEC in further
- cooperation with FEC and without ARQ, referred to as C-FEC
- cooperative HARQ with network codes, referred to as C-HARQ-NC
- cooperative HARQ without network codes, referred to as C-HARQ.

In other words, we are interested in comparing these techniques in order to observe the behavior of network coding in cooperative (H)ARQ.

Normally the works available in the literature focus at the performances of C-ARQ. Furthermore, in these works the performances are often compared to that of non-ARQ schemes. Unfortunately these works do not take into account the fact that the retransmissions cost extra energy. In order to account for this, we derive a metric that is called **true** E_b/N_0 . More explanations follow in Chapter 2 of this thesis.

The contributions of this work are as follows:

1. energetic fair criterion derivation and application to point-to-point networks STBC, STBC-FEC, and STBC-ARQ schemes
2. cooperative relay scheme combined with FEC and ARQ performance analysis and evaluation via finite state machine
3. various cooperative ARQ protocols definition and QoS evaluation in energetic fair context

The publications during this work:

Conference Proceedings

- A. Vanyan, F. Bassi, A. Herry, P. Duhamel, "Coding, diversity and ARQ in fading channels: a case-study performance comparison", IEEE Symposium on Personal, Indoor, Mobile, and Radio Communications, (PIMRC), September 2013.

Chapter 2

Energetic fair criterion derivation

In this chapter we compare the performances of ARQ combined with STBC schemes in an energetic fair context, for a one-source one-destination network scenario. The following schemes will be compared: ARQ, STBC-ARQ, STBC. We use the notation ARQ for the schemes that consist only of retransmissions (without PHY layer coding); STBC-ARQ to denote the schemes where the erroneous uncoded data are retransmitted from multiple antennas; and finally STBC to denote the schemes where there are no retransmissions occur, and the data is sent from multiple antennas to attain diversity. Usually in the literature these schemes are compared to each other without taking into account the fact that retransmissions have a cost in terms of energy consumption per successful information bit. In other words, the comparisons in the literature are limited to those as a function of SNR or E_b/N_0 . However, in a scheme without retransmissions, the total energy consumed in order to successfully receive one information bit, is different from the energy per successful information bit in ARQ or HARQ schemes. In order to make those comparisons fair, we introduce a new metric, that takes into account the above mentioned fact, and makes the comparisons between these schemes energetically fair.

The aims of this chapter are twofold: a) derive the energetic fair comparison metric, b) apply to the point-to-point networks and perform comparisons between them. The metrics obtained in this chapter will then be used to describe cooperative relay schemes.

The simplest form is Type I HARQ [42], where a fixed-rate error correcting code is applied on the data, and the encoded packets are processed as in conventional ARQ.

Not surprisingly, the most efficient use of the channel resources is obtained for mild error correction capability at the physical layer, adaptively enhanced, in case of bad channel occurrences, by the ARQ retransmissions [43].

As it is known from the literature, physical-layer reliability can be enhanced as well by the use of Multiple-Input Multiple-Output (MIMO) systems [44,45]. Used in combination with ARQ techniques, they provide a cross-layer error control strategy whose performance can be characterized by the diversity-multiplexing-delay trade-off [46]. We are interested here in the diversity gain achieved by full-rate Space Time Block Coding (STBC) [47,48]. When used in combination with ARQ protocols, STBC enhances physical layer reliability – as channel coding does in HARQ systems – without adding systematic redundancy to the data stream.

Performances are assessed considering chosen Quality of Service (QoS) metrics: the efficiency, defined as the ratio between the data-link goodput and the physical layer throughput; the Packet Error Probability (PER) at the network layer; the delay at the network layer, defined as the average number of data-link fragments necessary to the delivery of a network packet. These quantities are usually evaluated in terms of the channel SNR.

2.1 Considered setting

We consider radio transmission of BPSK modulated binary streams in a Rayleigh fast flat fading channel. The i -th received symbol is given by

$$y_i = \sqrt{\frac{E_s}{N_0}} h_i s_i + n_i, \quad (2.1)$$

where s_i is the i -th information symbol, $\frac{E_s}{N_0}$ is the per-symbol transmit signal to noise ratio (i.e. it does not include the code rate at the PHY layer), $h_i \sim \mathcal{CN}(0, 1)$ is the instantaneous channel gain, assumed to be i.i.d. across fading blocks, and $n_i \sim \mathcal{CN}(0, 1)$ models the additive white Gaussian noise, i.i.d. across channel transmissions. We assume Maximum Likelihood (ML) demodulation at the receiver.

Consider two transmission schemes, dubbed HARQ and STBC-ARQ, respectively. In both cases, each network (IP) packet is segmented at the transmitter side into N_{FRAG} data-link (MAC) fragments of equal length, each one consisting in L_{FRAG} information bits. The maximal number of transmissions per data-link fragment is fixed to N_S . It is assumed that the feedback channel between the transmitter and the receiver is perfect. If a data-link fragment fails correct reception after N_S transmissions, the remaining fragments belonging to the same network packet are removed from the transmitting queue.

2.1.1 HARQ transmission system

This is the classical Type I HARQ transmission scheme, as defined in [42] and considered in [20, 21]. A header and a Cyclic Redundancy Check (CRC) are added to each L_{FRAG} -length data-link uncoded fragment. The resulting information unit is encoded at the physical layer using a rate R_c convolutional code ($0 < R_c \leq 1$), which results in encoded FRAG of length $\frac{L_{\text{FRAG}}}{R_c}$. The binary data stream is modulated and transmitted over the channel. At the receiver side the same actions take place in reverse order: after demodulation and decoding, CRC check is performed, and a control message is sent to the transmitter to acknowledge correct/incorrect recovery of the fragment (ACK/NACK message). If the sender receives a NACK and the maximum number N_S of transmissions has not been reached yet, retransmission of the entire coded fragment occurs. The receiver is not equipped with a buffer, and erroneous fragments are immediately discarded (*i.e.*, not used for successive combining decoding).

2.1.2 STBC-ARQ transmission system

The header and CRC are added to each L -length fragment, which is modulated, and re-shaped according to the STBC matrix. We assume two transmit and one receive antennas,

and we consider the 2×1 full-rate Alamouti code [47], with the coding matrix shown below

$$G = \begin{pmatrix} x_1 & x_2 \\ -x_2^* & x_1^* \end{pmatrix}, \quad (2.2)$$

where $(\cdot)^*$ represents complex conjugation. We assume Maximum Likelihood (ML) STBC decoding. At the receiver side, after CRC check, a control message (ACK/NACK) is sent to the transmitter, to acknowledge correct recovery or request retransmission. No fragment combining is assumed at the receiver.

2.2 Theoretical performance analysis

The theoretical analysis of the HARQ and STBC-ARQ transmission systems is based on the results in [2, 20, 21]. We are interested in the derivation of the network-layer Packet Error Rate (PER), of the network-layer delay, and of the efficiency. These metrics depend on the Bit Error Rate (BER), defined as the proportion of transmitted information bits (considered at the input of the channel encoder/modulator) resulting erroneous (after channel decoding/demodulation). For the STBC-ARQ scheme the BER has expression [19].

$$\text{BER}_{\text{STBC}} = p^2 (1 + 2(1 - p)), \quad (2.3)$$

with $p = \frac{1}{2} - \frac{1}{2} \left(1 + \frac{2}{\frac{E_s}{N_0}}\right)^{-1/2}$, and where $\frac{E_s}{N_0}$ is the energy spent per transmitted symbol. The BER_{HARQ} needs to be evaluated by simulation, for each coding rate R_c of interest. The probability of a FRAG error is denoted by π , and defined as the probability that a fragment is incorrectly received at the destination after one transmission.

$$\pi = 1 - (1 - \text{BER})^{\text{L}_{\text{FRAG}}}. \quad (2.4)$$

The MAC layer frame error rate is given by π^{N_s} , i.e. the fragment is dropped after the maximal transmissions are expired.

The PER at the IP layer is defined as the frequency of non-decodable packets at the receiver network layer. A network layer packet is declared non-decodable as soon as one of its fragments is rejected at the data-link layer. The PER is hence expressed by:

$$\text{PER} = 1 - (1 - \pi^{\text{N}_s})^{\text{N}_{\text{FRAG}}}. \quad (2.5)$$

The average number \bar{T} of transmissions of a fragment is evaluated as:

$$\bar{T}_{\text{MAC}} = \sum_{k=1}^{\text{N}_s} k (1 - \pi) \pi^{k-1} + \text{N}_s \pi^{\text{N}_s} \quad (2.6)$$

The first term in (2.6) expresses the average number of transmissions of a *successfully received* fragment. The network-layer delay n is defined as the average number of fragment

transmissions associated to a successfully decoded network packet. Its expression is given by:

$$\bar{S}_{IP} = N_{FRAG} \left(\sum_{k=1}^{N_S} k \frac{(1 - \pi) \pi^{k-1}}{1 - \pi^{N_S}} \right). \quad (2.7)$$

The fraction in (2.7) denotes the frequency with which a successfully received fragment is transmitted exactly k times.

In the remainder of this chapter, we drop the index of \bar{S}_{IP} , and simply denote it by \bar{S} .

The efficiency η is defined as the ratio between the data-link goodput and the physical throughput¹. It expresses the average of the channel uses necessary to convey one data-link information bit at the destination, thus measuring the cost of the cross-layer error control strategy. A small value of η indicates that the system introduces a large amount of redundancy (channel coding parity bits *and* retransmissions of the fragments) in order to contrast the effects of the radio link. It is easy to verify that the general efficiency η_{gen} takes form

$$\eta_{gen} = \rho R_c \frac{(1 - \pi^{N_S})}{T}, \quad (2.8)$$

where ρ is a coefficient accounting for the overhead generated by the headers and the CRC of the transmission unit, and R is the rate of the convolutional code ($R = 1$ in the STBC-ARQ scheme).

The performance of the two systems will be assessed by comparison of the QoS metrics PER and \bar{S} , which are conventionally expressed as functions of the average energy per symbol $\frac{E_s}{N_0}$. This quantity, however, is not an exhaustive analysis metric for the considered systems, since it does not provide a measure of the energetic cost of the error protection strategy. Therefore, we define a new metric, $\frac{\bar{E}_s}{N_0}$, as the average energy spent by the system to convey to the destination one correct symbol. It is expressed by:

$$\frac{\bar{E}_s}{N_0} = \frac{E_s}{N_0} \frac{1}{\eta_{gen}}, \quad (2.9)$$

and will be used for performance assessment of the two schemes in the following section.

2.3 Numerical results

The simulation setup is as follows: each network packet consists of 360 information bits; it is fragmented into $N_{FRAG} = 3$ data-link layer fragments, of length $L_{FRAG} = 120$ information bits each. Results do not account for the overhead introduced by the header and the CRC: we make the assumption $\rho = 1$ in (2.8). Performance metrics are obtained both by scheme simulation and theoretical evaluation, making use of the expressions provided in Section 2.2.

¹With *data-link goodput* we refer to the average number of information bits (*i.e.* belonging to data-link fragments) correctly received per time unit. With *physical throughput* we refer to the average number of bits in transit over the channel per time unit.

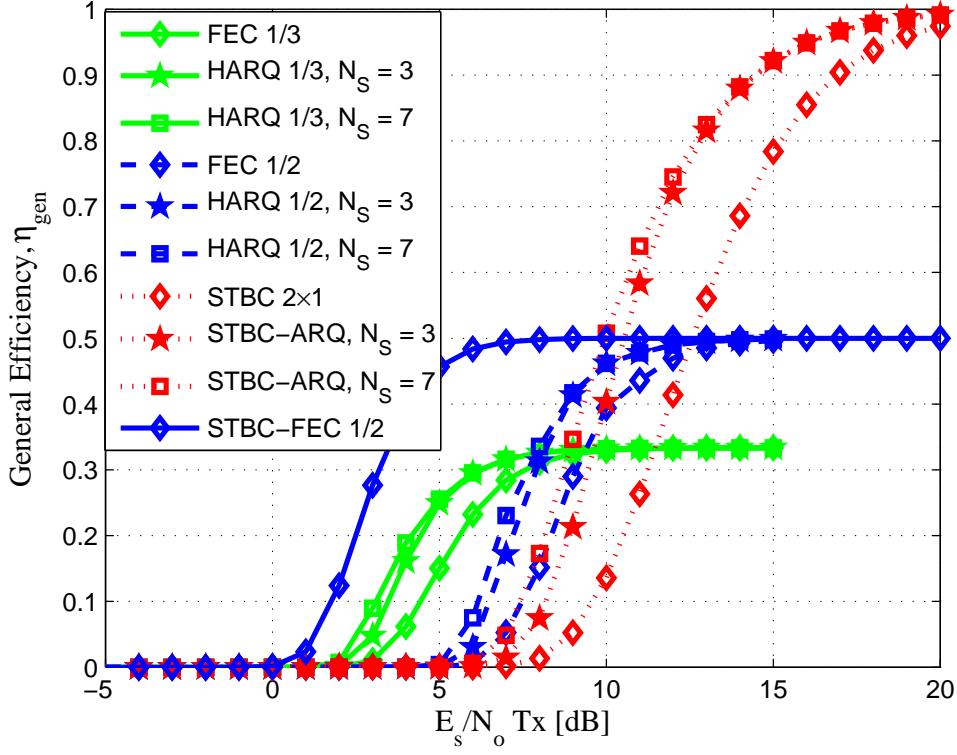


Figure 2.1: Efficiency of the FEC, HARQ, STBC, STBC-ARQ systems, $N_S \in \{3, 7\}$

The theoretical evaluation of the HARQ scheme is obtained injecting, where appropriate, the values of the BER obtained by simulation of the physical layer. In the HARQ scheme, each fragment is encoded with a convolutional code of rate $R_c = 1/3$ or of rate $R_c = 1/2$ (the transmission of a data-link fragment thus requires L_{FRAG}/R_c bits over the channel). The HARQ and STBC-ARQ schemes are considered for $N_S \in \{3, 7\}$. They are compared with the standard Forward Error Correction (FEC) scheme, obtained using the same convolutional code of rate R_c ; with the STBC scheme; with the STBC-FEC scheme, where the convolutional code of rate R is applied before modulation. In Figures 2.1, 2.2, 2.3, 2.4 and 2.5 the performance of the STBC/STBC-ARQ family is depicted using red, dotted lines; of the FEC/HARQ family for rate $R_c = 1/2$ using blue, dashed lines, and of the STBC-FEC scheme for rate 1/2 using blue, solid line; of the FEC/HARQ family for rate $R_c = 1/3$ using green, solid lines.

2.3.1 Efficiency η

We start the performance analysis considering the efficiency η , defined in (2.8), expressed as a function of $\frac{E_s}{N_0}$. The results are depicted in Figure 2.1.

Denote with η_{\max} the maximum achievable value of η ; after (2.8) it holds $\eta_{\max} = 1$ for the STBC/STBC-ARQ family, and $\eta_{\max} = R_c$ for the FEC/HARQ family and for the

System	$\frac{E_s}{N_0}(0.1 \eta_{\max})$	$\frac{E_s}{N_0}(0.8 \eta_{\max})$
STBC	9.06 dB	15.03 dB
STBC-ARQ, $N_S = 3$	8.14 dB	12.85 dB
STBC-ARQ, $N_S = 7$	7.37 dB	12.71 dB
FEC $R_c = 1/2$	6.87 dB	10.02 dB
HARQ $R_c = 1/2$, $N_S = 3$	6.07 dB	8.74 dB
HARQ $R_c = 1/2$, $N_S = 7$	5.61 dB	8.68 dB
STBC-FEC $R_c = 1/2$	1.52 dB	4.05 dB
FEC $R_c = 1/3$	3.52 dB	6.53 dB
HARQ $R_c = 1/3$, $N_S = 3$	2.68 dB	5.30 dB
HARQ $R_c = 1/3$, $N_S = 7$	2.30 dB	5.19 dB

Table 2.1: Linear region extreme points for the all considered setups.

STBC-FEC setup with rate R_c . The analysis of η allows to identify the operating regions of each system, defined as follows.

1. The *zero-efficiency* region, corresponding to very low SNR. If performed, retransmissions are attaining the maximum number N_S .
2. The *linear* region, corresponding to moderate SNR, where η grows roughly linearly with $\frac{E_b}{N_0}$. If performed, retransmissions help the delivery of correct data. Saturation at N_S retransmissions is less and less frequent as $\frac{E_b}{N_0}$ increases. We approximately identify the linear region as the operating points between $\eta = 0.1 \eta_{\max}$ and $\eta = 0.8 \eta_{\max}$. The limits of the linear region for all setups are detailed in Table 2.1.
3. The *full-efficiency* region, corresponding to high SNR, where the conditions of propagation are most favorable, and (nearly) η_{\max} is achieved. If performed, retransmissions are very rare events.

Remark that for the STBC, FEC, and STBC-FEC schemes the variation of η with $\frac{E_b}{N_0}$ is determined only by the improvement of the data-link goodput (see (2.8)), the physical layer throughput being constant across the three operating regions. On the contrary for the HARQ and STBC-ARQ schemes, for a given value N_S , the variation of η is affected both by the increase of data-link goodput *and* by the decrease of the physical layer throughput, as $\frac{E_b}{N_0}$ increases.

The results in Figure 2.1 and in Table 2.1 show the behavior of η for the considered schemes. At the two extremes, the STBC setup achieves the biggest efficiency ($\eta_{\max} = 1$), but very slowly; the STBC-FEC $R_c = 1/2$ scheme achieves a smaller efficiency ($\eta_{\max} = R_c$), but very quickly. The comparison of η within the STBC/STBC-ARQ and the FEC/HARQ families allows to infer that ARQ error control is beneficial, since it anticipates the linear region with respect to the physical layer error protection baseline (STBC and FEC, respectively). The biggest gain, however, is observed in the transition from no to $N_S = 2$ allowed retransmissions, suggesting that ARQ provides only a residual gain, which is already (nearly) drained out for very modest values of N_S .

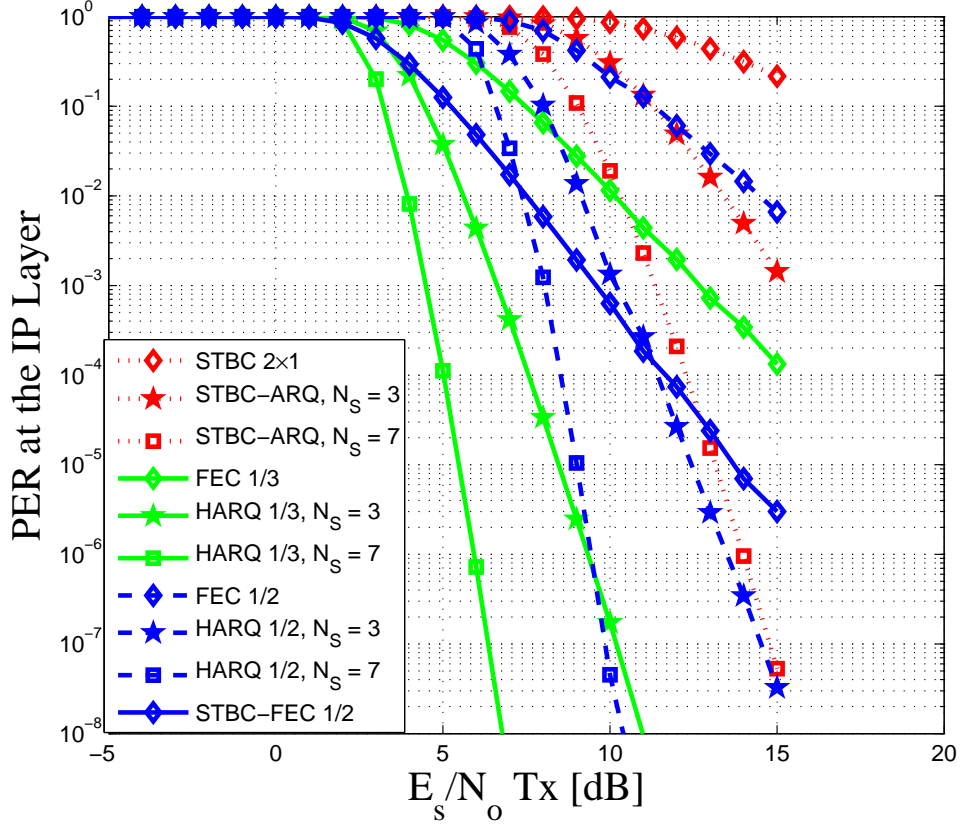


Figure 2.2: PER for the FEC, HARQ, STBC, STBC-ARQ systems, $N_S \in \{3, 7\}$, as function of the $\frac{E_s}{N_0}$

2.3.2 Packet Error Rate

System	$\bar{\xi}$	η	$\eta/0.8 \eta_{\max}$
STBC	15.55 dB	0.556	0.695
STBC-ARQ, $N_S = 3$	13.31 dB	0.587	0.734
STBC-ARQ, $N_S = 7$	12.96 dB	0.635	0.794
FEC $R_c = 1/2$	14.06 dB	0.392	0.98
HARQ $R_c = 1/2$, $N_S = 3$	12.86 dB	0.411	1.027
HARQ $R_c = 1/2$, $N_S = 7$	12.78 dB	0.419	1.045
STBC-FEC $R_c = 1/2$	8.2 dB	0.281	0.937
FEC $R_c = 1/3$	12.34 dB	0.232	0.87
HARQ $R_c = 1/3$, $N_S = 3$	11.09 dB	0.246	0.922
HARQ $R_c = 1/3$, $N_S = 7$	10.92 dB	0.255	0.956

Table 2.2: *Maximum efficiency* points for the all considered setups.

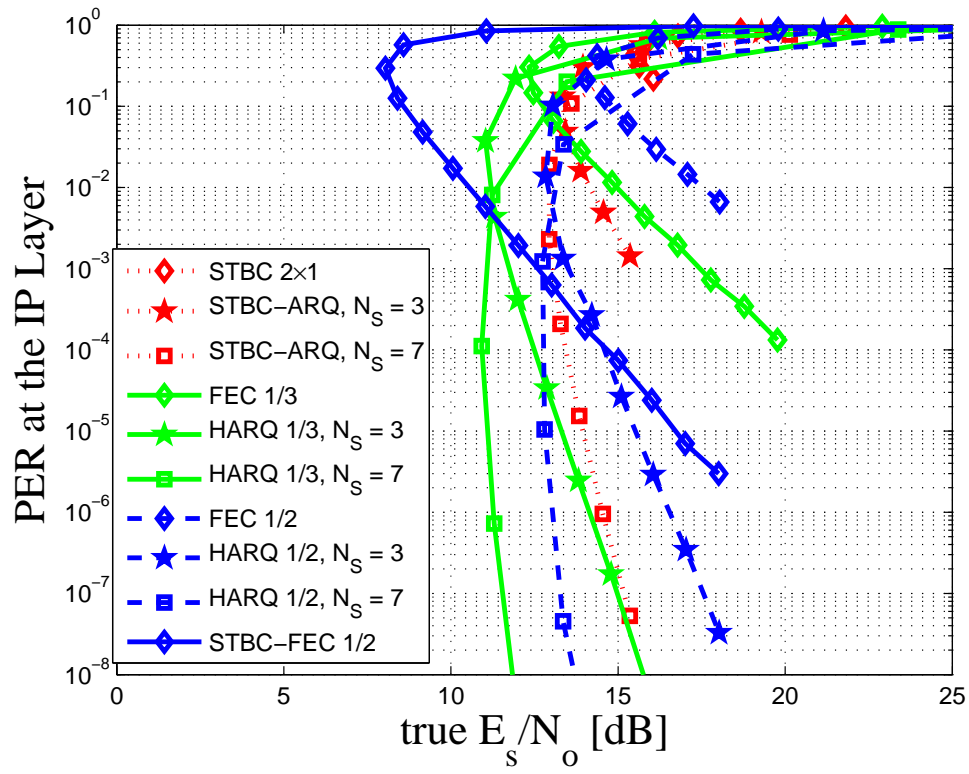


Figure 2.3: PER for the FEC, HARQ, STBC, STBC-ARQ systems, $N_S \in \{3, 7\}$, as function of the $\frac{\bar{E}_s}{N_0}$

The PER, defined by (2.5), is represented in Figure 2.2 as a function of $\frac{E_s}{N_0}$, for all the considered schemes. As expected, in all cases the ARQ strategy allows consistent improvement of the performance, especially transitioning from no to very limited N_S . The performance gain due to retransmissions increases as the target PER decreases. For fixed target PER, the gain obtained increasing N_S is more pronounced in the schemes with weaker physical layer error protection.

Expressing the target PER as a function of $\frac{E_s}{N_0}$ (Figure 2.2) allows the designer, for given N_0 , to determine the energy that needs to be allocated per each use of the channel, in order to match the required QoS. Expressing the target PER as a function of $\frac{\bar{E}_s}{N_0}$ (Figure 2.3) allows, by taking into account the cost of error protection, to analyze the total energy expense for correct information bit, at the chosen QoS operating point.

The C shape of the curves is explained noticing that the same value of $\frac{\bar{E}_s}{N_0}$ corresponds to two possible functioning regimes:

1. the *high SNR* regime, where $\frac{E_s}{N_0}$ has a high value, and the efficiency η is large. This regime is described by the bottom arm of the C-shaped curves in Figure 2.3;
2. the *low SNR* regime, where $\frac{E_s}{N_0}$ is low, and the efficiency η is small. This regime is described by the upper arm of the C-shaped curves.

For each curve, the transition point between the two regimes is the leftmost vertex of the C-shape. This indicates the *maximum efficiency* point, defined as:

$\bar{\xi} = \min_{(\frac{E_s}{N_0}, \eta)} \frac{\bar{E}_s}{N_0}$, which marks the best trade-off between energy cost and efficiency, *i.e.* the most economic use of the allocated energy. Comparison of the values of $\bar{\xi}$ in Table 2.2 with the values of η in Table 2.1 shows that $\bar{\xi}$ roughly corresponds to the point of maximum efficiency in the linear region.

Compare, in Figure 2.3, the performances of FEC, STBC-FEC, and HARQ $N_S = 7$ schemes, $R_c = 1/2$. For $\text{PER} > 10^{-3}$, thanks to diversity gain (increase η without affecting $\frac{E_s}{N_0}$ nor the physical layer throughput), the STBC-FEC scheme outperforms the other setups. For $\text{PER} < 10^{-1}$ the STBC-FEC setup needs, to improve QoA, to boost η by increasing $\frac{E_s}{N_0}$ (see Section ??). The energetic cost $\frac{\bar{E}_s}{N_0}$ per correct information bit thus increases fast as target PER decreases. Also, due to the more favourable propagation conditions, the applied FEC becomes overprotective for most of the fragment transmissions. The same considerations hold true for the FEC scheme as well. Consider now the HARQ, $N_S = 7$ scheme, for $\text{PER} < 10^{-1}$: improved η results *both* from increasing $\frac{E_s}{N_0}$ and decreasing the requests of retransmissions (see Section 2.3.1). Being able to resort to retransmission in case of unfavourable channel occurrences restrains the need to increase $\frac{E_b}{N_0}$ to the point of making the channel code, on average, overprotective. As a result, the energetic cost $\frac{\bar{E}_s}{N_0}$ per correct information bit increases very slowly as QoS demand tightens.

Consider now the performance of the STBC-ARQ for $N_S = 7$ and HARQ $R_c = 1/2$, $N_S = 3$ schemes, for target PER in the interval $[10^{-1}, 10^{-2}]$. As evident from inspection of Figure 2.2, the HARQ scheme requires lower energy per transmitted bit. Figure 2.3, however, shows that when we consider the total transmission energy budget the two schemes achieve roughly the same performance: the bigger redundancy introduced by the

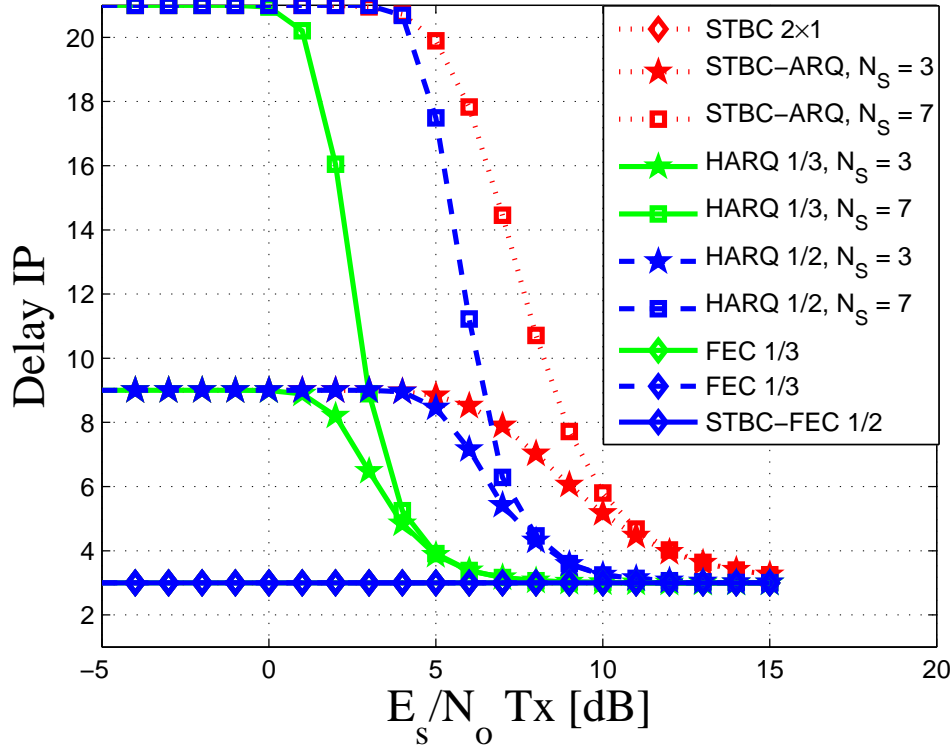


Figure 2.4: Network layer delay for the FEC, HARQ, STBC, STBC-ARQ $N_S \in \{3, 7\}$, as function of the $\frac{E_s}{N_0}$

increased number of retransmissions in the STBC-ARQ scheme has about the same cost as the channel-coding redundancy employed in the HARQ scheme. Moreover, as the QoS requirement becomes more constraining ($\text{PER} < 10^{-2}$), the STBC-ARQ scheme outperforms the HARQ scheme, indicating that as the propagation condition improves (*i.e.* for increasing $\frac{E_s}{N_0}$), rate-adaptation via retransmission results in more energy-efficient than systematic channel error protection.

2.3.3 Network layer delay

In this section we consider the network-layer average delay per successfully decoded IP packet, \bar{S} , defined in (2.7), and depicted in Figure 2.4, as a function of $\frac{E_s}{N_0}$. As $\frac{E_s}{N_0}$ increases, retransmissions become more rare, and \bar{S} decreases, to converge at $\bar{S} = N_{FRAG}$ as soon as the system enters the full-efficiency region (compare with Figure 2.1).

For applications where delay of delivery is a constraining QoS parameter, as *e.g.* in real-time services, the designer needs to determine the optimal trade-off between energy efficiency and timeliness of data reception. To this aim, the representation of \bar{S} as a function of $\frac{E_s}{N_0}$ (Figure 2.5) allows to easily identify the required system parameters. As in Figure 2.3, the curves exhibit the C-shape, where the vertex represents the maximum

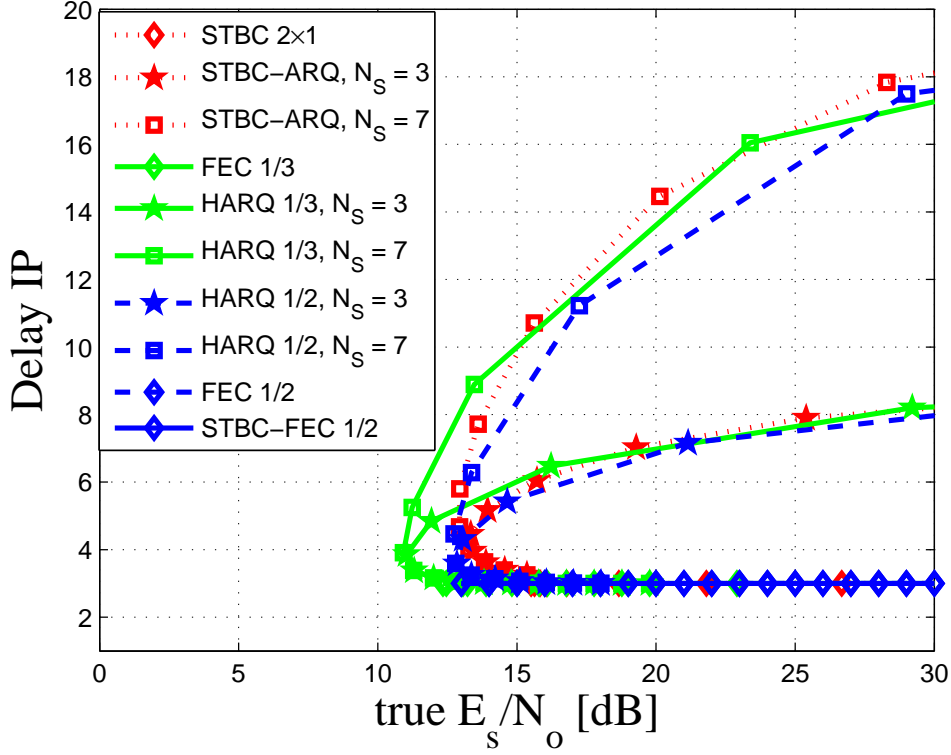


Figure 2.5: Network layer delay for the FEC, HARQ, STBC, STBC-ARQ systems, $N_S \{3, 7\}$, as function of the $\frac{\bar{E}_s}{N_0}$

efficiency point $\bar{\xi}$. The high SNR regime is again described by the bottom arm of the C.

Compare the STBC-ARQ scheme for $N_S = 7$ and the HARQ, $R_c = 1/2$ scheme, for $N_S = 3$. For values of target $\text{PER} < 10^{-2}$, attained for $\frac{\bar{E}_s}{N_0} > 13$ dB in the high SNR regime for both schemes (bottom arm of the C-shape in Figure 2.3) the values of the network-layer delay \bar{S} are comparable (bottom arm of the C-shape in Figure 2.5, $\frac{\bar{E}_s}{N_0} > 13$ dB). The design choice is then made in order to maximize energy efficiency, and the STBC-ARQ scheme is preferred. Now consider an application much more tolerant to packet loss than to delay (*e.g.* delivery of real-time video streams), with target $\text{PER} > 10^{-2}$. The operating regions in Figures 2.3 and 2.5 are identified as the upper arm of the C-shape, for $\frac{\bar{E}_s}{N_0} > 13$ dB. Inspection of Figure 2.5 reveals that, in this case, the HARQ scheme is preferable.

In this chapter we have considered propagation in fast-fading channels, and we have compared the QoS performance of different setups, in the FEC/HARQ and the STBC/STBC-ARQ families. The QoS metrics have been conventionally expressed as functions of the transmit E_s/N_0 per channel use, and as functions of the true E_s/N_0 , here introduced. This complementary analysis enables energetic profiling of the operating points of the setups, thus providing a valuable design tool, especially in case of energy-constrained applications. This tool can be applied for any network performance evaluations, *e.g.* for the cooperative

wireless networks.

Chapter 3

Deterministic Protocol Design and Analysis via FSM

3.1 Background

In this chapter we develop the system analysis tools for the cooperative relay networks, which are applied to the simplest network that consists of three nodes: a source, a relay and a destination. In other words, we build a framework and test its feasibility on the toy example in the most optimal and scalable way, in order to be able to apply this framework to larger schemes in the next chapter. We also use the metric derived in the previous section to assess the system performances in an energetic fair context.

In any networking system, the existence of a communication protocol is essential as it defines the set of rules by which the system behaves. For this reason, we first start this chapter by defining protocol for our scheme. As it will be shown later, there are many different protocols that could be deployed, but we have to limit our choice to those that are a) more realistic, b) efficient (in terms of the used resources and bandwidth), c) easily comprehensible.

In the literature there exist two major types in the protocol design: a) *deterministic*: *i.e.* as the name suggests, the actions taken by each node pre-defined, and take place in a time-division manner [10, 11, 20, 21]; and b) *probabilistic*: *i.e.* all the nodes of a system transmit data with certain probabilities during the same timeslot, which consequently leads to collisions, [49–51].

The *probabilistic* protocols, on the other hand, are closer to the realistic environments, e.g. for real-time wireless ad-hoc and mobile networks. However, their design has to deal with the collision probability [50] (unless orthogonal channels are used), since there is a possibility that several nodes transmit at the same time, causing a system collision at the receiver. In fact, the *deterministic* protocols can be viewed as a special case of the *probabilistic* ones, where the simultaneous retransmission probability is equal to zero for the transmitting nodes.

In this chapter we consider deterministic protocol design, and assume that all nodes in the network listen to the feedback messages sent by the receiving nodes. More details on this will be given in the protocol design section, 3.3.1. We design and examine several

types deterministic protocols, which differ based on the operation mode of the relay and the destination. As it will be shown, even for this simple configuration with one relay there will be many possible design choices, and this becomes even more complicated when the number of nodes (source or relay) is increased. In addition to that, the destination can also work in various modes: it can either combine or drop all the FRAGs that are received. Obviously combining the FRAGs is better for the error performance of the scheme, but needs a receive buffer to store all the copies. This, as a result, takes more computation time. The second technique does not spend much processing time and need not a receiving buffer, which in some systems can be very useful. In this chapter we will also compare the performances of the protocols with and without combining at the destination.

Furthermore, we are interested in making energetic fair comparisons between the schemes where there are no retransmissions involved in contradiction to the schemes with retransmissions, which is done by using the energetic fair criterion derived in the previous chapter.

As a rule, majority of the works that deal with performance evaluations of wireless networks, derive the QoS metrics using a combinatorial [20, 21] approach. However, even for a point-to-point communications this becomes quite complicated, and time-consuming, especially for the schemes with retransmissions. Albeit, there are some works that deduce the performance metrics via finite state machine (FSM) [49–51]. This approach becomes reasonable when the number of nodes in the network is larger than two. The authors in [51] model the Source-to-Relay, Source-to-Destination, and Relay-to-Destination channels via the FSM. As a matter of fact, they obtain a two-state Markovian process, based on which they compute the throughput, average delay and delay jitter.

The authors in [49] evaluate the tradeoff between the throughput (defined as the average number of frames successfully received in the destination node per time-slot) and the efficiency of the bandwidth utilization in a multi-relay network with undefined number of relays. The analysis is applied to two cases: in one the relays are the part of the infrastructure, and in the other - are not. The authors analyze and describe the N -hop ARQ wireless cooperative communication system via an FSM. Since the authors use multiple relay scenario, they define states as the number of relays that correctly received the FRAG. An additional $N + 1$ -th state is defined as the state, representing the successful frame decoding at the Destination. They use the transition probability matrix, based on which the steady state vector is computed. The latter is used in the performance metrics (*i.e.* the throughput, the average retransmission rate) computation. The authors are able to derive an optimal metric, which allows reducing the retransmission rate at the base station with a small reduction of the throughput in the single-destination case.

Another work, where the authors develop the system with the help of finite state machine is [50]. In this paper the strategy for the cooperating neighbors of a cooperative network, where collisions are possible, is optimized. The destination can correctly decode the data only if there is exactly one relay that both makes a retransmission and has a good channel (the channel has two states 'ON', which accounts for the collision; and 'OFF', which does not cause any interference). It is worth mentioning that in this paper the authors considered a probabilistic protocol, which basically means that all the nodes transmit with a certain probability.

3.1.1 Finite state machines

All the analysis of the systems is done via finite state machines. We show that it is possible to a) describe the protocol transmitting properties using the FSM, in further text referred to as **transmitting FSM**, and b) evaluate the performance metrics using a more detailed FSM, referred to as **performance analysis FSM**. It will be shown in this chapter, that this approach simplifies a lot the performance assessment, as the traditional (combinatorial) approach gets too complicated. However, the performance analysis FSM also can get complicated, which is due to the fact that we use ARQ (and will be explained in further sections). To overcome this difficulty, we then try to find an equivalent FSM, which will be more compact description, hence will be more manageable.

Recall that a finite state machine is a mathematical model of computation used to describe a system. Finite state systems can be modelled by *Mealy* machines that produce outputs on their state transitions after receiving inputs. In other words, finite state machine consists of states, inputs and outputs. The number of states is fixed; when an input is executed, the state is changed and an output is possibly produced. The machine can be in one state at a time. Finite state machines are widely used when designing computer programs, but also have their uses in engineering, biology, linguistics and other sciences thanks to their ability to recognize sequences.

Definition 7. *A finite state machine FSM is a quintuple*

$$FSM = (I, O, S, \delta, \lambda)$$

where I, O , and S are finite nonempty sets of input symbols, output symbols, states, respectively, [52].

$\delta : S \times I \rightarrow S$ is the state transition function

$\lambda : S \times I \rightarrow O$ is the output function

An FSM can be graphically represented by a state transition diagram: a directed graph whose vertices correspond to the states of the machine and whose edges correspond to the state transitions; each edge is labeled with the input and output associated with the transition. An FSM can also be illustrated via a state table with one row for each state and one column for each input symbol. For a combination of a present state and input symbol, the corresponding entry in the table specifies the next state and output.

FSM can be used to model a random system that changes states according to a transition rule that only depends on the current state and the output. Let us consider a random system (the simplest Markov chain example) and model it using FSM: the probabilities of weather conditions (modeled as either 'rainy' or 'sunny'), given the weather on the preceding day. The state of the FSM is defined by the weather condition, i.e. ' S_1 =sunny' or ' S_2 =rainy'. Let us consider the following transition probabilities:

- $\Pr(\text{it is rainy}|\text{it was sunny}) = \Pr(S_1 \rightarrow S_2) = 0.9$
- $\Pr(\text{it's sunny}|\text{it was rainy}) = \Pr(S_2 \rightarrow S_1) = 0.5$
- $\Pr(\text{it's sunny}|\text{it was sunny}) = \Pr(S_1 \rightarrow S_1) = 0.1$

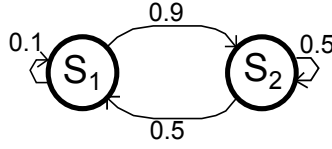


Figure 3.1: Finite State Machine Example for Markov chain process, describing weather conditions. The transitions between states are probabilistic.

- $\Pr(\text{it's rainy} | \text{it was rainy}) = \Pr(S_2 \rightarrow S_2) = 0.5$

The FSM, describing this random process is given in Figure 3.1.1, where the state transitions occur according to the current state of the machine, with a given probability.

As mentioned earlier, we use two different types of FSM in this work: a) one that describes the protocol; b) the second one that is used to derive the performance analysis. Below a description of each is given.

FSM for the protocol description, or the transmit FSM

This FSM serves to the aim of simply illustrating how the system reacts to a specific input. For this reason, the states of this machine are defined via the actions related to the transmitters: e.g. "source silent, relay silent". The transitions between the states are based on the feedback messages only, i.e. the ACK/NACKs.

FSM for the performance analysis, or the performance analysis FSM

This FSM serves to the aim of performance analysis of the considered schemes. In any scheme where retransmissions are present, the transmitting node(s) have to keep track of how many transmissions happened. After each transmission, the respective counter is decremented, thus updating the system status. Consequently, in order to pass from one state of the system to another, one would need to know: a) the last status of the counters; b) the feedback message(s). Therefore, it is sufficient to define the state of this FSM by the values of the counters at source(s) and at the relay(s). As a result, the transitions between the states depend on the feedback messages, and the actions are based on them. Our FSMs have countable finite number of states and thus correspond to Finite State Markov Chains (FSMC).

In the rest of this work, we refer to the **performance analysis FSM** as FSM or the **FSMC**, unless otherwise stated. Below is given a brief summary of how the performance metrics are obtained with the help of FSMC:

1. Derive the transition probability matrix

Definition 8. *The stochastic transition probability matrix (or Markov matrix) is a square matrix, whose elements represent the transition probabilities from one state of the FSM to another.*

In other words, these elements are the *one-step transition probabilities* of the Markov process. The stochastic matrix is a *finite* square matrix, whose order (number of rows) is equal to the number of states in the FSMC [53].

In our notation, the probability to go from state i to state j is represented by the element of the matrix, located at the i -th column and j -th row. Following the definition of the states that will be given for each protocol, the transition probabilities between states will be determined by the probability of observing control messages (ACK or NACK) sent by the decoders.

Consider the following simple example. The source can transmit a given FRAG, composed by L_{FRAG} information bits, at most 3 times at the destination. The destination attempts decoding after each transmission, and sends a control message (ACK or NACK) back to the source. As soon as the ACK message is observed or 3 transmissions have been performed, the system halts. The FSMC associated with this simple scheme has only 4 states: state S_i , $i \in \{1, 2, 3\}$ represent the ongoing transmission i , and state S_0 represents the halt condition. The system starts in state S_1 (first transmission). It is easy to see that the stochastic matrix takes the following form

$$P = \begin{pmatrix} 1 & 0 & 0 & 0 \\ (1 - \pi_1) & 0 & \pi_1 & 0 \\ (1 - \pi_2) & 0 & 0 & \pi_2 \\ 1 & 0 & 0 & 0 \end{pmatrix} \quad (3.1)$$

The first line in (3.1) represents the fact that once in the halt state S_0 the system does not evolve. The second line in (3.1) represents the fact that the system in state S_1 (first transmission) can reach either the halt state S_0 (after reception of ACK from the decoder), or move to the second transmission state S_2 (after reception of NACK). The probability of transitioning to S_2 is denoted by π_1 , and corresponds to the probability that the decoder was not able to recover the first received fragment. Assuming that the bit error rate on the considered channel is denoted by BER, the term π_1 is given by Eq. (2.4). The third line in (3.1) represents the fact that the system is in state S_2 (second transmission), and can reach either S_0 (reception of ACK) or S_3 (reception of NACK). The probability π_2 corresponds to the probability that the decoder was not able to recover the fragment, after 2 transmissions have been performed. If the system does not perform combining at the decoder, we have $\pi_2 = \pi_1$. Otherwise, if the decoder performs combining, we have that $\pi_2 \leq \pi_1$.

In order to evaluate these probabilities one could either use analytical approach, or use simulations. The analytical approach is feasible when the PHY layer BER can be evaluated in a closed form, with a precise equation. But when, for example, channel codes are used, the analytical expressions ([54]) in the form of an upper bound cannot be used to derive an exact expression.

For this reason, in order to evaluate the transition probabilities and insert them in the stochastic matrices, the BER at the physical layer are evaluated by simulations, both for no combining and combining. Then these BERs are converted to fragment error probabilities π_i (i.e. probability that one transmission fragment transmission fails) and inserted as appropriate in the stochastic transition matrix. Please remark, that the index i denotes

the number of copies of the FRAG at the decoder, and is used uniquely in the case when the destination performs combining.

Detailed explanation on how the physical layer BER is evaluated is provided in Appendix A.

2. Computation of the steady state vector The steady state vector is a vector of the size of the number of states of the FSMC. Each element represents the probability of being in each state of the FSMC, after an asymptotically long run of the machine. The steady state vector can be derived from the stochastic transition probability matrix as the eigenvector associated to the eigenvalue 1. One has

$$\begin{aligned} P \mathbf{p} &= \lambda \mathbf{p}, \quad \lambda = 1 \\ \sum_{i=1}^{N_{\text{states}}} p_i &= 1 \end{aligned} \tag{3.2}$$

where the term P represents the matrix, the term \mathbf{p} is the eigenvector and $\lambda = 1$ is the eigenvalue 1.

3. Evaluation of the performance metrics For each FSMC associated to a communication protocol, we evaluate the steady state vector. From the steady state vector we obtain the FER for example, by simply taking the probability to be in the state which leads to the FRAG drop event. Please notice that this computation strongly depends on the definition of the FSM states, and on the transition probabilities definition.

All the FSMs developed in this work are valid and generalized for any number of transmissions per node. However, the definition of the states changes slightly when the number of sources (i.e. nodes that transmit independent data to the destination), increases.

3.2 One Source, One Destination, no combining

Let us demonstrate how the FSM can be used to evaluate the performances of the very simple case, with one source and one destination. It will be shown in this section, that the performance evaluation metrics that were used in the previous chapter to evaluate the FER, and the average delay per successful and per FRAG (\bar{S} and \bar{T}), can be obtained also using the FSM approach. Let us first remind the protocol.

3.2.1 Protocol description

The nodes communicate using feedback channel. The source has the right to transmit the FRAG N times, after which the the current FRAG is dropped and a new one is transmitted. The feedback channel is assumed to be perfect, and sends a control message K , taking values in $\{\text{ACK}, \text{NACK}\}$. For the sake of simplicity, we detail only the case where the destination does not perform combining. Note that combining at the destination only affects the definition of the transition probabilities in the stochastic matrix, and hence

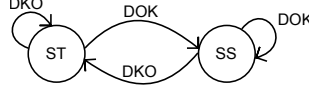


Figure 3.2: FSM describing the protocol behavior, 1s-1d. The notation "ST" and "SS" mean source transmits, and source is silent, respectively. The source status is either silent or transmits. The transitions from one state to another happen based on the acknowledgements of the system.

can easily be analyzed with the same tools as presented in this section. The channel is AWGN with fully-interleaved Rayleigh fading coefficients, and the modulation scheme is BPSK. The source transmits FRAGs of length L_{FRAG} information bits. The source has an incrementing counter of transmissions, A , taking values in $\{1, \dots, N\}$. The counter is set to one at the first transmission. Before performing a retransmission the source checks the control message K from the destination and the value of the counter A . If $K = \text{ACK}$ the FRAG has been correctly received, and it is not retransmitted. If $K = \text{NACK}$ and $A = N$ the maximum number of transmissions has already been reached, and the FRAG is dropped. The transmission of a new information FRAG then begins.

We are interested in evaluating the performance QoS metrics, which were defined in the first chapter of this work: the FER, the average delay per successful FRAG, and the average delay per FRAG. All the metrics in this part of the work are at the MAC layer.

3.2.2 System model

The system model is that given in Section (2.1) of Chapter (2).

3.2.3 Transmit FSM

The FSM that describes the transmission procedure of this protocol is given in Figure (3.2).

3.2.4 Performance evaluation with FSMC

The random variable K is defined on the space $\{\text{NACK}, \text{ACK}\}$, such that:

$$\begin{aligned} P(K = \text{NACK}) &= \pi \\ P(K = \text{ACK}) &= 1 - \pi. \end{aligned} \tag{3.3}$$

Term π is the probability that one FRAG transmission from the source to the destination fails, and is evaluated, similarly to the Chapter 2, Eq. (??), as $\pi = (1 - \text{BER})^{L_{FRAG}}$. Due to the channel memoryless property, the outcomes of the event $P(K = k)$ are independent from each other.

First we recall the derivation of the QoS metrics with the combinatorial approach, as performed in Chapter 2. The FER in such a protocol is defined as the ratio between the number of incorrectly decoded FRAGs and total number of transmitted FRAGs. It has

been evaluated in Chapter 2, Eq. (??), as:

$$FER = \pi^N$$

For the delay computations, we need the average number of FRAGs that are transmitted when a FRAG is correctly received, \bar{S} , and the average number of total transmitted FRAGs, denoted by \bar{T} . Using the results in Chapter 2, Eq. (??), these terms are easily deduced:

$$\bar{S} = \frac{\sum_{i=1}^N i \cdot (1 - \pi) \pi^{(i-1)}}{1 - \pi^N}$$

The average number of transmissions per failed FRAG can be computed as the sum of FRAGs transmitted per successful FRAG and number of FRAGs per failed FRAG. The first term in this sum is a product of \bar{S} and the probability of a successful FRAG. The second term is a product of total number of transmissions per FRAG and the probability that it fails. Then, the term \bar{T} can be obtained as:

$$\bar{T} = \bar{S}(1 - \pi^N) + N\pi^N$$

We now show how the performance can be evaluated using the FSMC.

3.2.4.1 Definition of the states

The FSMC is defined as a collection of distinct states, whose definition is chosen as explained in the following. Each state is associated with transition probabilities towards the other states. The stochastic matrix P is defined as the matrix of the transition probabilities, and is defined as $P_{ij} = P(\text{transition from } i \text{ to } j)$.

We define the state as a possible configuration of the pair (ongoing transmission number, last observed ACK/NACK) = (A_t, W_{t-1}) . Note that by construction of the transmission procedure a state with $1 < A_t \leq N$ can be visited only if the previous observed outcome was failure, i.e. $W_{t-1} = \text{NACK}$. As a consequence, states $(A_t = a, W_{t-1} = \text{ACK})$, for $a \in \{2, \dots, N\}$ do not exist.

The state $(A_t = 1, W_{t-1} = \text{ACK})$ represents the first trial of a new FRAG transmission, which starts after the previous FRAG has been successfully received. The state $(A_t = 1, W_{t-1} = \text{NACK})$ represents the first trial of a new FRAG transmission, which starts after the previous FRAG has not been successfully received. The total number of states is $N + 1$. Their numbering is done according to Table (3.1).

3.2.4.2 Transition probabilities evaluation

The transitions depend on the current state and on the realization of the outcome W_t (the result of the current transmission A_t). The probability of transition is associated to the probability of the realization of W_t .

State number	State definition
S_1	$W_{t-1} = \text{ACK}, A_t = 1$
S_2	$W_{t-1} = \text{NACK}, A_t = 2$
S_3	$W_{t-1} = \text{NACK}, A_t = 3$
\dots	\dots
S_N	$W_{t-1} = \text{NACK}, A_t = N$
S_{N+1}	$W_{t-1} = \text{NACK}, A_t = 1$

Table 3.1: Numbering of original FSM states for one source, one destination scheme

From state $(A_t = 1, W_{t-1} = *)$

States S_1 and S_{N+1} can transition to state S_1 (in case of successful trial) and to S_2 (in case of unsuccessful trial).

$$\begin{aligned} P_{1,1} &= P_{2,1} = P(W_t = 1) = (1 - \pi), \\ P_{1,2} &= P_{2,3} = P(W_t = 0) = \pi \end{aligned} \quad (3.4)$$

From state $(A_t = a, W_{t-1} = \text{NACK}), 1 < a < N$

Any state S_i ($A_t = a, W_{t-1} = \text{NACK}$) with $1 \leq a \leq N$ can transition to the state S_1 in case of successful outcome $W_t = \text{ACK}$, and to the state $i + 1$ ($A_t = a + 1, W_{t-1} = \text{NACK}$) in case of unsuccessful outcome.

$$\begin{aligned} P_{a,1} &= P(W_t = 1) = (1 - \pi), \\ P_{a,a+1} &= P(W_t = 0) = \pi \end{aligned} \quad (3.5)$$

From state $(A_t = N, W_{t-1} = \text{NACK})$

State S_N can transition either to S_1 (last transmission successful) or to S_{N+1} (last transmission unsuccessful).

$$\begin{aligned} P_{N,1} &= P(W_{t-1} = \text{ACK}) = (1 - \pi), \\ P_{N,N+1} &= P(W_{t-1} = \text{NACK}) = \pi \end{aligned} \quad (3.6)$$

The transition probability matrix takes hence the form:

$$P = \begin{pmatrix} 1 - \pi & \pi & 0 & 0 & 0 & \dots & 0 \\ 1 - \pi & 0 & \pi & 0 & 0 & \dots & 0 \\ 1 - \pi & 0 & 0 & \pi & 0 & \dots & 0 \\ 1 - \pi & 0 & 0 & 0 & \pi & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 1 - \pi & \pi & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (3.7)$$

3.2.4.3 Performance Evaluation

We use the steady state probability vector in order to evaluate the FER and the delay of the system. The steady state vector \mathbf{p} is of length $N + 1$ in this case, and its computation is similar to that described in the previous section.

Frame Error Rate In order to find the frame error rate, the requested states in the FSM are: the state where the system transmits a new FRAG after a success, and the state where the system transmits a new FRAG after a failure. The FER is the probability of a transmission of the new FRAG after the previous failed, and the averaged over the total number of successful and failed FRAGs. Denoting the steady state probability to be in state i by p_i , we obtain that the FER of this scenario is:

$$\text{FER} = \frac{p_{N+1}}{p_1 + p_{N+1}} \quad (3.8)$$

Average number of transmitted FRAGs per successful FRAG The average number of transmitted FRAGs per successful FRAG for this cooperative scenario is defined as the average number of transmitted FRAGs from the source and the relay, in order to successfully receive one FRAG:

$$\begin{aligned} \bar{S} &= \sum_{i=1}^N i \cdot P(i \text{ transmissions} | \text{FRAG is OK}) \\ &= \sum_{i=1}^N i \cdot \frac{P(\text{to be in state 1} | i\text{-th transmission})}{P(\text{to be in state 1})} \\ &= \sum_{i=2}^N i \cdot \frac{(1-\pi)p_i}{p_1} + (1-\pi) \frac{p_1 + p_{N+1}}{p_1} \end{aligned} \quad (3.9)$$

Average number of transmitted FRAGs per FRAG To determine the average number of transmissions per FRAG:

$$\bar{T} = \bar{S} \frac{p_1}{p_1 + p_{N+1}} + N \frac{p_{N+1}}{p_1 + p_{N+1}} \quad (3.10)$$

3.2.5 Simulation Validation

This section aimed at describing the theoretical derivations, which are shown to match the combinatorial approach, and hence, the simulations. In Figures (3.3) the FER at the MAC layer is given.

3.3 One Source, One Relay, One Destination Scheme, no combining

We consider three-node cooperative system with one source, one relay and one destination, as shown in Figure (3.4). The model can generalize a small wireless network with infrastructure, for example a Wireless Local Area Network (WLAN). The source and the destination communicate via a direct link, and via a relay node, which is situated between the source and the destination. The relay listens to the transmissions of source, and retransmits data in case if need arises.

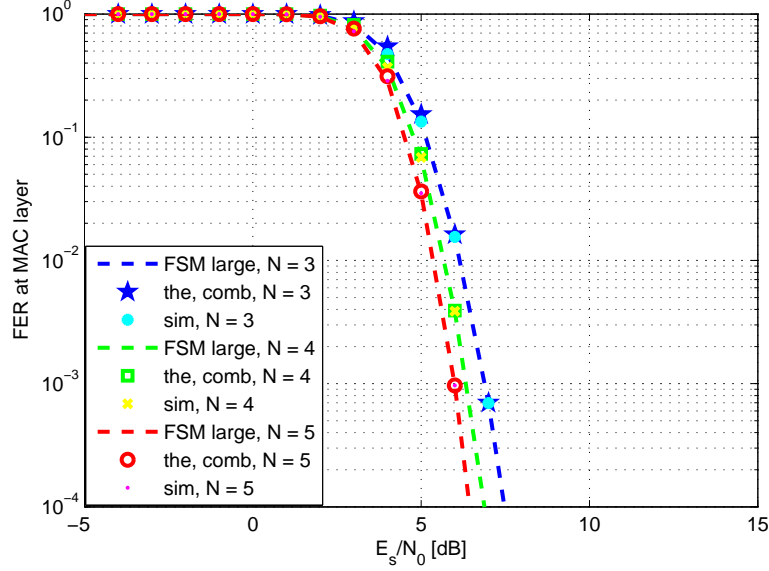


Figure 3.3: One source, one destination FER at the MAC layer in Rayleigh fully interleaved channel. Comparison of simulated results versus theory combinatorics and theory using FSM

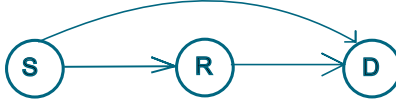


Figure 3.4: Cooperative relay network with one source, one relay and one destination

3.3.1 Protocol Description

This part of the work aims at describing the communications protocol and the associated assumptions. At the source, the fragment, containing information bits, is padded with a CRC. The resulting FRAG is encoded with a FEC of rate R_c , modulated, and transmitted from the source in broadcast mode.

Upon the reception of a FRAG, all the receiver nodes (i.e. the relay and the destination) send feedback messages: ACK/NACKs via a feedback channel. The feedback channel is assumed to be perfect and without any delays. We also consider that the CRC are also perfectly received. All the nodes of the network listen to the feedback messages, and as a result, all nodes know who has to transmit next (this is contained in the protocol definition).

Upon a reception of a negative acknowledgement from the destination, the transmitting nodes retransmit the data if they are authorized to. The relay has the priority to retransmit first, since it is assumed that it has shorter distance to the destination. The relay and the source have retransmission counters, which are decremented as soon as the node transmits. The retransmissions are assigned on a per-node basis, that is: the source has maximum

Relay Strategy	
Demodulate-and-FWD	Decode-and-FWD
1. Demodulate	1. Demodulate
2. Modulate	2. Decode (FEC)
3. Forward	3. Check CRC
	4. IF CRC is OK
	5. Add CRC
	6. Encode (FEC)
	7. Modulate
	8. Forward

Table 3.2: DCF and DMF protocols in a nutshell

N_S transmissions and the relay has maximum N_R transmissions. This gives in total $N = N_S + N_R$ transmissions per FRAG.

As before, we assume that the receivers do not perform combining.

Behavior of the Relay

In any communication system there can be multiple options to define the protocol, especially when there are multiple nodes and retransmissions involved. Nonetheless, there is a property: **the behavior of the relay**, which classifies the cooperative protocols into several types, among which we will consider two: 1) Decode-and-Forward and 2) Demodulate-and-Forward. Below the description of each protocol is given.

Decode-and-Forward (DCF)

The relay always decodes the FRAG and checks the CRC. As soon as the relay has successfully decoded a FRAG, it stores this version and does not drop until the end of the retransmission phase. The relay is authorized to retransmit the FRAG to the destination only if it has a correctly decoded copy, and has still retransmission credit. If the FRAG was not correctly received at the relay, or the relay has expired its retransmission credit, then the privilege to retransmit is granted to the source.

Demodulate-and-Forward (DMF)

The relay never decodes the received FRAG. It only demodulates, modulates and emits. In such a way, there can be a propagation of an erroneous FRAG, since there is no way the relay can check the CRC of what it received. The relay retransmits first, until its retransmission credit is exhausted, then the retransmission privilege passes to the source.

The summary of the sequence of relay actions is given in Table 3.3.1.

Behavior of the Protocol

In this section the set of protocol actions are described for each relay strategy (DCF or DMF) on a per timeslot basis.

Timeslot 1

the source transmits the data in a broadcast manner to relay and destination. After receiving the data, and depending on the relay operation mode, the respective node(s) checks the CRC. All the nodes in the network hear the control messages (ACK/NACK) and are able to determine from which node it comes (due to a pilot sequence). If after the first timeslot the destination has successfully received the data, then the source sends the next FRAG.

Timeslot 2

- A. if D correctly received the data, \Rightarrow the source generates a new FRAG and the same procedure repeats
- B. if D did not correctly receive the data, \Rightarrow then, depending on the behavior of the relay, retransmission occur either from the relay or the source. In the DCF case the relay transmits at most N_R times only if it has successfully received the FRAG. Otherwise, the source transmits until either D or R receive the data; or the credit of transmissions is expired. The FRAG is dropped at the destination if a) the number of transmissions is expired at both source and relay, b) if source expired the number of retransmissions, and the relay did not receive the data. In the DMF scheme the relay always transmits in the first turn, at most N_R times. Then, if it wastes all the credit and D still needs retransmissions, the source transmits at most N_S times. If the FRAG is not received at destination after the credits of retransmission are expired at all the nodes, the FRAG is dropped.

Furthermore, the destination can either perform or not combination of the FRAGs that it receives. That is, for each of the relay strategies (DCF, or DMF) there are two possible destination behaviors that are studied in this work.

3.3.2 System Model

This section aims at describing the channel model between the nodes in the network, and the received signal model. The symbols received at i -th position at the relay and destination are expressed below (for simplicity of notations, the index i is dropped):

$$\begin{aligned} y_{SD} &= h_{SD}x_m + n_{SD} \\ y_{SR} &= h_{SR}x_m + n_{SR} \\ y_{RD} &= h_{RD}\hat{x}_m + n_{RD} \end{aligned} \tag{3.11}$$

where $m \in [0, \dots, M]$ and M is the constellation size, h_{XY} and n_{XY} are the fading and noise coefficients from node X to node Y , respectively. We consider only fast fading

channels. For $M = 2$ PSK, the constellation has two symbols $X = [x_1, x_2]$, where the bits are symbols as follows:

- $x_1 = -\sqrt{E_b}$ for $c_i = 1$
- $x_2 = \sqrt{E_b}$ for $c_i = 0$

where the terms c_i denote i -th encoded bit in a FRAG, and $E_b = 1$ is the energy per symbol.

The complex noise coefficient consists of AWGN i.i.d. samples with $\mu_{noise,XY} = 0$ and $\sigma_{noise,XY}^2 = N_0/2$ per dimension. Rayleigh fully interleaved channels are assumed. The complex fading coefficients are i.i.d. normal variables with zero mean ($\mu_{fading,XY} = 0$) and unit variance ($\sigma_{fading,XY}^2 = 1$). Please note, that the data emitted from the relay (and denoted by \hat{x}_m), has a different notation, because it is an estimate of the original symbol that comes from the source. This means, that \hat{x}_m can be either correctly or wrongly decoded at the relay.

The data is encoded using rate $R_c = 1/2$ convolutional codes, with constraint length $K = 5$ and generator polynomials: $g_1 = [11111]$ and $g_2 = [11011]$. The demodulators both at the relay and at the destination output soft sequence, which is then fed to the soft input Viterbi decoder.

The physical layer BERs used to evaluate the terms π_{SD}, π_{RD} needed in the stochastic matrix are obtained by simulation, as detailed in Appendix A. In the case of the DCF protocol they are simply obtained as

$$\begin{aligned}\pi_{SD} &= 1 - (1 - \text{BER}_{SD})^{\text{L}_{\text{FRAG}}} \\ \pi_{RD} &= 1 - (1 - \text{BER}_{RD})^{\text{L}_{\text{FRAG}}}\end{aligned}$$

However, in the DMF protocol the source to relay transmission can result in error, which may propagate to the destination during the relay to destination transmission. To account for this fact, the transmission of a single FRAG from the source to the relay, and from the relay to the destination is simulated at the PHY layer. In other words, this term does not simply depend on the BER of the RD channel, but also on the error propagation of the SR transmission. Taking into account this, the state transitions are given below.

3.3.3 Performance Analysis using Combinatorial Approach

As it was mentioned in the previous sections of this work, combinatorial approach could also be used in order to derive the system performance metrics. This approach consists of listing all the events, associated with a FRAG drop event, and summing these events.

Demodulate and Forward (DMF)

We also obtained the theoretical expressions for the DMF protocol FER, average delay per successful FRAG, and average delay per FRAG using combinatorics. Although quite straight forward, this approach is not generic enough, and can be very time consuming when used for larger schemes. This served as a motivation to investigate the FSM approach.

We remind that the definitions to the metrics are given in Chapter 1, Eq. (1.7, 1.8, 1.9).

According to the definition of the FER, it is the product of probability that all transmissions of the source failed, and the probability that all transmissions of the relay failed, and it is given by:

$$FER = (\pi_{SD}^{N_S}) \cdot (\pi_{RD}^{N_R}) \quad (3.12)$$

The average delay per successful FRAG at the MAC layer is given by:

$$\bar{S}_{MAC} = \frac{1 - \pi_{SD} + \sum_{i=2}^{N_R+1} i \cdot (1 - \pi_{RD}) \pi_{SD} \pi_{RD}^{i-2} + \sum_{i=N_R+2}^{N_S+N_R} i \cdot (1 - \pi_{SD}) \pi_{SD}^{i-N_R-1} \pi_{RD}^{N_R}}{1 - FER} \quad (3.13)$$

The average delay per FRAG at the MAC layer is given by:

$$\bar{T}_{MAC} = \frac{\bar{S}_{MAC}}{1 - FER} + \frac{(\pi_{SD}^{N_S}) \cdot (\pi_{RD}^{N_R})}{FER} \quad (3.14)$$

By denoting the total number of transmitted coded bits from S and R, by \bar{n} , we notice that it consists of three different terms:

- \bar{n}_{SD} ,
- \bar{n}_{RD} ,
- and all the number of FRAGs per failed FRAG

Let us first remind the formula of the goodput here:

$$\zeta = \frac{\text{average number of successfully received information bits}}{\text{time unit}^{-1}} \quad (3.15)$$

It is assumed that transmission of one uncoded FRAG takes 1 second. Accordingly, the transmission of a FRAG coded with rate R channel codes takes $1/R$ seconds. Please notice that $0 < R \leq 1$.

$$\begin{aligned} \zeta &= \frac{(1 - FER) \cdot L}{\bar{n}_{SD} \cdot L + \bar{n}_{RD} \cdot L + (\frac{N_S}{R_{SD}} + \frac{N_R}{R_{RD}}) \pi_{SD}^{N_S} \pi_{RD}^{N_R} \cdot L} \\ &= \frac{(1 - FER)}{\bar{n}_{SD} + \bar{n}_{RD} + (\frac{N_S}{R_{SD}} + \frac{N_R}{R_{RD}}) \pi_{SD}^{N_S} \pi_{RD}^{N_R}} \end{aligned} \quad (3.16)$$

where L is the FRAG length, R_{RD} the channel encoder rate at the relay, and R_{SD} the channel encoder rate at the source. The number of transmitted bits per correctly received

FRAG as a result of RD transmission is given by:

$$\begin{aligned}
\bar{n}_{RD} &= \sum_{t=2}^{N_R+1} \frac{1}{R_{RD}} t \cdot \Pr\{\text{FRAG is OK in } t\text{-th tx} \mid \text{FRAG is OK}\} \\
&= \frac{1}{R_{RD}} \sum_{t=2}^{N_R+1} t \cdot \frac{\Pr\{\text{FRAG is OK} \mid \text{FRAG is OK in } t\text{-th tx}\} \Pr\{\text{FRAG is OK in } t\text{-th tx}\}}{\Pr\{\text{FRAG is OK}\}} \\
&= \frac{1}{R_{RD}} \sum_{t=2}^{N_R+1} t \cdot (1 - \pi_{RD}) \pi_{RD}^{t-2} \pi_{SD}
\end{aligned} \tag{3.17}$$

The number of transmitted bits per correctly received FRAG as a result of SD transmission is given by:

$$\begin{aligned}
\bar{n}_{SD} &= \sum_{k=N_R+2}^{N_R+N_S} \frac{1}{R_{SD}} k \cdot \Pr\{\text{FRAG is OK in } k\text{-th tx} \mid \text{FRAG is OK}\} \\
&= \frac{1}{R_{SD}} \sum_{k=N_R+2}^{N_R+N_S} k \cdot \frac{\Pr\{\text{FRAG is OK} \mid \text{FRAG is OK in } k\text{-th tx}\} \Pr\{\text{FRAG is OK in } k\text{-th tx}\}}{\Pr\{\text{FRAG is OK}\}} \\
&= \frac{1}{R_{SD}} \left(\sum_{k=N_R+2}^{N_R+N_S} k \cdot (1 - \pi_{SD}) \pi_{SD}^{k-N_R-2} \pi_{SD} \pi_{RD}^{N_R} + 1 - \pi_{SD} \right)
\end{aligned} \tag{3.18}$$

By using all these formulas, the final expression of the term \bar{n} becomes:

$$\begin{aligned}
\bar{n} &= \frac{1}{R_{SD}} \left(\sum_{k=N_R+2}^{N_R+N_S} k \cdot (1 - \pi_{SD}) \pi_{SD}^{k-N_R-2} \pi_{SD} \pi_{RD}^{N_R} + 1 - \pi_{SD} \right) \\
&\quad + \frac{1}{R_{RD}} \sum_{t=2}^{N_R+1} t \cdot (1 - \pi_{RD}) \pi_{RD}^{t-2} \pi_{SD} \\
&\quad + \left(\frac{N_S}{R_{SD}} + \frac{N_R}{R_{RD}} \right) \pi_{SD}^{N_S} \pi_{RD}^{N_R}
\end{aligned} \tag{3.19}$$

Finally, by inserting Equation (3.19) into Equation (3.16), we obtain the goodput G .

Decode and Forward (DCF)

In this section the theoretical performances of the DCF protocol are derived using the combinatorial approach. As it will be shown, the analysis becomes more complicated even for this simple scenario, as there are many events to be considered. This demonstration is done using the FER. The metric definition is that declared in Chapter (1) Eq. (1.7).

For example, let us consider the DCF protocol FER derivation. Assume that the PHY layer error probabilities for any link between X and Y are given with the term π_{XY} . This term can be evaluated either analytically, as we did for example in Eq. 2.4 in the first

chapter of this thesis; or by simulations if there is no closed-form expression. The maximal number of transmissions per node is denoted by k , i.e. $N_S = N_R = N$. It is assumed that the destination performs no combining, so the terms π_{XY} remain the same after each retransmission.

Now let us list all the events that are associated with a FRAG drop event, when we use DCF protocol:

$$\begin{aligned} \text{FER} &= (\pi_{SR}\pi_{SD})^N + \sum_{i=1}^N (\pi_{SR}\pi_{SD})^{(i-1)}(1 - \pi_{SR})\pi_{SD}^{N-(i-1)}\pi_{RD}^N \\ &= (\pi_{SR}\pi_{SD})^N + \sum_{i=1}^N \pi_{SR}^{(i-1)}(1 - \pi_{SR})\pi_{SD}^N\pi_{RD}^N \end{aligned} \quad (3.20)$$

The same strategy can, in principle, be employed also for the evaluation of the delay \bar{S} and of the average number of transmissions per fragment \bar{T} .

At the first sight, this approach seems quite straightforward and simple. However, let us assume that we also perform combining at the destination. In this case, the probability of error after each transmission is smaller than that of the previous transmission. This means, that we would have an expression, containing a big sum with multiple elements, each of them being a different probability.

Furthermore, in this simple scenario we have only one source and one relay. If we add one node, the expression becomes more complex (which will be demonstrated for the two source, one relay scenario as well).

3.4 Analysis of the One Source, One Relay, One Destination, no combining Scheme using FSMs

The example outlined in the previous section and evaluated using a combinatorial method is now taken as an example to evaluate the pros and cons of the models using FSMs, an analysis which will ultimately result in the proposal of specific probabilistic protocols with nice properties (Chapter 5).

3.4.1 Transmit FSM, DCF relay mode

The protocol used for transmitting the data can be modeled via a finite state machine, where each state represents the action to be undertaken by each node in the protocol, and the transitions are labeled with inputs (i.e. positive or negative acknowledgements). The FSM for an infinite number of source FRAGs is illustrated in Figure 3.5.

The transmit FSM described in this Section is the functional description of the protocol, and serves the purpose of illustrating its different actions. We suppose here the transmission of a single information fragment from the source to the destination node. We introduce also the following variation with respect to the protocol described in Section 3.3.1: while the relay has a finite retransmission credit, the source can retransmit until the fragment is correctly acknowledged by the destination.

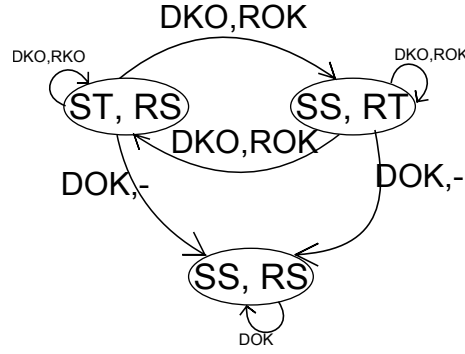


Figure 3.5: Transmit FSM for one source, one relay, and one destination scheme. There are three states; 1) source transmits and relay is silent, 2) source is silent and the relay transmits, 3) both source and relay are silent. In order to transit from state 1 to state 2, the inputs have to be "destination is ACK-ed and relay is NACK-ed".

We can identify three distinct operational phases. In the first phase, represented by state 1, the source is transmitting, and the relay is silent. This can happen either in time-slot 1, when the new fragment is broadcasted for the first time; or in subsequent time-slots, when the source is performing retransmissions and the relay is silent. At the beginning of the protocol, the source retransmits until the relay manages to correctly decode the broadcasted fragment. This event is signaled by the input ROK, that entails transition to the second phase of the protocol, represented by state 2. In state 2 the relay is in charge of retransmissions, and the source is silent. The system stays in this state as long as the relay has retransmission credit left, and the fragment is not acknowledged by the destination. When the relay retransmission credit expires (event signaled by the input RKO), the system transitions back to state 1, where the source retransmits and the relay is silent, until the fragment is correctly acknowledged by the destination. State 3 is the end state of the system, representing correct reception of the information fragment. It can be reached both from states 1 and 2, as soon as the ACK control message is observed from the destination (corresponding to input DOK).

3.4.2 Performance evaluation with FSMC

In this section we define the FSMC that helps in the performance metrics derivations. As we could see from the Section 3.3.1, the protocol can be described by a Markov process where each transition to the state at time instance t depends only on the state at time $t - 1$. In other words, the decision whether or not to retransmit and from which node, only depends on what happened in the previous timeslot: that is a) what were the error control messages from the relay and destination, b) what is the status of the relay (applicable only for the DCF protocol).

As we could see from the protocols description (Section 3.3.1), there is certain information that needs to be known in each timeslot in order to give the system description:

1. how many retransmissions have happened from the source and the relay in the past,

2. furthermore, in the case of the DCF protocol it is necessary to note the the previous status of the relay.

The finite state machine for DCF and DMF protocols is different because the states definitions are different too. The FSM takes inputs and outputs from the sets given below (for both protocols):

$$\begin{aligned} I &\in \{[DOK, ROK], [DOK, RKO], [DKO, ROK], [DKO, RKO]\} \\ O &\in \{[SS, RS], [ST, RS], [SS, RT]\} \end{aligned} \quad (3.21)$$

The notation **XOK** means that the node **X** successfully received the FRAG and sent an ACK, whereas **XKO**: that it sent a NACK. The notions **XS** and **XT** define: "node **X** is silent" and "node **X** transmits, respectively.

3.4.3 Performance evaluation with FSMC, DMF protocol

We begin deriving the FSMC for the DMF, no combining at the decoders protocol. The strategy is similar to the one adopted for the one source, one destination scheme.

Definition of the states

We consider that there is one state where the FRAG was correctly decoded at destination and the source transmits a new FRAG; and another state which is associated to the dropping of the FRAG and transmitting a new one. We define the state of the original FSM with the following variables:

- the status of the destination in the previous transmission, W_{t-1}
- the number of transmissions that happens in the current state A_t

The original FSM has $N_S + N_R + 1$ states, where the first state is associated to the first transmission of the FRAG after a successful decoding of the previous FRAG; and the last, $N_S + N_R + 1$ -th state is associated to the first transmission of the FRAG after the previous FRAG failed. The states definitions are given in Table 5.3.

Transition probabilities evaluation

The transition probabilities between the states of the FSM are evaluated using the probability that one FRAG transmission at a given channel fails. We again recall that the latter is denoted by π_{XY} for the transmission of a FRAG at channel XY . As it was already mentioned in the Section (3.3.2) of this chapter, in the DMF protocol the term π_{RD} also takes into account the transmission at the SR link, which is then demodulated at the relay, and then sent to the destination. The destination decoder takes into account the fact that the SR transmission could have been erroneous, and decodes accordingly. The terms π_{XY} are evaluated using simulations.

The state transitions depend on the current transmission result ($W_t = ACK$ or $W_t = NACK$), associated to the current transmission A_t . As a reminder, we refer to the

State number	State definition
1	$W_{t-1} = \text{ACK}, A_t = 1$
2	$W_{t-1} = \text{NACK}, A_t = 2$
3	$W_{t-1} = \text{NACK}, A_t = 3$
4	$W_{t-1} = \text{NACK}, A_t = 4$
...	...
$N_R + 1$	$W_{t-1} = \text{NACK}, A_t = N_R + 1$
$N_R + 2$	$W_{t-1} = \text{NACK}, A_t = N_R + 2$
$N_R + 3$	$W_{t-1} = \text{NACK}, A_t = N_R + 3$
...	...
$N_R + N_S$	$W_{t-1} = \text{NACK}, A_t = N_S + N_R$
$N_R + N_S + 1$	$W_{t-1} = \text{NACK}, A_t = 1$

Table 3.3: DMF FSM Original States

probability of one FRAG transmission failure at XY channel as π_{XY} . The elements in the transition probability matrix P are evaluated as follows.

From state S_1 The state S_1 can transition to states S_1 and S_2 in the case of successful and unsuccessful transmission, respectively.

$$\begin{aligned} P_{1,1} &= 1 - \pi_{SD}, \\ P_{1,2} &= \pi_{SD}, \end{aligned} \tag{3.22}$$

From states $S_i, i \in \{2, \dots, N_R + 1\}$

$$\begin{aligned} P_{i,1} &= 1 - \pi_{RD}, \\ P_{i,i+1} &= \pi_{RD}, \end{aligned} \tag{3.23}$$

From states $S_i, i \in \{N_R + 2, \dots, N_S + N_R\}$

$$\begin{aligned} P_{i,1} &= 1 - \pi_{SD}, \\ P_{i,i+1} &= \pi_{SD}, \end{aligned} \tag{3.24}$$

From state $S_{N_S+N_R+1}$

$$\begin{aligned} P_{N_S+N_R,1} &= 1 - \pi_{SD}, \\ P_{N_S+N_R,2} &= \pi_{SD}, \end{aligned} \tag{3.25}$$

The transition probability matrix of the FSM for DMF protocol is hence

$$P = \begin{pmatrix} 1 - \pi_{SD} & \pi_{SD} & 0 & \dots & 0 & 0 \\ 1 - \pi_{RD} & 0 & \pi_{RD} & 0 & 0 & 0 \\ 1 - \pi_{RD} & 0 & 0 & \pi_{RD} & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 1 - \pi_{SD} & 0 & 0 & 0 & 0 & 0 \\ 1 - \pi_{SD} & \pi_{SD} & 0 & 0 & 0 & 0 \end{pmatrix} \quad (3.26)$$

Performance Evaluations

In this section we describe how to evaluate the performance metrics using the transition probability matrix. Let \mathbf{p} denote the steady state vector.

We again remind our reader that the definitions to the metrics that are derived in this section are given in Eq. (1.7, 1.8, 1.9).

The Frame Error Rate The states that are needed to compute the FER, are the state where the system transmits a new FRAG after the previous failed, and the state where the system transmits a new FRAG after the previous was successfully received. The FER is the probability that the FRAG fails, averaged over the sum of probabilities that the FRAG was not successful and was successful. In other words:

$$FER = \frac{p_1}{p_1 + p_{N_S + N_R + 1}} \quad (3.27)$$

where the terms p_i denote the steady state probability to be in state i after a long run of the FSM; N_R and N_S the total number of transmissions granted to the relay and to the source, respectively.

Average number of transmissions to successfully receive one FRAG, \bar{S} This is the formula using the original FSM steady state probabilities:

$$\begin{aligned} \bar{S}_{MAC} &= \sum_{t=1}^{N_R + N_S + 1} \Pr\{t \text{ transmissions happened} \mid \text{DOK}\} \Pr\{t \text{ transmissions happened}\} \\ &= \sum_{t=1}^{N_R + N_S + 1} \frac{\Pr\{t \text{ transmissions happened, DOK}\} \Pr\{t \text{ transmissions happened}\}}{\Pr\{\text{DOK}\}} \\ &= \frac{\sum_{t=2}^{N_R + 1} t \cdot (1 - \pi_{RD}) p_t + \sum_{t=N_R + 2}^{N_R + N_S} t \cdot (1 - \pi_{SD}) p_t + (1 - \pi_{SD})(p_1 + p_{N_S + N_R + 1})}{p_1} \end{aligned} \quad (3.28)$$

where the terms p_i denote the steady state probability to be in state i after a long run of the FSM; π_{XY} are the elementary FRAG error probabilities on the X to Y links; N_R and N_S the total number of transmissions granted to the relay and to the source, respectively.

Average number of transmissions per FRAG The average of the total number of transmitted FRAGs is computed as the sum of FRAGs transmitted per successfully received FRAG and the number of FRAGs transmitted when the FRAG fails. The first term of this sum consists of the product of the term \bar{S}_{MAC} and the probability that the FRAG was OK. The second term consists of the product between the total number of FRAGs transmitted when the FRAG fails, and the probability that the FRAG fails.

This is the formula using the original FSM steady state probabilities:

$$\bar{T}_{\text{MAC}} = \bar{S} \frac{p_1}{p_1 + p_{N_S+N_R+1}} + (N_S + N_R + 1) \frac{p_{N_S+N_R+1}}{p_1 + p_{N_S+N_R+1}} \quad (3.29)$$

Goodput In order to derive the goodput using the target FSM, let us refer to its definition in Equation (3.15). Given that the steady state probability to be in state i , is denoted by p_i , it can be easily deduced:

$$\zeta = \frac{\frac{p_1}{p_1 + p_{N_S+N_R+1}}}{\frac{\sum_{t=2}^{N_R+1} t \cdot p_t \cdot (1 - \pi_{RD})}{R_{RD}} + \frac{\sum_{k=N_R+2}^{N_R+N_S+1} k \cdot p_k \cdot (1 - \pi_{SD}) + (1 - \pi_{SD})}{R_{SD}} + \left(\frac{N_S}{R_{SD}} + \frac{N_R}{R_{RD}}\right) p_{N_S+N_R+1}} \quad (3.30)$$

3.4.4 Performance evaluation with FSMC, DCF protocol

We follow a slightly different strategy for the evaluation of the FSMC associated with the DCF protocol.

Definition of the states

As mentioned earlier, for the FSM that helps in analysis of the QoS metrics, we need to know the status of the counters, and depending on the protocol:

- *DCF - S* : $\{C_S, C_R, R_{\text{status}}\}$. We need to mention the last status of the relay in the state definition of the finite state machine because the relay drops erroneous fragments and keeps it only when correctly received. In order to be coherent with this assumption, this information is also needed.

The finite state machine for the DCF protocol is given in Figure 3.6. The transitions between the states of the FSM are labeled with inputs and outputs. The inputs are the control messages of the system (*i.e.* the ACK/NACKs received at the destination and the relay). The outputs are the actions taken based on the input.

The following notations are used in the figures:

- *DKO, RKO* - destination sent a NACK, relay sent NACK
- *DKO, ROK* - destination sent a NACK, relay sent ACK
- *DOK, -* - destination sent a ACK, consequently the relay's feedback does not matter. With this input the system has to proceed the next FRAG.

Since the number of states depends on the retransmissions credit, leading to a very large FSM when this number increases, and in order to simplify the illustration of the generalized FSMs, we graphically represent the equivalent states (i.e. where the system is in a loop and ends up in the similar state after each iteration of this loop) into one state. Please notice, that, however, this does not reduce the number of states, but only simplifies the FSM illustration, Figure 3.6.

There are two types of transitions in the FSM, marked with straight arrows, and round arrows. In the case when the finite state machine loops in the current state, with the output state being unchanged, the round arrows are used.

Each time a retransmission happens, the machine transits from one state to another, producing an output. Consider 3.6. Let us assume that in the first timeslot the relay and the destination did not receive the fragment. The source transmitted twice, so the system lands in set of states 6, Figure 3.1.1.

During the last transmission from the source, the relay receives the FRAG. Since the relay has never transmitted previously, it will be granted a retransmission, i.e.: the input is **"DKO, ROK"**, and the corresponding action **"Source Silent, Relay Transmits"**. As a result, the system will land in another state, which belongs to the set of states 7.

Based on the representation of the finite state machines, we develop the algorithm to automatically generate the FSM and to compute the stochastic transition probability matrix, for any number of transmissions. In this chapter we detail the algorithm for the DCF protocol, which is more complicated and has more challenges in its development. The FSM and stochastic matrix generation algorithms for the DMF scheme can be easily derived in a similar manner, thus are not further detailed.

On the number of states in the FSMC

A natural question that arises, is "how will the number of states in the performance evaluation FSM increase when we increase the number of transmissions per node?"

By noticing the connection between the number of retransmissions granted per node, and the number of states in the finite state machine, it becomes obvious, that this approach would become too complicated for analysis when the number of transmissions increases. For example, let us have a look at the proportion between N_S, N_R and the DCF protocol states number growth:

1. $N_S = 2, N_R = 1 \Rightarrow N_{\text{states}_{\text{total}}} = 4$
2. $N_S = 3, N_R = 2 \Rightarrow N_{\text{states}_{\text{total}}} = 9$
3. $N_S = 4, N_R = 3 \Rightarrow N_{\text{states}_{\text{total}}} = 16$
4. $N_S = 5, N_R = 4 \Rightarrow N_{\text{states}_{\text{total}}} = 25$
5. ...
6. $N_S = 11, N_R = 10 \Rightarrow N_{\text{states}_{\text{total}}} = 121$
7. ...

$$8. N_S = 21, N_R = 20 \Rightarrow N_{\text{states}_{\text{total}}} = 400$$

It can thus be summarized, that:

1. by increasing the number of allowed retransmissions from 1 to 20, the number of states in the FSM for DCF increases 100 times
2. by denoting the number of retransmissions by μ , it can easily be deduced that:

$$\lim_{\mu \rightarrow \infty} (\mu + 1)^2 = \infty \quad (3.31)$$

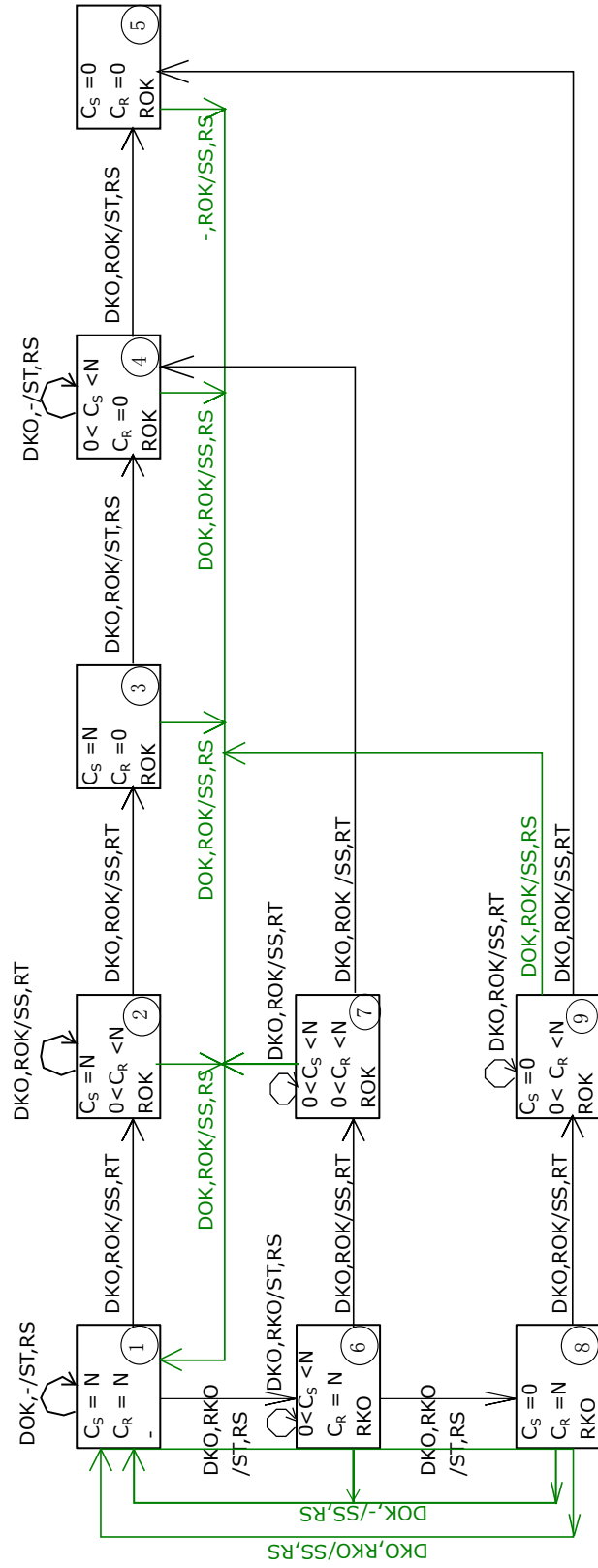


Figure 3.6: DCF protocol FSM of the one source, one relay, and one destination scheme

Transition probabilities evaluation

The stochastic transition probability matrix is given in a general form for the DCF protocol Table 3.4, for $N_R > 1, N_R \neq N_S$. The terms π_i (Table 3.4) and $\pi_{i,XY}$ (Table ??) in the stochastic matrices represent the probability of that there is at least one bit error in the i -th FRAG, after it has been transmitted from node X to node Y . In case if the destination performs combining of FRAGs, these probabilities are different (i.e. probability of error decreases as the number of copies is increased).

In order to evaluate these probabilities one could either use analytical approach, or using simulations. The analytical approach is feasible when the PHY layer BER can be evaluated in a closed form, with a precise equation. But when, for example, channel codes are used, the analytical expressions ([54]) in the form of an upper bound cannot be used to derive an exact expression.

For this reason, in order to evaluate the transition probabilities and insert them in the stochastic matrices, the BER at the physical layer are evaluated by simulations. Then these BERs are converted to fragment error probability (i.e. probability that one transmission fragment transmission fails) and inserted in the stochastic transition matrix. We recall the expression below:

$$\pi_i = 1 - (1 - \text{BER})^{L_{\text{FRAG}}}, \quad (3.32)$$

with $i \in [1, \dots, N]$ and the term BER is evaluated by simulations. Please remark, that the index i denotes which transmission is this, and is used uniquely in the case when the destination performs combining (i.e. the probability to decode the FRAG incorrectly is smaller in each following transmission). When no combining is performed at the destination, these terms are all equivalent for each transmission.

Table 3.4: Generalized Stochastic Transition Probability Matrix for 1 Source, 1 Relay, 1 Destination Scheme, DCF Protocol

	1	2	3	...	$N + 1$...	$2N + 1$...	$N_{\text{StatesTotal}}$
1	$1 - \pi_1$	$1 - \pi_2$	$1 - P_3$...	$1 - \pi_{N+1}$...	1	...	1
2	$\pi_1(1 - \pi_1)$	0	0	...	0	...	0	...	0
3	0	π_2	0	...	0	...	0	...	0
...
$N_{\text{StatesTotal}}$	0	0	0	...	0	...	0	...	0

Algorithmic Generation of the FSM and Stochastic Matrix

For a given $N_S \neq N_R$ the FSM and the stochastic matrix can be easily obtained using the algorithm that is described below.

```

for  $i = 1 \rightarrow N_S$  do
     $i = i + 1$ 
     $N_{\text{States}_{\text{Total}}} = N_{\text{States}_{\text{Total}}} + N_R + (N_S - i)$ 
end for

```

After having the number of states, they have to be enumerated systematically (from left to right):

```

statenumber = 1;
for  $i = 1 \rightarrow N_S$  do
    for  $j = 1 \rightarrow N_R + (N_S - i)$  do
        FSM(ii,jj) = statenumber;
        statenumber = statenumber + 1;
    end for
end for

```

Finally, below we give the summary of steps explaining how to fill in the transition probability matrix for any N_S and N_R , by keeping in mind that:

1. Starting from the second row of the FSM, each first state of FSM's each row is achieved when DKO,RKO: in other words, both D and R did not successfully receive the data.
2. Then, the transition from the first to the second state in each row of the FSM is achieved when DKO,ROK: in other words only R received successfully the data.
3. Then, the consequent transitions from the second state of each row to the next respective state are achieved by the event DKO. (No need to mention the relay again since it is already OK).
4. The transitions from the last state of each row to the state 1 are achieved by the event DOK.
5. The transition from the state that is located in the last row's first position to state 1 is achieved when DKO,RKO or DOK: in other words either the D receives the data, or both D and R fail.
6. The transition probability from any other state (that does not belong to the previous two cases) to state 1 is always equal to 1. Respectively the transition probabilities from this state to any other state (or their sum) have to be equal to 0.

Moreover, the transition probability from state i to state j , $P(i \rightarrow j)$ is given by the stochastic transition matrix element that is located in the j -th column of i -th row.

Performance Evaluations

The derivation of the frame error rate using the steady state probabilities, is explained below:

Frame Error Rate The Frame Error Rate (FER), is defined as the average number of dropped FRAGs divided by the total number of transmitted FRAGs. In order to compute the FER with the help of the steady-state probability vector, the following has to be done:

1. multiply the steady probabilities to be in the last state of each row of the FSM by the transition probability of going to state 1 with an error event at D
2. multiply the steady state probability to be in state that is located at the $N_R + 1$ -th row's 1st column of the FSM, by the transition probability to state 1 with an error event both at D and R
3. sum the terms obtained in steps 1 and 2.

The algorithm of the FER computation is given below:

```

for  $i = 1 \rightarrow N_S - 1$  do
    FER = FER + Pr{to be in last state of  $i$ -th row}  $\times P$ 
     $i = i + 1$ 
end for
FER = FER + Pr{to be in the 1st state of the  $N_S$ -th row}  $P^2$ 

```

3.5 Numerical Results

In this section the Monte-Carlo simulated results are discussed. Please notice that the analytical results for the FER obtained in Section 3.4.2 are validated by the simulations.

First we demonstrate with the FER example in Figure (3.7) that the theoretical evaluations using the FSM analysis match the simulations. In further we will plot only one curve, as both theory and simulations are the same.

Starting from this section and in all the further sections of this thesis, we use the following color schemes in the graphical representations: 1) DCF_{1s1r1d} red; 2) DMF_{1s1r1d} blue; 3) DCF_{2s1r1d} green; 4) DMF_{2s1r1d} black.

In order to show what are the main improvements and changes brought by the HARQ schemes, we also simulate the schemes without HARQ: a) 1 FRAG transmission scheme, equivalent to sending a FEC coded FRAG in a point-to-point network; b) 2 FRAGs transmission, where one transmission happens from the source and one from the relay: the destination combines these two versions and feeds LLR to the decoder.

We then discuss the cooperative HARQ for the defined DCF and DMF protocols.

In Table 3.5 we give the simulation configurations that apply for all the schemes.

Figure 3.8 illustrates the comparison of the schemes: a) with 1 FRAG (no combining, no ARQ), and b) 2 FRAG (with combining, no ARQ). As one could see, the scheme with combining at the destination outperforms the single FRAG transmission with FEC, by offering a diversity-type 6 dB gain.

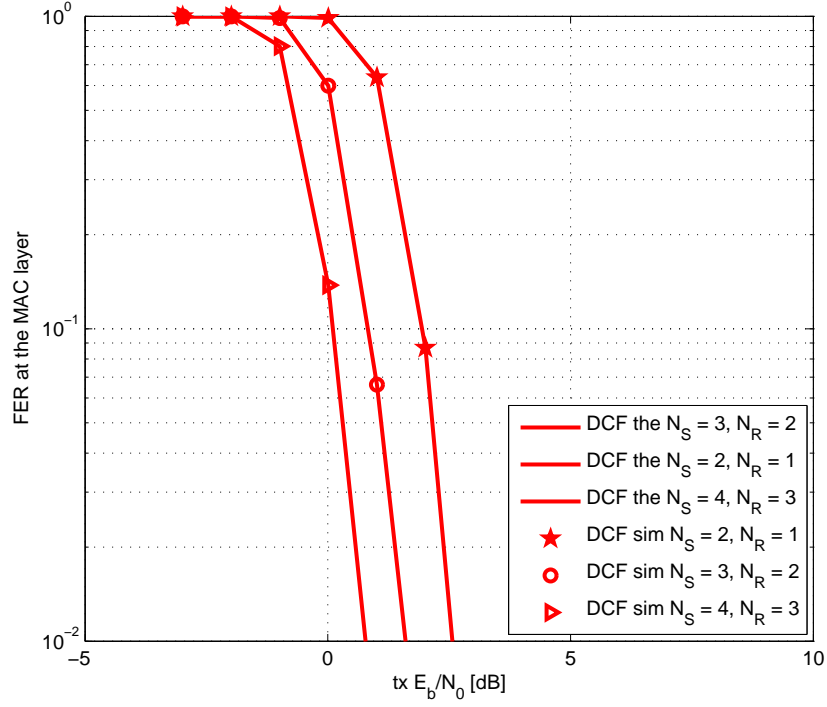


Figure 3.7: Comparison of theoretical and simulated results of DCF protocol FER, 1s-1r-1d

Table 3.5: Simulation configurations, 1-s-1-r-1-d scheme

<i>Parameter</i>	<i>Value</i>
FRAG length [information bits]	$L_F = 120$
Channel code rate	$R_c = \frac{1}{2}$
encoded FRAG length [coded bits]	$L_C = \frac{L_F}{R_c} = 360$
Constraint length	$K = 5$
Trellis structure	$g_1 = [11111], g_2 = [11011]$
Number of retransmissions per node	$\mu \in 0, 1, 2$

In order to see what happens when the retransmissions are added on top, we plot the results of the schemes without ARQ along with the DCF protocol, Figure 3.15.

3.5.1 Combining versus no combining

In this section the frame error rate of the DCF protocol is compared for two different cases: a) destination performs combining; b) the destination does not perform combining, Figure 3.9. These comparisons are done both as a function of transmitted E_s/N_0 , Figure 3.9; and plotted versus the **true** E_s/N_0 , Figure 3.10.

As it can be observed in Figure 3.9, the FER is plotted for $N_S = 2, N_R = 1$ (star markers) and $N_S = 3, N_R = 2$ (o markers). The group of curves with solid lines represents the no-combining at D case; and the dotted line curves are the ones with combining at the destination. When we compare the no-combining and combining curves for say $N_S = 3, N_R = 2$, we can see that there is a large improvement of around 6 dB, brought by combining.

Similar behavior is observed, when we compare these two techniques in an energetic fair context, Figure 3.10. In other words, combining at the destination improves the FER significantly both when we consider the transmitted signal-to-noise ratio, and when the total energy per transmitted bit is considered. Thus, the tradeoff lies between the receiver buffer implementation and the FER improvement of around 6 dB.

Furthermore, we compare these schemes to the simple transmission of a FEC-encoded FRAG in a point-to-point network, and to the 2 FRAGs combining (i.e. when destination receives two copies of the same FRAG, encoded with FEC) as a function of transmitted

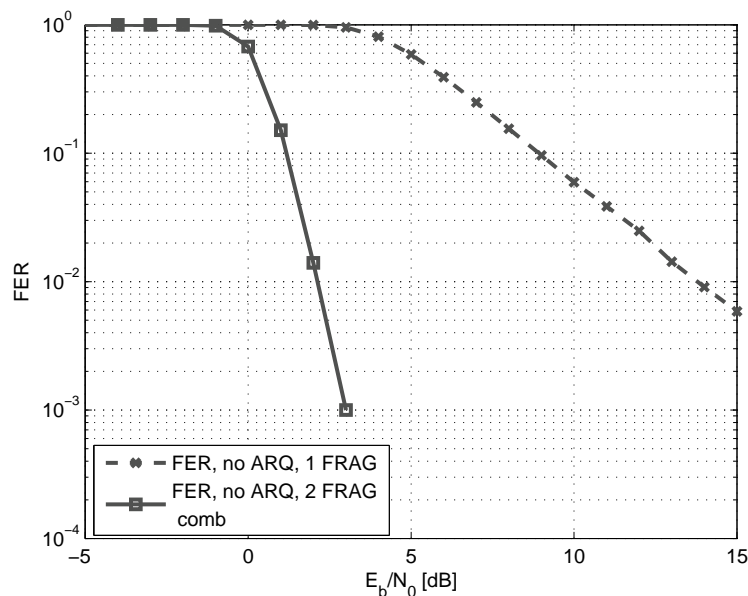


Figure 3.8: Frame Error Rate of scheme with one and two FRAG transmission in Rayleigh channel. FEC of rate $\frac{1}{2}$

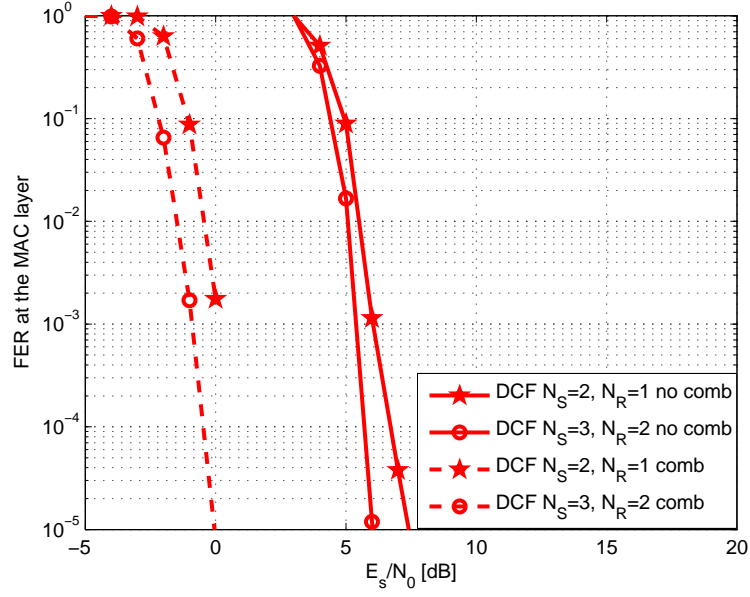


Figure 3.9: DCF protocol FER at the MAC layer as a function of E_s/N_0 in Rayleigh channel, for one source, one relay, one destination scheme

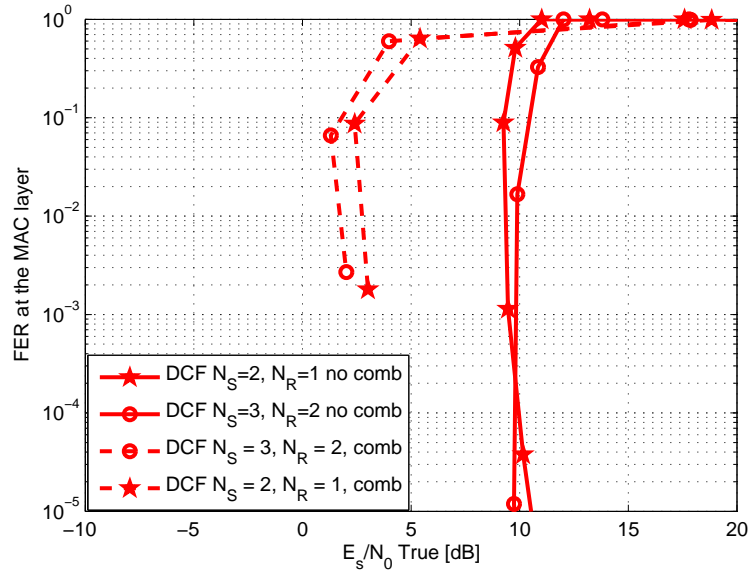


Figure 3.10: DCF protocol FER with and without combining at D; at the MAC layer plotted versus of **true** E_s/N_0 in Rayleigh channel, for one source, one relay, one destination scheme

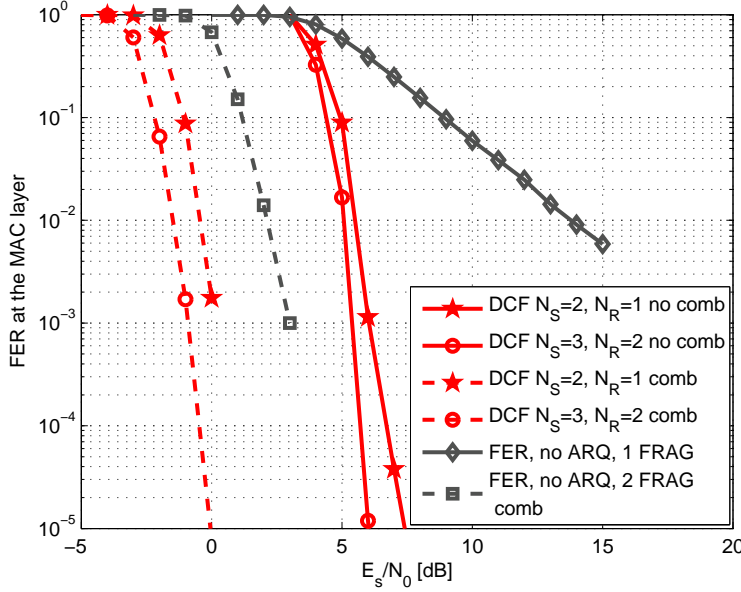


Figure 3.11: DCF protocol FER at the MAC layer as a function of E_s/N_0 , with and without combining compared to FEC and C-ARQ in Rayleigh channel, for one source, one relay, one destination scheme

E_s/N_0 , Figure 3.11; and versus the **true** E_s/N_0 , Figure 3.12. We can see that in the DCF protocol, the improvement offered by the combining at the destination is again around 6 dB. Furthermore, when we add the curves of the 2 FRAG combining, Figure 3.12, it is observed, that 2 FRAG combining performance is 2.5 dB worse than that of the $N_S = 2, N_R = 1$ DCF protocol with combining (the dotted curve with red stars).

It can also be noticed from Figure 3.12, that when there is no combining at all (i.e. the solid grey curve with diamonds), the C-shape is more vivid, whereas for the cases when there are retransmissions present, the C-shape has more moderate ascent. This is explained by the fact, that when there are retransmissions, the total energy expenditure spent per successfully received information bit is fluctuates around a steady level (e.g. around 10 dB for the target FER $[10^(-1), 10^(-5)]$ for the red solid curve with stars). In other words, the HARQ provides adaptivity of the scheme to the channel conditions, and thus spends the total power in a smarter way.

3.5.2 Efficiency

This part of the work aims at discussing the efficiency behavior for different numbers of retransmissions. Two types of curves are presented: the general efficiency, η_{gen} , and the ARQ-efficiency, η_{ARQ} which were defined in the first chapter. As a reminder, the first term takes into account the retransmissions and the code rate, whereas the second term takes into account only the retransmissions.

First of all, we can see from Figure (3.13), that the ARQ-efficiency achieves value 1

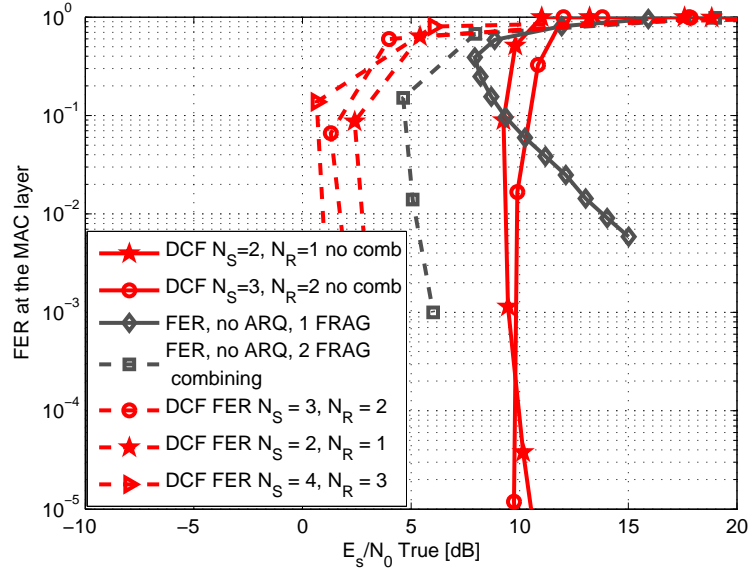


Figure 3.12: DCF protocol FER with and without combining at D at the MAC layer plotted versus **true** E_s/N_0 in Rayleigh channel with and without combining compared to FEC and C-ARQ, for one source, one relay, one destination scheme

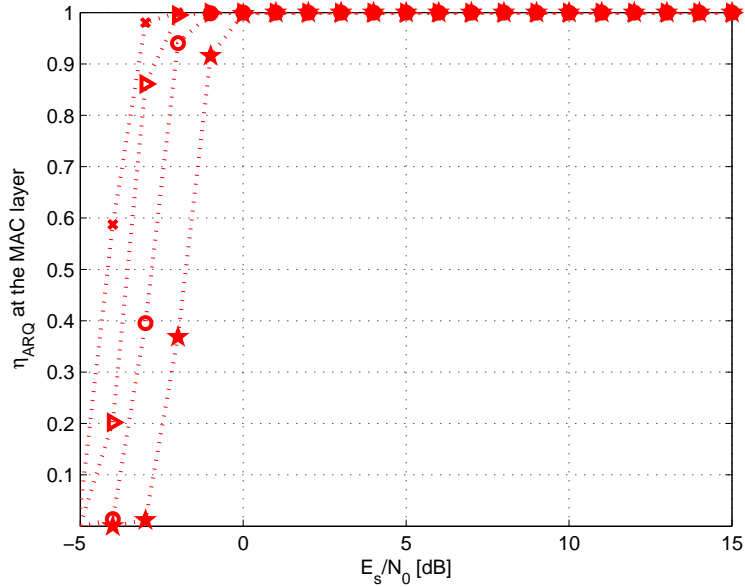


Figure 3.13: DCF protocol ARQ-efficiency, η_{ARQ} , at the MAC layer as a function of E_b/N_0 in Rayleigh channel, for one source, one relay, one destination scheme

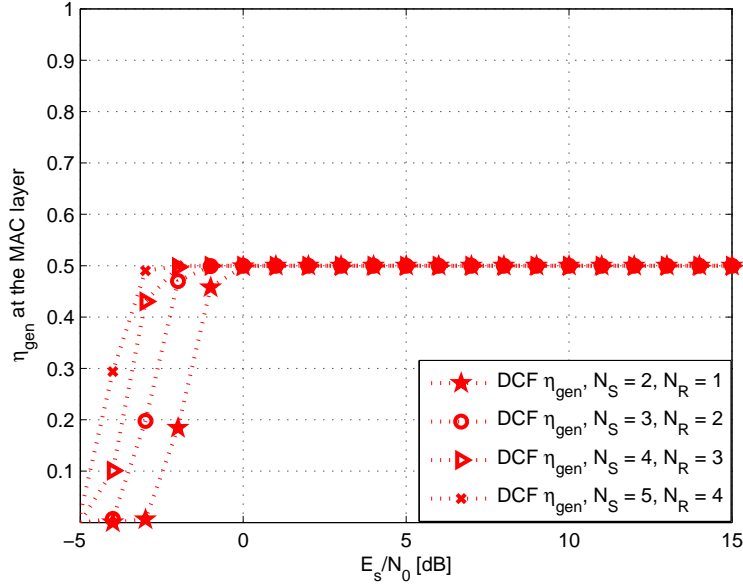


Figure 3.14: DCF protocol general efficiency, η_{gen} , at the MAC layer as a function of E_s/N_0 in Rayleigh channel, for one source, one relay, one destination scheme

because the PHY layer channel code rate is not taken into account.

By looking at Figure (3.13), one can see, that the increasing number of transmissions from $N_S = 2, N_R = 1$ to $N_S = 5, N_R = 4$ results in a large improvement, varying from $E_s/N_0 = 3 - 4$ dB. This means that by increasing the number of transmissions, the scheme provides adaptation to channel conditions, resulting in more received information bits, and consequently, in increased ARQ-efficiency.

Similar behavior is noticed for the general efficiency, Figure (3.14) with the difference that now the maximal attainable value is equal to the code rate $R_c = 0.5$. Furthermore, for both efficiencies we can see that when we continue increasing the retransmissions number, the effect is not as large as for the first time: i.e. take the ARQ-efficiency value $\eta_{ARQ} = 0.36$ for $N_S = 2, N_R = 1$ and for $N_S = 3, N_R = 2$. There is around 1 dB improvement in terms of the E_s/N_0 . Whereas when we take a look at $N_S = 4, N_R = 3$ and $N_S = 3, N_R = 2$ for the same efficiency, we see that the improvement is less significant (approximately 0.7 dB).

3.5.3 FER

In this part of the results section we compare the FER of the DCF protocol for various number of transmissions, with combining at the destination (please note that in the remainder of this work all the addressed curves represent the ones with combining at the destination, unless otherwise stated). Furthermore, we introduce the curves as a function of E_b/N_0 and E_s/N_0 without loss of generality. The relationship between the two is given by: $E_b/N_0[linear] = \frac{1}{R_c} \cdot 10^{(\frac{E_s/N_0[dB]}{10})}$, where R_c is the code rate.

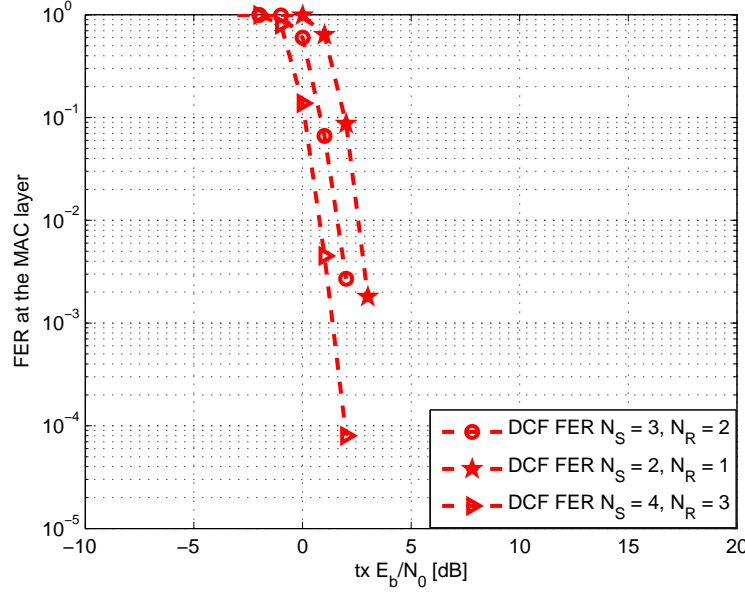


Figure 3.15: DCF protocol FER at the MAC layer as a function of E_b/N_0 , 1-s-1-r-1-d

The following notions are used:

$N_S = 2, N_R = 1$ curves with red stars markers

$N_S = 3, N_R = 2$ curves with red o markers

$N_S = 4, N_R = 3$ curves with red triangle markers

We can observe from Figure 3.15 the effect that increasing the transmissions number brings: take for example $N_S = 2, N_R = 1$ curve at $\text{FER} = 0.01$ at $E_b/N_0 = 3$ dB and compare it to the curve with $N_S = 4, N_R = 3$ at the same FER which is achieved at $E_b/N_0 = 1$ dB. So, there is around 2 dB improvement for the target $\text{FER} = 0.01$. Furthermore, it is noticed, that increasing the number of transmissions from $N_S = 2, N_R = 1$ to $N_S = 3, N_R = 2$ improves the FER performance by only 1 dB.

We can see that the curves plotted as a function of E_s/N_0 are shifted to the left, compared to the ones plotted as a function of E_b/N_0 . This is explained by the fact that the transmitted energy per symbol is less compared to the transmitted energy per information bit. In other words, each information bit gets $\frac{1}{R_c}$ times more energy because the information sequence is encoded with FEC.

Furthermore, in Figure 3.16 we also plot the curve for $N_S = 5, N_R = 4$ transmissions (curve with dot marker). Take the FER value 0.0197: the $N_S = 5, N_R = 4$ scheme achieves this FER at $E_s/N_0 = -3$ dB, whereas the smallest number of transmissions, $N_S = 2, N_R = 1$ achieves the same FER at around $E_s/N_0 = 0$ dB. In other words, the improvement brought by increasing the transmissions per FRAG from 3 to 9 is equal to 3 dB. Then, it is observed, that when the number of transmissions is $N_S = 3, N_R = 2$, then

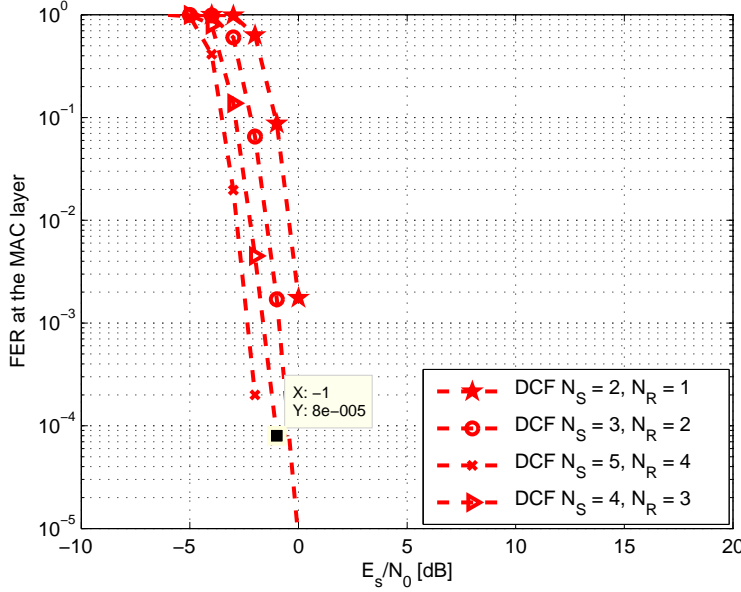


Figure 3.16: DCF protocol FER at the MAC layer as a function of E_s/N_0 , 1-s-1-r-1-d

this same $\text{FER} = 0.0197$ is achieved at $E_s/N_0 = -1.5$ dB; and for $N_S = 4, N_R = 3$ at $E_s/N_0 = -2.5$ dB. This implies, that increasing the number of transmissions by This is illustrated in Table (3.6), where N denotes the total number of transmissions per FRAG. As it can be concluded from the table, when the number of transmissions per FRAG becomes too large, i.e. $N = 9$, the improvement compared to the $N = 7$ case is not as big as it is for $N = 7$ to $N = 5$ comparison. Moreover, for the given value of $\text{FER} = 0.0197$, the ARQ-efficiency values (Figure 3.14) for these schemes are close to 1. In other words, the improvement brought by increase of retransmissions, almost does not affect the efficiency. This is illustrated in Table 3.6.

Further inquiry is, how the above described improvements in the FER affect the delay; and will be discussed in the next section.

Before moving forward, let us remark another observation: in Figure 3.16, take the $N_S = 3, N_R = 2$ and $N_S = 4, N_R = 3$ curves at $E_s/N_0 = -2$ dB. We see that in the first case we obtain $\text{FER} = 0.065$, whereas in the second case it is equal to 0.0045. Then compare the same curves at $E_s/N_0 = -1$ dB: in the first case he $\text{FER} = 0.001$, whereas for the $N_S = 4, N_R = 3$ we obtain $\text{FER} = 8 \cdot 10^{(-5)}$. It can be concluded that at high E_s/N_0 , increasing the number of retransmissions gives a diversity-type gain.

3.5.4 FER in an energetic fair context

Now let us consider the FER of the DCF protocol, plotted as a function of **true** E_b/N_0 and **true** E_s/N_0 . Please be reminded, that the **true** E_b/N_0 is computed using the η_{ARQ} ; and the **true** E_s/N_0 using η_{gen} . This is done because when we compute the **true** E_b/N_0 , the PHY layer code rate is taken into account in the $E_b/N_0[\text{linear}] = \frac{1}{R_c} \cdot \frac{E_s}{N_0}[\text{linear}]$. As

Table 3.6: Improvement achieved by increased transmissions, DCF, 1-s-1-r-1-d

FER = 0.0197				
N_S	N_R	N	E_s/N_0 , [dB]	η_{ARQ}
2	1	3	-0.3	0.98
3	2	5	-1.5	0.95
4	3	7	-2.5	0.9403
5	4	9	-3	0.9803

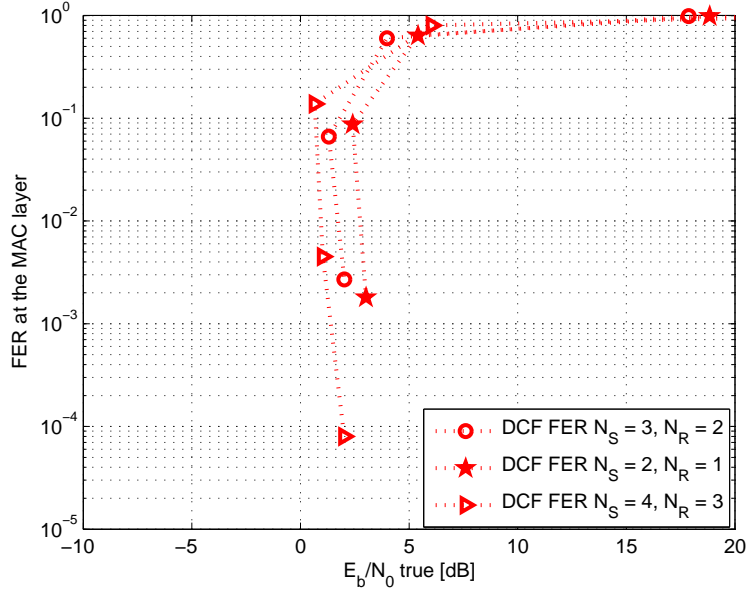


Figure 3.17: DCF protocol FER at the MAC layer versus **true** E_b/N_0 , 1-s-1-r-1-d

a result, to omit redundancy, we need to consider the efficiency that does not take into account the code rate: the η_{ARQ} .

In Figures 3.17 and 3.18 it can be observed, that the curves have the C-shape, that has been discussed earlier in the second chapter. We can see from Figure 3.17, that the difference between the $N_S = 4, N_R = 3$ and $N_S = 2, N_R = 1$ is around 2 dB of **true** E_b/N_0 .

Then let us take a value of $\eta_{gen} = 0.5$ at $E_s/N_0 = -1$ dB in Figure 3.14. The respective value of the FER at this E_s/N_0 value is equal to $8 \cdot 10^{(-5)}$, Figure 3.16. We then notice that in Figure 3.18, for this target FER $8 \cdot 10^{(-5)}$, the total energy expenditure per successfully received information bit is 2,011 dB. This is validated by the Equation 2.9:

$$\begin{aligned}
 \text{true } E_s/N_0[\text{dB}] &= E_s/N_0[\text{dB}] - \eta_{gen}[\text{dB}] \\
 &= -1 - 10\log_{10}(0.5) \\
 &= 2.011[\text{dB}]
 \end{aligned} \tag{3.33}$$

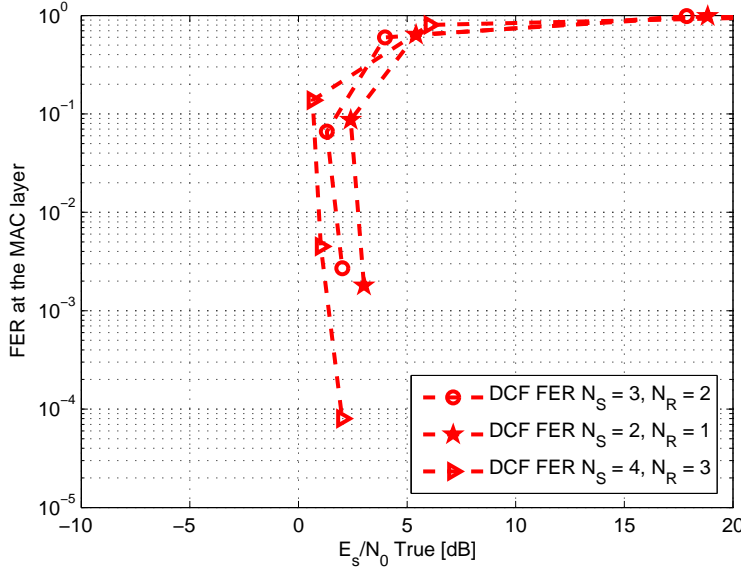


Figure 3.18: DCF protocol FER at the MAC layer versus **true** E_s/N_0 , 1-s-1-r-1-d

3.5.5 Delay

In Figures 3.19 and 3.20 we present the delay plotted as a function of E_b/N_0 and E_s/N_0 , respectively. We can see that starting from 2 dB E_b/N_0 , the delay of the $N_S = 4, N_R = 3$, $N_S = 3, N_R = 2$ and $N_S = 2, N_R = 1$ configurations is the same. This is explained by the fact that when the channel quality improves, on average the number of FRAGs to successfully receive one FRAG is the same for these configurations. This happens, because at some point (i.e. for $E_b/N_0 > 2$ dB), no matter how many retransmissions we assign to the system, the delay remains the same due to the adaptability of the scheme: i.e. on average we spend the same number of retransmissions in both cases.

This behavior is identical to the Figure 3.20, where the delay is plotted as a function of E_s/N_0 .

3.5.6 Delay in an energetic fair context

We then compare the DCF protocol behavior plotted versus the energetic fair criterions in Figures 3.21 and 3.22. As we can observe from the figures, we again obtain the C-shape of the delay curves as in the previous chapter for a point-to-point network delay. Furthermore, we see that for $N_S = 4, N_R = 3$ curve (with red triangle markers), the curve does not converge and have the C-shape. This happens because for the simulated delay evaluation, in very bad channel quality, the average number of successful FRAGs is zero. This leads to infinite delay and therefore is not plotted on the figure. As a result, when we plot the delay as a function of the **true** E_b/N_0 and **true** E_s/N_0 , we see that the curve does not obtain the C-shape.

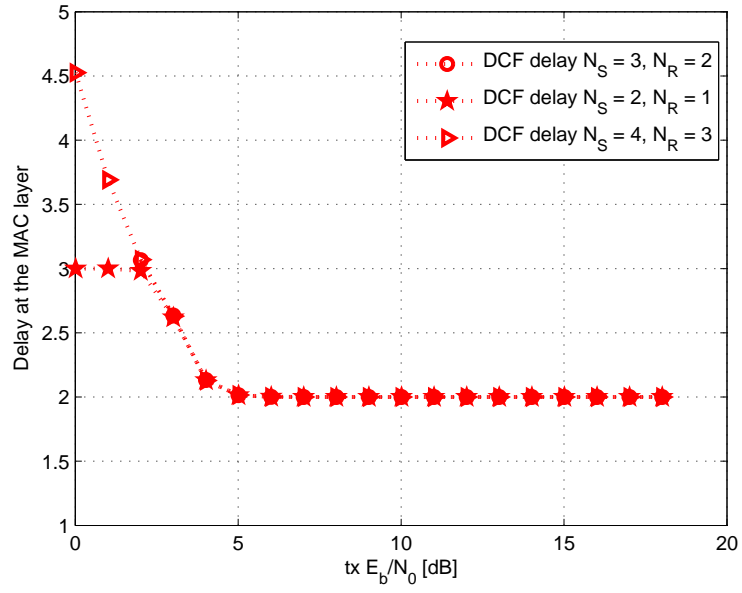


Figure 3.19: DCF protocol delay at the MAC layer as a function of E_b/N_0 , 1-s-1-r-1-d

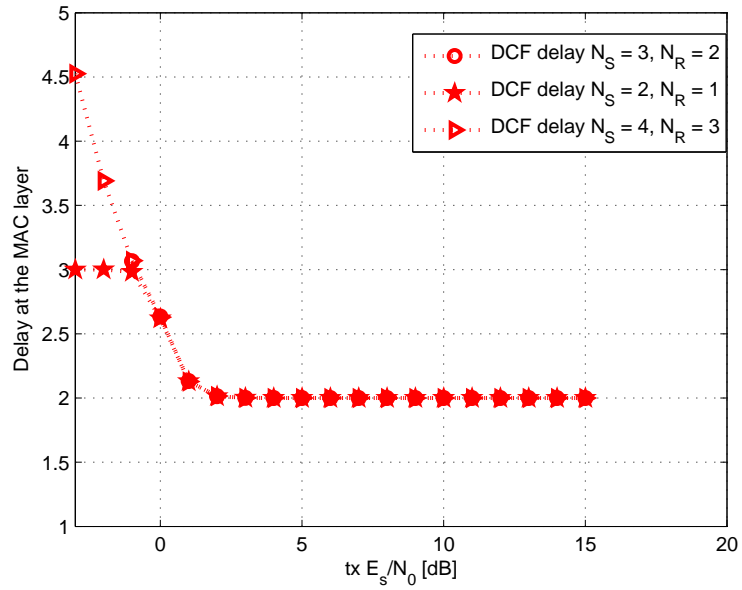


Figure 3.20: DCF protocol delay at the MAC layer as a function of E_s/N_0 , 1-s-1-r-1-d

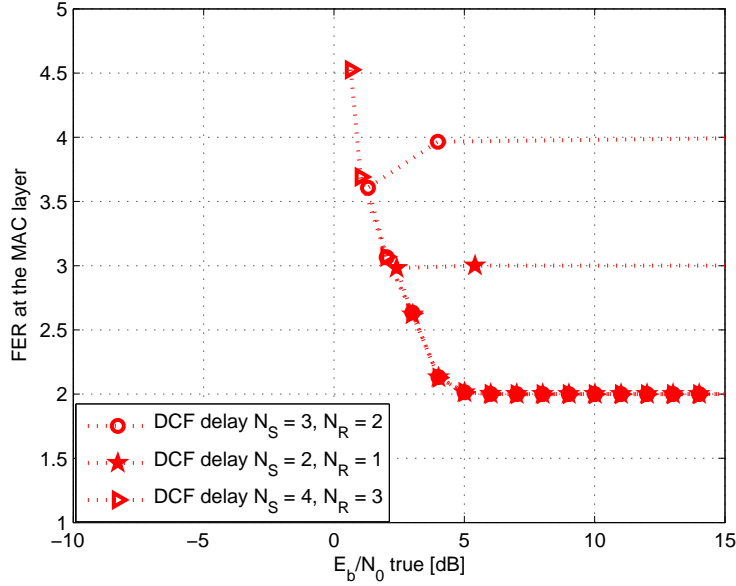


Figure 3.21: DCF protocol delay at the MAC layer versus **true** E_b/N_0 , 1-s-1-r-1-d

But what can be clearly observed, is that both in Figure 3.21 and 3.22, the delay fluctuations are larger for $N_S = 4, N_R = 3$ than for the $N_S = 3, N_R = 2$ case. This means that when the signal quality is bad, the delay of the scheme with many retransmissions can be quite larger than that of the scheme with $N_S = 3, N_R = 2$. As a consequence, in delay-tolerant systems it makes perfectly sense to use large number of transmissions in order to obtain better FER with the cost of high delays; but on the contrary, when the delay of the system becomes crucial design consideration (e.g. in video streaming); then it is better to use less number of transmissions to obtain lower delay with a tradeoff of FER (recall from Figure 3.16 that for $N_S = 3, N_R = 2$ we had FER=0.065 as opposed to the $N_S = 4, N_R = 3$ where the FER=0.0045).

Take Figure 3.22 delay for the $N_S = 4, N_R = 3$ for the **true** $E_s/N_0 = 0.64$ dB and notice that the maximal efficacy point for the general efficiency in Figure 3.14 is obtained at $E_s/N_0 = -2$ dB and is equal to 0.49. The delay at this E_s/N_0 value is equal to 3.691, Figure 3.20. Then have a look at Figure 3.22, and notice, that the **true** E_s/N_0 value for the delay 3.69 is equal to 1.028. This means that the best tradeoff between the smartest energy expenditure and the delay that can be achieved in this configuration, is given at the point **true** $E_s/N_0 = 1.028$ dB.

3.5.7 DCF versus DMF

In this part of the simulated results section, we plot the DCF and the DMF protocols results together. This part of the simulated results has the following goals: a) discuss the DMF protocol results for the FER, delay, and the efficiency; b) compare these results to the DCF protocol.

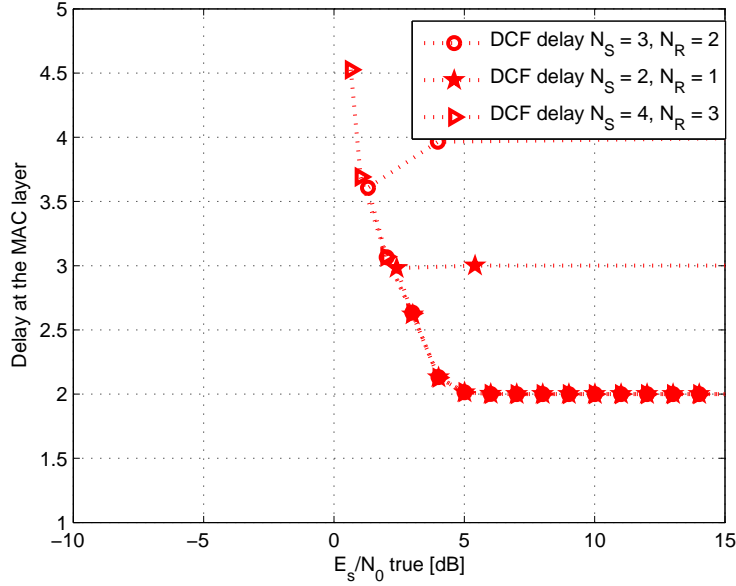


Figure 3.22: DCF protocol delay at the MAC layer versus **true** E_s/N_0 , 1-s-1-r-1-d

3.5.7.1 Efficiency

In Figures 3.23 and 3.24 the ARQ and general efficiencies are given for the comparison of DCF and DMF protocols in fully interleaved Rayleigh fading channels. It can be seen that in both cases, the DCF protocol outperforms DMF, by offering the same efficiency at lower E_s/N_0 .

By having a look at Figure 3.23 we can see that DMF protocol with $N_S = 4, N_R = 3$ efficiency is very close to that of DCF protocol's with $N_S = 3, N_R = 2$. This is explained by the fact, that DMF protocol spends more transmissions to successfully convey one FRAG, rather than DCF protocol.

3.5.7.2 FER

The FER of the DMF and DCF protocols is plotted together as a function of E_b/N_0 and of E_s/N_0 in Figures 3.25 and 3.26, respectively.

First, we observe that the DMF protocol behavior when the number of retransmissions is increased,

We also notice, by looking at Figures 3.25 and 3.26, that the difference for the two protocols FER remains 2 dB in favor of the DCF protocol. Take for example target FER = 0.0017 in Figure 3.25, for the $N_S = 4, N_R = 3$ cases for both protocols. We notice that DCF obtains this value at $E_b/N_0 = 1$ dB, whereas the DMF at $E_b/N_0 = 3$ dB. This improvement proposed by DCF protocol comes with the tradeoff of computational complexity at the relay (i.e. the time and manipulations to decode the FRAG, check it, etc.). Furthermore, we are posing the following question: will this improvement offered

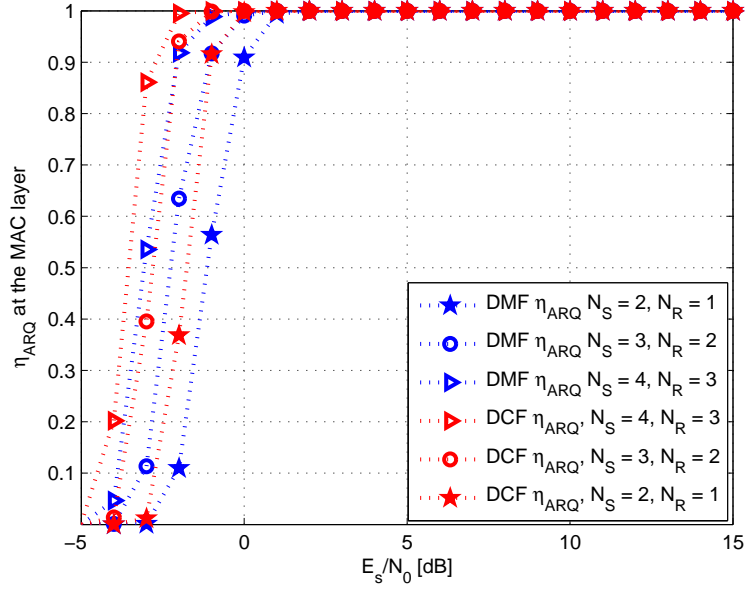


Figure 3.23: DCF and DMF protocol ARQ-efficiency comparison at the MAC layer as a function of E_b/N_0 , 1-s-1-r-1-d

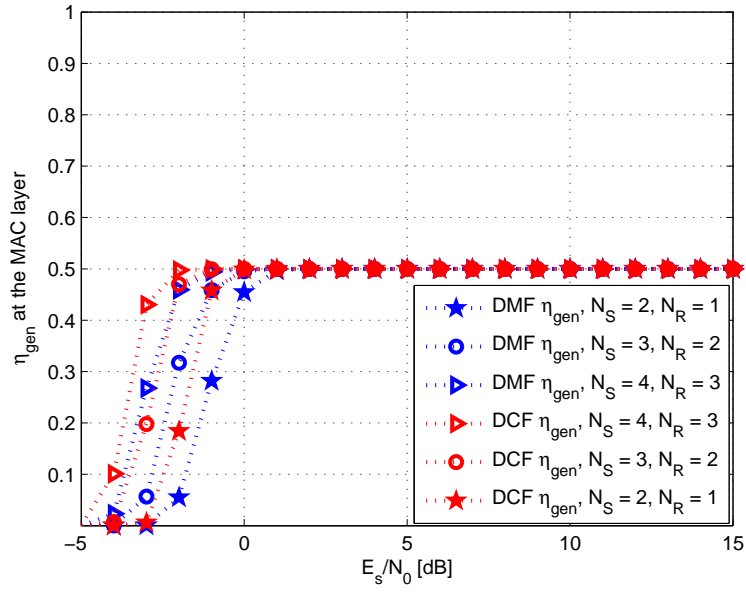


Figure 3.24: DCF and DMF protocol general efficiency comparison at the MAC layer as a function of E_b/N_0 , 1-s-1-r-1-d

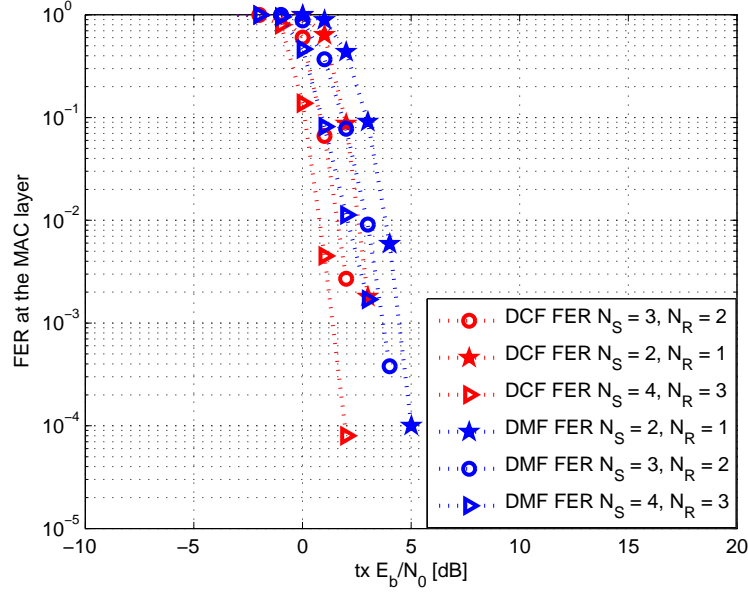


Figure 3.25: DCF and DMF protocol FER comparison at the MAC layer as a function of E_b/N_0 , 1-s-1-r-1-d

by DCF protocol in terms of the FER come also at the expense of the delay? In other words, will the delay of DCF be bigger than that of DMF or not? We will come back to this point in the delay discussion section.

Interestingly, we also notice from Figure 3.26, that in low $E_s/N_0 \in [-5; -2]$ dB, the DMF protocol with $N_S = 4, N_R = 3$ FER is almost identical to that of DCF with $N_S = 3, N_R = 2$, and the DMF $N_S = 3, N_R = 2$ to that of DCF with $N_S = 2, N_R = 1$. But as the channel quality improves, this does not hold true anymore. This is explained by the fact, that when the channel quality is very bad, the relay operation mode does not affect the performance much (i.e. because almost nothing is received, so the FER in both cases is close to 1), but however, when the channel becomes less corrupted, the operation mode of relay can improve the performance by 2 dB.

3.5.7.3 FER in an energetic fair context

We can see that in an energetic fair context (Figures 3.27 and 3.28), the behavior of the DMF protocol is quite similar to that of the DCF protocol.

We also notice from Figure 3.27, that the DMF curves with $N_S = 4, N_R = 3$, have more explicit C-form, rather than the DCF curves. In other words, the DCF curves slope with $N_S = 4, N_R = 3$ stays horizontal in the range of **true** $E_b/N_0 \in [0.6; 1.5]$ dB, whereas the DMF curve in the similar configuration in the range **true** $E_b/N_0 \in [1.38; 4]$ dB changes quite dramatically. This means, that the total energy expenditure per successfully conveyed information bit in the DMF protocol jumps very abruptly to a larger value, than in the DCF protocol. This again is explained by the protocol design: in the DMF

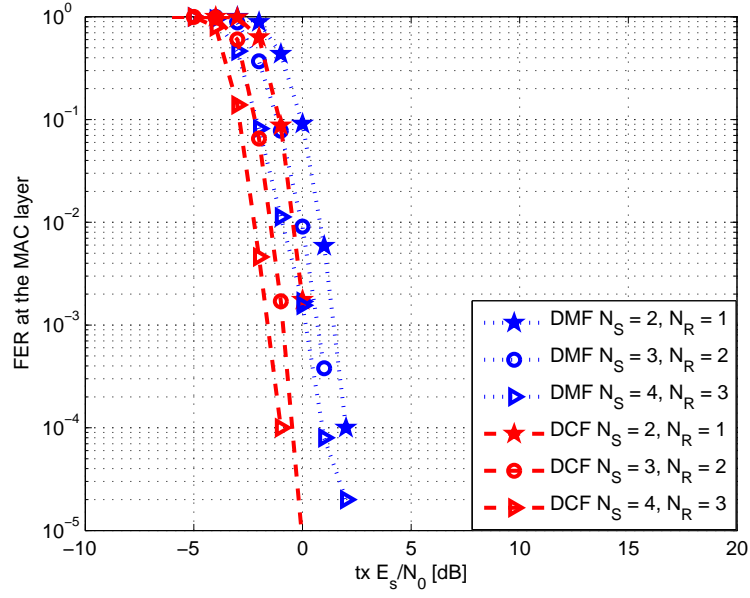


Figure 3.26: DCF and DMF protocol FER comparison at the MAC layer as a function of E_s/N_0 , 1-s-1-r-1-d

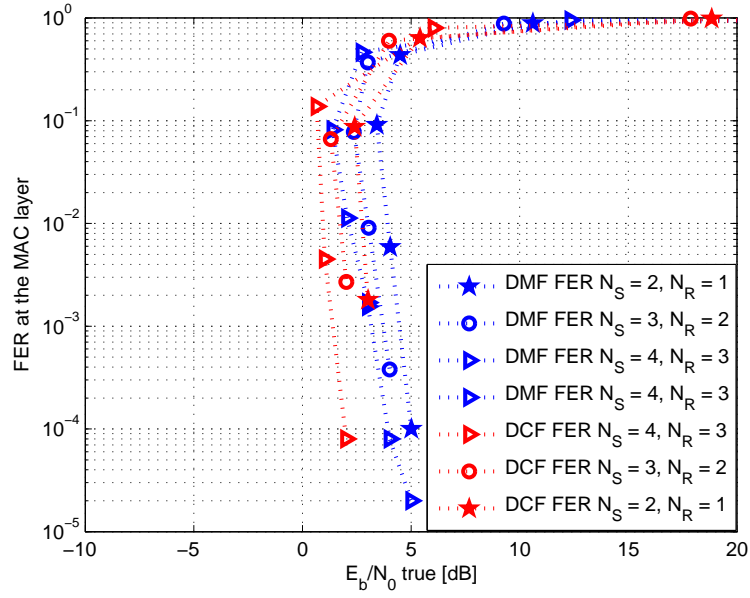


Figure 3.27: DCF and DMF protocol FER comparison at the MAC layer versus **true** E_b/N_0 , 1-s-1-r-1-d

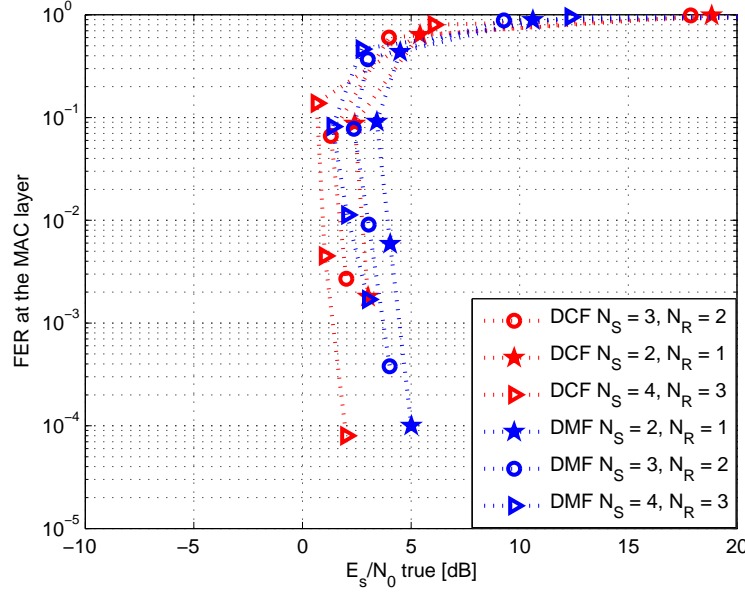


Figure 3.28: DCF and DMF protocol FER comparison at the MAC layer versus **true** E_s/N_0 , 1-s-1-r-1-d

protocol, if a FRAG is NACKed, the relay will **always** be granted the priority to retransmit maximal amount of allowed times, N_R , and only after that the source will be granted the permission to retransmit the FRAG N_S times at most. This means, that, despite the fact that the decoder at the destination takes into account the possible errors on the $S \rightarrow D$ links, still DMF over-wastes the energy per successful information bit, because it wastes retransmissions.

3.5.7.4 Delay

In this section our main concern is regarding the DMF delay behavior compared to that of DCF. We are mainly interested whether the improvement of FER attained by the DCF protocol comes at the expense of the delay or not. For this reason, we compare the delays of the protocols as a function of E_s/N_0 and E_b/N_0 in Figures 3.29 and 3.30, respectively.

Take the value of delay equal to 4 with $N_S = 4$, $N_R = 3$ for the DCF and DMF protocols in Figure 3.30. The DMF protocol achieves this value at $E_s/N_0 = 0$ dB, whereas DCF at around $E_s/N_0 = -2$ dB. Furthermore, the delay of DCF protocol in his configuration at $E_s/N_0 = 0$ dB drops to 2.6, showing that the DCF protocol delay is lower than that of the DMF protocol. This happens, because 1) in the definition of the delay we do not compute the time spent on coding-decoding manipulations at the relay in the DCF protocol; 2) DMF always sends FRAGs, resulting in errors at the destination when the channel quality is low-to-medium (i.e. $E_s/N_0 \in [-5; 0]$). This means that instead of letting the source retransmit the correct copy of the FRAG, the priority is given to the relay, who will transmit a corrupted copy of the FRAG, leading to decoding errors at the

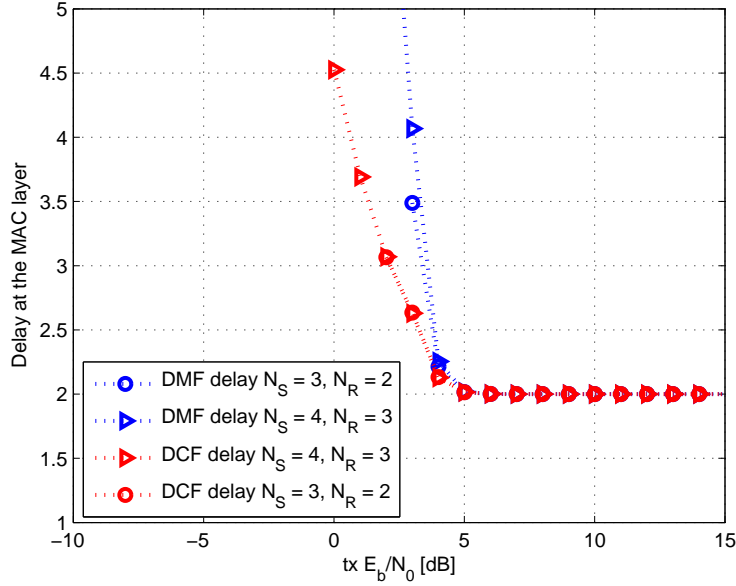


Figure 3.29: DCF and DMF protocol delay comparison at the MAC layer as a function of E_b/N_0 , 1-s-1-r-1-d

destination. As a consequence, in the DMF protocol there will be more retransmissions needed to successfully receive the FRAG in the same channel conditions, than in the DCF protocol.

3.5.7.5 Delay in an energetic fair context

Now let us consider the energetic fair comparisons of the delay for DCF and DMF protocols. Figures 3.31 and 3.32 represent the comparisons plotted versus the **true** E_b/N_0 and the **true** E_s/N_0 , respectively. We again can observe the C-shape of the curves.

We also notice, that the delay of the DMF scheme can go up to values larger than 5 in bad channel conditions, which is clearly due to the fact that the DMF will have more errors.

3.6 Conclusions

The aims of this chapter were the following:

- I. study different communications protocols for three-hop cooperative network
- II. develop analytical tools for deriving performance evaluation metrics, i.e. FER, delay, efficiency at the MAC layer
- III. apply the energetic fair criterion developed in Chapter 2 to the cooperative relay network

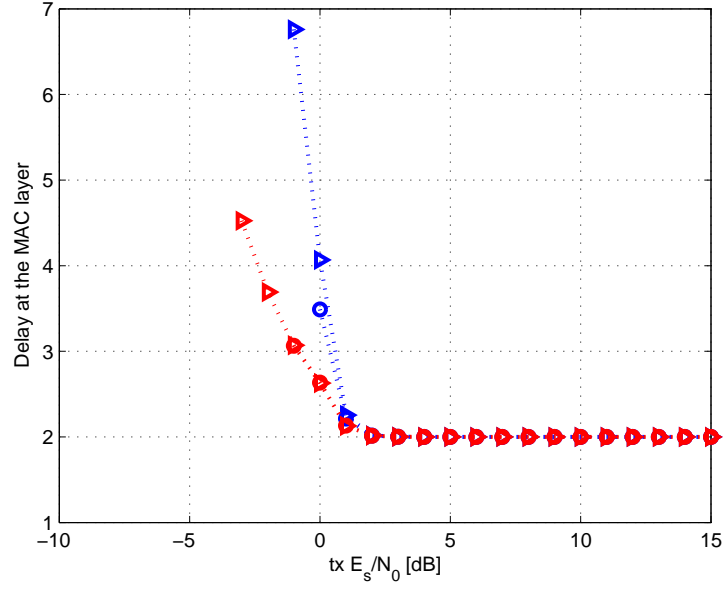


Figure 3.30: DCF and DMF protocol delay comparison at the MAC layer as a function of E_s/N_0 , 1-s-1-r-1-d

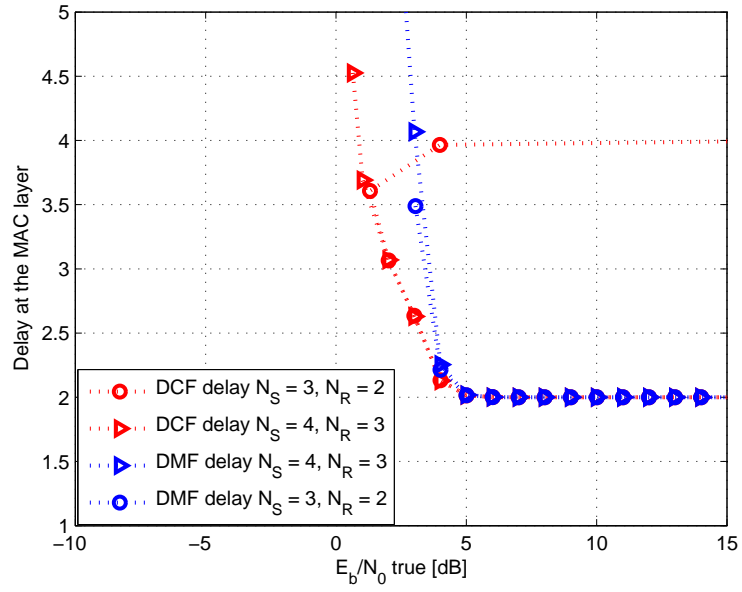


Figure 3.31: DCF and DMF protocol delay comparison at the MAC layer versus **true** E_b/N_0 , 1-s-1-r-1-d

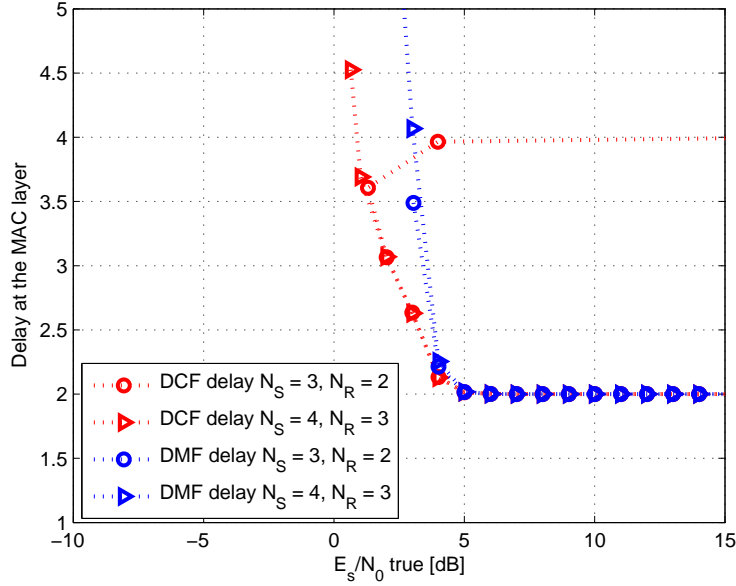


Figure 3.32: DCF and DMF protocol delay comparison at the MAC layer versus **true** E_s/N_0 , 1-s-1-r-1-d

IV. study the behavior of the protocols for various number of retransmissions

V. in the context of 1-source-1-relay-1-destination network, compare the developed protocols and discuss the advantages of each

We defined and studied various protocols, based on a) the destination behavior (combining or not); b) relay behavior (DCF or DMF). Analytical FER has been computed with the help of the FSM with and without combining at the destination for DCF and DMF protocols and validated by simulations. However it has been later revealed that as the number of retransmissions is increased, the number of states becomes too large, and the theoretical analysis is no longer possible. For this reason, we perform Monte-Carlo simulations in order to evaluate the FER, delay and the efficiency of the protocols. Furthermore, in the first part of this work it has been shown that the combining obviously provides better performance, which actually goes up to 6 dB. Based on this, all further comparisons are being done with the combining at the destination. In this context, we have compared the DCF and DMF protocols metrics and discussed their behavior as the number of retransmissions is increased. Indeed, it becomes clear, that DCF protocol has definitely more advantages in terms of the performance metrics. On the contrary, these advantages come with the expense of computational manipulations at the relay.

For all the protocols we performed evaluations with the help of the metric developed in Chapter 2, true E_b/N_0 , which actually gives the glance on the problem from another angle and allows for more conclusions.

Summarizing the tools and conclusions of this chapter, we move on to the protocol design and analysis via FSM for more complicated cooperative schemes, with the following questions in mind: "how will the number of states increase in the FSM when we increase the number of nodes and retransmissions?"; "is it useful to make the XOR-based NC of the data at the relay, when we already have HARQ?"

Chapter 4

Multuser cooperative schemes

4.1 Introduction

In Chapter 3 we have shown that the performance of cooperative schemes implementing HARQ strategies can be described using a FSMC. We considered so far very simple examples of cooperative networks, with a single user, or source node. In this part of the work we extend our analysis to cooperative networks with two sources with symmetric properties.

As soon as more nodes are introduced in the network implementing HARQ, both the design and the analysis problems increase in complexity. In particular, the introduction of the possibility of physical layer network coding at the relay serving multiple sources increases the number of possible design choices for the protocol. The FSMC analysis of systems implementing combining at the decoder, moreover, is made complex by the fact that the definition of the states needs to account not only for the number of retransmissions of the same FRAG, but also for the number and nature (direct retransmission vs network-coded retransmission) of the received copies.

The aim of this chapter is hence to introduce a structured strategy for the derivation of the FSMC associated to multuser cooperative schemes.

4.1.1 Protocol design choices

We consider systems implementing the following design strategy. We consider systems with two users. Each source sends independent data to the destination in orthogonal timeslots. The sources send information fragments in permanence, that is after correct reception of the current FRAG, or after the maximum number of allowed transmissions has expired, the sources generate a new information FRAG to be transmitted. We refer to the new FRAGs as the new generation.

In Chapter 3 only one source was generating information, and the system had to wait until current the FRAG is successfully received or not, to introduce a new generation FRAG in the transmission queue. When the number of emitting nodes increases, we choose another approach, in order to preserve each user from unnecessary delays in reception. The information sources are seen as independent, i.e. if source i has successfully received

data, it has to generate new FRAG and send it in its allocated timeslot, independently on whether or not source $j \neq i$ status. The FSMC analysis framework has to take into account that the sources have to share the same medium among each other (in this work on a time division basis), and this information will have an impact on the state definition as well.

As discussed above, the number of design choices in the protocol definition can be high. In order to facilitate the task of the description of the chosen solution, we identify a hierarchical procedure helping to define the behaviour of each component of the system.

The description of the communication protocol in case of multiple source nodes can be complex. In order to help its description we establish a hierarchy of design choices that need to be made, as follows:

1. If any, how does the relay operate? (Decode-and-Forward, Demodulate-and-Forward)?
2. If any, how does the relay decoder operate, i.e. combining the fragments or no?
3. How does the source decoder operate at the destination?
4. How many retransmissions per source, per device (i.e.: how many times the source can transmit the same FRAG, and how many times can a relay retransmit the same FRAG)?
5. Which transmitting node is granted the timeslot to emit data?
6. Once the node is granted transmission, what does it send in the current timeslot, i.e. network-coded fragment or simple fragment transmission?
7. How are the new generation fragments handled?

4.1.2 Representation of the Protocol using a FSMC

We are interested in defining the FSMC associated to the communication protocol such that:

1. each state is deterministically associated to an action taken by the communication protocol (i.e. transmission from one of the nodes);
2. the transition from one state to another is a function of the pair (current state, input). The input is given by the control messages (from the destination and the relays), instantaneously received after the action (transmission) takes place. We assume that all the nodes receive all control messages;
3. the transition probabilities depend on the probabilities of the inputs, i.e. on the probabilities of the outcomes of the decoding operations. In order to ensure the Markov property, it is necessary to guarantee that the transition probabilities for a given state i depend only on the current state and on the input.

In order to provide a definition of the states of the FSMC such that the above is satisfied, we need to be able to appropriately identify the variables that govern the communication protocol. In order to do this operation, we make use, in this chapter, of an intermediate operation: we are going to provide a functional description of the communication protocol, in the form of a flowchart, or block diagram. The flowchart can be easily constructed following the hierarchical description of the protocol detailed above. The structure of the flowchart will be used in the following to define the states of the FSMC, which can be grouped in classes (corresponding to the same action). This strategy allows to provide exhaustive but compact description of FSMC that might contain hundreds of states.

In order to introduce the details of the FSMC derivation, we begin the chapter with the very simple example of a scheme composed by two source nodes and one destination, without relays.

4.2 Two Source, One Destination Scheme

We describe first the case with two sources and one destination (no relays). This very simple scheme is used to describe the strategy that will be implemented in the following for the derivation of the FSMC associated with more complex protocols.

4.2.1 Protocol definition

The communication protocol for the two source one relay scheme is described using the hierarchical description presented in Section 4.1:

1. (no relay)
2. (no relay)
3. The decoder performs separately the decoding of each source's data, based on the detection of all the observations that has been received from the respective source (combining). The counter of the received observations of the current fragment from source i is denoted by A_i .
4. Each fragment belonging to the same source is granted a maximal number of transmissions N . Each source has a counter of remaining transmissions, S_i , that is decremented each time a (re)transmission occurs.
5. The right to transmit is granted to source 1 or source 2, depending on whom has transmitted less since the beginning of communication. Let G_i be the counter of total transmissions from source i from the beginning of communication. Source 1 is granted medium access if $G_1 \leq G_2$. Otherwise, source 2 is granted medium access. G_i can have a maximum value of G .
6. The source which is granted the timeslot retransmits either the current fragment, if $S_i < N$ or transmits a new generation fragment if $S_i = N$.

7. Source i generates a new fragment as soon as it receives an ACK (decoding successful at the destination), or as soon as $S_i = 0$ (all allowed retransmissions have been performed). As soon as source i buffers a newly generated fragment (note: buffering is not transmission) the counters need to be updated to $A_i = 0$ and $S_i = N$.

4.2.2 System model and decoding error probabilities

We consider the same system model considered in Chapter 3. The sources generate information FRAGs of length L_{FRAG} information bits. A CRC is added before FEC encoding with rate R_c , and BPSK modulation is considered. Between source 1 and destination and source 2 and destination we assume a fully-interleaved Rayleigh fading channel. The decoders (one dedicated to source 1, one dedicated to source 2) at the destination perform combining. Define $\pi_k^{(1)}$ as the probability of failure of decoding of a FRAG from source 1, when decoding is performed combining k copies. The term $\pi_k^{(1)}$ can be evaluated via simulation of the BER on the channel, as done in Chapter 3 and detailed in Appendix A.

4.2.3 Functional description of the protocol

The description of the protocol allows to determine a vector V or variables that determine its behaviour. In principle, we could define each state in the FSMC as any possible configuration of this vector. For the example of this section, we have $V = [A_1; A_2; S_1; S_2; G_1; G_2]$ with $A_i \in \{0, \dots, N\}$, $S_i = \{0, \dots, N\}$. The total number of states in the FSM would be $|\mathcal{V}| = (N + 1)^2 \cdot (N + 1)^2 \cdot G^2$.

We are going to provide a functional description of the protocol, that will be used to derive a FSMC with a reduced number of states. Since the communication protocol is deterministic, it can be described with the help of a block diagram, which consists of special blocks (representing the decoders), representing operations with multiple possible outputs (the decoder block has fixed input, and several possible outputs, associated to the probability of decoding error).

The flowchart represents the protocol in action for each timeslot, and is composed by the following subroutines, that are executed in the following hierarchical order:

1. stipulation of medium access right (conditional statements on elements of V)
2. transmission subroutines (for each transmitting agent a different subrouting is required)
3. decoding subroutine.

Moreover, the system variables update has to be performed as soon as applicable. Conditional statements are complete (i.e. when a conditional statement appears, an action is stipulated for each possible outcome). As a result, each possible loop in the flowchart (or execution of the pseudocode function) from entry-point to entry-point entails exactly one transmission and one decoding operations. The aim of this intermediate representation is to reduce the size $|\mathcal{V}|$. In order to attain this, the flowchart will be redesigned several times, by changing the definition of the system variables vector V . Each modification:

- changes the form of conditional statements on elements of V
- changes the form of system variable updating
- leaves the structure of all possible loops in the flowchart unaltered
- combines any several conditional check blocks, which form an empty loop and do not contain any blocks in between them, into one big conditional check.

In Figure 4.1 the first version of the block-diagram is given. This diagram can be further simplified by noticing, that:

- For $S_i \neq 0$, A_i and S_i have always the same value, and are always updated together. Hence, one of them is redundant and can be eliminated. Therefore, V is redefined as: $V = [A_1; A_2; G_1; G_2]$, $|\mathcal{V}| = (N + 1)^2 G^2$. Notice that A_i is now involved in the conditional expressions where S_i was. Recall that it is used in the computation of the probability of decoding error, too. For these reasons, the range of values $\{0, \dots, N\}$ cannot be changed.
- The G_i s are used only to denote the order in which the sources transmit, and since there are only two sources, these variables can be replaced by one Boolean variable taking values $\{1, 2\}$ ($G == 1$ replacing the old conditional $G_1 \leq G_2$). Hence we define $V = (A_1; A_2; G)$, $|\mathcal{V}| = (N + 1)^2 \cdot 2$.

The second version of the diagram (please note, that the conditions which are described above are respected), is given in Figure 4.2.

4.2.4 Finite state machine representation

From the block-diagram it is easy to get the FSM representation. The list of states is obtained by systematic enumeration of the elements in V . By the rules of construction of the diagram, a state D_i (corresponding to $V^{(0)}$), forces the system to loop over one and only one path, from entry-point to entry-point. The loop contains all the necessary updates to $V^{(0)}$, so that we can determine the following configuration $V^{(+1)}$, mapped in state D_j (note: since the loop contains the decoding operation, which has probabilistic output, there are in general several possible $V^{(+1)}$). The transition probabilities $[P]_{ij}$ are derived as functions of the probability of the decoder output.

A systematic and compact description of the FSM can be achieved via classification of the states. We can then partition the list of the states according to the loop they walk in the diagram. (Note: when you consider a loop in the diagram, the decoder is considered as a single block. Its internal structure determines the transitions). There are 4 possible routes in the diagram, then we are going to define 4 classes of states. Systematic enumeration of the elements of the classes builds the state list. For each state in a given class, systematic enumeration of the possible transitions builds the transition probability matrix.

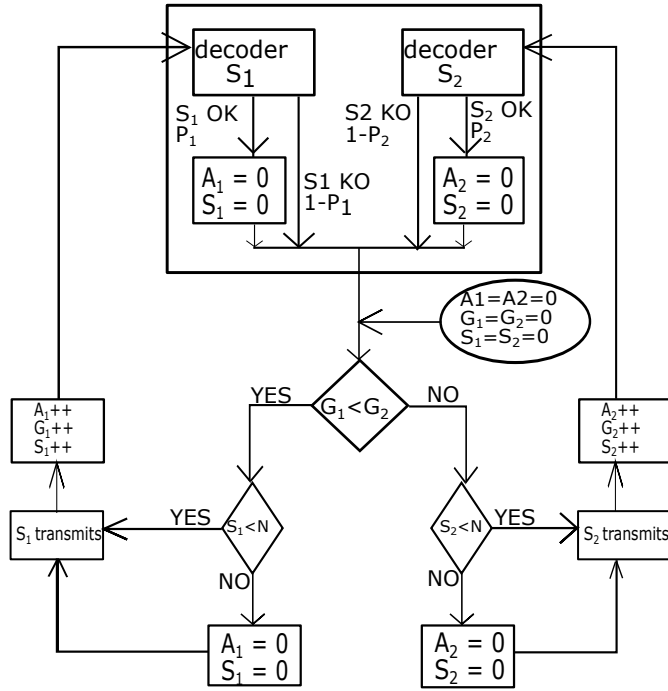


Figure 4.1: Example: block-diagram for the scheme with two sources and one destination

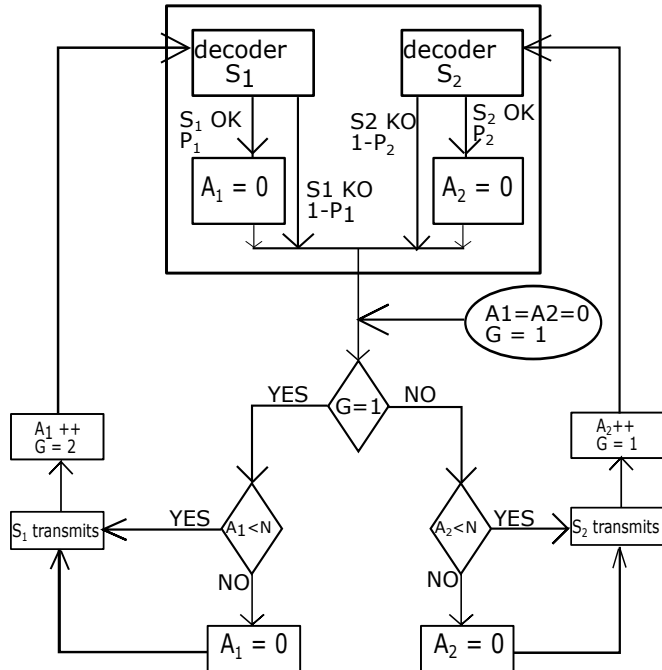


Figure 4.2: Example: block-diagram for the scheme with two sources and one destination compact form

Class \mathcal{A} Definition of the states belonging to this class.

$$\mathcal{A} = \{\alpha : G = 1; A_1 < N\} \quad (4.1)$$

For all the states in \mathcal{A} , the output is "Source 1 Transmits". As one could notice, the generic state $\alpha \in \mathcal{A}$ can be represented as:

$$\alpha = (A_1 = a_1; A_2 = a_2; G = 1) \quad (4.2)$$

Now, in order to evaluate all the possible transitions from this class of states, let us have a look at the decoder block in Figure 4.2. We can see that there are two possible transitions. So, the FSM moves to the new (arrival) states by updating the variables in the consequent blocks of the diagram (including the decoder loops):

$$\alpha_1 = (A_1 = 0; A_2 = a_2; G = 2), \text{ with probability } P_1 \quad (4.3)$$

$$\alpha_2 = (A_1 = a_1 + 1; A_2 = a_2; G = 2), \text{ with probability } (1 - P_1) \quad (4.4)$$

Recall that P_1 can be evaluated empirically as a function of $a_1 + 1$ (the total number of combined fragments at the decoder depends on the state α and its output).

Class \mathcal{B} Definition of the states belonging to this class:

$$\mathcal{B} = \{\beta : G = 2; A_2 < N\} \quad (4.5)$$

For all the states in \mathcal{B} , the output is "Source 2 Transmits".

In order to evaluate the state transitions, first we define the generic state $\beta \in \mathcal{B}$:

$$\beta = (A_1 = b_1; A_2 = a_2; G = 1) \quad (4.6)$$

As in the previous case, by looking at the decoder, we see that there are two possible transitions. The arrival states are given:

$$\beta_1 = (A_1 = b_1; A_2 = 0; G = 1), \text{ with probability } P_2 \quad (4.7)$$

$$\beta_2 = (A_1 = b_1; A_2 = b_2 + 1; G = 1), \text{ with probability } (1 - P_2) \quad (4.8)$$

Recall that P_2 can be evaluated empirically as a function of $b_2 + 1$.

Class \mathcal{C} Definition of the states belonging to this class:

$$\mathcal{C} = \{\gamma : G = 1; A_1 = N\} \quad (4.9)$$

For all the states in \mathcal{C} , the output is "Source 1 Transmits". Now we need to evaluate the possible transitions. The generic state $\gamma \in \mathcal{C}$ has form:

$$\gamma = (A_1 = N; A_2 = c_2; G = 1) \quad (4.10)$$

Looking at the decoder, one could notice, that there are two possible transitions. The FSM moves to the new (arrival) states by updating the variables in the consequent blocks of the diagram (including the decoder loops):

$$\gamma_1 = (A_1 = 0; A_2 = c_2; G = 2), \text{ with probability } P_1 \quad (4.11)$$

$$\gamma_2 = (A_1 = 1; A_2 = c_2; G = 2), \text{ with probability } (1 - P_1) \quad (4.12)$$

Recall that P_1 can be evaluated empirically as a function of 1.

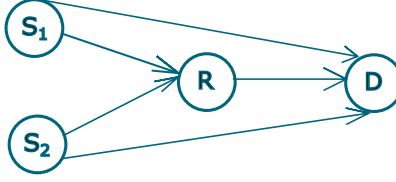


Figure 4.3: Cooperative relay network with two sources, one relay and one destination

Class \mathcal{D} Definition of the states belonging to this class:

$$\mathcal{D} = \{\delta : G = 2; A_2 = N\} \quad (4.13)$$

For all the states in \mathcal{D} , the output is "Source 2 Transmits". Now we need to evaluate the possible transitions. The generic state $\delta \in \mathcal{D}$ is given:

$$\delta = (A_1 = d_1; A_2 = N; G = 2) \quad (4.14)$$

Looking at the decoder, that there are two possible transitions. Similarly to the previous case, the arrival states are evaluated by following the respective route in the diagram:

$$\delta_1 = (A_1 = d_1; A_2 = 0; G = 1) \text{ with probability } P_2 \quad (4.15)$$

$$\delta_2 = (A_1 = d_1; A_2 = 1; G = 1) \text{ with probability } (1 - P_2) \quad (4.16)$$

Recall that P_2 can be evaluated empirically as a function of 1.

Evaluation of the performance

Similarly to the previous chapter, in order to obtain the FER, one has to evaluate the stationary probabilities of the states associated with the "Drop FRAG" event. These states are easily identified by looking at the loops in the block-diagram. For the present example, they are all the elements in the set $\mathcal{C} \cup \mathcal{D}$.

4.3 Two Source, One Relay, One Destination Scheme, DCF

In this section we use the framework developed for the two source and one destination scheme, by adding one relay to the network topology, Figure (4.3). The channels between any two nodes of the network are i.i.d. Rayleigh fully interleaved with AWGN. Since the relay is located between the source and the destination, the channels between S_i -R and R-D have twice the SNR of S_i -D.

4.3.1 Definition of the communication protocol

In this section we first define the priorities list for any protocol, that has to be respected and that defines the behavior of the system. We first detail it with the simple example of scheme without relays, and then do the same procedure for the scheme with two sources, one relay and one destination. We refer to the new FRAGs as the new generation in the

further text of this work. We also define the following counters: transmission at the source i , which increments with each transmission of source i by A_i ; the counter of the source i 's data at the relay by B_i , which is incremented as soon as the relay transmits the data of source i . If the relay transmits the XOR, both counters B_i are incremented.

Let us now list the priorities, based on the questions set in Section 4.1.1:

1. Relay operates in the DCF mode: it sends the data only if it has correctly received it. In order to distinguish whether or not the relay is allowed to retransmit data, we define the binary variable $R_i \in \{0, 1\}$, which takes values based on whether or not the relay correctly received the fragment of i -th source, S_i :

$$R_i = \begin{cases} 0, & \text{if the FRAG of } S_i \text{ the relay has a valid copy of the active fragment from source } i \\ 1, & \text{if the FRAG of } S_i \text{ the relay has a valid copy of the active fragment from source } i \end{cases}$$

2. The relay decoder does not perform combining (i.e. it decodes only on the basis of the last received message from the source).
3. The source decoder performs combining of all pertinent messages received in the past. See below for details.
4. Each source is allowed to transmit the FRAG N times, i.e. $A_i \in \{0, 1, \dots, N\}$.

The relay is allowed to transmit the FRAG of each source i N times, i.e. $B_i \in \{0, 1, \dots, N\}$

5. The relay is granted transmission right in priority, if $R_1 = R_2 = 1$. If this is not the case, then we have to check which source's fragment has the priority to be transmitted.

The general transmission counters are initialized to $G_{S_1} = 0$ and $G_{S_2} = 0$ in the beginning of new generation FRAG transmission. These counters are used for the timeslot allocation (i.e. "whose turn is it") and are incremented as soon as a fragment, associated with the corresponding source, is emitted. If $G_{S_1} \leq G_{S_2}$, then transmission right is given to a copy of the active fragment from source 1. If $R_1 = 1$, it is the relay sending, otherwise it is the source. To simplify the notation, we define a variable G , which is equal to 1 if it is source 1 data to be transmitted, and to 2 if it is source 2's data. When the relay sends XOR, this variable stays unchanged.

6. If the relay is granted transmission rights because $R_1 = R_2 = 1$, then the network coded packet of the fragments is sent. In this case both transmission counters B_1 and B_2 are incremented. In all the other cases, the device granted transmission right sends a direct copy of the active fragment.

In other words:

IF

$$\left. \begin{array}{l} R_1 = 1 \text{ AND } R_2 = 1 \\ B_1 < N \text{ AND } B_2 < N \end{array} \right\} \text{ TRUE} \Rightarrow \text{the relay retransmits XOR}$$

ELSEIF

$$\left. \begin{array}{l} R_1 = 1 \text{ AND } R_2 = 0 \\ B_1 < N \\ G = 1 \end{array} \right\} \text{TRUE} \Rightarrow \text{the relay retransmits FRAG of } S_1$$

ELSEIF

$$\left. \begin{array}{l} R_1 = 0 \text{ AND } R_2 = 1 \\ B_2 < N \\ G = 2 \end{array} \right\} \text{TRUE} \Rightarrow \text{the relay retransmits FRAG of } S_2$$

ELSEIF

$$\left. \begin{array}{l} R_1 = 0 \text{ AND } R_2 = 0 \\ G = 1 \end{array} \right\} \text{TRUE} \Rightarrow \text{the } S_1 \text{ is allowed to (re)transmit}$$

ELSEIF

$$\left. \begin{array}{l} R_1 = 0 \text{ AND } R_2 = 0 \\ G = 2 \end{array} \right\} \text{TRUE} \Rightarrow \text{the } S_2 \text{ is allowed to (re)transmit}$$

ELSEIF

$$\left. \begin{array}{l} R_1 = 1 \text{ AND } R_2 = 0 \\ B_1 < N \\ G = 2 \end{array} \right\} \text{TRUE} \Rightarrow \text{the } S_2 \text{ is allowed to (re)transmit}$$

ELSEIF

$$\left. \begin{array}{l} R_1 = 0 \text{ AND } R_2 = 1 \\ B_2 < N \\ G = 1 \end{array} \right\} \text{TRUE} \Rightarrow \text{the } S_1 \text{ is allowed to (re)transmit}$$

7. A new generation fragment at source i is buffered as soon as the current fragment is correctly decoded, or as soon as both the source and the relay have expired the maximum number of allowed transmissions.

In this section we defined the set of rules by which the system operates in the DCF protocol. As it can be seen, each of the rules has its respective priority, which will simplify the transition from this definition to the algorithm block diagram.

4.3.2 System model and decoder design

In order to get the decoding error probabilities for the two source, one relay, one destination scheme, we have to derive the likelihoods to be fed to each channel decoder. The demodulation is being done jointly. The input of the demodulator are the direct copies of the active messages received in the past (this includes also network coded packets for which one of the messages have already been decoded), and the all the network coded packets involving the active fragments. To limit complexity, network coded packets involving already dropped fragments are discarded by the decoder.

Detailed explanations on the functioning of the decoder is provided in Section B.1 in Appendix B.

4.3.3 Functional description of the protocol

In the previous section we prioritized the actions of the system in the hierarchical order. Due to this classification, we can design the algorithm block diagram. The simplified block-diagram is given in Figure (4.5). Operating the substitution on G_1 and G_2 with the binary variable G , the system variables vector takes form $V = [R_1; R_2; A_1; A_2; B_1; B_2; G; T_1; T_2; T_3]$. We have $|\mathcal{V}| = 2^2 \cdot (N + 1)^4 \cdot 2 \cdot (2N)^2 \cdot N$.

Unlike the simplest cooperative relay network, this scheme requires more variables, as it could be observed in the previous sections. In such case, the number of states, defined in subsection ??, is much bigger.

4.3.4 FSMC description

By looking at Figure 4.6. As one could see, there are various paths that can be followed in the block diagram, which are highlighted in red, blue, green, yellow, cyan, and magenta colors. As it can be seen in the figure, the system is symmetric since we have two sources, which makes the classification procedure simpler.

For example, in order to define a class of the states in the FSM, let us follow the red lines, which are the result of the following conditional block checks:

1. $R_1 = 1$ and $R_2 = 1 \Rightarrow \text{YES}$
2. $B_1 < N$ and $B_2 < N \Rightarrow \text{YES}$

In order to define a class, the specific path has to be followed within the block-diagram. For example, by following the green, dark blue, magenta, yellow and cyan routes in the FSM, we obtain new classes. These classes have their symmetric ones that are related to source 2. Another set of classes, that is not highlighted in the figure, will be defined by following the following path:

1. $R_1 = 1$ and $R_2 = 1 \Rightarrow \text{YES}$
2. $B_1 < N$ and $B_2 < N \Rightarrow \text{NO}$
3. ...

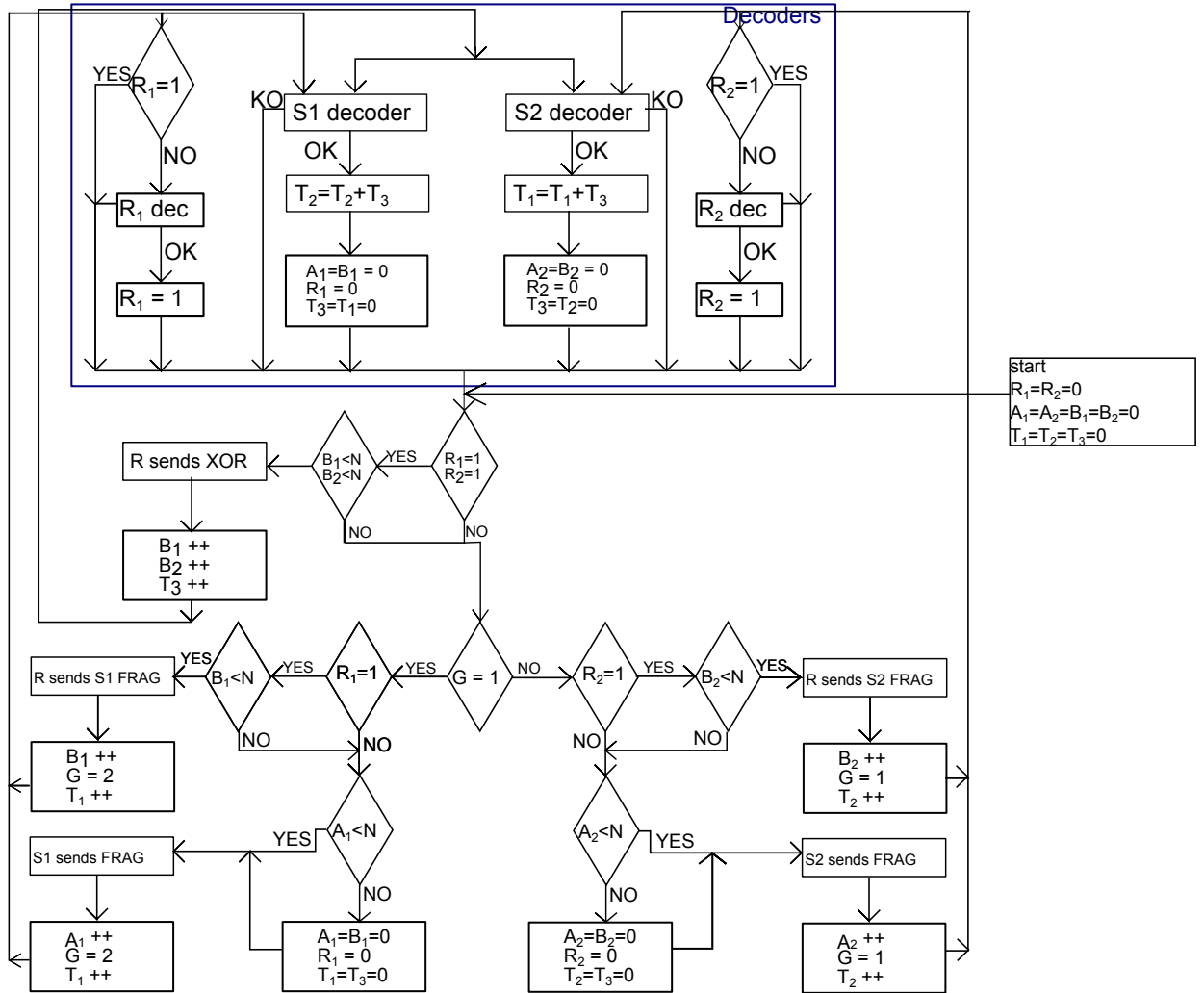


Figure 4.4: Block-diagram DCF protocol of the 2 source 1 relay 1 destination scheme

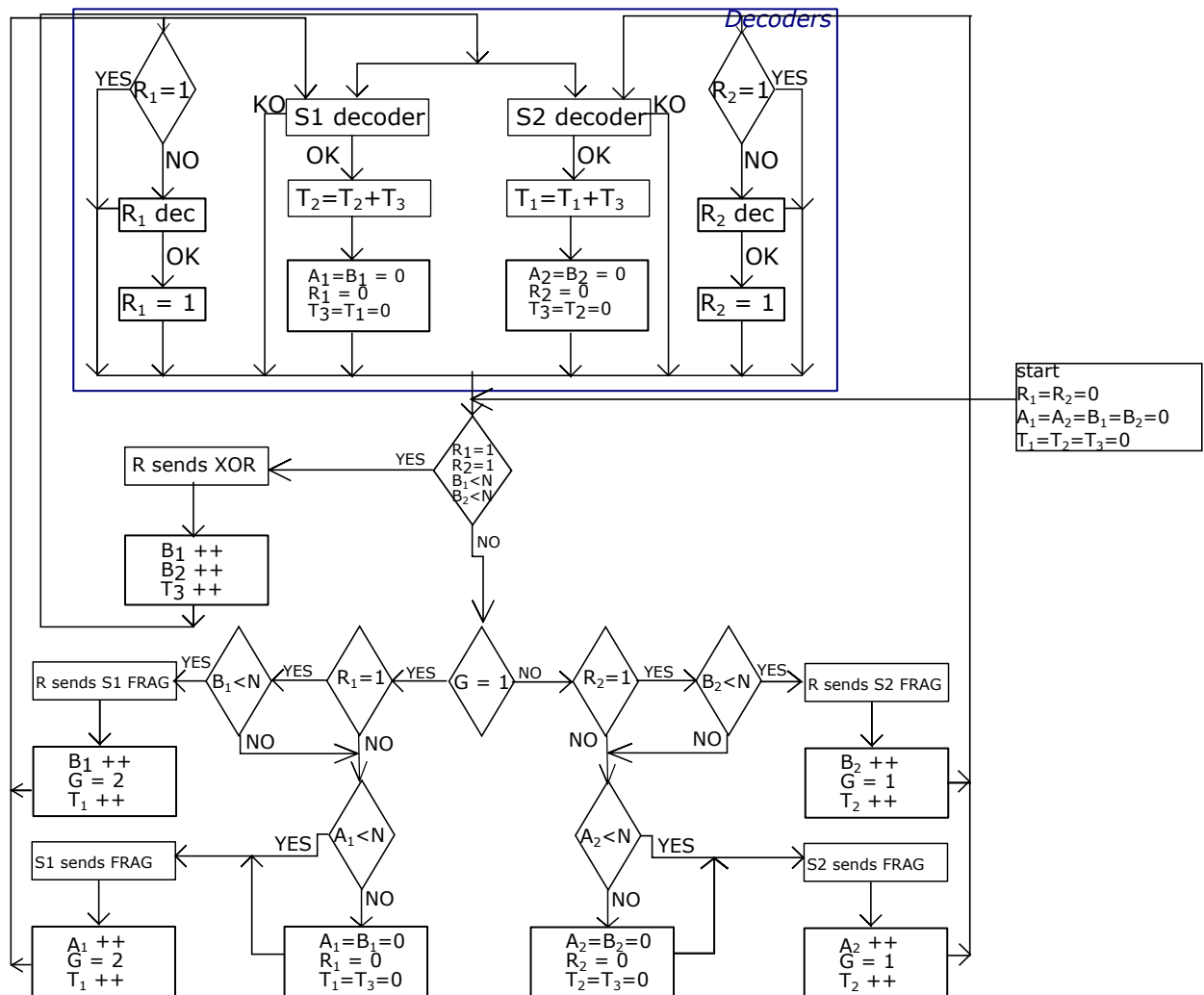


Figure 4.5: Block-diagram DCF protocol of the 2 source 1 relay 1 destination scheme: compact form

This route gives a set of classes, that coincide with the green, dark blue, magenta, yellow, and cyan routes in the block diagram.

The classes of the FSMC are hence defined as follows:

Class \mathcal{A} Definition of the states belonging to this class.

$$\mathcal{A} = \{\alpha : R_1 = 1, R_2 = 1, B_1 < N, B_2 < N\} \quad (4.17)$$

which corresponds to the red route in Figure 4.6. For all the states in \mathcal{A} , the output is "Relay Transmits XORed FRAG". The generic state $\alpha \in \mathcal{A}$ can be represented as:

$$\alpha = (R_1 = 1, R_2 = 1, A_1 = a_1, A_2 = a_2, B_1 = b_1, B_2 = b_2, G = g, T_1 = t_1, T_2 = t_2, T_3 = t_3) \quad (4.18)$$

There are four possible transitions, depending on the outputs of the decoder block, 4.4. Consequently, there are four output states.

$$\begin{aligned} \alpha_1 &= (R_1 = 1, R_2 = 1, A_1 = a_1, A_2 = a_2, B_1 = b_1 + 1, B_2 = b_2 + 1, G = g, T_1 = t_1, T_2 = t_2, T_3 = t_3 + 1) \\ \alpha_2 &= (R_1 = 0, R_2 = 1, A_1 = 0, A_2 = a_2, B_1 = 0, B_2 = b_2 + 1, G = g, T_1 = 0, T_2 = t_2 + t_3, T_3 = 0) \\ \alpha_3 &= (R_1 = 1, R_2 = 0, A_1 = a_1, A_2 = 0, B_1 = b_1 + 1, B_2 = 0, G = g, T_1 = t_1 + t_3, T_2 = 0, T_3 = 0) \\ \alpha_4 &= (R_1 = 0, R_2 = 0, A_1 = 0, A_2 = 0, B_1 = 0, B_2 = 0, G = g, T_1 = 0, T_2 = 0, T_3 = 0) \end{aligned}$$

with probabilities P_1, P_2, P_3, P_4 , respectively, which can be evaluated as functions of $T_1 = t_1, T_2 = t_2, T_3 = t_3 + 1$.

Class \mathcal{B} Definition of the states belonging to this class.

$$\mathcal{B} = \{\beta : \beta \notin \mathcal{A}, G = 1, R_1 = 1, B_1 < N\} \quad (4.19)$$

which corresponds to the dark blue route in Figure 4.6. The generic state $\beta \in \mathcal{B}$ can be represented as:

$$\beta = (R_1 = 1, R_2 = r_2, A_1 = a_1, A_2 = a_2, B_1 = b_1, B_2 = b_2, G = 1, T_1 = t_1, T_2 = t_2, T_3 = t_3) \quad (4.20)$$

The output action is always "Relay Transmits FRAG of S_1 ". There are two possible transitions, depending on the outputs of the decoder block, 4.4. Consequently, there are two output states:

$$\begin{aligned} \beta_1 &= (R_1 = 1, R_2 = r_2, A_1 = a_1, A_2 = a_2, B_1 = b_1 + 1, B_2 = b_2, G = 2, T_1 = t_1 + 1, T_2 = t_2, T_3 = t_3) \\ \beta_2 &= (R_1 = 0, R_2 = r_2, A_1 = 0, A_2 = a_2, B_1 = 0, B_2 = b_2, G = 2, T_1 = 0, T_2 = t_2 + t_3, T_3 = 0) \end{aligned}$$

where the transition probabilities are evaluated as functions of $T_1 = t_1 + 1, T_2 = t_2$, and $T_3 = t_3$.

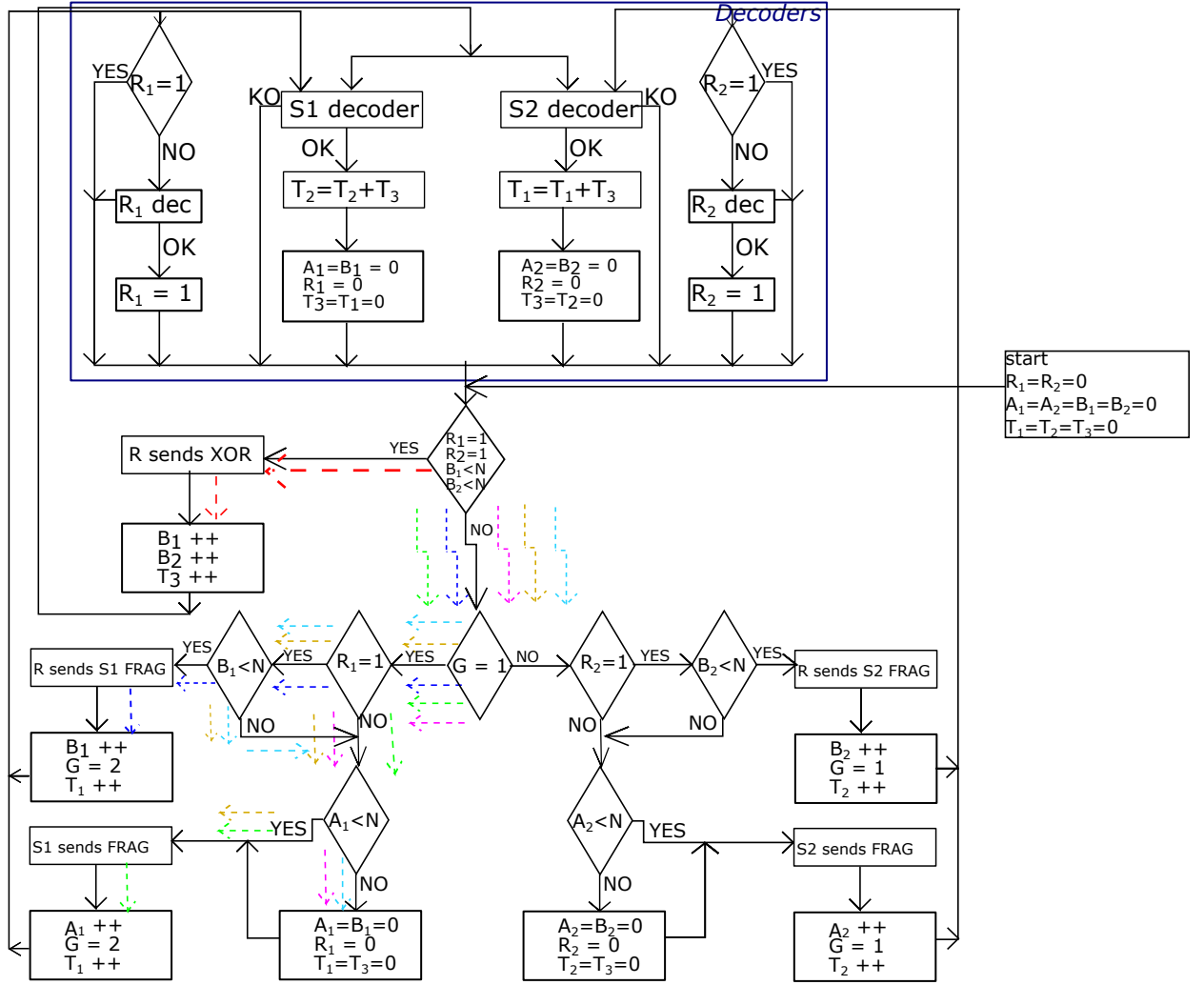


Figure 4.6: Block-diagram DCF protocol of the 2 source 1 relay 1 destination scheme: states classification

Class \mathcal{B}' Definition of the states belonging to this class.

$$\mathcal{B}' = \{\beta' : \beta' \notin \mathcal{A}, G = 2, R_2 = 1, B_2 < N\} \quad (4.21)$$

The generic state $\beta' \in \mathcal{B}'$ can be represented as:

$$\beta' = (R_1 = r_1, R_2 = 1, A_1 = a_1, A_2 = a_2, B_1 = b_1, B_2 = b_2, G = 1, T_1 = t_1, T_2 = t_2, T_3 = t_3) \quad (4.22)$$

The output is always "Relay Transmits FRAG of S_2 ", and the state transitions are symmetric to those of class \mathcal{B} .

Class \mathcal{C} Definition of the states belonging to this class.

$$\mathcal{C} = \{\gamma : \gamma \notin \mathcal{A}, G = 1, R_1 = 0, A_1 < N\} \quad (4.23)$$

which corresponds to the green route in Figure 4.6. The generic state $\gamma \in \mathcal{C}$ can be represented as:

$$\gamma = (R_1 = 0, R_2 = r_2, A_1 = a_1, A_2 = a_2, B_1 = b_1, B_2 = b_2, G = 1, T_1 = t_1, T_2 = t_2, T_3 = t_3) \quad (4.24)$$

The output action is always " S_1 Transmits FRAG of S_1 ". There are three possible transitions, depending on the outputs of the decoder block, 4.4. Consequently, there are three output states:

$$\begin{aligned} \gamma_1 &= (R_1 = 0, R_2 = r_2, A_1 = a_1 + 1, A_2 = a_2, B_1 = b_1, B_2 = b_2, G = 2, T_1 = t_1 + 1, T_2 = t_2, T_3 = 0) \\ \gamma_2 &= (R_1 = 1, R_2 = r_2, A_1 = a_1 + 1, A_2 = a_2, B_1 = b_1, B_2 = b_2, G = 2, T_1 = t_1 + 1, T_2 = t_2, T_3 = 0) \\ \gamma_3 &= (R_1 = 0, R_2 = r_2, A_1 = 0, A_2 = a_2, B_1 = 0, B_2 = b_2, G = 2, T_1 = 0, T_2 = t_2 + t_3, T_3 = 0) \end{aligned}$$

where the transition probabilities are evaluated as functions of $T_1 = t_1 + 1$, $T_2 = t_2$, and $T_3 = t_3$.

Class \mathcal{C}' Definition of the states belonging to this class.

$$\mathcal{C}' = \{\gamma' : \gamma' \notin \mathcal{A}, \gamma \notin \mathcal{B}, G = 2, R_2 = 0, A_2 < N\} \quad (4.25)$$

The generic state $\gamma \in \mathcal{C}$ can be represented as:

$$\gamma' = (R_1 = r_1, R_2 = 0, A_1 = a_1, A_2 = a_2, B_1 = b_1, B_2 = b_2, G = 2, T_1 = t_1, T_2 = t_2, T_3 = t_3) \quad (4.26)$$

The output action is always " S_2 Transmits FRAG of S_2 ", and the transition probabilities are symmetric to those of class \mathcal{C} , and are evaluated as functions of $T_1 = t_1$, $T_2 = t_2 + 1$, and $T_3 = t_3$.

Class \mathcal{D} Definition of the states belonging to this class.

$$\mathcal{D} = \{\delta : \delta \notin \mathcal{A}, G = 1, R_1 = 1, B_1 = N, A_1 < N\} \quad (4.27)$$

which corresponds to the yellow route in Figure 4.6. The generic state $\delta \in \mathcal{D}$ can be represented as:

$$\delta = (R_1 = 1, R_2 = r_2, A_1 = a_1, A_2 = a_2, B_1 = N, B_2 = b_2, G = 1, T_1 = t_1, T_2 = t_2, T_3 = t_3) \quad (4.28)$$

The output action is always " S_1 Transmits FRAG of S_1 ".

There are two possible transitions, depending on the outputs of the decoder block, 4.4. Consequently, there are two output states:

$$\begin{aligned} \delta_1 &= (R_1 = 0, R_2 = r_2, A_1 = a_1 + 1, A_2 = a_2, B_1 = b_1, B_2 = b_2, G = 2, T_1 = t_1 + 1, T_2 = t_2, T_3 = 0) \\ \delta_2 &= (R_1 = 0, R_2 = r_2, A_1 = 0, A_2 = a_2, B_1 = 0, B_2 = b_2, G = 2, T_1 = 0, T_2 = t_2 + t_3, T_3 = 0) \end{aligned}$$

Class \mathcal{D}' Definition of the states belonging to this class.

$$\mathcal{D}' = \{\delta' : \delta' \notin \mathcal{A}, \delta' \notin \mathcal{B}, \delta' \notin \mathcal{C}, G = 2, R_2 = 1, B_2 = N, A_2 < N\} \quad (4.29)$$

The generic state $\delta \in \mathcal{D}'$ can be represented as:

$$\delta' = (R_1 = r_1, R_2 = r_1, A_1 = a_1, A_2 = a_2, B_1 = b_1, B_2 = N, G = 2, T_1 = t_1, T_2 = t_2, T_3 = t_3) \quad (4.30)$$

The output action is always " S_2 Transmits FRAG of S_2 ". The state transitions are symmetrical to those of class \mathcal{D} .

Class \mathcal{E} Definition of the states belonging to this class.

$$\mathcal{E} = \{\epsilon : \epsilon \notin \mathcal{A}, G = 1, R_1 = 0, A_1 = N\} \quad (4.31)$$

which corresponds to the magenta route in Figure 4.6. The generic state $\epsilon \in \mathcal{E}$ can be represented as:

$$\epsilon = (R_1 = 0, R_2 = r_2, A_1 = N, A_2 = a_2, B_1 = b_1, B_2 = b_2, G = 1, T_1 = t_1, T_2 = t_2, T_3 = t_3) \quad (4.32)$$

The output action is always " S_1 Transmits new FRAG of S_1 ".

There are three possible transitions, depending on the outputs of the decoder block, 4.4. Consequently, there are three output states:

$$\begin{aligned} \epsilon_1 &= (R_1 = 0, R_2 = r_2, A_1 = 1, A_2 = a_2, B_1 = b_1, B_2 = b_2, G = 2, T_1 = 1, T_2 = t_2, T_3 = 0) \\ \epsilon_2 &= (R_1 = 1, R_2 = r_2, A_1 = 1, A_2 = a_2, B_1 = 0, B_2 = b_2, G = 2, T_1 = 1, T_2 = t_2, T_3 = 0) \\ \epsilon_3 &= (R_1 = 0, R_2 = r_2, A_1 = 0, A_2 = a_2, B_1 = 0, B_2 = b_2, G = 2, T_1 = 0, T_2 = t_2 + t_3, T_3 = 0) \end{aligned}$$

with probabilities evaluated as functions of the parameters $T_1 = 1, T_2 = t_2, T_3 = 0$.

Class \mathcal{E}' Definition of the states belonging to this class.

$$\mathcal{E}' = \{\epsilon' : G = 2, R_2 = 0, A_2 = N\} \quad (4.33)$$

The generic state $\epsilon \in \mathcal{E}$ can be represented as:

$$\epsilon' = (R_1 = r_1, R_2 = 0, A_1 = a_1, A_2 = N, B_1 = b_1, B_2 = b_2, G = 2, T_1 = t_1, T_2 = t_2, T_3 = t_3) \quad (4.34)$$

The output action is always " S_2 Transmits new FRAG of S_2 ". The state transitions are symmetric to those of class \mathcal{E} and the transition probabilities evaluated as functions of the parameters $T_1 = t_1, T_2 = 1, T_3 = 0$.

Class \mathcal{Z} Definition of the states belonging to this class.

$$\mathcal{Z} = \{\zeta : \zeta \notin \mathcal{A}, G = 1, R_1 = 1, B_1 = N, A_1 = N\} \quad (4.35)$$

which corresponds to the sky blue route in Figure 4.6. The generic state $\zeta \in \mathcal{Z}$ can be represented as:

$$\zeta = (R_1 = 1, R_2 = r_2, A_1 = N, A_2 = a_2, B_1 = N, B_2 = b_2, G = 1, T_1 = t_1, T_2 = t_2, T_3 = t_3) \quad (4.36)$$

The output action is always " S_1 Transmits new FRAG of S_1 ".

There are three possible transitions, depending on the outputs of the decoder block, 4.4. Consequently, there are three output states:

$$\begin{aligned} \zeta_1 &= (R_1 = 0, R_2 = r_2, A_1 = 1, A_2 = a_2, B_1 = 0, B_2 = b_2, G = 2, T_1 = 1, T_2 = t_2, T_3 = 0) \\ \zeta_2 &= (R_1 = 1, R_2 = r_2, A_1 = 1, A_2 = a_2, B_1 = 0, B_2 = b_2, G = 2, T_1 = 1, T_2 = t_2, T_3 = 0) \\ \zeta_3 &= (R_1 = 0, R_2 = r_2, A_1 = 0, A_2 = a_2, B_1 = 0, B_2 = b_2, G = 2, T_1 = 0, T_2 = t_2 + t_3, T_3 = 0) \end{aligned}$$

with probabilities evaluated as functions of the parameters $T_1 = 1, T_2 = t_2, T_3 = 0$.

Class \mathcal{Z}' Definition of the states belonging to this class.

$$\mathcal{Z}' = \{\zeta' : G = 2, R_2 = 1, B_2 = N, A_2 = N\} \quad (4.37)$$

The generic state $\zeta' \in \mathcal{Z}'$ can be represented as:

$$\zeta' = (R_1 = r_1, R_2 = 1, A_1 = a_1, A_2 = N, B_1 = b_1, B_2 = N, G = 2, T_1 = t_1, T_2 = t_2, T_3 = t_3) \quad (4.38)$$

The output action is always " S_2 Transmits new FRAG of S_2 ". The state transitions are symmetric to those of class \mathcal{Z} and the probabilities evaluated as functions of the parameters $T_1 = t_1, T_2 = 1, T_3 = 0$.

4.3.5 Stochastic Transition Matrix

In order to evaluate the stochastic transition probability matrix, the all possible states are generated and classified algorithmically, with the help of matlab. After the classification step, we have to eliminate the states that are not visited (i.e. states that do not exist). Please notice, that the number of states depends on the number of transmissions, which in consequence, leads to a very large amount of states when the number of transmissions, N is increased from 1 to say, 2. In other words, for the $N = 1$ case the finite state machine has 2304 states, which are then reduced to 84 after elimination of unvisited states. For $N = 2$ case we have initially 48600 possible states, which then reduces to 17638 states. Furthermore, in order to evaluate the steady state vector for such a finite state machine, one would need to compute the eigenvector decomposition of the stochastic matrix with dimensions 17638×17638 , which is quite a long computational procedure, and obviously, as the number of transmissions becomes larger, it becomes nearly impossible and too time consuming to obtain the theoretical values of the protocol performance.

For this reason, in this chapter of the thesis, the results are obtained via simulations.

4.3.6 Transmitting FSM for DCF Protocol

We distinguish between two different FSMs: one for performance evaluation of the protocols as we did in the previous section; and another to describe the transmission procedure. This latter FSM is derived with the help of the block diagram by following its branches and conditional statements.

The state definition of this FSM is given by the node that transmits and what it transmits: i.e. source 1 transmits its data; relay transmits XOR of the two FRAGs. This FSM can be derived using the block diagram given in Figure 4.6. By following the flowchart's blocks, one can easily derive the transitions between the states.

In order to simplify the notation of the transitions between the states and not use all the variables used in the block diagram, we define binary variables, as shown below:

$$\begin{aligned}
 \xi &\in \{0;1\} \\
 \sigma_1 &\in \{0;1\} \\
 \sigma_2 &\in \{0;1\} \\
 p_1 &\in \{0;1\} \\
 p_2 &\in \{0;1\} \\
 G &\in \{1;2\}
 \end{aligned} \tag{4.39}$$

where the term ξ takes values as follows:

$$\xi = \begin{cases} 1, & \text{if } B_1 < N_S, B_2 < N_S, R_1 = 1, R_2 = 1 \\ 0, & \text{otherwise} \end{cases} \tag{4.40}$$

The terms σ_i take their values based on the counters A_i defined in the protocol and used in the block diagram in Figure 4.6.

$$\sigma_i = \begin{cases} 1, & \text{if } A_i < N_S \\ 0, & \text{otherwise} \end{cases} \quad (4.41)$$

In a similar manner, the terms p_i take their values:

$$p_i = \begin{cases} 1, & \text{if } B_i < N_S \text{ and } R_i = 1 \\ 0, & \text{otherwise} \end{cases} \quad (4.42)$$

The graphical representation of the transmitting FSM is given in Figure 4.7.

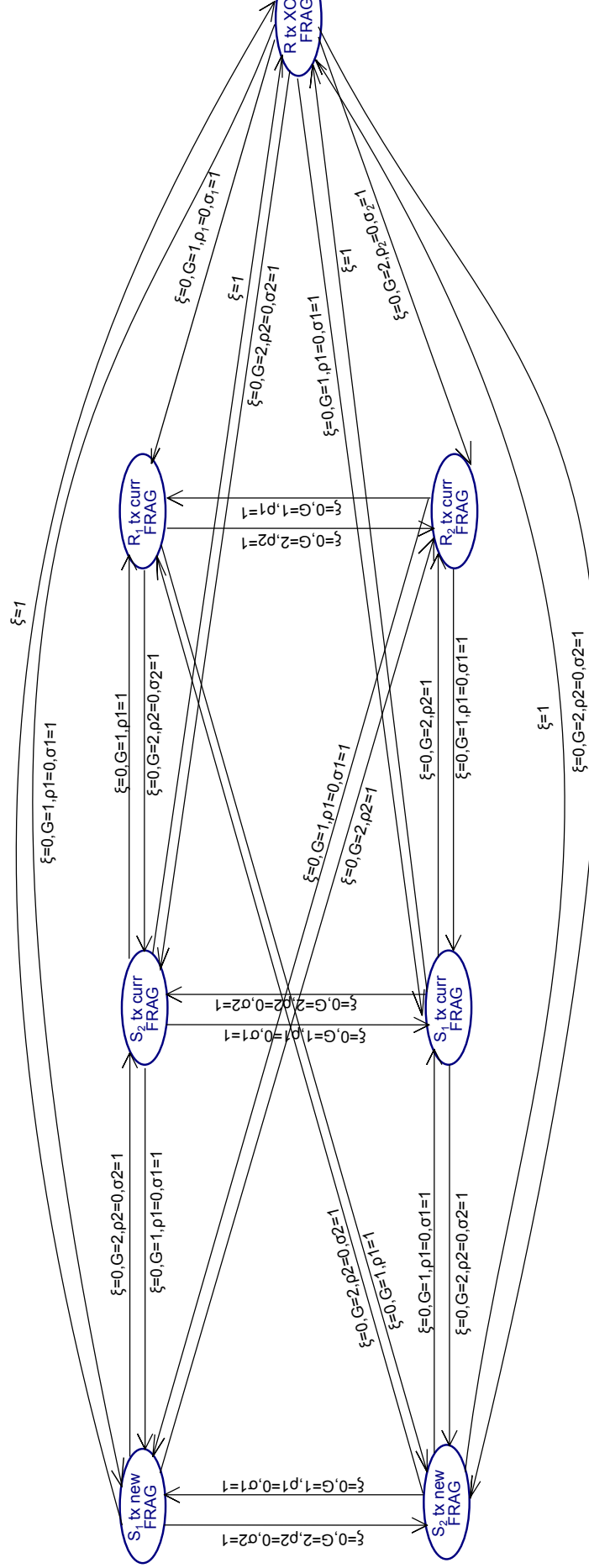


Figure 4.7: Transmitting FSM for the deterministic DCF Protocol

4.4 Two sources, one relay, one destination scheme, DMF

This section aims at defining and describing the DMF protocol, and its behavior for the scheme with two sources, one relay and a destination. We first give the set of rules and priorities in a brief format, and then extend and explain in more details.

4.5 Communication protocol definition

1. *Relay mode* - DMF: the relay sends network coded packets if both are requested, and direct copy in other cases.
2. *Relay decoder* - does not perform combining and only stores the last received copy of the FRAG.
3. *Decoders of sources at the destination* - perform combining of all pertinent messages received in the past.
4. *Number of retransmissions* - Each source is allowed to transmit the FRAG N times. The relay is allowed to transmit the FRAG of each source i N times.
5. *Priority to retransmit* - the relay is given priority if it has data, otherwise the respective source who transmitted less
6. *Retransmitted data* - the relay sends XOR if both sources are KO, otherwise the direct copy of the FRAG is sent from the device granted the priority
7. *New generation* - is sent as soon as the FRAG is correctly received or the number of retransmissions is expired.

Now let us see some of the rules in more details. The first several rules are quite straightforward, and do not need much explanations, whereas the others have to be detailed more. We also explain which counters are needed in order to implement this protocol.

1. The relay operates in Demodulate-and-Forward mode by XORing all the FRAGs that it receives. In other words, it only demodulates, XORs (if applicable), re-modulates and sends the data.
2. The relay is assumed to keep only the last received FRAG of each source.
3. The source decoder performs combining of all pertinent messages received in the past.
4. As soon as a new FRAG is generated at each source, the transmission counters at the source and the relay are initialized: $A_i \in \{0, 1, \dots, N\}$ - for the source; and $B_i \in \{0, 1, \dots, N\}$ - for the relay. Each node in the network is allowed to transmit the FRAG N times.

5. The relay is always given the priority to retransmit if it has the data and retransmissions credit. This choice is made because the relay has smaller distance to the destination. In order to know whether or not the relay can transmit, we define a variable $R_i \in [0, 1]$ that takes the values based on:

$$R_i = 1 \rightarrow \begin{cases} \text{if } S_i \rightarrow D \text{ is KO AND} \\ \text{R received a copy of } S_i \text{ FRAG AND} \\ B_i < N \end{cases}$$

As soon as a transmission from the relay happens, the counters B_1 and/or B_2 are incremented (when the relay sends the network coded version of the FRAGs, both counters are decremented).

If the relay cannot transmit the network coded version of the two FRAGs, then we check which FRAG has been transmitted the least. To keep the track of this information, we initialize general transmission counters to $G_{S_1} = 0$ and $G_{S_2} = 0$ in the beginning of a new generation FRAG transmission. These counters are incremented as soon as a FRAG, associated to the corresponding source, is emitted. If $G_i \leq G_j$, then transmission right is given to a copy of the active fragment of source i :

- from the relay if it has the active FRAG copy and the counter $B_i < N$
 - from the source if the relay cannot transmit
6. If both FRAGs of S_1 and S_2 need to be retransmitted, then the relay sends the XOR. In this case both transmission counters B_1 and B_2 are incremented. In all the other cases, the device granted transmission right sends a direct copy of the active fragment.
 7. A new generation fragment at source i is buffered as soon as the current fragment is correctly decoded, or as soon as both the source and the relay have expired the maximum number of allowed transmissions.

4.6 System model and decoder design

This section aims at describing the decoder for S_1 and S_2 when the relay works in the DMF regime, i.e. there can be error propagation issue. For this reason, let us consider the sequence of following vector has been received at the destination after several instances of the communication: $[\bar{y}_{S_1D}, \bar{y}_{S_2D}, \bar{y}_{S_1RD}, \bar{y}_{S_2RD}, \bar{y}_{RD_{XOR}}]$.

Please notice the difference: in contrast to the DCF protocol we have to separate the vectors that represent the symbols vector, received at the destination and coming from S_i , from R (S_i 's data), from R (the XOR). This happens because the relay can send erroneous data, and this has to be taken into account during the decoding. So, we will have five vectors in total. Their notations are given below.

Notations

\bar{y}_{S_1D} represents the vector of the x_1 symbol observations at the destination that results from transmission from the $S_1 \rightarrow D$ link

\bar{y}_{S_2D} represents the vector of the x_2 symbol observations at the destination that results from transmission from the $S_2 \rightarrow D$ link

\bar{y}_{S_1RD} represents the vector of the x_1 symbol observations at the destination that results from an indirect transmission through the relay link, $S_1 \rightarrow R \rightarrow D$

\bar{y}_{S_2RD} represents the vector of the x_2 symbol observations at the destination that results from an indirect transmission through the relay link, $S_2 \rightarrow R \rightarrow D$

$\bar{y}_{RD_{XOR}}$ represents the vector of the $x_1 \oplus x_2$ symbol observations at the destination that results from an indirect transmission through the relay link, $R \rightarrow D$

At the timeslot T , the direct observations received at the destination from source i , are given by :

$$y_{S_iD} = x_{S_i}h_{S_iD} + n_{S_iD} \quad (4.43)$$

The in the meantime, the relay listens to the source transmission, and receives the following symbols:

$$y_{S_iR} = x_{S_i}h_{S_iR} + n_{S_iR} \quad (4.44)$$

Furthermore, the relay performs an ML-demodulation on the symbols that it received:

$$\hat{x}_{S_iR} = \underset{b \in \{0,1\}}{\operatorname{argmin}} \{ ||y_{S_iR} - (1 - 2b)h_{S_iR}||^2 \} \quad (4.45)$$

Then these detected symbols are modulated, according to the BPSK mapping and sent to the destination in the dedicated timeslot.

$$y_{S_iRD} = (1 - 2\hat{x}_{S_iR})h_{S_iRD} + n_{S_iRD} \quad (4.46)$$

In some cases, which will be detailed later on, the relay sends the XOR of the two observations:

$$w = (1 - 2[\hat{x}_{S_1R} \oplus \hat{x}_{S_2R}])h_{RD_{XOR}} + n_{RD_{XOR}} \quad (4.47)$$

In the DMF protocol the relay does not know whether the data that it received is correct or not. Now let us define the following random variables:

$$a = \begin{cases} 0, & \text{if } R \text{ did not demodulate correctly the FRAG of } S_1 \\ 1, & \text{if } R \text{ demodulated correctly the FRAG of } S_1 \end{cases}$$

$$b = \begin{cases} 0, & \text{if } R \text{ did not demodulate correctly the FRAG of } S_2 \\ 1, & \text{if } R \text{ demodulated correctly the FRAG of } S_2 \end{cases}$$

The LLR of the i -th source S_i decoder at the destination is given by:

$$\text{LLR}_{S_1} = \log \frac{p(\bar{y}_{S_1D}, \bar{y}_{S_2D}, \bar{y}_{S_1RD}, \bar{y}_{S_2RD}, \bar{w} | x_i = 1)}{p(\bar{y}_{S_1D}, \bar{y}_{S_2D}, \bar{y}_{S_1RD}, \bar{y}_{S_2RD}, \bar{w} | x_i = -1)} \quad (4.48)$$

Detailed derivation of the explicit form of (4.48) is provided in Section B.2 in Appendix B.

4.7 Functional representation of the protocol

The block diagram for the DMF protocol is given in Figure 4.8, which is quite similar to that of the DCF protocol.

4.8 FSMC description

Operating the substitution on G_1 and G_2 with the binary variable G , the system variables vector takes form $V = [R_1; R_2; A_1; A_2; B_1; B_2; G; T_1; T_2; T_3; T_4; T_5]$. We have $|\mathcal{V}| = 2^2 \cdot (N+1)^4 \cdot 2 \cdot (N)^2 \cdot (2N)^2 \cdot (N) = 8 \cdot (N+1)^4 \cdot 4N^5$. The classes of the FSM are defined as follows:

Class \mathcal{A} Definition of the states belonging to this class.

$$\mathcal{A} = \{\alpha : R_1 = 1, R_2 = 1, B_1 < N, B_2 < N\} \quad (4.49)$$

As one could see by investigating Figure 4.8, for all the states in \mathcal{A} , the output is "Relay Transmits XORed FRAG". The generic state $\alpha \in \mathcal{A}$ can be represented as (for convenience, let $\mathbf{R} = (R_1, R_2)$; $\mathbf{A} = (A_1, A_2)$; $\mathbf{B} = (B_1, B_2)$; $\mathbf{T} = (T_1, T_2, \dots, T_5)$):

$$\alpha = (\mathbf{R} = (1, 1), \mathbf{A} = (a_1, a_2), \mathbf{B} = (b_1, b_2), G = g, \mathbf{T} = (t_1, t_2, t_3, t_4, t_5)) \quad (4.50)$$

There are four possible transitions, depending on the outputs of the decoder block, 4.4. Consequently, there are four output states.

$$\begin{aligned} \alpha_1 &= (\mathbf{R} = (1, 1), \mathbf{A} = (a_1, a_2), \mathbf{B} = (b_1 + 1, b_2 + 1), G = g, \mathbf{T} = (t_1, t_2, t_3, t_4, t_5 + 1)) \\ \alpha_2 &= (\mathbf{R} = (0, 1), \mathbf{A} = (0, a_2), \mathbf{B} = (0, b_2 + 1), G = g, \mathbf{T} = (0, t_2 + t_5, 0, t_4, 0)) \\ \alpha_3 &= (\mathbf{R} = (1, 0), \mathbf{A} = (a_1, 0), \mathbf{B} = (b_1 + 1, 0), G = g, \mathbf{T} = (t_1 + t_5, 0, 0, t_4, 0)) \\ \alpha_4 &= (\mathbf{R} = (0, 0), \mathbf{A} = (0, 0), \mathbf{B} = (0, 0), G = g, \mathbf{T} = (0, 0, 0, 0, 0)) \end{aligned}$$

with probabilities P_1, P_2, P_3, P_4 , respectively, which can be evaluated as functions of $T_1 = t_1, T_2 = t_2, T_3 = t_3, T_4 = t_4, T_5 = t_5 + 1$.

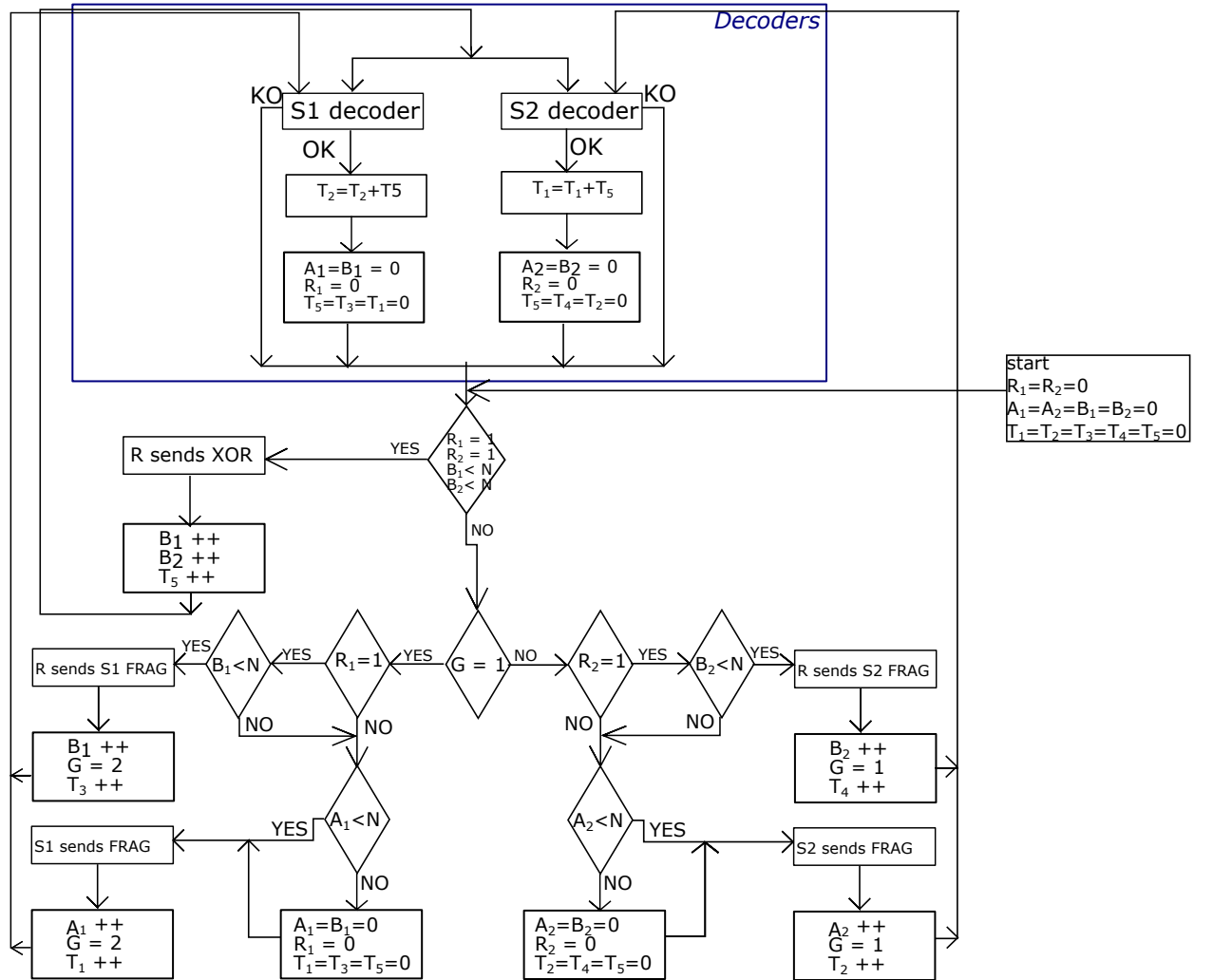


Figure 4.8: Algorithm of the 2 Source 1 Relay 1 Destination Scheme DMF Protocol

Class \mathcal{B} Definition of the states belonging to this class.

$$\mathcal{B} = \{\beta : \beta \notin \mathcal{A}, G = 1, R_1 = 1, B_1 < N\} \quad (4.51)$$

The generic state $\beta \in \mathcal{B}$ can be represented as:

$$\beta = (\mathbf{R} = (1, r_2), \mathbf{A} = (a_1, a_2), \mathbf{B} = (b_1, b_2), G = 1, \mathbf{T} = (t_1, t_2, t_3, t_4, t_5 + 1)) \quad (4.52)$$

The output action is always "Relay Transmits FRAG of S_1 ". There are two possible transitions, depending on the outputs of the decoder block, 4.4. Consequently, there are two output states:

$$\begin{aligned} \beta_1 &= (\mathbf{R} = (1, r_2), \mathbf{A} = (a_1, a_2), \mathbf{B} = (b_1 + 1, b_2), G = 2, \mathbf{T} = (t_1, t_2, t_3 + 1, t_4, t_5)) \\ \beta_2 &= (\mathbf{R} = (0, r_2), \mathbf{A} = (0, a_2), \mathbf{B} = (0, b_2), G = 2, \mathbf{T} = (0, t_2 + t_5, 0, t_4, 0)) \end{aligned}$$

where the transition probabilities are evaluated as functions of $T_1 = t_1 + 1$, $T_2 = t_2$, $T_3 = t_3$, $T_4 = t_4$, $T_5 = t_5 + 1$.

Class \mathcal{B}' Definition of the states belonging to this class.

$$\mathcal{B}' = \{\beta' : \beta' \notin \mathcal{A}, G = 2, R_2 = 1, B_2 < N\} \quad (4.53)$$

The generic state $\beta' \in \mathcal{B}'$ can be represented as:

$$\beta' = (\mathbf{R} = (r_1, 1), \mathbf{A} = (a_1, a_2), \mathbf{B} = (b_1, b_2), G = 1, \mathbf{T} = (t_1, t_2, t_3, t_4 + 1, t_5)) \quad (4.54)$$

The output is always "Relay Transmits FRAG of S_2 ", and the state transitions are symmetric to those of class \mathcal{B} .

Class \mathcal{C} Definition of the states belonging to this class.

$$\mathcal{C} = \{\gamma : \gamma \notin \mathcal{A}, G = 1, R_1 = 0, A_1 < N\} \quad (4.55)$$

which corresponds to the green route in Figure 4.6. The generic state $\gamma \in \mathcal{C}$ can be represented as:

$$\gamma = (\mathbf{R} = (0, r_2), \mathbf{A} = (a_1, a_2), \mathbf{B} = (b_1, b_2), G = 1, \mathbf{T} = (t_1, t_2, t_3, t_4, t_5)) \quad (4.56)$$

The output action is always " S_1 Transmits FRAG of S_1 ".

There are three possible transitions, depending on the outputs of the decoder block, 4.4. Consequently, there are three output states:

$$\begin{aligned} \gamma_1 &= (\mathbf{R} = (0, r_2), \mathbf{A} = (a_1 + 1, a_2), \mathbf{B} = (b_1, b_2), G = 2, \mathbf{T} = (t_1 + 1, t_2, t_3, t_4, t_5)) \\ \gamma_2 &= (\mathbf{R} = (1, r_2), \mathbf{A} = (a_1 + 1, a_2), \mathbf{B} = (b_1, b_2), G = 2, \mathbf{T} = (t_1 + 1, t_2, t_3, t_4, t_5)) \\ \gamma_3 &= (\mathbf{R} = (0, r_2), \mathbf{A} = (0, a_2), \mathbf{B} = (0, b_2), G = 2, \mathbf{T} = (0, t_2 + t_5, 0, t_4, 0)) \end{aligned}$$

where the transition probabilities are evaluated as functions of $T_1 = t_1 + 1$, $T_2 = t_2$, $T_3 = t_3$, $T_4 = t_4$, and $T_5 = t_5$.

Class \mathcal{C}' Definition of the states belonging to this class.

$$\mathcal{C}' = \{\gamma' : \gamma' \notin \mathcal{A}, \gamma \notin \mathcal{B}, G = 2, R_2 = 0, A_2 < N\} \quad (4.57)$$

The generic state $\gamma \in \mathcal{C}$ can be represented as:

$$\gamma' = (\mathbf{R} = (r_1, 0), \mathbf{A} = (a_1, a_2), \mathbf{B} = (b_1, b_2), G = 2, \mathbf{T} = (t_1, t_2, t_3, t_4, t_5)) \quad (4.58)$$

The output action is always "S₂ Transmits FRAG of S₂", and the transition probabilities are symmetric to those of class \mathcal{C} , and are evaluated as functions of $T_1 = t_1$, $T_2 = t_2 + 1$, and $T_3 = t_3$.

Class \mathcal{D} Definition of the states belonging to this class.

$$\mathcal{D} = \{\delta : \delta \notin \mathcal{A}, G = 1, R_1 = 1, B_1 = N, A_1 < N\} \quad (4.59)$$

The generic state $\delta \in \mathcal{D}$ can be represented as:

$$\delta = (\mathbf{R} = (1, r_2), \mathbf{A} = (a_1, a_2), \mathbf{B} = (N, b_2), G = 1, \mathbf{T} = (t_1, t_2, t_3, t_4, t_5)) \quad (4.60)$$

The output action is always "S₁ Transmits FRAG of S₁".

There are two possible transitions, depending on the outputs of the decoder block, 4.4. Consequently, there are two output states:

$$\begin{aligned} \delta_1 &= (\mathbf{R} = (0, r_2), \mathbf{A} = (a_1 + 1, a_2), \mathbf{B} = (b_1, b_2), G = 2, \mathbf{T} = (t_1 + 1, t_2, t_3, t_4, t_5)) \\ \delta_2 &= (\mathbf{R} = (0, r_2), \mathbf{A} = (0, a_2), \mathbf{B} = (0, b_2), G = 2, \mathbf{T} = (0, t_2 + t_5, t_3, t_4, 0)) \end{aligned}$$

Class \mathcal{D}' Definition of the states belonging to this class.

$$\mathcal{D}' = \{\delta' : \delta' \notin \mathcal{A}, \delta \notin \mathcal{B}, \delta \notin \mathcal{C}, G = 2, R_2 = 1, B_2 = N, A_2 < N\} \quad (4.61)$$

The generic state $\delta \in \mathcal{D}'$ can be represented as:

$$\delta' = (\mathbf{R} = (r_1, r_2), \mathbf{A} = (a_1, a_2), \mathbf{B} = (b_1, N), G = 2, \mathbf{T} = (t_1, t_2, t_3, t_4, t_5)) \quad (4.62)$$

The output action is always "S₂ Transmits FRAG of S₂". The state transitions are symmetrical to those of class \mathcal{D} .

Class \mathcal{E} Definition of the states belonging to this class.

$$\mathcal{E} = \{\epsilon : \epsilon \notin \mathcal{A}, G = 1, R_1 = 0, A_1 = N\} \quad (4.63)$$

The generic state $\epsilon \in \mathcal{E}$ can be represented as:

$$\epsilon = (\mathbf{R} = (0, r_2), \mathbf{A} = (N, a_2), \mathbf{B} = (b_1, b_2), G = 1, \mathbf{T} = (t_1, t_2, t_3, t_4, t_5)) \quad (4.64)$$

The output action is always "S₁ Transmits new FRAG of S₁".

There are three possible transitions, depending on the outputs of the decoder block. Consequently, there are three output states:

$$\begin{aligned}\epsilon_1 &= (\mathbf{R} = (0, r_2), \mathbf{A} = (1, a_2), \mathbf{B} = (b_1, b_2), G = 2, \mathbf{T} = (1, t_2, t_3, t_4, 0)) \\ \epsilon_2 &= (\mathbf{R} = (1, r_2), \mathbf{A} = (1, a_2), \mathbf{B} = (0, b_2), G = 2, \mathbf{T} = (1, t_2, t_3, t_4, 0)) \\ \epsilon_3 &= (\mathbf{R} = (0, r_2), \mathbf{A} = (0, a_2), \mathbf{B} = (0, b_2), G = 2, \mathbf{T} = (0, t_2 + t_5, 0, t_4, 0))\end{aligned}$$

with probabilities evaluated as functions of the parameters $T_1 = 1$, $T_2 = t_2$, $T_3 = 0$, $T_4 = t_4$, $T_5 = t_5$.

Class \mathcal{E}' Definition of the states belonging to this class.

$$\mathcal{E}' = \{\epsilon' : G = 2, R_2 = 0, A_2 = N\} \quad (4.65)$$

The generic state $\epsilon \in \mathcal{E}$ can be represented as:

$$\epsilon' = (\mathbf{R} = (r_1, 0), \mathbf{A} = (a_1, N), \mathbf{B} = (b_1, b_2), G = 2, \mathbf{T} = (t_1, t_2, t_3, t_4, t_5)) \quad (4.66)$$

The output action is always "S₂ Transmits new FRAG of S₂". The state transitions are symmetric to those of class \mathcal{E} and the transition probabilities evaluated as functions of the parameters $T_1 = t_1$, $T_2 = 1$, $T_3 = 0$, $T_4 = t_4$, $T_5 = t_5$.

Class \mathcal{Z} Definition of the states belonging to this class.

$$\mathcal{Z} = \{\zeta : \zeta \notin \mathcal{A}, G = 1, R_1 = 1, B_1 = N, A_1 = N\} \quad (4.67)$$

which corresponds to the sky blue route in Figure 4.6. The generic state $\zeta \in \mathcal{Z}$ can be represented as:

$$\zeta = (\mathbf{R} = (1, r_2), \mathbf{A} = (N, a_2), \mathbf{B} = (N, b_2), G = 1, \mathbf{T} = (t_1, t_2, t_3, t_4, t_5)) \quad (4.68)$$

The output action is always "S₁ Transmits new FRAG of S₁".

There are three possible transitions, depending on the outputs of the decoder block, 4.4. Consequently, there are three output states:

$$\begin{aligned}\zeta_1 &= (\mathbf{R} = (0, r_2), \mathbf{A} = (1, a_2), \mathbf{B} = (0, b_2), G = 2, \mathbf{T} = (1, t_2, t_3, t_4, 0)) \\ \zeta_2 &= (\mathbf{R} = (1, r_2), \mathbf{A} = (1, a_2), \mathbf{B} = (0, b_2), G = 2, \mathbf{T} = (1, t_2, t_3, t_4, 0)) \\ \zeta_3 &= (\mathbf{R} = (0, r_2), \mathbf{A} = (0, a_2), \mathbf{B} = (0, b_2), G = 2, \mathbf{T} = (0, t_2 + t_5, 0, t_4, 0))\end{aligned}$$

with probabilities evaluated as functions of the parameters $T_1 = 1$, $T_2 = t_2$, $T_3 = 0$, $T_4 = t_4$, $T_5 = t_5$.

Class \mathcal{Z}' Definition of the states belonging to this class.

$$\mathcal{Z}' = \{\zeta' : G = 2, R_2 = 1, B_2 = N, A_2 = N\} \quad (4.69)$$

The generic state $\zeta' \in \mathcal{Z}'$ can be represented as:

$$\zeta = (\mathbf{R} = (r_1, 1), \mathbf{A} = (a_1, N), \mathbf{B} = (b_1, N), G = 2, \mathbf{T} = (t_1, t_2, t_3, t_4, t_5)) \quad (4.70)$$

The output action is always " S_2 Transmits new FRAG of S_2 ". The state transitions are symmetric to those of class \mathcal{Z} and the probabilities evaluated as functions of the parameters $T_1 = t_1, T_2 = 1, T_3 = 0, T_4 = t_4, T_5 = t_5$.

4.8.1 On the number of states in the FSM

In this part of the work we discuss the growth of the FSM states number when the number of nodes is increased. As we could see from the scheme with three nodes, the number of variables and the values that each variable takes have a direct influence on the number of states in the FSM. As soon as we increase the number of states, the number of variables in a states grew from 3 to 10 for the DCF protocol. This growth is explained by the fact, that the state needs to have more information in it.

Below are summarized the number of states for the 2 source 1 relay and 1 destination scheme.

1. $N_S = N_R = 1$ number of states is 84
2. $N_S = N_R = 2$ number of states is 17638

In order to evaluate the number of states in the FSM for larger numbers of N_S and N_R , we would have to run time consuming simulations in order to eliminate the non-existent states, that originate from randomly listing all the possible counter values. Since only for $N_R = 2$ the number is quite large, we conclude that using FSM for the cooperative protocol design with this definition of the states although possible but is not practical. For this reason in the remainder of this work the performance metrics are evaluated by simulations.

4.9 Numerical Results

In this section we introduce the simulated results for the DCF and DMF protocols. Before doing that, we briefly introduce the configurations of the simulation setup.

Each source generates independent frames (PDUs) that consist of $L = 120$ information bits. These frames are then encoded to FRAGs that consist of L/R_c coded bits, that are then mapped according to BPSK constellation and emitted in orthogonal timeslots.

The channel codes are convolutional codes with rate $R_c = 1/2$. It is assumed that the feedback messages are perfectly known at each node and are instantaneously received.

The channel is Rayleigh fully interleaved with addition of white Gaussian noise, whose coefficients are i.i.d.

The number of transmissions per node that we consider in this setting, is $N_S = N_R \in \{1, 2, 3\}$.

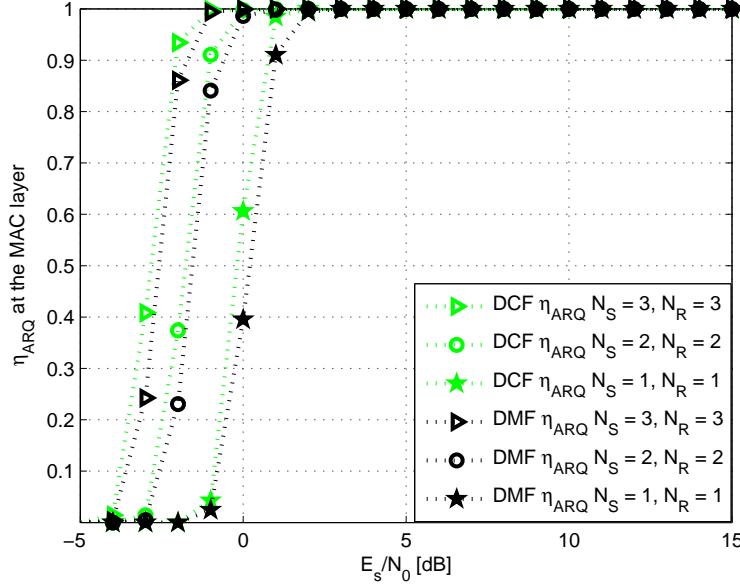


Figure 4.9: DCF and DMF protocol ARQ-efficiency, η_{ARQ} , comparison at the MAC layer as a function of E_b/N_0 , 2-s-1-r-1-d

4.9.1 Efficiency

We start the comparisons of the protocols by plotting the efficiencies: η_{ARQ} and η_{gen} in Figures 4.9 and 4.10, respectively.

We notice that in both cases the gap between the two protocols is almost negligible, on the range of 0.1 – 0.2 dB, with DMF losing to DCF. This is explained by the fact, that in DMF protocol erroneous FRAGs can propagate to the destination although the decoder tries to correct them.

4.9.2 FER

In this part of the results section we compare the FER of the DCF and DMF protocols for various number of transmissions, with combining at the destination (please note that in the remainder of this work all the addressed curves represent the ones with combining at the destination, unless otherwise stated). The aims of these comparisons are two-fold: a) provide an insight on the system behavior as the number of transmissions is increased; b) compare the DCF and DMF protocols.

Please be reminded that the relay in all the curves shown in this section performs XOR operation according to the protocol rules, unless otherwise stated. Please also notice that since the system is symmetric and we obtain the same curves for both S_1 and S_2 , we always plot only one curve to save some space and make the curves readable.

In Figures 4.11 and 4.12 the FER of the DCF and DMF protocols is compared as a function of E_b/N_0 and E_s/N_0 , respectively. Interestingly, the first thing that strikes the

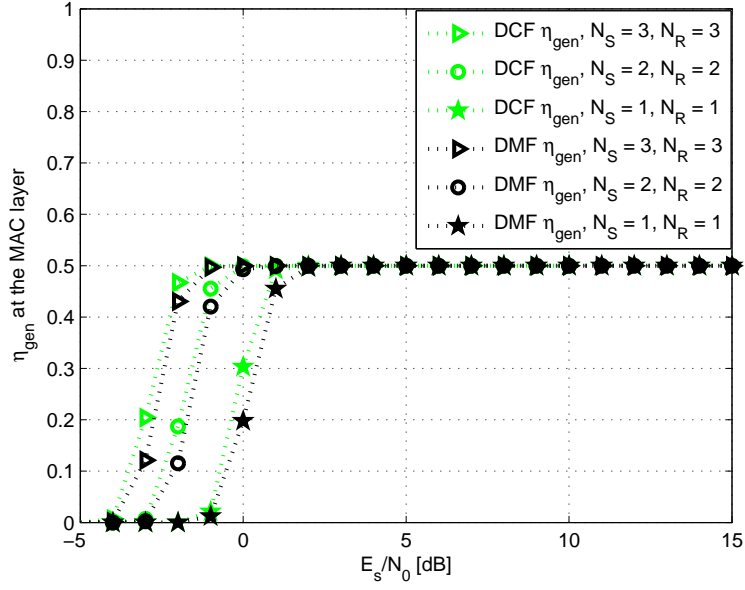


Figure 4.10: DCF and DMF protocol general efficiency, η_{gen} , comparison at the MAC layer as a function of E_b/N_0 , 2-s-1-r-1-d

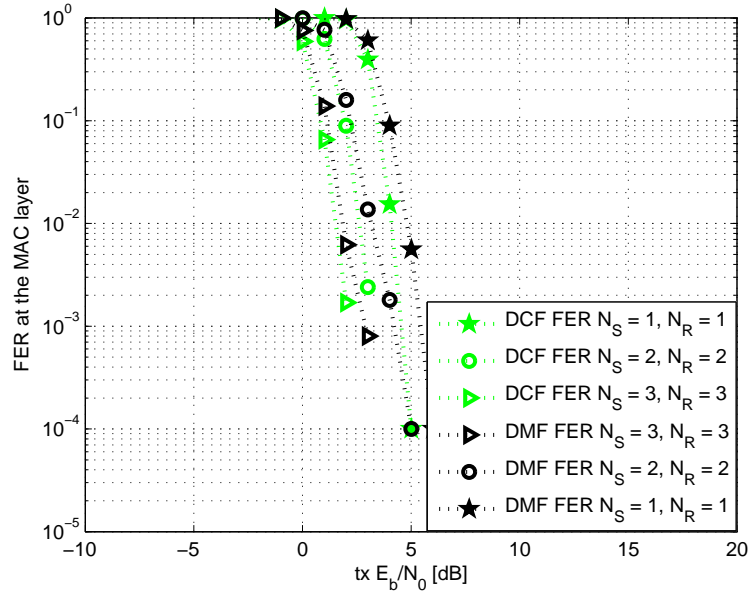


Figure 4.11: DCF and DMF protocol FER comparison at the MAC layer as a function of E_b/N_0 , 2-s-1-r-1-d

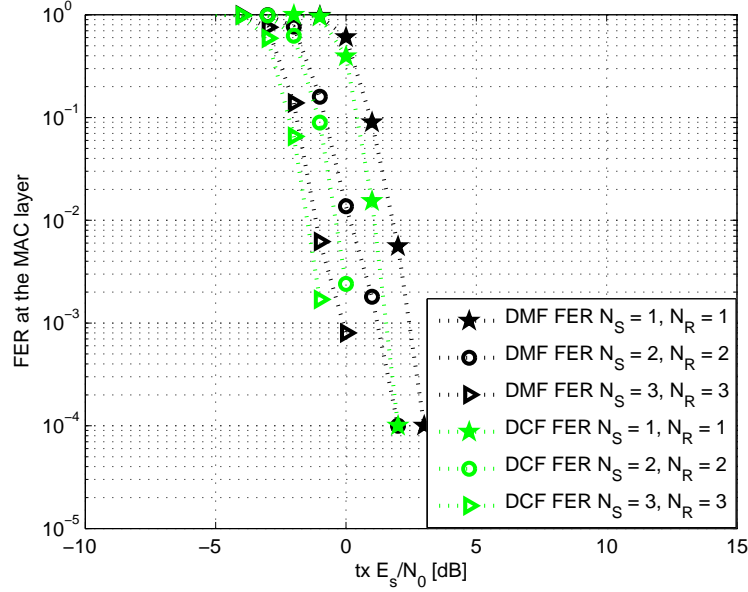


Figure 4.12: DCF and DMF protocol FER comparison at the MAC layer as a function of E_s/N_0 , 2-s-1-r-1-d

eye is that the DMF protocol for any number of N_S and N_R falls off less steeply than the DCF protocol. This is explained by the fact, that despite the decoder taking into account the possible errors at the source to relay links, the DMF protocol has more errors than the DCF. It can be seen that for both protocols, increasing the number of transmissions by 2, results in approximately 1 dB gain.

Let us now have a look at Figure 4.12 and compare the two protocols when the gap between them is the biggest. Take the DMF curve and compare it to the DCF with $N_S = 1, N_R = 1$ at $\text{FER} = 0.0001$: DCF protocol achieves this FER at $E_s/N_0 = 2$ dB, and DMF at $E_s/N_0 = 1$ dB. As we might remember from the section with one source, one relay and one destination, DMF protocol performed much worse, by having around 2 dB gap. Also, contrary to the three-hop network, we see that at low E_s/N_0 values the two protocols have roughly the same FER: for example, see the DCF and DMF FER with $N_S = 2, N_R = 2$ for the range of $E_s/N_0 \in [-5; -1]$ - what we see is that DMF in bad channel conditions achieves the same performance as DCF with the **same** number of transmissions as DCF. Whereas as you might recall from the one source, one relay and one destination section, DMF even if had similar performance to DCF, it was achieved with more retransmissions.

4.9.3 FER in an energetic fair context

Here we plot the FER of both protocols in an energetic fair context. In Figures 4.13 and 4.14 the X -axis represents the **true** E_b/N_0 and **true** E_s/N_0 , respectively.

Notice that the DMF protocol tends to have a more vivid C-shape as the channel

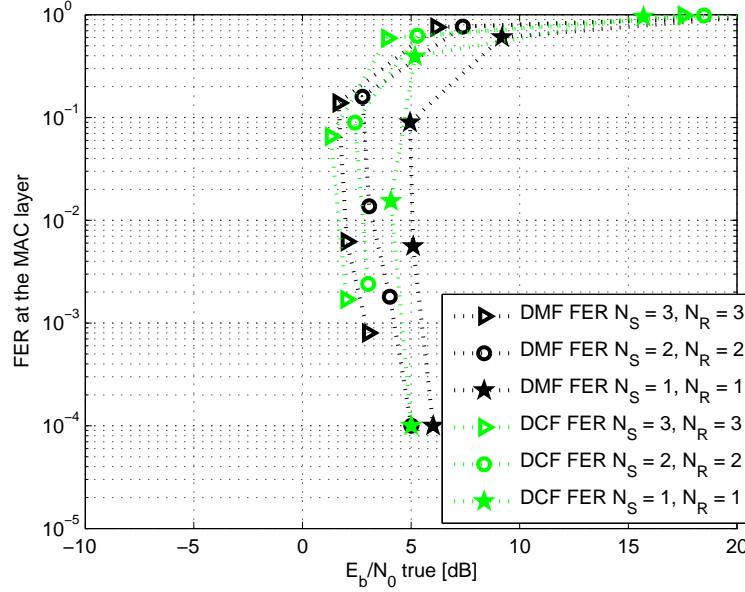


Figure 4.13: DCF and DMF protocol FER comparison at the MAC layer versus **true** E_b/N_0 , 2-s-1-r-1-d

quality increases, thus over-spending the energy per successful information bit. This happens because in the DMF protocol, the destination has more decoding errors (i.e. sends NACKs more frequently), and this results in waste of energy. In other words, each retransmission costs energy, and since there are more retransmissions in DMF than in DCF, the curves have more distinct C-shape.

Take a look at the maximal general efficacy point of DMF with $N_S = 2, N_R = 2$ in Figure 4.10, which is equal to 0.42 and obtained at $E_s/N_0 = -1$ dB. Then see the FER attained at this E_s/N_0 value: it is equal to 0.15, Figure 4.12. We then see that this value in Figure 4.14 corresponds to the **true** $E_s/N_0 = 2.76$ dB. This point shows the best possible energy expenditure. Making similar comparison for DCF with $N_S = 2, N_R = 2$, we conclude that the maximal efficacy point is achieved at **true** $E_s/N_0 = 2.417$ dB. This implies that when the true energy is considered, there is almost no difference between the two protocols.

4.9.4 Delay

Now let us focus on the delay comparisons. We could see that the FER of the two protocols improves each time that the number of transmissions is increased. Interestingly, from the delay curves we observe that for the DCF case the delay with $N_S = 2, N_R = 2$ and $N_S = 3, N_R = 3$ is almost the same. This is explained by the fact that on average both schemes spend the same amount of FRAGs to successfully receive one Figure 4.15.

By looking at Figures 4.15 and 4.16, it also becomes evident that the delay both schemes in the range of $E_b/N_0 \in [2.01; 5]$ dB (Figure 4.15) is practically equal for any

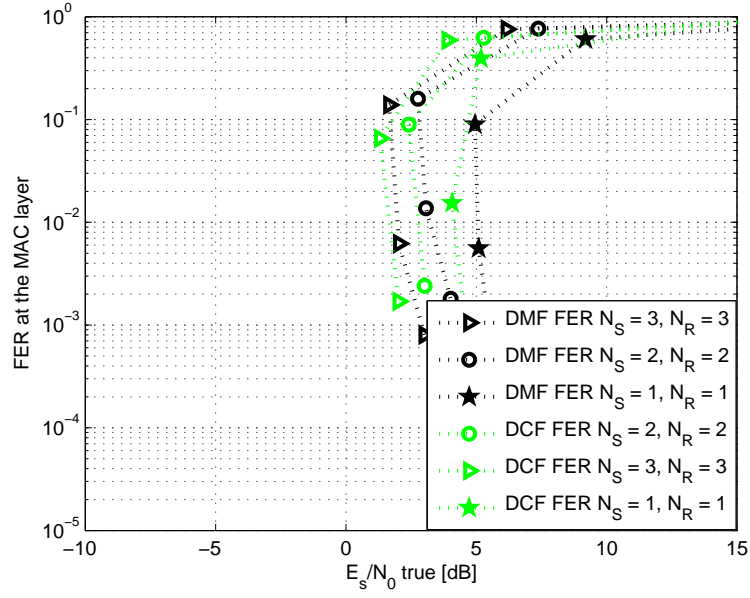


Figure 4.14: DCF and DMF protocol FER comparison at the MAC layer versus **true** E_s/N_0 , 2-s-1-r-1-d

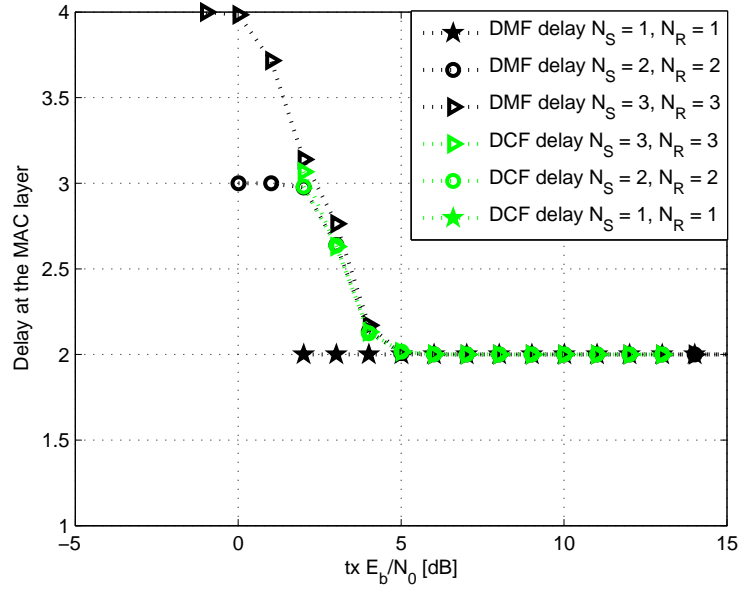


Figure 4.15: DCF and DMF protocol delay comparison at the MAC layer as a function of E_b/N_0 , 2-s-1-r-1-d

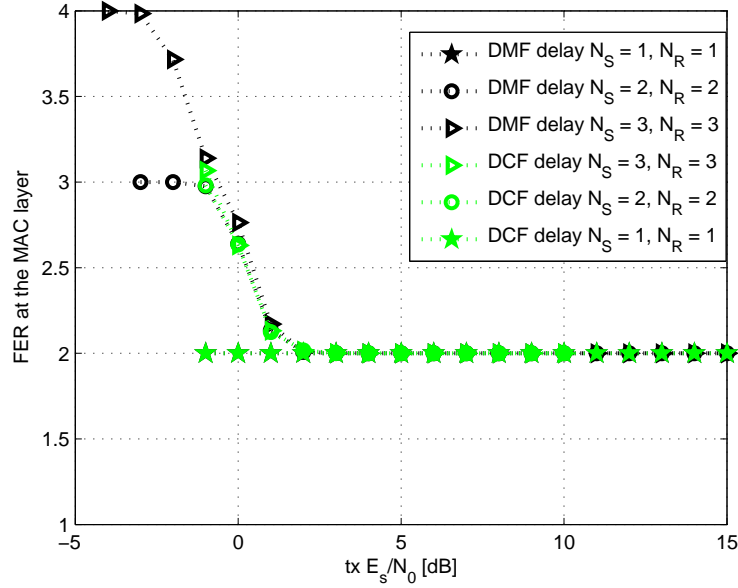


Figure 4.16: DCF and DMF protocol delay comparison at the MAC layer as a function of E_s/N_0 , 2-s-1-r-1-d

number of retransmissions.

For example, see the DCF and DMF delay in Figure 4.15, at $E_b/N_0 = 2.01$ dB is equal to 3.067 and to 3.1 for DCF and DMF, respectively.

4.9.5 Delay in an energetic fair context

When we consider the delay of both schemes in an energetic fair comparison, we see that the gap again is almost negligible in both protocols, for the same numbers of N_S and N_R .

By taking the DCF and DMF protocols maximal efficacy point for the $N_S = 2, N_R = 2$ configuration, it can be observed, that the delay of DCF at this point is equal to 2.98 and DMF to 2.9.

We can also see that both protocols, when the number of transmissions is increased to $N_S = 3, N_R = 3$, can have twice the delay of the schemes with $N_S = 2, N_R = 2$. This puts a constraint in terms of the delay: for the systems with lower level of tolerance to the delay we can increase the number of transmissions in order to attain better FER; but when the delay tolerance is very restricted, than we have to carefully choose the number of transmissions.

4.9.6 2-S-1-R-1-D versus 1-S-1-R-1-D

In order to compare the usage of binary network codes in this simple cooperative network to the case when the sources are treated separately (i.e. the relay sends direct observation to the destination), we consider the comparisons of the FER for the two cooperative relay

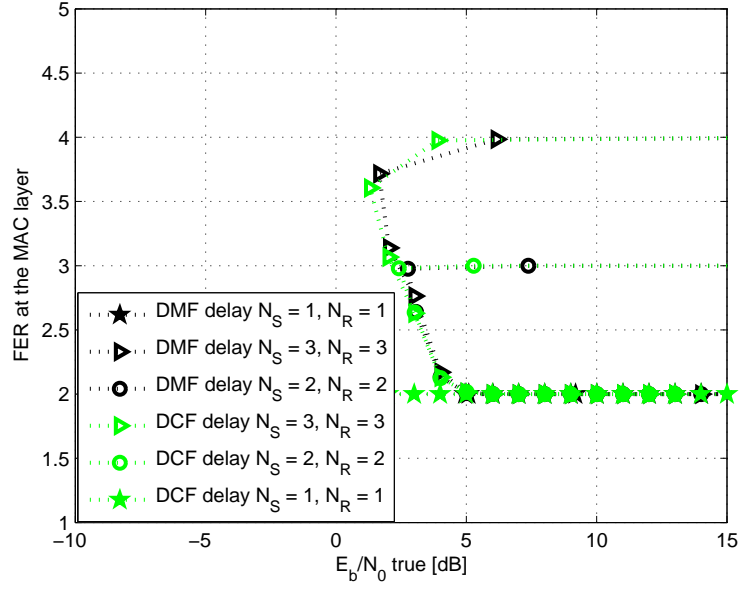


Figure 4.17: DCF and DMF protocol delay comparison at the MAC layer versus **true** E_b/N_0 , 2-s-1-r-1-d

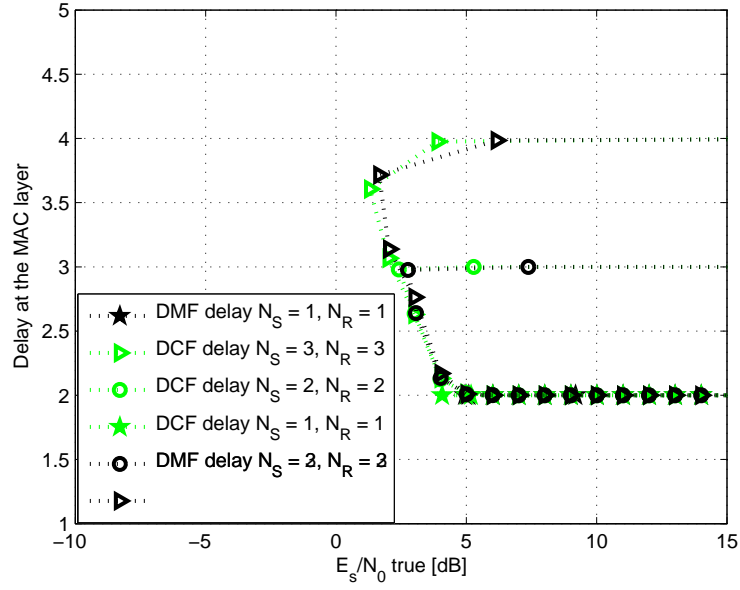


Figure 4.18: DCF and DMF protocol delay comparison at the MAC layer versus **true** E_s/N_0 , 2-s-1-r-1-d

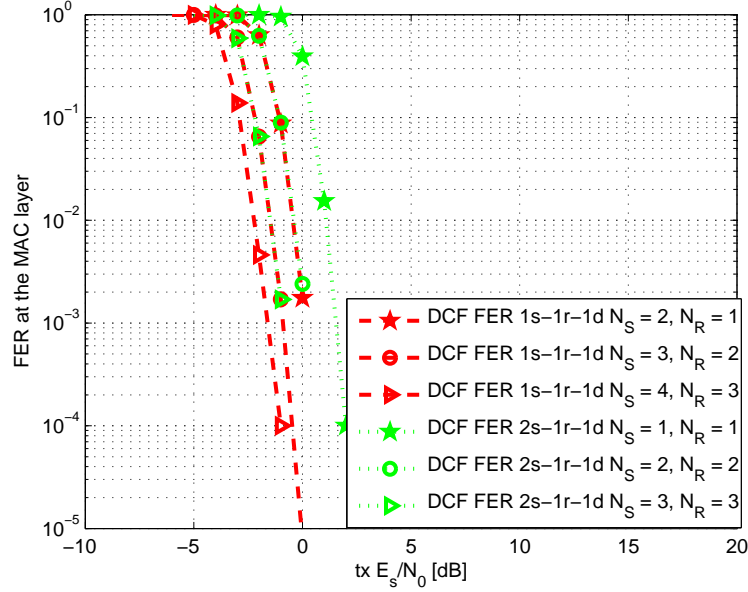


Figure 4.19: DCF protocol FER comparison for 2-S-1-R-1-D and 1-S-1-R-1-D networks at the MAC layer as a function of E_s/N_0

schemes studied in this thesis. First we discuss the FER, Figure 4.19. It can be observed that the FER of the three-hop scheme with $N_S = 2, N_R = 1$ is isomorph to that of four-hop scheme with $N_S = 2, N_R = 2$. This implies, that with less transmission per FRAG the three-node network attains the same FER. Similarly, this happens for the three-hop network with $N_S = 3, N_R = 2$ and four-hop network with $N_S = 3, N_R = 3$. According to this result, it is deduced that the FER of each source can be significantly improved with less retransmissions, if the relay simply forwards the data instead of XORing it.

Then by looking at Figure 4.20, it becomes clear that the delay in both schemes has the same values. As a result it turns out, that treating the data in single manner does not affect the delay.

Now let us finally compare these results in an energetic fair context for the FER, Figure 4.21.

Similarly, we see that when the network is treated as one-source-one-relay-one-destination, the same performance can be achieved with lower transmissions number.

So it can be concluded that XORing the data in such a simple cooperative network is not justified by any of the performance metrics.

4.10 Conclusions

This chapter had the following goals:

- I. study the protocols defined for the 1-s-1-r-1-d scheme for 2-s-1-r-1-d scheme

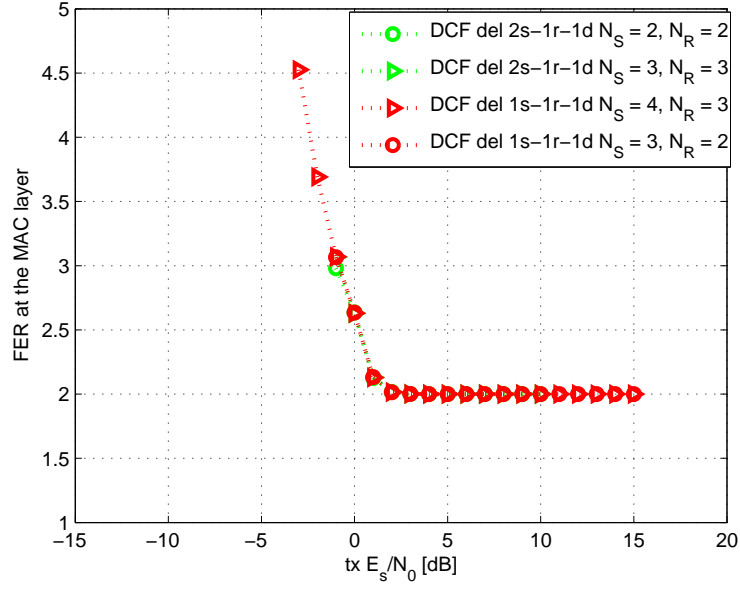


Figure 4.20: DCF protocol delay comparison for 2-S-1-R-1-D and 1-S-1-R-1-D networks at the MAC layer as a function of E_s/N_0

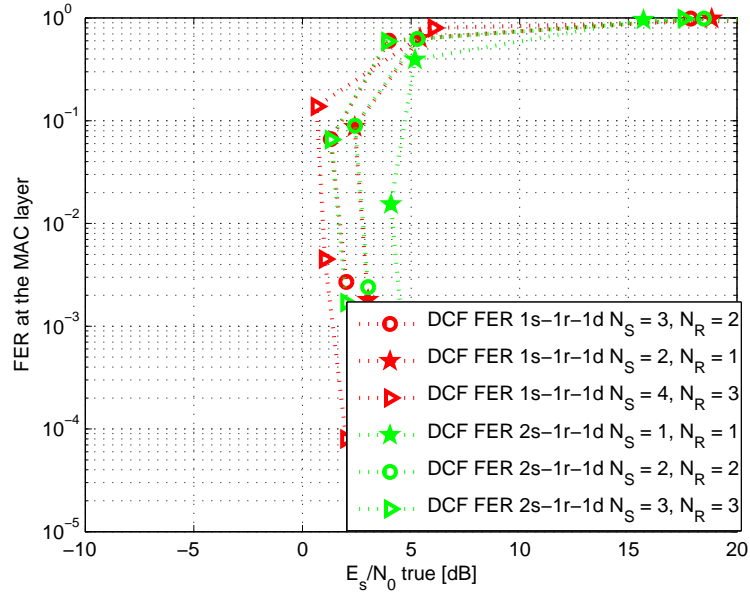


Figure 4.21: DCF protocol FER comparison for 2-S-1-R-1-D and 1-S-1-R-1-D networks at the MAC layer as a function of **true** E_s/N_0

- II. develop analytical tools for deriving performance evaluation metrics, i.e. FER, delay, efficiency at the MAC layer
- III. apply the energetic fair criterion developed in Chapter 2 to the cooperative relay network
- IV. study the behavior of the protocols for various number of retransmissions
- V. in the context of 2-source-1-relay-1-destination network, compare the developed protocols and discuss the advantages of each

First of all, we defined the DCF and DMF protocols in the context of multi-source cooperative network. There are several differences that arise when the number of nodes is increased: a) instead of treating the nodes as single networks, treat the sources jointly, i.e. perform XOR at the relay. Then, we also had the problem of alternating between the sources, i.e. not making one source wait for another. We develop analytical framework to analyze the performance metrics, but as it has been proven, this approach becomes non tractable and practically cannot be used. This mainly happens due to the definition of the states in the FSM and the number of variables per state. For this reason we consider only simulated metrics. We further perform energetic fair comparisons, which allow us drawing conclusions about the total energy expenditure per successful information bit. We could also see that for both protocols, we have improvements in the FER when the number of transmissions is increased. However, the delay and the efficiency is not improved significantly. Moreover, the delay becomes larger when the number of transmissions is increased, creating a tradeoff between FER and the delay. This constraint can be overcome by implementing the small number of transmissions in bad channel quality; and high number of transmissions when the quality of channel improves.

Overall this thesis has shown that the DCF protocol has advantages compared to the DMF both in the simplest cooperative networks, and in the more complicated ones. This work has the goal of a study of cooperative HARQ protocols QoS metrics in cooperative networks context. Although many works study cooperative networks, most of them focus only on some QoS metrics. This work has the aim of performing performance evaluations of the QoS in cooperative HARQ, which allows us to have a broader view on the performance of the system. Furthermore, the few works existing in the literature on the subject of cooperative HARQ (which also includes FEC) normally perform the analysis using combinatorial approach or simply by upper bounding the BER. However, those approaches have not come up with a framework to fully analyze the performance metrics of cooperative HARQ. Our goal is to fill this gap by providing system analysis tools and studying their feasibility. The main contributions of this work are summarized as follows:

- 1. A metric is derived that allows for computing the total energy expenditure per successful information bit. This metric is first tested on simple point-to-point networks, for HARQ and STBC protocols combination. We show that this metric permits having a different view on the performance metrics, and face other aspects, which are not revealed when the comparisons are done as a function of E_s/N_0 or E_b/N_0 .

2. We then analyze the protocol performance metrics by designing finite state machines. The theoretical FER has been computed for the smallest cooperative network and shown to be validated by Monte-Carlo simulations. This proves that the FSM analysis of the protocol can be done in theory. However, the complexity of this approach grows proportionally to the number of nodes in the network and the number of transmissions per node. Nevertheless, future works on this subject could be addressed towards the probabilistic protocol design for cooperative networks, with the aim of simplifying the analytical computations and produce smaller FSM. However this aspect has to be treated as a different problem since probabilistic protocols deal with other problems.
3. Extensive simulations are conducted for the various cooperative networks, in order to verify the efficiency of the proposed protocols and observe the behavior of the protocols when the number of transmissions is increased. It is shown that mostly the protocols improve performance

We develop and study two different cooperative schemes: $1 - S - 1 - R - 1 - D$ and $2 - S - 1 - R - 1 - D$. For each of these schemes we define two protocols and study the frame error rate, efficiency, and the delay for different number of transmissions. Moreover, we derive and propose the decoder for each protocol. This work answers the following questions:

- is network coding useful in with cooperative HARQ?
- how the protocols behave as the number of transmissions increases?
- which of the protocols is better: DCF or DMF (both in the $1 - S - 1 - R - 1 - D$ and $2 - S - 1 - R - 1 - D$ networks)
- how to fairly compare the cooperative HARQ schemes to the simple FEC?
- is it feasible to analyze the cooperative HARQ metrics using finite state automata?

In this chapter we studied the analytical performances of cooperative relay schemes using FSM as a framework. In the next chapter we address the questions of minimizing the FSM and optimizing the performances.

Chapter 5

Reducing the number of states in the FSM

In Chapter 3 we have introduced an analysis framework based on FSM, able to describe the performance of simple networks composed by one source, one relay and one destination. The same approach has been considered in Chapter 4 for the analysis of more complex schemes, composed by two source nodes, one relay and one destination. We showed how the number of states in the FSMC associated to the considered communication protocols in the two-sources networks increases very fast, as the protocol becomes more sophisticated (*e.g.* as effect of DCF mode at the relay, of combining at the receiver, of an increased number of allowed retransmissions). The approach considered so far, hence, is not scalable to more complex networks, where the number of emitting nodes (sources and/or relays) is increased.

This chapter introduces a modified FSM analysis framework able to better scale with the considered network size and protocol complexity. By treating the simple networks considered in Chapter 3, we provide a proof of concept of the feasibility and efficiency of this alternative strategy.

In the first part of this Chapter we illustrate the following. Suppose a given network topology, and an appropriate deterministic communication protocol (*e.g.* the *deterministic* protocols discussed in Chapter 3), achieving a performance that we measure in terms of MAC-layer FER, average number of transmissions, and goodput. On the same network topology it does exist a *probabilistic* communication protocol able to achieve, for opportune tuning of its design parameters, the same performance of the deterministic one. This finding is interesting, since the FSMC description of the probabilistic protocol is very compact, and can be used, thanks to the equivalence results, to provide the performance analysis of the deterministic protocol as well. This finally allows to derive a low-complexity optimization procedure of the design parameters of both protocols, as it will be detailed in the second part of this Chapter.

5.1 One source, one destination, no combining

This Section aims to motivate and introduce the new performance analysis framework, using the simplest possible example. We consider point-to-point communication between the source and the destination. A deterministic communication protocol has been introduced for this topology in Section 3.2. We briefly recall the deterministic protocol: the source can send the same fragment to the destination up to N times. After each transmission, the destination sends a control message (ACK, NACK) to the source node, to signal whether the fragment has been correctly decoded. We do not consider combining at the decoder. The source stops the retransmission of the current fragment as soon as an ACK is received by the destination, or the maximum number of allowed transmissions N has been reached. In Section 3.2 the behaviour of the deterministic protocol has been described using its associated FSMC, which in the following will be denoted as the *target* FSMC. In this Section we aim to find an *equivalent* FSMC representation with a smaller number of states than the target FSMC, but still able to describe the performance of the deterministic protocol, measured in terms of FER and average number of transmissions per fragment.

Since the target FSMC is irreducible, no state in the target FSMC can be eliminated, so the only solution is to change the definition of the states. We would like to define the equivalent FSMC as composed by only three states:

State S_1 first transmission of a new FRAG after a successful reception

State S_2 retransmissions of current FRAG

State S_3 first transmission of a new FRAG after a failed reception of the previous one.

As before, each state needs to be associated to a transmission. The state input W_t , taking values on $\{\text{ACK}, \text{NACK}\}$, determines the transition. We ask whether the equivalent FSMC with this definition of the states can be associated to deterministic protocol, and in order to do so we consider the transitions between the states.

According to the protocol, states S_2 and S_3 can both transition to state S_1 . These transitions depend only on the current state, and on the observed output of the transmission W_t . States S_1 and S_3 can both transition to state S_2 . These transition depend only on the current state, and on the observed output of the transmission W_t . State S_2 can transition to state S_3 . From the functioning of the deterministic protocol, however, we know that the transition from $S_2 \rightarrow S_3$ can happen (with probability π) only if the maximum number of retransmissions N has been reached, that is only if a self-loop $S_2 \rightarrow S_2$ has happened $N - 1$ times after the first time the system has reached state S_2 . This implies that the transition to state S_3 depends on the observed output W_t , but also on the permanence time in the state. This violates the Markov property that transitions depend only on the current state and on the input. It is concluded that an equivalent FSMC in this form cannot be associated to the deterministic protocol. (Example: let $N = 3$, and suppose the system has just entered state S_2 , meaning that the second transmission is about to happen. The probability of transition to S_3 is zero, since even in case of failure of the current transmission, another one is allowed by the deterministic protocol. The

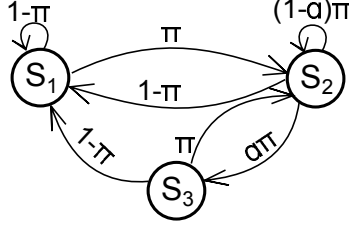


Figure 5.1: FSM for one source one destination ARQ protocol. $S_1 = (W_{t-1} = 0)$, $S_2 = (W_{t-1} = 0, J_{t-1} = 1)$, $S_3 = (W_{t-1} = 0, J_{t-1} = 0)$

output $W_t = \text{NACK}$ determines then a transition of the state on itself. The following transmission happens, yielding $W_t = \text{NACK}$. This time the system, in the same state and with the same output as before, transitions to state S_3).

5.1.1 Equivalent Probabilistic Protocol

We decide to adopt another strategy, and to define an equivalent protocol for the same network topology, such that it can be associated to the desired three-state FSMC. We define it using a probabilistic allocation of the retransmissions. The probabilistic protocol works as follows: if the first transmission is unsuccessful, a second transmission is granted with probability 1. If the i -th transmission is unsuccessful, for $i \geq 2$, another transmission is granted with probability $(1 - \alpha)$. To model this rule, we define a random variable J_t on support $\{0, 1\}$. The drawing of J_t in time-slot t determines whether another transmission will be allowed in time-slot $t+1$. We have $P(J_1) = 1$ in time-slot 1, and $P(J_t = 1) = (1 - \alpha)$ in time-slot $t > 1$. We suppose that no combining is performed at the destination.

The probabilistic protocol is associated to a FSMC with three states, defined as follows.

State S_1 : $S_1 = (W_{t-1} = \text{ACK}, J_{t-1} = *)$ is associated to the first transmission of a new FRAG, after the previous FRAG was successfully received at the destination

State S_2 : $S_2 = (W_{t-1} = \text{NACK}, J_{t-1} = 1)$ is associated with the repetition of the transmission, after the previous transmissions of the FRAG failed and the right to retransmit again was granted

State S_3 : $S_3 = (W_{t-1} = \text{NACK}, J_{t-1} = 0)$ is associated to the first transmission of a new FRAG, after the last transmission of the previous FRAG was unsuccessful, and the right to retransmit was not granted.

The FSMC is illustrated in Fig. 5.1. The transition between states depends on the outcome W_t of the current transmission, and on the realization of J_t . The transition probabilities for this FSMC are evaluated as follows:

$$P_{i,1} = P(W_t = \text{ACK}) = (1 - \pi), \quad i \in \{1, 2, 3\} \quad (5.1)$$

where the term π denotes the probability that one FRAG transmission on the source to destination channel fails. The states S_1 and S_3 can reach state S_2 after a failed FRAG

transmission:

$$P_{i,2} = P(W_t = \text{NACK}) = \pi, \quad i \in \{1, 3\} \quad (5.2)$$

The state S_2 can reach the state S_3 , if $W_t = \text{NACK}$ and $J_t = 0$, and loop onto itself if $W_t = \text{NACK}$ and $J_t = 1$.

$$\begin{aligned} P_{2,2} &= P(W_t = \text{NACK}, J_t = 1) = \pi(1 - \alpha), \\ P_{2,3} &= P(W_t = \text{NACK}, J_t = 0) = \pi\alpha \end{aligned} \quad (5.3)$$

Thus, the transition matrix takes the form:

$$P = \begin{pmatrix} 1 - \pi & \pi & 0 \\ 1 - \pi & (1 - \alpha)\pi & \alpha\pi \\ 1 - \pi & \pi & 0 \end{pmatrix}. \quad (5.4)$$

Evaluation of α

We now need to evaluate the parameter α of the probabilistic protocol. Our goal is to use the small FSMC associated to the probabilistic protocol to describe the performance of the deterministic protocol of Section 3.2. In order to do so, we need to tune α such that the FER and the average number of transmissions \bar{T} in the probabilistic protocol match the FER and \bar{T} obtained with the deterministic protocol, for N maximum transmissions. In both cases FER and \bar{T} depend on the steady state probabilities associated to S_1 (transmission of a new FRAG after success) and to the last state (transmission of a new FRAG after failure). In order to achieve the equivalence, it is sufficient to impose the value of α such that the probability of terminating the retransmission phase in the probabilistic protocol matches the probability of terminating the retransmission phase in the deterministic protocol. Notice that α is defined, in the probabilistic protocol, as the probability of not being allowed to perform another transmission, once the retransmission phase has begun (this means that we failed the first and the second transmissions). In the deterministic protocol, we were not allowed to perform another transmission only if we were in state N of the target FSMC. As a consequence, α is evaluated as the probability of being in state N of the target FSMC, knowing that we already entered the retransmissions phase, that is, knowing that we are either in state 2, 3, ... or N of the target FSMC. Denoting as $\bar{\mathbf{p}}$ is the steady state vector of the target FSMC associated to the deterministic protocol, α is then evaluated as:

$$\begin{aligned} \alpha &= P(\text{current state} = N \mid \text{current state} \in \{2, 3, \dots, N\}) \\ &= \frac{P(\text{current state} = N, \text{current state} \in \{2, 3, \dots, N\})}{P(\text{current state} \in \{2, 3, \dots, N\})} \\ &= \frac{P(\text{current state} = N)}{P(\text{current state} \in \{2, 3, \dots, N\})} \\ &= \frac{\bar{p}_N}{\sum_{i=2}^N \bar{p}_i} \end{aligned} \quad (5.5)$$

The derivation of the parameter α can be performed without making use of the steady state vector $\bar{\mathbf{p}}$ of the target FSMC, since the steady state probability of being in one of the states can be evaluated directly. Let A_t denote the number of the transmission effectuated at time-slot t , in the deterministic protocol. The probability α is the probability that the current transmission is the last allowed (*i.e.* the N -th transmission), knowing that the previous one was not the last allowed. The expression of α can be evaluated as follows:

$$\begin{aligned}
\alpha &= P(A_t = N | 1 \leq A_{t-1} \leq N-1) \\
&= \frac{P(A_t = N, 1 \leq A_{t-1} \leq N-1)}{P(A_t = N | 1 \leq A_{t-1} \leq N-1)} \\
&= \frac{P(A_t = N)}{\sum_{i=2}^N P(A_t = i)} \\
&= \frac{\pi^{(N-1)}}{\sum_{i=2}^N \pi^{(i-1)}} \tag{5.6}
\end{aligned}$$

5.1.2 Performance evaluations

In Section 3.2 we showed how the performance of the deterministic protocol can be evaluated as a function of the steady state vector of the target FSMC. Similarly, we derive here the performance of the probabilistic protocol as a function of the steady state vector of the equivalent FSMC.

The Frame Error Rate

In order to evaluate the FER of the equivalent protocol, using the equivalent FSMC, it is required to evaluate the steady state vector \mathbf{p} . Similarly to the original FSM definition, we have to compute the probability that a new FRAG is transmitted after the previous FRAG failed, averaged over the sum of probability denoting the transmission of a new FRAG after the previous FRAG failed and probability denoting the transmission of a new FRAG after the previous FRAG was OK. By denoting the steady state probability to be in state i by p_i , and using its definition given in Section 1.2, we obtain the FER:

$$\text{FER} = \frac{p_1}{p_1 + p_3} \tag{5.7}$$

The average number of transmissions per successful FRAG (delay)

The average number of transmissions per successfully received fragment can be evaluated, following the definition given in Section 1.2, as follows. Notice that the probability of being at the first transmission is $P(1\text{-st transmission}) = p_1 + p_3$. The probability of being at the second transmission can be evaluated as

$$P(2\text{-nd tran.}) = P(W_t = \text{NACK} | 1\text{-st tran.})P(1\text{-st tran.}) = \pi(p_1 + p_3).$$

Similarly, one has

$$P(3\text{-rd tran.}) = P(W_t = \text{NACK and } J_S = 1 | 2\text{-nd tran.})P(2\text{-nd tran.}) = \pi(1-\alpha)P(2\text{-nd tran.}).$$

The average number of transmissions per successful FRAG is hence easily evaluated as

$$\begin{aligned}\bar{S}_{\text{MAC}} &= \sum_{i=1}^{\infty} i \cdot \Pr\{i \text{ transmissions performed} \mid \text{FRAG is successfully received}\} \\ &= \sum_{i=1}^{\infty} \frac{i \cdot \Pr\{\text{be in state 1} \mid \text{after } i\text{-th tran.}\}}{\Pr\{\text{be in state 1}\}} \\ &= (1-\pi)\frac{p_1+p_3}{p_1} + \sum_{i=2}^{\infty} i(1-\pi)\frac{p_1+p_3}{p_1}\pi^{i-1}(1-\alpha)^{i-2}\end{aligned}\quad (5.8)$$

The average number of transmissions per FRAG

In a similar way, using its definition given in Section 1.2, we obtain the average number of transmissions per FRAG:

$$\begin{aligned}\bar{T}_{\text{MAC}} &= \frac{p_1}{p_1+p_3}\bar{S}_{\text{MAC}} + \frac{p_3}{p_1+p_3}\sum_{i=1}^{\infty} i \cdot P(i \text{ tran. perf} \mid \text{FRAG wrongly decoded}) \\ &= \frac{p_1}{p_1+p_3}\bar{S}_{\text{MAC}} + \alpha\pi\sum_{i=2}^{\infty} i \cdot \pi^{i-1}(1-\alpha)^{i-2}\end{aligned}\quad (5.10)$$

5.1.3 Simulation validation

In the previous Section we provided the expressions of the FER, \bar{T} , \bar{S} associated to the probabilistic protocol, as functions of the steady state vector of the equivalent FSMC. In (5.5) and (5.6) we provided the expressions used to evaluate the parameter α to be used in order to ensure that the probabilistic protocol yields the same performance of the deterministic protocol with maximum number of transmissions fixed to N .

In order to validate the above analysis, we consider a simulated example. We consider the deterministic protocol, and we evaluate the FER for $N = 5$, for different values of the probability π . We compare in Table 5.1 the predicted values of the FER obtained using the combinatorial approach (3.4) (label in data: **FER_comb**), using the target FSMC (3.8) (label in data: **FER_target**), using the FSMC associated to the probabilistic protocol (5.7), for equivalent α (for α as in (5.5) for label **FER_equiv**, and for α as in (5.6) for label **FER_equiv_2**), and finally simulation results (label in data: **FER_sim**).

Table 5.2 shows the same results for what concerns the prediction of the value of \bar{T} . It can be observed that the evaluation using the FSMC of the equivalent experiment, with α evaluated to yield the same FER as the target FSMC, provides a prediction of \bar{T} which is very close to the one of the target FSMC, but not exactly the same (but always bigger). This is the effect that we observe due to the fact that in the equivalent protocol the probability of performing a number of trials bigger than N is not negligible.

π	FER_comb	FER_target	FER_equiv	FER_equiv_2	FER_sim
0.90	0.59049	0.59049	0.59049	0.59049	0.58665
0.80	0.32768	0.32768	0.32768	0.32768	0.32409
0.70	0.16807	0.16807	0.16807	0.16807	0.16677
0.60	0.07776	0.07776	0.07776	0.07776	0.07711
0.50	0.03125	0.03125	0.03125	0.03125	0.03194
0.40	0.01024	0.01024	0.01024	0.01024	0.00989
0.30	0.00243	0.00243	0.00243	0.00243	0.00245
0.20	0.00032	0.00032	0.00032	0.00032	0.00031
0.10	0.00001	0.00001	0.00001	0.00001	

Table 5.1: Theoretical prediction of the FER and simulation results, $N = 5$.

π	AV_T_comb	AV_T_target	AV_T_equiv	AV_T_sim
0.90	4.09510	4.09510	4.28588	4.13321
0.80	3.36160	3.36160	3.50035	3.36121
0.70	2.77310	2.77310	2.86789	2.79169
0.60	2.30560	2.30560	2.36516	2.30864
0.50	1.93750	1.93750	1.97083	1.93350
0.40	1.64960	1.64960	1.66536	1.64490
0.30	1.42510	1.42510	1.43082	1.42567
0.20	1.24960	1.24960	1.25088	1.24959
0.10	1.11110	1.11110	1.11119	1.11111

Table 5.2: Theoretical \overline{T} prediction and simulation results, $N = 5$.

5.2 One source, one relay, one destination, no combining

In this Section we consider the same network topology considered in Section 3.3, where one source and one destination communicate aided by one relay. We briefly recall the considered deterministic protocol. The source and the relay are allowed to transmit maximum N_S and N_R times, respectively. The relay works in DMF mode. If the first transmission from the source is unsuccessful, the relay is granted the right to retransmit first. Retransmissions from the source happens only after the relay has spent its maximum number of allowed transmissions N_S . No combining is performed at the decoders.

As done in the previous example, we derive a probabilistic communication protocol for the same network topology considered in Section 3.3, and we tune its parameters in order to provide the same performance.

5.2.1 Equivalent Probabilistic Protocol

We describe the probabilistic protocol, and its associated FSMC, composed by four states. The probabilistic protocol for the one source, one relay, one destination topology works as follows. We define two control variables, J_t^S and J_t^R , taking values on the support $\{0, 1\}$. The realization of the random variables J_t^S and J_t^R in time-slot t determines whether the source and the relay are allowed to transmit in time-slot $t + 1$. If both $J_t^S = 1$ and $J_t^R = 1$ the relay has priority in retransmission in time-slot $t + 1$, and the source is silent. We suppose that all nodes share the same seed, so that the same drawing of both J_t^S and J_t^R can be obtained, at each time-slot, at all nodes. This allows to avoid collisions due to the simultaneous attempt of transmission by two nodes.

In the first time-slot the source generates and transmits a FRAG to the relay and the destination, and $J_1^R = 1$, $J_1^S = 1$ with probability 1. As a consequence, the relay is granted transmission with probability 1 in time-slot 2. In time-slot 2, $J_2^S = 1$ with probability 1 and $J_2^R = 1$ with probability $(1 - \beta)$, so that the relay keeps control of the retransmission phase in time-slot 3 with probability $(1 - \beta)$, or passes it to the source with probability β . The relay keeps retransmitting, and the same probability of drawing J_t^R and J_t^S persists, until $J_i^R = 0$ is observed in time-slot i , and the control of the retransmission phase passes to the source in time-slot $i + 1$. In the following, $J_t^R = 0$ with probability 1 for $t \geq i + 1$, and $J_t^S = 1$ with probability $(1 - \alpha)$ for $t \geq i + 1$, that is the source keeps control of the retransmission phase, until $J_t^S = 0$ is observed, and the current fragment is dropped. No combining is performed at the receivers.

The definition of the states of the FSMC associated to the probabilistic protocol is given by the configuration of the variables W_t (state input), J_t^S and J_t^R . The states are listed in Table (5.3). State S_1 represents the first transmission of a new FRAG, after the previous has been successfully received ($W_{t-1} = \text{ACK}$). State S_2 represents the retransmission phase, when the relay is granted the right to retransmit. State S_3 represents the retransmission phase, when the source is granted the right to retransmit. State S_4 represents the first transmission of a new FRAG, after the previous has been dropped because of the termination of the retransmission phase of the source.

Recall that π_{SD} represents the probability that the fragment is not successfully decoded after a (re)transmission from the source, and that π_{RD} represents the probability that the

Table 5.3: DMF Equivalent FSM States

State number	State definition
S_1	$W_{t-1} = \text{ACK}, J_{t-1}^S = *, J_{t-1}^R = *$
S_2	$W_{t-1} = \text{NACK}, J_{t-1}^S = 1, J_{t-1}^R = 1$
S_3	$W_{t-1} = \text{NACK}, J_{t-1}^S = 1, J_{t-1}^R = 0$
S_4	$W_{t-1} = \text{NACK}, J_{t-1}^S = 0, J_{t-1}^R = 0$

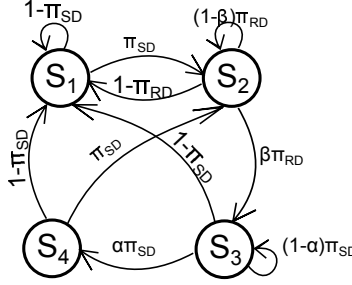


Figure 5.2: FSM for one source, one relay, and one destination DMF equivalent protocol. The terms α and β denote the probabilities that the source/relay is not allowed to transmit anymore

fragment is not successfully decoded after a retransmission from the relay. The transition probability matrix of the equivalent FSMC is given in Eq. (5.11):

$$P = \begin{pmatrix} 1 - \pi_{SD} & \pi_{SD} & 0 & 0 \\ 1 - \pi_{RD} & (1 - \beta)\pi_{RD} & \beta \cdot \pi_{RD} & 0 \\ 1 - \pi_{SD} & 0 & (1 - \alpha)\pi_{SD} & \alpha \cdot \pi_{SD} \\ 1 - \pi_{SD} & \pi_{SD} & 0 & 0 \end{pmatrix} \quad (5.11)$$

The FSM is illustrated in Fig. 5.2.

Evaluation of α and β

We evaluate now the expressions of the design parameters α and β of the probabilistic protocol evaluated in order to guarantee that the performance of the deterministic protocol with maximum number of retransmissions N_S and N_R is matched. The procedure to derive them is similar to the one delineated for the case of the one source, one destination scheme. The parameters, expressed as functions of the steady state vector $\bar{\mathbf{p}}$ of the target FSMC associated to the deterministic protocol, and as functions of $\pi_{SD}, \pi_{RD}, N_S, N_R$ only, take form

$$\alpha = \frac{\bar{p}_{N_R+N_R}}{\sum_{i=N_R+2}^{N_S+N_R} \bar{p}_i} = \frac{\pi_{SD} \pi_{RD}^{N_R} \pi_{SD}^{N_S-2}}{\sum_{i=N_R+2}^{N_R+N_S} \pi_{SD} \pi_{RD}^{N_R} \pi_{SD}^{i-N_R-2}} \quad (5.12)$$

$$\beta = \frac{\bar{p}_{N_R+1}}{\sum_{i=2}^{N_R+1} \bar{p}_i} = \frac{\pi_{SD} \pi_{RD}^{N_R-1}}{\sum_{i=2}^{N_R+1} \pi_{SD} \pi_{RD}^{i-2}}. \quad (5.13)$$

5.2.2 Performance Evaluations

In this Section we provide the expressions of the performance metrics of the probabilistic protocol, evaluated using the FSMC. The strategy followed to derive them is similar to the one described for the one source, one destination scheme. Let \mathbf{p} denote the steady state vector of the FSMC associated to the probabilistic protocol.

FER

$$\text{FER} = \frac{p_4}{p_1 + p_4} \quad (5.14)$$

Average number of transmitted FRAGs per successful FRAG (delay)

In the probabilistic protocol the transmissions can happen infinitely. For this reason, the computation of the term \bar{S}_{MAC} takes a slightly different form. All the events of the successful reception of the FRAG have to be computed.

$$\begin{aligned} \bar{S}_{\text{MAC}} = & (1 - \pi_{SD}) \frac{p_1 + p_4}{p_1} + 2 (1 - \pi_{RD}) \frac{p_1 + p_4}{p_1} \pi_{SD} \\ & + \sum_{t=3}^{\infty} t \frac{p_1 + p_4}{p_1} \pi_{SD} \left((1 - \pi_{RD}) (\pi_{RD} (1 - \beta))^{t-2} \right. \\ & \left. + (1 - \pi_{SD}) \sum_{k=0}^{t-3} \beta \pi_{rd} (\pi_{RD} (1 - \beta))^k (\pi_{SD} (1 - \alpha))^{t-3-k} \right) \end{aligned} \quad (5.15)$$

Average number of transmissions per FRAG

In order to compute the total number of transmissions per FRAG, again we have to take into account the fact that there is a chance that one of the nodes will infinitely transmit. In other words, we do not know the exact amount of times that the source can transmit and the number of times that the relay will transmit. For this reason we consider all the possible combinations between them.

$$\begin{aligned} \bar{F} = & \frac{p_1 + p_4}{p_4} \pi_{SD} \left(3 \beta \pi_{RD} \alpha \pi_{SD} \right. \\ & \left. + \sum_{t=4}^{\infty} t \left(\sum_{k=0}^{t-3} \alpha \pi_{SD} ((1 - \beta) \pi_{RD})^{t-3-k} \beta \pi_{RD} ((1 - \alpha) \pi_{SD})^k \right) \right) \end{aligned} \quad (5.16)$$

$$\bar{T} = (1 - \text{FER}) \bar{S} + \text{FER} \bar{F} \quad (5.17)$$

5.3 Performance optimization

In Section 5.2 we introduced a probabilistic communication protocol, for the setup composed by one source, one relay, one destination. The relay works in DMF mode, and no combining is considered at the destination. The probabilistic protocol is an alternative to the deterministic protocol introduced for the same setup in Section 3.3.

For a given physical-layer error protection scheme (modulation scheme and FEC code), the design parameters affecting the performance of the deterministic protocol are the number of allowed transmissions from the source and from the relay, N_S and N_R , respectively. Similarly, the design parameters in the probabilistic protocol are the probabilities $(1 - \alpha)$ and $(1 - \beta)$ that another retransmission is granted to the relay or to the source. In this Section we address, for both protocols, the problem of design parameter optimization. The QoS metric that we choose to consider is the MAC-layer goodput at the destination, which is defined as

$$\zeta = \# \text{ correct information bits } [\text{tu}^{-1}], \quad (5.18)$$

where tu denotes the time unit. We define the time unit tu as the amount of time necessary to complete the transmission of an uncoded fragment (L information bits) on a given transmission link. Under this assumption, it is easy to see that the goodput ζ can be evaluated (both for the deterministic and probabilistic protocols) as

$$\zeta = R_c \frac{(1 - \text{FER})}{\bar{T}}. \quad (5.19)$$

Before discussing optimization, we introduce the considered model for the channel propagation conditions.

5.3.1 Physical layer setup

The reference scenario considered for protocol optimization explicitly accounts for the path loss effects in propagation. We assume that the source node is placed at distance d_0 from the destination node. In the following, we consider distance measurements normalized with respect to d_0 . Let d and $(1 - d)$ denote the normalized distances between the relay and the destination, and between the source and the relay, respectively. Medium access control is managed via time-division multiplexing. We assume that the nodes are synchronized, and time-slots of duration t seconds. As before, information is transmitted using MAC layer fragments of length L information bits. As before, we will assume $L = 120$.

We consider the BPSK modulation scheme. The same modulation scheme is adopted between each communicating pair. As before, let x denote the generic modulation symbol, with unitary symbol energy. We assume that the channel between any two pairs of communicating nodes X and Y is statistically independent on all the other channels. The channel is degraded by independent AWGN noise, path loss, and multipath fading. The propagation model is described as

$$y_{XY} = \sqrt{\ell(d_{XY})E_s}h_{XY}x + n_{XY} \quad (5.20)$$

where x is the modulation symbol, and E_s is the per symbol energy; h_{XY} is the Rayleigh fading coefficient, such that $h_{XY} \sim \mathcal{CN}(0, 1)$; n_{XY} is the AWGN noise term, such that

2/3	1/2	2/5	1/3
11111111	11111111	11111111	11111111
10101010	11111111	11111111	11111111
00000000	00000000	11001100	11111111

Table 5.4: Considered puncturing matrices and corresponding code rates.

$n_{XY} \sim \mathcal{CN}(0, N_o)$; $\ell(d_{XY})$ is the path-loss coefficient, dependent on the normalized (with respect to d_0) distance d_{XY} between the transmitting nodes. The path-loss coefficient is assumed to take the form $\ell(d_{XY}) = d_{XY}^{-\alpha}$, with $\alpha \in [2, 4]$. Notice that $\ell(d_{XY}) = \ell(d_0/d_0) = 1$ when X is the source and Y is the destination. We assume that perfect Channel State Information (CSI) is available at the receiver, and coherent demodulation. The receive per-symbol signal-to-noise ratio γ_s takes hence form

$$\gamma_s = \frac{\ell(d_{XY})|h_{XY}|^2 E_s}{N_o}. \quad (5.21)$$

We assume the same transmit E_s/N_o on all transmitting nodes. The receive γ_s at a receiving node depends on the distance from the emitter, and on the path loss coefficient α . The path loss coefficient α is assumed constant on the SR and RD links. We consider in this setting fully-interleaved fading, *i.e.* ideal interleaving of the modulated symbols before transmission, and independent realizations of the fading coefficients h on each symbol.

Before modulation, the MAC fragment is encoded using a rate R_c channel code. In this Section we consider Rate Compatible Punctured Convolutional (RCPC) codes ???. The chosen mother code is a $R_c = 1/3$ convolutional code with constraint length $K = 7$, memory $M = 6$ with generator polynomials (in octal form)

$$g_1 = 0133, \quad g_2 = 0171, \quad g_3 = 0145. \quad (5.22)$$

The puncturing period is $P = 8$. The considered puncturing matrices ?? are described in Table 5.4.

Figure 5.3.1 describes the behaviour of the elementary fragment error probability π_{SD} on the source-destination link, as a function of the transmit E_s/N_o . The π_{SD} values have been obtained via Monte Carlo simulation of the transmission of coded data packets on the source-destination link, for the setting described above. In our protocol we are obviously interested in considering configurations where $\pi_{RD} < \pi_{SD}$. Notice that π_{RD} depends not only on the transmit E_s/N_o at the relay and on the normalized distance d of the relay node from the destination node, but also on the error propagation effect through the relay. Recall that in the DMF operating mode the relay does not perform channel decoding on the received fragment, but simply demodulates, modulates and forwards the coded symbols. This implies that the received symbol sequence at the destination node is corrupted both by the effects of propagation on the source-relay and the relay-destination links. In the following, we will consider a setup where the relay is close to the source node (normalized distance of the relay from the destination is set to $d = 0.85$). The considered path-loss coefficient is set to $\alpha = 3.6$, corresponding to propagation in urban environment. The

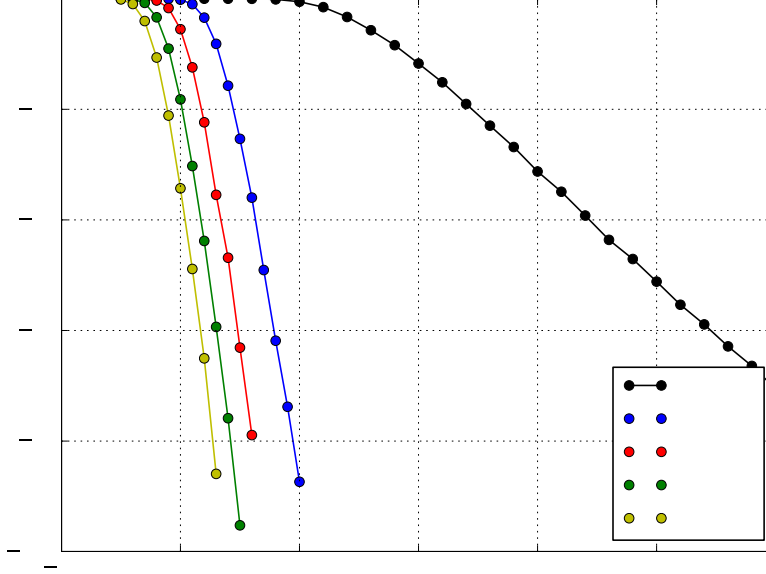


Figure 5.3: Simulated values of π_{SD} for various code rates.

values of π_{RD} , obtained via Monte Carlo simulation of the transmission of packets over the source-relay-destination link, are plotted in Figure 5.3.1 for various code rates (marked lines), and compared with the corresponding values of π_{SD} (unmarked lines). Figure 5.3.1 allows to compare the effects of the error propagation phenomenon in presence and absence of FEC coding. For low transmit E_s/N_o $\pi_{RD} < \pi_{SD}$, for all considered code rates. As E_s/N_o increases, however, π_{RD} improves slower than π_{SD} , whenever FEC is applied. This effect is due to the channel decoding operation at the destination.

5.3.2 Optimization of the deterministic protocol

In this section we consider the problem of optimization of the design parameters for the deterministic protocol, described in Section 3.3. We recall that in the deterministic protocol the relay has priority in retransmitting over the source, and that the maximum amount of allowed transmissions are N_S from the source, and N_R from the relay. No combining is performed at the receiver. The reference propagation condition on the channels is the one described in Section 5.3.1. For each considered value of the transmit E_s/N_o , we want to find the pair (N_S, N_R) maximizing the goodput ζ (5.19), *i.e.* to find the solution to the discrete optimization problem

$$\arg \max_{N_S, N_R} R_c \frac{(1 - \text{FER})}{\bar{T}}. \quad (5.23)$$

The FSMC representation of the performance of the protocol is very useful in order to provide a fast numerical solution to the optimization problem (5.23). The goodput ζ

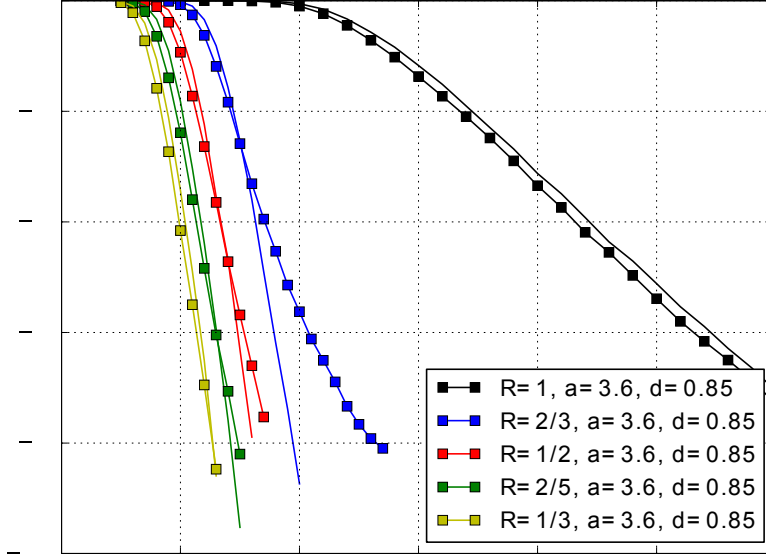


Figure 5.4: Simulated values of π_{SD} and π_{RD} for various code rates. The normalized distance between relay and destination is $d = 0.85$, and the path-loss exponent is $\alpha = 3.6$. Marked lines represent the behaviour of π_{RD} , and unmarked lines represent the behaviour of π_{SD} .

can in fact be easily computed using (5.19). The optimization algorithm works as follows. A maximum value \mathcal{M} for N_S and N_R is imposed. For each considered value of E_s/N_o , and for each possible configuration of the pair (N_S, N_R) , the goodput ζ is evaluated. This allows to discover the pair (N_S, N_R) maximizing the goodput. Notice that this procedure requires to evaluate the steady state vector of the big FSMC associated to the deterministic protocol \mathcal{M}^2 times for each considered E_s/N_o point of interest.

While the complexity of this calculation is not very large for the very simple setup considered in this Section, it becomes fast intractable when the number of states in the FSMC increases, because of the numerical burden determined by several evaluations of the steady state vector associated to a large probability transition matrix. (In our example, a very large probability transition matrix would be determined by choosing a large \mathcal{M} .) The much more compact FSMC representation of the probabilistic protocol can be used to effectively reduce the complexity of the protocol optimization. In Section 5.2.1 we showed that there always exist a pair of parameters (α, β) such that the FER and \bar{T} of the probabilistic protocol match the FER and \bar{T} of the deterministic protocol. Moreover, we showed how α and β can be evaluated without building the steady state vector of the big FSMC associated to the deterministic protocol. In order to solve the problem (5.23) we proceed hence as follows. For each considered value of E_b/N_o , and for each possible configuration of the pair (N_S, N_R) , we evaluate the equivalent pair (α, β) , and we then

proceed to evaluate the goodput using the FSMC associated to the probabilistic protocol. Notice that this requires, as before, to evaluate the steady state vector \mathcal{M}^2 times. The size of the probability transition matrix of the equivalent FSMC, however, is much smaller than the one associated to the deterministic protocol. Moreover, it is constant for each considered pair (N_S, N_R) , that is, it is independent on \mathcal{M} . For this reasons solving the optimization problem using the equivalent FSMC associated to the probabilistic protocol is computationally more convenient.

Numerical results

This Section presents the results of the optimization problem (5.23), solved using the equivalent FSMC representation of the protocol, as described above. The considered conditions on the physical channels are described in Section 5.3.1.

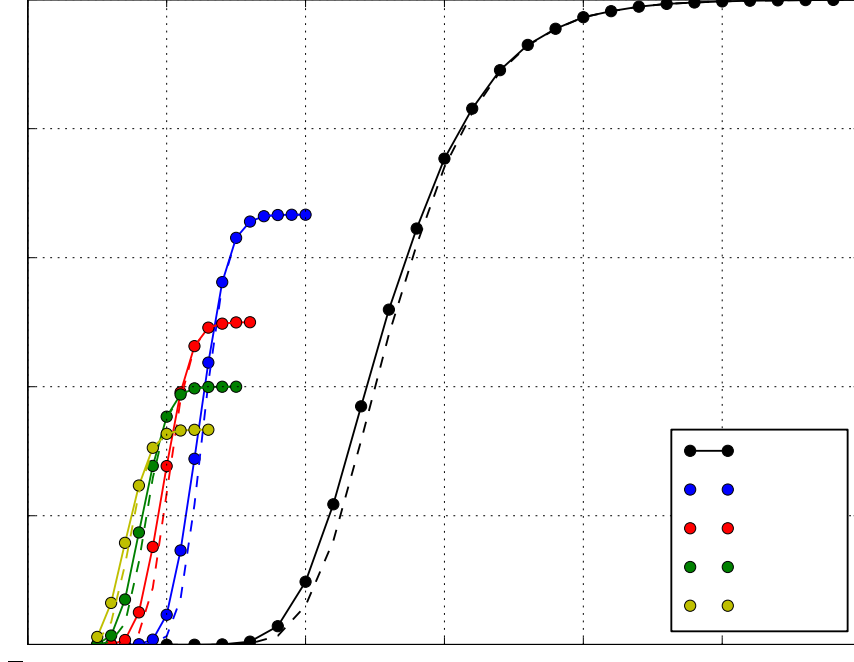


Figure 5.5: Optimized goodput for the 1s-1r-1d deterministic protocol, DMF mode, no combining, as a function of the transmit E_s/N_o at the source and at the relay (marked curves). Dashed curves represent the performance of the scheme without relay.

Figure 5.5 shows the optimized goodput ζ , as a function of the transmit E_s/N_o at the source and at the relay, for various code rates (marked lines). The dashed lines in 5.5 represent the optimized performance of a system without the relay, *i.e.* of a system where retransmission can be performed only from the source node. The comparison shows

that the presence of the relay brings some performance improvement. The performance improvement is very mild, and concentrated in the low E_s/N_o region. This is due to the behaviour of the considered π_{SD} and π_{RD} , depicted in Figure 5.3.1, and on the fact that π_{SD} improves faster than π_{RD} with E_s/N_o : for any considered code rate, as soon as $\pi_{RD} > \pi_{SD}$ the optimized system allocates all retransmissions on the source-destination channel, which becomes the most reliable route.

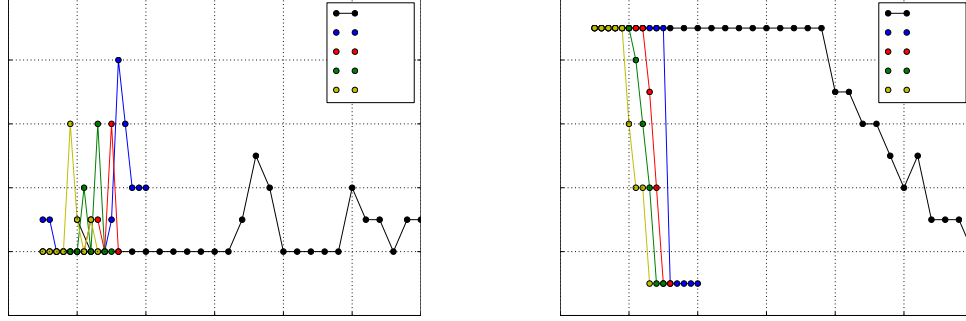


Figure 5.6: Optimum values of N_S (left) and N_R (right) for the 1s-1r-1d protocol, DMF mode, no combining, as a function of the transmit E_s/N_o at the source and at the relay.

This behaviour of the optimum parameters N_S and N_R is depicted in Figure 5.6. The optimum values of N_R are plotted on the right side. For low E_s/N_o , where $\pi_{RD} < \pi_{SD}$, the relay is granted the maximum number \mathcal{M} of allowed transmissions, while N_S is kept low. As E_s/N_o increases, and the difference between π_{RD} and π_{SD} diminishes, the budget of allowed retransmissions is gradually deplaced on the source. When the quality on the channel becomes good (high E_s/N_o) the optimum number of total allowed transmissions decreases.

Figure 5.7 depicts the FER associated to the optimized goodput, as a function of the transmit E_s/N_o at the source and at the destination, for all considered code rates. Not surprisingly, maximization of the goodput guarantees that the FER is restrained, and thus decreasing as E_s/N_o increases, except for very small values of FER ($\text{FER} < 10^{11}$). This effect is primarily due to the discrete nature of the protocol parameters.

Finally, Figure 5.8 depicts the behaviour of the delay \bar{S} (left) and of the average number of transmissions per fragment \bar{T} (right) of the optimized scheme, as functions of the transmit E_s/N_o at the source and at the relay. As expected, both \bar{S} and \bar{T} decreases as the quality on the channels improves.

5.3.3 Optimization of the probabilistic protocol

In this Section we address the problem of optimizing the design parameters of the probabilistic protocol. Recall that in the probabilistic protocol the relay has priority in retransmission over the source; the first retransmission from the relay is always granted,

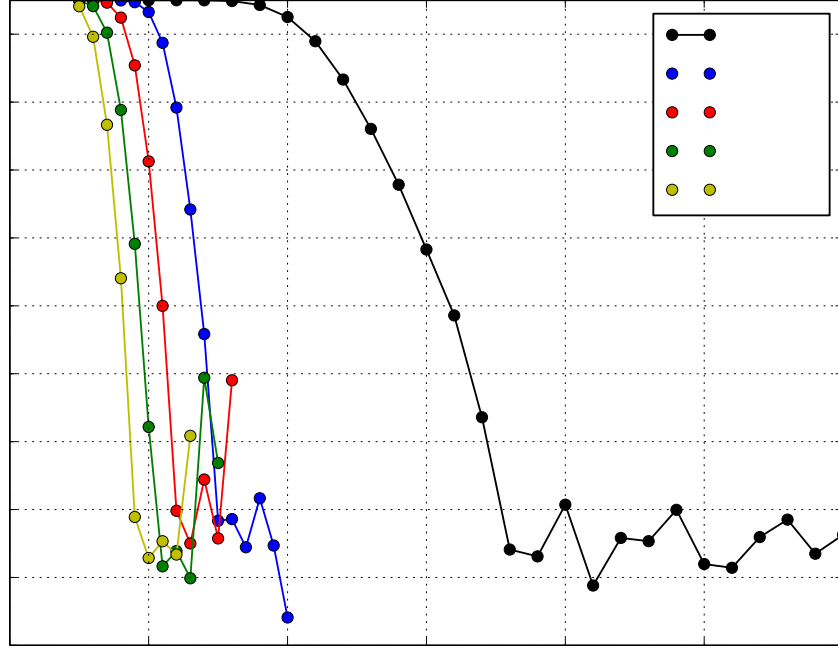


Figure 5.7: FER for the optimized 1s-1r-1d deterministic protocol, DMF mode, no combining, as a function of the transmit E_s/N_o at the source and at the relay (marked curves).

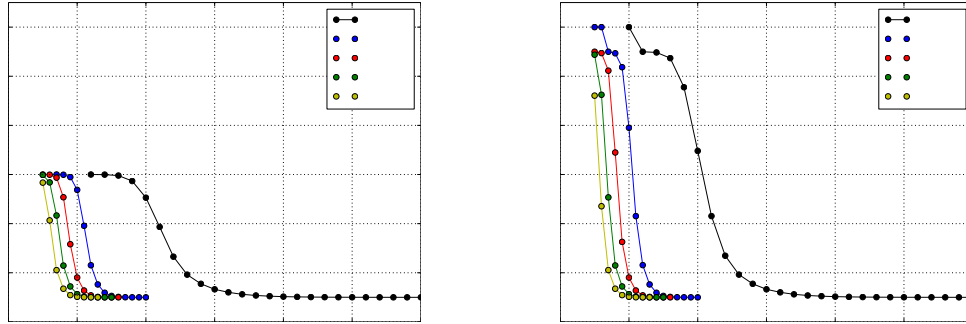


Figure 5.8: \bar{S} (left) and \bar{T} (right) for the optimized 1s-1r-1d protocol, DMF mode, no combining, as a function of the transmit E_s/N_o at the source and at the relay.

and subsequent are authorized with probability $(1 - \alpha)$. Retransmissions from the source

begin when the relay is first denied the possibility to retransmit. The first retransmission from the source is granted with probability 1, and subsequent retransmissions are authorized with probability $(1 - \beta)$. No combining is performed at the receiver.

In Section 5.3.2 we addressed the problem of optimizing the design parameters of the deterministic protocol, in order to maximize the goodput. We used the equivalent FSMC associated to the probabilistic protocol only to improve the efficiency of the optimization procedure, on the basis of the equivalence results derived in the first part of this Chapter.

The substantial difference between the probabilistic and the deterministic protocols resides in the nature of their design parameters (real parameters in the probabilistic protocol, discrete parameters in the deterministic protocol). In this Section we aim to investigate whether this difference, able to provide the probabilistic protocol with increased design freedom, is able to improve the achievable performance of the probabilistic protocol with respect to the achievable performance of the deterministic protocol. In order to do so, we consider the goodput optimization problem

$$\arg \max_{\alpha, \beta} R_c \frac{(1 - \text{FER})}{\bar{T}}. \quad (5.24)$$

The optimization problem (5.24) can be solved numerically, using, in principle, the same strategy employed for the deterministic protocol (*i.e.* exploring many possible configurations of the pair (α, β) , for each considered value of E_s/N_o). The real nature of the parameters, however, makes this brute-force evaluation computationally very expensive.

In order to make the optimization problem tractable also for the probabilistic protocol, we consider the following strategy, in two phases. The first phase consists in the optimization of the deterministic protocol, which, as seen in Section 5.3.2, can be performed with reduced computational cost. The first phase allows to evaluate a pair (α, β) equivalent to the optimum pair (N_S, N_R) associated to the deterministic protocol. The pair (α, β) represents the initialization point of the optimization procedure carried out in the second phase, where the search window is limited to a small interval centered around the initialization point. This strategy allows to increase the precision with which the considered (α, β) vary, restraining the complexity of the computation. It should be underlined, however, that this optimization procedure is a greedy algorithm, not able to guarantee that the result corresponds to the best achievable performance.

Numerical results

In this Section we illustrate the results of the goodput optimization of the probabilistic protocol, with the greedy algorithm described in the previous Section, and we compare them with the optimized deterministic protocol.

Figure 5.9 depicts the optimized goodput of the probabilistic protocol as function of the E_s/N_o at the source and at the relay. The performance (marked curves) is compared with the achievable performance of the deterministic protocol (dashed curves). For low E_s/N_o the probabilistic protocol allows to achieve a small improvement, which vanishes as the propagation conditions on the channels get more favorable.

In Figure 5.10 it is depicted the behaviour of the FER for the optimized probabilistic protocol (marked lines), and it is compared with the achievable FER of the optimized

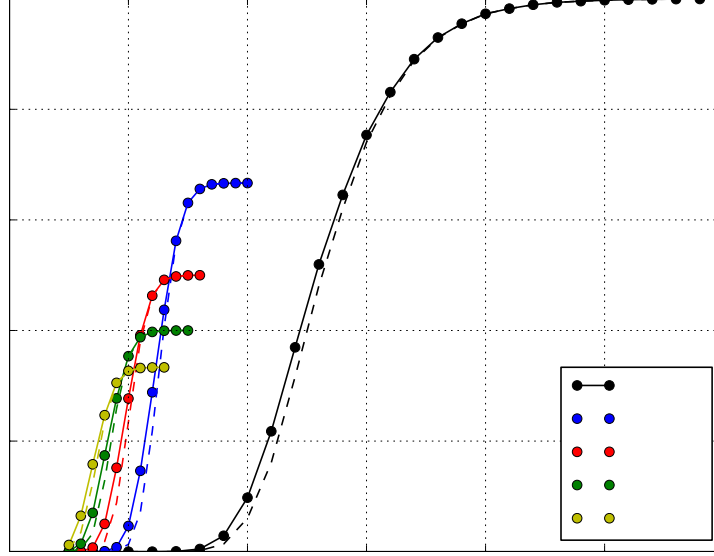


Figure 5.9: Optimized goodput for the 1s-1r-1d probabilistic protocol, DMF mode, no combining, as a function of the transmit E_s/N_o at the source and at the relay (marked lines). Dashed lines represent the optimum goodput of the deterministic protocol.

deterministic protocol. On this QoS metric the improvement of the probabilistic over the deterministic protocol is remarkable. For low E_s/N_o values an improvement of the performance is remarked also with respect to the average number of transmissions per fragment \bar{T} , as shown in Figure 5.11. This comes at the expenses of a increase of the delay \bar{S} .

5.3.4 Conclusions

In this Chapter we addressed the problem of devising a framework based on FSM that allows the analysis of the performance of cooperative networks, with constrained complexity. In order to do so, we showed that it is possible, for a given network topology and a given deterministic communication protocol, to design a probabilistic communication protocol achieving the same performances. The FSMC associated to the probabilistic protocol, which contains only a small number of states, can hence be used, for appropriate tuning of the parameters, to obtain the prediction of the performance of the deterministic protocol as well.

This performance analysis strategy makes then the problem of the optimization of the deterministic protocol parameters computationally tractable.

In the second part of the Chapter we compared the achievable performances of the deterministic and of the probabilistic protocols, defined on the same network topology.

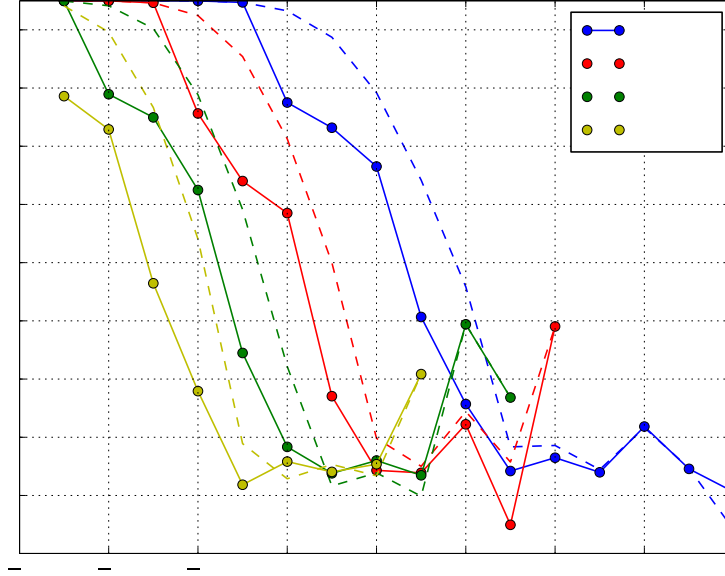


Figure 5.10: FER for the optimized 1s-1r-1d probabilistic protocol, DMF mode, no combining, as a function of the transmit E_s/N_o at the source and at the relay (marked lines). Dashed lines represent the FER achievable by the optimized deterministic protocol.

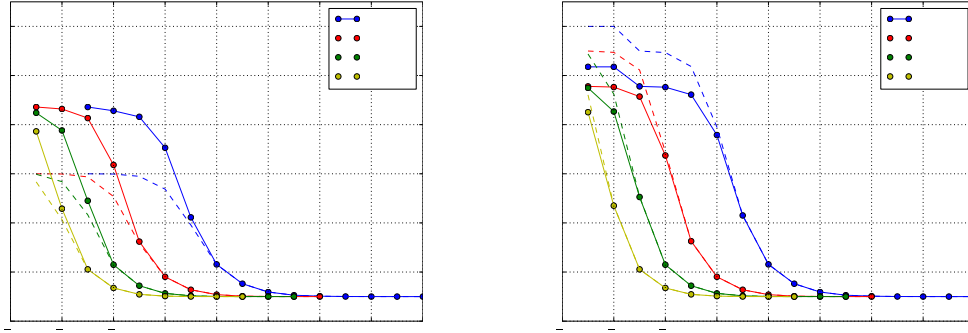


Figure 5.11: \bar{S} (left) and \bar{T} (right) for the optimized probabilistic 1s-1r-1d protocol, DMF mode, no combining, as a function of the transmit E_s/N_o at the source and at the relay (marked lines). Dashed lines represent the performance achievable by the optimized deterministic protocol.

The real nature of the design parameters in the probabilistic protocol provides more design freedom, and hence allows a better performance, visible above all in the decreased

achievable FER, over the deterministic protocol.

The results of this study, where we considered simple schemes only, serve as a proof of concept of feasibility in more complex configurations (three-nodes networks with DCF relay mode, and for multi-source, multi-relay networks), for which the approach described in Chapters 3 and 4 is not tractable.

Chapter 6

Conclusions and Future Work

This thesis had the goal of studying and analyzing the cooperative HARQ (C-HARQ) protocols at the MAC layer. Having been introduced in 1971 by Van der Meulen [24], the cooperative relay techniques have now become very useful, and a part of LTE-A standard. Relaying strategies can be extremely useful when the transmitter and the receiver are separated by a large distance. In the literature the cooperation can be classified depending on the relay behavior into the following groups: Amplify-and-Forward (AF) [55], Decode-and-Forward (DCF) [56–59], and Demodulate-and-Forward (DMF), [60]. The extensive advance in the relaying has driven the research to another technique, namely the Cooperative ARQ. It has been recently shown that the relay in cooperative networks, if used properly, can serve to obtain diversity gains [10], [11] [12] and throughput increase [1]. However, the existing works usually study either only cooperative ARQ (C-ARQ) [5,15,27,36]; or simply the effect of network codes in cooperation (without retransmissions) [11,12]. In this work we are questioning whether the relay is useful when we add both ARQ and channel codes. We provide extensive analysis, performance evaluation and optimization of cooperative HARQ protocols with/without network codes.

In Chapter 1 we give the literature overview and motivation for this work. Knowing that we are comparing schemes with and without ARQ, we derive a metric that provides for energy efficient analysis of these schemes. We demonstrate that this metric can be used for any network analysis (point-to-point, cooperative relay networks), and that it is more informative than the comparisons, existing in the literature. This metric is called the true E_b/N_0 and is derived in Chapter 2. With the help of the true E_b/N_0 the system designers can study and decide in which channel quality they are wasting the energy and assign it appropriately.

We ask ourselves the question whether it is possible to analyze the QoS of large cooperative networks with multiple sources, multiple relays and with retransmissions. To answer this, we first study C-HARQ in simplest networks that consist of one source, one relay and one destination in Chapter 3. In any multi-node network, it is crucial to be able to evaluate the QoS metrics. However there is no generalized framework existing, that would allow doing this in an efficient manner without having to derive expressions for each specific case. It is widely known that finite state machines (FSM) are simple tools to describe a mathematical model. In this thesis we develop this framework using FSM.

We show that the frame error rate, efficiency, delay, and the goodput can be obtained by computing the steady state vector of the FSM. We also show that the number of states in the FSM depends strongly on the definition of the states and the protocol. We perform a study on the number of states of the FSM and its relation to the transmissions number, we show that this approach also becomes too cumbersome. We do this demonstration using the two source, one relay and one destination scheme. So the question is: now that we know how to analyze QoS of a cooperative network, using the FSM, how can we reduce the number of states?

In order to provide a framework for any cooperative network with multiple nodes, we develop in Chapter 5 an equivalent probabilistic protocol, which gives the same performances as the original protocol. The corresponding FSM of the equivalent protocol can obtain a very small FSM, which is easy to analyze. Moreover, this equivalent FSM's number of states stays the same no matter how many retransmissions are present in the system. We had the goal of finding the most optimal combinations of PHY layer code rates and the number of transmissions granted per node in order to obtain optimal goodput with restraining the maximal value of the FER.

In beginning the Chapter 4 we used a proof of concept, that an equivalent FSM can be derived, in such a way, that the performance metrics are equal or better than those of the original FSM. We then extend this to the DMF protocol with one source, one relay and one destination. We compare the equivalent protocol metrics to those of the original, developed in Chapter 3. The QoS metrics that we obtain using the equivalent FSM analysis, are then optimized as follows: the goodput is maximized subject to a constraint on the FER. We study the achievable FER and goodput for given signal-to-noise ratio, by finding the most optimal code rates and transmissions credits at the source and at the relay. Moreover, we take into account the fact that the relay is situated in between the source and the destination, henceforth, the distance source-relay and relay-destination is twice smaller. As a consequence, the channels on those links are better than the direct source-destination link. We show that due to the relay it is possible to obtain better performances than in a single source single destination network.

The tools and analysis developed in this thesis can be used for extended networks with multiple sources and multiple relays (e.g. [10–12]) with (H)ARQ.

We proved in this thesis that the relay **is** useful, if exploited in a proper way, and we give the tools to measure this. The future works can study the effects of the relay when there is network coding, and when there are many nodes. Due to the extensive analysis of this work, we propose the tool that reduces the computational time of the QoS parameters.

More specifically, the following subjects must be treated:

- the results of the last chapter can be also studied for the DCF protocol in one source, one relay, one destination networks. This will help the designers to better understand the impact of the relay work mode on the cooperation in terms of the goodput, efficiency, delay and error rate.
- cooperative HARQ protocols for multi-source and one-relay networks QoS evaluations: with and without network coding. What will be the effect of network codes on the goodput and the FER.

- What is the best network code, that accounts for the error propagation, and in the meantime reduces the error probability?
- further studies on cooperative HARQ protocols for multi-source and multi-relay networks can be carried out and the effect of multiple relays can be exploited. More interestingly, are the retransmissions helping in the networks with multiple relays or network codes are sufficient.

All the above mentioned questions can be studied easily due to the framework developed in this thesis, that is generic and easily extendable for larger networks. The cooperative systems can then be compared to those without ARQ using the energy efficient metric that we developed. This measure can be useful in power allocation problems for multinode networks, where the energy consumption of the nodes has to be used in economic manner (e.e. in wireless sensor networks where the battery life of the nodes is limited).

Appendices

Appendix A

Evaluation of the elementary fragment error rate

In this Appendix we describe the procedure used in order to obtain the probability of decoding error π_{XY} of a fragment, after a single transmission on the physical channel X to Y.

Recall that the received modulated symbol at the destination can be expressed as

$$y_{XY} = h x + n, \quad (\text{A.1})$$

where x is the emitted modulated symbol, with average energy E_s , h is the Rayleigh fading coefficient, $h \sim \mathcal{CN}(0, 1)$, and n is the AWGN noise symbol, $n \sim \mathcal{CN}(0, N_o)$. We consider BPSK modulation. Coherent demodulation and perfect Channel State Information are assumed at the decoder. Before modulation and transmission each information fragment is encoded using a convolutional code of rate R . At the destination the coded fragment is soft demodulated, and the LLRs relative to the received symbols are fed to the Viterbi decoder.

As we detailed in Chapter 3, several choices can be made in the design of the communication protocol. In particular, the choice of the operation mode of the relay (DMF or DCF) and the choice whether to combine previously received fragments (combining vs no combining) affect the way in which the LLRs input at the decoder are evaluated.

A.1 No Combining

The destination computes symbol-by-symbol LLRs of the received sequence. Say destination received symbol y at i -th position. In order to derive the LLRs, it is considered that at destination there is always only one copy of the FRAG, which means that the soft output of the demodulator takes the LLR based on the gaussian pdf:

$$\begin{aligned} \Lambda_{\text{no combining}} &= \log(P(y|x)) \\ &= \log\left(\frac{\frac{1}{\sqrt{2\pi}\sigma_n} \exp(-\frac{|y+\sqrt{E_b}h|^2}{\sigma_n^2})}{\frac{1}{\sqrt{2\pi}\sigma} \exp(-\frac{|y-\sqrt{E_b}h|^2}{\sigma_n^2})}\right) \end{aligned} \quad (\text{A.2})$$

So, the destination receives the FRAGs and feeds the following LLRs to the Viterbi decoder:

$$\Lambda_{\text{no combining}} = ||y - \sqrt{E_b}h||^2 - ||y + \sqrt{E_b}h||^2 \quad (\text{A.3})$$

A.2 Demodulate-and-Forward with Combining

The symbol received at the relay from the source is given by Eq. 3.11. The relay demodulates the data as shown below:

$$\hat{x}_{SR} = \underset{x_m, m \in [1, \dots, M]}{\text{argmin}} ||y_{SR} - \sqrt{E_b}h_{SR}x_m||^2 \quad (\text{A.4})$$

The relay then performs BPSK modulation on the detected symbol and transmits it to destination:

$$y_{RD,k} = \sqrt{E_b}\hat{x}_{k,SR}h_{k,RD} + n_{k,RD} \quad (\text{A.5})$$

where the index k denotes the k -th combined symbol at the destination. Now let us denote the following event:

$$C = \begin{cases} 0, & \text{when the relay did not correctly demodulate the FRAG coming from the source,} \\ 1, & \text{when the relay correctly demodulated the FRAG coming from the source} \end{cases}$$

The event C is considered at the destination decoder, which computes the number of FRAGs received from the source (i.e. direct transmission), and the number of FRAGs coming from the relay (i.e. indirect transmission, that involves the event C in the computation).

Then, by denoting all the symbols received from source and relay at the destination at timeslot t by y , and considering the fact that these copies are i.i.d, the conditional pdf becomes:

$$P(y|x_m) = \prod_{i=1}^{T_S} P(y_{i,SD}|x_m) \prod_{k=1}^{T_R} P(y_{k,RD}|x_m) \quad (\text{A.6})$$

where the term T_S denotes the total number of the FRAGs that have been received from the source, and T_R the total number of FRAGs that have been received from the relay. Since the transmissions of the source and the relay are independent (i.e. the channels between the source-destination, and relay-destination are i.i.d.), we could represent the pdf of the received symbols sequence as a product of two pdfs: one representing the symbols received from the source, and another the symbols received from the relay.

In Eq. A.6 the term $P(\mathbf{y}_{SD}|x_m)$ is given by Gaussian pdf with Rayleigh fading, so we only need to find the term $P(\mathbf{y}_{RD}|x_m)$, which can be done using marginalization:

$$\begin{aligned} P(y_{k,RD}|x_m) &= \sum_{c \in \{0,1\}} P(y_{k,RD}|c, x_m)P(c|x_m) \\ &= P(y_{k,RD}|c=0, x_m)P(c=0|x_m) + P(y_{k,RD}|c=1, x_m)P(c=1|x_m) \end{aligned} \quad (\text{A.7})$$

$$P(c = 0|x_m) = Q\left(\sqrt{\frac{E_b||h_{k,SR}||^2}{N_0}}\right) \quad (\text{A.8})$$

$$P(c = 1|x_m) = 1 - Q\left(\sqrt{\frac{E_b||h_{k,SR}||^2}{N_0}}\right)$$

$$\begin{aligned} P(\mathbf{y}|x_m) &= \prod_{i=1}^{T_S} \frac{1}{\sqrt{2\pi}\sigma_n} \exp\left(-\frac{||y_{i,SD} - x_m h_{i,SD}||^2}{2\sigma_n^2}\right) \\ &\quad \prod_{k=1}^{T_R} P(y_{k,RD}|c = 0, x_m)P(c = 0|x_m) + P(y_{k,RD}|c = 1, x_m)P(c = 1|x_m) \\ &= \left(\frac{1}{\sqrt{2\pi}\sigma_n}\right)^{T_S} \prod_{i=1}^{T_S} \exp\left(-\frac{||y_{i,SD} - \sqrt{E_b}x_m h_{i,SD}||^2}{2\sigma_n^2}\right) \\ &\quad \left(\frac{1}{\sqrt{2\pi}\sigma_n}\right)^{T_R} \prod_{k=1}^{T_R} \exp(2\sigma_n^2) \left[\exp\left(-||y_{k,RD} - \sqrt{E_b}\tilde{x}_m h_{k,RD}||^2\right) Q\left(\sqrt{\frac{E_b||h_{k,SR}||^2}{N_0}}\right) \right. \\ &\quad \left. + \exp\left(-||y_{k,RD} - \sqrt{E_b}x_m h_{k,RD}||^2\right) \left(1 - Q\left(\sqrt{\frac{E_b||h_{k,SR}||^2}{N_0}}\right)\right) \right] \end{aligned} \quad (\text{A.9})$$

where the term \tilde{x}_m denotes the symbol which has been incorrectly detected at relay, and $E_b = 1$ the energy per symbol.

So, the LR becomes:

$$\begin{aligned} LR_{DMF} &= \prod_{i=1}^{T_S} \frac{e(-||y_{i,SD} + \sqrt{E_b}h_{i,SD}||^2)}{e(-||y_{i,SD} - \sqrt{E_b}h_{i,SD}||^2)} \\ &\quad \prod_{k=1}^{T_R} \frac{e(-||y_{k,RD} - \sqrt{E_b}h_{k,RD}||^2) Q\left(\sqrt{\frac{E_b||h_{k,SR}||^2}{N_0}}\right) + e(-||y_{k,RD} + \sqrt{E_b}h_{k,RD}||^2) \left(1 - Q\left(\sqrt{\frac{E_b||h_{k,SR}||^2}{N_0}}\right)\right)}{e(-||y_{k,RD} + \sqrt{E_b}h_{k,RD}||^2) Q\left(\sqrt{\frac{E_b||h_{k,SR}||^2}{N_0}}\right) + e(-||y_{k,RD} - \sqrt{E_b}h_{k,RD}||^2) \left(1 - Q\left(\sqrt{\frac{E_b||h_{k,SR}||^2}{N_0}}\right)\right)} \end{aligned} \quad (\text{A.10})$$

By taking the log of the expression in Eq. A.10, the soft LLR for the DMF technique is obtained as:

$$\begin{aligned} \Lambda_{DMF} &= \sum_{i=1}^{T_S} \left(||y_{i,SD} - \sqrt{E_b}h_{i,SD}||^2 - ||y_{i,SD} + \sqrt{E_b}h_{i,SD}||^2 \right) \\ &+ \sum_{k=1}^{T_R} \ln\left\{ \left(\exp\left(-||y_{k,RD} - \sqrt{E_b}h_{k,RD}||^2\right) - \exp\left(-||y_{k,RD} + \sqrt{E_b}h_{k,RD}||^2\right) \right) Q\left(\sqrt{\frac{E_b||h_{k,SR}||^2}{N_0}}\right) \right. \\ &+ \exp\left(-||y_{k,RD} + \sqrt{E_b}h_{k,RD}||^2\right) \} \\ &- \sum_{k=1}^{T_R} \ln\left\{ \left(\exp\left(-||y_{k,RD} + \sqrt{E_b}h_{k,RD}||^2\right) - \exp\left(-||y_{k,RD} - \sqrt{E_b}h_{k,RD}||^2\right) \right) Q\left(\sqrt{\frac{E_b||h_{k,SR}||^2}{N_0}}\right) \right. \\ &+ \exp\left(-||y_{k,RD} - \sqrt{E_b}h_{k,RD}||^2\right) \} \end{aligned} \quad (\text{A.11})$$

A.3 Decode-and-Forward with Combining

As explained in the previous sections, in the DCF protocol the relay forwards only the data that is correct, meaning that all the bits of the FRAG are received correctly at the relay. This is equivalent of sending the FRAG on a direct link to the destination.

At the i -th transmission the symbol received at D is:

$$y_i = h_i x_m + n_i \quad (\text{A.12})$$

where h_i and n_i are the received fading and noise coefficients of the i -th combined FRAG. Since the decoder processes the FRAGs in sequence (symbol by symbol), the time index is dropped, and each of the terms denotes one coefficient.

Since we have AWGN with Rayleigh fast fading, the received symbol copies are i.i.d., consequently, the receiver computes LLRs, based on:

$$\begin{aligned} P(\mathbf{y}|x_m) &= \\ &= \prod_{i=1}^N P(y_i|x_m) \\ &= \prod_{i=1}^N \frac{1}{\sqrt{2\pi}\sigma_n} \exp\left(-\frac{\|y_i - \sqrt{E_b}x_m h_i\|^2}{2\sigma_n^2}\right) \end{aligned} \quad (\text{A.13})$$

where σ_n is the noise variance.

The Likelihood Ratio is given by:

$$\begin{aligned} LR &= \prod_{i=1}^N \frac{\exp(-\|y_i + \sqrt{E_b}h_i\|^2)}{\exp(-\|y_i - \sqrt{E_b}h_i\|^2)} \\ &= \prod_{i=1}^N \exp\left(\|y_i - \sqrt{E_b}h_i\|^2 - \|y_i + \sqrt{E_b}h_i\|^2\right) \end{aligned} \quad (\text{A.14})$$

By taking the natural \log , the metric becomes:

$$\begin{aligned} \Lambda_{DCF_1} &= \log \prod_{i=1}^N \exp\left(\|y_i - \sqrt{E_b}h_i\|^2 - \|y_i + \sqrt{E_b}h_i\|^2\right) \\ &= \sum_{i=1}^N \left(\|y_i - \sqrt{E_b}h_i\|^2 - \|y_i + \sqrt{E_b}h_i\|^2\right) \end{aligned} \quad (\text{A.15})$$

The Viterbi decoder takes as input these LLR estimates for each symbol and performs the Viterbi decoding based on these soft inputs.

Appendix B

Evaluation of the elementary fragment error rate, multisource case

In this Appendix we extend the procedure to evaluate the LLRs to be fed to the Viterbi decoder at the destination. We consider here the case of multisource schemes, where network-coded packets may be forwarded by the relay.

B.1 DCF protocol

Let us assume that the destination has received T_1 FRAGs belonging to the source 1 (either from $S1$ or from the relay), T_2 FRAGs belonging to the source 2 (either from $S2$ or from the relay), and T_3 FRAGs coming from the relay and representing the XOR between the FRAGs of $S1$ and $S2$.

The joint demodulator evaluates the following (per symbol) likelihood:

$$p(\overline{y_1}, \overline{y_2}, \overline{w} | x_1, x_2) = \prod_{k=1}^{T_1} p(y_{1_k} | x_1) \prod_{j=1}^{T_2} p(y_{2_j} | x_2) \prod_{i=1}^{T_1} p(w_i | x_1, x_2) \quad (\text{B.1})$$

where y_i is the vector of the observed direct copies of the active symbol x_i , and w is the vector of the observed network coded packet.

The channel decoder of source 1 needs, as input, the likelihood ratio relative to the likelihood

$$p(\overline{y_1}, \overline{y_2}, \overline{w} | x_1) \quad (\text{B.2})$$

which is evaluated as follows:

$$p(\overline{y_1}, \overline{y_2}, \overline{w}|x_1) = \sum_{x_2 \in \{-1, 1\}} p(y_1, y_2, w, x_2|x_1) \quad (\text{B.3a})$$

$$= \sum_{x_2 \in \{-1, 1\}} p(y_1, y_2, w|x_1, x_2)p(x_2|x_1) \quad (\text{B.3b})$$

$$= \frac{1}{2} \prod_{k=1}^{T_1} p(y_{1_k}|x_1) \sum_{x_2 \in \{-1, 1\}} \left(\prod_{j=1}^{T_2} p(y_{2_j}|x_2) \prod_{i=1}^{T_3} p(w_i|x_1, x_2) \right) \quad (\text{B.3c})$$

where Eq. (B.3c) is obtained by substitution of Eq.(B.1) in Eq.(B.3b), and noticing that $p(x_2|x_1) = p(x_2) = 1/2$. The probability distribution of the output of the decoder can be then evaluated as a function of the parameters T_1 ; T_2 ; T_3 .

Using the modulation given in Section ??, we can easily obtain the respective LLRs for the source 1 and source 2 decoders.

The LLR of source 1 is computed the following way:

$$\begin{aligned} LLR_{S_1} &= \log \left(\frac{p(y_1, y_2, w|x_1 = -1)}{p(y_1, y_2, w|x_1 = 1)} \right) \\ &= \log \left(\frac{\prod_{k=1}^{T_1} p(y_{1_k}|x_1 = -1) \sum_{x_2 \in \{-1, 1\}} \left(\prod_{j=1}^{T_2} p(y_{2_j}|x_2) \prod_{i=1}^{T_3} p(w_i|x_1 = -1, x_2) \right)}{\prod_{k=1}^{T_1} p(y_{1_k}|x_1 = 1) \sum_{x_2 \in \{-1, 1\}} \left(\prod_{j=1}^{T_2} p(y_{2_j}|x_2) \prod_{i=1}^{T_3} p(w_i|x_1 = 1, x_2) \right)} \right) \end{aligned} \quad (\text{B.4})$$

Since the channel is Rayleigh fully interleaved fading with AWGN, we can easily obtain

the closed-form expression of the decoder of source 1:

$$\begin{aligned}
LLR_{S_1} &= \log \left(\prod_{k=1}^{T_1} \frac{p(y_{1_k}|x_1 = -1)}{p(y_{1_k}|x_1 = 1)} \right) \\
&+ \log \left(\frac{\sum_{x_2 \in \{-1,1\}} \left(\prod_{j=1}^{T_2} p(y_{2_j}|x_2) \prod_{i=1}^{T_3} p(w_i|x_1 = -1, x_2) \right)}{\sum_{x_2 \in \{-1,1\}} \left(\prod_{j=1}^{T_2} p(y_{2_j}|x_2) \prod_{i=1}^{T_3} p(w_i|x_1 = 1, x_2) \right)} \right) \\
&= \sum_{k=1}^{T_1} \log \left(\frac{\frac{1}{\sqrt{2\pi}\sigma_n} \exp \left(-\frac{\|y_{1_k} + h_{1_k}\|^2}{2\sigma_n^2} \right)}{\frac{1}{\sqrt{2\pi}\sigma_n} \exp \left(-\frac{\|y_{1_k} - h_{1_k}\|^2}{2\sigma_n^2} \right)} \right) \\
&+ \log \left(\frac{\sum_{x_2 \in \{-1,1\}} \left(\prod_{j=1}^{T_2} \frac{1}{\sqrt{2\pi}\sigma_n} \exp \left(-\frac{\|y_{2_j} - x_2 h_{2_j}\|^2}{2\sigma_n^2} \right) \prod_{i=1}^{T_3} \frac{1}{\sqrt{2\pi}\sigma_n} \exp \left(-\frac{\|w_i - (1-2(1 \oplus x_2))h_{3_i}\|^2}{2\sigma_n^2} \right) \right)}{\sum_{x_2 \in \{-1,1\}} \left(\prod_{j=1}^{T_2} \frac{1}{\sqrt{2\pi}\sigma_n} \exp \left(-\frac{\|y_{2_j} - x_2 h_{2_j}\|^2}{2\sigma_n^2} \right) \prod_{i=1}^{T_3} \frac{1}{\sqrt{2\pi}\sigma_n} \exp \left(-\frac{\|w_i - (1-2(0 \oplus x_2))h_{3_i}\|^2}{2\sigma_n^2} \right) \right)} \right) \\
&= \sum_{k=1}^{T_1} \left(\frac{\|y_{1_k} - h_{1_k}\|^2}{2\sigma_n^2} - \frac{\|y_{1_k} + h_{1_k}\|^2}{2\sigma_n^2} \right) \\
&+ \log \left(\frac{\sum_{x_2 \in \{-1,1\}} \left(\prod_{j=1}^{T_2} \exp \left(-\frac{\|y_{2_j} - x_2 h_{2_j}\|^2}{2\sigma_n^2} \right) \prod_{i=1}^{T_3} \exp \left(-\frac{\|w_i - (1-2(1 \oplus x_2))h_{3_i}\|^2}{2\sigma_n^2} \right) \right)}{\sum_{x_2 \in \{-1,1\}} \left(\prod_{j=1}^{T_2} \exp \left(-\frac{\|y_{2_j} - x_2 h_{2_j}\|^2}{2\sigma_n^2} \right) \prod_{i=1}^{T_3} \exp \left(-\frac{\|w_i - (1-2(0 \oplus x_2))h_{3_i}\|^2}{2\sigma_n^2} \right) \right)} \right) \quad (B.5)
\end{aligned}$$

The derivation of the source 2 LLR is similar, so we directly give the closed-form expression below:

$$\begin{aligned}
LLR_{S_2} &= \sum_{j=1}^{T_2} \left(\frac{\|y_{2_j} - h_{2_j}\|^2}{2\sigma_n^2} - \frac{\|y_{2_j} + h_{2_j}\|^2}{2\sigma_n^2} \right) \\
&+ \log \left(\frac{\sum_{x_1 \in \{-1,1\}} \left(\prod_{k=1}^{T_1} \exp \left(-\frac{\|y_{1_k} - x_1 h_{1_k}\|^2}{2\sigma_n^2} \right) \prod_{i=1}^{T_3} \exp \left(-\frac{\|w_i - (1-2(1 \oplus x_1))h_{3_i}\|^2}{2\sigma_n^2} \right) \right)}{\sum_{x_1 \in \{-1,1\}} \left(\prod_{k=1}^{T_1} \exp \left(-\frac{\|y_{1_k} - x_1 h_{1_k}\|^2}{2\sigma_n^2} \right) \prod_{i=1}^{T_3} \exp \left(-\frac{\|w_i - (1-2(0 \oplus x_1))h_{3_i}\|^2}{2\sigma_n^2} \right) \right)} \right) \quad (B.6)
\end{aligned}$$

B.2 DCF protocol

Let us consider the derivation of S_1 decoder, which can then be easily derived for the S_2 . Our task is to obtain the conditional pdf $p(\bar{y}_{S_1D}, \bar{y}_{S_2D}, \bar{y}_{S_1RD}, \bar{y}_{S_2RD}, \bar{w}|x_1)$ in order to compute the LLR in Eq. (4.48). So we have to derive this term from the $p(\bar{y}_{S_1D}, \bar{y}_{S_2D}, \bar{y}_{S_1RD}, \bar{y}_{S_2RD}, \bar{w}|x_1, x_2, a, b)$.

First of all, due to the fact that the channel is memoryless, which consequently makes the observed symbols independent from each other, the joint conditional pdf can be represented as a product of pdfs:

$$\begin{aligned}
p(\bar{y}_{S_1D}, \bar{y}_{S_2D}, \bar{y}_{S_1RD}, \bar{y}_{S_2RD}, \bar{w}|x_1, x_2) &= \prod_{i=1}^{T_1} p(y_{S_1D,i}|x_1) \prod_{j=1}^{T_2} p(y_{S_2D,j}|x_2) \\
&\cdot \sum_{a \in \{0,1\}} \prod_{k=1}^{T_3} p(y_{S_1RD,k}|x_1, a) p(a|x_1) \\
&\cdot \sum_{b \in \{0,1\}} \prod_{n=1}^{T_4} p(y_{S_2RD,n}|x_2, b) p(b|x_2) \\
&\cdot \sum_{a,b \in \{00,\dots,11\}} \prod_{m=1}^{T_5} p(w_m|x_1, x_2, a, b) p(a, b|x_1, x_2)
\end{aligned}$$

Then, in order to obtain the conditional pdf for the S_1 decoder at the destination, the term obtained in Eq. B.7 is marginalized over all values of x_2 :

$$\begin{aligned}
p(\bar{y}_{S_1D}, \bar{y}_{S_2D}, \bar{y}_{S_1RD}, \bar{y}_{S_2RD}, \bar{w}|x_1) &= \sum_{x_2 \in \{0,1\}} p(\bar{y}_{S_1D}, \bar{y}_{S_2D}, \bar{y}_{S_1RD}, \bar{y}_{S_2RD}, \bar{w}|x_1, x_2) p(x_2|x_1) \\
&= \sum_{x_2 \in \{0,1\}} p(\bar{y}_{S_1D}, \bar{y}_{S_2D}, \bar{y}_{S_1RD}, \bar{y}_{S_2RD}, \bar{w}|x_1, x_2) p(x_2) \\
&= \sum_{x_2 \in \{0,1\}} p(x_2) \left[\prod_{i=1}^{T_1} p(y_{S_1D,i}|x_1) \prod_{j=1}^{T_2} p(y_{S_2D,j}|x_2) \right. \\
&\cdot \sum_{a \in \{0,1\}} \prod_{k=1}^{T_3} p(y_{S_1RD,k}|x_1, a) p(a|x_1) \\
&\cdot \sum_{b \in \{0,1\}} \prod_{n=1}^{T_4} p(y_{S_2RD,n}|x_2, b) p(b|x_2) \\
&\cdot \left. \sum_{a,b \in \{00,\dots,11\}} \prod_{m=1}^{T_5} p(w_m|x_1, x_2, a, b) p(a, b|x_1, x_2) \right] \quad (\text{B.8})
\end{aligned}$$

where the terms $p(a|x_1)$, $p(b|x_2)$, and $p(a, b|x_1, x_2)$ are given by Eq. (B.9), (B.10), and (B.11), respectively. They are based on the Gaussian Q function computations, channel properties:

$$\begin{aligned}
p(a = 0|x_1) &= Q\left(\sqrt{\frac{2E_b||h_{S_1R}||^2}{N_0}}\right) \\
p(a = 1|x_1) &= 1 - Q\left(\sqrt{\frac{2E_b||h_{S_1R}||^2}{N_0}}\right)
\end{aligned} \quad (\text{B.9})$$

In a similar way the conditional probability for the random variable b is given:

$$\begin{aligned} p(b=0|x_2) &= Q\left(\sqrt{\frac{2E_b||h_{S_2R}||^2}{N_0}}\right) \\ p(b=1|x_2) &= 1 - Q\left(\sqrt{\frac{2E_b||h_{S_2R}||^2}{N_0}}\right) \end{aligned} \quad (\text{B.10})$$

$$\begin{aligned} p(a=0, b=0|x_1, x_2) &= Q\left(\sqrt{\frac{2E_b||h_{S_1R}||^2}{N_0}}\right) Q\left(\sqrt{\frac{2E_b||h_{S_2R}||^2}{N_0}}\right) \\ p(a=0, b=1|x_1, x_2) &= Q\left(\sqrt{\frac{2E_b||h_{S_1R}||^2}{N_0}}\right) \left(1 - Q\left(\sqrt{\frac{2E_b||h_{S_2R}||^2}{N_0}}\right)\right) \\ p(a=1, b=0|x_1, x_2) &= \left(1 - Q\left(\sqrt{\frac{2E_b||h_{S_1R}||^2}{N_0}}\right)\right) Q\left(\sqrt{\frac{2E_b||h_{S_2R}||^2}{N_0}}\right) \\ p(a=1, b=1|x_1, x_2) &= \left(1 - Q\left(\sqrt{\frac{2E_b||h_{S_1R}||^2}{N_0}}\right)\right) \left(1 - Q\left(\sqrt{\frac{2E_b||h_{S_2R}||^2}{N_0}}\right)\right) \end{aligned} \quad (\text{B.11})$$

The terms h_{S_1R} and h_{S_2R} are assumed to be known at the destination.

Now, in order to obtain the closed-form of the S_1 decoder, we inject the result of Eq. (B.8) into (4.48):

$$\begin{aligned} \text{LLR}_{S_1} &= \log\left(\frac{\sum_{x_2 \in \{0,1\}} p(x_2) \left[\prod_{i=1}^{T_1} p(y_{S_1D,i}|x_1=1) \prod_{j=1}^{T_2} p(y_{S_2D,j}|x_2) \sum_{a \in \{0,1\}} \prod_{k=1}^{T_3} p(y_{S_1RD,k}|x_1=1, a) p(a|x_1=1) \right]}{\sum_{x_2 \in \{0,1\}} p(x_2) \left[\prod_{i=1}^{T_1} p(y_{S_1D,i}|x_1=-1) \prod_{j=1}^{T_2} p(y_{S_2D,j}|x_2) \sum_{a \in \{0,1\}} \prod_{k=1}^{T_3} p(y_{S_1RD,k}|x_1=-1, a) p(a|x_1=-1) \right]} \right. \\ &\quad \cdot \frac{\sum_{b \in \{0,1\}} \prod_{n=1}^{T_4} p(y_{S_2RD,n}|x_2, b) p(b|x_2) \sum_{a, b \in \{00, \dots, 11\}} \prod_{m=1}^{T_5} p(w_m|x_1=-1, x_2, a, b) p(a, b|x_1=1, x_2)}{\sum_{b \in \{0,1\}} \prod_{n=1}^{T_4} p(y_{S_2RD,n}|x_2, b) p(b|x_2) \sum_{a, b \in \{00, \dots, 11\}} \prod_{m=1}^{T_5} p(w_m|x_1=1, x_2, a, b) p(a, b|x_1=-1, x_2)} \Big) \end{aligned} \quad (\text{B.12})$$

Notation Meaning

N_S	Number of transmissions granted per source
N_R	Number of transmissions granted per relay
M	Number of constellation symbols
K	FEC constraint length
R_c	FEC Rate
R_m	Modulation rate
L_{FRAG}	Number of information bits in FRAG
L_C	Number of encoded bits in FRAG, $L_C = L_F/R_c$
L_M	Number of bits in encoded and modulated FRAG, $L_C = L_F R_m/R_c$
B	Information bits in FRAG, $B = [b_1, b_2, \dots, b_{L_F}]$
C	Encoded FRAG bits, $C = [c_1, c_2, \dots, c_{L_C}]$
X	Set of symbols belonging to the constellation $X = [x_1, \dots, x_M]$
Y	Set of the received symbols at the t -th timeslot at the i -th transmission $Y = [y_{t_1}, y_{t_2}, \dots, y_{t_N}]$
Λ	Set of the soft estimated of the MAP demodulator at the t -th timeslot at the i -th transmission $\Lambda = [\Lambda_{t_1}, \Lambda_{t_2}, \dots, \Lambda_{t_N}]$
n_i	AWGN noise sample at the t -th timeslot of the i -th received copy of the FRAG
h_i	Rayleigh fading coefficient at the t -th timeslot of the i -th received copy of the FRAG
σ_R^2	Rayleigh fading coefficient variance
μ_n	AWGN coefficient mean
σ_n^2	AWGN coefficient variance
PCK	IP layer packet
$FRAG$	MAC layer fragment, encoded with convolutional codes of rate R_c

Bibliography

- [1] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, “Network Information Flow,” *IEEE Transactions on Information Theory*, vol. 46, 2000.
- [2] S. Lin and D. Costello, *Error Control Coding: Fundamentals and Applications*. Prentice Hall, 1983.
- [3] T. Ho and et. al, “The benefits of coding over routing in a randomized setting,” *Proceedings of IEEE ISIT, Yokohama, Japan*, p. 442, 2003.
- [4] T. Ho and et. al, “A random linear network coding approach to multicast,” *IEEE Transactions on Information Theory*, vol. 52, no. 10, pp. 4413–4430, 2006.
- [5] S. Katti and et al., “Xors in the air: Practical wireless network coding,” *IEEE/ACM Trans. Networking*, vol. 16, no. 3, pp. 497–510, 2008.
- [6] S. Katti, S. Gollakota, and D. Katabi, “Network coded wireless architecture,” *Ph.D. Dissertation, Massachusetts Institute of Technology*, 2008.
- [7] M. Xiao and M. Skoglund, “Design of network codes for multiple-user multiple-relay wireless networks,” in *Proceedings of IEEE International Symposium on Information Theory (ISIT)*, pp. 2562–2566, June 2009.
- [8] M. Xiao and M. Skoglund, “M-user cooperative wireless communication based on nonbinary network codes,” in *Proceedings of IEEE Information Theory Workshop, ITW*, pp. 316–320, June 2009.
- [9] J. L. Rebelatto, B. F. Uchôa-Filho, Y. Li, and B. Vucetic, “Generalized distributed network coding based on nonbinary linear block codes for multi-user cooperative communications,” in *Available at: <http://arxiv.org/abs/1004.2757>*, December 2010.
- [10] M. Iezzi, M. D. Renzo, and F. Graziosi, “Network Code Design from Unequal Error Protection Coding: Channel Aware Receiver Design and Diversity Analysis,” in *IEEE International Conference on Communications (ICC)*.
- [11] M. Iezzi, M. D. Renzo, and F. Graziosi, “Closed Form Error Probability of Network Coded Cooperative Wireless Networks with ChannelAware Detectors,” in *Global Telecommunications Conference (GLOBECOM 2011), IEEE*, December 2011.

- [12] M. Iezzi, M. D. Renzo, and F. Graziosi, *Flexible Network Codes Design for Cooperative Diversity*.
- [13] R. Koetter and M. Medard, "An algebraic approach to network coding," *IEEE/ACM Trans. Networking*, vol. 11, no. 5, pp. 782–795, 2003.
- [14] S. Katti, S. Gollakota, and D. Katabi, "Symbol-level network coding for wireless mesh networks," *Proceedings of ACM SIGCOMM, Seattle, USA*, vol. 16, no. 3, pp. 401–412, 2008.
- [15] S. Katti, S. Gollakota, and D. Katabi, "Embracing wireless interference: Analog network coding," *Proceedings of ACM SIGCOMM, Kyoto, Japan*, pp. 397–408, 2007.
- [16] R. Koetter and F. Kschischang, "Coding for errors and erasures in random network coding," *IEEE Transactions on Information Theory*, vol. 54, no. 8, pp. 3579–3591, 2008.
- [17] T. Ho, M. Médard, J. Shi, M. Effros, and D. Karger, "On randomized network coding," *41st Annual Allerton Conference on Communication, Control, and Computing*, October 2003.
- [18] H. X. Nguyen, H. H. Nguyen, and T. Le-Ngoc, "Signal transmission with unequal error protection in wireless relay networks," *IEEE Trans. Veh. Technol.*, pp. 2166–2178, 2010.
- [19] S. M. Alamouti, "A Simple Transmit Diversity Technique for Wireless Communications," *IEEE Journal on selected areas in Communication*, vol. 16, October 1998.
- [20] A. L. Duc, C. L. Martret, and P. Ciblat, "Packet-Error Rate and Efficiency Closed-Form Expressions for Cross-Layer Hybrid ARQ schemes," 2009.
- [21] A. L. Duc, C. L. Martret, and P. Ciblat, "Delay and Jitter Closed-Form Expressions for Cross-Layer Hybrid ARQ Schemes," 2009.
- [22] E. Zimmermann, P. Herhold, and G. Fettweis, "On the performance of cooperative relaying protocols in wireless networks," *European Transactions on Telecommunications (ETT)*, vol. 16, pp. 17–35, January-February 2005.
- [23] M. Dohler and Y. Li, *Cooperative Communications: Hardware, Channel and PHY*. February 2010.
- [24] E. C. V. D. Meulen, "Three-terminal communication channels," *Advances in Applied Probability*, vol. 3, pp. 120–154, Spring 1971.
- [25] R. W. Yeung, "Multilevel diversity coding with distortion," *IEEE Transactions on Information Theory*, vol. 41, pp. 412–422, March 1995.
- [26] R. W. Yeung and Z. Zhang, "On symmetrical multilevel diversity coding," *IEEE Transactions on Information Theory*, vol. 45, pp. 609–621, March 1999.

- [27] W. Ni, Z. Chen, and I. B. Collings, "Cooperative Hybrid ARQ in Wireless Decode and Forward Relay Networks," in *IEEE 71st Vehicular Technology Conference (VTC)*, Spring 2009.
- [28] M. Xiao, M. Médard, and T. Aulin, "A binary coding approach for combination networks and general erasure networks," June 2007.
- [29] S. Maheshwar, Z. Li, and B. Li, "Bounding the coding advantage of combination network coding in undirected networks," *IEEE Transactions on Information Theory*, vol. 58, February 2012.
- [30] T. X. Vu, Q. B. V. Nguyen, M. D. Renzo, and P. Duhamel, "Performance analysis of relay networks with channel code in low snr regime," in *IEEE 14th Workshop on Signal Processing Advances in Wireless Communications*, June 2013.
- [31] A. S. Avestimehr and D. N. C. Tse, "Outage capacity of the fading relay channel in the low-snr regime," in *IEEE Transactions on Information Theory*, vol. 53, p. 14011415, April 2007.
- [32] K. Tourki, H. Yang, and M. Alouini, "Accurate outage analysis of incremental decode-and-forward opportunistic relaying," in *IEEE Transactions on Wireless Communications*, vol. 10, p. 1021–1025, April 2011.
- [33] T. X. Vu, M. D. Renzo, and P. Duhamel, "Optimal and low complexity iterative joint network/channel decoding for the multiple access relay networks," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3312–3315, May 2011.
- [34] C. Hausl and J. Hagenauer, "Iterative network and channel decoding for the two-way relay channel," in *In Proc. of IEEE International Conference on Communications*, pp. 1568–1573, 2006.
- [35] R. Thobaben, "Joint network/channel coding for multi-user hybrid-arq," in *In Proc. Int. ITG Conf. Source and Channel Coding*, pp. 1–6, 2008.
- [36] Z. Zhang, T. Lv, X. Su, and H. Gao, "Dual xor in the air: A network coding based retransmission scheme for wireless broadcasting," pp. 1–6, June 2011.
- [37] Y. Sun, Y. Li, and X. Wang, "Cooperative Hybrid ARQ Protocol with Network Coding," in *Fourth International Conference on Communications and Networking in China (ChinaCOM)*, pp. 1–5, August 2009.
- [38] D. Nguyen, T. Tran, T. Nguyen, and B. Bose, "Wireless broadcast using network coding," *IEEE Transactions on Vehicular Technology*, vol. 58, p. 914925, February.
- [39] A. Goldsmith and S. Wicker, *Design challenges for energy constrained ad-hoc wireless networks*, vol. 9. August 2002.

- [40] M. Vuran and I. Akyildiz, "Cross-layer analysis of error control in wireless sensor networks," *SECON 06: Sensor and Ad Hoc Communications and Networks*, vol. 2, p. 585594, September 2006.
- [41] M. C. Vuran and I. F. Akyildiz, "Error Control in Wireless Sensor Networks: A Cross Layer Analysis," *IEEE/ACM Transactions on Networking*, vol. 17, August 2009.
- [42] C. G. Lott, O. Milenkovic, and E. Soljanin, "Hybrid ARQ: theory, state of the art and future directions," in *Proc. of IEEE Inf. Theory Workshop on Information Theory for Wireless Networks*, (Solstrand, Norway), pp. 1–5, 2007.
- [43] P. Wu and N. Jindal, "Coding versus ARQ in fading channels: how reliable should the PHY be?," *IEEE Trans. on Communications*, vol. 59, no. 12, pp. 3363–3374, 2011.
- [44] E. Telatar, "Capacity of multi-antenna gaussian channels," *AT&T-Bell Labs, Tech. Rep.*, 1995.
- [45] G. J. Foschini and M. Gans, "On the limits of wireless communication in a fading environment when using multiple antennas," *Wireless Personal Communications*, vol. 6, pp. 311–335, 1998.
- [46] H. E. Gamal, G. Caire, and M. O. Damen, "The MIMO ARQ channel: diversity-multiplexing-delay tradeoff," *IEEE Trans. on Information Theory*, vol. 52, no. 8, 2006.
- [47] S. M. Alamouti, "A simple transmit diversity technique for wireless communications," *IEEE J. on Selected Areas in Communications*, vol. 16, no. 8, pp. 1451–1458, 1998.
- [48] V. Tarokh, H. Jafarkhani, and A. R. Calderbank, "Space-Time Block Coding for wireless communications: performance results," *IEEE J. on Selected Areas in Communications*, vol. 17, March 1999.
- [49] Juan J. Alcaraz and Joan Garcia Haro, "Performance Evaluation of Multiple-Relay Cooperative ARQ Strategies for Mobile Networks," 2009.
- [50] Lixiang Xiong, Lavy Libman, and Guoqiang Mao, "Optimal Strategies for Cooperative MAC-Layer Retransmission in Wireless Networks," pp. 1495–1500, 2008.
- [51] Mehrdad Dianati, Xinhua Ling, Kshirasagar Naik, and Xuemin (Sherman) Shen, "A Node-Cooperative ARQ Scheme for Wireless Ad Hoc Networks," *IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY*, vol. 55, May 2006.
- [52] D. Lee and M. Yannakakis, "Principles and methods of testing finite state machines," *Proceedings of the IEEE*, vol. 84, pp. 1090 – 1123, August 1996.
- [53] H. M. Taylor and S. Karlin, *An Introduction To Stochastic Modeling*. Academic Press, 2005.
- [54] J. Proakis, *Digital Communications*. McGraw-Hill Companies, 1989.

- [55] S. Agnihotri, S. Jaggi, and M. Chen, "Amplify-and-forward in wireless relay networks," *IEEE Transactions on Information Theory*, vol. 53, pp. 3302 – 3318, 2007.
- [56] J. Zhao, M. Kuhn, A. Wittneben, and G. Bauch, "Cooperative transmission schemes for decode-and-forward relaying," 2007.
- [57] T. Wang, A. Cano, G. B. Giannakis, and J. N. Laneman, "High-performance cooperative demodulation with decode-and-forward relays," *IEEE Transactions on Communications*, vol. 55, pp. 1427 – 1438, 2007.
- [58] M. C. Ju and I. Kim, "Ml performance analysis of the decodeandforward protocol in cooperative diversity networks," *IEEE Transactions on Wireless Communications*, vol. 8, p. 3855 – 3867, 2009.
- [59] D. Liang, S. X. Ng, and L. Hanzo, "Relayinduced error propagation reduction for decodeandforward cooperative communications," *IEEE GLOBECOM*, vol. 8, pp. 1 – 5, 2010.
- [60] W. H. Alouane, N. Hamdi, and S. Meherzi, "Accurate bep of adaptive demodulate-and-forward relaying over rayleigh fading channels," *IEEE Symposium on Computers and Communications (ISCC)*, 2012.