



THÈSE

Pour obtenir le grade de

**DOCTEUR DE LA COMMUNAUTE UNIVERSITE
GRENOBLE ALPES**

Spécialité : **Nano Electronique Nano Technologies**

et de :

DOCTEUR (Ph.D) DE L'UNIVERSITE de YAOUNDE 1

Spécialité : **Informatique**

**préparée dans le cadre d'une cotutelle entre la
Communauté Université Grenoble Alpes et
l'Université de Yaoundé 1**

Arrêté ministériel : le 06 janvier 2005 - 07 août 2006 (France)
le 23 septembre 1999 - 11 avril 2001 (Cameroun)

Présentée par

TCHUANI TCHAKONTE Diane

Thèse codirigée par

Emmanuel SIMEU et Maurice TCHUENTE

préparée au sein des Laboratoires **TIMA** et **UMMISCO**
dans les Écoles Doctorales **ED EEATS** et **CRFD STG**

Minimisation de la consommation d'énergie des réseaux de capteurs dans les applications de couverture de cibles

Thèse soutenue publiquement le « **19/07/2019** »,
devant le jury composé de :

M. Médard FOGUE

Professeur IUT/FV Université de Dschang, Président

M. Emmanuel TANYI

Professeur Université de Buea, Rapporteur

M. KOSAI RAOOF

Professeur Université du Maine le Mans, Rapporteur

M. René NDOUNDAM

Maître de Conférences Université de Yaoundé 1, Examineur

M. Thomas DJOTIO

Maître de Conférences ENSP – Université de Yaoundé 1, Examineur

M. Serge STINCKWICH

Maître de Conférences Université de Caen Normandie, Examineur

M. Emmanuel SIMEU

Maître de Conférences HDR Université Grenoble Alpes, Co-directeur de thèse

M. Maurice TCHUENTE

Professeur Université de Yaoundé 1, Co-directeur de thèse



UNIVERSITÉ DE YAOUNDÉ I Faculté des Sciences Division de la Programmation et du Suivi des Activités Académiques		THE UNIVERSITY OF YAOUNDE I Faculty of Science Division of Programming and Follow-up of Academic Affairs
LISTE DES ENSEIGNANTS PERMANENTS		LIST OF PERMANENT TEACHING STAFF

ANNÉE ACADEMIQUE 2017/2018

(Par Département et par Grade)

DATE D'ACTUALISATION : 5 Octobre 2018

ADMINISTRATION

DOYEN : AWONO ONANA Charles, *Professeur*

VICE-DOYEN / DPSAA : DONGO Etienne, *Professeur*

VICE-DOYEN / DSSE : AJEAGAH Gidéon AGHAINDUM, *Maître de Conférences*

VICE-DOYEN / DRC : ABOSSOLO Monique, *Maitre de Conférences*

Chef Division Administrative et Financière : NDOYE FOE Marie C. F., *Maitre de Conférences*

Chef Division des Affaires Académiques, de la Scolarité et de la Recherche MBAZE MEVA'A
 Luc Léonard, *Maitre de Conférences*

1- DÉPARTEMENT DE BIOCHIMIE (BC) (41)			
N°	NOMS ET PRÉNOMS	GRADE	OBSERVATIONS
1	FEKAM BOYOM Fabrice	Professeur	En poste
2	MBACHAM FON Wilfried	Professeur	En poste
3	MOUNDIPA FEWOU Paul	Professeur	Chef de Département
4	NINTCHOM PENLAP V. épouse BENG	Professeur	En poste
5	OBEN Julius ENYONG	Professeur	En poste
6	ATOGHO Barbara Mma	Maître de Conférences	En poste
7	BELINGA née NDOYE FOE M. C. F.	Maître de Conférences	Chef DAF / FS
8	BIGOGA DIAGA Jude	Maître de Conférences	En poste
9	BOUDJEKO Thaddée	Maître de Conférences	En poste
10	EFFA NNOMO Pierre	Maître de Conférences	En poste
11	FOKOU Elie	Maître de Conférences	En poste
12	KANSCI Germain	Maître de Conférences	En poste
13	NANA Louise épouse WAKAM	Maître de Conférences	En poste
14	NGONDI Judith Laure	Maître de Conférences	En poste
15	NGUEFACK Julienne	Maître de Conférences	En poste
16	NJAYOU Frédéric Nico	Maître de Conférences	En poste
17	ACHU Merci BIH	Chargée de Cours	En poste
18	DEMMANO Gustave	Chargé de Cours	En poste
19	DJOKAM TAMO Rosine	Chargée de Cours	En poste
20	DJUIDJE NGOUNOUE Marcelline	Chargée de Cours	En poste
21	DJUIKWO NKONGA Ruth Viviane	Chargée de Cours	En poste
22	EVEHE BEBANDOUE Marie-Solange	Chargée de Cours	En poste
23	EWANE Cécile Anne	Chargée de Cours	En poste
24	KOTUE KAPTUE Charles	Chargé de Cours	En poste
25	LUNGA Paul KEILAH	Chargé de Cours	En poste
26	MBONG ANGIE M. Mary Anne	Chargée de Cours	En poste

27	MOFOR née TEUGWA Clotilde	Chargée de Cours	Inspecteur de Service MINESUP
28	NJAYOU Frédéric Nico	Chargé de Cours	En poste
29	Palmer MASUMBE NETONGO	Chargé de Cours	En poste
30	TCHANA KOUATCHOUA Angèle	Chargée de Cours	En poste
31	PACHANGOU NSANGO Sylvain	Chargé de Cours	En poste
32	DONGMO LEKAGNE Joseph Blaise	Chargé de Cours	En poste
33	FONKOUA Martin	Chargé de Cours	En poste
34	BEBOY EDZENGUELE Sara Nathalie	Chargée de Cours	En poste
35	DAKOLE DABOY Charles	Chargée de Cours	En poste
36	MANANGA Marlyse Joséphine	Chargée de Cours	En poste
37	MBOUCHE FANMOE Marceline Joëlle	Assistante	En poste
38	BEBEE Fadimatou	Assistante	En poste
39	TIENTCHEU DJOKAM Leopold	Assistant	En poste

2- DÉPARTEMENT DE BIOLOGIE ET PHYSIOLOGIE ANIMALES (BPA) (44)			
1	BILONG BILONG Charles-Félix	Professeur	Chef de Département
2	DIMO Théophile	Professeur	En Poste
3	DJIETO LORDON Champlain	Professeur	En poste
4	ESSOMBA née NTSAMA MBALA	Professeur	<i>VDoyen/FMSB/UYI</i>
5	FOMENA Abraham	Professeur	En Poste
6	KAMTCHOUING Pierre	Professeur	EN POSTE
7	NJAMEN Dieudonné	Professeur	En poste
8	NJIOKOU Flobert	Professeur	En Poste
9	NOLA Moïse	Professeur	En poste
10	TAN Paul VERNYUY	Professeur	En poste
11	TCHUEM TCHUENTE Louis Albert	Professeur	<i>Coord. Progr. MINSANTE</i>
12	AJEAGAH Gidéon AGHAINDUM	Maître de Conférences	Chef Service DPER
13	DZEUFIEU DJOMENI Paul Désiré	Maître de Conférences	En poste
14	FOTO MENBOHAN Samuel	Maître de Conférences	En poste
15	KAMGANG René	Maître de Conférences	<i>C.S. MINRESI</i>
16	KEKEUNOU Sévior	Maître de Conférences	En poste
17	MEGNEKOU Rosette	Maître de Conférences	En poste
18	MONY Ruth épouse NTONE	Maître de Conférences	En Poste
19	TOMBI Jeannette	Maître de Conférences	En poste
20	ZEBAZE TOGOUET Serge Hubert ZEBAZE TOGOUET Serge Hubert	Maître de Conférences	En poste
21	ALENE Désirée Chantal	Chargée de Cours	En poste
22	ATSAMO Albert Donatien	Chargée de Cours	En poste
23	BELLET EDIMO Oscar Roger	Chargé de Cours	En poste
24	BILANDA Danielle Claude	Chargée de Cours	En poste
25	DJIOGUE Séfirin	Chargée de Cours	En poste
26	DONFACK Mireille	Chargée de Cours	En poste
27	GOUNOUE KAMKUMO Raceline	Chargée de Cours	En poste
28	LEKEUFACK FOLEFACK Guy B.	Chargé de Cours	En poste
29	MAHOB Raymond Joseph	Chargé de Cours	En poste
30	MBENOUN MASSE Paul Serge	Chargé de Cours	En poste
31	MOUNGANG LucianeMarlyse	Chargée de Cours	En poste

32	MVEYO NDANKEU Yves Patrick	Chargée de Cours	En poste
33	NGOUATEU KENFACK Omer Bébé	Chargé de Cours	En poste
34	NGUEGUIM TSOFAK Florence	Chargée de Cours	En poste
35	NGUEMBOK	Chargé de Cours	En poste
36	NJATSA Hermine épouse MEGAPTCHE	Chargée de Cours	En Poste
37	NJUA Clarisse Yafi	Chargée de Cours	CD/UBa
38	NOAH EWOTI Olive Vivien	Chargée de Cours	En poste
39	TADU Zephyrin	Chargée de Cours	En poste
40	YEDE	Chargée de Cours	En poste
41	ETEME ENAMA Serge	Assistant	En poste
42	KANDEDA KAVAYE Antoine	Assistant	En poste
43	KOGA MANG DOBARA	Assistant	En poste

3- DÉPARTEMENT DE BIOLOGIE ET PHYSIOLOGIE VÉGÉTALES (BPV) (26)

1	AMBANG Zachée	Professeur	Chef Division/UYII
2	BELL Joseph Martin	Professeur	En poste
3	YOUMBI Emmanuel	Professeur	Chef de Département
4	MOSSEBO Dominique Claude	Professeur	En poste
5	BIYE Elvire Hortense	Maître de Conférences	En poste
6	DJOCGOUE Pierre François	Maître de Conférences	En poste
7	KENGNE NOUMSI Ives Magloire	Maître de Conférences	En poste
8	MALA Armand William	Maître de Conférences	En poste
9	NDONGO BEKOLO	Maître de Conférences	<i>CE / MINRESI</i>
10	NGONKEU MAGAPTCHE Eddy L.	Maître de Conférences	En poste
11	ZAPFACK Louis	Maître de Conférences	En poste
12	MBARGA BINDZI Marie Alain	Maître de Conférences	CT/Univ Dschang
13	MBOLO Marie	Maître de Conférences	En poste
14	ANGONI Hyacinthe	Chargée de Cours	En poste
15	MAHBOU SOMO TOUKAM. Gabriel	Chargé de Cours	En poste
16	ONANA JEAN MICHEL	Chargé de Cours	En poste
17	GOMANDJE Christelle	Chargée de Cours	En poste
18	NGODO MELINGUI Jean Baptiste	Chargé de Cours	En poste
19	NGALLE Hermine BILLE	Chargée de Cours	En poste
20	NGOUO Lucas Vincent	Chargé de Cours	En poste
21	NSOM ZAMO Annie Claude épouse PIAL	Chargée de Cours	<i>Expert national /UNESCO</i>
22	TONFACK Libert Brice	Chargé de Cours	En poste
23	TSOATA Esaïe	Chargé de Cours	En poste
24	DJEUANI Astride Carole	Assistante	En poste
25	MAFFO MAFFO Nicole Liliane	Assistante	En poste
26	NNANGA MEBENGA Ruth Laure	Assistante	En poste
27	NOUKEU KOUAKAM Armelle	Assistante	En poste

4- DÉPARTEMENT DE CHIMIE INORGANIQUE (CI) (33)

1	AGWARA ONDOH Moïse	Professeur	<i>Vice Recteur Univ ,Bamenda</i>
2	ELIMBI Antoine	Professeur	En poste
3	Florence UFI CHINJE épouse MELO	Professeur	<i>RECTEUR Univ.Ngaoundere</i>
4	GHOOGOMU Paul MINGO	Professeur	<i>Directeur Cabinet PM</i>
5	LAMINSI Samuel	Professeur	En poste
6	NANSEU Njiki Charles Péguy	Professeur	En poste
7	NDIFON Peter TEKE	Professeur	<i>IS1 MINRESI/Chef de Département</i>

8	NENWA Justin	Professeur	En poste
9	NGAMENI Emmanuel	Professeur	<i>DOYEN FS Univ. Dschang</i>
10	BABALE née DJAM DOUDOU	Maître de Conférences	<i>Chargée Mission P.R.</i>
11	DJOUFAC WOUMFO Emmanuel	Maître de Conférences	En poste
12	KEMMEGNE MBOUGUEM Jean C.	Maître de Conférences	En poste
13	KONG SAKEO	Maître de Conférences	<i>Chargé de Mission au P. M.</i>
14	NDIKONTAR Maurice KOR	Maître de Conférences	<i>Vice-Doyen Univ. Bamenda</i>
15	NGOMO Horace MANGA	Maître de Conférences	VC/UB
16	NJIOMOU C. épse DJANGANG	Maître de Conférences	En poste
17	YOUNANG Elie	Maître de Conférences	En poste
18	ACAYANKA Elie	Chargé de Cours	En poste
19	EMADACK Alphonse	Chargé de Cours	En poste
20	KAMGANG YOUNBI Georges	Chargé de Cours	En poste
21	NDI NSAMI Julius	Chargée de Cours	En poste
22	NJOYA Dayirou	Chargé de Cours	En poste
23	PABOUDAM GBAMBIE A.	Chargée de Cours	En poste
24	TCHAKOUTE KOUAMO Hervé	Chargé de Cours	En poste
25	BELIBI BELIBI Placide Désiré	Chargé de Cours	En poste
26	CHEUMANI YONA Arnaud M.	Chargé de Cours	En poste
27	NYAMEN Linda Dyorisse	Chargée de Cours	En poste
28	KENNE DEDZO GUSTAVE	Chargé de Cours	En poste
29	KOUOTOU DAOUDA	Chargé de Cours	En poste
30	MAKON Thomas Beauregard	Chargé de Cours	En poste
31	MBEY Jean Aime	Chargé de Cours	En poste
32	NCHIMI NONO KATIA	Chargé de Cours	En poste
33	NEBA nee NDOSIRI Bridget NDOYE	Chargé de Cours	En poste

5- DÉPARTEMENT DE CHIMIE ORGANIQUE (CO) (34)

1	DONGO Etienne	Professeur	Vice-Doyen / DPSAA
2	GHOHOMU TIH Robert Ralph	Professeur	Dir IBAF/UDS
3	MBAFOR Joseph	Professeur	En poste
5	NGOUELA Silvere Augustin	Professeur	En poste
6	NKENGFACK Augustin Ephraïm	Professeur	Chef de Département
7	NYASSE Barthélemy	Professeur	<i>Directeur/UN</i>
8	PEGNYEMB Dieudonné Emmanuel	Professeur	<i>Directeur/ MINESUP</i>
9	WANDJI Jean	Professeur	En poste
10	Alex de Théodore ATCHADE	Maître de Conférences	<i>DEPE/ Rectorat/Uyi</i>
11	FOLEFOC Gabriel NGOSONG	Maître de Conférences	<i>En poste</i>
12	KEUMEDJIO Félix	Maître de Conférences	En poste
13	KOUAM Jacques	Maître de Conférences	En poste
14	MBAZOA née DJAMA Céline	Maître de Conférences	En poste
15	NOUNGOUE TCHAMO Diderot	Maître de Conférences	En poste
16	TCHOUANKEU Jean-Claude	Maître de Conférences	<i>VR/ UYII</i>
17	YANKEP Emmanuel	Maître de Conférences	En poste
18	TIH née NGO BILONG E. Anastasie	Maître de Conférences	En poste
19	MKOUNGA Pierre	Maître de Conférences	En poste
20	NGO MBING Joséphine	Maître de Conférences	En poste
21	TABOPDA KUATE Turibio	Maître de Conférences	En poste
22	KEUMOGNE Marguerite	Maître de Conférences	En poste
23	AMBASSA Pantaléon	Chargé de Cours	En poste
24	EYONG Kenneth OBEN	Chargé de Cours	En poste
25	FOTSO WABO Ghislain	Chargé de Cours	En poste

26	KAMTO Eutrophe Le Doux	Chargé de Cours	En poste
27	NGONO BIKOBO Dominique Serge	Chargé de Cours	En poste
28	NOTE LOUGBOT Olivier Placide	Chargé de Cours	Chef Service/Minesup
29	OUAHOUE WACHE Blandine M.	Chargée de Cours	En poste
30	TAGATSING FOTSING Maurice	Chargé de Cours	En poste
31	ZONDENDEGOUNBA Ernestine	Chargée de Cours	En poste
32	NGOMO Orléans	Chargée de Cours	En poste
33	NGNINTEDO Dominique	Assistant	En poste
6- DÉPARTEMENT D'INFORMATIQUE (IN) (25)			
1	ATSA ETOUNDI Roger	Professeur	<i>Chef DivSys.</i>
2	FOUDA NDJODO Marcel Laurent	Professeur	<i>Chef Dpt ENS/Chef DivSys.MINESUP</i>
3	NDOUNDAM René	Maître de Conférences	En poste
4	KOUOKAM KOUOKAM E. A.	Chargé de Cours	En poste
5	CHEDOM FOTSO Donatien	Chargé de Cours	En poste
6	MELATAGIA YONTA Paulin	Chargé de Cours	En poste
7	MOTO MPONG Serge Alain	Chargé de Cours	En poste
8	TINDO Gilbert	Chargé de Cours	En poste
9	TSOPZE Norbert	Chargé de Cours	En poste
10	WAKU KOUAMOU Jules	Chargé de Cours	En poste
11	TAPAMO Hyppolite	Chargé de Cours	En poste
12	ABESSOLO ALO'O Gislain	Assistant	En poste
13	BAYEM Jacques Narcisse	Assistant	En poste
14	DJOUWE MEFFEJA Merline Flore	Assistante	En poste
15	DOMGA KOMGUEM Rodrigue	Assistant	En poste
16	EBELE Serge	Assistant	En poste
17	HAMZA Adamou	Assistant	En poste
18	KAMDEM KENGNE Christiane	Assistante	En poste
19	KAMGUEU Patrick Olivier	Assistant	En poste
20	KENFACK DONGMO Clauvice V.	Assistant	En poste
21	MEYEMDOU Nadège Sylvianne	Assistante	En poste
22	MONTHÉ DJIADEU Valéry M.	Assistant	En poste
23	JIOMEKONG AZANZI Fidel	Assistant	En poste
7- DÉPARTEMENT DE MATHÉMATIQUES (MA) (35)			
1	BEKOLLE David	Professeur	<i>Vice-Recteur UN</i>
2	BITJONG NDOMBOL	Professeur	<i>En poste</i>
3	DOSSA COSSY Marcel	Professeur	En poste
4	AYISSI Raoult Domingo	Maître de Conférences	Chef de Département
5	EMVUDU WONO Yves S.	Maître de Conférences	<i>CD/ MINESUP /Chef de Département (IN)</i>
6	NKUIMI JUGNIA Célestin	Maître de Conférences	En poste
7	NOUNDJEU Pierre	Maître de Conférences	En poste
8	TCHAPNDA NJABO Sophonie B.	Maître de Conférences	Directeur/AIMS Rwanda
9	AGHOUEKENG JIOFACK Jean Gérard	Chargé de Cours	Chef Cellule MINPLAMAT
10	CHENDJOU Gilbert	Chargé de Cours	En poste
11	FOMEKONG Christophe	Chargé de Cours	En poste
12	KIANPI Maurice	Chargé de Cours	En poste
13	KIKI Maxime Armand	Chargé de Cours	En poste
14	MBAKOP Guy Merlin	Chargé de Cours	En poste
15	MBANG Joseph	Chargé de Cours	En poste
16	MBEHOU Mohamed	Chargé de Cours	En poste

17	MBELE BIDIMA Martin Ledoux	Chargé de Cours	En poste
18	MENGUE MENGUE David Joe	Chargé de Cours	En poste
19	NGUEFACK Bernard	Chargé de Cours	En poste
20	POLA DOUNDOU Emmanuel	Chargé de Cours	En poste
21	TAKAM SOH Patrice	Chargé de Cours	En poste
22	TCHANGANG Roger Duclos	Chargé de Cours	En poste
23	TCHOUNDJA Edgar Landry	Chargé de Cours	En poste
24	TETSADJIO TCHILEPECK M. E.	Chargée de Cours	En poste
25	TIAYA TSAGUE N. Anne-Marie	Chargée de Cours	En poste
26	DJIADEU NGAHA Michel	Assistant	En poste
27	MBIAKOP Hilaire George	Assistant	En poste
28	NIMPA PEFOUNKEU Romain	Assistant	En poste
29	TANG AHANDA Barnabé	Assistant	Directeur/MINTP
8- DÉPARTEMENT DE MICROBIOLOGIE (MIB) (13)			
1	ESSIA NGANG Jean Justin	Professeur	DRV/IMPM
2	ETOA François Xavier	Professeur	Chef de Département Recteur Université de Douala
3	NWAGA Dieudonné M.	Maître de Conférences	En poste
4	NYEGUE Maximilienne Ascension	Maître de Conférences	En poste
5	SADO KAMDEM Sylvain Leroy	Maître de Conférences	En poste
6	BOYOMO ONANA	Maître de Conférences	En poste
7	RIWOM Sara Honorine	Maître de Conférences	En poste
8	BODA Maurice	Chargé de Cours	En poste
9	BOUGNOM Blaise Pascal	Chargé de Cours	En poste
10	ESSONO OBOUGOU Germain G.	Chargé de Cours	En poste
11	NJIKI BIKOÏ Jacky	Chargée de Cours	En poste
12	TCHIKOUA Roger	Chargé de Cours	En poste
9.DEPARTEMENT DE PYSIQUE(PHY)			
1	ESSIMBI ZOBO Bernard	Professeur	En poste
2	KOFANE Timoléon Crépin	Professeur	En poste
3	NDJAKA Jean Marie Bienvenu	Professeur	Chef de Département
4	NJOMO Donatien	Professeur	En poste
5	PEMHA Elkana	Professeur	En poste
6	TABOD Charles TABOD	Professeur	Doyen Univ/Bda
7	TCHAWOUA Clément	Professeur	En poste
8	WOAFO Paul	Professeur	En poste
9	EKOBENA FOUDA Henri Paul	Maître de Conférences	<i>Chef Division. UN</i>
10	NJANDJOCK NOUCK Philippe	Maître de Conférences	<i>Sous Directeur/ MINRESI</i>
11	BIYA MOTTO Frédéric	Maître de Conférences	DG/HYDRO Mekin
12	BEN- BOLIE Germain Hubert	Maître de Conférences	CD/ENS/UN
13	DJUIDJE KENMOE épouse ALOYEM	Maître de Conférences	En poste
14	NANA NBENDJO Blaise	Maître de Conférences	En poste
15	NOUAYOU Robert	Maître de Conférences	En poste
16	SIEWE SIEWE Martin	Maître de Conférences	En poste
17	ZEKENG Serge Sylvain	Maître de Conférences	En poste
18	EYEBE FOUDA Jean sire	Maître de Conférences	En poste
19	FEWO Serge Ibraïd	Maître de Conférences	En poste
20	HONA Jacques	Maître de Conférences	En poste
21	OUMAROU BOUBA	Maître de Conférences	<i>En poste</i>
22	SAIDOU	Maître de Conférences	Sous Directeur/Minresi

23	SIMO Elie	Maître de Conférences	En poste
24	BODO Bernard	Chargé de Cours	En poste
25	EDONGUE HERVAIS	Chargé de Cours	En poste
26	FOUEDJIO David	Chargé de Cours	En poste
27	MBANE BIOUELE	Chargé de Cours	En poste
28	MBINACK Clément	Chargé de Cours	En poste
29	MBONO SAMBA Yves Christian U.	Chargé de Cours	En poste
30	NDOP Joseph	Chargé de Cours	En poste
31	OBOUNOU Marcel	Chargé de Cours	DA/Univ Inter Etat/Sangmalima
32	TABI Conrad Bertrand	Chargé de Cours	En poste
33	TCHOFFO Fidèle	Chargé de Cours	En poste
34	VONDOU DerbetiniAppolinaire	Chargé de Cours	En poste
35	WOULACHE Rosalie Laure	Chargée de Cours	En poste
36	ABDOURAHIMI	Chargé de Cours	En poste
37	ENYEGUE A NYAM épouse BELINGA	Chargée de Cours	En poste
38	WAKATA née BEYA Annie	Chargée de Cours	<i>Sous Directeur/ MINESUP</i>
39	MVOGO ALAIN	Chargé de Cours	<i>En poste</i>
40	CHAMANI Roméo	Assistant	En poste
41	MLI JOELLE LARISSA	Assistante	<i>En poste</i>
10- DÉPARTEMENT DE SCIENCES DE LA TERRE (ST) (42)			
1	NDJIGUI Paul Désiré	Professeur	Chef de Département
2	BITOM Dieudonné	Professeur	<i>Doyen / FASA / UDs</i>
3	NZENTI Jean-Paul	Professeur	En poste
4	KAMGANG Pierre	Professeur	En poste
5	MEDJO EKO Robert	Professeur	<i>Coseiller Technique/UYII</i>
6	FOUATEU Rose épouse YONGUE	Maître de Conférences	En poste
7	NDAM NGOUPAYOU Jules-Remy	Maître de Conférences	En poste
8	NGOS III Simon	Maître de Conférences	DAAC/Uma
9	NJILAH Isaac KONFOR	Maître de Conférences	En poste
10	NKOUMBOU Charles	Maître de Conférences	En poste
11	TEMDJIM Robert	Maître de Conférences	En poste
12	YENE ATANGANA Joseph Q.	Maître de Conférences	<i>Chef Div. /MINTP</i>
13	ABOSSOLO née ANGUE Monique	Maître de Conférences	<i>Chef div. DAASR / FS</i>
14	GHOGOMU Richard TANWI	Maître de Conférences	CD/UMa
15	MOUNDI Amidou	Maître de Conférences	<i>Chef Div. MINIMDT</i>
16	ONANA Vincent	Maître de Conférences	En poste
17	TCHOUANKOUE Jean-Pierre	Maître de Conférences	En poste
18	ZO'O ZAME Philémon	Maître de Conférences	<i>DG/ART</i>
19	MOUNDI Amidou	Maître de Conférences	<i>CT/ MINIMDT</i>
20	BEKOA Etienne	Chargé de Cours	<i>En poste</i>
21	BISSO Dieudonne	Chargé de Cours	<i>Directeur/Projet Barrage Memve'ele</i>
22	ESSONO Jean	Chargé de Cours	<i>En poste</i>
23	EKOMANE Emile	Chargé de Cours	<i>En pste</i>
24	FUH Calistus Gentry	Chargée de cours	<i>Sec. D'Etat/MINIMDT</i>
25	GANNO Sylvestre	Chargé de Cours	En poste
26	LAMILEN BILLA Daniel	Chargé de Cours	En poste
27	MBIDA YEM	Chargé de Cours	<i>En poste</i>
28	MINYEM Dieudonné-Lucien	Chargé de Cours	<i>CD/Uma</i>
29	MOUAFO Lucas	Chargé de Cours	En poste
31	NGO BELNOUN Rose Noël	Chargée de Cours	En poste

32	NGO BIDJECK Louise Marie	Chargée de Cours	En poste
33	NGUETCHOUA Gabriel	Chargé de Cours	CEA/MINRESI
34	NYECK Bruno	Chargé de Cours	En poste
35	TCHAKOUNTE J. épouse NOUMBEM	Chargée de Cours	CT / MINRESI
36	METANG Victor	Chargé de cours	En poste
37	NOMO NEGUE Emmanuel	Chargé de cours	En poste
38	TCHAPTCHET TCHATO De P.	Chargé de cours	En poste
39	TEHNA Nathanaël	Chargé de cours	En poste
40	TEMGA Jean Pierre	Chargé de cours	En poste
41	MBESSE CECILE OLIVE	Chargée de cours	En poste
42	ELISE SABABA	Chargé de cours	En poste
43	EYONG JOHN TAKEM	Assistant	En poste
44	ANABA ONANA Achille Basile	Assistant	En poste

Répartition chiffrée des Enseignants de la Faculté des Sciences de l'Université de Yaoundé I

NOMBRE D'ENSEIGNANTS					
DÉPARTEMENT	Professeurs	Maîtres de Conférences	Chargés de Cours	Assistants	Total
B.C.	5 (1)	10 (5)	20 (10)	3 (1)	39 (16)
B.P.A.	11 (1)	9 (3)	20 (8)	3 (5)	43 (17)
B.P.V.	4 (0)	9(2)	10 (2)	4 (4)	27 (8)
C.I.	9(1)	8(2)	16 (4)	0 (2)	33 (9)
C.O.	8 (0)	13 (3)	8 (2)	1 (0)	30 (5)
I.N.	3 (0)	1 (0)	8 (0)	12 (3)	24 (3)
M.A.	3 (0)	5 (0)	18 (1)	4 (0)	30 (1)
M.B.	2 (0)	5 (2)	6 (2)	0 (0)	13 (4)
P.H.	8 (0)	17 (0)	15 (2)	2 (1)	42 (3)
S.T.	5 (0)	15 (2)	22 (3)	2 (0)	44 (5)
Total	58 (3)	92(19)	144 (33)	31(16)	325(71)

Soit un total de **325(71)** dont :

- Professeurs **58 (3)**
- Maîtres de Conférences **92 (19)**
- Chargés de Cours **144 (33)**
- Assistants **31 (16)**

() = Nombre de Femmes

Remerciements

Je tiens tout d'abord à remercier mes encadrateurs, Professeur Maurice TCHUENTE et Professeur Emmanuel SIMEU, sans qui je n'aurais jamais vécu cette aventure passionnante qu'est la thèse de doctorat. Je remercie Professeur TCHUENTE et Professeur SIMEU d'avoir accepté de diriger mes travaux de thèse, de m'avoir guidée, conseillée, soutenue et accompagnée durant toutes ces années de thèse. Je remercie Professeur Maurice TCHUENTE et Professeur Emmanuel SIMEU de m'avoir permis d'effectuer cette thèse en cotutelle entre l'Université de Yaoundé I et la Communauté Université Grenoble Alpes. Je les remercie pour l'aide précieuse qu'ils m'ont apportée, pour leur disponibilité, pour leur engagement et pour la patience dont ils ont fait preuve envers moi.

Je remercie Professeur Médard FOGUE d'avoir accepté de présider à mon Jury de soutenance. Je remercie Professeur Emmanuel TANYI, Professeur ROOF KOSAI, Professeur Thomas DJOTIO, Professeur René NDOUNAM et Professeur Serge STINKWICH d'avoir accepté d'examiner mon travail et de participer à mon Jury de soutenance.

Je remercie l'Université de Yaoundé I pour la formation et l'encadrement dont j'ai bénéficié. Je remercie notamment le Département d'Informatique et ses enseignants qui m'ont donné toutes les connaissances académiques et en particulier une formation en microcontrôleurs, nécessaires pour m'engager dans cette thèse de doctorat. Je remercie le Centre d'Excellence Africain en Technologies de l'Information et de la Communication (CETIC) pour le support financier apporté à ma thèse. Je remercie l'Institut de Recherche pour le Développement (IRD), l'Unité de Modélisation Mathématique et Informatique des Systèmes Complexes (UMMISCO) et le Laboratoire International de Recherche en Informatique et Mathématiques Appliquées (LIRIMA) pour leur soutien financier et académique. Je remercie Docteur Guillaume CHELIUS qui m'a initié et formé dans le domaine des réseaux de capteurs et dont l'encadrement a permis l'obtention d'un Master dans cette spécialité. Je tiens aussi à remercier tous mes collègues et camarades de l'Université de Yaoundé I qui m'ont aidée et encouragée tout au long de cette thèse.

Au cours de ces années de thèse, j'ai effectué plusieurs séjours à l'Université Grenoble Alpes, au sein du laboratoire des Techniques de l'Informatique et de la Microélectronique pour l'Architecture des systèmes intégrés (TIMA). Je profite de cette occasion pour remercier le laboratoire TIMA et en particulier l'équipe RMS (Reliable Mixed signal Systems) pour le soutien financier apporté à la réalisation de cette thèse. Je remercie aussi les membres de l'équipe RMS et l'administration du laboratoire TIMA pour l'accueil chaleureux qu'ils m'ont réservé lors de chacun de mes séjours parmi eux.

Je remercie l'Agence Universitaire de la Francophonie (AUF) qui a financé ma thèse dans le cadre du programme *Horizons Francophones Sciences Fondamentales* durant les années 2012 à 2015. En plus de financer mes séjours dans les deux universités de ma cotutelle, l'AUF m'a accompagnée en

organisant des séminaires de formation et d'encadrement. Je remercie particulièrement les membres du Comité de Pilotage de ce programme et Mme Sylvie MOUANGUE pour leur dévouement.

Cette thèse s'est déroulée sous le regard bienveillant du mécène, Docteur Paul K. FOKAM, promoteur de l'Université PKFokam Institute of Excellence, qui a joué un rôle de catalyseur par ses encouragements et son soutien financier. Je remercie sincèrement Docteur Paul K. FOKAM d'avoir cru en moi et de m'avoir donné un cadre de travail épanouissant au sein de l'Université PKFokam Institute of Excellence. Je remercie le Vice-Président Professeur Jean WOUAFO KAMGA, le Recteur Professeur Thomas NJINE et toute l'équipe dirigeante de l'Université PKFokam Institute of Excellence pour m'avoir toujours soutenue et avoir toujours donné la priorité à mes travaux de recherche plutôt qu'à mes heures d'enseignement en me permettant d'effectuer de longs séjours de recherche hors du Cameroun sans suspension de mon traitement salarial. Je remercie tous mes collègues de l'Université PKFokam Institute of Excellence pour leur soutien et leur encouragement.

Durant toutes ces années de thèse, j'ai été entourée de personnes merveilleuses qui ont cru en moi et m'ont épaulée. Je dis un grand merci à tous mes proches, mes parents, mes frères et sœurs et mes amis en particulier Dorcas, Léon, Valery, Raymond, Mariama, Charlène et Rodrigue. Je remercie tous ceux et celles, qui de près ou de loin, m'ont permis d'arriver au bout de cette thèse. Je remercie ma fille chérie, Marie-ange, qui a été très courageuse et m'a ainsi permis de donner le meilleur de moi dans ce travail de thèse.

A papa Joseph et maman Jacqueline TCHAKONTE

Résumé

Les réseaux de capteurs sans fil (WSN) se composent de petits nœuds de capteurs dotés de microcontrôleurs intégrés, de radios à faible consommation, de batteries et de capteurs utilisés pour surveiller les conditions environnementales telles que la température, la pression, l'humidité et les vibrations. De nos jours, ces réseaux sont utilisés dans un large éventail d'applications militaires, sanitaires, domestiques, urbaines, industrielles et environnementales. Les applications de couverture de cibles sont celles où plusieurs points d'intérêt appelés cibles doivent être surveillés en permanence par des nœuds de capteurs. Dans la plupart des applications de couverture de cibles, les nœuds de capteurs ont une ressource d'énergie limitée et il est donc essentiel de gérer efficacement leur consommation d'énergie afin de prolonger la durée de vie du réseau. Une approche commune pour résoudre ce problème consiste à alterner le fonctionnement des nœuds de capteurs entre le mode actif et le mode veille. Le problème de former des sous-ensembles de nœuds de capteurs qui seront activés successivement pendant des durées définies tandis que les autres nœuds seront en veille, afin de maximiser la durée de vie du réseau, est NP-difficile et appelé MLCP (Maximum Lifetime Coverage Problem). Dans cette thèse, notre objectif est de proposer de nouvelles heuristiques pour le MLCP tout en considérant des hypothèses plus réalistes sur la durée de vie et la consommation énergétique des nœuds de capteurs. Tout d'abord, nous proposons deux heuristiques gloutonnes en supposant que les nœuds de capteurs n'ont pas nécessairement la même durée de vie. La première heuristique est basée sur une méthode adaptative tandis que la seconde utilise l'idée de liste noire, ce qui permet d'optimiser la gestion des cibles critiques, c'est-à-dire qui sont couvertes par un nombre minimal de capteurs. Ensuite, en considérant que l'énergie consommée par les nœuds de capteurs mis en veille n'est pas négligeable, nous proposons une troisième heuristique gloutonne qui prend en compte l'énergie résiduelle des nœuds de capteurs dans le choix des nœuds à activer. Puis, nous proposons une approche récurrente pour les réseaux réguliers. Nous étudions ensuite une famille d'instances difficiles du MLCP, à savoir la sous-famille des réseaux réguliers constitués d'anneaux de taille impaire. Nous proposons pour cette sous-classe, une approche analytique permettant d'obtenir des solutions efficaces dont nous conjecturons l'optimalité. Enfin, nous développons un système de surveillance de la pollution atmosphérique et de détection d'incendie basé sur un réseau de capteurs sans fil et nous évaluons le gain en termes de durée de vie du réseau lorsqu'un algorithme pour le MLCP est intégré dans un tel système.

Mots clés

Réseaux de capteurs sans fil, surveillance de cibles, consommation d'énergie, durée de vie, ensembles couvrants, heuristiques gloutonnes, cibles critiques, liste noire, réseaux réguliers.

Abstract

Wireless Sensor Networks (WSN) consist of tiny sensor nodes with embedded microcontrollers, low power radios, battery cells and sensors which are used to monitor environmental conditions such as temperature, pressure, humidity, and vibration. Today, these networks are used in a wide range of military, health, domestic, urban, industrial and environmental applications. Target coverage applications are those where several points of interest called targets must be continuously monitored by sensor nodes. In most target coverage applications, sensor nodes have a limited amount of energy and it is therefore critical to efficiently manage their energy consumption in order to extend the network lifetime. A common approach to tackle this problem is to alternate the operation of sensor nodes between active and sleep mode. The scheduling of appropriate subsets of sleep/active sensor nodes in order to maximize the network lifetime is an NP-hard problem called Maximum Lifetime Coverage Problem (MLCP). In this thesis, we aim at proposing new heuristics to the MLCP, while considering more realistic assumptions on lifetime and energy consumption of sensor nodes. Firstly, we propose two greedy heuristics with the assumption that the sensor nodes do not necessarily have the same lifetime. The first heuristic is based on an adaptive method while the second uses the idea of blacklist, which allows to optimize the management of least covered targets called critical targets. Secondly, by considering that the energy consumed by sensor nodes put in sleep mode is not negligible, we propose a third greedy heuristic that takes into account the remaining energy of the sensor nodes in the choice of the nodes to activate. Then, we propose a recurring approach for regular networks. We then study a family of hard instances of the MLCP, namely the sub-family of regular networks composed of odd-sized rings. We propose for this subclass, an analytical approach to obtain effective solutions which we conjecture optimality. Finally, we develop a system for air pollution monitoring and fire detection based on a wireless sensor network and we evaluate the network lifetime gain when an algorithm for the MLCP is integrated in such a system.

Keywords

Wireless sensor networks, target monitoring, energy consumption, lifetime, cover sets, greedy heuristics, critical targets, black list, regular networks.

Tables des matières

Remerciements.....	x
Résumé.....	xii
Abstract.....	xiii
Chapitre 1 Introduction	1
1.1 Contexte.....	1
1.2 Défis liés à la mise en oeuvre des réseaux de capteurs sans fil.....	2
1.3 Objectifs de la thèse	4
1.4 Organisation du mémoire	5
Chapitre 2 Réseaux de capteurs sans fil.....	7
2.1 Modules d'un nœud de capteur sans fil.....	7
2.1.1 Le module de traitement.....	7
2.1.2 Le module de communication	8
2.1.3 Le module de captage	13
2.1.4 La source d'énergie	14
2.2 Plateformes de nœud de capteur	14
2.3 Applications des réseaux de capteurs.....	15
Chapitre 3 Etat de l'art du problème de couverture de cibles par les réseaux de capteurs sans fil	19
3.1 Introduction	19
3.1.1 La couverture de surface.....	19
3.1.2 La couverture de cibles	20
3.1.3 La couverture de barrière.....	20
3.2 Problème de couverture (MLCP)	21
3.3 Heuristiques de l'approche par couvrants disjoints.....	23
3.3.1 Heuristique basée sur les couvrants minimaux.....	23
3.3.2 L'algorithme MCMCC (Most Constrained - Minimally Constraining Coverage).....	24
3.3.3 L'algorithme MC-MIP (Maximum Covers using Mixed Integer Programming).....	25
3.3.4 Heuristique basée sur la méthode Simplexe.....	26
3.3.5 L'algorithme GAMDSC (Genetic Algorithm for Maximum Disjoint Set Covers)	27
3.3.6 L'algorithme Flowing-GA (Flowing Genetic Algorithm)	28

3.4	Heuristiques de l'approche par couvrants non disjoints	29
3.4.1	L'algorithme Greedy-MSC (Greedy Maximum Set Cover heuristic).....	29
3.4.2	L'algorithme Static-CCF (Static Critical Control Factor).....	30
3.4.3	L'algorithme Dynamic-CCF (Dynamic Critical Control Factor)	31
3.5	Conclusion.....	31
Chapitre 4	Nouvelles heuristiques pour le MLCP	33
4.1	Introduction	33
4.2	Heuristiques proposées	39
4.2.1	Approche réactive.....	39
4.2.2	Algorithme adaptatif.....	40
4.2.3	Algorithme blacklist	45
4.3	Simulations et analyse des résultats	49
4.4	Conclusion.....	53
Chapitre 5	Prise en compte de la consommation en veille dans le MLCP	55
5.1	Introduction	55
5.2	Borne supérieure de la durée de vie du réseau	58
5.3	Algorithme pour MLCP avec prise en compte de l'énergie en veille	59
5.4	Simulations et analyse des résultats	62
5.5	Conclusion.....	70
Chapitre 6	Le MLCP dans le cas des réseaux réguliers.....	73
6.1	Introduction	73
6.2	Famille des réseaux en anneau impair : exemple d'instances difficiles.....	74
6.3	Heuristique pour la famille des réseaux en anneau impair	77
6.3.1	Méthode analytique de résolution pour les capteurs de même énergie.....	78
6.3.2	Heuristique OddRingNetworks	80
6.4	Applications des réseaux en anneau.....	82
6.5	Conclusion.....	83
Chapitre 7	Réseaux de capteurs pour la surveillance de la pollution atmosphérique	85
7.1	Introduction	85
7.2	Principaux polluants et leurs conséquences.....	86
7.3	Etat de l'art des systèmes de surveillance de la pollution de l'air basés sur les réseaux de capteurs	88
7.4	Implémentation d'un système de surveillance de la qualité de l'air et de détection d'incendies.....	91
7.4.1	Architecture du système	92
7.4.2	Architecture d'un nœud de capteur.....	92
7.4.3	Architecture de la station de base	94
7.4.4	Protocole de communication	95
7.4.5	Base de données et application Web	97

7.5	Contrainte d'énergie dans les systèmes de surveillance de la qualité de l'air	100
7.6	Conclusion.....	103
Chapitre 8	Conclusion générale.....	105
8.1	Rappel des enjeux.....	105
8.2	Contributions.....	105
8.3	Perspectives	107
Bibliographie		108

Table des figures

Figure 1- Les principaux modules d'un noeud de capteur	2
Figure 2 - Exemple de contrôle de topologie.....	4
Figure 3 - Topologie d'un réseau utilisant une technologie radio courte portée	10
Figure 4 - Topologie d'un réseau utilisant la technologie radio SIGFOX	11
Figure 5 - Topologie d'un réseau utilisant la technologie radio LoRa.....	13
Figure 6 - Exemples de capteurs omnidirectionnels	13
Figure 7 - Exemples de nœuds de capteurs.....	15
Figure 8 - Solution d'une instance du MLCP	22
Figure 9 - Une instance difficile de MLCP	23
Figure 10 - Une instance de MLCP	24
Figure 11 - Représentation d'une solution de MLCP sous forme de chromosome	27
Figure 12 - Variation de la capacité en fonction de la température et du courant de décharge.....	34
Figure 13 - Courbes de décharge d'une pile pour différentes puissances et températures	35
Figure 14 - L'approche classique et l'approche réactive	40
Figure 15 - Algorithme adaptatif.....	43
Figure 16 - Une instance de MLCP	44
Figure 17 - Exécution 1 de l'approche réactive avec l'algorithme adaptatif.....	44
Figure 18 - Exécution 2 de l'approche réactive avec l'algorithme adaptatif.....	45
Figure 19 - Algorithme Blacklist.....	48
Figure 20 - Exécution de l'approche réactive avec l'algorithme blacklist	49
Figure 21 - Boîte à moustache.....	50
Figure 22 - Performance des algorithmes pour 150 capteurs et écart-type = 10.....	52
Figure 23 - Performance des algorithmes pour 150 capteurs et écart-type = 20.....	53
Figure 24 - Performance des algorithmes pour 300 capteurs et écart-type = 10.....	53
Figure 25 - Performance des algorithmes pour 300 capteurs et écart-type = 20.....	53
Figure 26 - Approche réactive	60
Figure 27 - Algorithme Energy-Blacklist	62
Figure 28 - Performance des algorithmes pour 150 capteurs, écart-type = 500 et $r=0$	65
Figure 29 - Performance des algorithmes pour 150 capteurs, écart-type = 500 et $r=0.1$	65
Figure 30 - Performance des algorithmes pour 300 capteurs, écart-type = 500 et $r=0$	65
Figure 31 - Performance des algorithmes pour 300 capteurs, écart-type = 500 et $r=0.1$	65
Figure 32 - Performance des algorithmes pour 150 capteurs, écart-type = 1000 et $r=0$	66
Figure 33 - Performance des algorithmes pour 150 capteurs, écart-type = 1000 et $r=0.1$	66
Figure 34 - Performance des algorithmes pour 300 capteurs, écart-type = 1000 et $r=0$	66
Figure 35 - Performance des algorithmes pour 300 capteurs, écart-type = 1000 et $r=0.1$	66
Figure 36 - Performance de Static-CCF Algorithm pour 300 capteurs et écart-type = 500	69

Figure 37 - Performance de Dynamic-CCF Algorithm pour 300 capteurs et écart-type = 500	69
Figure 38 - Performance de MCS- based Algorithm pour 300 capteurs et écart-type = 500	69
Figure 39 - Performance de Greedy-MSC Algorithm pour 300 capteurs et écart-type = 500	69
Figure 40 - Performance de Adaptive Algorithm pour 300 capteurs et écart-type = 500	69
Figure 41 - Performance de Blacklist Algorithm pour 300 capteurs et écart-type = 500	69
Figure 42 - Performance de Energy-Blacklist Algorithm pour 300 capteurs et écart-type = 500	70
Figure 43 - Une instance difficile du MLCP	74
Figure 44 - Une instance de la famille $\mathcal{F}$	74
Figure 45 - Une instance de la famille $\mathcal{F}$ où $n=3$	76
Figure 46 - Algorithme OddRingNetworks	81
Figure 47 - Exemple d'application des réseaux en anneau	83
Figure 48 - Comparaison des systèmes de surveillance de la qualité de l'air [83]	89
Figure 49 - Architecture générale du système	92
Figure 50 - Architecture d'un nœud de capteur	93
Figure 51 - Matériels utilisés pour la conception d'un nœud de capteur	94
Figure 52 - Architecture d'une station de base	95
Figure 53 - Station de base à partir de Raspberry Pi 3	95
Figure 54 - Eléments du protocole de communication	97
Figure 55 - Architecture logicielle du système	98
Figure 56 - Structure de la base de données	99
Figure 57 - Une interface Grafana	100
Figure 58 - Visualisation des métriques	100

Liste des tableaux

Tableau 1- Paramètres de simulation 1.....	51
Tableau 2 - Ratio d'énergie entre un capteur en veille et un capteur actif.....	57
Tableau 3 - Paramètres de simulation 2.....	63
Tableau 4 – Modes de fonctionnement d'un nœud de capteur	101
Tableau 5 – Consommation d'un nœud de capteur et ratio d'énergie	102

Liste des abréviations

COV: Composés Organiques Volatils

CSMN: Conventional Stationary Monitoring Network

CSN: Community Sensor Network

CSS: Chirp Spread Spectrum

GPL: Gaz de Pétrole Liquéfié

GPRS: General Packet Radio Service

GPS: Global Positioning System

GSM: Global System for Mobile communication

ISM: Industriel, Scientifique et Médical

LORA: Long Range

LPWA: Low-Power Wide-Area

MLCP : Maximum Lifetime Coverage Problem

RAM: Random Access Memory

RISC: Reduced Instruction Set Computer

SCADA: Supervisory Control And Data Acquisition

SGBD: Système de Gestion de Base de Données

SQL: Structured Query Language

SRAM: Static Random Access Memory

SSH: Secure Shell

SSN: Static Sensor Network

TDOA: Time Difference Of Arrival

TNGAPMS: The Next Generation Air Pollution Monitoring System

VNC: Virtual Network Connection

WSN: Wireless Sensor Network

Chapitre 1 Introduction

1.1 Contexte

Un réseau de capteurs sans fil ou WSN (*Wireless Sensor Network* en anglais) est un réseau sans fil constitué de plusieurs nœuds de capteurs et/ou d'actionneurs et d'une ou plusieurs stations de collecte ou puits [1], [2]. Les nœuds de capteurs sont des équipements électroniques peu coûteux, de petite taille et de faible puissance, capables de mesurer ou contrôler des phénomènes physiques (température, l'humidité, la luminosité, etc.) et de transmettre les données collectées vers une ou plusieurs stations de collecte via un canal sans fil. La station de collecte peut jouer le rôle de station de base et traiter l'information reçue localement ou jouer le rôle de passerelle et transmettre les données reçues vers des ordinateurs sur Internet pour traitement. Les réseaux de capteurs sont une ressource inestimable pour l'Internet des Objets ou IoT (*Internet of Things* en anglais) car ils fournissent les services d'acquisition de données aux utilisateurs de l'Internet des Objets [3], [4]. L'Internet des Objets est défini comme l'interconnexion entre les objets du quotidien et l'Internet dans le but de créer de nouvelles formes de communication entre les objets et les personnes ou entre les objets uniquement [5], [6]. En 2008, le nombre d'objets connectés à Internet a dépassé le nombre de personnes connectés à Internet et Cisco estime que d'ici 2020, le nombre d'objets connectés à Internet sera sept fois supérieur à la population mondiale [7]. Compte tenu de cette croissance accélérée de l'Internet des Objets, l'intérêt des communautés scientifique et industrielle pour les réseaux de capteurs sans fil ne cesse de croître.

Un nœud de capteur sans fil est composé de quatre modules principaux: un module de captage, un module de traitement, un module de communication sans fil et une source d'alimentation [8], [9]. Le module de captage est constitué d'un ou de plusieurs capteurs chargés de mesurer diverses données environnementales pouvant être transformées en grandeurs numériques. Il peut aussi contenir des actionneurs capables d'agir sur l'environnement (déclencher une commande d'arrosage, sonner une alarme, éteindre une lampe ...). Le module de traitement est composé d'un processeur et d'une mémoire. Le processeur est chargé de traiter les données et d'exécuter les programmes stockés dans la mémoire du nœud de capteur. Le module de communication sans fil utilise un canal sans fil pour échanger (émettre/recevoir) les informations avec la station de base ou avec d'autres nœuds de capteurs. La source d'énergie est une batterie ou une pile chargée d'alimenter les autres modules du nœud de capteur. L'architecture d'un nœud de capteur est illustrée à la figure 1.

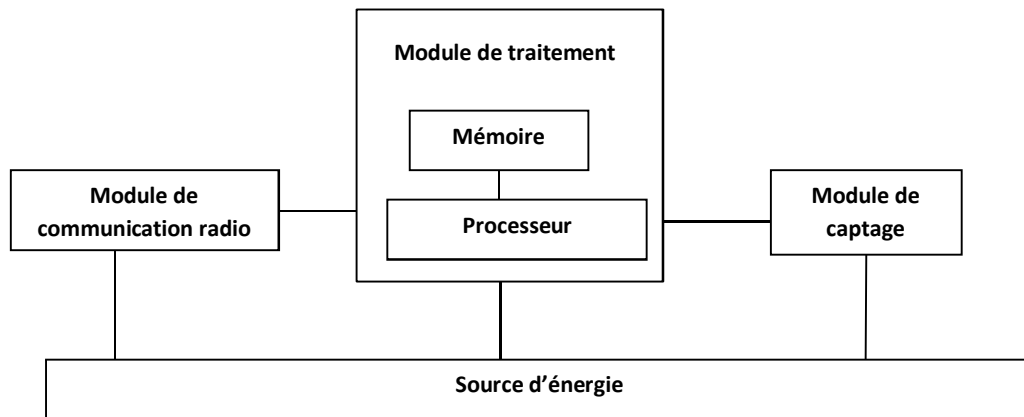


Figure 1- Les principaux modules d'un nœud de capteur

Les réseaux de capteurs sans fil ont de nombreuses applications dans divers domaines de la société. Parmi ces applications, nous pouvons citer les applications médicales, la surveillance environnementale, l'agriculture intelligente, les transports intelligents, les applications militaires, les applications industrielles, et les applications domotiques [10], [11]. Ces applications peuvent être regroupées en deux catégories en fonction de la mobilité des nœuds du réseau. On distingue alors les applications de suivi et les applications de couverture. Dans les applications de suivi, les nœuds de capteurs sont fixés sur des cibles mobiles et envoient des informations relatives aux mouvements de ces cibles (position, vitesse, direction) et à l'environnement autour des cibles (température, présence d'un gaz, ...) à la station de base. Parmi les applications de suivi, nous pouvons citer entre autres les applications de surveillance des troupes militaires, les applications de surveillance du bétail, les applications médicales de surveillance des patients à domicile. Dans les applications de couverture, les nœuds de capteurs sont statiques. Ces nœuds sont déployés sur une zone d'intérêt et transmettent à la station de base les informations relatives aux paramètres physiques à surveiller. Dans certains cas, toute la surface de la zone d'intérêt doit être surveillée, on parle de couverture de surface, tandis que dans d'autres cas, seuls certains points de la surface (appelés cibles) doivent être surveillés, on parle alors de couverture de cibles.

1.2 Défis liés à la mise en oeuvre des réseaux de capteurs sans fil

Malgré leur diversité, les applications des réseaux de capteurs soulèvent les mêmes défis, à savoir entre autres la qualité de service, la tolérance aux fautes, le passage à l'échelle, l'auto-maintenance, l'auto-configuration, la sécurité et la gestion de l'énergie [2], [12].

La qualité de service ou QoS (Quality of Service) fait référence aux capacités réseau qui résultent en la satisfaction des utilisateurs. Elle est définie comme un ensemble d'exigences de service à satisfaire par le réseau lors du transport d'un flux de paquets de la source à la destination. C'est donc une garantie par le réseau de fournir à l'utilisateur un ensemble de services prédéfinis et mesurables comme le délai, la bande passante et le taux de perte de paquets. La tolérance aux fautes est une propriété importante des réseaux de capteurs car les nœuds de capteurs peuvent être à court d'énergie ou être endommagés. Il est alors nécessaire que le réseau de capteurs dans son ensemble

soit capable de tolérer ce type de pannes. Une des méthodes utilisées pour rendre le réseau robuste est de déployer plus de nœuds de capteurs que nécessaire afin d'exploiter la redondance. La robustesse du réseau suppose aussi l'auto-maintenance et l'auto-configuration du réseau. En effet, en cas de panne d'un nœud, la configuration du réseau change et les autres nœuds doivent se réorganiser afin de continuer d'assurer leur service sans une intervention humaine. La configuration du réseau peut aussi changer du fait de l'ajout d'un ou plusieurs nœuds de capteurs selon les applications. Le réseau doit alors être capable de gérer le passage à l'échelle c'est-à-dire les situations où le nombre de nœuds est très élevé, comme par exemple dans les applications de détection des feux de forêts. Par ailleurs, comme pour tout réseau sans fil, la question de la sécurité se pose dans les réseaux de capteurs. On distingue deux aspects de sécurité : la sécurité des données transmises en termes de confidentialité et d'authenticité, et la sécurité des nœuds de capteurs qui peuvent être endommagés (voire détruits), attaqués (pour en extraire les données) ou manipulés (en installant des programmes malveillants). Enfin, la gestion de l'énergie est une question cruciale pour les applications des réseaux de capteurs où les nœuds sont déployés dans des zones hostiles, ou difficiles d'accès ou lorsque le réseau a une forte densité. Dans ces applications, la ressource énergétique est très limitée car il est difficile voire impossible de recharger ou de remplacer les batteries des nœuds de capteurs une fois qu'ils sont déployés.

Du fait de leur forte contrainte énergétique, une problématique majeure des réseaux de capteurs est la durée de vie du réseau. [13]. La durée de vie peut être liée au taux de perte des nœuds du réseau et se définit alors comme la durée jusqu'à ce que le nombre de nœuds actifs soit inférieur à un certain seuil. La durée de vie du réseau peut aussi être liée à la couverture de cibles. Dans ce cas, elle se définit comme la durée pendant laquelle le réseau garantit un taux de couverture de la zone d'intérêt supérieur à un seuil donné. Ainsi l'optimisation de la durée de vie d'un réseau de capteurs est directement liée à l'optimisation de la consommation énergétique des nœuds de capteurs.

Il existe plusieurs techniques d'optimisation de la consommation d'énergie des nœuds de capteurs [14]. Etant donné que le module de communication sans fil est celui qui consomme le plus d'énergie, l'objectif principal de ces techniques d'optimisation est de minimiser le temps d'activité du module radio des différents nœuds du réseau. On peut regrouper ces techniques en deux classes : celles basées sur la réduction de données et celles basées sur le basculement veille/activité. Dans les techniques de réduction de données, l'idée est de réduire la taille des données transmises au puits. Les techniques de veille/activité les plus connues sont le facteur de marche (*duty cycle* en anglais) et le contrôle de topologie. La méthode de facteur de marche consiste à adapter l'état du module radio du nœud de capteur en fonction de l'activité du réseau afin de minimiser l'écoute passive (*idle listening* en anglais) et d'augmenter le temps de veille. Ainsi, le nœud de capteur peut par exemple activer son module radio sur demande lorsqu'il a un message à émettre ou à recevoir, puis le remettre en veille. La technique de contrôle de topologie exploite la redondance pour adapter dynamiquement la topologie du réseau en minimisant le nombre de nœuds actifs tout en garantissant les besoins de l'application [15], [16], [17]. Ainsi, lorsque les nœuds de capteurs sont déployés de façon redondante, certains nœuds sont mis en veille (module radio en veille) tandis que d'autres, actifs, assurent les fonctions du réseau. La figure 2 illustre la technique de contrôle de topologie. Afin d'assurer la couverture de la zone d'intérêt, un seul capteur de chaque carré reste actif tandis que les autres capteurs sont inactifs. Cette technique est très utile dans les applications

de couverture telles que la surveillance environnementale et l'agriculture intelligente. Les deux techniques de veille/activité peuvent être combinées de sorte que les nœuds actifs de la méthode de contrôle de topologie appliquent la méthode de facteur de marche pour réaliser leurs tâches.

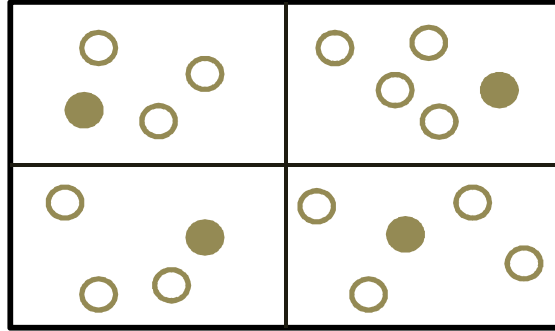


Figure 2 - Exemple de contrôle de topologie

1.3 Objectifs de la thèse

Dans cette thèse, nous nous intéressons au problème de l'optimisation de la consommation d'énergie des réseaux de capteurs dans le cas des applications de couverture de cibles. Notre objectif est de proposer des solutions permettant de prolonger la durée de vie du réseau, c'est-à-dire la durée pendant laquelle chaque cible est couverte par au moins un capteur. Nous considérons que la technologie radio utilisée par le réseau est une technologie longue portée et faible consommation (LPWA). De ce fait, nous ne tenons pas compte des problématiques de connexité du réseau et de routage puisque chaque nœud de capteur peut atteindre la station en un saut sans l'intermédiaire des autres nœuds du réseau. Par ailleurs, nous considérons un déploiement aléatoire des nœuds de capteurs. Ainsi, les problématiques de dimensionnement et de déploiement déterministe visant à optimiser la taille du réseau ne sont pas abordés dans ce travail.

L'approche d'optimisation de l'énergie que nous considérons est basée sur la veille/activité. En bref, il est question dans cette thèse de proposer des méthodes pour former des sous-ensembles de nœuds de capteurs qui seront activés successivement pendant des durées définies tandis que les autres nœuds seront en veille, de sorte à maximiser la durée pendant laquelle chaque cible sera couverte par au moins un nœud capteur. Ce problème connu sous le nom de MLCP (*Maximum Lifetime Coverage Problem* en anglais) a été démontré NP-difficile et largement étudié dans la littérature sans prise en compte de l'énergie consommée par les capteurs qui sont en veille [18], [19], [20]. Une approche de résolution de MLCP est de chercher à former le nombre maximal de couvrants disjoints. Pour cela, une idée est de former consécutivement des couvrants de taille minimale [21], mais cette méthode ne permet pas toujours d'obtenir le nombre maximal de couvrants disjoints. Une seconde méthode est d'utiliser la programmation linéaire [22], mais sa complexité est très élevée voire exponentielle. Une troisième méthode est l'approche génétique [23], [24] qui est stochastique et par conséquent, il est difficile de connaître sa performance. En utilisant la technique gloutonne, il est possible de construire les couvrants en se focalisant à chaque étape sur la cible la moins couverte appelée cible critique ([25], [20]), cependant il peut y avoir

plusieurs cibles critiques. Une autre approche pour le MLCP est de former le nombre maximal de couvrants non disjoints de sorte que chaque capteur appartienne à au plus w couvrants. L'approche par couvrants non disjoints permet d'obtenir une durée de vie du réseau plus longue que celle obtenue par l'approche par couvrants disjoints. Les algorithmes qui implémentent cette approche construisent les couvrants de façon gloutonne en se basant sur la relation entre un nœud capteur et les cibles critiques [26]. Cependant, ces algorithmes utilisent des paramètres dont les valeurs adéquates dépendent de l'instance du problème. Nous proposons dans ce travail une nouvelle approche de construction des couvrants, l'approche réactive, qui construit un nouveau couvrant uniquement en cas de panne du couvrant en activité. Cette nouvelle approche permet de s'adapter aux instances où les nœuds de capteur n'ont pas la même durée de vie et aux instances où la durée de vie des nœuds n'est pas connue a priori. Une première implémentation de cette approche consiste à remplacer dans le couvrant précédent, le nœud de capteur en panne ou à cours d'énergie et à maintenir tous les autres nœuds de ce couvrant en activité : c'est l'algorithme adaptatif. Une seconde implémentation de la nouvelle approche consiste à définir lors de la construction d'un couvrant, une liste noire de nœuds de capteur à éviter afin de réduire le nombre de nœuds de capteur qui couvrent les cibles critiques : c'est l'algorithme blacklist.

Dans la pratique, un nœud de capteur en veille doit rallumer son module radio de temps en temps (sur demande ou sur rendez-vous) pour pouvoir communiquer avec les autres nœuds notamment avec la station de base et ainsi pouvoir participer à l'activité du réseau au moment opportun. Nous proposons des algorithmes approchés pour le MLCP lorsque la consommation énergétique des capteurs en veille est non négligeable. Dans ce contexte, on observe qu'en activant successivement les nœuds qui couvrent une cible dans l'ordre croissant de leurs énergies, le temps total pendant lequel cette cible est couverte est maximal. En nous appuyant sur cette observation et sur le concept de liste noire, nous proposons un nouvel algorithme approché basé sur l'approche réactive, l'algorithme Energy-blacklist. Cependant, nous constatons que cet algorithme ne donne pas de bons résultats pour la famille des réseaux réguliers constitués d'anneaux de taille impaire. Nous proposons une méthode analytique pour cette famille en nous basant sur les autormorphismes de graphe.

1.4 Organisation du mémoire

La suite de ce mémoire est organisée de la manière suivante. Dans le chapitre 2, nous étudions les réseaux de capteur sans fil. Pour cela, nous présentons en détail les différents modules d'un nœud de capteur ainsi que la consommation énergétique de chacun d'eux. Après avoir donné quelques exemples de plateformes de nœuds de capteur, nous présentons quelques applications basées sur les réseaux de capteurs sans fil. Cette étude nous permet de comprendre l'importance de l'optimisation de la consommation d'énergie des réseaux de capteurs sans fil.

Dans le chapitre 3, nous introduisons le problème de couverture et nous présentons le problème de la maximisation de la durée de vie du réseau sous contrainte de couverture (MLCP). Nous étudions ensuite les différents algorithmes approchés pour MLCP présents dans la littérature et nous examinons les limites de chacun d'eux. Cela nous amène à constater que ces algorithmes ne sont pas adaptés au cas où les nœuds de capteur n'ont pas la même durée de vie ou lorsque les durées de

vie des nœuds ne sont pas connus à l'avance. Or ces situations sont celles que l'on rencontre le plus en environnement réel.

Dans le chapitre 4, nous supposons que les nœuds n'ont pas la même durée de vie et nous proposons une nouvelle approche de résolution de MLCP, l'approche réactive. Cette approche construit un couvrant uniquement après la défaillance du couvrant en activité, ce qui permet de s'adapter au cas où les durées de vie des nœuds ne sont pas connues a priori. Nous proposons ensuite deux algorithmes basés sur l'approche réactive à savoir: l'algorithme adaptatif et l'algorithme blacklist. L'algorithme adaptatif cherche à réparer localement la panne du couvrant défectueux tandis que l'algorithme blacklist met en liste noire les nœuds susceptibles de raccourcir la durée de vie du réseau. Les résultats de simulations montrent que ces deux algorithmes sont meilleurs que les algorithmes existants dans la littérature.

Dans le chapitre 5, nous considérons l'énergie consommée par les capteurs en veille et nous montrons que la durée de couverture d'une cible est maximale lorsque les capteurs qui la couvrent sont activés dans l'ordre croissant de leurs énergies initiales. Partant de ce constat et en utilisant l'idée de liste noire, nous proposons un nouvel algorithme pour le MLCP : l'algorithme Energy-blacklist. A travers les résultats des simulations, nous constatons que cet algorithme est meilleur que les précédents lorsque la consommation d'énergie des nœuds en veille est non nulle.

Dans le chapitre 6, nous définissons la famille des réseaux réguliers constitués par les anneaux de taille impaire qui est une famille d'instances difficiles de MLCP pour lesquelles les algorithmes précédents et ceux de la littérature ne trouvent pas de solutions optimales. Nous proposons une méthode analytique basée sur les automorphismes de graphe et sur une propriété fondamentale des réseaux réguliers. Nous conjecturons que cette méthode est optimale pour les instances où les nœuds ont la même énergie initiale.

Dans le chapitre 7, nous nous intéressons aux réseaux de capteurs sans fil appliqués à la surveillance de la pollution atmosphérique. Nous faisons l'état de l'art des systèmes de surveillance de la qualité de l'air basés sur les réseaux de capteurs sans fil et nous implémentons un système de ce type. En plus de surveiller la qualité de l'air, notre système permet de détecter les incendies. Ce système intègre un algorithme du MLCP permettant d'augmenter l'autonomie énergétique et la robustesse du réseau de capteurs lorsque les nœuds de capteurs sont redondants.

Dans le chapitre 8, nous concluons cette thèse en rappelant les enjeux et les principales contributions. Puis nous présentons quelques perspectives.

Chapitre 2 Réseaux de capteurs sans fil

Les nœuds de capteur sans fil sont les éléments de base d'un réseau de capteurs sans fil. Etudier les modules d'un nœud de capteur permet de comprendre le fonctionnement des réseaux de capteurs et les problématiques rencontrés dans ce type de réseau, notamment la problématique d'énergie. Dans ce chapitre, nous présentons les modules d'un nœud de capteur ainsi que leur consommation énergétique. Par la suite, nous citons des exemples de plateformes de nœuds de capteur et nous présentons quelques unes des nombreuses applications des réseaux de capteurs sans fil qui existent de nos jours.

2.1 Modules d'un nœud de capteur sans fil

2.1.1 Le module de traitement

Le processeur est le cœur du nœud de capteur. Son rôle est de contrôler l'ensemble des procédures permettant à un nœud capteur de réaliser les tâches d'acquisition, de traitement, de stockage et de transmission des données collectées. Dans certains cas, le processeur peut également recevoir, analyser, agréger et relayer des données capturées par d'autres nœuds de capteurs afin d'alléger certaines tâches à la station de base ou pour des besoins de routage. Ainsi, il est responsable de l'exécution des différents protocoles de communication, des programmes de traitement de données et du système d'exploitation. Pour gérer l'interaction entre les différents modules, le processeur est constitué de deux interfaces qui sont l'interface avec le module de captage et l'interface avec le module de communication radio.

La mémoire d'un nœud de capteur contient différentes parties : une mémoire vive et une mémoire non volatile. La mémoire vive du nœud de capteur est la RAM (Random Access Memory). Elle sauvegarde entre autres les mesures prélevées par le capteur et les paquets reçus d'autres nœuds de capteurs. La mémoire RAM est très rapide mais son principal inconvénient est de perdre les données qu'elle contient en cas d'interruption de l'alimentation électrique. Le code des programmes est stocké dans la mémoire flash. La mémoire flash peut aussi stocker les données lorsque la mémoire RAM est insuffisante ou lorsque l'alimentation de la mémoire RAM doit être interrompue pour un certain temps. La mémoire flash nécessite un temps de lecture et d'écriture plus long et une consommation d'énergie plus élevée que la mémoire RAM.

Du point de vue matériel, le module de traitement est généralement de type microcontrôleurs car leurs caractéristiques sont adaptées aux systèmes embarqués. En effet, les microcontrôleurs sont faciles à connecter à d'autres équipements (capteurs par exemple), ils ont une faible consommation d'énergie, ils sont programmables et intègrent généralement une mémoire. De plus, les microcontrôleurs offrent la possibilité de réduire leur puissance de consommation grâce aux états dits de veille où seules certaines parties du microcontrôleur sont en marche tandis que les autres

sont à l'arrêt. Parmi les microcontrôleurs utilisés pour la conception des nœuds de capteur, nous pouvons citer les microcontrôleurs des familles Atmel ATmega et MSP430. Atmel ATmega est un microcontrôleur 8 bits conçu pour les systèmes embarqués et équipé d'interfaces externes pour les périphériques usuels (écran LCD, moteur, capteur, ...). MSP430 est une famille de microcontrôleurs 16 bits réputé pour sa faible consommation énergétique, il consomme beaucoup moins d'énergie que le microcontrôleur Atmel ATmega.

Un microcontrôleur fonctionne sous différents modes ou états. Chaque état est caractérisé par son degré de fonctionnalités (nombre de circuits actifs), sa puissance électrique de consommation et son temps de réveil (temps nécessaire pour retourner au mode de plein fonctionnement). En général, il existe trois états : Actif (Active), Idle et Veille (Sleep). Certains microcontrôleurs possèdent plusieurs états de veille. Dans ce cas, l'état de veille qui consomme le moins d'énergie est une veille plus prolongée. Les transitions entre les états nécessitent à la fois du temps et de l'énergie. Moins un état consomme de l'énergie et plus la transition vers le mode de plein fonctionnement consomme du temps et de l'énergie. Ainsi, il est parfois plus bénéfique de rester dans un état qui consomme plus plutôt que d'entrer dans un état qui consomme moins. Cela est illustré par l'exemple ci-après. Supposons qu'un microcontrôleur consomme les puissances suivantes : P_{active} en état Actif, P_{sleep} dans l'état de veille et $\frac{P_{sleep}+P_{active}}{2}$ lors de la transition entre les deux états. Supposons τ_{up} et τ_{down} les durées respectives de la transition de l'état de veille à l'état actif et inversement, t_1 l'instant où l'on décide de mettre le microcontrôleur en veille et t_{event} l'instant où le prochain événement se produit. Placer le microcontrôleur en veille entre les deux instants ne permet d'économiser de l'énergie que si : $(t_{event} - t_1) > \frac{1}{2} (\tau_{down}) + \frac{P_{active}+P_{sleep}}{P_{active}-P_{sleep}} + \tau_{up}$. Bien que les transitions entre les états consomment de l'énergie, il est toutefois primordial de mettre le microcontrôleur en veille dès que la date du prochain événement n'est pas imminente. En effet, supposons qu'un nœud de capteur soit alimenté par une batterie capable de fournir environ 1J et que le microcontrôleur soit toujours dans l'état actif (n'entre jamais en veille). Pour que la batterie soit capable d'alimenter le microcontrôleur pendant seulement une journée, celui-ci doit fonctionner à une puissance inférieure à $\frac{1}{24*60*60} \approx 11.5 \mu W$. Or, il n'existe pas de microcontrôleur consommant une puissance aussi faible dans son état actif. A titre d'exemple, le microcontrôleur MSP430 consomme environ 1.2 mW dans son état actif et 0.3 μW dans son état de veille le plus prolongé. Tandis que le microcontrôleur Atmel ATmega consomme environ 75 μW en mode veille et 15 mW en mode actif.

2.1.2 Le module de communication

Le module de communication est utilisé pour échanger les données entre un nœud de capteur et les autres éléments du réseau (nœuds de capteur et station de collecte). Plusieurs médias de transmission sans fil sont envisageables dans ce type de communication à savoir la fréquence radio, la communication optique et l'ultrason. La transmission par fréquence radio est la mieux adaptée aux réseaux de capteurs car ses caractéristiques correspondent mieux aux besoins des applications des réseaux de capteurs. La communication par fréquence radio a une longue portée, un débit de transmission de données élevé, un taux d'erreur acceptable et une faible consommation d'énergie. Le module de communication radio d'un nœud de capteur est composé d'un émetteur et d'un

récepteur. L'émetteur convertit une série de bits ou d'octets provenant du microcontrôleur en onde radio. Inversement, le récepteur convertit l'onde radio reçue en une série de bits ou d'octets et l'envoie au microcontrôleur. Un transceiver est un équipement électronique qui joue à la fois le rôle d'émetteur et de récepteur. Tout comme le microcontrôleur, le transceiver peut fonctionner sous différents modes. La plupart des transceivers distinguent quatre modes de fonctionnement : *Transmit*, *Receive*, *Idle* et *Sleep*. Dans l'état *Transmit*, l'émetteur est actif et l'antenne irradie de l'énergie. Dans l'état *Receive*, le récepteur est actif et en cours de réception de données. Lorsque le récepteur est actif sans recevoir effectivement des données, il est dans l'état *Idle*. Le transceiver est dans l'état *Sleep* lorsque plusieurs de ces composants (notamment l'émetteur et le récepteur) sont éteints. Certains transceivers ont plusieurs états *Sleep* qui se distinguent par le nombre de circuits éteints, le temps et l'énergie nécessaires pour quitter l'état.

Les transceivers sont utilisés pour communiquer selon une technologie sans fil. On peut regrouper les technologies radio en deux catégories : les technologies traditionnelles courte distance et les technologies faible puissance longue distance ou LPWA (Low Power Wide Area).

Parmi les technologies radio traditionnelles utilisées dans les réseaux de capteurs, on peut citer WIFI, Bluetooth et Zigbee. Les normes WiFi offre une portée d'environ 300 mètres et un débit de 11 Mbps à quelques Gbps). Par contre, cette technologie consomme beaucoup d'énergie et la durée de vie des nœuds de capteurs qui l'utilisent est limitée à quelques heures. Le Bluetooth offre aux nœuds de capteurs une durée de vie plus longue (quelques jours) que celle du WiFi. Il a une portée d'environ 100 mètres et offre un débit jusqu'à 3 Mbps. Cependant la taille du réseau qu'il peut former est limitée à 8 nœuds, y compris la station de base. Zigbee présente un faible coût énergétique ce qui prolonge théoriquement la durée de vie d'un nœud de capteur sur plusieurs années. De plus, il permet un réseau plus dense (jusqu'à 65536 nœuds) et une portée moyenne de 100 mètres avec un débit maximal de 250 Kbps. Les technologies radio traditionnelles ne sont pas idéales pour connecter des équipements de faible énergie répartis sur une grande superficie à cause de leur portée limitée à une centaine de mètres. La portée de ces technologies est améliorée en déployant un grand nombre de nœuds de capteurs qui sont connectés à la station de base par des communications multisaut. Ce surplus de nœuds entraîne une augmentation du coût de déploiement du réseau. Par ailleurs, le routage de messages réalisé au niveau de chaque nœud pour relayer les messages des nœuds voisins vers la station de base entraîne une consommation supplémentaire d'énergie [27]. La figure 3 présente la topologie maillée d'un réseau de capteurs sans fil utilisant une technologie radio courte portée.

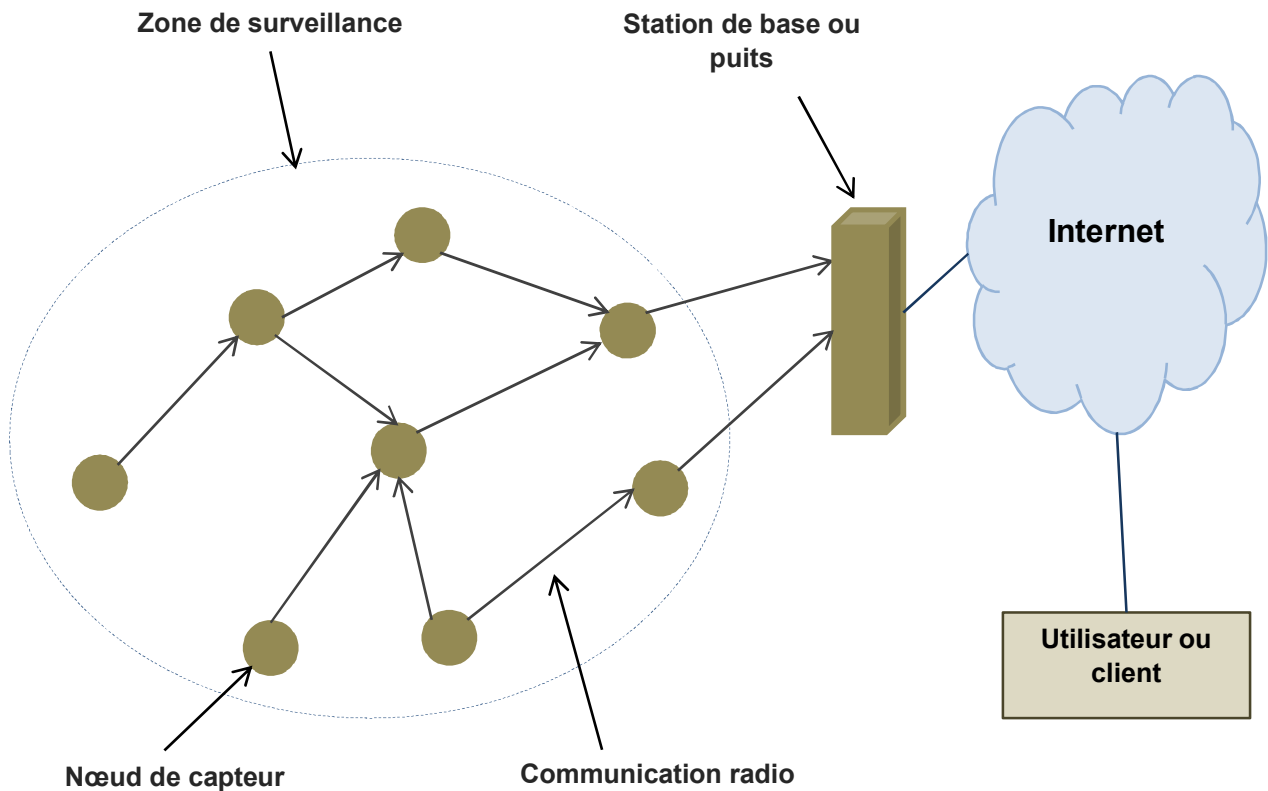


Figure 3 - Topologie d'un réseau utilisant une technologie radio courte portée

Contrairement aux technologies radio traditionnelles, les technologies LPWA permettent de connecter un grand nombre d'équipements répartis sur une longue distance, avec une faible consommation d'énergie et à faible coût [28], [29]. Les technologies LPWA offrent une portée allant de quelques kilomètres en zone urbaine à des dizaines de kilomètres en zone rurale [30]. Cette longue portée permet d'avoir une topologie en étoile où chaque nœud de capteur communique directement (à un saut) avec la station de base. Grâce à la topologie en étoile, non seulement les nœuds ne gaspillent plus leur énergie dans la réception et la retransmission des messages de leurs voisins, mais en plus ils n'ont plus besoin d'écouter fréquemment le canal pour savoir s'il y'a un message à retransmettre. Ainsi, les technologies LPWA permettent de réduire le temps passé dans les états *Transmit*, *Receive* and *Idle*, ce qui réduit la consommation énergétique du nœud de capteur. Les technologies LPWA sont moins coûteuses que les technologies traditionnelles car elles ne requièrent pas de nœuds supplémentaires utilisés comme relais vers la station de base. Parmi les technologies LPWA émergentes, on peut citer SIGFOX, LoRa, INGENU et TELENESA [28].

SIGFOX [31] offre une solution de connectivité LPWA basée sur ses technologies brevetées. Les opérateurs de réseau SIGFOX déploient les stations de base propriétaires et les relient à des serveurs (backend) connectés à Internet (cf. figure 4). Les nœuds de capteurs se connectent à ces stations de base en utilisant la modulation BPSK (Binary Phase Shift Keying) sur les fréquences 868MHz/902MHz. SIGFOX utilise la bande passante efficacement et subit de très faibles niveaux de bruit, ce qui entraîne une sensibilité élevée du récepteur, une consommation d'énergie ultra-faible

et un faible coût de conception de l'antenne. Cependant, son débit de 100 bps est plus faible que celui offert par la plupart des autres technologies LPWA, ce qui limite le nombre de cas d'utilisation pour SIGFOX. En outre, SIGFOX initialement ne supportait que la communication montante (uplink), mais a évolué plus tard vers une technologie bidirectionnelle. Une communication descendante (downlink) n'est possible qu'après une communication montante. A cause des régulations sur la bande libre ISM (Industrial, Scientific, Medical), les communications montantes du nœud de capteur vers la station de base sont limitées par jour à 140 messages de 12 octets chacun et les communications descendantes à 4 messages de 8 octets chacun. De ce fait, il n'est pas possible pour la station de base d'acquitter tous les messages reçus. Sans support adéquat pour les acquittements, la fiabilité de la communication montante est améliorée en utilisant des transmissions redondantes. Un message unique provenant d'un nœud de capteur est transmis plusieurs fois sur différents canaux de fréquence. La technologie SIGFOX fournit une portée de communication de 10 km en zone urbaine à 50 km en zone rurale [28], [31].

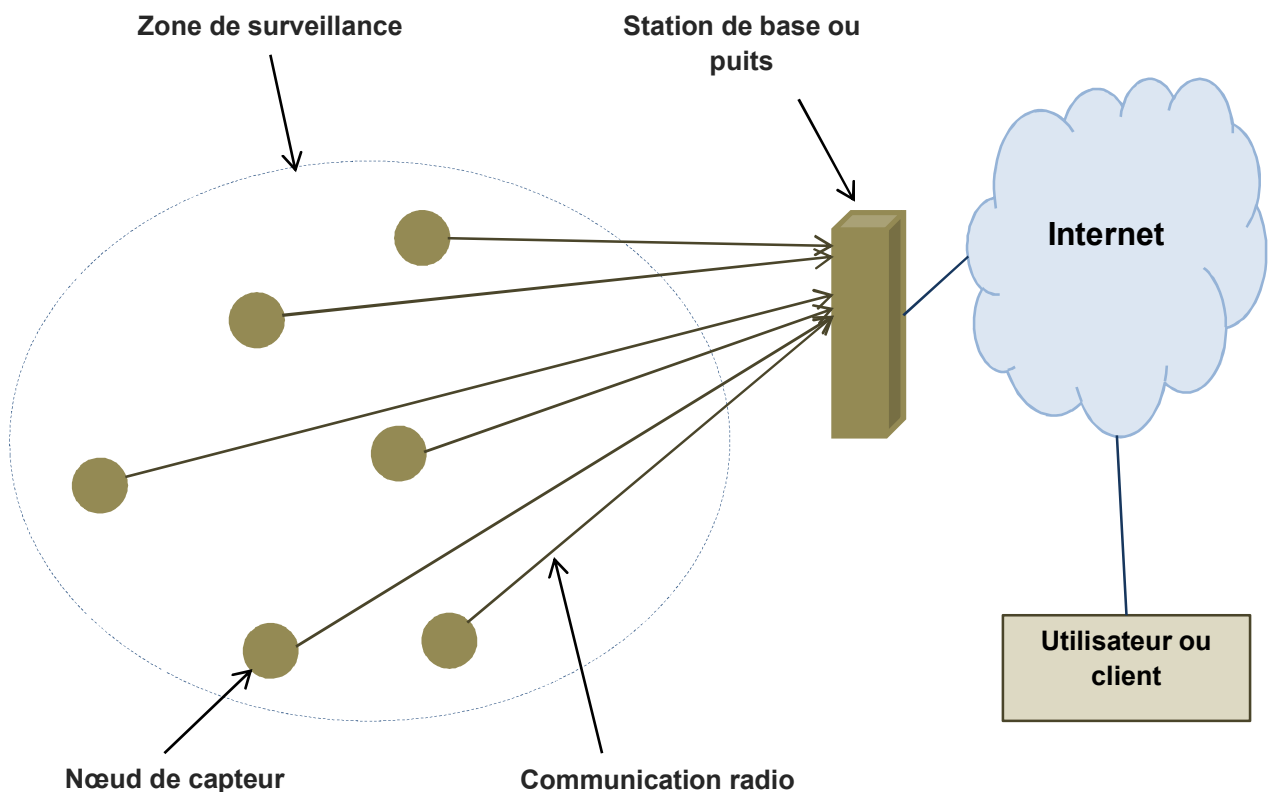


Figure 4 - Topologie d'un réseau utilisant la technologie radio SIGFOX

LoRa [32] est une technologie radio qui module les signaux dans la bande de fréquence 433MHz, 868 MHz/915MHz ISM en utilisant une technique propriétaire d'étalement de spectre développée et commercialisée par Semtech Corporation. Une communication bidirectionnelle est fournie par cette technique spéciale appelée CSS (Chirp Spread Spectrum) qui étale un signal d'entrée à bande étroite sur un canal de plus large bande passante. LoRa prend en charge plusieurs facteurs d'étalement (entre 7-12) pour décider du compromis entre la portée et le débit de données. Des

facteurs d'étalement plus élevés offrent une longue portée à un faible débit et vice versa. Le débit de données varie de 250 bps à 50 Kbps selon le facteur d'étalement et la bande passante du canal. En outre, plusieurs transmissions utilisant différents facteurs d'étalement peuvent être reçues simultanément par une station de base LoRa. En substance, les facteurs d'étalement fournissent un troisième degré de diversité après le temps et la fréquence. LoRa offre une portée de communication allant de 5 km en zone urbaine à 15 km en zone rurale. Les messages transmis par un nœud de capteur sont reçus par toutes les stations de base dans sa portée de communication, donnant lieu à une topologie en étoile d'étoiles (star-of-stars). Les messages reçus par les stations de base sont transmis vers un serveur central qui filtre ensuite les messages redondants. LoRa améliore le taux de messages reçus avec succès grâce à la réception multiple. De plus, LoRa exploite la réception multiple du même message par différentes stations de base pour la déterminer la position de nœud de capteur émetteur. A cet effet, la technique de localisation basée sur la différence de temps d'arrivée (TDOA) est utilisée au niveau des stations de base qui ont des horloges synchronisées entre elles. Un groupe d'intérêt appelé LORa™ Alliance et constitué de plusieurs partenaires industriels et commerciaux ont proposés LORAWAN, un standard qui définit l'architecture et les couches protocolaires au-dessus de la couche physique LoRa. LORAWAN définit trois classes de nœuds, la classe A, la classe B et la classe C. Les classes permettent de gérer le compromis entre la latence de la communication descendante et l'économie d'énergie. Le choix de la classe des nœuds du réseau dépend des applications et de leurs besoins. Les nœuds de la classe A permettent une communication bidirectionnelle où la transmission montante de chaque nœud est suivie de deux fenêtres courtes de réception de communication descendante. Le nœud initie une communication avec la station de base selon ses propres besoins. Les nœuds de classe A consomment peu d'énergie mais le temps de latence de la communication descendante est grand car toute communication de la station de base vers le nœud doit attendre jusqu'à la prochaine communication montante du nœud. En plus des fenêtres de réception des nœuds de classe A, les nœuds de classe B écoutent le canal pendant d'autres fenêtres planifiées périodiquement et dont les dates sont connues par la station de base. La planification des fenêtres se réalise à la réception d'une balise de synchronisation émise par la station de base. Ce mode de fonctionnement réduit le temps de latence de la communication descendante mais induit une consommation d'énergie supplémentaire car le nœud écoute plus longtemps le canal. Les nœuds de classe C sont ceux qui n'ont pas de contrainte d'énergie, ils écoutent continuellement le canal sauf quand ils émettent un message [33], [34]. La figure 5 présente la topologie en étoile d'étoiles d'un réseau de capteurs utilisant la technologie LoRa.

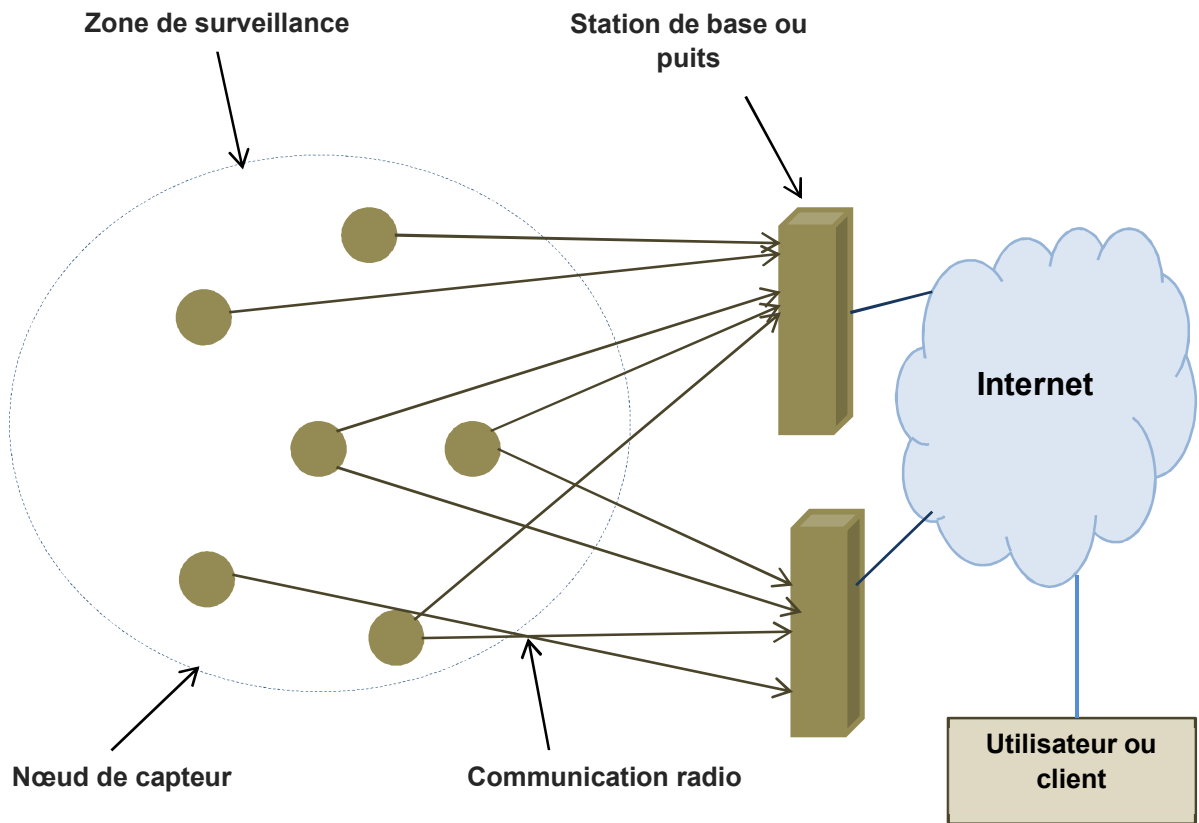


Figure 5 - Topologie d'un réseau utilisant la technologie radio LoRa

2.1.3 Le module de captage

Il existe une large gamme de paramètres physiques que les capteurs peuvent mesurer. Ces capteurs peuvent être regroupés en trois catégories [8]:

Les capteurs passifs et omnidirectionnels : Ce type capteur mesure une grandeur physique sans manipuler l'environnement par des sondages. Dans ce sens, il est dit passif. D'autre part, il n'existe pas de notion de direction car le paramètre physique peut être mesuré dans toutes les directions autour du capteur. Quelques exemples de capteurs de ce type sont les capteurs de température, les capteurs de fumée, les capteurs de lumière, de vibration, d'humidité et de produits chimiques. La figure 6 présente des exemples de capteur de température et de capteur de fumée.



Capteur de température DS18B20



Capteur de fumée MQ2

Figure 6 - Exemples de capteurs omnidirectionnels

Les capteurs passifs et directionnels : Ces capteurs sont aussi passifs mais ne peuvent mesurer un phénomène que dans une direction donnée. Un exemple de ce type de capteur est une caméra qui ne peut prendre des images que dans une direction particulière.

Les capteurs actifs : Les capteurs de cette catégorie doivent sonder activement l'environnement. Par exemple un radar génère des ondes de choc par de petites explosions.

Il est difficile d'estimer la consommation énergétique des capteurs à cause de la grande diversité de ces équipements. Pour certains capteurs tel que les capteurs de température ou de lumière, la consommation peut être ignorée en comparaison à celle des autres modules du nœud de capteur. Par contre les capteurs actifs tels que les sonars, consomment beaucoup d'énergie et peuvent nécessiter une source d'alimentation différente de celle du nœud de capteur. Estimer la consommation des capteurs requiert de connaître non seulement la nature des capteurs mais aussi l'application concernée car pour une courte période entre les prélèvements, le capteur consomme plus d'énergie que pour une période plus longue.

2.1.4 La source d'énergie

L'alimentation en énergie du nœud de capteur comporte essentiellement deux aspects : la conservation de l'énergie et l'approvisionnement en énergie. Il existe de nombreuses techniques pour économiser l'énergie du nœud de capteur conservée dans la batterie [14], [35]. Lorsque l'énergie disponible dans la batterie est épuisée, le nœud de capteur tombe en panne. Des techniques d'extraction d'énergie à partir de l'environnement sont utilisées pour augmenter l'autonomie énergétique des nœuds de capteurs. Les cellules solaires, la variation de température, les vibrations, la variation de pression, la circulation de l'air ou de fluides sont des exemples de sources d'énergie utilisées pour recharger les batteries des nœuds de capteurs [36], [37], [38].

2.2 Plateformes de nœud de capteur

Plusieurs plateformes de nœuds de capteurs ont été développées [39]. Certaines sont exploitées dans le cadre de projets de recherche académique et de développement tandis que d'autres sont des solutions commerciales. Parmi les plus populaires, nous pouvons citer MICA, EYES, TelosB et BTnode. Tous ces nœuds sont dotés de transceivers fonctionnant avec une technologie radio courte portée à savoir ZigBee. Les nœuds de la famille Mica sont Mica, Mica2, Mica2Dot. Ces nœuds de capteur possèdent un microcontrôleur de la famille Atmel et plusieurs capteurs (lumière, température,...). Il est aussi possible de les connecter à d'autres capteurs supplémentaires. Les nœuds de capteur EYES sont équipés de microcontrôleurs MSP430, et d'interfaces USB. Les nœuds BTnode sont dotés de microcontrôleurs Atmel ATmega et de mémoire flash de 128 Ko. Ces nœuds de capteurs fonctionnent avec des systèmes d'exploitation différents de ceux utilisés par les ordinateurs. En effet, les nœuds de capteurs ont une mémoire, une puissance de calcul et une ressource énergétique très limitées contrairement aux ordinateurs. Les systèmes d'exploitation les plus utilisés sur les différentes plateformes de nœuds de capteurs sont TinyOS et CONTIKI [8]. TinyOS est un système d'exploitation open-source spécialement développé pour les applications embarquées fonctionnant en réseau et, en particulier, pour les réseaux de capteurs sans fil. Contiki est un autre système d'exploitation open source multitâche développé pour une série de plate-

formes comprenant des microcontrôleurs tels que MSP430 et Atmel. La figure 7 présente un exemple de nœud de capteur MicaZ et TelosB.



Figure 7 - Exemples de nœuds de capteurs

Une alternative aux précédentes plateformes pour nœuds de capteur est de réaliser les nœuds de capteurs en assemblant divers équipements électroniques correspondant aux différents modules. En effet, il existe plusieurs cartes de développement dont la plus connue est la famille Arduino (UNO, DUO, MEGA, ...) qui contiennent un microcontrôleur, une interface USB pour la programmation et un ensemble de port permettant l'ajout d'autres périphériques. La carte de développement Arduino UNO [40] est dotée d'un microcontrôleur ATmega328P avec un processeur RISC 8 bits, 32 Ko de mémoire flash, et 2 Ko de mémoire vive statique (SRAM). Il existe aussi des cartes de communication radio notamment de type LORA qui peuvent être facilement connectées aux cartes Arduino [41]. Enfin, il est possible d'obtenir et de connecter facilement plusieurs types de capteurs (température, gaz, image, son, etc.) aux cartes Arduino.

2.3 Applications des réseaux de capteurs

Les réseaux de capteurs sans fil ont de nombreuses applications dans divers domaines de la société. Parmi ces applications, nous pouvons citer les applications médicales, la surveillance environnementale, l'agriculture intelligente, les transports intelligents, les applications militaires, les applications industrielles, et les applications domotiques [10], [11].

Les applications des réseaux de capteurs sans fil dans le domaine médical peuvent être regroupées en trois catégories : la surveillance des patients en environnement hospitalier, la surveillance à domicile des patients âgés ou atteints de maladies chroniques et enfin, la collecte à long terme des données pour des études cliniques [42], [43], [44]. Les réseaux de capteurs sans fil sont utilisés dans les hôpitaux pour la surveillance des signes vitaux des patients. Ces réseaux de capteurs ont plusieurs avantages par rapport au système traditionnel filaire de surveillance des patients. Parmi ces avantages, nous pouvons citer entre autres la facilité d'utilisation, la réduction des risques d'infections, la réduction des risques de pannes, la réduction de l'inconfort des patients dû à la présence des fils et l'amélioration de la mobilité des patients. Les soins de santé à domicile sont d'une importance capitale pour accroître l'autonomie et la qualité de vie des personnes âgées et des patients atteints de maladies chroniques telles que l'asthme, le diabète, l'obstruction pulmonaire, l'insuffisance cardiaque, etc. Des nœuds de capteur portés par ces personnes et déployés dans leur environnement de vie, permettent de collecter en temps réel les informations sur leurs états de santé.

Ces données collectées sont transmises via des technologies de communications (GSM, Internet, ...) à la famille et au personnel médical, permettant ainsi une surveillance à distance des patients et des personnes âgées. Les études pour la médecine et la santé publique ont été révolutionnées par l'usage des réseaux de capteurs. En effet, des capteurs sans fil portés sur le corps et connectés à Internet permettent de collecter continuellement et sur le long terme des données provenant de différents sujets d'étude dans leur quotidien. Sans les réseaux de capteurs, ce type de recherche ne pourrait être menée car certaines conditions de l'environnement naturel ne peuvent être reproduites en laboratoire et lorsque cela possible, la durée de l'expérience est limitée.

La préservation de l'environnement est l'un des défis majeurs de notre ère. Les réseaux de capteurs pour la surveillance environnementale sont déployés dans le but de contrôler et de protéger notre environnement [45], [46]. Ainsi, il existe des réseaux de capteurs pour des applications de mesure de la qualité de l'air et du taux de pollution en zone urbaine ou rurale, des applications de surveillance de glaciers pour étudier les effets du réchauffement climatique, des applications de surveillance de volcans pour détecter à l'avance des éruptions, des applications de surveillance de cours d'eau pour prévenir les inondations, des applications de surveillance d'érosions côtières pour étudier la montée du niveau des eaux, des applications de détection des feux de forêts, etc. Certaines applications des réseaux de capteurs permettent de surveiller le comportement de la faune dans l'habitat naturel de certaines espèces. D'autres applications des réseaux de capteurs sont utilisées dans le domaine de l'élevage pour la surveillance du bétail. Cette surveillance permet de contrôler l'état de santé du bétail, de détecter les maladies, de localiser les animaux et aussi de contrôler la qualité du produit final (viande, lait, ...).

L'agriculture intelligente ou agriculture de précision est une application des réseaux de capteurs qui permet de surveiller le sol, l'eau, les cultures et le climat dans un champ agricole afin d'ajuster au mieux les quantités de semences, d'eau, de fertilisants, de produits phytosanitaires et autres ressources nécessaires à la croissance des cultures [47], [48], [49], [50], [51]. Dans ce type d'application, les nœuds de capteurs sont incorporés dans la terre ou déposés à la surface du sol afin de mesurer les paramètres tels que l'humidité du sol, l'humidité des feuilles, le niveau d'ensoleillement, la température, etc. Les données transmises par les nœuds de capteurs à la station de base aident les agriculteurs à optimiser les quantités d'eau et de nutriments pour chaque parcelle du champ. Ainsi, l'agriculture intelligente a pour principaux objectifs: l'optimisation du rendement agricole, la lutte contre l'insuffisance alimentaire et la réduction des effets néfastes de l'agriculture sur l'environnement.

Les réseaux de capteurs ont été intégrés au transport routier pour donner naissance aux transports intelligents [52], [53], [54]. Les applications des transports intelligents peuvent être regroupées en deux classes : la surveillance du trafic et les systèmes de sécurité temps réel. Dans les systèmes de surveillance du trafic, des nœuds de capteurs sans fil sont intégrés aux routes et aux intersections afin de collecter des données sur le trafic. Par exemple, ils peuvent compter les véhicules dans les files d'attente pour ajuster les feux de circulation ou adapter le nombre de cabines de péage et de voies ouvertes [55],[56]. Dans les systèmes de sécurité, des réseaux de capteurs sont utilisés pour faire face à des situations telles que le freinage d'urgence, l'évitement des collisions et les mises en garde de conditions de conduite dangereuses (glace sur la route, croisement d'animaux, ...). Le transport intelligent inclut aussi les applications embarquées de guidage routier pour éviter les

embouteillages aux heures de pointe et d'assistance à la recherche d'une place de stationnement [57], [58].

Le domaine militaire est le premier à utiliser les réseaux de capteurs grâce notamment à divers projets lancés dès 1980 par la DARPA (Defense Advanced Research Projects Agency) aux Etats Unis d'Amérique. Parmi les applications des réseaux de capteurs dans le domaine militaire, nous pouvons citer la détection d'attaque nucléaire, chimique et biologique, la surveillance de zone de bataille, la localisation de cibles mobiles (soldats, secouristes, voitures ...), la détection du franchissement illégal des frontières, la détection de champs de mines, l'inspection d'un bâtiment par des robots dotés de nœuds de capteurs, etc [11], [10]. Dans ce type d'applications, la surveillance peut être passive ou active. Dans la surveillance active, les capteurs sont déployés sur une zone afin de mesurer les paramètres importants (produits chimiques, biologiques ou radioactifs, détections de mouvement ou de présence) et les transmettre vers le centre de commandement des opérations. Dans la surveillance active, les nœuds de capteurs sont portés par les agents ou les véhicules et transmettent les données concernant le statut du porteur et les paramètres de l'environnement; ce qui permet le suivi, la localisation et une meilleure gestion des troupes et du matériel sur le terrain.

Dans le domaine de l'industrie, un système d'acquisition et de contrôle de données ou SCADA (*Supervisory Control and Data Acquisition*) est un système de gestion à distance et à grande échelle permettant de traiter en temps réel un grand nombre de mesures et de contrôler à distance des installations techniques [59], [60], [61], [62]. Les systèmes SCADA sont utilisés entre autres dans la surveillance de processus industriels, la surveillance des réseaux d'approvisionnement en eau, la surveillance de commande de la production d'énergie électrique, la surveillance de canalisations (pipelines) de gaz, d'eau et pétrole [63], [64]. Dans les systèmes SCADA traditionnels, la plupart des capteurs sont connectés par des fils à une machine ou au réseau d'usine. L'intégration des réseaux de capteurs sans fil aux systèmes SCADA apporte de nombreux avantages notamment la réduction des coûts de maintenance et de déploiement, l'augmentation de la flexibilité et du passage à l'échelle et l'augmentation de la performance. Par exemple, dans la surveillance de pipeline souterrain ou sous-marin (pétrole, gaz ou eau), les nœuds de capteurs sont déployés à l'intérieur et autour du pipeline pour détecter les fuites, les explosions et autres anomalies. Les informations de localisation et de quantification de l'anomalie sont alors envoyées en temps réel à la station de base et peuvent être utilisées pour résoudre le problème de façon rapide et optimale.

La domotique vise à apporter des solutions techniques pour répondre aux besoins de confort (gestion d'énergie, optimisation de l'éclairage et chauffage), de sécurité (alarme) et de communication (commandes à distance, signaux visuels ou sonores, etc) que l'on peut retrouver dans les maisons, les hôtels, les lieux publics, etc. Ce domaine qui existe depuis les années 1970, a été révolutionné par l'intégration des réseaux de capteurs sans fil. Ainsi, il existe de nos jours de nombreuses applications domotiques à moindre coût permettant d'automatiser des tâches du quotidien, de réduire la consommation énergétique et d'assurer la sécurité de l'habitation [65], [66]. De plus, les réseaux de capteurs permettent non seulement de rendre une maison intelligente, mais aussi de la rendre connectée grâce à l'Internet des Objets (IoT). Des nœuds de capteurs de différents types (lumière, présence, température, image) et des actionneurs sont déployés dans la maison et connectés à Internet via la station de base. Ce réseau de capteurs envoie des informations sur

différents paramètres de la maison (chauffage, éclairage, présence d'un intrus, ...) au propriétaire via son smartphone et le propriétaire peut, où qu'il se trouve, actionner des commandes de la maison (réduire le chauffage, verrouiller une porte, ...) à partir de son smartphone. Dans le domaine public, les réseaux de capteurs sans fil permettent aussi la construction des bâtiments et infrastructures intelligents [67], [68], [69]. Les bâtiments gaspillent de grandes quantités d'énergie par un usage inefficace du chauffage, de la ventilation et de la climatisation. Une meilleure surveillance en temps réel et à haute résolution de la température, du débit d'air, de l'humidité et d'autres paramètres physiques dans un bâtiment par le biais d'un réseau de capteurs sans fil peut considérablement augmenter le niveau de confort des usagers et réduire la consommation d'énergie. De plus, les nœuds de capteurs peuvent être utilisés pour surveiller le niveau des paramètres mécaniques des bâtiments ou des ponts dans les zones sismiques. En mesurant les paramètres mécaniques comme la charge de flexion de poutres, il est possible de déterminer rapidement s'il est possible d'entrer dans un bâtiment donné après un tremblement de terre ou si le bâtiment est au bord de l'effondrement, ce qui constitue un avantage considérable pour le personnel de sauvetage. Dans ce type d'applications, les nœuds de capteurs sont incorporés dans le bâtiment ou le pont au moment de sa construction.

Chapitre 3 Etat de l'art du problème de couverture de cibles par les réseaux de capteurs sans fil

3.1 Introduction

Le problème de couverture reflète la qualité de service, en termes de surveillance, fournie par un réseau. Par exemple, dans une application de surveillance de forêt, une question liée au problème de couverture est de savoir quelles sont les chances qu'un feu déclenché à un endroit donné de la forêt soit détecté. Le problème de couverture peut être défini de différentes façons du fait de la diversité des applications des réseaux de capteurs. On distingue trois classes principales de problèmes de couverture en fonction de l'entité à couvrir: la couverture de surface ou de région, la couverture de cibles ou de points et la couverture de barrière ou de chemin [70], [19], [18], [71]. Dans la couverture de surface, l'objectif est de surveiller chaque point d'une surface donnée, en d'autres termes, chaque point de la surface d'intérêt doit être dans le rayon de sensibilité d'au moins un capteur. Dans la couverture de cibles, l'objectif est de surveiller un ensemble de points donnés, chaque cible doit donc être dans le rayon de sensibilité d'au moins un capteur. La couverture de barrière a pour but de déterminer si une région est couverte de manière suffisante pour s'assurer qu'un intrus qui passe à travers la région ne peut pas le faire sans être détecté.

3.1.1 La couverture de surface

Dans le problème de couverture de surface, l'objectif est de couvrir l'ensemble des points d'une surface donnée. Selon les exigences de l'application, la couverture peut être complète ou partielle. Si le nombre de capteurs est insuffisant pour assurer une couverture complète, l'objectif est de maximiser le taux de couverture de la surface.

Des applications militaires telles que la surveillance des champs de bataille nécessitent une couverture complète de surface. Dans ces applications, chaque point de la surface est couvert par au moins un nœud de capteur (couverture simple ou 1-couverture) ou par au moins k capteurs avec $k > 1$, (couverture multiple ou k -couverture). La couverture multiple est une extension de la couverture simple. Elle est requise dans les applications militaires de surveillance de zones de haute sécurité et des champs de bataille et plus généralement pour des applications qui exigent un niveau élevé de robustesse ou tolérance aux pannes. En effet, étant donné que la défaillance d'un seul nœud peut entraîner la perte ou la corruption de données importantes, un seul niveau de couverture n'est pas suffisant pour ces applications. Un défi majeur de la couverture complète de surface est de minimiser le nombre de nœuds de capteurs à déployer pour couvrir toute la surface.

Dans certaines applications, la couverture complète d'une zone donnée n'est pas nécessaire, auquel cas une couverture partielle assurant un degré de couverture donné est suffisante et acceptable. Une couverture partielle peut être définie comme l'ensemble des nœuds de capteurs qui couvrent au moins un pourcentage θ de la surface totale ; elle est alors appelée θ -couverture. Les applications de mesure de la température sont un exemple d'applications donnant lieu à une couverture partielle car il suffit de mesurer la température de 80% de la région pour connaître la température de toute la région.

3.1.2 La couverture de cibles

Dans de nombreuses applications, la surveillance de toute la zone n'est pas utile, seuls certains points particuliers ou cibles doivent être surveillés. En fonction de la criticité de l'application, chaque cible est couverte par au moins un nœud de capteur (couverture simple) ou par au moins k capteurs avec $k > 1$ (couverture multiple). Les applications de surveillance de points critiques dans une zone industrielle ou de points sensibles d'un champ agricole sont des exemples d'applications où la couverture de cibles est requise. De plus, la couverture de cibles est une autre forme de couverture partielle de surface lorsque l'ensemble des cibles correspond à un ensemble de points qui, lorsqu'ils sont couverts, permettent d'atteindre le taux de couverture de surface requis. C'est par exemple le cas pour les applications de surveillance de température, d'humidité, de détection de fumée ou de produits chimiques.

3.1.3 La couverture de barrière

Dans plusieurs applications, les capteurs ne sont pas déployés pour surveiller les événements à l'intérieur de la zone considérée mais plutôt pour détecter les intrus qui tentent de pénétrer dans cette zone. C'est par exemple le cas des applications dont l'objectif est de détecter les intrus lorsqu'ils traversent une frontière ou qu'ils pénètrent une zone protégée. Les exemples de telles applications sont le déploiement des nœuds de capteurs le long des frontières internationales pour détecter des intrusions illégales, autour des forêts pour détecter la propagation des feux de forêt, ou autour d'usines chimiques pour détecter la propagation de produits chimiques. On distingue deux types de couverture de barrière, la couverture complète et la couverture partielle. La couverture de barrière est dite complète s'il est impossible de traverser la barrière de capteurs sans être détecté par au moins un capteur. La couverture est partielle si la probabilité qu'un intrus traverse la barrière sans être détecté est inférieure à un seuil donné.

Dans les applications des réseaux de capteurs sans fil qui utilisent des technologies radio courte portée, le problème de couverture est soumis à une contrainte de connexité du réseau. Les données collectées par les nœuds de capteurs doivent alors être transmises en multisauf à la station de base. Cela implique que le déploiement des nœuds de capteurs doit non seulement respecter les exigences de couverture (surface, cible, barrière), mais aussi que le réseau déployé doit être connexe pour permettre la transmission des données. Dans le graphe connexe formé par le réseau, un arc existe entre deux sommets si la distance entre les deux nœuds de capteurs représentant ces sommets est inférieure à la portée de communication du module radio. Lorsque la contrainte de connexité est associée au problème de couverture, on parle alors de couverture connexe.

La durée de vie du réseau est une problématique majeure dans la plupart des applications des réseaux de capteurs sans fil [13]. Dans les applications où le problème de couverture se pose, la définition de la durée de vie du réseau dépend de celle du problème de couverture. De manière générale, la durée de vie du réseau est la durée pendant laquelle le réseau garantit la qualité de couverture requise par l'application. Ainsi, par exemple, pour la k -couverture de cibles, la durée de vie du réseau est la période pendant laquelle chaque cible est couverte par au moins k capteurs. Compte tenu de la contrainte énergétique des nœuds de capteurs, l'objectif principal est de maximiser la durée de vie du réseau. Dans un premier temps, les nœuds de capteurs sont déployés de façon redondante, c'est-à-dire que le nombre de nœuds déployés est supérieur à celui nécessaire pour couvrir la zone d'intérêt (surface, cibles, barrière). Ensuite, le contrôle de topologie est utilisée pour économiser l'énergie consommée par le réseau. Une technique de contrôle de topologie consiste à activer un sous-ensemble de nœuds capables de garantir la qualité de couverture requise et de mettre les autres nœuds en veille [15], [17]. Le sous-ensemble de nœuds actifs est appelé ensemble couvrant. Dans le cas de la couverture connexe, chaque ensemble couvrant doit former un graphe connexe. En formant plusieurs ensembles couvrants et en les activant successivement, la durée de vie du réseau est prolongée et égale à la somme des durées d'activation de tous les couvrants. Le problème de former une famille d'ensembles couvrants et de leur assigner des durées d'activation de sorte que la durée de vie du réseau soit maximale est appelé Maximum Lifetime Coverage Problem (MLCP) [18]. Ce problème est NP-difficile [20].

3.2 Problème de couverture (MLCP)

Soit $S = \{s_1, \dots, s_n\}$ un ensemble de capteurs et $T = \{t_1, \dots, t_m\}$ un ensemble de cibles. Chaque capteur s_i a une batterie de durée de vie b_i . Le capteur s_i couvre la cible t_j si la distance entre t_j et s_i est inférieure au rayon de sensibilité de s_i . Un ensemble couvrant C est un ensemble de capteurs qui couvrent conjointement toutes les cibles. La durée d'activation Lt d'un ensemble couvrant C est la durée pendant laquelle les capteurs appartenant à l'ensemble C sont en mode actif tandis que les autres capteurs sont en mode veille.

Le MLCP consiste à former une famille d'ensembles couvrants $C_j, j = 1 \dots p$ et déterminer leurs durées d'activation respectives $Lt_j, j = 1 \dots p$, de sorte à maximiser la durée de vie du réseau $Lt_1 + \dots + Lt_p$ sachant que si le capteur s_i appartient aux ensembles couvrants $C_{i_1}, \dots, C_{i_q}$ alors $Lt_{i_1} + \dots + Lt_{i_q}$ doit être inférieur à la durée de vie b_i de sa batterie [18].

Il est à noter que les couvrants ainsi formés sont activés les uns après les autres pendant leurs durées d'activation respectives. La figure 8 montre un réseau constitué de 4 capteurs s_1, s_2, s_3, s_4 et trois cibles (les triangles) t_1, t_2, t_3 . Supposons que $b_1 = b_2 = b_3 = b_4 = 1$. Une solution pour cette instance MLCP est de former deux ensembles couvrants $C_1 = \{s_1, s_3\}$ et $C_2 = \{s_2, s_4\}$ de durées d'activation respectives $Lt_1 = 1$ et $Lt_2 = 1$. Pendant une unité de temps, les capteurs s_1, s_3 (de l'ensemble couvrant C_1) sont en mode actif pendant que les autres sont veille. Pendant la seconde unité de temps, les capteurs s_2, s_4 (de l'ensemble couvrant C_2) sont mis en mode actif tandis que les autres sont placés en mode veille. Dans le cas de cet exemple, aucun capteur n'est en veille pendant ce temps car les capteurs s_1, s_3 sont à court de batterie. En supposant que l'énergie consommée par un capteur en veille est négligeable, cette stratégie permet d'obtenir une durée de vie du réseau

$Lt_1 + Lt_2$ égale à 2, ce qui représente le double de la durée de vie du réseau si tous les capteurs étaient actifs en même temps.

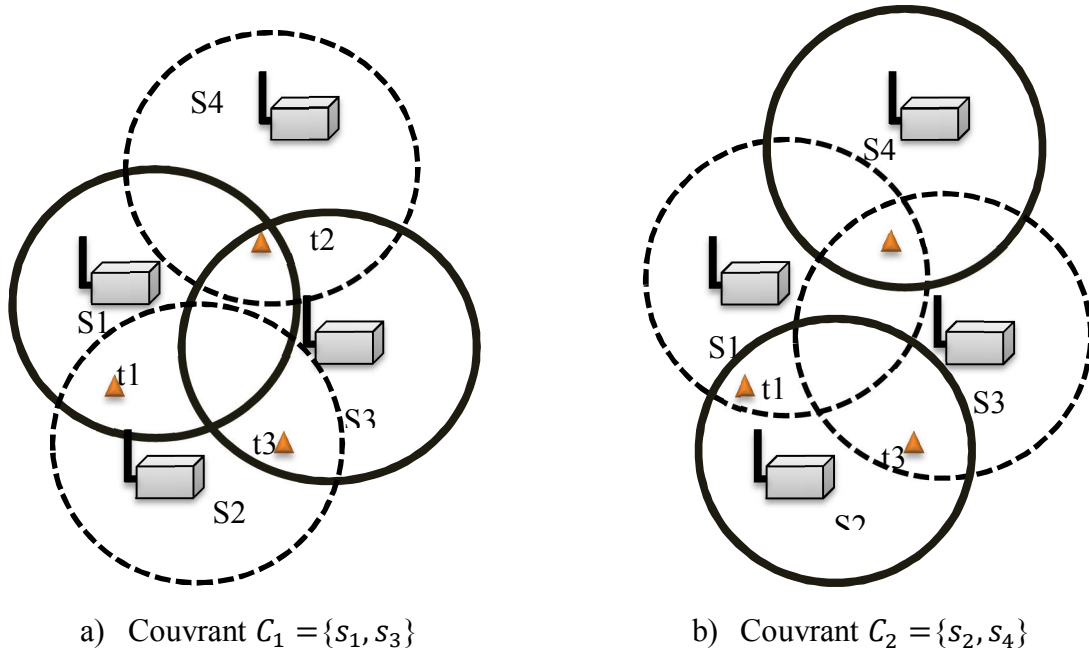
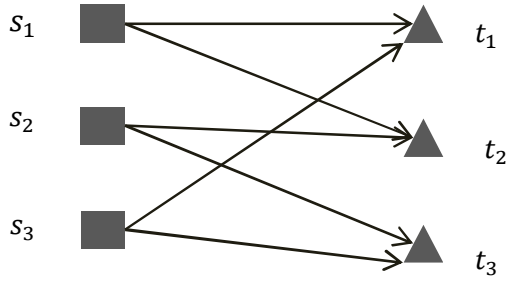


Figure 8 - Solution d'une instance du MLCP

Le MLCP étant NP-difficile [20], il existe plusieurs algorithmes approchés dans la littérature. Ces heuristiques peuvent être regroupées en deux catégories en fonction de l'approche utilisée. Dans la première approche, on forme un nombre maximal de couvrants disjoints. Le problème de former le nombre maximal de couvrants disjoints est aussi un problème NP-difficile [22]. Une sous-classe de la catégorie de l'approche par couvrants disjoints cherche à atteindre le nombre maximal de couvrants disjoints en formant une suite de couvrants minimaux. Le problème de former un couvrant de taille minimal est aussi un problème NP-difficile [21]. Dans la seconde approche, on cherche à former le nombre maximal de couvrants non disjoints tel que chaque capteur appartienne à au plus w couvrants. Ce problème est aussi NP-difficile [20]. Dans chacune des deux approches, plusieurs méthodes ont été proposées notamment la programmation linéaire [22], la méthode gloutonne [25], [20], [26] et les algorithmes génétiques [24], [23].

L'exemple très simple de la figure 9 ci-dessous montre que l'approche par couvrants non disjoints peut donner de meilleurs résultats que l'approche par couvrants disjoints [18]. Dans cet exemple, le réseau est formé de trois capteurs s_1, s_2, s_3 et de trois cibles t_1, t_2, t_3 tel que le capteur s_1 couvre les cibles t_1, t_2 , le capteur s_2 couvre les cibles t_2, t_3 et le capteur s_3 couvre les cibles t_1, t_3 . Les capteurs ont tous la même durée de vie $b = 2$. En organisant les capteurs en couvrants disjoints, nous obtenons un seul couvrant $C_1 = \{s_1, s_2\}$, ou $C_2 = \{s_1, s_3\}$, ou $C_3 = \{s_2, s_3\}$ avec une durée de vie du réseau égale à 2. Tandis qu'en formant des couvrants non disjoints avec $w = 2$, nous obtenons trois couvrants $C_1 = \{s_1, s_2\}$, $C_2 = \{s_1, s_3\}$, et $C_3 = \{s_2, s_3\}$ de durée d'activation égale 1 chacun, ce qui donne une durée de vie du réseau égale à 3. Dans la suite, nous présentons les heuristiques pour le MLCP par catégorie.



Ensemble de capteurs $S = \{s_1, s_2, s_3\}$; Ensemble de cibles $T = \{t_1, t_2, t_3\}$

Relation de couverture : $s_1 = \{t_1, t_2\}$; $s_2 = \{t_1, t_3\}$; $s_3 = \{t_2, t_3\}$

Figure 9 - Une instance difficile de MLCP

3.3 Heuristiques de l'approche par couvrants disjoints

3.3.1 Heuristique basée sur les couvrants minimaux

Dans cette heuristique, le problème de former le nombre maximal de couvrants disjoints est approché par le problème de former une succession de couvrants minimaux. Le problème de couvrant minimal étant NP-difficile, plusieurs algorithmes approchés ont été proposés dans la littérature. Parmi ces algorithmes, l'un des plus connus est l'algorithme glouton classique [21]. L'algorithme glouton classique du couvrant minimal consiste à sélectionner itérativement le capteur qui couvre le plus de cibles non couvertes par les capteurs sélectionnés précédemment. Cet algorithme garantit de former un couvrant dont la taille est au plus $\log(m)$ fois celle du couvrant minimal, où m est le nombre de cibles. Cependant, former une succession de couvrants minimaux ne permet pas toujours de former le nombre maximal de couvrants disjoints comme le montre l'exemple 2 de la figure 10. L'exemple 2 est un réseau composé de 4 capteurs s_1, s_2, s_3, s_4 et de 3 cibles t_1, t_2, t_3 . Le capteur s_1 couvre les cibles t_1, t_2 , le capteur s_2 couvre les cibles t_2, t_3 , le capteur s_3 couvre les cibles t_1, t_3 et le capteur s_4 couvre la cible t_3 . Tous les capteurs ont la même durée de vie $b = 1$.

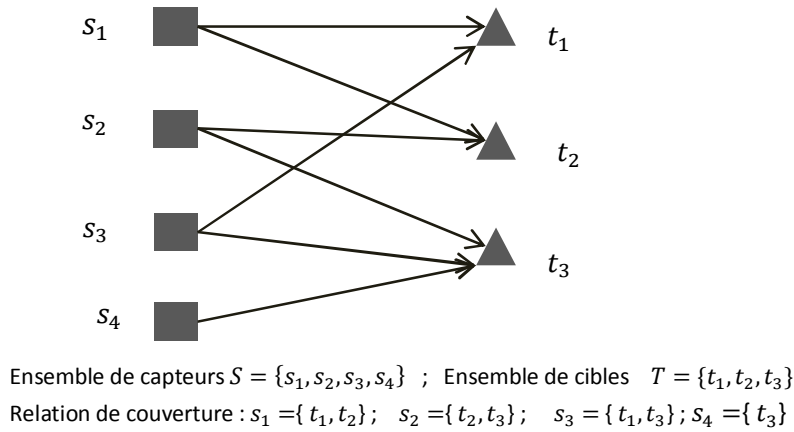


Figure 10 - Une instance de MLCP

Lors de la première itération, l'algorithme glouton classique du couvrant minimal peut former le couvrant $c_1 = \{s_1, s_2\}$ (car il est minimal). Les couvrants devant être disjoints, il n'y aura pas d'autres itérations car les capteurs résiduels à savoir s_3 et s_4 ne couvrent pas toutes les cibles. On aura donc une solution avec un seul couvrant alors qu'il est possible de former deux couvrants disjoints à savoir $C'_1 = \{s_1, s_3\}$ et $C'_2 = \{s_2, s_4\}$.

3.3.2 L'algorithme MCMCC (Most Constrained - Minimally Constraining Coverage)

L'algorithme MCMCC (Most Constrained - Minimally Constraining Coverage) proposé par S. Slijepcevic et M. Potkonjak [25] est basé sur la notion de cible critique. Une cible critique est dite critique si elle est couverte par le plus petit nombre de capteurs. Le nombre de capteurs qui couvrent une cible critique est une borne supérieure du nombre maximal de couvrants disjoints. En effet, si la cible critique est couverte par k capteurs, alors il est impossible de former plus de k couvrants disjoints. L'idée de base de cet algorithme est de minimiser la couverture des cibles au sein de chaque couvrant. Si dans un couvrant, il existe p capteurs qui couvrent la cible critique, il est impossible de former plus de $k - p + 1$ couvrants disjoints. A chaque round de l'algorithme, un couvrant est formé. Un round est constitué de plusieurs itérations, et à chaque itération, un capteur est sélectionné jusqu'à ce que toutes les cibles soient couvertes par le couvrant en construction. Au début de chaque round, l'ensemble V des capteurs *disponibles* est initialisé à l'ensemble des capteurs n'appartenant à aucun couvrant formé lors des rounds précédents. A chaque itération du round, l'ensemble E_{min} des cibles critiques est initialisé (cet ensemble change à chaque itération). L'une de ces cibles critiques, e_{min} , est sélectionnée. Pour chaque capteur qui couvre la cible critique e_{min} , la valeur d'une fonction objectif est calculée. Une valeur élevée de la fonction objectif pour un capteur indique qu'il est peu probable que des cibles critiques soient couvertes de façon redondante si ce capteur est choisi.

La fonction objectif utilise deux paramètres M , N et a pour but de :

- 1) favoriser les capteurs qui couvrent un grand nombre de cibles non couvertes par le couvrant en ajoutant la valeur M pour chaque cible non couverte,

- 2) favoriser les capteurs qui couvrent les cibles faiblement couvertes en soustrayant le nombre de capteurs de V qui couvrent les mêmes cibles non couvertes par le couvrant (ce nombre est petit pour les cibles peu couvertes),
- 3) favoriser les capteurs qui ne couvrent pas les cibles de façon redondante dans le couvrant en soustrayant N pour chaque cible déjà couverte, et
- 4) favoriser les capteurs qui couvrent de façon redondante les cibles fortement couvertes en ajoutant le nombre de capteurs qui peuvent couvrir ces cibles dans les prochains couvrants.

Le capteur qui couvre e_{min} et qui a la plus grande valeur de la fonction objectif est choisi dans le couvrant. Ce capteur, de même que tous les autres capteurs de l'ensemble V qui couvrent la cible critique e_{min} sont considérés comme *non disponibles* pour les itérations futures de ce round et sont donc supprimés de l'ensemble V . L'algorithme s'arrête lorsque les capteurs résiduels (n'appartenant à aucun couvrant) ne couvrent plus toutes les cibles.

La complexité de cette heuristique est $O(n^2)$, où n est le nombre de capteurs. Un inconvénient de cet algorithme est la difficulté de détermination des bonnes valeurs des paramètres M et N , alors qu'ils jouent un rôle important dans la formation des couvrants car ils influencent grandement le choix des capteurs. Les auteurs ne donnent pas d'indication sur les valeurs adéquates de ces paramètres en fonction de l'instance du problème. Par ailleurs, à une itération donnée, plusieurs capteurs peuvent avoir la même valeur de la fonction objectif, et la liberté de choix de l'un d'entre eux peut aboutir à une solution de mauvaise qualité.

3.3.3 L'algorithme MC-MIP (Maximum Covers using Mixed Integer Programming)

Le principe de base de l'algorithme MC-MIP (Maximum Covers using Mixed Integer Programming) [22] est de transformer le problème de trouver le nombre maximal de couvrants disjoints un problème de flot maximal (MFP : Maximum Flow Problem), de formuler le problème de flot maximal comme un problème de programmation mixte en nombres entiers (MIP : Mixed Integer Programming) que l'on résout par la méthode branch-and-bound).

Le problème classique de flot maximal est défini de la manière suivante. Soit un graphe connexe et orienté $G = (X, U)$ tel que chaque arc $u \in U$ est muni d'une capacité inférieure $b_u \in \mathbb{R}$ et d'une capacité supérieure $c_u \in \mathbb{R}$ ($b_u \leq c_u$), avec deux sommets particuliers, une source $s \in X$ et un puits $p \in X$ ayant un arc $u_0 = (p, s) \in U$ appelé arc de retour. Le problème de flot maximal consiste à attribuer à chaque arc $u \in U$ un nombre réel ou flux $\varphi(u)$ tel que $b_u \leq \varphi(u) \leq c_u$ et pour tout sommet $s \in X$, la somme des flux des arcs arrivant à s soit égale à la somme des flux des arcs partant de s de sorte à maximiser le flux de l'arc de retour.

La transformation du problème du nombre maximal de couvrants disjoints en un problème de flot maximal se fait ainsi qu'il suit.

Etape 1 : On crée un graphe orienté bipartite $G = (V, E)$ dont V est l'ensemble de tous les capteurs et de toutes les cibles et l'arc $(s, t) \in E$ si le capteur s couvre la cible t . La capacité de l'arc (s, t) vaut 1. Puis on ajoute au graphe G un sommet X et un arc (t, X) de capacité 1 pour toute cible t .

Etape 2 : On identifie une cible critique (la cible la moins couverte). Soit k le nombre de capteurs qui couvrent la cible critique. On crée k exemplaires du graphe à savoir $G_1, G_2, \dots, G_k$. Pour chaque sous-graphe G_j , tout sommet s_i du graphe G est noté s_{ji} afin de permettre une distinction entre les différents exemplaires de s_i .

Etape 3 : on crée un sommet source S . Puis pour chaque capteur s_i , on crée un sommet S_{0i} , un arc (S, S_{0i}) et un arc (S_{0i}, s_{ji}) pour tout sous-graphe G_j . La capacité des arcs (S, S_{0i}) et (S_{0i}, s_{ji}) est égale au degré du sommet s_i dans le graphe G .

Etape 4 : on crée deux sommets puits Y_1 et Y_2 . Pour tout sommet X_j (du sous graphe G_j), on ajoute un arc (X_j, Y_2) de capacité m (où m est le nombre de cibles). Pour tout sommet t_{ji} (de la cible t_i dans le sous graphe G_j), on ajoute un arc (t_{ji}, Y_1) de capacité n (où n est le nombre de capteurs).

La capacité affectée à chaque arc u correspond à sa capacité supérieure c_u et sa capacité inférieure b_u est égale à 0. Le flux $\varphi(u)$ de tout arc u est soumis à la contrainte $b_u \leq \varphi(u) \leq c_u$ excepté pour les arcs qui arrivent sur le puits Y_1 . Pour tout sommet v excepté la source S et les puits (Y_1 et Y_2), la somme des flux des arcs arrivant sur s doit être égale à la somme des flux des arcs partant de s . L'objectif de ce problème de flot maximal est de maximiser la somme des flux des arcs qui arrivent sur le puits Y_2 .

Soit α le flot sur le sommet Y_2 et m le nombre de cibles, l'algorithme MC-MIP construit $\frac{\alpha}{m}$ couvrants disjoints de la manière suivante. Pour chaque sous graphe G_j , si le flux de l'arc (X_j, Y_2) est non nul alors un couvrant est formé en sélectionnant tous les capteurs s_i tels que le flux de l'arc (S_{0i}, s_{ji}) est différent de 0. Un inconvénient de cette heuristique est sa complexité qui est dominée par celle de la résolution de la formulation MIP qui peut être exponentielle dans le pire cas. De plus la transformation du problème du nombre maximal de couvrants disjoints aboutit à un problème de flot maximal de taille beaucoup plus grande.

3.3.4 Heuristique basée sur la méthode Simplexe

L'idée de base de cette heuristique est de former successivement des solutions au problème du nombre maximal de couvrants disjoints de sorte que chaque solution contienne plus ou autant de couvrants disjoints que la solution précédente [72].

A chaque itération, l'heuristique construit un ensemble de couvrants disjoints. A la première itération, les capteurs sont classés dans un ordre initial. Pour former un couvrant, l'algorithme parcourt séquentiellement (sans retour en arrière) la liste des capteurs dans l'ordre et sélectionne le premier capteur qui couvre au moins une cible non couverte par le couvrant. Arrivé en fin de liste, si les capteurs sélectionnés ne couvrent pas toutes les cibles l'itération s'arrête sinon l'algorithme remet à leur place dans la liste résiduelle tous les capteurs redondants du couvrant formé. Un capteur est redondant dans un couvrant si toutes les cibles qu'il couvre sont couvertes par d'autres capteurs du couvrant. L'algorithme répète les parcours de la liste jusqu'à ce que les capteurs résiduels de la liste ne couvrent plus toutes les cibles. Le nombre de couvrants disjoints de la solution correspond au nombre de parcours fructueux réalisés.

A l'itération suivante, les capteurs sont réordonnés de la manière suivante : tous les capteurs résiduels de l'itération précédente sont placés en tête de liste suivis des capteurs sélectionnés lors de

l'itération précédente dans leur ordre de sélection. Puis on recommence le processus de parcours pour former les couvrants de la nouvelle solution. Les solutions sont produites ainsi successivement et l'algorithme s'arrête sous l'une des trois conditions.

- Il y a autant de couvrants que de capteurs qui couvrent la cible critique
- La solution obtenue est identique à la précédente
- Les k dernières itérations produisent le même nombre de couvrants disjoints

3.3.5 L'algorithme GAMDSC (Genetic Algorithm for Maximum Disjoint Set Covers)

L'algorithme GAMDSC (Genetic Algorithm for Maximum Disjoint Set Covers) est un algorithme génétique [23]. Un algorithme génétique (AG) est un algorithme stochastique d'optimisation basé sur les mécanismes de la sélection naturelle et de la génétique. Un AG recherche le ou les extrema d'une fonction définie sur un espace de données. Pour résoudre un problème d'optimisation donné, un ensemble de solutions est créé aléatoirement. Cet ensemble de solutions est appelé population initiale et chaque solution est un individu représenté sous la forme d'un chromosome. Chaque individu est évalué et classifié selon sa ressemblance avec la solution optimale grâce à une fonction fitness. Les meilleurs individus ont une meilleure chance d'être sélectionnés pour la reproduction. Une nouvelle population ou génération est créée suite à des opérations entre les individus (opérations de croisement, de mutation, etc.). Certains individus de la nouvelle génération sont meilleurs (plus proches de la solution optimale) que ceux de la génération précédente. Plusieurs générations sont formées successivement jusqu'à ce que la condition d'arrêt soit atteinte. La condition d'arrêt correspond à la formation du nombre maximal de générations fixé comme paramètre de l'algorithme. La solution retournée par l'algorithme est le meilleur individu de la dernière génération. Proposer un algorithme génétique consiste à définir une méthode de codage d'un individu, un mécanisme de création de la population initiale, une fonction fitness d'évaluation des individus, des opérateurs génétiques (croisement, mutation, ...), et des paramètres de dimensionnement tel que la taille de la population, le nombre maximal de générations, etc.

Dans l'algorithme GAMDSC, chaque individu est codé par un chromosome qui est un tableau de n nombres entiers (où n est le nombre de capteurs). Notons ub le nombre de capteurs qui couvrent la cible critique (rappelons que ub est une borne supérieure du nombre maximal de couvrants disjoints). Chaque élément du tableau est un gène du chromosome et contient une valeur entière entre 1 et ub . La valeur du gène i est l'index du couvrant auquel appartient le capteur s_i . La figure 11 montre le codage de l'individu correspondant à la solution optimale de l'exemple 2.

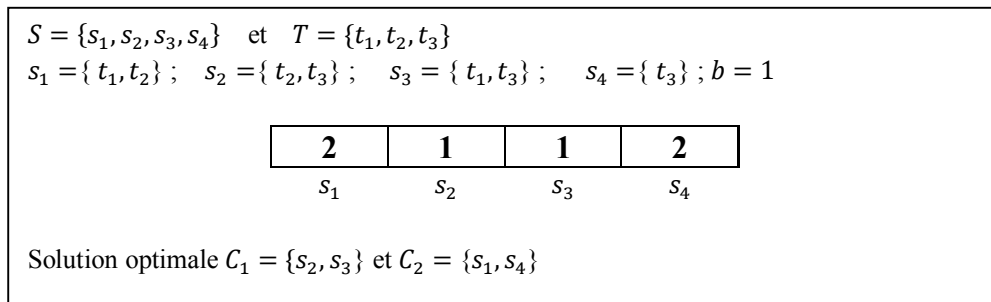


Figure 11 - Représentation d'une solution de MLCP sous forme de chromosome

L'algorithme GAMDSC définit la fonction fitness d'un individu comme le nombre de couvrants disjoints qu'il comporte. Si l'ensemble C_j des capteurs dont le gène porte la valeur j ne couvre pas toutes les cibles, alors C_j n'est pas compté dans la fonction fitness car il n'est pas un couvrant. L'opérateur de croisement adopté par l'algorithme GAMDSC est le croisement uniforme. Cet opérateur crée un individu fils en choisissant pour chaque gène, la valeur du gène correspondant d'un de ses parents avec la probabilité un demi. L'opérateur de mutation remplace chaque gène d'un fils par une valeur inférieure à ub avec la probabilité $1/n$. Enfin, cet algorithme définit un opérateur de dispersion dont le but est de s'assurer que tous les capteurs qui couvrent la cible critique appartiennent à des couvrants différents. L'opérateur de dispersion est appliqué à un individu fils après l'opérateur de mutation. Il parcourt les gènes de tous les capteurs de la cible critique et si deux capteurs ont le même gène, il remplace l'un d'eux par un indice de groupe non utilisé par les autres capteurs de la cible critique. Un inconvénient majeur de l'algorithme GAMDSC est que l'ensemble des gènes ayant la même valeur au sein d'un individu représente un groupe de capteurs qui n'est pas nécessairement un couvrant. De ce fait, si la taille du problème augmente, il sera difficile d'explorer tout l'espace de solutions. Or, la performance d'un AG est liée à la capacité à explorer tout l'espace de solutions. Par ailleurs, l'opérateur de dispersion est difficilement applicable dans les cas où il existe plusieurs cibles critiques car chacune d'elles a son propre ensemble de capteurs.

3.3.6 L'algorithme Flowing-GA (Flowing Genetic Algorithm)

L'algorithme Flowing-GA (Flowing Genetic Algorithm) [24] est un algorithme génétique. Bien qu'il utilise le même codage de l'individu que l'algorithme GAMDSC, l'algorithme Flowing-GA garantit que tous les capteurs dont les gènes ont la même valeur forment un couvrant à l'exception du dernier groupe (celui dont l'index est le plus élevé). Pour respecter cette contrainte, la valeur 0 est attribuée à tous les gènes de chaque individu de la population initiale. De façon itérative, les capteurs redondants du couvrant d'index i sont choisis aléatoirement pour former le couvrant d'index $i + 1$ (un capteur est redondant dans un couvrant si toutes les cibles qu'il couvre sont couvertes par d'autres capteurs de ce couvrant). La fonction fitness d'un individu est égale à son nombre de couvrants plus son taux de couverture. Le taux de couverture d'un individu est le pourcentage de cibles couvertes par son dernier groupe. Le taux de couverture est inférieur ou égal à 1 car le dernier groupe ne couvre pas nécessairement toutes les cibles. L'algorithme Flowing-GA utilise l'opérateur de croisement uniforme et garde après le croisement le meilleur individu entre les parents et le fils. Si le chromosome du fils ne respecte pas la contrainte spécifiée plus haut (tous les groupes, excepté le dernier, doivent être des couvrants), il est supprimé et seuls les parents sont comparés pour la sélection dans la nouvelle génération. Le taux de mutation des gènes d'un individu est égal au pourcentage d'individus de la population ayant la meilleure fitness. L'algorithme Flowing-GA introduit trois nouveaux opérateurs génétiques dont le but est d'améliorer la fitness des individus. Il s'agit des opérateurs Forward-flowing, Backward-flowing et Critical-flowing. L'opérateur Forward-flowing sélectionne aléatoirement un capteur appartenant à un couvrant. Si ce capteur est redondant alors il est placé dans le dernier groupe (le couvrant incomplet). L'opérateur Backward-flowing sélectionne un capteur au hasard dans chaque individu, si le capteur est redondant dans son couvrant, il est ramené vers un autre couvrant choisi aléatoirement. L'opérateur Critical-flowing déplace un capteur qui couvre une cible critique de son

couvrant vers le dernier groupe lorsque ce couvrant contient un autre capteur qui couvre la même cible critique. Cet opérateur permet de compléter rapidement le dernier couvrant.

Cet algorithme Flowing-GA effectue un meilleur parcours de l'espace de solutions que l'algorithme GAMDSC. De plus, il définit des opérateurs qui permettent d'améliorer la qualité des individus et ainsi de créer des individus qui sont plus proches de la solution optimale. Cependant, un inconvénient de l'algorithme Flowing-GA est que tous les individus de la population initiale sont identiques. En effet, une population initiale constituée d'individus différents permettrait une exploration plus complète et plus rapide de l'espace de solutions. Un autre inconvénient est le rejet systématique de tout nouvel individu qui ne respecte pas la contrainte d'un *bon* individu (tous les groupes, excepté le dernier, doivent être des couvrants). Considérons le cas où tous les individus créés lors d'une génération sont des *mauvais* individus, ils seront tous ignorés et il n'y aura aucune amélioration de qualité de la population. En définissant une stratégie de transformation d'un *mauvais* individu en un *bon* individu, on obtiendrait d'avantage de nouveaux individus et donc une meilleure exploration de l'espace de solutions.

3.4 Heuristiques de l'approche par couvrants non disjoints

3.4.1 L'algorithme Greedy-MSC (Greedy Maximum Set Cover heuristic)

L'algorithme Greedy-MSC [20] est un algorithme glouton qui utilise le concept de cible critique. Il définit un paramètre w qui représente la fraction de durée de vie qu'un capteur passe dans un couvrant. Tous les capteurs sont supposés avoir la même durée de vie $b = 1$. L'algorithme Greedy-MSC cherche donc à former le nombre maximal de couvrants non disjoints tel que chaque capteur appartienne à au plus $\frac{1}{w}$ couvrants. L'algorithme Greedy-MSC construit les couvrants de façon itérative. Chaque couvrant est construit en plusieurs itérations. Au début de la construction d'un couvrant, l'algorithme identifie l'ensemble S des capteurs dont l'énergie résiduelle est non nulle (c.-à-d. les capteurs appartenant à moins de $\frac{1}{w}$ couvrants). A chaque itération de la construction du couvrant, une cible critique est choisie, c'est la cible la moins couverte par les capteurs de S , à la fois en termes de nombre de capteurs et d'énergie résiduelle de ces capteurs. Parmi les capteurs qui couvrent cette cible critique, l'algorithme choisit celui qui a la plus grande contribution. Un capteur a une plus grande contribution s'il couvre le plus grand nombre de cibles non couvertes par le couvrant en construction et s'il a la plus grande énergie résiduelle. Le nouveau capteur est ajouté au couvrant et l'algorithme passe à l'itération suivante. Les itérations de la construction d'un couvrant s'arrêtent lorsque toutes les cibles sont couvertes par ce couvrant. Puis l'algorithme passe à la construction d'un nouveau couvrant. L'algorithme s'arrête lorsque les capteurs résiduels (dont l'énergie est non nulle) ne couvrent plus toutes les cibles. Les couvrants générés par l'algorithme sont activés pendant une durée w chacun. La complexité de cet algorithme est $O(km^2n)$ où k est le nombre de capteurs qui couvrent la cible la moins couverte, m est le nombre de cibles et n est le nombre de capteurs. Un inconvénient de l'algorithme Greedy-MSC est qu'il se focalise sur une seule cible critique lors du choix d'un nouveau capteur à ajouter dans le couvrant et ne prend pas en compte la relation entre les capteurs et les autres cibles critiques. De ce fait, lorsqu'il y a plusieurs cibles critiques, le choix de l'une d'entre elles peut conduire à une solution sous-optimale. Un autre inconvénient de l'algorithme Greedy-MSC est la détermination de la valeur du paramètre w . Les

auteurs ne donnent pas d'indications sur la méthode permettant de trouver la bonne valeur de ce paramètre duquel dépend la performance de l'algorithme.

3.4.2 L'algorithme Static-CCF (Static Critical Control Factor)

L'algorithme Static-CCF [26] est un algorithme glouton qui utilise aussi le concept de cible critique. Cet algorithme propose une stratégie de sélection des capteurs qui cherche à minimiser le nombre de capteurs qui couvrent les cibles critiques au sein d'un couvrant. L'algorithme définit une fonction coût CCF (Critical Control Factor) qui prend en compte trois critères à savoir le statut de couverture du capteur candidat, sa relation avec les cibles critiques et son énergie résiduelle. Le statut de couverture mesure le rapport entre le nombre de cibles nouvellement couvertes et le nombre de celles doublement couvertes par le capteur. La relation ou association entre un capteur et les cibles critiques est décrite par un degré de criticité qui est calculé une seule fois, au début de l'algorithme, et reste constant jusqu'à la fin de l'algorithme. Le facteur de criticité d'un capteur est lié à la criticité des cibles qu'il couvre. Moins une cible est couverte, plus elle est critique. Plus un capteur couvre des cibles critiques, plus il est critique. Chacun capteur peut appartenir à au plus w couvrants, où w est un paramètre de l'algorithme. L'algorithme suppose que chaque capteur a une énergie initiale égale à w (couvrants). Trois paramètres α, β, γ (avec $\alpha + \beta + \gamma = 1$) sont utilisés par la fonction coût pour pondérer chaque critère. Une valeur plus grande d'un paramètre permet d'accorder plus d'importance (plus de poids) au critère correspondant. Les valeurs de la fonction coût sont comprises entre 0 et 1. L'objectif de la fonction coût est de (1) favoriser les capteurs candidats qui couvrent le moins de cibles déjà couvertes par le couvrant en construction, (2) minimiser la probabilité de sélectionner les capteurs qui ont un degré élevé de criticité, et (3) favoriser les capteurs qui ont une grande énergie résiduelle. Les couvrants sont construits successivement. Un couvrant est construit en plusieurs itérations. A chaque itération de la construction d'un couvrant, la valeur de la fonction coût est calculée pour chaque capteur candidat. Un capteur est candidat si son énergie résiduelle est non nulle (c.-à-d. s'il appartient à moins de w couvrants) et s'il couvre au moins une cible non couverte par le couvrant en construction. Le capteur dont la valeur de la fonction coût est maximale est ajouté au couvrant. Les itérations sont répétées ainsi de suite jusqu'à ce que toutes les cibles soient couvertes par le couvrant. L'algorithme s'arrête lorsque les capteurs résiduels ne couvrent plus toutes les cibles. La complexité de cet algorithme est $O(wn^2m)$ où m est le nombre de cibles, n est le nombre de capteurs et w le nombre maximal de couvrants auxquels peut appartenir un capteur.

La performance de l'algorithme Static-CCF dépend du choix des valeurs de ces paramètres $(\alpha, \beta, \gamma, w)$. Trouver les valeurs optimales de ces paramètres pour une instance donnée du problème est un inconvénient de cet algorithme. Un autre inconvénient de cet algorithme est le fait que plusieurs capteurs peuvent avoir la même valeur de la fonction coût lors d'une itération. Dans ce cas, le choix aléatoire d'un de ces capteurs peut conduire à une solution sous-optimale. D'autre part, l'algorithme Static-CCF ne garantit pas l'absence de redondance de capteurs dans les couvrants. Les capteurs redondants dans un couvrant peuvent être utilisés ultérieurement pour former des couvrants supplémentaires.

3.4.3 L'algorithme Dynamic-CCF (Dynamic Critical Control Factor)

L'algorithme Dynamic-CCF [26] est une variante de l'algorithme Static-CCF. Tout comme l'algorithme Static-CCF, l'algorithme Dynamic-CCF définit une fonction coût CCF (Critical Control Factor) qui prend en compte trois critères à savoir le statut de couverture du capteur candidat, sa relation avec les cibles critiques et son énergie résiduelle, chacun étant pondéré par des paramètres α, β, γ tel que $\alpha + \beta + \gamma = 1$. Cependant, l'algorithme Dynamic-CCF décrit différemment la relation entre un capteur et les cibles critiques. Dans cet algorithme, la notion de cible critique est dynamique c'est-à-dire que pour chaque nouveau couvrant, les cibles critiques sont identifiées comme celles couvertes par le plus petit nombre de capteurs dont l'énergie résiduelle est non nulle. Ainsi, une cible qui n'était pas critique lors d'une itération donnée peut le devenir lors d'une autre itération. Le critère H_s qui représente la relation entre un capteur et les cibles critiques (pour le couvrant en construction), est une variable binaire qui indique s'il est prudent de choisir ce capteur dans le couvrant. Un capteur est sans danger s'il ne couvre pas une cible critique déjà couverte par le couvrant en construction. Le critère H_s vaut 1 si le capteur est sans danger et 0 sinon. Ce critère permet à l'algorithme d'éviter de couvrir doublement les cibles critiques dans les couvrants. La structure de l'algorithme Dynamic-CCF est la même que celle de l'algorithme Static-CCF. Au début de la construction d'un couvrant, les cibles critiques sont identifiées. A chaque itération, la valeur de la fonction coût CCF est calculée pour chaque capteur candidat. Un capteur est candidat si son énergie résiduelle est non nulle (c.-à-d. s'il appartient à moins de w couvrants) et s'il couvre au moins une cible non couverte par le couvrant en construction. Tous les capteurs qui couvrent des cibles critiques déjà couvertes sont pénalisés car leur critère H_s (car il est nul). Le capteur ayant la plus grande valeur de la fonction coût est sélectionné et ajouté au couvrant. Les itérations se poursuivent jusqu'à ce que le couvrant en construction couvre toutes les cibles. L'algorithme s'arrête lorsque les capteurs résiduels ne couvrent plus toutes les cibles. La complexité de cet algorithme est $O(wn^2m + nw^2 + wnm)$ où m est le nombre de cibles, n est le nombre de capteurs et w le nombre maximal de couvrants auxquels peut appartenir un capteur.

Tout comme l'algorithme Static-CCF, la performance de l'algorithme Dynamic-CCF dépend du choix des valeurs de ces paramètres $(\alpha, \beta, \gamma, w)$. Trouver les valeurs optimales de ces paramètres pour une instance donnée du problème est un inconvénient de cet algorithme. De même, un autre inconvénient de cet algorithme est le fait que plusieurs capteurs peuvent avoir la même valeur de la fonction coût lors d'une itération. Enfin, l'algorithme Dynamic-CCF ne garantit pas non plus l'absence de redondance de capteurs dans les couvrants.

3.5 Conclusion

Dans ce chapitre, nous avons présenté les trois principales catégories de problèmes de couverture à savoir la couverture de surface, la couverture de cibles et la couverture de chemin ou de barrière. Pour chaque catégorie, nous avons décrit les variantes possibles à savoir la k -couverture, la θ -couverture et la couverture connexe. Nous avons ensuite montré que le problème de couverture dans les réseaux de capteurs est fortement lié à l'optimisation de la consommation d'énergie des nœuds à cause de la ressource énergétique limitées de ces nœuds. Nous avons présenté le MLCP (Maximum Lifetime Coverage Problem) dont le but est de maximiser la durée de couverture offerte par le réseau. Nous nous sommes intéressés au MLCP dans le cas de la couverture de cibles et la

couverture de surface et nous avons montré que le MLCP pour la couverture de surface peut être ramené à un MLCP pour la couverture de cibles. Après avoir formellement défini le MLCP pour la couverture de cibles, nous avons présenté les deux approches d'approximation de ce problème NP-difficile à savoir l'approche par couvrants disjoints et l'approche par couvrants non disjoints. Nous avons noté que la seconde approche donne de meilleurs résultats que la première dans certains cas. Pour chaque approche, nous avons présenté certains algorithmes approchés existants dans la littérature. Ces algorithmes approchés utilisent différentes techniques parmi lesquelles la programmation linéaire, l'algorithme génétique et la technique gloutonne. Nous avons noté que la plupart de ces heuristiques utilisent la notion de cible critique définie comme la cible la moins couverte. Deux variantes de cette définition sont proposées : la variante statique et la variante dynamique. La variante statique définit la cible critique comme étant celle la moins couverte par les capteurs après le déploiement du réseau. De ce fait, les cibles critiques sont identifiées une seule fois et ne changent au cours de l'algorithme. La cible critique est dynamique lorsqu'elle est définie comme la cible couverte par le moins de capteurs *disponibles* à une étape donnée de l'algorithme. La notion de disponibilité d'un capteur est définie différemment selon les heuristiques. Nous avons remarqué que ces heuristiques présentent divers inconvénients. Les heuristiques basées sur la programmation linéaire ont l'inconvénient d'avoir une complexité élevée à savoir exponentielle dans le pire des cas. Les heuristiques génétiques qui sont stochastiques doivent leur performance à leur capacité à explorer tout l'espace de solutions. Cependant, il est difficile d'estimer la taille de l'espace de solutions exploré par ces heuristiques. D'autre part, les valeurs des paramètres tels que la taille de la population et le nombre maximal de générations sont difficiles à déterminer de façon optimale, or elles influencent grandement la qualité de la solution. De même, les algorithmes gloutons présentés dans ce chapitre utilisent pour la plupart des paramètres dont les valeurs optimales sont difficiles à déterminer. Par ailleurs, ces heuristiques gloutons se basent sur des fonctions coût ou objectif dont les valeurs peuvent être les mêmes pour plusieurs capteurs à une itération donnée. Lorsque cette situation se produit, aucune stratégie n'est proposée pour choisir efficacement l'un d'eux. Enfin, certaines heuristiques gloutonnes génèrent des couvrants qui peuvent contenir des capteurs redondants, or ces capteurs peuvent être utiles dans la génération de couvrants supplémentaires. De plus, toutes les heuristiques proposées dans ce chapitre supposent que les capteurs ont une durée de vie identique. Nous montrons dans la suite que cette hypothèse n'est pas réaliste et nous supposons que les durées de vie des capteurs ne sont pas identiques. Sous cette hypothèse, nous proposons deux nouvelles heuristiques gloutonnes qui utilisent la notion de cible critique sans faire usage de paramètres.

Chapitre 4 Nouvelles heuristiques pour le MLCP

4.1 Introduction

Dans le chapitre précédent, nous avons défini le MLCP (*Maximum Lifetime Coverage Problem*) comme le problème de former une famille d'ensembles couvrants $C_j, j = 1 \dots p$ et déterminer leurs durées d'activation respectives $Lt_j, j = 1 \dots p$, de sorte à maximiser la durée de vie du réseau $Lt_1 + \dots + Lt_p$ sachant que si le capteur s_i appartient aux ensembles couvrants $C_{i_1}, \dots, C_{i_q}$ alors $Lt_{i_1} + \dots + Lt_{i_q}$ doit être inférieur à la durée de vie b_i de sa batterie [18]. Par ailleurs, nous avons noté que les heuristiques présentées dans le chapitre précédent supposent que tous les capteurs ont la même durée de vie (c.-à-d. $b_i = b$ pour tout capteur s_i). Une question peut alors être posée : « cette hypothèse est-elle réaliste ? ». La réponse à cette question est importante car si cette hypothèse n'est pas réaliste, les performances attendues en théorie (suite à des simulations) seront inférieures à celle obtenues en pratique (en environnement réel). De plus, une hypothèse plus réaliste permet de proposer des heuristiques plus efficaces et mieux adaptées pour une implémentation en environnement réel. Les auteurs de ces heuristiques supposent que les capteurs ont la même durée de vie car ils considèrent que tous les capteurs déployés dans le réseau peuvent être dotés du même type de batterie (matériaux de fabrication, capacité,...etc). Ces auteurs supposent donc que si tous les capteurs ont « la même batterie » alors ils ont la même durée de vie puisqu'ils réalisent la même fonction (exécutent le même programme) dans la plupart des cas. Afin de déterminer si cette hypothèse est réaliste, nous étudions dans la suite les facteurs qui influencent la décharge d'une batterie.

Une batterie est caractérisée par sa capacité nominale et sa tension à vide (f.e.m). La capacité d'une batterie, exprimée en ampère-heure, est la charge ou quantité d'électricité maximale qu'elle peut débiter pendant sa durée de fonctionnement. L'énergie théorique maximale d'une batterie est égale à sa capacité multipliée sa tension à vide. Dans la pratique, la capacité d'une batterie n'est pas constante mais dépend de la température ambiante, de l'intensité du courant de décharge, et de la façon dont la décharge est réalisée (continu ou discontinu) [73]. La température joue un rôle à la fois sur la f.e.m de la pile, sur la conductivité de l'électrolyte et sur la cinétique des réactions électrochimiques. Par grand froid, les éléments perdent une partie importante de leur capacité. La figure 12.b présente la variation de la capacité de la pile Energizer [74] en fonction de la température pour différentes intensités de courant. On remarque sur cette figure que la capacité de la batterie augmente avec la température jusqu'à atteindre une valeur maximale (environ 3500 mAh). Cette figure montre que la capacité de la batterie varie aussi en fonction de l'intensité du courant débité. On observe qu'à température égale, la capacité est plus grande lorsque l'intensité du

courant est plus faible. La figure 12.a présente les valeurs de la capacité de la pile Energizer à 21°C pour différentes intensités de courant. On voit sur cette figure qu'à température égale, plus le courant de décharge est élevé, plus la capacité récupérée en pratique est faible. Ainsi nous notons sur cette figure que pour un courant de décharge de 500 mA, la capacité réelle est inférieure à 1500 mAh tandis qu'elle est supérieure à 2500 mAh pour un courant de décharge d'intensité 25 mA. Ce phénomène est appelé l'effet de capacité nominale (*rate capacity effect* en anglais) [75]. En plus de la température et de l'intensité du courant de décharge, un autre facteur qui influence la capacité d'une batterie est la façon dont la décharge est réalisée. La décharge peut être continue ou discontinue. En décharge continue, un courant de décharge d'intensité constante est fournie jusqu'à épuisement de la batterie. En décharge discontinue, l'intensité du courant de décharge varie au cours du temps entre une intensité non nulle (période d'activité) et une intensité nulle ou quasi nulle (période d'inactivité). Lorsque le courant de décharge passe d'une intensité élevée à une intensité plus faible, la batterie peut récupérer la charge perdue lors de la période de décharge précédente (à fort régime). Ce phénomène est connu sous le nom d'effet de récupération (*recovery effect* en anglais) [75]. Une expérimentation a été réalisée par Handy et Timmermann [76] pour montrer l'impact de l'effet de récupération sur la durée de vie d'une batterie. Considérons deux capteurs A et B tel que le capteur A consomme continuellement un courant de 5A (décharge continue) et le capteur B consomme périodiquement un courant de 20A pendant 0.25 minute et ne consomme rien pendant le reste de la minute. Les résultats des simulations montrent que le capteur le capteur A vit moins longtemps (280.1 minutes) que le capteur B (303.3 minutes) pourtant le courant de décharge de B est quatre fois plus élevé que celui de A [76].

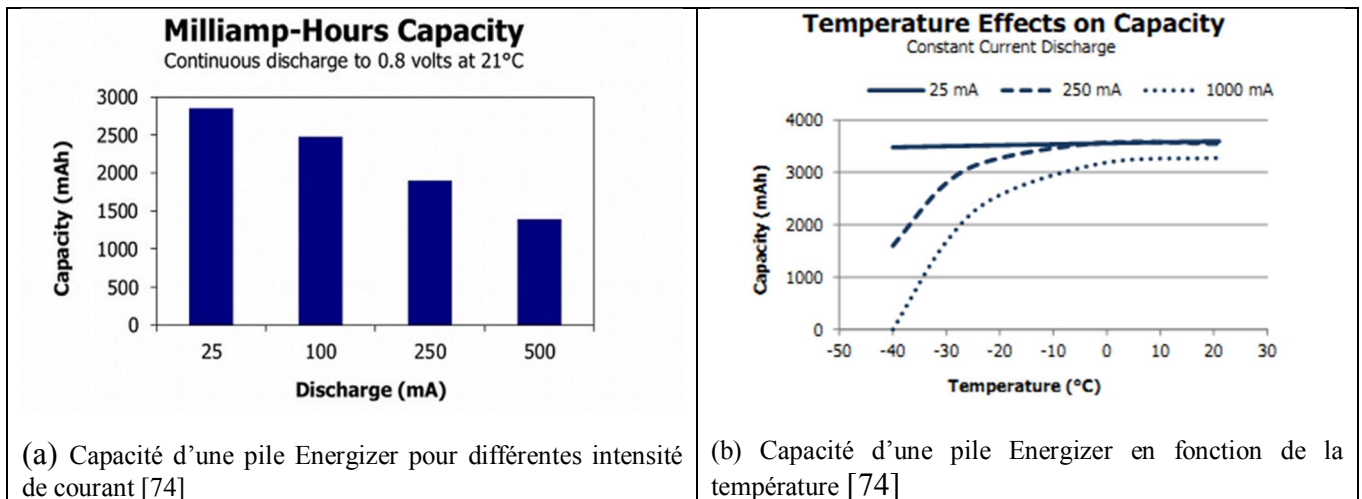


Figure 12 - Variation de la capacité en fonction de la température et du courant de décharge

Une autre caractéristique importante d'une batterie est sa courbe de décharge. La courbe de décharge d'une batterie est la représentation graphique de l'évolution de sa tension aux bornes pendant la décharge. Cette évolution est représentée soit en fonction du temps, soit en fonction de la capacité déchargée, dans des conditions fixées (généralement à courant constant sur une résistance donnée à une température donnée). La figure 13.a présente les courbes de décharge d'une pile Energizer pour différentes puissances. Nous remarquons sur cette figure qu'à la même température (21°C) la batterie a une durée de vie inférieure à 10 heures lorsqu'elle fournit une puissance de 250 mW tandis que sa durée de vie atteint environ 30 heures lorsqu'elle débite une puissance de 100 mW. Cela s'explique par le fait que plus le courant de décharge est faible et plus

la capacité de la batterie est élevée et la batterie dure plus longtemps. Considérons deux nœuds capteurs A et B dotés d'une batterie de ce type et une station de base tel que le capteur A soit très proche de la station de base et le capteur B soit très éloigné de la station de base. Puisque la puissance nécessaire pour émettre un message dépend de la distance entre l'émetteur et le récepteur, le capteur B aura une durée de vie plus courte que le capteur A bien que tous les deux capteurs réalisent la même activité. La figure 13.b présente les courbes de décharge de la pile Energizer à différentes températures pour le même courant de décharge (250 mA). On observe sur cette figure qu'à -20°C , la durée de la batterie est d'environ une heure, tandis qu'elle est d'environ 4 heures à 0°C et elle avoisine 8h à 21°C . Cette figure montre que la durée de vie d'une batterie augmente avec la température. Cela se justifie par le fait que la capacité augmente avec la température. Considérons le cas de deux capteurs A et B dotés chacun de cette batterie. Supposons que le capteur A soit placé à un endroit exposé au soleil tandis que le capteur B se trouve dans un endroit très froid, alors le capteur B aura une durée de vie inférieure à celle du capteur A bien que les deux capteurs effectuent la même activité.

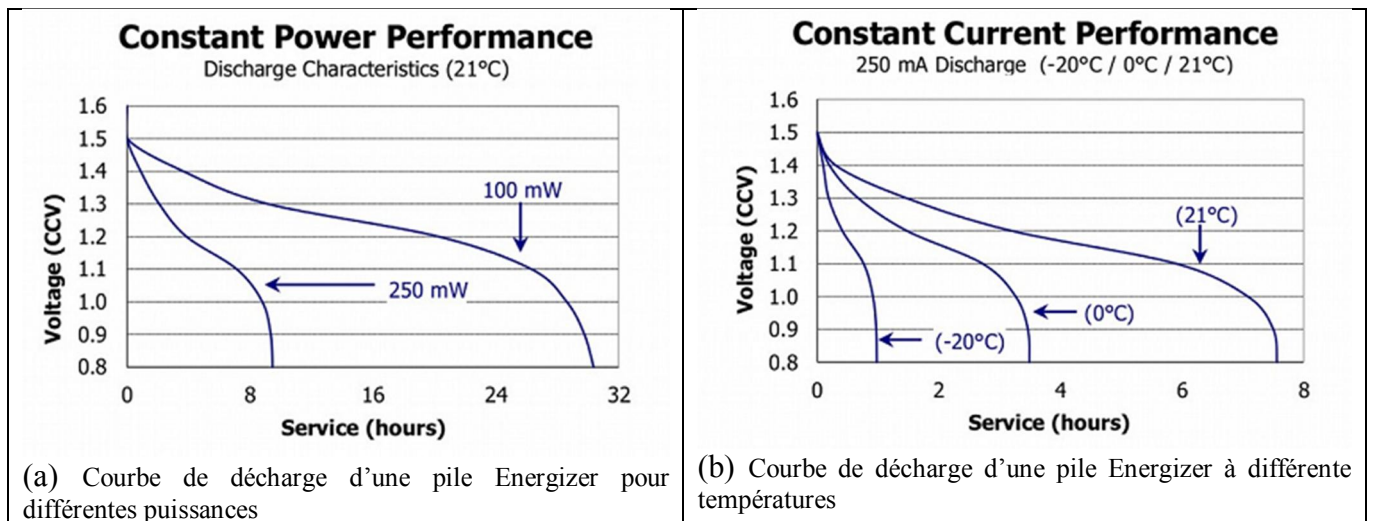


Figure 13 - Courbes de décharge d'une pile pour différentes puissances et températures

L'étude de la décharge d'une batterie montre que plusieurs capteurs dotés de batteries identiques et supposés réaliser la même fonction, n'ont pas la même durée de vie car la durée de vie d'un capteur dépend entre autres de la distance entre ce capteur et la station de base, du nombre de retransmissions effectuées pour un même message, de l'environnement autour du capteur et des périodes que le capteur passe en état de faible consommation. Nous déduisons de cela que l'hypothèse de la durée de vie identique des capteurs n'est pas réaliste. En outre, une panne peut se produire au niveau du capteur et entraîner son arrêt de fonctionnement (sa fin de vie). Si le module radio tombe en panne, les données collectées ne peuvent plus être transmises et le capteur est considéré comme défectueux. De même, le microcontrôleur ou le module de captage peuvent cesser de fonctionner empêchant ainsi le nœud de réaliser sa tâche. Enfin, un nœud de capteur peut être endommagé par des intempéries, des animaux, ou même des humains (volontairement ou involontairement). Ainsi, à cause du facteur panne, il n'est pas réaliste de supposer que tous les capteurs dotés du même type de batterie et accomplissant la même fonction ont la même durée de vie. Par ailleurs, dans la pratique, il est contraignant d'imposer que tous les nœuds d'un réseau utilisent le même type de batterie. Or, chaque type de batterie ayant ses propres caractéristiques

(courbes de décharge, effet de récupération, effet de la capacité nominale), deux batteries de types différents et utilisées dans des conditions identiques (température, courant de décharge, type de décharge) n'ont pas la même durée de vie. Une fois de plus, nous constatons qu'il n'est pas réaliste de supposer que tous les capteurs d'un réseau ont la même durée de vie.

Une autre question importante est de pouvoir estimer la durée de vie résiduelle d'une batterie. Connaître la durée de vie résiduelle des batteries des capteurs peut être utile pour prolonger la durée de vie du réseau. Nous avons vu que la durée de vie résiduelle d'une batterie est liée à son énergie résiduelle. La courbe de décharge d'une batterie permet de déterminer la charge résiduelle d'une batterie en fonction de la tension entre ses bornes. Toutefois, une courbe de décharge correspond à des conditions spécifiques (température, courant de décharge ou puissance, décharge continue) mais le fabricant de batteries ne fournit pas les courbes de décharge pour toutes les conditions de décharge possibles. De ce fait, utiliser les courbes de décharge pour estimer l'énergie résiduelle d'une batterie ne permet pas d'avoir une bonne précision. De plus, pour utiliser cette méthode, le nœud de capteur doit intégrer un composant matériel permettant de mesurer la tension aux bornes de la batterie, ce qui n'est pas le cas dans la grande majorité des plates-formes de capteurs car cela induit un coût supplémentaire. Une autre méthode pour déterminer l'énergie résiduelle d'une batterie est d'utiliser un modèle d'estimation de l'énergie consommée. L'idée est de calculer, à un instant t , la quantité d'énergie consommée par le nœud depuis sa mise en marche jusqu'à cet instant. L'énergie résiduelle à l'instant t est la différence entre l'énergie initiale de la batterie et l'énergie consommée par le nœud jusqu'à l'instant t . Il existe deux modèles permettant de calculer l'énergie consommée par un nœud jusqu'à un instant donné : le modèle linéaire et le modèle non linéaire. Dans le modèle linéaire [77], l'énergie consommée jusqu'à un instant donné est la somme des énergies consommées dans chacun des états dans lequel le capteur s'est retrouvé jusqu'à l'instant t . Dans ce modèle (1), I_m et t_m représentent le courant consommé et le temps passé par le nœud lorsque le microcontrôleur en mode actif, I_l et t_l représentent le courant consommé et le temps passé par le nœud lorsque le microcontrôleur en mode veille, I_t et t_t représentent le courant consommé et le temps passé par le nœud lorsque le module radio en mode transmission, I_r et t_r représentent le courant consommé et le temps passé par le nœud lorsque le module radio en mode réception, I_{ci} et t_{ci} représentent le courant consommé et le temps passé par le nœud lorsque les autres composants (module de captage, LEDs, ...) sont en activité et V est la tension de fonctionnement du nœud. Le principal inconvénient de ce modèle est qu'il ne prend pas en compte la non-linéarité des batteries, notamment l'effet de la capacité nominale et l'effet de la récupération.

$$\frac{E}{V} = I_m t_m + I_l t_l + I_t t_t + I_r t_r + \sum_i I_{ci} t_{ci} \quad (1)$$

Le modèle non linéaire est une amélioration du modèle linéaire qui prend en compte non seulement l'énergie consommée par le nœud dans les différents états mais aussi l'effet de la récupération [75], [78]. Dans ce modèle (2), $\sigma(t)$ est la charge consommée par le capteur à l'instant t , le premier terme représente l'énergie consommée par le nœud dans les différents états jusqu'à l'instant t et le second terme représente le comportement non linéaire de la batterie pendant les périodes de décharge et de récupération. Le paramètre β est le coefficient de diffusion des ions dans l'électrolyte. Compte tenu de la grande complexité en calcul et en espace mémoire de ce modèle, une version récursive [75] de ce modèle a été proposé pour permettre son implémentation dans un

nœud de capteur. Dans ce modèle récursif (3), $\sigma(L_n)$ est la charge consommée par la batterie à l'instant L_n et dépend récursivement de la charge consommée à l'instant $\sigma(L_{n-1})$ tel que la différence $L_n - L_{n-1} = \Delta$ est un paramètre constant qui représente la période à laquelle le nœud calcule son énergie résiduelle. Le premier terme représente l'énergie consommée par le nœud dans les différents états jusqu'à l'instant L_n et la fonction A dans le troisième terme représente le comportement non linéaire de la batterie. En termes de précision de l'estimation de l'énergie résiduelle, le modèle non linéaire original est le plus précis, suivi du modèle non linéaire récursif et enfin le modèle linéaire qui n'est pas réaliste [76].

$$\alpha(t) = \int_0^t i(\tau) d\tau + 2 \sum_{m=1}^{\infty} \int_0^t i(\tau) e^{-\beta^2 m^2 (t-\tau)} d\tau \quad (2)$$

$$\alpha(L_n) = \sum_{k=1}^n I_k t_k + \lambda \left(\alpha(L_{n-1}) - \sum_{k=1}^{n-1} I_k t_k \right) + 2 \cdot I_n \cdot A(L_n, L_{n-1} + t_n, L_{n-1}) \quad (3)$$

L'inconvénient des modèles d'estimation de la charge résiduelle d'une batterie est qu'ils ne permettent pas de déterminer a priori la durée de vie d'une batterie. En fait, pour connaître la durée de vie d'une batterie, l'activité complète du nœud (passages dans les différents états et temps de séjour dans chaque état) doit être connue en avance. Cependant, lorsqu'un nœud a un comportement périodique, sa durée de vie peut être estimée a priori à partir de ces modèles. Un comportement périodique suppose que sur chaque période, le pourcentage de temps que le capteur va séjourner dans chaque état est connu. Par exemple, sur chaque période Δ , le capteur passe 10% de temps en mode actif, 5% en mode transmission, 10% en mode réception et le reste en mode veille (faible consommation d'énergie). Lorsqu'un tel profil de consommation est connu, un des modèles ci-dessus peut être appliqué pour estimer la durée de vie résiduelle en calculant la date t à laquelle la capacité consommée est égale à la capacité initiale. Mais dans la pratique, il est difficile de connaître à l'avance l'activité complète d'un nœud. Dans certaines applications de couverture (surveillance environnementale, agriculture intelligente, ...), les nœuds du réseau peuvent avoir un comportement périodique. Par exemple, toutes les Δ minutes, les capteurs mesurent le phénomène physique observé, transmettent la mesure à la station de base, reçoivent un acquittement et se mettent en veille. Bien que ce comportement soit périodique, il est difficile d'estimer avec précision le temps qu'un capteur passe dans chaque état. Par exemple, on ne peut pas déterminer avec précision le temps qu'un capteur passe en attente de l'accès au canal radio ou en attente de la réception d'un message ou le nombre de retransmissions nécessaires pour l'envoi d'un message. Cette erreur d'estimation du temps de séjour périodique d'un capteur dans un état est encore plus grande dans le cas des réseaux multi-sauts où on ne peut pas connaître à l'avance le nombre de messages qu'un nœud sera amené à relayer vers la station de base.

Nous avons montré qu'il n'est pas réaliste de supposer que les capteurs ont la même durée de vie. Par conséquent, les heuristiques de l'approche par couvrants disjoints et celles de l'approche par couvrants non disjoints présentées dans le chapitre précédent n'auront pas dans la pratique les performances attendues. En effet, dans l'approche par couvrants disjoints, les heuristiques forment au préalable un ensemble de couvrants disjoints et les activent ensuite successivement. Dans la pratique, un nœud sera à court d'énergie avant les autres et le couvrant devra être remplacé par un

autre. Ce cycle sera répété jusqu'à ce que le dernier couvrant soit activé. Nous notons qu'au terme de ce processus, il existe dans chaque couvrant des capteurs ayant des énergies résiduelles non nulles. Ces capteurs peuvent être utilisés pour former davantage de couvrants et ainsi prolonger la durée de vie du réseau. Mais en appliquant l'approche par couvrants disjoints, les énergies résiduelles de ces capteurs sont gaspillées. Dans l'approche par couvrants disjoints, l'idée est de former un ensemble de couvrants tel que chaque capteur appartienne à au plus w couvrants et d'activer chaque couvrant pendant une durée b/w où b est la durée de vie des batteries. Il est difficile d'implémenter cette approche dans la pratique car toutes les batteries n'ont pas la même durée de vie et la durée de vie d'une batterie ne peut être estimée à l'avance avec précision. A partir du comportement périodique des nœuds et de la capacité nominale des batteries on peut estimer la valeur de b et implémenter l'approche par couvrants non disjoints. Cependant, dans la pratique les couvrants ne fonctionneront pas pendant la durée b/w (comme prévue) à cause de l'erreur sur l'estimation de la valeur de b . D'autre part, les couvrants auront des capteurs résiduels comme dans le cas de l'approche par couvrants disjoints et l'énergie résiduelle sera gaspillée au lieu d'être utilisée pour prolonger la durée de vie du réseau.

Contrairement aux heuristiques de la littérature, nous supposons dans la suite que le réseau est constitué d'un ensemble de capteurs $s_1, \dots, s_n$ avec des durées de vie respectives $b_1, b_2, \dots, b_n$. Cependant, tout comme les heuristiques existantes, nous supposons que l'énergie consommée par un capteur en veille est négligeable. De ce fait, la durée de vie b_i d'un capteur s_i correspond au temps total que le capteur s_i met (jusqu'à épuisement de sa batterie) pour surveiller les cibles qu'il couvre sans entrer en veille. Afin d'évaluer la performance des nouvelles heuristiques que nous proposons, il est nécessaire de déterminer une borne supérieure de la durée de vie du réseau de la solution optimale du MLCP.

Considérons un réseau constitué d'un ensemble de capteurs $S = \{s_1, \dots, s_n\}$ déployés pour couvrir un ensemble de cibles $T = \{t_1, \dots, t_m\}$ tel que chaque cible t soit couverte par un sous-ensemble $S_{[t]}$ de capteurs ($S_{[t]} \subset S$). Supposons que les capteurs $s_1, \dots, s_n$ ont des durées de vie respectives $b_1, b_2, \dots, b_n$. Si tous les capteurs sont activés au déploiement du réseau, alors la cible t a une durée de couverture maximale d_t égale à celle du capteur de plus grande durée de vie parmi les capteurs qui la couvrent. En d'autres termes, la durée de couverture maximale de la cible t en l'absence d'optimisation d'énergie est $d_t = \max_{s_i \in S_{[t]}} b_i$. Puisque la durée de vie du réseau est le temps pendant lequel toutes les cibles sont couvertes, la durée de vie du réseau si on n'applique aucune technique d'optimisation d'énergie, est $\Delta = \min_{t \in T} \max_{s_i \in S_{[t]}} b_i$. Cependant, si on optimise la consommation d'énergie en activant à chaque instant un sous-ensemble de capteurs tel que chaque cible soit couverte par un seul capteur alors la durée de couverture maximale D_t de la cible t est égale la somme des durées de vie des capteurs qui la couvrent, c'est-à-dire que $D_t = \sum_{s_i \in S_{[t]}} b_i$. Etant donné que certaines cibles peuvent être doublement couvertes par les capteurs d'un couvrant, alors la durée de vie du réseau lorsqu'on optimise la consommation d'énergie est inférieure ou égale à $\Delta_{sup} = \min_{t \in T} \sum_{s_i \in S_{[t]}} b_i$. Ainsi la durée de vie maximale du réseau qu'on peut obtenir en optimisant la consommation d'énergie est égale à la durée de vie Δ_{opt} de la solution optimale du MLCP tel que $\min_{t \in T} \max_{s_i \in S_{[t]}} b_i \leq \Delta_{opt} \leq \min_{t \in T} \sum_{s_i \in S_{[t]}} b_i$. Connaissant cette borne supérieure de la durée de vie optimale du réseau, il est possible d'évaluer la qualité d'une solution du MLCP.

Soit Δ_A la durée de vie du réseau d'une solution A d'une instance du MLCP. Plus le ratio $\frac{\Delta_A}{\Delta_{opt}}$ est proche de 1, plus la solution A est proche de l'optimum. Sachant que $\frac{\Delta_A}{\Delta_{opt}} \geq \frac{\Delta_A}{\Delta_{sup}}$, on en déduit que si le ratio $\frac{\Delta_A}{\Delta_{sup}}$ est proche de 1, alors la solution A est proche de l'optimum. Ainsi, la qualité d'une solution d'une instance du MLCP peut être estimée en comparant sa durée de vie à la borne supérieure Δ_{sup} de la durée de vie du réseau. Un ratio par rapport à la borne supérieure qui est proche de 1, indique que la solution est de bonne qualité. Nous utiliserons cette métrique dans la suite pour estimer la performance des nouvelles heuristiques et les comparer à celles de la littérature. Nous présentons dans la suite les nouvelles heuristiques que nous avons proposées pour le MLCP.

4.2 Heuristiques proposées

4.2.1 Approche réactive

Puisque dans la pratique les durée de vie des capteurs ne sont pas identiques, nous proposons une nouvelle approche de construction des couvrants appelée approche *réactive*. Contrairement aux heuristiques de la littérature qui forment tous les couvrants dans une première étape et les activent successivement dans une seconde étape, dans l'approche réactive, on forme un seul couvrant et on l'active. Un nouveau couvrant est formé au moment où un capteur du couvrant actif a une énergie résiduelle proche de 0. Ce processus est réitéré jusqu'à ce que les capteurs à énergie résiduelle non nulle ne couvrent plus toutes les cibles. Cette approche est plus adaptée à l'environnement réel et permet de réagir en temps réel en cas de panne. En effet, chaque capteur du couvrant actif informe périodiquement la station de base de son énergie résiduelle. Si la station de base ne reçoit pas le message périodique d'un capteur du couvrant actif, celui-ci est considéré comme défectueux et un nouveau couvrant est formé et activé. Par ailleurs, dans la nouvelle approche, la durée de vie d'un couvrant est égale à la durée de vie du premier capteur qui est à cours d'énergie. Il est important de noter que dans cette approche, les couvrants sont non disjoints mais contrairement à l'approche classique par couvrants non disjoints, le nombre de couvrants auxquels un capteur peut appartenir n'est pas fixé à l'avance et varie d'un capteur à l'autre. La figure 14 présente les diagrammes de l'approche classique et de l'approche réactive. On observe sur cette figure que l'approche réactive comporte trois étapes principales. L'étape 1 (*Former un couvrant*) est l'étape de construction d'un couvrant, c'est à cette étape qu'un algorithme de construction d'un couvrant est appelé. L'étape 2 (*Activer le couvrant*) consiste à activer tous les capteurs du couvrant généré à l'étape 1 et mettre les autres capteurs en veille. L'étape 3 consiste à recevoir périodiquement les énergies résiduelles des capteurs en activité. On passe de l'étape 1 à l'étape 3 si un capteur est à cours d'énergie et chaque cible est couverte par au moins un capteur dont l'énergie résiduelle est non nulle. En utilisant cette nouvelle approche, proposer une solution pour le MLCP consiste à définir l'algorithme de construction de couvrants qui sera exécuté à l'étape 1. Nous proposons dans la suite deux nouveaux algorithmes à cet effet : l'algorithme *adaptatif* et l'algorithme *blacklist*.

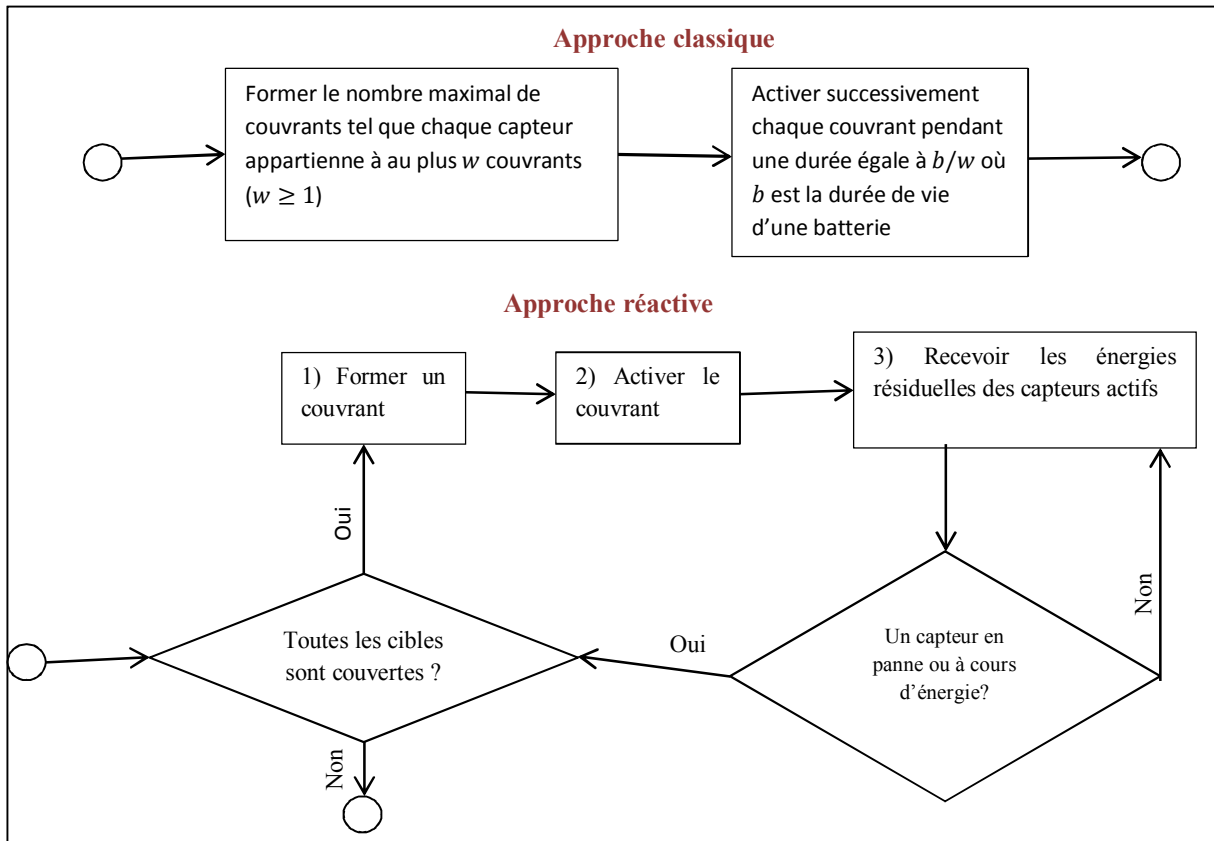


Figure 14 - L'approche classique et l'approche réactive

4.2.2 Algorithme adaptatif

L'algorithme adaptatif [79] est un algorithme glouton qui implémente l'étape 1 (*Former un couvrant*) de l'approche réactive décrite plus haut. Son objectif est de construire un couvrant. L'idée principale de l'algorithme adaptatif est d'obtenir un nouveau couvrant en réparant le couvrant précédent. Rappelons que dans l'approche réactive, un nouveau couvrant est formé lorsqu'un capteur du couvrant actif est à cours d'énergie ou tombe en panne. Lorsque cela se produit, le couvrant est alors défectueux puisque certaines cibles ne sont plus couvertes par ce couvrant. Les cibles non couvertes par le couvrant en cours sont appelées cibles orphelines. En remplaçant le capteur défectueux de sorte que toutes les cibles orphelines soient de nouveau couvertes, on obtient un nouveau couvrant. Lors de cette réparation, toutes les cibles couvertes par le capteur défectueux ne sont pas nécessairement orphelines car certaines cibles peuvent être doublement couvertes par les capteurs du couvrant. Cependant, il existe toujours au moins une cible orpheline et le couvrant obtenu après réparation est toujours différent du couvrant précédent. Le capteur défectueux peut être remplacé par un ou plusieurs capteurs et il peut arriver que certains capteurs du couvrant précédent deviennent redondants. Un capteur est redondant dans un couvrant si toutes les cibles qu'il couvre sont couvertes par un autre capteur du couvrant. Ainsi, en supprimant ce capteur du couvrant, toutes les cibles restent toujours couvertes. De ce fait, il est important après la réparation d'un couvrant de rechercher et de supprimer les capteurs redondants du nouveau couvrant. Ces capteurs redondants sont placés en mode veille comme les autres capteurs qui n'appartiennent pas au couvrant actif. En général dans l'algorithme adaptatif, un capteur actif reste en activité jusqu'à épuisement de sa batterie (ou en cas de panne) excepté lorsqu'il devient redondant suite à une

réparation de couvrant. D'autre part, seul le premier couvrant n'est pas explicitement le fruit d'un dépannage de couvrant car il est construit partant de zéro. Dans un souci d'uniformité, le premier couvrant est vu comme un couvrant obtenu suite à une réparation où toutes les cibles sont des cibles orphelines.

Afin de réparer un couvrant défectueux, l'algorithme adaptatif définit les critères de sélection du (ou des) capteur(s) à activer pour couvrir les cibles orphelines. La première étape est de déterminer l'ensemble des capteurs qui couvrent au moins une cible orpheline. Cet ensemble est un sous ensemble des capteurs en veille car aucun capteur actif ne couvre de cibles orphelines. La seconde étape est de compter le nombre de capteurs (en veille) qui couvre chaque cible orpheline. Connaissant le nombre de capteurs qui couvre chaque cible orpheline, on peut déterminer l'ensemble des cibles critiques. Une cible critique est une cible orpheline couverte par le plus petit nombre de capteurs. La troisième et dernière étape est de sélectionner le capteur qui couvre le plus grand nombre de cibles critiques. L'idée étant de couvrir en priorité les cibles critiques. Il peut arriver que plusieurs capteurs couvrent le nombre maximal de cibles critiques, dans ce cas, on sélectionne celui qui couvre le plus grand nombre de cibles orphelines en plus des cibles critiques. Si plusieurs capteurs remplissent ce critère, l'un d'eux est alors choisi aléatoirement. S'il reste des cibles orphelines après l'ajout de ce capteur, on recommence le processus depuis la première étape. Les capteurs sont ainsi ajoutés itérativement jusqu'à ce qui ne reste plus de cibles orphelines. On note que l'ensemble des cibles critiques est dynamique car il change en fonction des itérations. Une cible orpheline non critique à l'itération n peut le devenir à l'itération $n + 1$. Etant donné que lors d'une réparation, le nombre de capteurs qui couvrent une cible ne varie pas, les cibles critiques de l'itération n sont couvertes par moins de capteurs que les cibles critiques de l'itération $n + 1$. On constate alors que l'algorithme adaptatif couvre les cibles orphelines en partant des plus critiques vers les moins critiques. Ainsi, l'algorithme cherche à couvrir une cible orpheline couverte par m capteurs (itération $n + 1$) uniquement s'il n'existe plus de cibles orphelines couvertes par moins de m capteurs (itérations $1, \dots, n$). Une fois le couvrant réparé, l'algorithme recherche et supprime les capteurs redondants du nouveau couvrant. L'algorithme adaptatif est présenté à la figure 15.

L'algorithme adaptatif prend en entrée l'ensemble S des capteurs dont l'énergie résiduelle est non nulle, l'ensemble T des cibles et le couvrant C à réparer. Lors du premier appel de l'algorithme, l'ensemble S contient tous les capteurs du réseau, le couvrant C est l'ensemble vide et l'ensemble des cibles orphelines T_o est égal à l'ensemble des cibles T . Au fur et à mesure des appels de l'algorithme, la taille de l'ensemble S diminue car des capteurs épuisent leurs batteries et d'autres tombent en panne. De même, à partir du second appel de l'algorithme la taille de l'ensemble des cibles orphelines est strictement inférieure à celle de l'ensemble des cibles. Il est à noter que l'ensemble C est un sous-ensemble de l'ensemble de S (c.à.d. $S \cap C = C$). L'algorithme arrive à réparer le couvrant défectueux si toutes les cibles orphelines sont couvertes par au moins un capteur de S . Cette condition est vérifiée (instruction I1) avant de procéder à la construction du couvrant. Dans le cas contraire, l'algorithme retourne l'ensemble. L'instruction I2 recherche l'ensemble T_o des cibles orphelines. Au début de chaque itération de la boucle *tantque*, l'instruction I3 recherche l'ensemble $T_{critique}$ des cibles critiques qui sont les cibles orphelines couvertes par le moins de capteurs de l'ensemble S . L'instruction I4 filtre l'ensemble S pour former l'ensemble S_u des capteurs qui couvrent le plus grand nombre de cibles critiques. L'instruction I5 sélectionne le

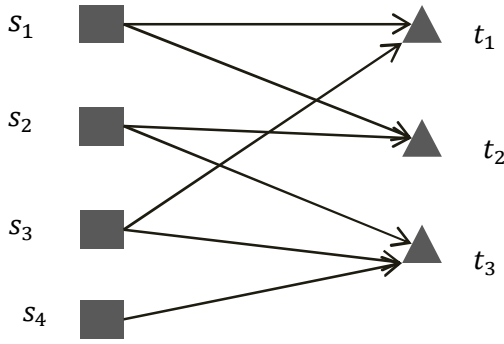
capteur de S_u qui couvre le plus grand nombre de cibles orphelines (critiques et non critiques). Une fois sélectionné, le capteur est ajouté au couvrant (instruction I6) et les cibles couvertes par ce capteur sont supprimées de l'ensemble T_o (Instruction I7). Une nouvelle itération de la boucle recommence si l'ensemble T_o n'est pas vide, sinon la boucle s'arrête. A la sortie de la boucle, l'instruction I8 exécute la procédure *suppression_redondance* afin de rechercher et de supprimer les capteurs redondants du couvrant formé. Pour chaque capteur du couvrant, la procédure *suppression_redondance* le supprime premièrement du couvrant, puis si une cible couverte par ce capteur n'est pas couverte par le couvrant, le capteur est réinséré dans le couvrant car cela signifie qu'il n'est pas redondant. La procédure *suppression_redondance* retourne donc un couvrant dépourvu de capteur redondant, c'est ce couvrant qui est retourné par l'algorithme adaptatif (instruction I9).

Lors du premier appel de l'algorithme adaptatif, on effectue au plus $m(m + n)$ opérations où m est le nombre de cibles et n est le nombre de capteurs. A partir du second appel, on effectue au plus $m'(m' + n)$ opérations où $m' \leq m$ car m' représente le nombre de cibles orphelines. Un inconvénient de l'algorithme adaptatif est qu'il peut y avoir plusieurs capteurs qui remplissent le critère de sélection et l'un d'eux est choisi aléatoirement. Le choix qui est fait à ce moment-là n'est pas nécessairement le plus judicieux.

Algorithme Adaptatif (S, T, C)**Input:** T : Ensemble de cibles ; S : Ensemble de capteurs ; C : Couvrant à réparer**Début**I1. **Si** $\exists$ une cible $t \in T$ et $\nexists$ un capteur $s \in S$ tel que s couvre t , **alors Retourner** $\emptyset$ I2. Identifier l'ensemble T_o des cibles orphelines (cibles non couvertes par les capteurs de C)**Tant que** $T_o \neq \emptyset$ **Faire**I3. Identifier l'ensemble des cibles critiques $T_{critique} \subset T_o$ (cibles couvertes par le nombre minimal de capteurs de S)I4. Identifier l'ensemble $S_u \subset S$ de capteurs qui couvrent le nombre maximal de cibles de $T_{critique}$ I5. Sélectionner le capteur $s \in S_u$ qui couvre le nombre maximal de cibles de T_o I6. Ajouter le capteur s au couvrant : $C = C \cup \{s\}$ I7. Supprimer de T_o toutes les cibles couvertes par le capteur s **Fintantque**I8. Appeler la procédure *suppression_redondance*(C, T) pour supprimer les capteurs redondantsI9. **Retourner** le couvrant C **FinAlgo**

Procédure *suppression_redondance* (C, T)**Input :** T : Ensemble de cibles ; C : Couvrant**Début****Pour tout** capteur $s \in C$ **Faire**Supprimer le capteur s du couvrant : $C = C - \{s\}$ **Pour toute** cible t couverte le capteur s **Si** la cible t n'est pas couverte par le couvrant C **Alors** Ajouter le capteur s au couvrant C : $C = C \cup \{s\}$ **Finsi****Finpour****Finpour****Retourner** le couvrant C **FinProc**

Figure 15 - Algorithme adaptatif



Ensemble de capteurs $S = \{s_1, s_2, s_3, s_4\}$; Ensemble de cibles $T = \{t_1, t_2, t_3\}$

Durées de vie : $b_1 = 2$; $b_2 = 1.5$; $b_3 = 3$; $b_4 = 2.5$

Relation de couverture : $s_1 = \{t_1, t_2\}$; $s_2 = \{t_1, t_2, t_3\}$; $s_3 = \{t_1, t_2, t_3\}$; $s_4 = \{t_3\}$

Figure 16 - Une instance de MLCP

Etape 1 : Premier appel de l'algorithme adaptatif : *adaptatif*(S, T, C)

$S = \{s_1, s_2, s_3, s_4\}$; $T = \{t_1, t_2, t_3\}$; $C = \emptyset$; $b_1 = 2$; $b_2 = 1.5$; $b_3 = 3$; $b_4 = 2.5$

Itération 1 : $T_o = \{t_1, t_2, t_3\}$; $T_{critique} = \{t_1, t_2\}$; $S_u = \{s_1\}$; on choisit s_1 ;

Itération 2 : $T_o = \{t_3\}$; $T_{critique} = \{t_3\}$; $S_u = \{s_2, s_3, s_4\}$; on choisit s_2 ;

On retourne $C = \{s_1, s_2\}$

Etape 2 : On active le couvrant C ,

Etape 3 : le capteur s_2 épuise sa batterie à la date $\Delta_1 = 1.5$

Etape 1 : Second appel de l'algorithme adaptatif : *adaptatif*(S, T, C)

$S = \{s_1, s_3, s_4\}$; $T = \{t_1, t_2, t_3\}$; $C = \{s_1\}$; $b_1 = 0.5$; $b_3 = 3$; $b_4 = 2.5$

Itération 1 : $T_o = \{t_3\}$; $T_{critique} = \{t_3\}$; $S_u = \{s_3, s_4\}$; on choisit s_3 ;

On retourne $C = \{s_1, s_3\}$

Etape 2 : On active le couvrant C ,

Etape 3 : le capteur s_1 épuise sa batterie à la date $\Delta_1 = 1.5 + 0.5 = 2$

$S = \{s_3, s_4\}$; $T = \{t_1, t_2, t_3\}$; $b_3 = 2.5$; $b_4 = 2.5$

Aucun capteur de S ne couvre la cible t_2 : Fin de vie du réseau

La durée de vie du réseau est $\Delta_1 = 2$

Figure 17 - Exécution 1 de l'approche réactive avec l'algorithme adaptatif

Etape 1 : Premier appel de l'algorithme adaptatif : *adaptatif* (S, T, C)
 $S = \{s_1, s_2, s_3, s_4\}; T = \{t_1, t_2, t_3\}; C = \emptyset; b_1 = 2; b_2 = 1.5; b_3 = 3; b_4 = 2.5$
 Itération 1 : $T_o = \{t_1, t_2, t_3\}; T_{critique} = \{t_1, t_2\}; S_u = \{s_1\}$; on choisit s_1 ;
 Itération 2 : $T_o = \{t_3\}; T_{critique} = \{t_3\}; S_u = \{s_2, s_3, s_4\}$; on choisit s_4 ;
 On retourne $C = \{s_1; s_4\}$
Etape 2 : On active le couvrant C ,
Etape 3 : le capteur s_1 épuise sa batterie à la date $\Delta_2 = 2$

Etape 1 : Second appel de l'algorithme adaptatif : *adaptatif* (S, T, C)
 $S = \{s_2, s_3, s_4\}; T = \{t_1, t_2, t_3\}; C = \{s_4\}; b_2 = 1.5; b_3 = 3; b_4 = 0.5$
 Itération 1 : $T_o = \{t_1, t_2\}; T_{critique} = \{t_1, t_2\}; S_u = \{s_2, s_3\}$; on choisit s_2 ;
 Itération 2 : $T_o = \{t_1\}; T_{critique} = \{t_1\}; S_u = \{s_3\}$; on choisit s_3 ;
 On supprime les capteurs redondants du couvrant $C = \{s_4; s_2; s_3\}$: s_4 est redondant
 On retourne $C = \{s_2; s_3\}$
Etape 2 : On active le couvrant C ,
Etape 3 : le capteur s_2 épuise sa batterie à la date $\Delta_2 = 2 + 1.5 = 3.5$

$S = \{s_3, s_4\}; T = \{t_1, t_2, t_3\}; b_3 = 1.5; b_4 = 0.5$
 Aucun capteur de S ne couvre la cible t_2 : Fin de vie du réseau

La durée de vie du réseau est $\Delta_2 = 3.5$

Figure 18 - Exécution 2 de l'approche réactive avec l'algorithme adaptatif

Afin de comprendre les limites de l'algorithme adaptatif, examinons la figure 17 et la figure 18 qui présentent deux traces d'exécutions de l'approche réactive avec l'algorithme adaptatif sur l'exemple de la figure 16. A l'itération 2 du premier appel de l'algorithme adaptatif, trois capteurs (s_2, s_3, s_4) remplissent les conditions pour être sélectionnés dans le couvrant. Sur la figure 17, le capteur choisi aléatoirement est le capteur s_2 , tandis que sur la figure 18, c'est le capteur s_4 qui est sélectionné aléatoirement. En comparant les deux traces d'exécutions, on constate que la durée de vie du réseau est $\Delta_1 = 2$ dans le premier cas et $\Delta_1 = 3.5$ dans le second cas, ce qui représente un gain de 75% par rapport à la durée de vie obtenu dans le premier cas. Cet exemple permet de voir que le choix aléatoire d'un capteur parmi plusieurs candidats peut avoir un effet considérable (positif ou négatif) sur la durée de vie du réseau. Nous constatons aussi que dans le second cas, le capteur choisi (s_4) ne couvre aucune des cibles critiques (t_1, t_2) de l'itération précédente. Cette observation laisse à penser qu'une meilleure gestion des cibles critiques peut permettre d'obtenir une plus grande durée de vie du réseau. C'est dans cette optique que nous proposons dans la suite un nouvel algorithme, à savoir l'algorithme *blacklist*.

4.2.3 Algorithme blacklist

L'algorithme blacklist est un algorithme glouton de construction de couvrant. Il est invoqué à l'étape 1 de l'approche réactive. L'objectif de cet algorithme est de minimiser le nombre de capteurs qui couvrent chaque cible critique dans le couvrant. Pour cela, l'algorithme se base sur l'utilisation d'une liste noire (*blacklist*). La liste noire contient les capteurs qui couvrent au moins

une cible critique déjà couverte par le couvrant en construction. Les capteurs en liste noire ne sont sélectionnés que lorsque les capteurs hors de la liste noire ne couvrent pas les cibles non encore couvertes par le couvrant en construction. Cette méthode permet donc d'éviter autant que possible qu'une cible critique ne soit doublement couverte par un couvrant, l'idéal étant que chaque cible critique soit couverte par un seul capteur du couvrant. Rappelons qu'une cible est critique si elle est couverte par le nombre minimal de capteurs. En utilisant judicieusement les capteurs qui couvrent ces cibles critiques, la durée de vie du réseau pourrait être davantage prolongée. Nous avons vu dans le chapitre précédent que certaines heuristiques de la littérature utilisent des fonctions coût pour pénaliser les capteurs qui couvrent les cibles critiques. Cependant, les autres paramètres de ces fonctions peuvent compenser cette pénalité au point où un capteur couvrant une cible critique déjà couverte soit sélectionné. Contrairement à cette technique (par fonction coût), les capteurs que l'on ne souhaite pas sélectionner sont clairement identifiés grâce à la liste noire et lorsque la situation l'impose, le choix de l'un d'entre eux peut se faire selon des critères bien définis. Un autre principe de base de l'algorithme blacklist est de couvrir en priorité toutes les cibles critiques avant de chercher à couvrir les autres cibles. Au début de la couverture des cibles critiques, la liste noire est vide et lorsque toutes les cibles critiques sont couvertes, cette liste est pleine. Étant donné que les capteurs en liste noire sont (en quelque sorte) exclus, on peut définir une autre forme de criticité. Il s'agit des cibles couvertes par le nombre minimal de capteurs hors de la liste noire. Afin de couvrir les cibles non critiques, l'algorithme cherche à couvrir en priorité la cible couverte par le nombre minimal de capteurs hors de la liste noire. Dans certains cas, ce nombre est nul, autrement dit, tous les capteurs qui couvrent cette cible se trouvent en liste noire. Cela implique qu'un capteur doit être pris en liste noire et par conséquent, une ou plusieurs cibles critiques seront couvertes doublement. Dans ce cas, l'algorithme choisit le capteur qui couvre le plus de nouvelles cibles afin de minimiser la probabilité que cette situation se reproduise. De manière générale, l'algorithme blacklist cherche à minimiser la taille des couvrants qu'il forme. Pour cela, lorsqu'un choix de capteurs se pose, la priorité est donnée à celui qui couvre le nombre maximal de nouvelles cibles (critiques dans un cas, et non critiques dans l'autre). Si plusieurs capteurs sont dans cette même situation, l'un d'eux est alors choisi aléatoirement.

L'algorithme blacklist est présenté à la figure 19. Il prend en entrée l'ensemble S des capteurs et l'ensemble T des cibles. La première instruction (I1) vérifie que toutes les cibles sont couvertes par au moins un capteur. Si tel n'est pas le cas, l'algorithme retourne l'ensemble vide. La seconde instruction (I2) identifie l'ensemble (T_C) des cibles critiques. Il s'agit des cibles couvertes par le moins de capteurs de S . Toutes les cibles n'appartenant pas à l'ensemble T_C sont ajoutées à l'ensemble T_N des cibles non critiques, à l'instruction I3. Ces cibles critiques et non critiques sont séparées ainsi car elles sont traitées différemment par l'algorithme. À l'instruction I4, le couvrant et la liste noire sont initialisés à l'ensemble vide. La première boucle *tantque* concerne les cibles critiques T_C . Elle s'arrête lorsque toutes les cibles critiques sont couvertes. L'instruction I5 sélectionne le capteur s hors de la liste noire ($s \in (S - B)$) qui couvre le plus de nouvelles cibles critiques. Si aucun capteur hors de la liste noire ne couvre nouvellement une cible critique, l'instruction I5 sélectionne le capteur s de la liste noire qui couvre le plus de nouvelles cibles critiques. Lorsque le capteur s est ainsi sélectionné, il est ajouté au couvrant (I6) et supprimé de son ensemble d'origine (S ou B) selon le cas (I9). De même, toutes les cibles couvertes par le nouveau capteur s sont supprimées de leur ensemble d'origine (T_C ou T_N) selon le cas (I7). Pour chaque

nouvelle cible critique couverte par s , tous les autres capteurs qui la couvrent sont déplacés de l'ensemble S vers la liste noire B (I18). Lorsque toutes les cibles critiques sont couvertes, on passe à la seconde boucle *tantque* qui cherche à couvrir les cibles non critiques. Si l'ensemble T_N est vide à la sortie de la première boucle, la seconde boucle est ignorée. A chaque itération, la cible t la moins couverte par les capteurs hors de la liste noire est sélectionnée (I10). S'il y'en a plusieurs, l'une d'entre elles est choisie aléatoirement. A l'instruction (I11), l'algorithme cherche un capteur qui couvre t et qui ne se trouve pas en la liste noire. Si plusieurs capteurs remplissent cette condition, on cherche celui qui couvre le plus de nouvelles cibles. S'il y'en a plusieurs, on choisit aléatoirement l'un d'eux. Par contre, si tous les capteurs qui couvrent la cible t se trouvent en liste noire, on sélectionne l'un d'entre eux selon le même critère que précédemment. Lorsque le capteur s est ainsi choisi, on l'ajoute dans le couvrant (I12), on le supprime de S et B selon le cas et on supprime de T_N toutes les cibles nouvellement couvertes. Si l'ensemble T_N n'est pas vide, on démarre une nouvelle itération. Les itérations de la seconde boucle se succèdent ainsi jusqu'à ce que toutes les cibles non critiques soient couvertes ($T_N = \emptyset$). A la sortie de la seconde boucle, toutes les cibles sont couvertes et l'algorithme appelle la procédure ***suppression_redondance***(C, T) (Fig. 3.4) pour rechercher et supprimer d'éventuels couvrants redondants du couvrant formé (I14). Enfin, l'algorithme retourne le couvrant (I15). La complexité de l'algorithme blacklist est en $O(mn)$ où m est le nombre de cibles et n est le nombre de capteurs.

Algorithme Blacklist (S, T)**Input:** T : Ensemble de cibles ; S : Ensemble de capteurs**Début**

- I1. **Si** $\exists$ une cible $t \in T$ et $\nexists$ un capteur $s \in S$ tel que s couvre t , **alors Retourner** $\emptyset$
- I2. Identifier l'ensemble T_C des cibles critiques (cibles couvertes par le nombre minimal de capteurs de S)
- I3. Former l'ensemble T_N des cibles non critiques ($T_N = T - T_C$)
- I4. Initialiser le couvrant et la liste noire B à l'ensemble vide : $B = \emptyset$; $C = \emptyset$

Tant que $T_C \neq \emptyset$ **Faire**

- I5. **Si** $\exists s \in (S - B)$ tel que s couvre une cible de T_C **Alors**
 - Sélectionner le capteur $s \in (S - B)$ qui couvre le nombre maximal de cibles de T_C
- Sinon**
 - Sélectionner le capteur $s \in B$ qui couvre le nombre maximal de cibles de T_C
- Finsi**
- I6. Ajouter le capteur s au couvrant : $C = C \cup \{s\}$
- I7. Supprimer de T_C et de T_N toutes les cibles couvertes par le capteur s
- I8. Ajouter à la liste noire (B) tous les capteurs qui couvrent les mêmes cibles critiques que le capteur s
- I9. Supprimer le capteur s de l'ensemble S ou de l'ensemble B selon le cas

Fintantque**Tant que** $T_N \neq \emptyset$ **Faire**

- I10. Choisir la cible $t \in T_N$ couverte par le nombre minimal de capteurs de $S - B$
- I11. **Si** aucun capteur de $S - B$ ne couvre la cible t **Alors**
 - Sélectionner le capteur $s \in B$ qui couvre la cible t et le nombre maximal d'autres cibles de T_N
- Sinon**
 - Sélectionner le capteur $s \in S - B$ qui couvre à t et le nombre maximal d'autres cibles de T_N
- Finsi**
- I12. Ajouter le capteur s au couvrant : $C = C \cup \{s\}$
- I13. Supprimer de T_N toutes les cibles couvertes par le capteur s
- I14. Supprimer le capteur s de l'ensemble S et de l'ensemble B selon le cas

Fintantque

- I14. Appeler la procédure *suppression_redondance*(C, T) pour supprimer les capteurs redondants de C

- I15. **Retourner** le couvrant C

FinAlgo

Figure 19 - Algorithme Blacklist

Etape 1 : Premier appel de l'algorithme blacklist : $blacklist(S, T)$
 $S = \{s_1, s_2, s_3, s_4\}; T = \{t_1, t_2, t_3\}; b_1 = 2; b_2 = 1.5; b_3 = 3; b_4 = 2.5$
 $B = \emptyset; C = \emptyset; T_C = \{t_1, t_2\}; T_N = \{t_3\}$
 Itération 1 boucle 1 : $S - B = \{s_1, s_2, s_3, s_4\}$; on choisit s_1 ;
 $B = \{s_2, s_3\}; T_C = \emptyset; S = \{s_2, s_3, s_4\}$
 Itération 1 boucle 2 : $T_N = \{t_3\}; S - B = \{s_4\}$ on choisit s_4 ;
 On retourne $C = \{s_1, s_4\}$
Etape 2 : On active le couvrant C ,
Etape 3 : le capteur s_1 épuise sa batterie à la date $\Delta_3 = 2$

Etape 1 : Second appel de l'algorithme blacklist : $blacklist(S, T)$
 $S = \{s_2, s_3, s_4\}; T = \{t_1, t_2, t_3\}; b_2 = 1.5; b_3 = 3; b_4 = 0.5$
 $B = \emptyset; C = \emptyset; T_C = \{t_1, t_2\}; T_N = \{t_3\}$
 Itération 1 boucle 1 : $S - B = \{s_2, s_3, s_4\}$; on choisit s_2 ;
 $B = \emptyset; T_C = \{t_1\}; T_N = \emptyset; S = \{s_3, s_4\}$
 Itération 2 boucle 1 : $S - B = \{s_3, s_4\}$; on choisit s_3 ;
 On retourne $C = \{s_2, s_3\}$
Etape 2 : On active le couvrant C ,
Etape 3 : le capteur s_2 épuise sa batterie à la date $\Delta_3 = 2 + 1.5 = 3.5$
 $S = \{s_3, s_4\}; T = \{t_1, t_2, t_3\}; b_3 = 1.5; b_4 = 0.5$
 Aucun capteur de S ne couvre la cible t_2 : Fin de vie du réseau

La durée de vie du réseau est $\Delta_3 = 3.5$

Figure 20 - Exécution de l'approche réactive avec l'algorithme blacklist

La figure 20 montre une trace d'exécution de l'approche réactive avec l'algorithme blacklist sur l'exemple de la figure 16. On observe que la durée de vie du réseau obtenue est $\Delta = 3.5$, c'est la même que celle obtenue par l'algorithme adaptatif lorsque le « bon » capteur a été choisi aléatoirement (Fig. 3.7). Cela montre que l'algorithme blacklist propose une meilleure gestion des cibles critiques que l'algorithme adaptatif. Il reste à présent à comparer cette durée de vie à la borne supérieure (définie plus haut) de la durée de vie optimale. Pour cela, on calcule premièrement la durée de couverture maximale de chaque cible selon la formule $D_t = \sum_{s_i \in S_{[t]}} b_i$. On obtient alors $D_{t_1} = b_1 + b_3 = 2 + 3 = 5$, $D_{t_2} = b_1 + b_2 = 2 + 1.5 = 3.5$ et $D_{t_3} = b_2 + b_3 + b_4 = 1.5 + 3 + 2.5 = 7$. Puis on identifie la plus petite de ces durées maximales de couverture. Dans ce cas, nous avons $\Delta_{sup} = 3.5$. Puisque $\Delta_{sup} = \Delta$, nous déduisons que l'algorithme blacklist a permis d'obtenir la durée de vie optimale du réseau dans le cas de l'exemple 2.

4.3 Simulations et analyse des résultats

Dans cette section, l'objectif est d'étudier les performances des heuristiques que nous proposons à savoir l'algorithme adaptatif et l'algorithme blacklist intégrés chacun dans l'approche réactive. Les performances de ces heuristiques sont comparées à certains algorithmes approchés gloutons de la littérature notamment les algorithmes Greedy-MS [20], Static-CCF [26], Dynamic CCF [26] et MCS-based. L'algorithme MCS-based est celui qui cherche à construire itérativement des couvrants minimaux en utilisant l'algorithme glouton classique du problème du couvrant minimal [21]. Ces

heuristiques sont simulées sur des réseaux de capteurs déployés uniformément sur une surface pour couvrir un ensemble de cibles elles-mêmes déployées uniformément sur la dite surface. La métrique utilisée pour évaluer la performance d'une solution est le ratio entre la durée de vie de la solution et la borne supérieure de la durée de vie du réseau présentée dans l'introduction. Lorsque ce ratio est proche de 1, la solution est proche de l'optimum et s'il vaut 1, la solution est optimale. Chaque algorithme est exécuté sur 100 réseaux différents et retourne 100 valeurs de ratio. Ces valeurs sont présentées sous la forme de boîte à moustache. Une boîte à moustache (*BoxPlot* en anglais) permet de visualiser la dispersion d'une série de valeurs en représentant le minimum, le premier quartile, la médiane, le troisième quartile et le maximum de la série comme présenté à la figure 21.

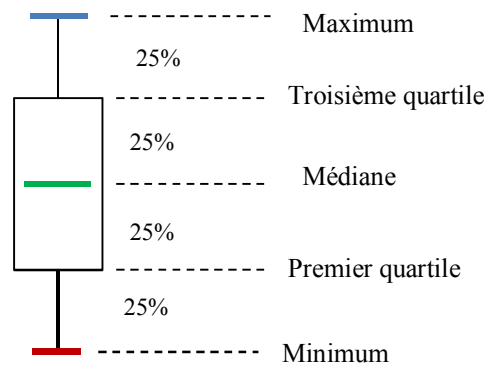


Figure 21 - Boîte à moustache

Les paramètres des simulations sont donnés dans le tableau 1. Nous considérons une surface de déploiement carré de 10m x10m et des capteurs dont le rayon de sensibilité est égal à 1m. Le nombre de cibles déployées uniformément est de 50. Cela correspond au nombre de cellules de dimensions 1m x 2m (sans chevauchement) qu'on peut obtenir en quadrillant la surface dans le but de la couvrir en totalité (une cellule pouvant être considérée comme une cible). Les réseaux simulés comptent 150 capteurs dans un cas et 300 capteurs dans l'autre. Le but du MLCP étant de prolonger la durée de vie du réseau, le nombre de capteurs doit être supérieur au nombre de cibles. Dans le cas des réseaux de 150 capteurs, une cible est couverte par plus ou moins trois capteurs compte tenu du déploiement uniforme à la fois des capteurs et des cibles. De même, une cible est couverte par environ 6 capteurs pour les réseaux de 300 capteurs. Nous voulons observer l'impact de la redondance de couverture des cibles sur la performance des différents algorithmes. Enfin, nous supposons que la durée de vie des capteurs suit une loi normale de moyenne 100 et d'écart-type 10 dans un cas et 20 dans l'autre. Ces deux valeurs de l'écart-type permettront d'observer l'impact de la différence entre les durées de vie des capteurs sur la performance des algorithmes. Ces simulations sont réalisées sous l'environnement Linux. Tous les algorithmes sont programmés en langage C++ de même que le générateur de réseau permettant de simuler le réseau. Le générateur de réseau génère un ensemble de coordonnées uniformément répartis pour simuler les cibles et un second ensemble pour simuler les capteurs. La loi uniforme de répartition des capteurs et des cibles, ainsi que la loi normale des durées de vie des capteurs sont implémentés par la librairie scientifique GNU appelée GSL (GNU Scientific Library). Lorsque les positions des cibles et des capteurs sont connues, le générateur de réseau détermine pour chaque cible, le sous-ensemble de capteurs qui la couvre en se basant sur le rayon de sensibilité et la distance entre cette cible et les capteurs. A la fin de son exécution, le générateur de réseau retourne une instance du MLCP constituée d'un ensemble

des capteurs, un ensemble des cibles et la relation de couverture entre les capteurs et les cibles. L'instance ainsi générée est passée à un algorithme pour être résolue. Nous avons implémenté les algorithmes de la littérature en considérant que les couvrants générés sont disjoints, autrement dit $w = 1$ (w est le nombre maximal de couvrants auxquels un capteur peut appartenir). Deux raisons justifient ce choix. D'une part, ces algorithmes supposent que les capteurs ont la même durée de vie et $\frac{1}{w}$ est la fraction de cette durée de vie que chaque capteur passe dans un couvrant. Or dans notre cas, les capteurs n'ayant pas tous la même durée de vie, il est plus simple de poser $w = 1$ car chaque capteur passe tout son temps d'activité dans un seul couvrant. D'autre part, les auteurs de ces articles ont montré par des simulations que de manière générale, les performances de ces algorithmes sont quasiment les mêmes lorsque les couvrants générés sont disjoints ($w = 1$) ou non disjoints ($w > 1$). De ce fait, le choix du paramètre w égal à 1 ne diminue pas la performance de ces algorithmes. Les valeurs des paramètres α, β, γ des algorithmes Static-CCF et Dynamic-CCF sont celles choisies par les auteurs de ces algorithmes pour les simulations de l'algorithme Static-CCF à savoir $\alpha = 0.35$, $\beta = 0.02$ et $\gamma = 0.63$ [26].

Tableau 1- Paramètres de simulation 1

Paramètre	valeur
Surface	10 x 10
Rayon de sensibilité	1
Nombre de cibles	50 (réparties uniformément sur la surface)
Nombre de capteurs	150 et 300 (déployés uniformément sur la surface)
Durée de vie des capteurs	Loi normale de moyenne 100 et d'écart-type 10 et 20

La figure 22 montre les performances des algorithmes pour des réseaux de 150 capteurs dont l'écart-type des durées de vie est égal à 10. On observe sur cette figure que les algorithmes de la littérature (Dynamic CCF, Static CCF, Greedy-MSC et MCS-based) ont des performances similaires et de moins bonne qualité que les nouveaux algorithmes (adaptatif et blacklist). En effet, les nouveaux algorithmes trouvent la solution optimale (ratio = 1) dans 75% des cas tandis que les anciens algorithmes ont un ratio supérieur à 0.8 dans 75% des cas environ. Ces résultats s'expliquent par le fait que les anciens algorithmes supposent que les durées de vie des capteurs sont identiques et par conséquent plusieurs capteurs résiduels (à énergie résiduelle non nulle) sont gaspillés.

La figure 23 montre les performances des algorithmes pour des réseaux de 150 capteurs dont l'écart-type des durées de vie est égal à 20. Tout comme sur la figure précédente, les anciens algorithmes ont des performances similaires et moins bonnes que celles des nouveaux algorithmes. Les nouveaux algorithmes trouvent la solution optimale dans 75% des cas tandis que les anciens algorithmes obtiennent un ratio supérieur à 0.6 dans 75% des cas. En comparant la figure 22 et la figure 23, on observe que la performance des anciens algorithmes diminue lorsque l'écart-type augmente. En effet, les anciens algorithmes donnent un ratio supérieur à 0.8 dans moins de 50% des cas lorsque l'écart-type est égal à 20 contre un ratio supérieur à 0.8 dans 75% des cas lorsque l'écart-type est égal à 10. Ce résultat est logique car plus l'écart-type est grand, plus les durées de vie des capteurs sont dispersées et plus l'énergie résiduelle des capteurs gaspillés est grande. Par

ailleurs, on note que la variation de la dispersion des durées de vie des capteurs n'a pas de conséquence sur la performance des nouveaux algorithmes.

La figure 24 montre les performances des algorithmes pour des réseaux de 300 capteurs dont l'écart-type des durées de vie est égal à 10. Une de plus les nouveaux algorithmes trouvent la solution optimale dans 75% des cas tandis que les anciens algorithmes ont un ratio supérieur à 0.8 dans moins de 75% des cas. Les nouveaux algorithmes donnent toujours de meilleurs résultats que les anciens même lorsque le nombre de capteurs augmentent. Par ailleurs, on comparant la figure 22 et la figure 24, on constate que l'augmentation du nombre de capteurs n'a pas d'effet sur la performance des nouveaux algorithmes contrairement aux anciens algorithmes. On observe que la performance des anciens algorithmes diminue lorsque le nombre de capteurs augmente. En effet, les anciens algorithmes donnent un ratio inférieur à 0.9 dans 75 % des cas pour 300 capteurs contre 50% des cas pour 150 capteurs.

Sur la figure 25, les réseaux sont constitués de 300 capteurs dont l'écart-type des durées de vie est égal à 20. On observe que les nouveaux algorithmes sont meilleurs que les anciens et donnent la solution optimale dans 75% des cas tandis que les anciens algorithmes donnent un ratio supérieur à 0.6 dans environ 75% des cas. Cette figure confirme l'impact de l'écart-type et du nombre de capteurs sur la performance des anciens algorithmes tel qu'observé que les figures précédentes. En comparant la figure 24 et la figure 25, on note que la performance des anciens algorithmes diminue lorsqu'on passe de l'écart-type 10 (ratio supérieur à 0.8 dans 75% des cas environ) à l'écart-type 20 (ratio inférieur à 0.7 dans 75% des cas). De même, en comparant la figure 23 et la figure 25, on constate que la performance des anciens algorithmes diminue lorsqu'on passe de 150 capteurs (ratio entre 0.6 et 0.9 dans 50% des cas) à 300 capteurs (ratio entre 0.6 et 0.7 dans 50% des cas).

En comparant les figures 22 à 25, on observe que l'algorithme blacklist donne un ratio supérieur à 0.8 dans 100% tandis que l'algorithme adaptatif donne un ratio supérieur 0.6 dans 100%. Cela signifie que l'algorithme blacklist est meilleur que l'algorithme adaptatif bien que tous deux atteignent l'optimum dans 75% des cas. Cette différence peut s'expliquer par le fait que l'algorithme blacklist a une meilleure gestion des cibles critiques que l'algorithme adaptatif.

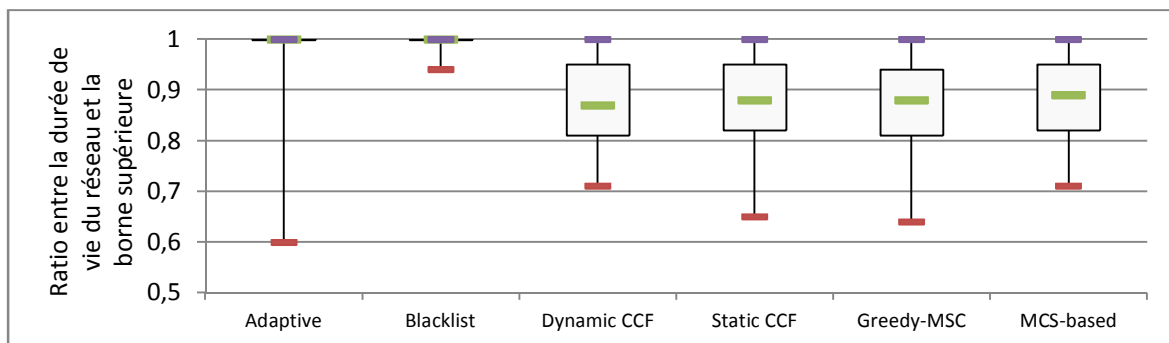


Figure 22 - Performance des algorithmes pour 150 capteurs et écart-type = 10

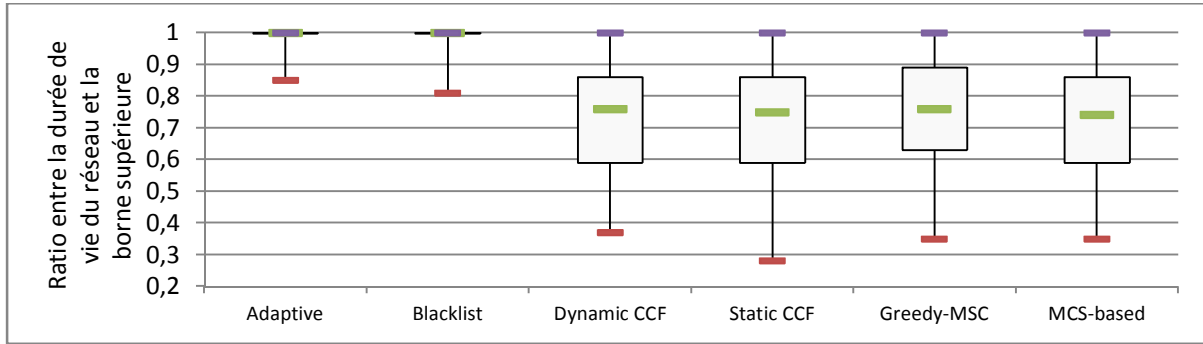


Figure 23 - Performance des algorithmes pour 150 capteurs et écart-type = 20

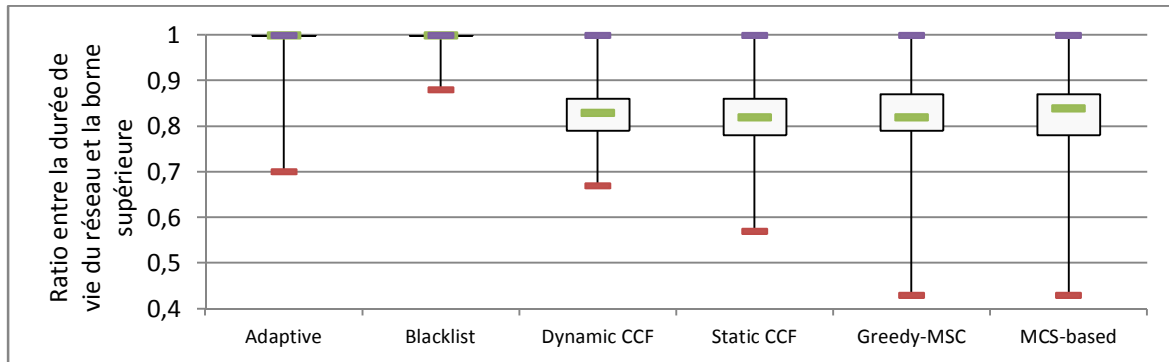


Figure 24 - Performance des algorithmes pour 300 capteurs et écart-type = 10

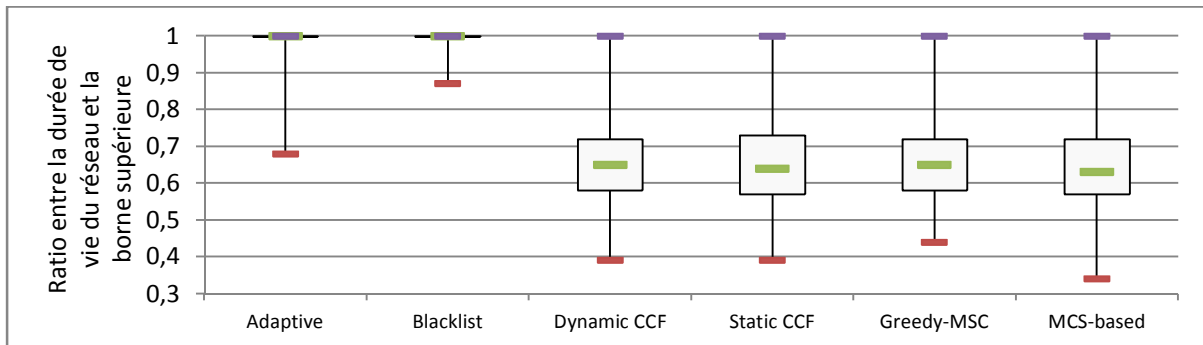


Figure 25 - Performance des algorithmes pour 300 capteurs et écart-type = 20

4.4 Conclusion

Dans ce chapitre, nous avons montré que l'hypothèse de durée de vie identique des capteurs n'est pas réaliste. Nous avons donc considéré une hypothèse plus réaliste sur les durées de vie des capteurs à savoir que chaque capteur a sa durée de vie propre. Partant de cette hypothèse, nous avons proposé une nouvelle approche de résolution du MLCP, appelée approche réactive. Contrairement à l'approche classique qui consiste à construire les couvrants en une étape et à les activer dans une seconde étape, l'approche réactive ne construit un nouveau couvrant que lorsque le couvrant actif est déféctueux. Cette approche a l'avantage d'être plus adaptée à l'environnement réel notamment en cas de panne d'un capteur. En s'appuyant sur l'approche réactive, nous avons proposé deux nouveaux algorithmes de construction des couvrants: l'algorithme adaptatif et l'algorithme blacklist. L'idée de base de l'algorithme adaptatif est de construire un nouveau couvrant en remplaçant le capteur déféctueux dans le couvrant précédent. L'algorithme blacklist

quant à lui utilise une liste noire afin de minimiser le nombre de capteurs qui couvrent une cible critique dans un couvrant. Les simulations réalisées permettent de constater que ces nouveaux algorithmes ont une performance supérieure à celles des anciens algorithmes de la littérature. Bien que les deux nouveaux algorithmes trouvent la solution optimale dans 75% des cas simulés, l'algorithme blacklist est meilleur que l'algorithme adaptatif quand on considère tous les 100% des cas simulés. Dans ce chapitre nous avons supposé comme pour les anciens algorithmes, que l'énergie consommée par un capteur en veille (n'appartenant pas au couvrant actif) est négligeable. En réalité, cette hypothèse n'est pas réaliste. Dans la suite, nous justifions cette affirmation et nous proposons une nouvelle heuristique pour le MLCP qui prend en compte la consommation d'énergie des capteurs en veille.

Chapitre 5 Prise en compte de la consommation en veille dans le MLCP

5.1 Introduction

Dans la littérature, il est généralement admis que l'énergie consommée par un capteur en état de veille est négligeable. Cette hypothèse est justifiée par le fait que l'énergie consommée par un microcontrôleur en veille est très faible par rapport à celle consommée lorsque le microcontrôleur est en mode actif. Par exemple, nous avons vu dans le chapitre 1 que le microcontrôleur Atmel ATmega consomme environ $75 \mu W$ en mode veille et $15 mW$ en mode actif, soit un ratio de 0.005. Ce ratio est encore plus faible si le capteur est en cours d'émission ou de réception. Prenons l'exemple du module LoRa Microchip RN2483 qui consomme $42.6 mW$ en mode réception, $116.7 mW$ en mode émission à la puissance maximale ($14 dBm$) et $30 \mu W$ en mode veille. Le ratio entre la consommation d'énergie du capteur (microcontrôleur et module radio) en mode veille et lorsqu'il est en mode réception est de 0.0018 et le ratio entre la consommation d'énergie du capteur en mode veille et lorsqu'il est en mode émission est de 0.0008. Cet exemple montre que l'on peut supposer que l'énergie consommée par un capteur en veille est négligeable.

Dans le MLCP, seuls quelques capteurs sont actifs pendant une période donnée tandis que les autres sont en veille. Dans la littérature, on suppose que l'énergie consommée par les capteurs en veille est négligeable. Cette hypothèse est-elle réaliste ? Pour répondre à cette question, étudions le comportement d'un capteur en mode actif et celui d'un capteur en mode veille. Lorsqu'un couvrant est activé, tous ses capteurs sont placés en mode actif. Le rôle des capteurs actifs est de surveiller l'ensemble des cibles. Durant la surveillance, un capteur passe par plusieurs modes de fonctionnement : mode capture de données, mode traitement des données, mode émission des données, mode réception et mode veille. Dans le mode capture de données, seuls le microcontrôleur et le module de captage sont actifs. Une fois la mesure prélevée, le capteur passe en mode traitement des données où le module de captage est éteint et le microcontrôleur seul est actif. Lorsque la donnée est traitée, le capteur allume sa radio en mode émission pour transmettre la donnée vers la station de base. Une fois le message envoyé, le capteur met sa radio en mode réception pour attendre le message d'acquiescement de la station de base. Lorsque ce message est reçu, le capteur passe en mode veille en éteignant sa radio et en plaçant son microcontrôleur en mode basse consommation. Cette séquence d'activité se déroule périodiquement. On voit donc qu'un capteur actif n'est pas toujours en mode actif mais se met de temps en temps en mode veille. Regardons ce qui se passe du côté des capteurs en veille, c'est-à-dire les capteurs qui n'appartiennent pas au couvrant en cours d'activité. Si un capteur en veille reste constamment en veille (radio éteinte), il ne pourra plus être contacté par la station de base. En d'autres termes, un

capteur constamment en mode veille ne peut plus être réveillé pour passer en mode actif. De ce fait, un capteur en veille doit maintenir le contact avec la station de base pour savoir à quel moment il devra se réveiller pour surveiller les cibles. Pour cela, le capteur en veille doit envoyer périodiquement des messages à la station pour l'informer de son bon état de fonctionnement et recevoir en retour un éventuel ordre de changement d'activité. Cela signifie qu'un capteur en veille passe aussi par plusieurs modes de fonctionnement : le mode émission de données, le mode réception de données, le mode traitement de données et le mode veille. Dans le mode émission, la radio du capteur est en mode émission et transmet un message de « signalement » à la station de base. Puis le capteur met sa radio en mode réception pour écouter la réponse de la station de base. Cette réponse peut être un simple acquittement ou un ordre de changement d'activité. Le capteur éteint sa radio et passe ensuite en mode traitement de données pour traiter le message reçu et agir en conséquence. Si le message est un simple acquittement, le capteur met son microcontrôleur en mode basse consommation d'énergie et entre de nouveau en mode veille. Cette séquence d'activité est répétée de façon périodique. En comparant le capteur en veille au capteur actif, on constate que les deux capteurs sont en réalité en activité et qu'un capteur en veille est en fait un capteur actif ayant une activité réduite. Partant de ce constat, on peut dire que le MLCP consiste à mettre un sous-ensemble de capteurs en mode d'activité normale tandis que les autres capteurs sont placés en mode d'activité réduite. Il reste à déterminer si l'énergie consommée par un capteur en activité réduite (que nous appelons capteur en veille) est négligeable comparée à l'énergie consommée par un capteur en activité normale (que nous appelons capteur actif).

Nous avons vu qu'un capteur en veille passe dans les mêmes modes de fonctionnement qu'un capteur actif, excepté le mode de capture de données. La différence principale entre les activités d'un capteur en veille et celui d'un capteur actif est le temps de passage dans chaque mode. En effet, sur une même période donnée, un capteur actif passe plus de temps en mode émission et réception qu'un capteur en veille. Cela s'explique par le fait que le capteur actif doit transmettre fréquemment des mesures à la station de base tandis que le capteur en veille envoie un message à la station de base de temps en temps en fonction de la latence de communication désirée. Le capteur actif passe aussi plus de temps en mode traitement que le capteur en veille car ce dernier reçoit moins de messages à analyser et aucune mesure prélevée à traiter. Sachant que les modes d'émission et de réception sont ceux qui consomment le plus d'énergie par rapport aux autres modes, nous pouvons comparer les énergies consommées en considérant uniquement le temps que chacun d'eux passe dans ces deux modes. Pour cela, considérons une période T et notons n_a le nombre de transmissions (émission et réception) faites par un capteur actif pendant la période T et n_v le nombre de transmissions faites par un capteur en veille pendant la même période. Puisque l'énergie consommée pour envoyer et recevoir un message est la même pour les deux capteurs, alors le ratio r entre l'énergie consommée par le capteur en veille et celle consommée par le capteur actif pendant la période T peut être estimée par la formule $r = \frac{n_v}{n_a}$. Le tableau 2 montre des exemples de valeurs de r pour une période $T = 20 \text{ min}$. Dans le cas 1, le capteur actif transmet un message par minute contre un message toutes les 5 minutes pour le capteur en veille. Le ratio de consommation d'énergie est de 0.2. Dans le cas 2, le capteur actif envoie un message par minute tandis que le capteur en veille envoie un message toutes les 10 minutes. Le ratio de consommation est de 0.1 et est plus faible que celui du cas 1, mais cela a une conséquence sur la communication entre la station de base et le capteur en veille. En fait, plus le nombre de messages émis par le

capteur en veille est petit et plus le temps de latence de la communication est grand. Plus le temps de latence de communication entre la station de base et le capteur est grand et plus la station de base doit attendre pour transmettre un ordre de changement d'état au capteur. Par exemple, si la station de base veut demander à un capteur en veille de devenir actif, elle devra attendre 5 minutes dans le cas 1, 10 minutes dans le cas 2 et 3 et 20 minutes dans le cas 4. Ce temps d'attente peut être préjudiciable pour les applications critiques car pendant ce temps d'attente, certaines cibles ne sont pas couvertes (d'où la nécessité de réveiller le capteur en veille). Pour une application de détection d'incendie par exemple, un incendie peut se déclencher et se propager considérablement en 10 ou 20 minutes pendant que la station de base attend qu'un capteur en veille se réveille pour lui demander de surveiller les cibles. Pour la plupart des applications, le ratio de consommation d'énergie entre un capteur en veille et un capteur actif est supérieur à 0.1 et peut être supérieur à 0.5 dans certaines applications peu critiques. Dans le cas 3 par exemple, un capteur actif envoie un message toutes les 5 minutes tandis qu'un capteur en veille le fait toutes les 10 minutes (deux fois moins). Ce cas peut se retrouver dans des applications de surveillance où 10 minutes sans surveillance n'entraîne pas de conséquences dramatiques (agriculture intelligente, surveillance environnementale, ... etc). Tout ceci montre que l'énergie consommée par un capteur en veille n'est pas négligeable par rapport à celle consommée par un capteur et l'hypothèse faite dans la littérature à ce propos n'est pas réaliste.

Tableau 2 - Ratio d'énergie entre un capteur en veille et un capteur actif

	Cas 1	Cas 2	Cas 3	Cas 5
n_v	4	2	2	1
n_a	20	20	4	4
r	0.2	0.1	0.5	0.25

Dans la suite nous supposons qu'un capteur actif consomme une énergie E_a sur chaque période T tandis qu'un capteur en veille consomme une énergie E_v durant la même période. Le ratio de consommation d'énergie r est donné par la formule $r = \frac{E_v}{E_a} = \frac{P_v}{P_a}$ où P_v et P_a sont les puissances moyennes respectives d'un capteur en veille et d'un capteur actif. On note que $r \in [0; 1]$. Nous supposons que les paramètres P_v et P_a sont connus et dépendent des contraintes de l'application telles que le temps de latence de la communication entre la station de base et les capteurs en veille, la fréquence de transmission des mesures du phénomène observé, ...etc. Nous avons vu dans le chapitre 3 que les méthodes d'estimation de l'énergie résiduelle des batteries ne permettent pas de déterminer a priori la durée de vie résiduelle des capteurs. Cette durée de vie peut néanmoins être estimée si le capteur a un profil périodique connu de consommation d'énergie. Cependant, d'après l'hypothèse ci-dessus, un capteur peut avoir deux profils de consommation d'énergie : profil actif avec une puissance P_a et profil en veille avec une puissance P_v . De plus, le temps qu'un capteur passe dans chacun de ces deux profils de consommation n'est pas connu à l'avance car il dépend de l'algorithme du MLCP. De ce fait, nous supposons dans la suite que les capteurs $s_1, \dots, s_n$ du MLCP ont des énergies initiales $E_1, E_2, \dots, E_n$ et qu'à chaque instant présent t , l'énergie résiduelle de chaque capteur est connue (car elle peut être estimée par les méthodes d'estimation présentées au chapitre 3). En considérant l'hypothèse de consommation non nulle des capteurs en veille, il est important de déterminer la nouvelle borne supérieure de la durée de vie optimale du réseau. En effet, la borne supérieure calculée au chapitre 3 est un cas particulier où $P_v = 0$ (c.-à-d. $r = 0$). Dans la section suivante nous calculons la borne supérieure de la durée de vie optimale du réseau

pour $r \in [0; 1]$. Cette borne supérieure permettra d'estimer la qualité d'une solution d'une instance du MLCP de façon plus réaliste.

5.2 Borne supérieure de la durée de vie du réseau

Le MLCP a pour but de prolonger la durée de vie du réseau en activant à chaque instant un sous-ensemble de capteurs tandis que les autres sont placés dans un mode de basse consommation. Dans le cas idéal, chaque cible est couverte à chaque instant par un seul capteur actif. La durée de vie correspondant au cas idéal est la borne supérieure de la durée de vie optimale du réseau. Cette borne supérieure est la plus petite des durées maximales de couverture des cibles. En d'autres termes, soit D_t la durée de couverture maximale de la cible t et $\Delta_{sup,r}$ la borne supérieure de la durée de vie du réseau lorsque $r \neq 0$, alors $\Delta_{sup,r} = \min_{t \in T} D_t$. Pour calculer la borne supérieure, il faut donc déterminer la durée de couverture maximale de chaque cible. Lorsque $P_v = 0$ (c-à-d $r = 0$), la durée de couverture maximale de la cible t est donnée par la formule $D_t = \sum_{s_i \in S[t]} \frac{E_i}{P_a}$. Cependant, lorsque $P_v \neq 0$, le calcul de la durée maximale de couverture d'une cible n'est pas trivial. La proposition 1 ci-après nous permet de calculer la durée de couverture maximale d'une cible.

Proposition 1

Soit t une cible couverte par p capteurs $s_0, s_1, \dots, s_{p-1}$ d'énergies respectives $E_0, E_1, \dots, E_{p-1}$. La durée de couverture D_t de t est maximale lorsque les p capteurs sont activés dans l'ordre croissant de leurs énergies.

Preuve

Considérons le cas de 2 capteurs s_0 et s_1 d'énergies E_0 et E_1 tel que $E_0 \leq E_1$. Soit Δ_c la durée de couverture lorsque les capteurs sont activés dans l'ordre croissant de leurs énergies (s_0, s_1) et Δ_d la durée de couverture lorsqu'ils sont activés dans l'ordre décroissant de leurs énergies (s_1, s_0). Alors, $\Delta_c = \frac{E_0(1-r)+E_1}{P_a}$ tandis que $\Delta_d = \frac{E_1(1-r)+E_0}{P_a}$ si $E_0 > r.E_1$ et $\Delta_d = \frac{E_1}{P_a}$ si $E_0 \leq r.E_1$. On constate que $\Delta_c > \Delta_d$. Donc la séquence d'activation s_0, s_1 a une durée de couverture plus grande que la séquence s_1, s_0 et la proposition 1 est vérifiée.

Pour $p > 2$, notons Seq la séquence $s_0, s_1, \dots, s_{i-2}, s_{i+1}, \dots, s_{p-1}$ et considérons deux capteurs s et s' d'énergies respectives E et E' . Supposons que $E < E'$. Nous avons vu que la durée de couverture de la séquence s, s' est plus grande que la durée de couverture de la séquence s', s . Insérons les deux capteurs dans la séquence, respectivement dans l'ordre croissant et l'ordre décroissant de leurs énergies. Soit Seq_1 la séquence $s_0, s_1, \dots, s_{i-2}, s, s', s_{i+1}, \dots, s_{p-1}$ et Seq_2 la séquence $s_0, s_1, \dots, s_{i-2}, s', s, s_{i+1}, \dots, s_{p-1}$. Soit D_1 et D_2 les dates où le capteur s_{i+1} est activé respectivement dans la séquence Seq_1 et la séquence Seq_2. Puisque $D_1 > D_2$, alors à la date D_2 , la séquence Seq_1 a un capteur de plus s^* d'énergie $(D_1 - D_2) * P_a$ par rapport à la séquence Seq_2. Donc la séquence Seq_1 a une durée de couverture plus longue que la séquence Seq_2. Ceci montre que dans une séquence d'activation optimale, deux capteurs consécutifs sont dans l'ordre croissant de leurs énergies, et par conséquent tous les capteurs sont dans l'ordre croissant de leurs énergies. ■

Soit t une cible couverte par p capteurs $s_0, s_1, \dots, s_{p-1}$ d'énergies respectives $E_0, E_1, \dots, E_{p-1}$ tel que $E_0 \leq E_1 \leq \dots \leq E_{p-1}$. D'après la proposition 1, la durée de couverture maximale de la cible t est obtenue lorsque les capteurs $s_0, s_1, \dots, s_{p-1}$ sont activés dans cet ordre, sachant qu'un seul capteur à la fois est actif et le capteur suivant est activé lorsque l'énergie du capteur courant est épuisée. Notons t'_i le temps pendant lequel le capteur s_i est en mode actif (c.-à-d. fonctionnant à la puissance P_a) et t_i le temps maximal de fonctionnement en mode actif, alors $t_i = E_i/P_a$. La durée de couverture maximale de la cible t est $D_t = t'_0 + t'_1 + \dots + t'_{p-1}$. Il reste à calculer les valeurs des t'_i . Pour le capteur s_0 , on a $t'_0 = t_0$ car c'est le premier capteur à être activé. Pour le capteur s_1 on a: $t'_1 = t_1 - r t'_0$ car au moment d'activer s_1 , son énergie résiduelle E'_1 est égale à son énergie initiale moins l'énergie qu'il a dépensée en veille lorsque le capteur s_0 était actif (c.-à-d. $E'_1 = E_1 - r E_0$). De même, pour le capteur s_2 , son énergie résiduelle E'_2 au moment de son activation est égale à son énergie initiale moins l'énergie qu'il a dépensée en veille pendant que s_0 et s_1 étaient actifs successivement (c.-à-d. $E'_2 = E_2 - r E_0 - r E'_1$), donc $t'_2 = t_2 - r(t'_0 + t'_1)$. De façon générale, on voit que $t'_i = t_i - r(t'_0 + t'_1 + \dots + t'_{i-1})$ pour $0 < i < p$ et $t'_0 = t_0$. En intégrant les t'_i à la formule $D_t = t'_0 + t'_1 + \dots + t'_{p-1}$, on trouve que la durée de couverture maximale de la cible t couverte par p capteurs $s_0, s_1, \dots, s_{p-1}$ d'énergies respectives $E_0, E_1, \dots, E_{p-1}$ tel que $E_0 \leq E_1 \leq \dots \leq E_{p-1}$ est donnée par la formule (1) ci-après où $t_i = \frac{E_i}{P_a}$.

$$\begin{cases} \text{si } p = 1, D_t = t_0 \\ \text{si } p = 2, D_t = t_0(1 - r) + t_1 \\ \text{si } p > 2, D_t = \sum_{i=0}^{p-1} t_i - t_0[1 - (1 - r)^{p-1}] - r \sum_{i=2}^{p-1} \sum_{j=1}^{i-1} t_j(1 - r)^{i-j-1} \end{cases} \quad (1)$$

La borne supérieure est donc $\Delta_{sup,r} = \min_{t \in T} D_t$ où la durée de couverture maximale D_t de chaque cible t est calculée selon la formule (1). Etant donné que la borne supérieure $\Delta_{sup,r}$ correspond à la durée de couverture maximale des cibles critiques, la formule (1) montre qu'une cible critique n'est pas nécessairement une cible couverte par le plus petit nombre de capteurs car la valeur de D_t dépend à la fois du nombre de capteurs qui couvrent la cible t et des énergies initiales des capteurs. Ainsi, une cible couverte par moins de capteurs peut avoir une durée de couverture supérieure à une autre cible couverte par plus de capteurs. Dans la suite, nous proposons un algorithme de construction de couvrants qui prend en compte l'énergie consommée par les capteurs en veille et identifie les cibles critiques d'après la formule (1) et non comme les cibles couvertes par le plus petit nombre de capteurs.

5.3 Algorithme pour MLCP avec prise en compte de l'énergie en veille

Tout comme l'algorithme Adaptatif et l'algorithme Blacklist présenté au chapitre 3, l'algorithme Energy-blacklist [80] proposé dans ce chapitre est utilisé dans le cadre de l'approche réactive (figure 26). Rappelons que dans l'approche réactive, un couvrant est formé à la suite de la panne du couvrant précédent. L'approche réactive est un processus itératif dont la première étape consiste à invoquer un algorithme de construction de couvrant. Puis le couvrant formé est activé dans la seconde étape et la troisième étape contrôle l'énergie résiduelle des capteurs actifs afin de relancer la construction d'un nouveau couvrant si un capteur actif est à cours d'énergie ou en panne. Puisque les capteurs en veille consomment de l'énergie, nous proposons une version différente de

l'approche réactive où à l'étape 3, on reçoit non seulement les énergies résiduelles des capteurs actifs mais aussi celles des capteurs en veille. L'information à propos des énergies résiduelles de tous les capteurs est donc disponible au moment de construire le couvrant à l'étape 1. Cette collecte d'information est réalisable sur le plan pratique car les capteurs en veille échangent périodiquement avec la station de base. Le nouvel algorithme appelé Energy-blacklist, est invoqué à l'étape 1 de l'approche réactive.

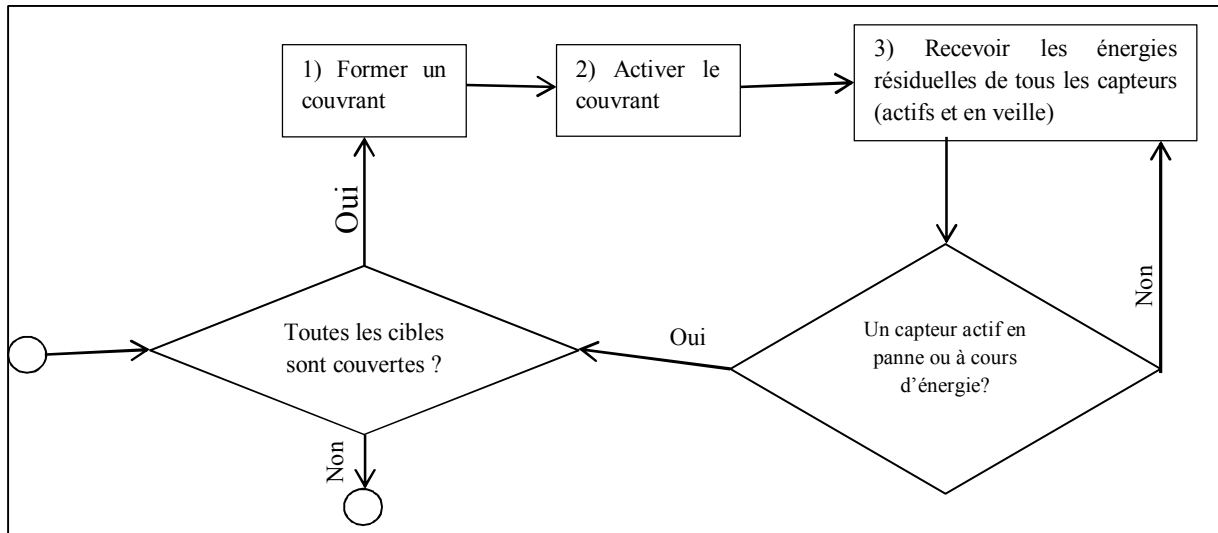


Figure 26 - Approche réactive

L'algorithme Energy-blacklist utilise l'idée de la liste noire comme l'algorithme Blacklist présenté dans le chapitre 3. La liste noire isole les capteurs qui couvrent les cibles critiques déjà couvertes par le couvrant en construction. Ce qui permet de réduire le nombre de capteurs qui couvrent chaque cible critique dans un couvrant. Contrairement à l'algorithme Blacklist qui définit une cible critique comme la cible couverte par le plus petit nombre de capteurs, l'algorithme Energy-blacklist définit la cible critique comme celle dont la durée de couverture maximale (donnée par la formule 1) est minimale. Par ailleurs, la notion de cible critique est statique car l'algorithme Energy-blacklist identifie les cibles critiques uniquement au premier appel (après le déploiement du réseau). L'ensemble des cibles critiques ne varie pas au fil des appels de l'algorithme contrairement aux algorithmes Blacklist et Adaptatif qui identifient les cibles critiques de façon dynamique (à chaque appel de l'algorithme). En plus de tenir compte de l'énergie résiduelle des capteurs pour identifier les cibles critiques, l'algorithme Energy-blacklist sélectionne aussi les capteurs en tenant compte de leurs énergies résiduelles. D'après la proposition 1, la durée de couverture d'une cible est maximale lorsque les capteurs qui la couvrent sont activés dans l'ordre croissant de leurs énergies. L'algorithme Energy-blacklist s'appuie sur cette proposition et sélectionne prioritairement les capteurs de plus faible énergie. Ainsi, l'algorithme Energy-blacklist forme des couvrants en se basant à la fois sur l'idée de la liste noire et sur la proposition 1. L'algorithme Energy-blacklist identifie une seule fois l'ensemble des cibles critiques. A chaque itération, il sélectionne le capteur hors de la liste noire dont l'énergie est minimale parmi les capteurs qui couvrent au moins une cible non couverte par le couvrant en construction. Si aucun capteur hors de la liste noire ne couvre de nouvelles cibles, un capteur est alors choisi en liste noire. Le capteur sélectionné en liste noire doit avoir la plus petite énergie parmi les capteurs qui couvrent au moins une nouvelle cible. S'il existe

plusieurs candidats (capteurs ayant une énergie minimale), le capteur qui couvre le plus de nouvelles cibles est choisi. Lorsqu'un capteur est sélectionné, l'algorithme met en liste noire tous les autres capteurs qui couvrent les mêmes cibles critiques que lui. Les capteurs sont sélectionnés itérativement jusqu'à ce que toutes les cibles soient couvertes. Lorsqu'un couvrant est formé, tous les capteurs redondants sont supprimés, s'il en existe. Rappelons qu'un capteur est redondant si toutes les cibles qu'il couvre sont couvertes par d'autres capteurs du couvrant.

L'algorithme Energy-blacklist est présenté à la figure 27. L'algorithme prend en entrée l'ensemble S des capteurs avec leurs énergies résiduelles et l'ensemble T des cibles et l'ensemble T_C des cibles critiques. Au premier appel de l'algorithme, l'ensemble T_C est vide. La première instruction (I1) vérifie que chaque cible de l'ensemble T est couverte par au moins un capteur de S . Si tel n'est pas le cas, l'algorithme retourne l'ensemble vide pour indiquer qu'il n'est pas possible de former un couvrant. Si l'ensemble T_C est vide, l'instruction I2 identifie les cibles critiques et les insère dans l'ensemble T_C . Les cibles critiques sont identifiées en utilisant la durée de couverture maximale de chaque cible donnée par la formule 1. L'instruction I3 initialise la liste noire et le couvrant à l'ensemble vide. La boucle *tantque* permet d'ajouter un capteur au couvrant à chaque itération. Cette boucle s'arrête lorsque toutes les cibles sont couvertes. L'instruction I4 choisit un capteur d'énergie minimale, hors de la liste noire et qui couvre au moins une nouvelle cible (critique ou non). Si aucun capteur en dehors de la liste noire ne couvre de nouvelles cibles, alors un capteur est choisi en liste noire selon le même critère. Le capteur choisi s_{min} est ajouté au couvrant (I5), toutes les cibles qu'il couvre sont supprimées de l'ensemble T (I6) et tous les autres capteurs qui couvrent les mêmes cibles critiques que s_{min} sont ajoutés à la liste noire (I7). Enfin, à l'instruction I8, le capteur s_{min} est supprimé de l'ensemble des capteurs et de la liste noire selon le cas. A la sortie de la boucle *tantque*, le couvrant est formé mais pourrait contenir des capteurs redondants. L'instruction I9 appelle la procédure ***suppression_redondance***(C, T) pour rechercher et supprimer d'éventuels capteurs redondants du couvrant. L'algorithme retourne le nouveau couvrant (I10) et se termine. La complexité de l'algorithme Energy-blacklist est en $O(mn)$ où m est le nombre de cibles et n est le nombre de capteurs.

Algorithme Energy-Blacklist (S, T, T_C)**Input:** T : Ensemble de cibles ; S : Ensemble de capteurs ; T_C : Ensemble des cibles critiques**Début**11. **Si** $\exists$ une cible $t \in T$ et $\nexists$ un capteur $s \in S$ tel que s couvre t , **alors Retourner** $\emptyset$ 12. **Si** $T_C = \emptyset$ **alors** Identifier les cibles critiques et les insérer dans l'ensemble T_C 13. Initialiser le couvrant et la liste noire B à l'ensemble vide : $B = \emptyset$; $C = \emptyset$ **Tant que** $T \neq \emptyset$ **Faire**14. **Si** $\exists s \in (S \setminus B)$ tel que s couvre une cible de T **Alors**Sélectionner le capteur $s_{min} \in (S \setminus B)$ qui couvre au moins une cible de T et dont l'énergie résiduelle est minimale**Sinon**Sélectionner le capteur $s_{min} \in B$ qui couvre au moins une cible de T et dont l'énergie résiduelle est minimale**Finsi**15. Ajouter le capteur s_{min} au couvrant : $C = C \cup \{s_{min}\}$ 16. Supprimer de T toutes les cibles couvertes par le capteur s_{min} 17. Ajouter à la liste noire (B) tous les capteurs qui couvrent les mêmes cibles critiques que le capteur s_{min} 18. Supprimer le capteur s de l'ensemble S et de l'ensemble B selon le cas**Fintantque**19. Appeler la procédure *suppression_redondance*(C, T) pour supprimer les capteurs redondants de C 110. **Retourner** le couvrant C **FinAlgo****Procédure *suppression_redondance* (C, T)****Input :** T : Ensemble de cibles ; C : Couvrant**Début****Pour tout** capteur $s \in C$ **Faire**Supprimer le capteur s du couvrant : $C = C \setminus \{s\}$ **Pour toute** cible t couverte le capteur s **Si** la cible t n'est pas couverte par le couvrant C **Alors** Ajouter le capteur s au couvrant C : $C = C \cup \{s\}$ **Finsi****Finpour****Finpour****Retourner** le couvrant C **FinProc**

Figure 27 - Algorithme Energy-Blacklist

5.4 Simulations et analyse des résultats

L'objectif de cette section est d'analyser les performances du nouvel algorithme Energy-blacklist et de comparer cet algorithme aux algorithmes Blacklist et Adaptatif du chapitre 3 et aux algorithmes Greedy-MS, Static-CCF, Dynamic CCF et MCS-based de la littérature. Les simulations réalisées dans ce chapitre sont similaires à celles du chapitre 3. Les heuristiques sont simulées sur des réseaux de capteurs déployés uniformément sur une surface pour couvrir un ensemble de cibles

elles-mêmes déployées uniformément sur la dite surface. La métrique utilisée pour évaluer la performance d'une solution est le ratio entre la durée de vie de la solution et la borne supérieure de la durée de vie du réseau. Cette borne supérieure est calculée à base de la formule 1, tel qu'expliqué dans la section II. Chaque algorithme est exécuté sur 100 réseaux différents et retourne 100 valeurs de ratio. Ces valeurs sont présentées sous la forme de boîte à moustache. La principale différence entre ces simulations et celles du chapitre 3 est que nous simulons aussi la consommation des capteurs en veille. Ainsi, tandis que l'énergie d'un capteur actif est décrémenté de $P_a \cdot \tau$ où τ est la durée du couvrant en cours d'activité, l'énergie d'un capteur en veille est décrémenté de $r \cdot P_a \cdot \tau$. Les valeurs du ratio r varient de 0 à 0.9. Lorsque $r = 0$, les capteurs en veille ne consomment pas d'énergie et les résultats obtenus permettent d'observer le comportement du nouvel algorithme dans ce contexte. Lorsque $r \neq 0$, nous pouvons observer les performances de l'algorithme Energy-blacklist et des autres lorsque les capteurs en veille consomment de l'énergie, sachant que plus r est grand et plus l'énergie consommée en veille est grande. Les paramètres des simulations sont donnés dans le tableau 3. Nous considérons une surface de déploiement carré de 10m x10m et des capteurs dont le rayon de sensibilité est égal à 1m. Le nombre de cibles déployées uniformément est de 50. Les réseaux simulés comptent 150 capteurs dans un cas et 300 capteurs dans l'autre. Enfin, nous supposons que l'énergie des capteurs suit une loi normale de moyenne 5085 mWh (qui correspond à l'énergie d'une pile alcaline jetable 9V de capacité typique 565 mAh et approximativement à l'énergie d'une pile Lithium-ion rechargeable 8.4V de capacité typique 600 mAh) et d'écart-type 500 dans un cas et 1000 dans l'autre.

Tableau 3 - Paramètres de simulation 2

Paramètre	valeur
Surface	10 x 10
Rayon de sensibilité	1
Nombre de cibles	50 (réparties uniformément sur la surface)
Nombre de capteurs	150 et 300 (déployés uniformément sur la surface)
Energie des capteurs	Loi normale de moyenne 5085 mWh et d'écart-type 500 et 1000
Ratio $r = \frac{P_v}{P_a}$	0 - 0.1 - 0.2 - 0.3 - 0.4 - 0.5 - 0.6 - 0.7 - 0.8 - 0.9

La figure 28 présente les performances des algorithmes pour des réseaux de 150 capteurs dont l'écart-type des énergies est 500 et la consommation des capteurs en veille est nulle ($r = 0$). Nous observons que l'algorithme Energy-blacklist donne la solution optimale dans 50% des cas et un ratio à l'optimum supérieur à 0.9 dans 75%. Les algorithmes Blacklist et Adaptatif atteignent l'optimum dans 75% des cas environ. Les anciens algorithmes (Greedy-MS, Static-CCF, Dynamic-CCF, et MCS-based algorithm) ont un ratio à l'optimum supérieur ou égal à 0.9 dans 50% des cas. Ceci montre que les nouveaux algorithmes (Adaptatif, Blacklist et Energy-blacklist) sont meilleurs que les algorithmes de la littérature lorsque $r = 0$. Cela s'explique par le fait que les anciens algorithmes ne tiennent pas compte des énergies résiduelles des capteurs après la fin de vie des couvrants qui ont été formés à l'étape 1 de l'approche classique. Tandis que l'approche réactive utilisée pour les trois nouveaux algorithmes permet d'exploiter ces énergies résiduelles. La figure 28 montre aussi que les algorithmes Blacklist et Adaptatif sont meilleurs que l'algorithme Energy-blacklist lorsque $r = 0$. Ce résultat est logique car l'algorithme Energy-blacklist se base sur la proposition 1 pour former les couvrants en choisissant le capteur d'énergie minimale. Or, lorsque

$r = 0$, la durée de couverture d'une cible ne change pas en fonction de l'ordre d'activation de ses capteurs. Ainsi, dans certains cas, le capteur sélectionné par l'algorithme à une itération donné n'est pas le plus approprié. Néanmoins, comme le montre la figure 28, l'algorithme Energy-blacklist a une très bonne performance (ratio à l'optimum supérieur à 0.9 dans 75% des cas).

La figure 29 présente les performances des algorithmes pour des réseaux de 150 capteurs dont l'écart-type des énergies est 500 et la consommation des capteurs en veille est égale à 10% de la consommation des capteurs actifs ($r = 0.1$). On observe sur cette figure que l'algorithme Energy-blacklist donne un ratio à l'optimum supérieur à 0.9 dans plus de 70% tandis que tous les autres algorithmes obtiennent un ratio à l'optimum supérieur à 0.9 dans environ 50% des cas. L'algorithme Energy-blacklist a donc une meilleure performance que les autres algorithmes lorsque $r = 0.1$. On constate aussi sur cette figure que les performances des algorithmes Blacklist et Adaptatif diminuent lorsque l'on passe de $r = 0$ à $r = 0.1$. Ce résultat s'explique par le fait que ces deux algorithmes ne prennent pas en compte l'énergie consommée par les capteurs en veille dans le choix des capteurs à activer. Car d'après la proposition 1, la durée de couverture d'une cible dépend de l'ordre dans lequel les capteurs qui la couvrent sont activés.

La figure 30 montre la performance des algorithmes pour des réseaux de 300 capteurs dont l'écart-type des énergies est 500 et la consommation des capteurs en veille est nulle ($r = 0$). On observe sur cette figure que l'algorithme energy-blacklist donne un ratio à l'optimum supérieur à 0.9 dans plus de 70% des cas. Les algorithmes Blacklist et Adaptatif atteignent l'optimum dans 75% environ. Les anciens algorithmes (Greedy-MS, Static-CCF, Dynamic-CCF, et MCS-based algorithm) quant à eux, ont un ratio à l'optimum supérieur à 0.8 dans 50% des cas. Cela montre que les deux algorithmes Blacklist et Adaptatif sont meilleurs que l'algorithme Energy-blacklist qui est lui-même meilleur que les anciens algorithmes lorsque $r = 0$.

La figure 31 présente les performances des algorithmes pour des réseaux de 300 capteurs dont l'écart-type des énergies est 500 et la consommation des capteurs en veille est égale à 10% de la consommation des capteurs actifs ($r = 0.1$). On observe sur cette figure que l'algorithme Energy-blacklist donne un ratio à l'optimum supérieur à 0.9 dans 75% des cas tandis que les algorithmes Blacklist et Adaptatif donnent un ratio à l'optimum supérieur à 0.8 dans 75% des cas et les anciens algorithmes ont un ratio à l'optimum supérieur à 0.8 dans 50% des cas. Cela montre que l'algorithme Energy-blacklist est meilleur que les autres algorithmes lorsque $r = 0.1$.

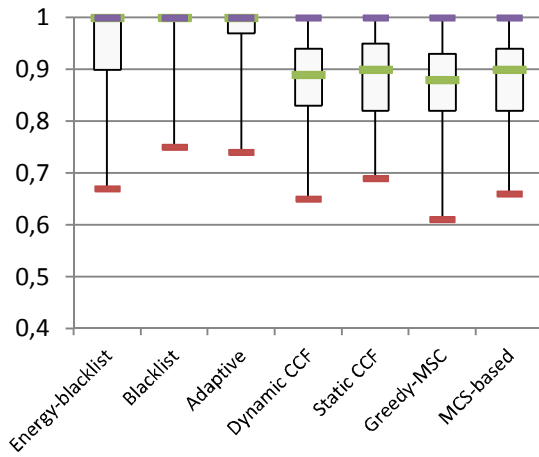


Figure 28 - Performance des algorithmes pour 150 capteurs, écart-type = 500 et $r=0$

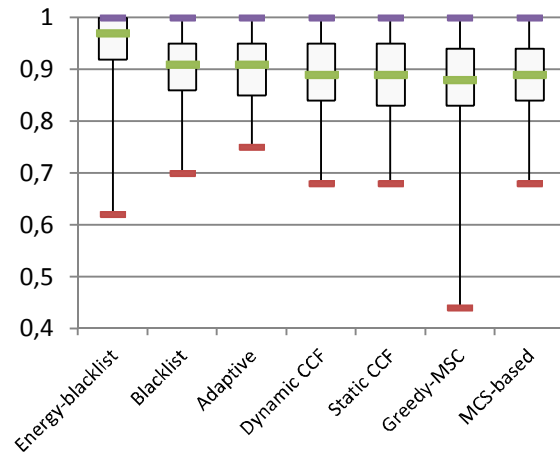


Figure 29 - Performance des algorithmes pour 150 capteurs, écart-type = 500 et $r=0.1$

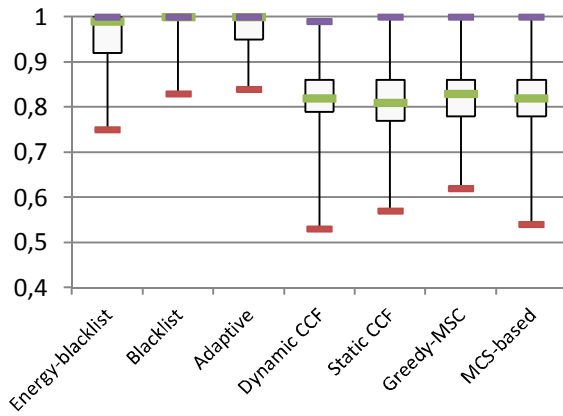


Figure 30 - Performance des algorithmes pour 300 capteurs, écart-type = 500 et $r=0$

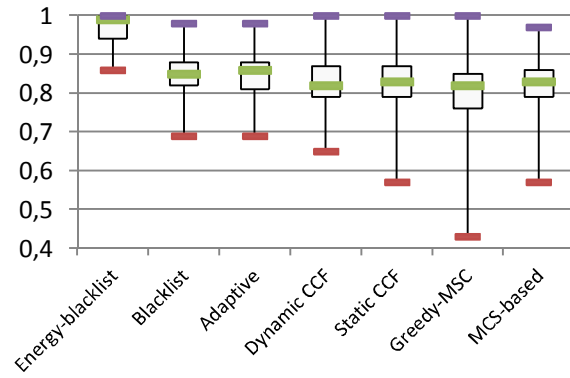


Figure 31 - Performance des algorithmes pour 300 capteurs, écart-type = 500 et $r=0.1$

La figure 32 présente les performances des différents algorithmes pour des réseaux de 150 capteurs dont l'écart-type des énergies est 1000 et la consommation des capteurs en veille est nulle ($r = 0$). Les algorithmes Blacklist et Adaptatif atteignent l'optimum dans 75% des cas, l'algorithme Energy-blacklist a un ratio supérieur à 0.9 dans plus de 75% des cas et les anciens algorithmes (Greedy-MSC, Static-CCF, Dynamic-CCF, et MCS-based algorithm) ont un ratio à l'optimum supérieur à 0.7 dans environ 50% des cas. Cela montre que les algorithmes Blacklist et Adaptatif sont meilleurs que l'algorithme Energy-blacklist qui lui-même est meilleur que les anciens algorithmes lorsque $r = 0$.

La figure 33 présente les performances des algorithmes pour des réseaux de 150 capteurs dont l'écart-type des énergies est 1000 et la consommation des capteurs en veille est égale à 10% de la consommation des capteurs actifs ($r = 0.1$). L'algorithme Energy-blacklist donne un ratio à l'optimum supérieur à 0.9 dans plus de 75% des cas, les algorithmes Blacklist et Adaptatif donnent un ratio à l'optimum supérieur à 0.9 dans 50% et les anciens algorithmes obtiennent un ratio à l'optimum supérieur à 0.7 dans 50% des cas environ. Ce résultat montre que l'algorithme Energy-blacklist est le meilleur lorsque $r = 0.1$.

La figure 34 montre la performance des algorithmes pour des réseaux de 300 capteurs dont l'écart-type des énergies est 1000 et la consommation des capteurs en veille est nulle ($r = 0$). Les algorithmes Blacklist et Adaptatif atteignent l'optimum dans 75% des cas environ, l'algorithme Energy-blacklist obtient un ratio supérieur à 0.9 dans plus de 75% des cas et les anciens algorithmes ont un ratio à l'optimum supérieur à 0.6 dans environ 50% des cas. On en déduit donc que les algorithmes Blacklist et Adaptatif sont meilleurs que l'algorithme Energy-blacklist qui lui-même est meilleur que les anciens algorithmes lorsque $r = 0$.

La figure 35 présente les performances des algorithmes pour des réseaux de 300 capteurs dont l'écart-type des énergies est 1000 et la consommation des capteurs en veille est égale à 10% de la consommation des capteurs actifs ($r = 0,1$). L'algorithme Energy-blacklist donne un ratio à l'optimum supérieur à 0.9, les algorithmes Blacklist et Adaptatif donnent un ratio à l'optimum supérieur à 0.8 dans 75% des cas et les anciens algorithmes donnent un ratio supérieur à 0.6 dans environ 75% des cas. Cela montre que l'algorithme Energy-blacklist est le meilleur lorsque $r = 0.1$.

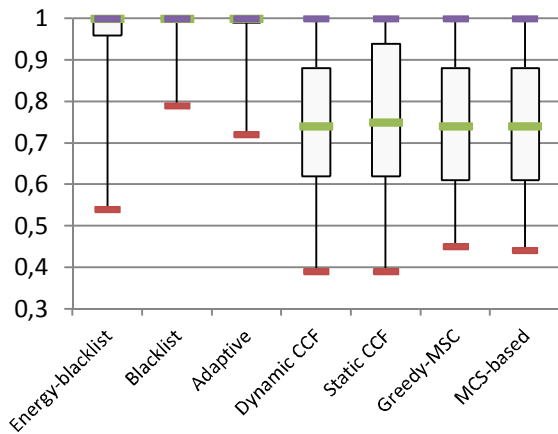


Figure 32 - Performance des algorithmes pour 150 capteurs, écart-type = 1000 et $r=0$

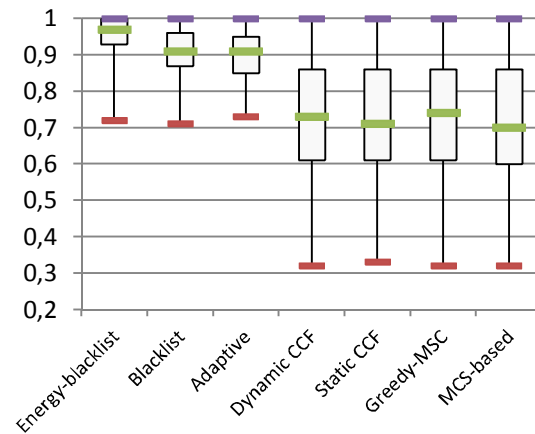


Figure 33 - Performance des algorithmes pour 150 capteurs, écart-type = 1000 et $r=0.1$

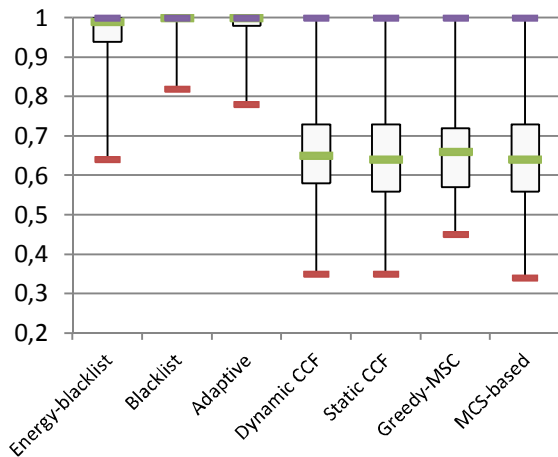


Figure 34 - Performance des algorithmes pour 300 capteurs, écart-type = 1000 et $r=0$

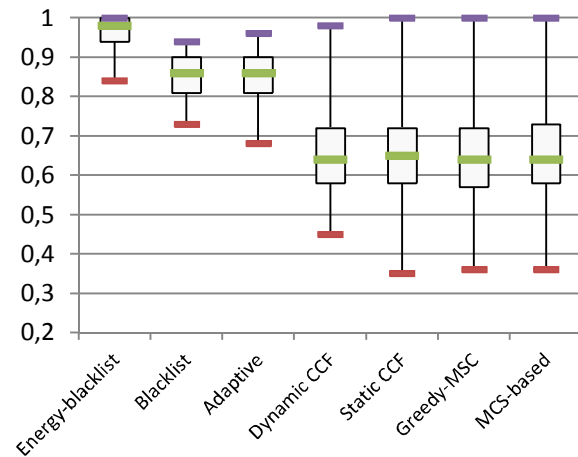


Figure 35 - Performance des algorithmes pour 300 capteurs, écart-type = 1000 et $r=0.1$

En comparant les figures ci-dessus, on aboutit aux observations ci-après :

1. L'algorithme Energy-blacklist donne un ratio à l'optimum supérieur à 0.9 dans au moins 75% des cas et atteint l'optimum dans environ 50% des cas, quel que soit la taille du réseau, la dispersion des énergies des capteurs et que la consommation en veille soit nulle ou pas. De plus, cet algorithme est le meilleur lorsque la consommation en veille est non nulle.
2. Les algorithmes Blacklist et Adaptatif sont les meilleurs lorsque la consommation en veille est nulle, quel que soit la taille du réseau et la dispersion des énergies des capteurs. Lorsque la consommation en veille n'est pas nulle, ces deux algorithmes sont meilleurs que les anciens algorithmes. Cependant, les performances de ces algorithmes diminuent lorsque l'énergie consommée en veille n'est plus nulle, notamment pour $r = 0.1$. Lorsque l'énergie consommée en veille est nulle, les performances des algorithmes Blacklist et Adaptatif ne varient pas en fonction de la taille du réseau, ni de la dispersion des énergies des capteurs. Ce qui n'est pas le cas lorsque la consommation en veille est non nulle. En effet, lorsque $r = 0.1$, les performances de ces deux algorithmes diminuent lorsque le nombre de capteurs augmente.
3. Pour un nombre de capteurs donné et pour une dispersion des énergies donnée, les anciens algorithmes (Greedy-MS, Static-CCF, Dynamic-CCF, et MCS-based algorithm) ont des performances qui ne varient presque pas lorsque r passe de 0 à 0.1. Par contre, pour une valeur donnée de r ($r = 0$ ou $r = 0.1$), les performances de ces algorithmes diminuent lorsque le nombre de capteurs augmente et lorsque la dispersion des énergies augmente.
4. Lorsque la consommation en veille est nulle, on aboutit aux mêmes observations faites au chapitre 3 concernant les algorithmes Blacklist, Adaptatif et les anciens algorithmes à savoir que les algorithmes Blacklist, Adaptatif sont meilleurs que les anciens algorithmes et que l'algorithme Blacklist donne des résultats légèrement supérieurs que l'algorithme Adaptatif.

Les figures ci-dessous présentent les performances des algorithmes Static-CCF, Dynamic-CCF, MCS-based, Greedy-MS, Adaptatif, Blacklist et Energy-blacklist pour des réseaux de 300 capteurs dont l'écart-type des énergies est 500 lorsque les valeurs de r varient entre 0 et 0.9. On observe sur ces figures que les performances des anciens algorithmes ne varient presque pas en fonction de l'énergie consommée par les capteurs en veille. Par contre les performances des algorithmes Blacklist et Adaptatif varient en fonction de l'énergie consommée en veille. Lorsque la consommation en veille est nulle, les deux algorithmes sont proches de l'optimum ou égaux à l'optimum dans plus 75% des cas. Leurs performances diminuent considérablement lorsque $r = 0.1$ et $r = 0.2$. A partir de $r = 0.3$, les performances de ces deux algorithmes augmentent progressivement avec en fonction de r . Lorsque $r = 0.9$, le ratio à l'optimum de ces algorithmes est supérieur à 0.9 dans 100% des cas. Cette variation s'explique par le fait que plus r est grand et plus l'énergie consommée par un capteur en veille est proche de celle consommée par un capteur actif. De ce fait, la quantité d'énergie perdue (à cause de la non prise en compte de la consommation en veille) diminue au fur et à mesure que r augmente. Le cas extrême (voire improbable) où les capteurs en veille consomment la même énergie que les capteurs actifs ($r = 1$) permet d'illustrer ce phénomène. Si $r = 1$, il est impossible de prolonger la durée de vie du réseau car un capteur en veille a la même durée de vie (ou dépense la même quantité d'énergie) qu'il soit en veille ou actif. Ainsi, activer tous les capteurs à la fois (c.-à-d. aucune optimisation de l'énergie) donne la solution

optimale tout comme activer successivement les couvrants jusqu'à ce que tous les capteurs soient à cours d'énergie. C'est pour cette raison que les algorithmes Blacklist et Adaptatif se rapprochent de l'optimum lorsque r se rapproche de 1. Cependant, cette variation n'existe pas dans le cas des anciens algorithmes car les couvrants sont formés en une seule étape et activer successivement. De ce fait, les capteurs sont activés uniquement pendant la durée spécifiée par l'algorithme et sont mis en veille le reste du temps. Illustrons cela par l'approche par couvrants disjoint, les capteurs n'appartenant à aucun couvrant sont mis en veille et ne sont jamais activés. De même, les capteurs résiduels des couvrants formés sont aussi mis en veille et ne sont plus activés. Etant donné que la valeur de r influence uniquement la durée de vie des capteurs qui sont mis en veille et activés/réactivés par la suite, l'effet de la variation de r est très peu visible sur la performance des anciens algorithmes.

Contrairement aux autres algorithmes, la figure 42 montre que la performance de l'algorithme Energy-blacklist augmente lorsque la valeur de r augmente. Quel que soit la valeur de r , l'algorithme Energy-blacklist donne un ratio à l'optimum supérieur à 0.9 dans plus de 75% des cas et atteint l'optimum dans environ 50%. A partir de $r = 0.5$, l'algorithme donne un ratio supérieur ou égal à 0.9 dans 100% des cas et lorsque $r = 0.9$, l'algorithme atteint l'optimum dans 100% des cas. En observant ces résultats de simulations, nous notons que l'algorithme Energy-blacklist est meilleur que les autres lorsque la consommation en veille est non nulle ($r \neq 0$).

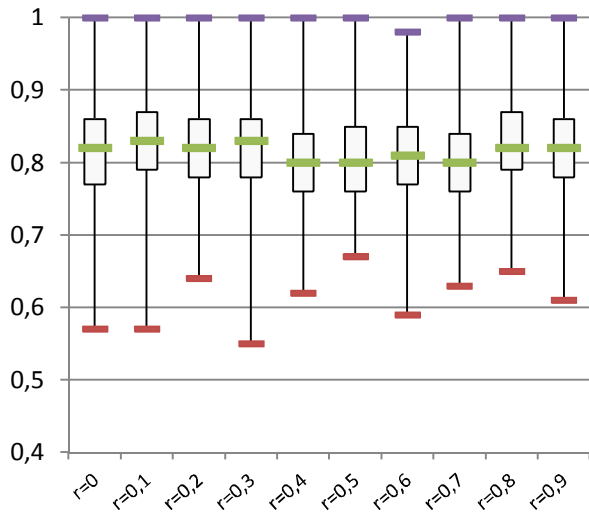


Figure 36 - Performance de Static-CCF Algorithm pour 300 capteurs et écart-type = 500

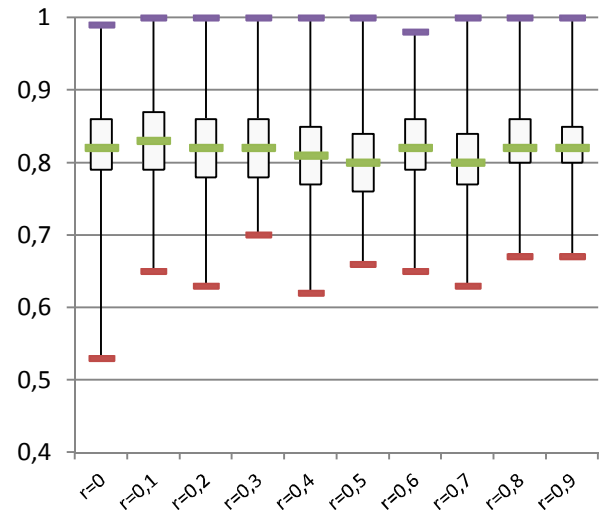


Figure 37 - Performance de Dynamic-CCF Algorithm pour 300 capteurs et écart-type = 500

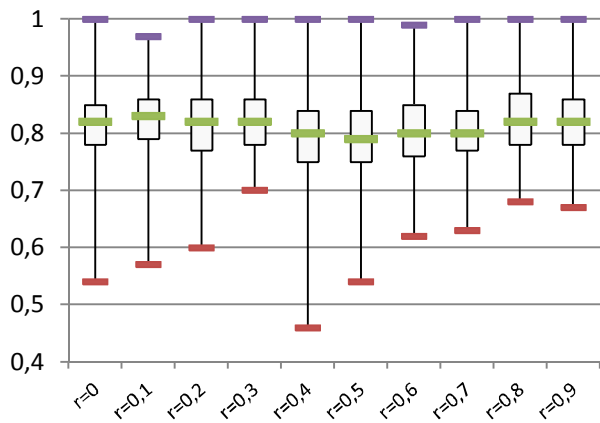


Figure 38 - Performance de MCS-based Algorithm pour 300 capteurs et écart-type = 500

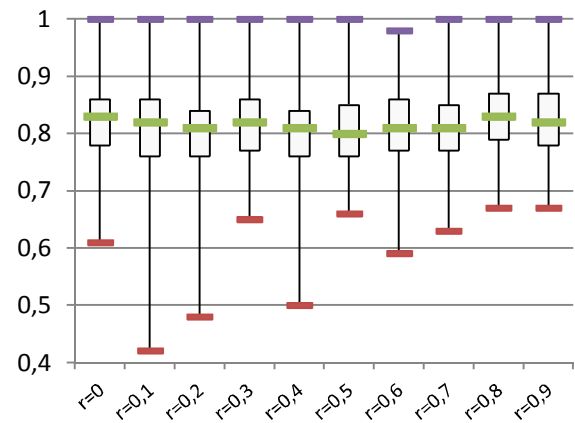


Figure 39 - Performance de Greedy-MSC Algorithm pour 300 capteurs et écart-type = 500

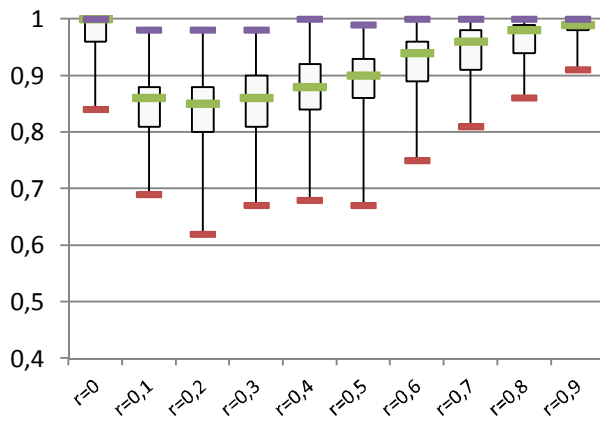


Figure 40 - Performance de Adaptive Algorithm pour 300 capteurs et écart-type = 500

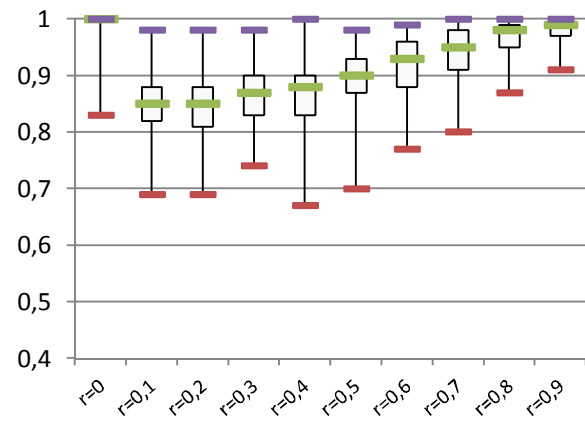


Figure 41 - Performance de Blacklist Algorithm pour 300 capteurs et écart-type = 500

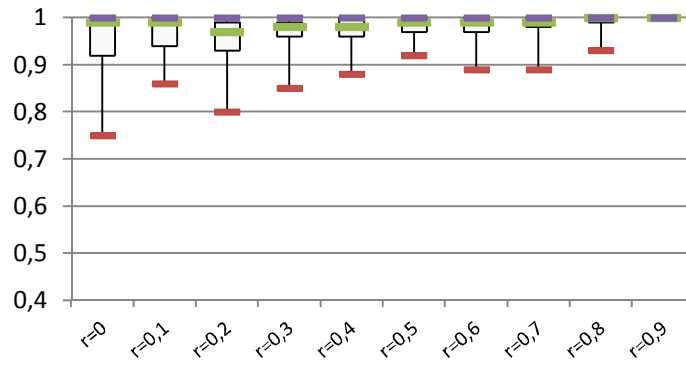


Figure 42 - Performance de Energy-Blacklist Algorithm pour 300 capteurs et écart-type = 500

5.5 Conclusion

Dans ce chapitre, nous avons montré que dans le MLCP, il n'est pas réaliste de supposer que l'énergie consommée par les capteurs en veille est nulle. Car les capteurs dits « en veille » sont en réalité des capteurs ayant une activité réduite par rapport à l'activité des capteurs dits « actifs ». Nous avons donc introduit la consommation en veille dans le MLCP en considérant que le ratio entre l'énergie consommée par un capteur en veille et l'énergie consommée par un capteur actif est $r = \frac{P_v}{P_a}$ où P_v et P_a sont les puissances moyennes respectives d'un capteur en veille et d'un capteur actif, tel que $r \in [0; 1]$. Nous avons ensuite donné une proposition (proposition 1) utile pour déterminer une borne supérieure de la durée de vie optimale du réseau. Cette proposition stipule que la durée de couverture maximale d'une cible est atteinte lorsque les capteurs qui la couvrent sont activés dans l'ordre croissant de leurs énergies. La borne supérieure obtenue à partir de cette proposition permet non seulement d'évaluer la qualité d'une solution d'une instance du MLCP, mais aussi d'identifier les cibles critiques car une cible critique n'est pas nécessairement celle couverte par le plus petit nombre de capteurs. Nous avons par la suite proposé un nouvel algorithme, l'algorithme Energy-blacklist, qui tient compte de l'énergie consommée par les capteurs en veille dans la construction des couvrants. Cet algorithme est invoqué à l'étape 1 de l'approche réactive qui dans ce cas, collecte les énergies résiduelles de tous les capteurs (actifs et en veille). L'algorithme Energy-blacklist se base sur la proposition 1 et sur l'idée de la liste noire utilisée premièrement par l'algorithme Blacklist. D'autre part, cet algorithme utilise la notion de cibles critiques statiques et identifie les cibles critiques une seule fois en tenant compte du nombre de capteurs qui les couvrent et aussi des énergies de ces capteurs. Nous avons réalisé des simulations afin d'évaluer la performance de l'algorithme Energy-blacklist et de comparer cet algorithme aux algorithmes Blacklist et Adaptatif, de même qu'aux algorithmes Static-CCF, Dynamic-CCF, MCS-based, Greedy-MSC de la littérature. Les résultats des simulations ont montré que l'algorithme Energy-blacklist est meilleur que les autres algorithmes lorsque la consommation des capteurs en veille est non nulle. De plus, l'algorithme Energy-blacklist donne un ratio à l'optimum supérieur à 0.9 dans 75%, quels que soient l'énergie consommée en veille, le nombre de capteurs du réseau et la dispersion des énergies de ces capteurs. Nous avons aussi constaté que les algorithmes Blacklist et Adaptatif sont meilleurs que les anciens algorithmes, quelle que soit l'énergie consommée en veille et ils donnent un ratio à l'optimum supérieur à 0.8 dans 75% des cas lorsque la consommation

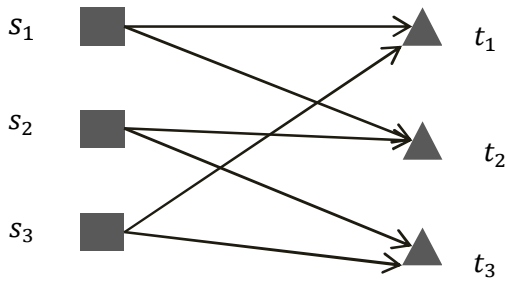
d'énergie en veille est non nulle. Les trois nouveaux algorithmes Energy-blacklist, Adaptatif et Blacklist sont des algorithmes basés sur l'approche réactive. Dans cette approche, un nouveau couvrant est formé uniquement lorsque le couvrant en cours est défaillant. Bien que cette approche donne de bons résultats dans la plupart des cas comme le montrent les simulations, il existe cependant une famille d'instances du MLCP pour laquelle l'approche réactive n'est pas aussi efficace : il s'agit de la famille des réseaux en anneau impair. Dans la suite, nous définissons la famille des réseaux en anneau impair et nous proposons des solutions de meilleure qualité que celles données par les algorithmes basés sur l'approche réactive.

Chapitre 6 Le MLCP dans le cas des réseaux réguliers

6.1 Introduction

L'approche réactive, introduite au chapitre 2, est une technique qui consiste à construire un nouveau couvrant uniquement lorsque le couvrant en cours d'activité est défaillant suite à l'épuisement de la batterie d'un capteur actif ou à une panne d'un capteur actif. L'avantage de cette approche est qu'elle permet d'exploiter les énergies résiduelles des capteurs pour former davantage de couvrants. L'approche réactive diffère ainsi de l'approche classique par couvrants disjoints ou non disjoints où les couvrants sont formés en une étape et les énergies résiduelles des capteurs à la fin de l'activation de ces couvrants ne sont pas prises en compte. Les simulations réalisées dans les chapitres 3 et 4 ont montré que les algorithmes basés sur l'approche réactive à savoir Energy-blacklist, Blacklist et Adaptatif, sont plus performants que les algorithmes de la littérature basés sur l'approche classique. Cependant, l'approche réactive n'est pas adaptée pour certaines instances du MLCP comme le montre l'exemple ci-après.

Considérons l'instance du MLCP de la figure 43 où le réseau est formé de trois capteurs s_1, s_2, s_3 et de trois cibles t_1, t_2, t_3 tel que le capteur s_1 couvre les cibles t_1, t_2 , le capteur s_2 couvre les cibles t_2, t_3 et le capteur s_3 couvre les cibles t_1, t_3 . Les énergies respectives des capteurs s_1, s_2, s_3 sont $E_1 = 10$; $E_2 = 8$ et $E_3 = 9$. La puissance d'un capteur actif est $P_a = 1$ et la consommation d'énergie d'un capteur actif est nulle ($r = 0$). Sur cette instance, les algorithmes Energy-blacklist, Blacklist et Adaptatif forment deux couvrants successifs $C_1 = \{s_1, s_2\}$ et $C_2 = \{s_1, s_3\}$ de durées respectives $Lt_1 = 8$ et $Lt_2 = 2$. Cette solution donne une durée de vie du réseau $\Delta_1 = 10$. Or, la solution S_2 formée des couvrants successifs $C_1 = \{s_1, s_2\}$ de durée $Lt_1 = 4$, $C_2 = \{s_2, s_3\}$ de durée $Lt_2 = 4$, $C_3 = \{s_1, s_3\}$ de durée $Lt_3 = 5$, donne une durée de vie $\Delta_2 = 13$. La solution S_2 donne donc une durée de vie du réseau supérieure de 30% à celle donnée par les algorithmes basés sur l'approche réactive. On observe que, contrairement à l'approche réactive, les couvrants sont remplacés avant l'épuisement de la batterie d'un capteur dans la solution S_2 . Cet exemple montre que pour certaines instances difficiles du MLCP, l'approche réactive donne des solutions sous-optimales. Dans la suite, nous définissons une famille d'instances difficiles, à savoir la famille des réseaux en anneau impair qui est une sous-classe des réseaux réguliers, et nous proposons une nouvelle heuristique qui trouve de meilleures solutions pour les instances de cette famille que celles données par les algorithmes basés sur l'approche réactive.



Ensemble de capteurs $S = \{s_1, s_2, s_3\}$; Ensemble de cibles $T = \{t_1, t_2, t_3\}$

Relation de couverture : $s_1 = \{t_1, t_2\}$; $s_2 = \{t_1, t_3\}$; $s_3 = \{t_2, t_3\}$

$E_1 = 10$; $E_2 = 8$; $E_3 = 9$; $r = 0$; $P_a = 1$

Figure 43 - Une instance difficile du MLCP

6.2 Famille des réseaux en anneau impair: exemple d'instances difficiles

La famille $\mathcal{F}$ des réseaux en anneau impair est définie ainsi qu'il suit.

Définition 1

Une instance $\mathcal{F}n$ de $\mathcal{F}$, avec $n = 2p + 1$ ($p \in \mathbb{N}^*$), est un réseau de n capteurs $s_1, \dots, s_n$ et de n cibles $t_1, \dots, t_n$ reliés les uns aux autres comme illustré à la figure suivante. Dans ce type de réseaux, chaque capteur couvre exactement deux cibles et chaque cible est couverte exactement par deux capteurs.

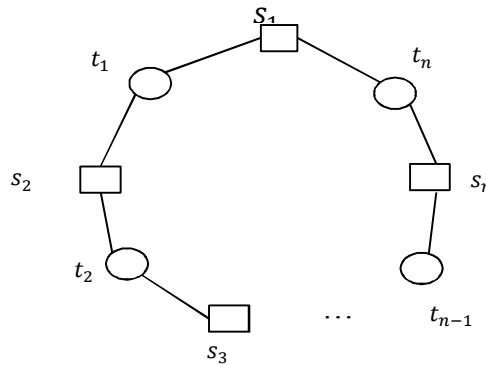


Figure 44 - Une instance de la famille $\mathcal{F}$

Il est important de noter qu'un réseau en anneau pair (c.-à-d. $n = 2p, p \in \mathbb{N}^*$) n'est pas une instance difficile du MLCP. Lorsque les capteurs ont la même énergie initiale E , la solution optimale est formée de deux couvrants disjoints, le premier constitué des capteurs d'indices pairs et le second constitué des capteurs d'indices impairs. Le premier couvrant est activé pendant $\frac{E}{P_a}$ unités

de temps tandis le second est activé pendant $\frac{E(1-r)}{P_a}$ unités de temps. La durée de vie optimale du réseau est donc égale à $(2-r) * \frac{E}{P_a}$.

La figure 45 montre une instance de la famille $\mathcal{F}$ où $n = 3$. Cette figure est aussi une autre représentation de l'exemple de la figure 43 ci-dessus. Lorsque les capteurs ont une énergie $E = 1$, une puissance en mode actif $P_a = 1$ et une consommation en veille nulle ($r = 0$), la solution optimale de cette instance du MLCP a une durée de vie du réseau égale à $\frac{3}{2}$ et est formée de trois couvrants $C_1 = \{s_1, s_2\}$, $C_2 = \{s_2, s_3\}$ et $C_3 = \{s_1, s_3\}$, de durée d'activation $\frac{1}{2}$ chacun. En effet, la solution optimale consomme 2 unités d'énergie (pour 2 capteurs actifs) pendant au moins 1 unité de temps, c'est-à-dire avant que le premier capteur épuise sa batterie. Alors, il reste en tout au plus 1 unité d'énergie pour activer 2 capteurs. La seconde étape dure au plus $\frac{1}{2}$ unité de temps. Ainsi, la durée de vie du réseau $\frac{3}{2}$ est optimale. Les algorithmes de la littérature basés sur l'approche classique par couvrants non disjoints (Greedy-MS, Static-CCF et Dynamic-CCF) trouvent cette solution optimale lorsque le paramètre w est égal à 2 (w spécifie le nombre maximal de couvrants auxquels peut appartenir un capteur). Cependant, les auteurs de ces algorithmes n'expliquent pas comment déterminer la valeur appropriée de w . Pour cela, nous proposons une approche systématique pour trouver une solution pour toute instance de la famille $\mathcal{F}$ lorsque les capteurs ont la même énergie E et la consommation d'énergie en veille est nulle ($r = 0$). Soit une instance $\mathcal{F}_n$ de la famille $\mathcal{F}$ avec $n = 2p + 1$, il y'a n couvrants non disjoints de taille minimale et chaque capteur appartient à $p + 1$ d'entre eux. Si tous les n couvrants sont activés, alors chaque capteur appartient à $p + 1$ couvrants. En posant $w = p + 1$, chaque couvrant consomme $\frac{E}{p+1}$ unités d'énergie de chaque capteur. Alors, la durée d'activation de chaque couvrant est $\frac{1}{p+1} * \frac{E}{P_a}$ et la durée de vie du réseau de cette solution est $\frac{n}{p+1} * \frac{E}{P_a}$. On note que pour l'instance où $n = 3$, cette approche donne $w = 2$ et cela correspond à la valeur choisie par les auteurs des algorithmes basés sur l'approche classique par couvrants non disjoints pour cette instance.

Lorsque les capteurs ont la même énergie E et l'énergie consommée en veille n'est pas nulle ($r \neq 0$), la proposition suivante donne la solution optimale de toute instance de la famille $\mathcal{F}$ où $n = 3$ (Fig. 5.3).

Proposition 2

*La solution optimale de toute instance de la famille $\mathcal{F}$ où $n = 3$ et les capteurs ont la même énergie E , a une durée de vie du réseau égale à $\frac{3-r}{2} * \frac{E}{P_a}$ et est constituée des couvrants $C_1 = \{s_1, s_2\}$, $C_2 = \{s_2, s_3\}$ et $C_3 = \{s_1, s_3\}$ de durées d'activation respectives $Lt_1 = \frac{1}{2} * \frac{E}{P_a}$, $Lt_2 = \frac{1}{2} * \frac{E}{P_a}$, et $Lt_3 = \frac{1-r}{2} * \frac{E}{P_a}$.*

Preuve

Notons x la durée pendant laquelle deux capteurs sont actifs et un est en veille, et y la durée pendant laquelle deux capteurs sont actifs et un capteur a épuisé son énergie. L'objectif est de

maximiser la durée de vie du réseau $x + y$ en respectant les contraintes suivantes :

- $x \geq \frac{E}{P_a}$ car il faut au moins $\frac{E}{P_a}$ unités de temps pour épuiser une batterie
- $2xP_a + xrP_a + 2yP_a \leq 3E$ car l'énergie totale disponible dans les trois capteurs est $3E$.

La solution de ce problème de programmation linéaire peut être obtenue graphiquement et est

$x = \frac{E}{P_a}$, $y = \frac{E}{2P_a}(1 - r)$. Cela correspond à la durée de vie $\frac{3-r}{2} * \frac{E}{P_a}$ de la solution ci-dessus. Cette solution est donc optimale.

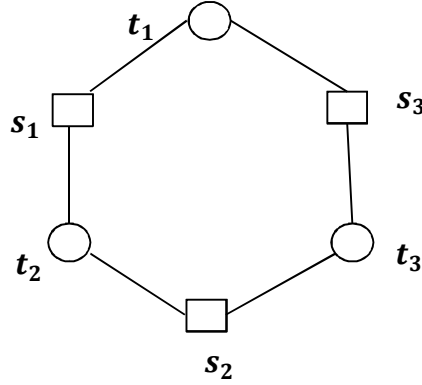


Figure 45 - Une instance de la famille $\mathcal{F}$ où $n=3$

La famille $\mathcal{F}$ est une sous-classe de la famille des réseaux réguliers que nous définissons ci-dessous. Soit $G = (S \cup T, U)$ le réseau dont l'ensemble des sommets est formé des ensembles S des capteurs et T des cibles et l'arête $(s, t) \in U$ si le capteur s couvre la cible t .

Définition 2

Un automorphisme de G est une permutation σ de $S \cup T$, tel que $(s, t) \in U$ si et seulement si $(\sigma(s), \sigma(t)) \in U$. En d'autres termes, un automorphisme est une permutation de l'ensemble des sommets qui préserve l'ensemble des arêtes.

Définition 3

G est dit régulier si tout capteur peut être transformé en n'importe quel autre capteur par un automorphisme. En d'autres termes, pour tout couple (s, s') de capteurs, il existe un automorphisme σ tel que $\sigma(s) = s'$.

Définissons la notion de résidu d'un capteur comme suit.

Définition 4

Le résidu d'un capteur s_i correspondant à un algorithme pour une instance du MLCP est la quantité d'énergie résiduelle de s_i à la fin de l'exécution de l'algorithme.

Les réseaux réguliers dont les capteurs ont la même énergie ont la propriété fondamentale énoncée par la proposition 3.

Proposition 3

Tout réseau régulier dont les capteurs ont la même énergie admet une solution optimale de résidu nul.

Preuve

Démontrons ce résultat par l'absurde. Supposons qu'un algorithme optimal A de durée de vie maximale Δ_{opt} sur un réseau régulier G , a un résidu non nul e sur le capteur s^* . Considérons les automorphismes $\sigma^{(j)}$, $j = 1, \dots, n$ tel que $\sigma^{(j)}(s^*) = s_j$. Soient $A^{(1)}, A^{(2)}, \dots, A^{(n)}$ les algorithmes obtenus à partir de A en appliquant les automorphismes $\sigma^{(j)}$, $j = 1, \dots, n$ et en divisant les énergies de chaque capteur par $w = n$. Alors, chaque algorithme $A^{(j)}$ admet un résidu $\frac{e}{n}$ sur le capteur s_j (car $s_j = \sigma^{(j)}(s^*)$ et A laisse un résidu e sur le capteur s^*). Il est donc possible de former un nouveau couvrant et la durée de vie du réseau de la nouvelle solution est $\Delta_{opt} + \frac{e}{P_a * n}$, où P_a est la puissance consommée en mode actif. Cela conduit à une contradiction puisque Δ_{opt} est la durée de vie maximale du réseau. Cela montre que toute solution optimale d'un réseau régulier dont les capteurs ont la même énergie, a un résidu nul.

Considérons un algorithme A défini sur un réseau régulier et associé à un couvrant C tel qu'il existe un seul capteur s^* avec un résidu non nul. Comme ci-dessus, il est possible en utilisant les automorphismes appropriés, de générer $|C|$ couvrants de durées d'activation $\frac{1}{|C|} * \frac{E}{P_a}$ tel que chaque couvrant C_i produit un résidu sur un capteur différent de C . Ainsi, tous les capteurs du couvrant C ont un résidu $\frac{E}{|C|}$ et forment un couvrant supplémentaire. Nous obtenons ainsi une solution à résidu nul de durée de vie du réseau égale à $\frac{|C|+1}{|C|} * \frac{E}{P_a}$. Cette méthode peut s'appliquer sur l'instance de la famille $\mathcal{F}$ où $n = 3$, $E = 1$, $P_a = 1$ et $r = 0$ (Fig. 5.4) de la façon suivante. Soit $C = \{s_1, s_2\}$ le couvrant formé par l'algorithme A . Cet algorithme laisse un résidu égal à 1 sur le capteur s_3 . Par l'automorphisme $\sigma^{(1)}$ qui transforme s_1 en s_2 , s_2 en s_3 et s_3 en s_1 , on génère le couvrant $C_1 = \{s_2, s_3\}$ de durée d'activation $\frac{1}{|C|} * \frac{E}{P_a} = \frac{1}{2}$ qui laisse un résidu $\frac{E}{|C|} = \frac{1}{2}$ sur le capteur s_1 . Puis, par l'automorphisme $\sigma^{(2)}$ qui transforme s_1 en s_3 , s_2 en s_1 et s_3 en s_2 , on génère le couvrant $C_2 = \{s_3, s_1\}$ de durée d'activation $\frac{1}{|C|} * \frac{E}{P_a} = \frac{1}{2}$ qui laisse un résidu $\frac{E}{|C|} = \frac{1}{2}$ sur le capteur s_2 . Tous les capteurs de C à savoir s_1 et s_2 ont donc un résidu $\frac{1}{2}$ et forment un nouveau couvrant $C_3 = \{s_1, s_2\}$. La solution donnée par cette méthode est formée de trois couvrants $C_1 = \{s_2, s_3\}$, $C_2 = \{s_3, s_1\}$ et $C_3 = \{s_1, s_2\}$ de durées d'activation $\frac{1}{2}$ chacun. Nous avons montré précédemment que cette solution est optimale pour cette instance du MLCP. L'inconvénient de cette méthode est qu'elle ne s'applique qu'aux réseaux réguliers pour lesquels un algorithme laisse un seul capteur avec un résidu non nul. Dans la suite, nous proposons une méthode similaire à celle-ci et qui s'applique à toute instance de la famille $\mathcal{F}$ où les capteurs ont la même énergie.

6.3 Heuristique pour la famille des réseaux en anneau impair

Dans cette section, nous proposons une méthode analytique basée sur les automorphismes et sur la proposition 3, qui permet de trouver une solution pour toute instance de la famille $\mathcal{F}$ dont les

capteurs ont la même énergie initiale et ce, quelle que soit la consommation en veille ($0 \leq r < 1$). Etant donné que l'hypothèse d'énergie identique pour tous les capteurs n'est pas réaliste, nous présentons par la suite une heuristique pour toute instance de la famille $\mathcal{F}$, quelles que soient les énergies initiales des capteurs. Cette heuristique est basée sur la méthode analytique.

6.3.1 Méthode analytique de résolution pour les capteurs de même énergie

Soient :

$\mathcal{F}n$, ($n = 2p + 1$) une instance de la famille $\mathcal{F}$,

E l'énergie initiale des capteurs,

P_a la puissance consommée et

$r \in [0; 1[$, le ratio entre la puissance consommée en veille et la puissance consommée en mode actif.

Définissons l'automorphisme σ ainsi qu'il suit:

$$\begin{cases} \sigma(s_n) = s_1 \\ \sigma(s_i) = s_{i+1} \text{ pour } i = 1, \dots, n-1 \end{cases}$$

Soit $C_i = \{s_{i1}, s_{i2}, s_{i3}, \dots, s_{il}\}$ un sous-ensemble de capteurs, définissons la fonction f telle que si $f(C_i) = C_j$ alors $C_j = \{\sigma(s_{i1}), \sigma(s_{i2}), \sigma(s_{i3}), \dots, \sigma(s_{il})\}$.

Considérons les couvrants $C_1, C_2, \dots, C_n$ tel que $C_1 = \{s_1, s_3, s_5, \dots, s_{n-1}\}$ est de taille minimale et $C_{j+1} = f(C_j)$, pour $j = 1, \dots, n-1$. Notons Lt_j la durée d'activation du couvrant C_j et posons $Lt_j = \gamma_j * \frac{E}{P_a}$, avec $\gamma_j \in \mathbb{R}^*$, $j = 1, \dots, n$. Les couvrants sont activés dans l'ordre $C_1, C_2, \dots, C_n$.

D'après la proposition 3, l'instance $\mathcal{F}n$ admet une solution optimale de résidu nul. En se basant sur cette proposition, l'idée de cette méthode est de trouver une solution de résidu nul car toute solution de $\mathcal{F}n$ avec un résidu non nul est nécessairement sous-optimale. Pour cela, nous cherchons une solution où chaque capteur a une énergie nulle après l'activation du dernier couvrant auquel il appartient. Pour chaque capteur s_i , notons K^i l'indice du dernier couvrant auquel il appartient. Alors, l'énergie totale consommée par le capteur s_i est $P_a * \sum_{j=1}^{K^i} Lt_j * (\delta_{ij} + r * \overline{\delta_{ij}})$, où $\delta_{ij} = 1$ si le capteur s_i appartient au couvrant C_j et $\delta_{ij} = 0$ sinon. En effet, l'énergie consommée par le capteur s_i durant l'activation du couvrant C_j est égale à $P_a * Lt_j$ si le capteur s_i appartient au couvrant C_j et est égale à $r * P_a * Lt_j$ sinon. En supposant que tout capteur s_i a une énergie nulle au terme de l'activation du dernier couvrant auquel il appartient, on obtient le système d'équation suivant :

$$\begin{cases} \sum_{j=1}^{K^1} \gamma_j * (\delta_{1j} + r * \overline{\delta_{1j}}) = 1 \\ \sum_{j=1}^{K^2} \gamma_j * (\delta_{2j} + r * \overline{\delta_{2j}}) = 1 \\ \vdots \\ \sum_{j=1}^{K^n} \gamma_j * (\delta_{nj} + r * \overline{\delta_{nj}}) = 1 \end{cases} \quad (1)$$

Une solution $(\gamma_1, \gamma_2, \dots, \gamma_n)$ de ce système d'équations (1) est:

$$\begin{cases} \gamma_j = \frac{1}{(p+1)+r*(p-1)} \text{ pour } j = 1, \dots, n-1 \\ \gamma_n = \frac{1-r}{(p+1)+r*(p-1)} \end{cases} \quad (S_1)$$

Preuve

Soit $\alpha = \frac{1}{(p+1)+r*(p-1)}$. Chaque capteur appartient à $p+1$ couvrants. Le dernier couvrant auquel appartient un capteur est soit C_{n-1} ou C_n . Si le dernier couvrant contenant un capteur est C_{n-1} , alors après l'activation de C_{n-1} , ce capteur a consommé l'énergie $\alpha((p+1)+r*(p-1)) * E = E$ et son énergie résiduelle est donc nulle après sa dernière activation. Si le dernier couvrant auquel appartient un capteur est C_n , alors l'énergie consommée par ce capteur après l'activation de C_n est $\alpha(p+(1-r)+r*p) * E = E$, son énergie résiduelle est donc nulle après sa dernière activation.

La durée de vie de cette solution est $\Delta = \frac{n-r}{(p+1)+r*(p-1)} * \frac{E}{P_a}$. Cette solution généralise celle de l'instance de la famille $\mathcal{F}$ où $n = 3$. D'après la proposition 2, la solution obtenue par cette méthode est optimale pour $n = 3$. Nous conjecturons que cette méthode donne une solution optimale pour toute instance $\mathcal{F}n$, $n = 2p+1$, de la famille $\mathcal{F}$ où les capteurs ont la même énergie initiale.

Considérons une autre méthode utilisant l'automorphisme σ et la fonction f définis ci-dessus. En se basant sur la proposition 3, cherchons une solution où chaque capteur a un résidu nul après l'activation du dernier couvrant, c'est-à-dire le couvrant C_n . Alors, on obtient le système d'équations (2) ci-après :

$$\begin{cases} \sum_{j=1}^n \gamma_j * (\delta_{1j} + r * \overline{\delta_{1j}}) = 1 \\ \sum_{j=1}^n \gamma_j * (\delta_{2j} + r * \overline{\delta_{2j}}) = 1 \\ \vdots \\ \sum_{j=1}^n \gamma_j * (\delta_{nj} + r * \overline{\delta_{nj}}) = 1 \end{cases} \quad (2)$$

La solution $(\gamma_1, \gamma_2, \dots, \gamma_n)$ du système d'équations (2) est: $\gamma_j = \frac{1}{(p+1)+r*p}$ pour $j = 1, \dots, n$. En effet, chaque capteur appartient à $p+1$ couvrants et n'appartient pas à p couvrants. La durée de vie de cette solution est $\Delta' = \frac{n}{(p+1)+r*p} * \frac{E}{P_a}$.

En comparant ces deux solutions, nous observons que $\Delta = \Delta'$ pour $r = 0$, alors que $\Delta > \Delta'$ pour $0 < r < 1$. Cela signifie que les deux solutions retournent la même durée de vie du réseau lorsque l'énergie consommée en veille est nulle tandis que la première solution est meilleure que la seconde lorsque l'énergie consommée en veille est non nulle. Ce résultat est logique car le temps total que les capteurs passent en mode veille est plus grand dans la seconde solution que dans la première. Ainsi, lorsque l'énergie consommée en veille est non nulle, ce temps supplémentaire passé en veille réduit la durée de vie du réseau de la seconde solution. Rappelons que d'après la proposition 3, toute solution optimale a un résidu nul mais toute solution dont le résidu est nul n'est pas nécessairement optimale. Bien que les deux solutions sont à résidu nul, nous avons conjecturé que la première solution est optimale pour toute instance $\mathcal{F}n$ où les capteurs ont la même énergie initiale

et nous constatons que la seconde solution est sous-optimale pour $0 < r < 1$.

Notons Δ'' la durée de vie de la solution obtenue par les algorithmes basés sur l'approche réactive (Energy-Blacklist, Blacklist et Adaptatif). En comparant cette durée de vie à celle de la première solution basée les automorphismes (Δ), nous notons que $\Delta = \frac{n-r}{(p+1)+r*(p-1)} \Delta''$, car $\Delta'' = \frac{E}{P_a}$. Sachant que $\frac{n-r}{(p+1)+r*(p-1)} = 1 + g_r(p)$ où $\frac{1-r}{2} \leq g_p(r) \leq \frac{1-r}{1+r}$, nous déduisons que la première solution basée sur les automorphismes est meilleure que la solution basée sur l'approche réactive ($\Delta \geq \Delta''$) et le gain obtenu par la première solution basée sur les automorphismes est supérieur ou égal à $\frac{1-r}{2}$ fois la durée de vie de la solution basée sur l'approche réactive.

La méthode analytique basée sur les automorphismes et sur la proposition 3 tel que chaque capteur ait un résidu après l'activation du dernier couvrant auquel il appartient, est donc meilleure que les algorithmes basés sur l'approche réactive. Cependant, cette méthode ne s'applique qu'aux instances de la famille $\mathcal{F}$ où tous les capteurs ont la même énergie initiale. Or, nous avons montré dans les chapitres précédents, qu'il n'est réaliste de supposer que les capteurs ont la même énergie initiale. Dans la suite, nous proposons une nouvelle heuristique qui s'appuie sur cette méthode analytique afin de trouver une solution pour toute instance de la famille $\mathcal{F}$.

6.3.2 Heuristique OddRingNetworks

L'heuristique OddRingNetworks que nous proposons pour les instances de la famille $\mathcal{F}$ est une méthode itérative basée sur la méthode analytique présentée ci-dessus. Cette heuristique prend en entrée un ensemble de cibles $t_1, t_2, \dots, t_n$ et un ensemble de capteurs $s_1, s_2, \dots, s_n$ avec les énergies initiales respectives $E_1, E_2, \dots, E_n$ où $n = 2p + 1$ et le ratio r entre la puissance en mode veille et la puissance en mode actif est tel que $0 \leq r < 1$. Au début son exécution, l'heuristique détermine l'énergie minimale E_{min} des capteurs. Puis, on applique la méthode analytique en considérant que tous les capteurs ont la même énergie E_{min} . Les couvrants formés sont alors activés successivement pendant les durées spécifiées par la méthode analytique. Après l'activation du dernier couvrant, tous les capteurs d'énergie initiale E_{min} ont épuisé leurs batteries tandis que les autres capteurs ont des énergies résiduelles non nulles. L'ensemble des capteurs résiduels et l'ensemble de cibles ne forment plus un réseau en anneau impair. Si chaque cible est couverte par au moins un capteur, l'heuristique forme un nouveau couvrant en sélectionnant à chaque itération le capteur d'énergie minimale qui couvre le plus de nouvelles cibles (cibles non couvertes par le couvrant en construction). Lorsque le couvrant est formé, on recherche d'éventuels capteurs et on les supprime du couvrant en utilisant la procédure (*suppression_redondance*) présentée dans les chapitres 3 et 4. Le couvrant ainsi formé est activé jusqu'à ce qu'un de ses capteurs épuise son énergie. On recommence la construction et l'activation de nouveaux couvrants jusqu'à ce qu'une cible ne soit plus couverte par un capteur. La complexité de cette heuristique est polynomiale. L'heuristique OddRingNetworks est présentée à la figure 46.

Algorithme OddRingNetworks (S, T)**Input:** T : Ensemble de cibles ; S : Ensemble de capteurs ;**Début**I1. Poser $E_{min} = \min_{s_i \in S} E_i$ I2. Former les couvrants $C_1, C_2, \dots, C_n$ tel que C_1 est un couvrant de taille minimale et $C_{j+1} = f(C_j)$, pour $j = 1, \dots, n-1$ I3. Activer successivement les couvrants C_j pendant $Lt_j = \frac{1}{(p+1)+r*(p-1)} * \frac{E_{min}}{P_a}$ unités de temps, pour $j = 1, \dots, n-1$ et le couvrant C_n pendant la durée $Lt_n = \frac{1-r}{(p+1)+r*(p-1)} * \frac{E_{min}}{P_a}$ unités de tempsI4. **Pour tout** capteur $s_i \in S$ **Faire** **Si** $E_{i,n} = 0$ **Alors** Supprimer s_i de l'ensemble S des capteurs ($S = S \setminus \{s_i\}$) **Finsi** ($E_{i,n}$ est l'énergie résiduelle du capteur après l'activation du couvrant C_n) **Fpour**I5. $k = n + 1$ **Tant que** chaque cible de T est couverte par un capteur de S **Faire** I6. $UNCOV = T$ **Tant que** $UNCOV \neq \emptyset$ **Faire** I7. Sélectionner le capteur s_{min} qui couvre le plus de nouvelles cibles et dont l'énergie est minimale I8. Ajouter le capteur s_{min} au couvrant : $C_k = C_k \cup \{s_{min}\}$ I9. Supprimer de $UNCOV$ toutes les cibles couvertes par le capteur s_{min} **Fintantque** I10. Appeler la procédure *suppression_redondance*(C_k, T) pour supprimer les capteurs redondants de C_k I11. Activer le couvrant C_k jusqu'à ce qu'un capteur épuise sa batterie I12. **Pour tout** capteur $s_i \in S$ **Faire** **Si** $E_{i,k} = 0$ **Alors** Supprimer s_i de l'ensemble S des capteurs ($S = S \setminus \{s_i\}$) **Finsi** **Fpour** I13. $k = k + 1$ **Fintantque****FinAlgo**Les fonctions σ et f sont telles que :

$$\begin{cases} \sigma(s_n) = s_1 \\ \sigma(s_i) = s_{i+1} \text{ pour } i = 1, \dots, n-1 \end{cases}$$

et

$$f(C_i) = \{\sigma(s_{i1}), \sigma(s_{i2}), \sigma(s_{i3}), \dots, \sigma(s_{il})\}, \text{ avec } C_i = \{s_{i1}, s_{i2}, s_{i3}, \dots, s_{il}\}$$

Figure 46 - Algorithme OddRingNetworks

En appliquant l'heuristique OddRingNetworks sur l'exemple de la figure 43, on pose $E_{min} = 8$ à l'instruction I1. A l'instruction I2, on forme le couvrant minimal $C_1 = \{s_1, s_2\}$ et les couvrants $C_2 = f(C_1) = \{s_2, s_3\}$ et $C_3 = f(C_2) = \{s_1, s_3\}$. L'instruction I3 active successivement les couvrants C_1 et C_2 pendant la durée $Lt_1 = Lt_2 = \frac{1}{(p+1)+r*(p-1)} * \frac{E_{min}}{P_a} = 4$ chacun car $p = 1$, $P_a = 1$ et $r = 0$. Ensuite le couvrant C_3 est activé pendant $Lt_3 = \frac{1-r}{(p+1)+r*(p-1)} * \frac{E_{min}}{P_a} = 4$. A l'instruction I4, on supprime le capteur s_2 car son énergie est épuisée. Après l'activation du

couvrant C_3 , l'énergie résiduelle du capteur s_3 est $E_{3,3} = 1$ et celle du capteur s_1 est $E_{1,3} = 2$. A l'instruction I5, on pose $k = 4$, puis on entre dans la première boucle **tantque** car chaque cible est couverte par au moins un capteur. Dans la boucle interne **tantque**, on sélectionne itérativement les capteurs s_3 puis s_1 pour former le couvrant et $C_4 = \{s_1, s_3\}$. Puisque ce couvrant ne comporte pas de capteur redondant, on l'active pendant $Lt_4 = 1$ car le capteur s_3 épuise sa batterie après cette durée de temps. Après l'activation du couvrant C_4 , la cible t_3 n'est couverte par aucun capteur, on sort de la boucle externe **tantque** et l'exécution de l'heuristique s'arrête. L'heuristique a donc donné une durée de vie du réseau $\Delta = 13$. Comme on l'a montré à l'introduction, cette solution améliore de 30% la solution donnée par les algorithmes Energy-Blacklist, Blacklist et Adapdatif, basés sur l'approche réactive. Il est difficile de comparer l'heuristique OddRingNetworks aux heuristiques de la littérature basées sur l'approche classique par couvrants non disjoints car leurs auteurs n'expliquent pas comment choisir la valeur du paramètre w qui indique le nombre maximal de couvrants auxquels peut appartenir un capteur.

Lorsque tous les capteurs ont la même énergie initiale, l'heuristique n'entre pas dans la boucle externe **tantque** et la solution de l'heuristique OddRingNetworks est meilleure que celle donnée par les algorithmes basés sur l'approche réactive. Ce résultat a été démontré dans la sous-section précédente. Nous observons d'après l'exécution sur l'exemple de la figure 43 que l'heuristique OddRingNetwork est meilleure que les algorithmes basés sur l'approche réactive lorsque les capteurs n'ont pas la même énergie. Nous déduisons que pour la famille des réseaux en anneau impair, l'heuristique OddRingNetwork est plus meilleure que les algorithmes Energy-Blacklist, Blacklist, Adapdatif et les autres algorithmes de la littérature basés sur l'approche classique par couvrants disjoints et non disjoints.

6.4 Applications des réseaux en anneau

Nous avons présenté la famille des réseaux en anneau impair comme une famille d'instances difficiles contrairement aux réseaux en anneau pair. Lorsque les capteurs ont la même énergie initiale, nous avons vu que la durée de vie maximale d'un réseau en anneau pair est de $(2 - r) * \frac{E}{P_a}$

et celle d'un anneau impair est au moins égale à $\frac{n-r}{(p+1)+r*(p-1)} * \frac{E}{P_a}$ (où $n = 2p + 1$). Cela implique que lorsque $r = 0$ et en organisant le réseau sous la forme d'un anneau, la durée de vie du réseau est le double de celle d'un capteur si n est pair et les trois demi de celle d'un capteur si n est impair. Cela est un avantage car dans le cas des réseaux en anneau (pair ou impair), le nombre de capteurs est égal au nombre de cibles. Ainsi, pour couvrir n cibles, on peut déployer uniquement n capteurs et obtenir tout de même une durée de vie du réseau supérieure à celle d'un capteur. Les réseaux en anneau sont donc adaptés aux applications où l'on peut choisir de façon déterministe la topologie du réseau et où l'on souhaite déployer un petit nombre de capteurs. Les applications de couverture de surface sont un exemple d'applications pour lesquelles les réseaux en anneau sont adaptés. Supposons que l'on souhaite surveiller une surface de taille $M \times N$ par des capteurs dont le rayon de sensibilité est égal à R_s . Quadrillons la surface par des colonnes de largeur R_s et des lignes de hauteur R_s . Chaque cellule du quadrillage représente une cible. La cible est supposée placée au centre de la cellule. Une cible est couverte par un capteur si la totalité (ou quasi-totalité) de la surface de la cellule qu'elle représente est couverte par ce capteur. Les capteurs sont ensuite

disposés sur le quadrillage comme illustré à la figure 47 ci-dessous. Sur cette figure, les cibles sont affichées sous forme de points noirs tandis que les capteurs sont représentés par des carrés colorés. Les capteurs de même couleur, ainsi que les cibles placées entre eux forment un réseau en anneau, excepté les capteurs (jaunes) au centre de la grille. On compte deux réseaux en anneau et un réseau linéaire sur cette figure. Cette méthode permet donc de disposer des capteurs sur une surface de sorte à couvrir toute la cible tout en formant un certain nombre de réseaux en anneau. Les réseaux formés peuvent être pairs ou impairs et le MLCP est traité pour chacun d'eux de façon totalement indépendante des autres. Pour chaque réseau en anneau impair, l'heuristique OddRingNetwork est exécutée. Pour les réseaux en anneau pair, une version modifiée de cette heuristique peut être utilisée. Dans cette version, seuls deux couvrants disjoints sont formés au départ et activés ensuite. Puis, l'heuristique se poursuit à partir de l'instruction I5 tel que présenté à la figure 46. Pour le cas du réseau linéaire, l'heuristique OddRingNetwork peut aussi être utilisée en commençant l'exécution à partir de l'instruction I5.

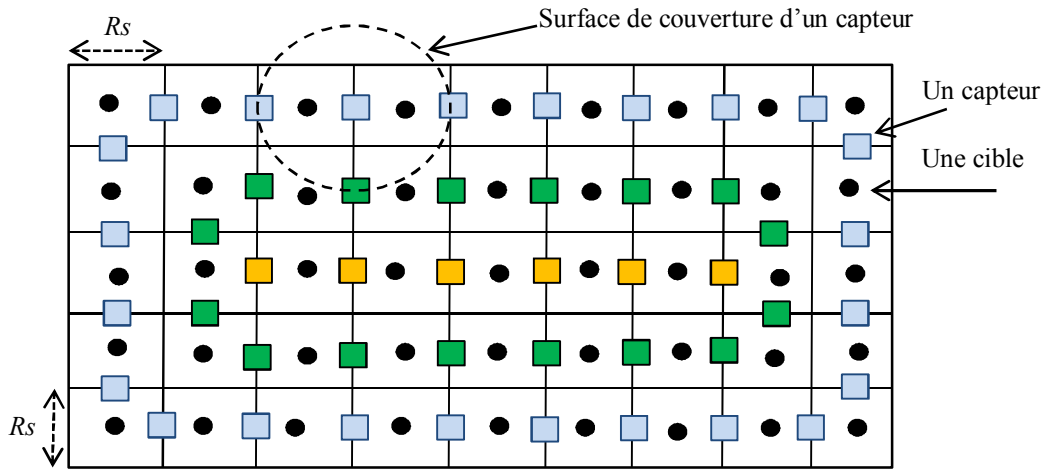


Figure 47 - Exemple d'application des réseaux en anneau

6.5 Conclusion

Dans ce chapitre, nous avons étudié la famille des réseaux en anneau impair. Il s'agit d'une famille d'instances difficiles pour lesquelles les algorithmes Energy-blacklist, Blacklist et Adaptatif, présentés dans les chapitres 3 et 4, trouvent des solutions sous-optimales. De même, les algorithmes de la littérature ne trouvent pas de solutions optimales pour les instances de cette famille. Après avoir défini la famille des réseaux en anneau impair, nous avons démontré une propriété fondamentale à savoir que toute solution optimale d'une instance de cette famille ne laisse aucune énergie résiduelle sur un capteur lorsque les capteurs ont la même énergie initiale. En se basant sur cette propriété et sur les automorphismes de graphe, nous avons proposé une méthode analytique qui donne une solution optimale pour les réseaux en anneau de 3 capteurs et 3 cibles. Nous avons conjecturé que cette solution est optimale pour toute instance de la famille des réseaux en anneau impair. Nous avons ensuite montré que cette solution est meilleure que celle donnée par les algorithmes basés sur l'approche réactive et par les algorithmes de la littérature. Puisque cette méthode analytique ne s'applique qu'aux instances où les capteurs ont la même énergie initiale, nous avons proposé une nouvelle heuristique, OddRingNetwork, qui s'appuie sur cette méthode

analytique et qui permet d'obtenir une solution pour toute instance de la famille des réseaux en anneau impair, quelles que soient les énergies initiales des capteurs. A travers un exemple, nous avons observé que l'heuristique OddRingNetwork est meilleure que les algorithmes Energy-blacklist, Blacklist et Adaptatif. Enfin, nous avons présenté des champs d'application des réseaux en anneau (pair ou impair), notamment les applications impliquant la couverture de surface. Nous avons proposé une méthode permettant de couvrir une surface par plusieurs réseaux en anneau et un réseau linéaire.

Chapitre 7 Réseaux de capteurs pour la surveillance de la pollution atmosphérique

7.1 Introduction

La pollution est définie comme étant l'introduction directe ou indirecte, par suite de l'activité humaine, de substances ou de chaleur dans l'air, l'eau ou le sol, susceptibles de porter atteinte à la santé humaine ou à la qualité des écosystèmes aquatiques ou des écosystèmes terrestres, qui entraînent des détériorations aux biens matériels, une détérioration ou une entrave à l'agrément de l'environnement ou à d'autres utilisations légitimes de ce dernier [81]. La pollution est en général un sous-produit de l'activité humaine qui peut toucher l'atmosphère, le sol ou les eaux. La pollution atmosphérique est l'introduction par l'homme, directement ou indirectement, dans l'atmosphère et les espaces clos, de substances ayant des conséquences préjudiciables de nature à mettre en danger la santé humaine, à nuire aux ressources biologiques et aux écosystèmes, à influencer sur les changements climatiques, à détériorer les biens matériels, à provoquer des nuisances olfactives excessives [82]. De nos jours, la préservation de la qualité de l'atmosphère est devenue un enjeu économique, sanitaire et environnemental planétaire. D'après un communiqué de presse du 2 Mai 2018, l'Organisation Mondiale de la Santé (OMS) révèle que sept millions de personnes meurent chaque année dans le monde à cause de la pollution de l'air et neuf personnes sur dix respirent un air contenant des niveaux élevés de polluants. L'OMS estime aussi que plus de 90% des décès dus à la pollution de l'air se produisent dans des pays à revenu faible ou intermédiaire, principalement en Asie et en Afrique, suivis des pays à revenu faible ou intermédiaire des Régions de la Méditerranée orientale, de l'Europe et des Amériques [83]. Ces estimations proviennent de la base de données de l'OMS sur la qualité de l'air ambiant à laquelle plus de 4300 villes de 108 pays sont inscrites. Depuis 2016, plus d'un millier de villes supplémentaires ont rejoint la base de données de l'OMS, ce qui montre que les pays sont plus nombreux que jamais à mesurer la pollution de l'air et à prendre des mesures pour la réduire. Cependant, le manque de données relatives à la pollution de l'air est criant en Afrique. En Afrique, seuls 8 pays parmi les 47 pays de la région transmettent des données relatives à la pollution atmosphérique à l'OMS. Pourtant les gouvernements africains sont soucieux de la protection de l'atmosphère à l'instar du Cameroun. Le décret n°2011/2582/PM du 23 août 2011 [84] fixe les modalités de protection de l'environnement au Cameroun et instaure l'implantation de stations de mesure et de contrôle de la qualité de l'air sur des sites où la santé et l'environnement font l'objet d'une protection particulière et des sites qui sont susceptibles de donner une représentation valable de la pollution de l'air sur une grande parcelle du territoire. Ces stations de mesure et de contrôle de la qualité de l'air sont très fiables, précises et capables de mesurer une large gamme de polluants. Cependant, ces stations de mesure ont une grande taille, un poids lourd et un coût très élevé de l'ordre de plusieurs dizaines de millions de francs CFA [85].

Cela entraîne une limitation du nombre de stations pouvant être déployés pour couvrir l'ensemble des villes du pays. De plus, ces stations de mesure fournissent les données sur la pollution de l'air par heure voire par jour à cause des modèles chronographes qu'ils utilisent. Par conséquent, les cartes de la pollution atmosphérique construites à partir de ces stations de mesure ont une très faible résolution spatio-temporelle [85]. Contrairement à ces stations de mesure et de contrôle de la qualité de l'air, les réseaux de capteurs sans fil permettent de nos jours de mesurer la qualité de l'air à un coût abordable, avec une fréquence de collecte de l'ordre de quelques minutes et sur de plus grandes superficies. La surveillance de la qualité de l'air par les réseaux de capteurs sans fil est la mieux adaptée pour les pays en voie de développement en général et le Cameroun en particulier.

Dans la suite, nous présentons les principaux polluants de l'air et leurs effets sur la santé. Par la suite, nous faisons l'état de l'art des systèmes de surveillance de la pollution de l'air basés sur les réseaux de capteurs sans fil. Puis, nous présentons les détails d'implémentation d'un système de surveillance de la qualité que nous avons réalisé. En plus de mesurer la qualité de l'air, ce système permet de détecter les incendies afin de lutter contre les incendies de marché qui font des ravages dans les villes du Cameroun. Enfin, nous discutons de la problématique d'énergie dans les systèmes de surveillance de la qualité de l'air basés sur les réseaux de capteurs en prenant le système que nous avons développé comme exemple. Nous montrons que la redondance de nœuds de capteurs et les algorithmes du MLCP peuvent permettre d'augmenter la robustesse et l'autonomie énergétique de tels systèmes. Nous terminons par une conclusion.

7.2 Principaux polluants et leurs conséquences

Le plomb ajouté à l'essence est le polluant le plus néfaste des émissions de véhicules. Le plomb, un poison insidieux qui agit lentement, est facilement absorbé par l'organisme, qui le retient. Le plomb des émissions présent dans l'air, lorsqu'on le respire, s'accumule dans les tissus cellulaires et entraîne anémie, hypertension et perte irréversible de certaines fonctions cérébrales, notamment chez les nourrissons et les enfants. Il a été démontré qu'une exposition au plomb réduit le quotient intellectuel des enfants de 2 à 3 points par 100 microgrammes additionnels de plomb par litre de sang. Les enfants, notamment en bas âges, courent le plus grand risque car, par rapport aux adultes, leur appareil digestif absorbe beaucoup plus facilement le plomb qui s'accumule par ailleurs dans le sol sur lequel ils jouent ainsi que sur leurs vêtements et leurs jouets. Les conséquences ne s'arrêtent pas aux maladies mais comprennent aussi une altération irréversible des capacités mentales. Ce sont les enfants les plus démunis qui forment la population la plus à risque car la malnutrition augmente l'absorption de plomb. Le plomb entraîne une pression artérielle élevée, des maladies cardiovasculaires et des troubles neurologiques et rénaux chez les adultes [86].

Les trois polluants issus de carburants les plus nocifs qui affectent le système respiratoire sont: le dioxyde de soufre (SO_2), les oxydes d'azote (NO_x) et les particules fines (PM_{10} et $\text{PM}_{2.5}$). Le dioxyde de soufre (SO_2) est un gaz irritant qui provient essentiellement de la combustion des matières fossiles contenant du soufre (comme le charbon). Il augmente la fréquence des bronchites et des infections pulmonaires. Les oxydes d'azote (NO_x) tendent à accroître les symptômes respiratoires et à altérer les fonctions pulmonaires. Le dioxyde d'azote (NO_2) peut entraîner une altération de la fonction respiratoire et une hyper activité bronchique. Chez les enfants et les

asthmatiques, il peut augmenter la sensibilité des bronches aux infections microbiennes. Les particules en suspension d'un diamètre inférieur à $10\mu m$ (PM_{10}), en particulier les particules ultra fines d'un diamètre inférieur à $2.5\mu m$ ($PM_{2.5}$), qui traversent le filtre du larynx, s'accumulent dans les poumons et irritent les voies respiratoires. Certaines particules peuvent avoir des propriétés cancérogènes. La plus grande quantité de ces trois polluants est produite par la combustion du diesel [86].

Le monoxyde de carbone (CO) est un gaz invisible et inodore qui est issu de la combustion incomplète de combustibles fossiles. Les combustibles concernés sont le bois, le charbon, le charbon de bois, le pétrole, le gaz naturel, l'essence, le kérosène et le propane. Le monoxyde de carbone est produit en plus grande quantité lors de la combustion de carburant. Ce gaz peut aussi être émis par n'importe quel appareil de chauffage (plaques ou cuisinières à gaz, four à pétrole,...) mal entretenu ou endommagé. Comme l'oxygène, le monoxyde de carbone pénètre le corps par les poumons lorsque l'on respire. Il vient prendre sa place dans les globules rouges, réduisant ainsi le taux d'oxygène arrivant au cœur, au cerveau et aux autres organes vitaux. En cas de concentration élevée, le monoxyde de carbone peut entraîner la mort très rapidement. Une exposition à un faible niveau de monoxyde de carbone pendant une longue période est également très dangereuse et provoque souvent des symptômes souvent confondus avec ceux de la grippe : des maux de tête, des vertiges et de la fatigue.

La famille des Composés Organiques Volatils (COV) regroupe plusieurs milliers de composés (hydrocarbures, solvants, ...) aux caractéristiques très variables. Ils ont un impact direct sur la santé (certains sont toxiques ou cancérogènes). Ce sont des gaz et des vapeurs qui contiennent du carbone, comme les vapeurs d'essence et les solvants. Ils interviennent dans le processus de formation d'ozone dans la basse atmosphère et participent donc au réchauffement de la planète. Les Composés Organiques Volatils (COV) entrent dans la composition des carburants mais aussi de nombreux produits courants: peintures, encres, colles, détachants, cosmétiques, solvants, etc. pour des usages ménagers, professionnels ou industriels (pour ces raisons, leur présence dans l'air intérieur peut aussi être importante). Ils sont émis lors de la combustion de combustibles (notamment dans les gaz d'échappement), ou par évaporation lors de leur fabrication, de leur stockage et de leur utilisation. Les effets des COV sont très variables selon la nature du polluant envisagé. Ils vont de la simple gêne olfactive et une irritation, à une diminution de la capacité respiratoire. Le benzène est un composé cancérogène reconnu.

L'Ozone troposphérique (O_3) est une forme particulière de l'oxygène. Contrairement aux autres polluants, l'ozone n'est pas émis par une source particulière mais résulte de la transformation photochimique de certains polluants de l'atmosphère, issus principalement du transport routier (NO_x et COV), en présence des rayonnements ultra-violet solaires. L'ozone entre en action le plus souvent quand les températures sont en hausse. L'ozone au sol irrite les voies respiratoires et les yeux. Une gêne respiratoire, une toux et une respiration sifflante sont des symptômes d'une exposition à une concentration élevée d'ozone. Sont particulièrement vulnérables les enfants et les personnes âgées, ainsi que les personnes souffrant de troubles respiratoires et cardiaques.

7.3 Etat de l'art des systèmes de surveillance de la pollution de l'air basés sur les réseaux de capteurs

Les systèmes de surveillance de la qualité de l'air basés sur les réseaux de capteurs sans fil sont connus sous le nom de TNGAPMS (The Next Generation Air Pollution Monitoring System) [85]. Grâce à l'évolution de la technologie, des capteurs portables, de faible coût, de petite taille et de temps de réponse rapide (ordre des secondes ou des minutes) sont désormais disponibles. Ces capteurs sont ceux utilisés dans les TNGAPMS. Cependant, aucun de ces capteurs ne fournit la même précision que les instruments utilisés par les stations conventionnelles de mesure et de contrôle de la qualité de l'air. Les TNGAPMS peuvent être regroupés en trois catégories selon les porteurs des nœuds de capteurs à savoir les systèmes SSN (Static Sensor Network), les systèmes CSN (Community Sensor Network) et les systèmes VSN (Vehicle Sensor Network). Dans les systèmes SSN, les nœuds de capteurs sont statiques et fixés sur les murs, les poteaux des feux de signalisation ou des lampadaires et d'autres positions judicieusement choisies. Dans les systèmes CSN, les nœuds de capteurs sont portés par les utilisateurs. Ces nœuds de capteurs sont associés aux téléphones intelligents (smartphones) des utilisateurs afin de collecter, analyser et distribuer l'état de la pollution de l'air. Dans les systèmes VSN, les nœuds de capteurs sont placés sur les moyens de transport public tels que les bus, les trains et les taxis. Grâce à la mobilité des véhicules, un nœud de capteur est capable de couvrir une large zone géographique. Chacune de ces trois classes de TNGAPMS présente des avantages et des inconvénients. Dans certains cas, les systèmes SSN n'ont pas de contrainte énergétique car les nœuds peuvent être branchés sur le réseau électrique (courant d'alimentation des feux de signalisation, des lampadaires ou des maisons), être alimentés par des panneaux solaires. Dans les autres cas, par exemple lorsque le réseau de capteurs est déployé dans une zone où le réseau électrique n'est pas inaccessible, la contrainte d'énergie devient importante. Le choix de la position des nœuds de capteurs est un inconvénient des systèmes SSN car l'emplacement des nœuds influence la carte de la pollution produite par le système. Par ailleurs, un grand nombre de nœuds de capteurs doit être déployé afin d'assurer une large couverture géographique et une bonne résolution spatiale. Les systèmes CSN ont l'avantage d'utiliser le module GPS, le réseau cellulaire et même la puissance de calcul du téléphone de l'utilisateur ; ce qui réduit le coût du système. De plus, la mobilité des utilisateurs permet de couvrir une large zone géographique. Un inconvénient des systèmes CSN est la faible fiabilité des données collectées car les nœuds de capteurs sont placés dans les poches et les sacs à main des utilisateurs. De plus, les utilisateurs passent un temps considérable à l'intérieur (bureaux, maisons, ...). Par ailleurs, les nœuds étant portés par des personnes, la contrainte de poids et de taille est très forte dans cette classe de TNGAPMS. Parmi les avantages des systèmes VSN, on peut citer l'absence de contrainte énergétique (car les nœuds de capteurs sont alimentés par la batterie des véhicules), de la contrainte de poids et de la contrainte de taille. De plus, la grande mobilité des nœuds de capteurs permet d'assurer une très large couverture géographique. Un inconvénient des systèmes VSN est l'utilisation de GPS permettant de localiser les nœuds de capteurs (du fait de leur mobilité). Un autre inconvénient est le problème de redondance des données lorsque le véhicule se retrouve dans des embouteillages ou stationné dans un parking. En considérant les avantages et inconvénients des classes de TNGAPMS et des systèmes de surveillance conventionnels CSMN (Conventional Stationary Monitoring Network), il ressort qu'un compromis entre le coût d'un nœud, la qualité des

données, la résolution spatiale et la résolution temporelle doit être fait comme le montre la figure 48.

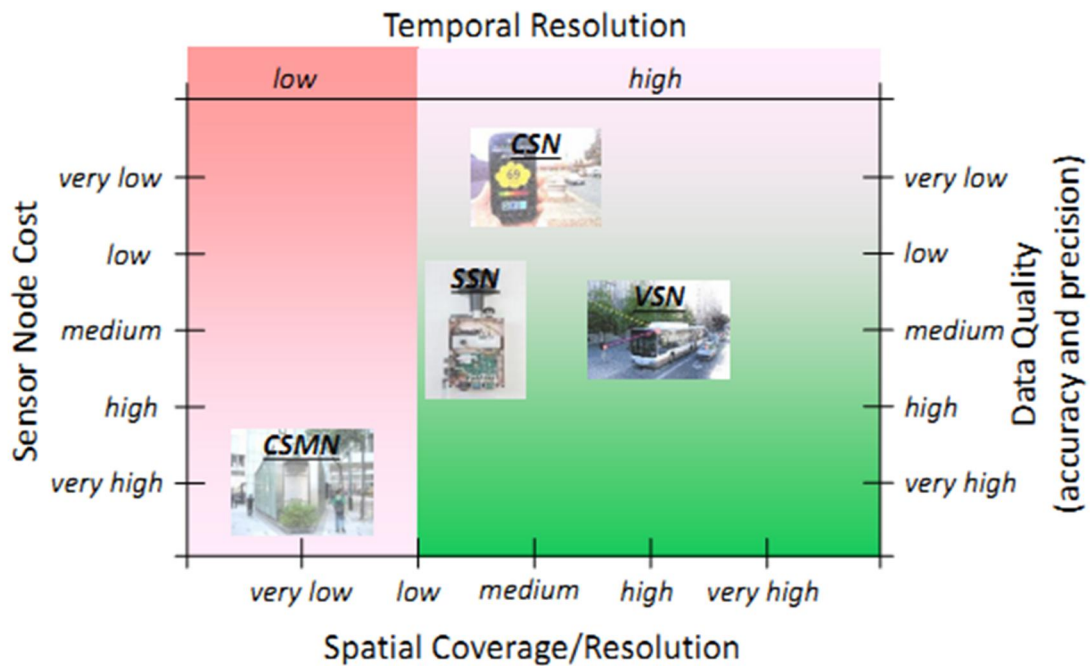


Figure 48 - Comparaison des systèmes de surveillance de la qualité de l'air [85]

De nombreux systèmes de surveillance de la qualité de l'air basés sur les réseaux de capteurs sans fil (TNGAPMS) ont été réalisés au cours de cette décennie. Plus d'une vingtaine de TNGAPMS ont été répertoriés par Yi et al. [85] et Pavani et Rao [87]. Parmi ces systèmes, nous pouvons citer les exemples ci-après appartenant à des catégories différentes (SSN, VSN et CSN).

T. Ahuja, V. Jain et S. Gupta [88] ont développé un système SSN pour surveiller la qualité de l'air à des endroits sensibles de la ville de New Delhi. Il s'agit notamment d'une décharge, d'une zone industrielle, d'un parc public et d'une école. Un nœud de capteur est placé dans chacune de ces zones. Dans l'architecture de ce système, un nœud de capteur est formé d'une carte Arduino Uno sur lequel est monté trois capteurs de gaz à savoir deux capteurs MQ135 et un capteur MQ3. Le capteur de gaz MQ135 permet de mesurer la qualité de l'air en termes de quantité de d'ammoniac, de dioxyde d'azote, de benzène et de fumée. Le capteur MQ3 est utilisé pour mesurer la quantité d'alcool dans l'air. La carte Arduino est connectée à une carte WIFI et transmet les données collectées sur le Cloud notamment sur la plateforme *thething.io*. Cette plateforme sauvegarde les données sur une base de données et offre des fonctionnalités de visualisation et d'analyse des données. De plus, cette plateforme envoie les alertes par SMS ou courrier électronique à ses utilisateurs à propos de la qualité de l'air sur les différentes zones surveillées lorsque les seuils fixés sont dépassés. Dans ce système, les données sont mesurées et envoyées toutes les deux minutes.

S. Abraham et X. Li [89] ont conçu un système SSN pour mesurer la qualité de l'air en environnement intérieur (maison d'habitation, immeuble, ... etc.). Ce système est composé de plusieurs nœuds de capteurs, d'une station de base, d'une base de données et d'un serveur web. La

station de base reçoit périodiquement les mesures provenant des différents nœuds de capteur. Ces données sont envoyées à un serveur de base de données pour le stockage. Le serveur web fournit une interface web aux utilisateurs pour accéder aux données. Un nœud de capteur est composé d'une carte Arduino Uno, d'un module radio ZigBee, d'un capteur MG811 pour mesurer la quantité de dioxyde de carbone, d'un capteur TGS602 qui mesure la quantité de Composés Organiques Volatils (COV), d'un capteur MQ7 pour la mesure de la quantité de monoxyde de carbone, d'un capteur MQ131 pour mesurer la quantité d'ozone et d'un capteur RTH03 pour la mesure de la température et de l'humidité.

S. Mansour et al. [90] ont proposé un système SSN pour mesurer la qualité de l'air en zone urbaine et industrielle. L'architecture de ce système comprend des nœuds de capteurs, une passerelle, un serveur de base de données, un serveur web, et un système d'aide à la décision. Le nœud de capteur est la plateforme Waspote fabriquée par Libelium. Waspote intègre plusieurs capteurs de gaz notamment le monoxyde de carbone, le dioxyde de carbone, le dioxyde d'azote, l'ammoniac, le méthane, etc. Le module radio utilisé est ZigBee. La passerelle est la plateforme Meshlium de Libelium. Cette passerelle dotée d'un module radio WIFI collecte les données provenant des nœuds de capteur et les envoie via Internet vers le serveur de base de données. Les utilisateurs qui s'enregistrent sur le serveur web et reçoivent des alertes par SMS ou courrier électronique lorsque les seuils qu'ils ont définis sont atteints. En plus des informations temps réel, les utilisateurs peuvent aussi accéder aux données sauvegardées précédemment. Le système d'aide à la décision permet de prédire la future qualité de l'air en construisant des modèles de prédiction qui utilisent l'historique des données.

A. Kadri et al. [91] ont développé un système SSN pour surveiller et analyser la qualité de l'air dans la ville de Doha au Qatar. Ce système est composé de deux éléments principaux : les nœuds de capteur et un serveur. Chaque nœud de capteur est doté d'un module GPRS (General Packet Radio Service) pour la connexion au réseau cellulaire et des capteurs de gaz pouvant mesurer le niveau d'ozone, de dioxyde d'azote et de monoxyde de carbone. Ces nœuds de capteurs comportent aussi des capteurs de température et de l'humidité. Les nœuds de capteurs sont alimentés par des panneaux solaires. Le serveur collecte les données envoyées par les nœuds de capteurs, il les stocke dans la base de données. Un site Internet et une application mobile permettent d'afficher en temps réel les données collectées, ainsi que des rapports sur les mesures antérieures. Les interfaces sont conçues pour permettre au grand public de visualiser et de comprendre l'état de la qualité de l'air dans les zones observées. Quatre capteurs ont été déployés sur quatre zones différentes et prélèvent les mesures toutes les cinq minutes.

J. Kang et J. Kim [92] ont conçu un système VSN pour mesurer la qualité de l'air dans la ville d'Incheon en Corée du Sud. Ils ont fabriqué un nœud de capteurs composé d'un microcontrôleur ATmega128, d'un module radio d'une portée d'un kilomètre, d'un GPS et des capteurs de température, de Composés Organiques Volatils (COV), de dioxyde d'azote et d'Ozone. Le nœud de capteur est installé sur un bus qui se déplace dans la ville. Le GPS donne la position du nœud au moment de la mesure de la quantité des polluants. Un serveur collecte les données envoyées par le nœud de capteur et fournit une interface de visualisation.

S. Devarakonda et al. [93] ont proposé un système VSN et un système CSN pour mesurer la qualité de l'air dans les villes de New Jersey et New York aux USA. Le système VSN est constitué de nœuds de capteurs placés dans des infrastructures de transport public, notamment des bus. Un nœud de capteur est composé d'une carte Arduino Mega128, d'un GPS pour localiser le nœud, d'un modem cellulaire SIM5218 3G/GPRS pour connecter le nœud au réseau cellulaire, d'un capteur MQ7 pour mesurer la quantité de monoxyde de carbone. Le nœud est alimenté par la batterie du bus sur lequel il se trouve. Les données prélevées par les nœuds de capteurs sont envoyées sur un serveur du Cloud. Le système CSN proposé est constitué de nœuds de capteurs connectés par Bluetooth aux téléphones des utilisateurs. Un nœud de capteur est une plateforme NODE dotée de capteur de monoxyde de carbone. Les mesures prélevées sont envoyées par Bluetooth au téléphone qui les transmet via le réseau cellulaire vers un serveur du Cloud. Dans les deux systèmes, les données reçues par le serveur sont traitées et affichées sous forme de carte de pollution. Une interface web et une application mobile permettent aux utilisateurs de visualiser ces cartes en temps réel.

7.4 Implémentation d'un système de surveillance de la qualité de l'air et de détection d'incendies

Dans cette section, nous proposons un système de surveillance de la qualité de l'air basé sur les réseaux de capteurs sans fil (TNGAPMS) de la classe des systèmes SSN car les nœuds de capteurs sont statiques. Dans ce système nous considérons uniquement le monoxyde de carbone à titre d'exemple mais d'autres polluants peuvent être contrôlés suite à une adaptation du système. En plus de surveiller le niveau de monoxyde de carbone dans l'air, ce système permet à la fois de prévenir les incendies grâce à la détection de fuite du gaz inflammable GPL (Gaz de Pétrole Liquéfié) composé de butane et de propane, mais aussi à détecter les débuts d'incendies grâce au contrôle de la température ambiante et de la présence de fumée. A chacun des produits contrôlés est associé un niveau limite à ne dépasser. Lorsque le niveau d'un produit dépasse le seuil qui lui a été associé, une alarme est déclenchée par le nœud de capteur qui a détecté le dépassement de seuil. Le rôle de cette alarme est d'inciter les personnes présentes à évacuer les lieux au plus vite et à appeler le service des pompiers. Au même moment, un courrier électronique est envoyé aux différents utilisateurs du système pour les informer du dépassement de seuil. Durant le fonctionnement du système, les données recueillies sont envoyées sur un serveur et stockées dans une base de données. Ces données peuvent être visualisées en temps réel via une interface web par les utilisateurs du système. Ce système peut être utilisé par le gouvernement, les entreprises et les particuliers pour la surveillance des lieux publics comme les marchés, les industries et les maisons d'habitation. Dans les grandes villes du Cameroun (Yaoundé, Douala, Maroua, etc.), les incendies de marchés, de bâtiments publics et de maisons d'habitation sont fréquentes et entraînent de lourdes pertes financières et parfois des pertes en vies humaines. Un tel système permettra à la fois de surveiller la qualité de l'air (en termes de niveau de monoxyde de carbone pour cet exemple) et de limiter les dégâts causés par les incendies grâce à une détection précoce.

7.4.1 Architecture du système

L'architecture du système que nous proposons est illustrée à la figure 49. Les nœuds de capteurs sont placés à des endroits appropriés et transmettent de façon périodique les mesures vers la station de base en utilisant la technologie LoRa. Cette technologie permet aux nœuds de capteurs de communiquer avec la station à plusieurs kilomètres de distance (allant de 5 km en zone urbaine à 15 km en zone rurale). Aucune communication ne se fait entre deux nœuds de capteurs. Il existe une communication bidirectionnelle entre chaque nœud de capteur et la station de base. La station de base est dotée à la fois d'un modem LoRa et d'une carte WiFi. Cela lui permet de récupérer les données envoyées par les nœuds de capteurs et de les transmettre vers un serveur du centre de contrôle et de surveillance via Internet. La station de base joue ainsi un rôle de passerelle entre le réseau de capteurs et le serveur. Le serveur qui réceptionne les données est placé dans les locaux du centre de contrôle et de surveillance. Un Système de Gestion de Base de Données (SGBD) est installé sur ce serveur et permet de stocker et de manipuler les données récoltées par le réseau de capteurs. Une application web est aussi installée sur le serveur afin de visualiser, sous forme de courbes, les mesures prélevées en temps réel.

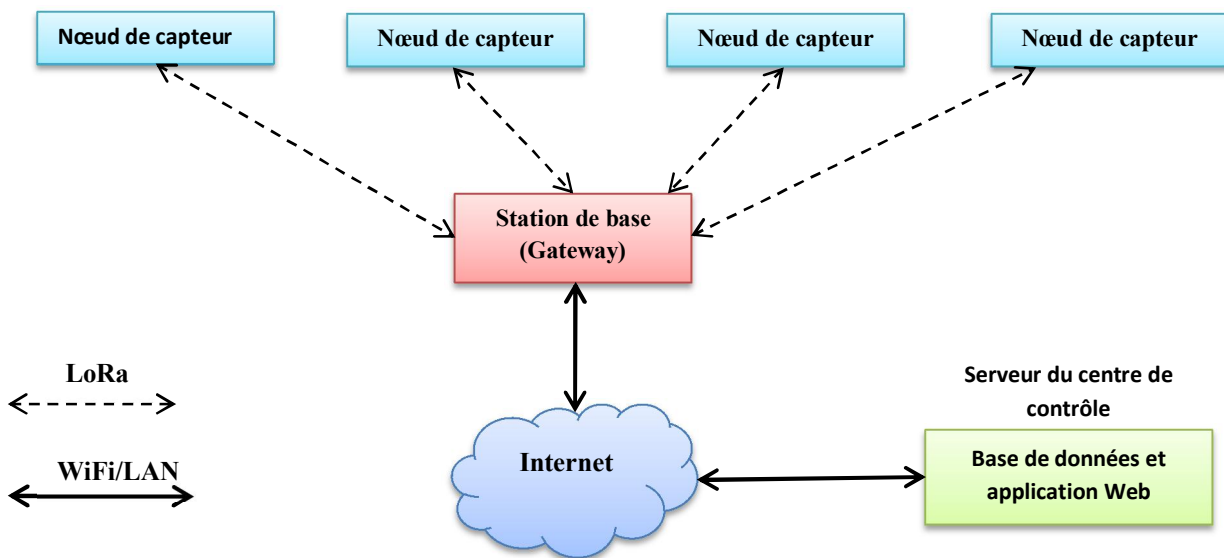


Figure 49 - Architecture générale du système

7.4.2 Architecture d'un nœud de capteur

La figure 50 présente l'architecture d'un nœud de capteur. Un nœud de capteur est constitué d'une carte Arduino Uno [40] qui comporte un microcontrôleur ATmega328 8 bits ayant une fréquence d'horloge de 16 MHz et une mémoire Flash programmable de 32 Ko. Le microcontrôleur ATmega328 consomme 1.1 mA en mode actif, 0.35 mA au repos (mode Idle) et moins de 1 μA en mode veille et fonctionne à un voltage maximal de 5.5V. Un nœud de capteur comporte aussi un modem LoRa SX1276 [41] intégrée à une carte fille Dragino afin de faciliter la connexion avec la carte Arduino. Ce modem consomme en mode transmission entre 20 mA (pour une puissance de +7dBm) et 120 mA (pour une puissance de +20 dBm). En mode réception, le modem consomme

entre 10.8 et 12 mA tandis qu'en mode veille, il consomme moins de $1\mu A$. Le débit de communication peut aller jusqu'à 300 kpbs. Le nœud de capteur est aussi constitué d'un capteur de température, d'un capteur de gaz et d'une alarme. Le capteur de température est le thermomètre numérique DS18B20 [94] qui renvoie une valeur sur 9 à 12 bits. Le capteur DS18B20 consomme 750 nA en standby avec un voltage d'au plus 5.5V. Le capteur de gaz est le capteur Grove MQ2 [17]. Il est capable de détecter le méthane, le GPL, la fumée, le monoxyde de carbone, l'alcool, le dihydrogène. Ce capteur fonctionne sous une tension maximale de 5V et consomme environ 160 mA. La valeur renvoyée par le capteur est une tension, plus elle est grande, plus la concentration du gaz est élevée. Le capteur MQ2 peut mesurer une concentration de gaz allant de 100 ppm (parts per million) à 10000 ppm en d'autres termes une concentration minimale de 0.01% et une concentration maximale de 1%. Enfin, le nœud de capteur est alimenté par une batterie rechargeable 9V connectée au port jack de la carte Arduino et peut aussi être alimenté par le réseau électrique via un adaptateur 5V grâce au port USB de la carte Arduino. La figure 51 présente les images des différents composants utilisés pour la conception d'un nœud de capteur. Sur le plan logiciel, le nœud de capteur est programmé en langage C++ via l'IDE open-source Arduino qui comporte un ensemble de bibliothèques facilitant l'utilisation des composants de carte et des périphériques qui y sont connectés. De nouvelles bibliothèques sont ajoutées régulièrement grâce à une large communauté de développeurs Arduino.

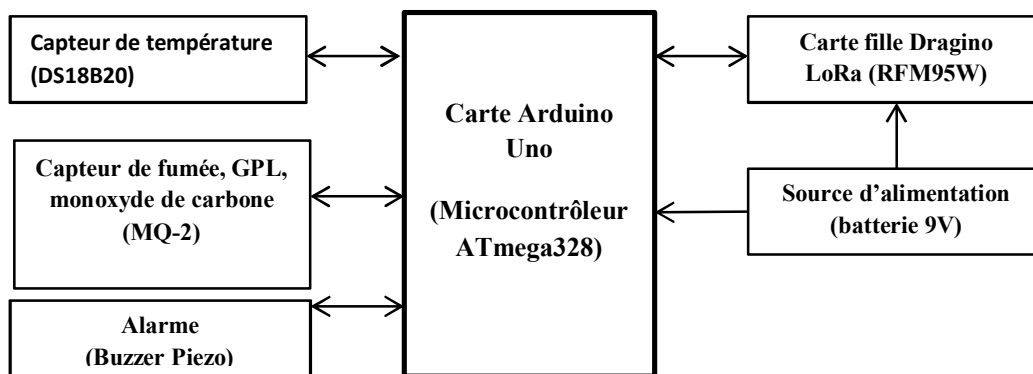


Figure 50 - Architecture d'un nœud de capteur

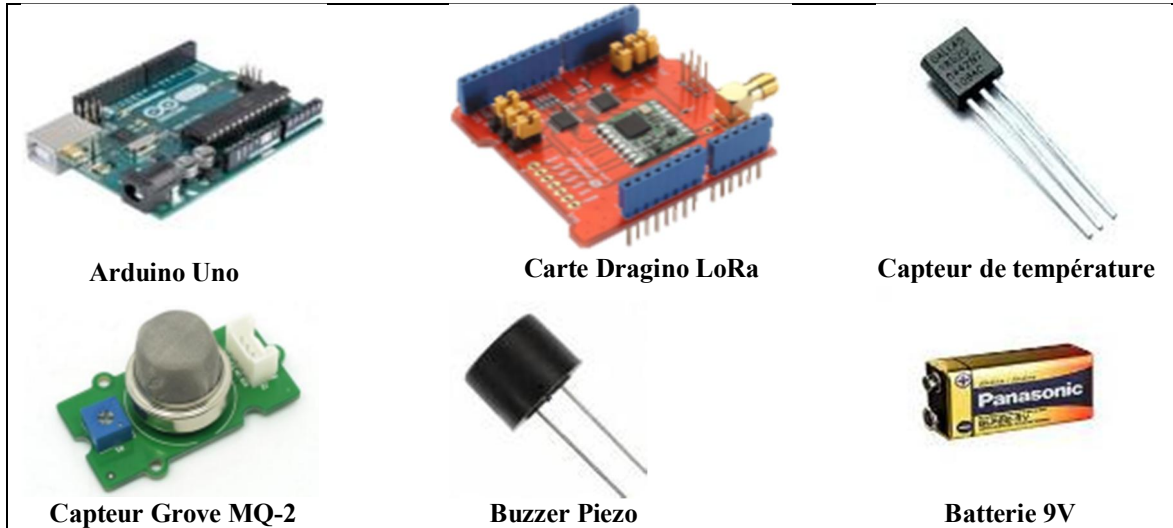


Figure 51 - Matériels utilisés pour la conception d'un nœud de capteur

7.4.3 Architecture de la station de base

La station de base est constituée d'un ordinateur sur carte (System on Chip) appelé Raspberry Pi 3 modèle B connecté à des périphériques d'entrée/sortie (moniteur, clavier, souris) et d'un modem LoRa SX1276 (figure 52). Le Raspberry Pi 3 [95] est constitué entre autre d'un processeur 64 bits, d'une fréquence d'horloge de 1.2 GHz, d'une mémoire RAM de 1 Go, d'une carte graphique de 400 MHz, d'une carte Ethernet, d'une carte WiFi, d'une carte Bluetooth, de quatre ports USB et d'un port HDMI. Une carte SD externe, généralement de 8 Go, est utilisée comme mémoire de stockage des applications et des données. Le Raspberry consomme un courant de 2.5 A et est alimenté par un adaptateur 5V connecté au réseau électrique. Le modem LoRa SX1276 [41] est le même que celui utilisé par le nœud de capteur. La station de base fonctionne sous le système d'exploitation Raspbian qui est une distribution du système Linux. La figure 53 présente des images d'un Raspberry Pi 3 connecté à un modem LoRa et à des périphériques d'entrée/sortie et l'interface de bureau du système Raspbian. Il est possible d'utiliser le Raspberry Pi sans périphériques d'entrée/sortie (moniteur, clavier, souris) en se connectant à distance depuis un autre ordinateur. En effet, SSH (Secure Shell) est une fonctionnalité de Linux qui permet d'ouvrir une session de terminal sur le Raspberry Pi à partir d'une ligne de commande d'un autre ordinateur. L'inconvénient de SSH est qu'il ne gère pas l'interface graphique. Afin de résoudre cette limitation, Raspbian intègre un serveur VNC (Virtual Network Connection) qui permet de télécommander à distance le Raspberry Pi tout en affichant son bureau. Il suffit pour cela d'activer la fonctionnalité VNC du Raspberry Pi (qui est désactivée par défaut) et d'installer un client VNC, en général RealVNC, sur la machine distante. Grâce au contrôle à distance du Raspberry Pi et sa petite taille, ce dernier peut être placé à l'extérieur (à condition qu'une borne WiFi se trouve à proximité) et alimenté par un panneau solaire par exemple. Il peut ainsi être contrôlé depuis une machine située au centre de contrôle.

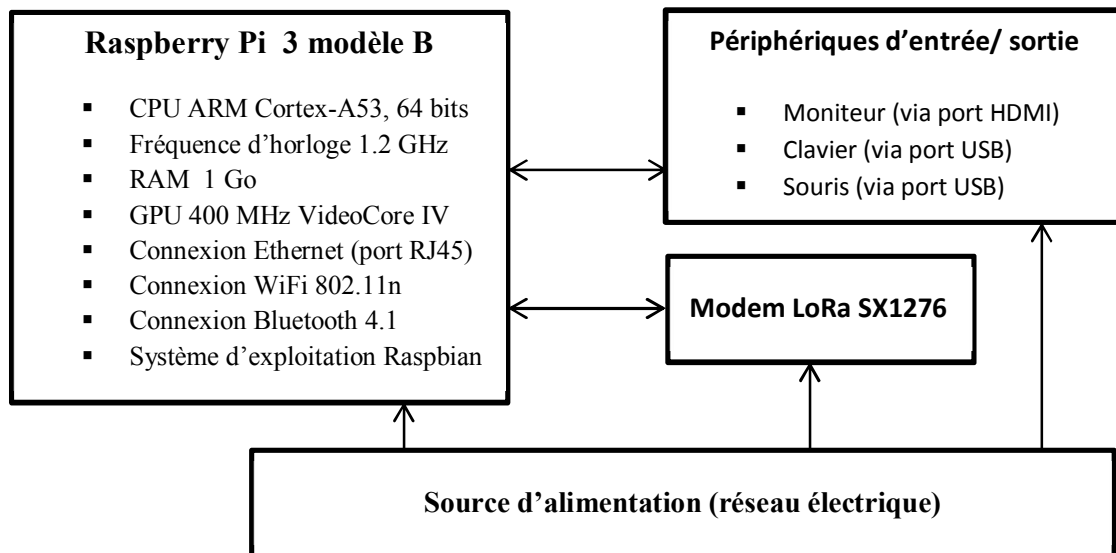


Figure 52 - Architecture d'une station de base

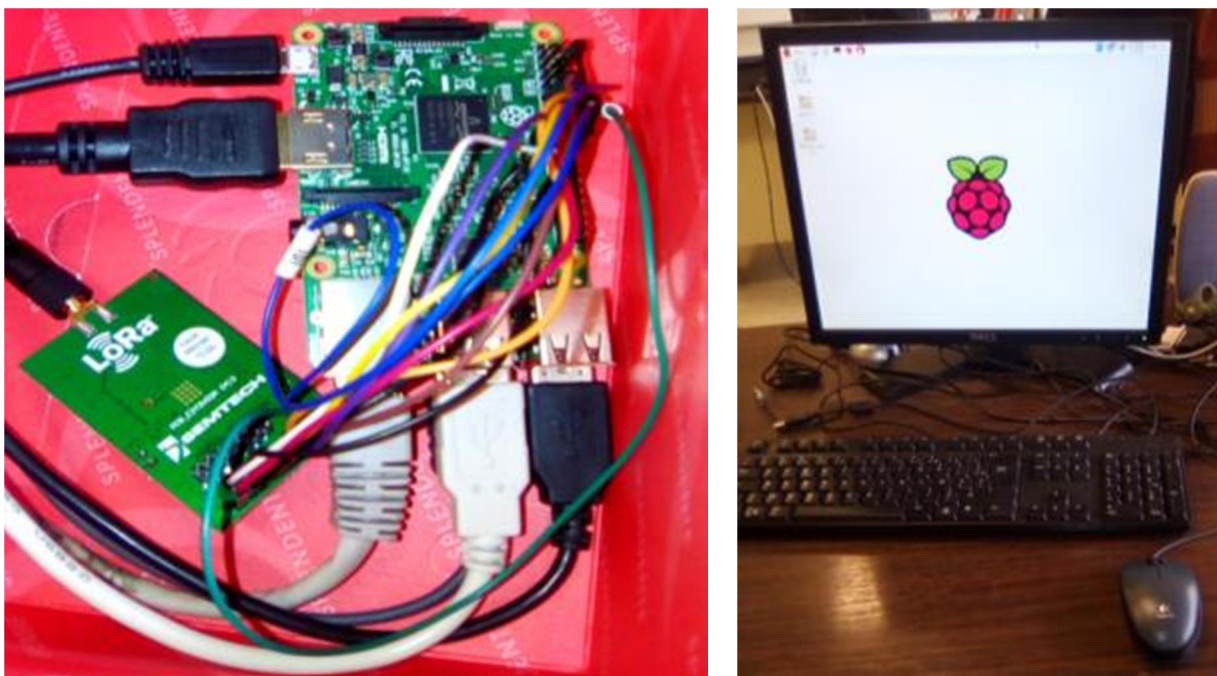


Figure 53 - Station de base à partir de Raspberry Pi 3

7.4.4 Protocole de communication

Lors de déploiement du réseau de capteurs, la station de base préalablement configurée allume sa radio pour recevoir les messages provenant des nœuds de capteurs. Les nœuds de capteurs sont configurés et activés à leur tour. Lors de la configuration d'un nœud de capteur, une adresse lui est attribuée, de même que les identifiants des cibles qui se trouvent dans son rayon de couverture (car la position du nœud de capteur est connue). L'adresse de la station de base est aussi configurée, c'est la seule adresse à laquelle le nœud peut envoyer un message. Chaque message est composé de deux parties principales : une partie entête (HEADER) et une partie données (DATA). La partie

entête contient le type du message (TYPE), l'adresse de l'émetteur (SRC_ADDR), l'adresse du destinataire (DST_ADDR) et un numéro de séquence (SEQNO). Lorsqu'un nœud de capteur est activé, il envoie un message de type JOIN à la station de base pour s'identifier auprès de la station de base et se faire enregistrer. La partie DATA du message JOIN contient la liste (TAG_LIST) et le nombre (TAG_NB) de cibles que le nœud de capteur couvre. La station de base informe le nœud de capteur de son enregistrement par l'envoi d'un message ACK_JOIN dont la partie DATA contient le temps (START_TIME) après lequel le nœud de capteur devra recontacter pour connaître son statut en envoyant un message de type NODE_STATE. La durée de ce temps d'attente est donnée en paramètre par l'utilisateur et permet à la station de base d'enregistrer tous les capteurs, de former d'éventuels couvrants pour la gestion de l'énergie du réseau et de lancer le démarrage effectif de la surveillance. Cela correspond à la période de configuration du réseau. A la fin de la période de configuration du réseau, la station de base exécute un algorithme du MLCP pour construire un couvrant. Cet algorithme permet de sélectionner les nœuds de capteurs qui seront activés tandis que les autres seront en veille. Deux statuts sont définis pour déterminer le mode de fonctionnement d'un nœud: le statut Veille et le statut Actif. Un capteur a le statut Actif s'il appartient au couvrant formé et le statut Veille sinon. Deux périodes sont fixées par l'utilisateur : la période NODE_STATE_PER et la période DATA_PER. La période NODE_STATE_PER détermine le temps après lequel le nœud de capteur doit envoyer un message de type NODE_STATE à la station de base. Ce message informe la station de base de l'énergie résiduelle (ENERGIE) du nœud de capteur. Ce message est envoyé par tous les capteurs (en veille et actifs). C'est le seul type de message envoyé par les capteurs en veille. La période DATA_PER détermine le temps après lequel un nœud de capteur actif envoie un message de type SAMPLE qui contient les mesures des phénomènes observés (température, monoxyde de carbone etc.). Cette période correspond à la période d'échantillonnage du système. La partie DATA d'un message de type SAMPLE est constitué des champs TEMP, CO, SMOKE et GPL contenant respectivement les mesures de température (en degré Celsius), de monoxyde de carbone, de fumée et de GPL (en ppm). Lorsqu'un message de type SAMPLE est envoyé par un nœud de capteur, celui-ci attend un message d'acquiescement de type ACK de la station de base. La partie DATA d'un message de type ACK contient le numéro de séquence (SEQNO) du message qui est acquitté et un champ ALARM qui demande au nœud de déclencher ou d'arrêter son alarme. La station de base envoie un message de type ACK dont le champ ALARM vaut 1 (ON) lorsqu'une des mesures envoyées par le nœud de capteur dépasse le seuil défini. A la réception de ce message, le nœud de capteur fait sonner son alarme (buzzer) tout en continuant d'envoyer les mesures toutes les DATA_PER unités de temps. Le nœud de capteur arrête son alarme lorsqu'il reçoit un message de type ACK dont le champ ALARM vaut 0 (OFF), preuve que les mesures sont retombées en dessous des seuils définis. Lorsqu'un message de type NODE_STATE est envoyé par un nœud de capteur, la station de base répond en envoyant un message de type NETWORK_CONFIG. La partie DATA d'un message de ce type contient les valeurs des périodes DATA_PER et NODE_STATE_PER, ainsi que le statut (STATUT) du nœud de capteur qui varie en fonction du couvrant courant. Un nouveau couvrant est construit chaque fois qu'un capteur actif (statut = Actif) a épuisé son énergie ou est tombé en panne. Un message de type NETWORK_CONFIG permet à la station de base d'informer un nœud d'un éventuel changement de statut. Ainsi, un nœud de capteur de statut Veille peut passer au statut Actif après réception d'un message de type NETWORK_CONFIG et vice-versa. La valeur de la période NODE_STATE_PER doit être judicieusement choisie par l'utilisateur car elle détermine le

délai maximal pour informer un nœud de capteur en veille de son changement de statut. Après chaque message envoyé par un nœud de capteur, un message est attendu de la part de la station de base en guise de réponse. Tant que le nœud de capteur ne reçoit pas de réponse, il attend pendant une durée aléatoire et retransmet le message. La figure 54 récapitule les formats des différents messages et les interactions entre la station de base et un nœud de capteur.

Format général d'un message

HEADER	DATA
--------	------

Format de la partie entête (Header)

TYPE	DST_ADDR	SRC_ADDR	SEQNO
------	----------	----------	-------

TYPE = {JOIN, ACK_JOIN, NODE_STATE, NETWORK_CONFIG, SAMPLE, ACK }

Format d'un message de type JOIN

HEADER	TAG_NB	TAG_LIST
--------	--------	----------

Format d'un message de type ACK_JOIN

HEADER	START_TIME
--------	------------

Format d'un message de type NODE_STATE

HEADER	ENERGY
--------	--------

Format d'un message de type NETWORK_CONFIG

HEADER	NODE_STATE_PER	DATA_PER	STATUT
--------	----------------	----------	--------

Format d'un message de type SAMPLE

HEADER	TEMP	CO	SMOKE	GPL
--------	------	----	-------	-----

Format d'un message de type ACK

HEADER	SEQNO	ALARM
--------	-------	-------

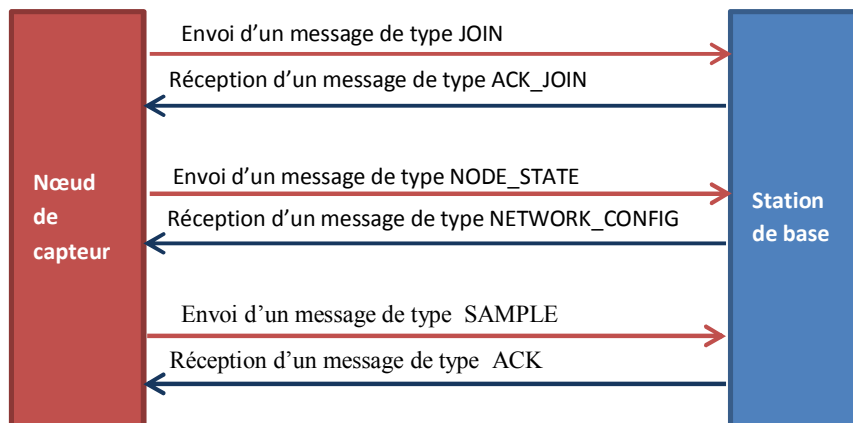


Figure 54 - Eléments du protocole de communication

7.4.5 Base de données et application Web

Les données provenant des nœuds de capteurs sont envoyées par la station de base vers le serveur situé au centre de contrôle et de surveillance. Sur ce serveur est installé un gestionnaire de base de données (SGBD) pour le stockage et la manipulation des données reçues. La base de données est exploitée par une application Web pour la visualisation des données sous forme de graphe et pour la

gestion des alertes. Le SGBD choisi pour ce système est PostgreSQL et l'application Web est Grafana. La figure 55 présente cette architecture logicielle.

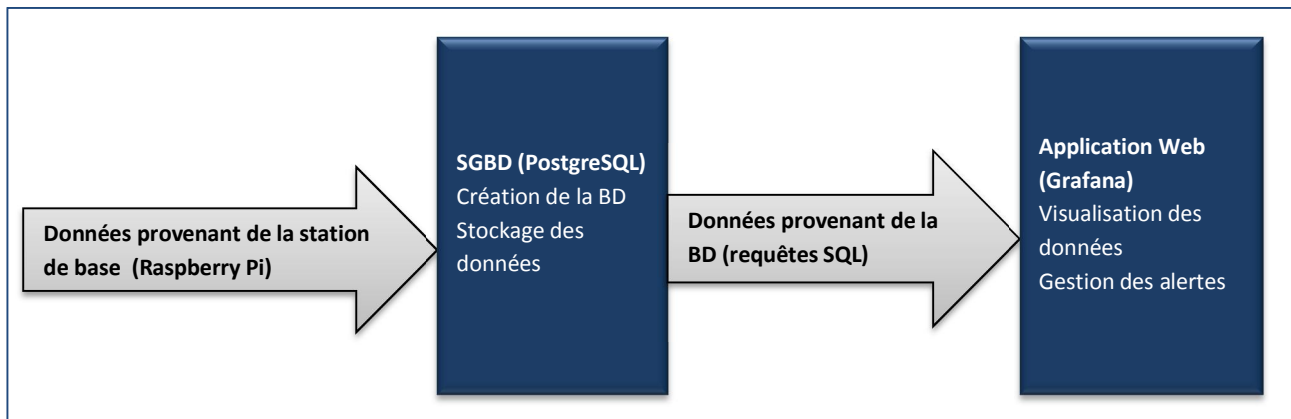


Figure 55 - Architecture logicielle du système

PostgreSQL [96] est un système de gestion de bases de données relationnel robuste et puissant, aux fonctionnalités riches et avancées et capable de manipuler en toute fiabilité de gros volumes de données. Il a été prouvé que PostgreSQL est hautement évolutif, à la fois dans la quantité de données qu'il peut gérer et dans le nombre d'utilisateurs simultanés qu'il peut gérer. PostgreSQL est un logiciel libre et publié sous la licence Open Source PostgreSQL qui permet d'utiliser, copier, modifier et distribuer ce logiciel et sa documentation à quelque fin que ce soit, sans frais et sans accord écrit, à condition de mentionner les créateurs de PostgreSQL. PostgreSQL fonctionne sur les principaux systèmes d'exploitation : Linux, MacOS, Solaris et Windows. Il est extrêmement respectueux des standards, se conformant à plus de 160 sur les 179 fonctionnalités obligatoires de la norme SQL:2011 depuis sa version 10 en 2017. Il supporte plus d'une douzaine de langages de programmation, dont Java, Perl, Python, Ruby, Tcl, C / C ++. PostgreSQL fonctionne selon une architecture client/serveur, il est ainsi constitué :

- **d'une partie serveur**, c'est-à-dire une application fonctionnant sur la machine hébergeant la base de données (le serveur de bases de données) capable de traiter les requêtes des clients. Le serveur de base de données PostgreSQL est *postgres*. Pour qu'une application cliente puisse accéder à une base de données, elle se connecte (via un réseau ou localement) à une instance *postgres* en cours d'exécution. L'instance *postgres* démarre alors un processus serveur distinct pour gérer la connexion.
- **d'une partie client** installée sur toutes les machines nécessitant d'accéder au serveur de base de données. Dans notre système, un client PostgreSQL est installé sur le Raspberry Pi et sur le serveur du centre de contrôle (un client PostgreSQL peut fonctionner sur la même machine que le serveur PostgreSQL). Les clients (les machines sur lesquelles le client PostgreSQL est installé) interrogent le serveur de bases de données à l'aide de requêtes SQL.

La structure de la base de données créée pour ce système est composée d'une table représentée à la figure 56. Cette table comporte sept champs avec pour clé primaire le champ *sample_id* qui est l'identifiant d'une prise de mesure reçue par la station de base. Le champ *sample_date* représente l'instant (la date et l'heure) de la réception de la prise de mesure par la station de base. Le champ

node_id est l'identifiant du nœud de capteur qui a effectué la prise de mesure et les champs *CO*, *TEMP*, *SMOKE* et *GPL* représentent respectivement les mesures de monoxyde de carbone, de température, de fumée et de gaz GPL prélevées par le nœud de capteur.

SAMPLE
Sample_id: int primary key
Node_id: varchar (10)
Sample_date: timestamp
CO: real
TEMP: real
SMOKE: real
GPL: real

Figure 56 - Structure de la base de données

Grafana [97] est un logiciel open source d'analyse et de visualisation de données qui dépendent du temps. Il est le plus souvent utilisé pour visualiser des données de séries chronologiques pour l'analyse des infrastructures et des applications, mais aussi dans d'autres domaines, notamment les capteurs industriels, la domotique, la météo et le contrôle des processus. Il est très bien adapté pour visualiser des mesures en provenance de capteurs ou d'objets connectés. Grafana expose dans des tableaux de bord paramétrables les métriques brutes ou agrégées provenant de multiples sources. Il dispose de capacités de présentation très poussées et permet de visualiser les métriques sous forme de tableaux et de graphes de plusieurs types (courbes, histogrammes, camembert,...). Grafana supporte de nombreuses sources de données, entre autres les bases de données de séries temporelles (Graphite, InfluxDB, OpenTSDB, ...) et les bases de données MySQL et PostgreSQL. Grafana est un logiciel libre et open source sous licence Apache qui autorise la modification et la distribution du code sous toute forme et oblige le maintien du copyright lors de toute modification. La première version de Grafana date de janvier 2014 et depuis sa version 4 (actuellement version 5), Grafana permet de placer des alarmes sur des métriques en définissant de façon simple des seuils d'alertes et les actions associées notamment l'envoi de courriers électroniques. Lorsqu'une alerte est définie, une ligne rouge matérialise naturellement les seuils sur les graphes. Très peu de configurations sont requises pour utiliser les nombreux canaux de notifications disponibles. Pour activer les notifications par courrier électronique par exemple, il suffit de configurer les paramètres SMTP dans la configuration Grafana. Les notifications par e-mail téléchargent une image du graphique d'alerte et l'insère sous forme de pièce jointe au courrier électronique.

Pour ce système, Grafana est installé sur le serveur du centre de contrôle et configuré pour la visualisation en temps réel des données gérées par le SGBD PostgreSQL installé sur la même machine. La figure 57 présente une interface Grafana sur laquelle est affiché un tableau de bord qui contient des courbes des mesures de température provenant de deux nœuds de capteurs. Les points sur les courbes représentent les données reçues. Les valeurs minimales, maximales et moyennes, de même que les valeurs courantes sont affichées en dessous des courbes. Le seuil de l'alerte est représenté par la ligne rouge. Lorsque ce seuil est franchi, un courrier électronique contenant une image des courbes est envoyé aux utilisateurs. Ce tableau de bord est configuré pour afficher les données de la journée en cours et rafraîchir les graphes toutes les minutes (en haut et à droite de

l'interface). Cette configuration peut être changée en un clic. La figure 58 présente plus clairement les courbes de température sur un intervalle d'une heure. La simplicité d'utilisation de Grafana permet d'ajouter facilement de nouveaux graphes au tableau de bord. De plus, cette interface permet à tout utilisateur du système de visualiser les graphes pour différentes périodes de temps (jour, semaine, mois, année, etc.) et de gérer les alertes.

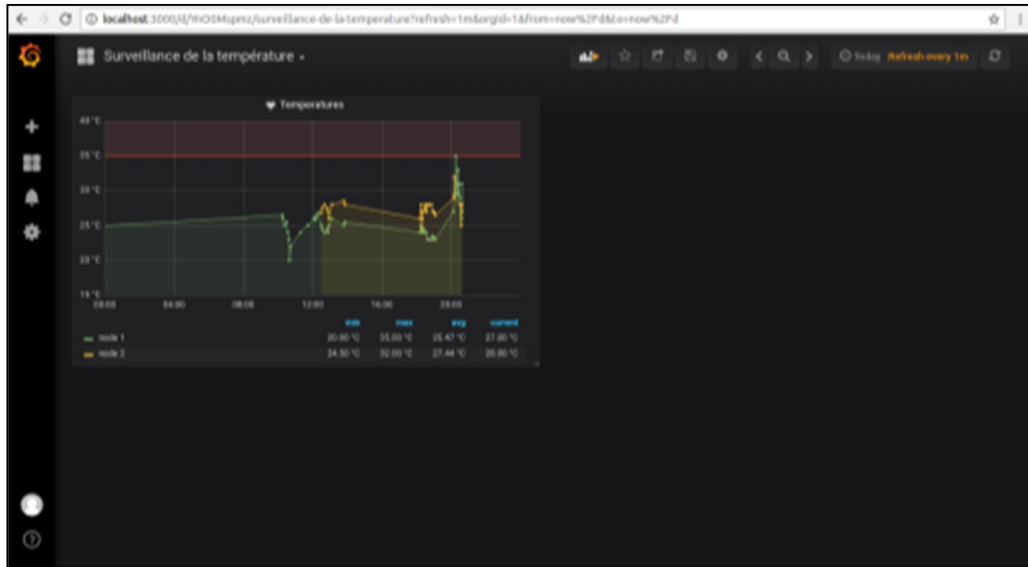


Figure 57 - Une interface Grafana

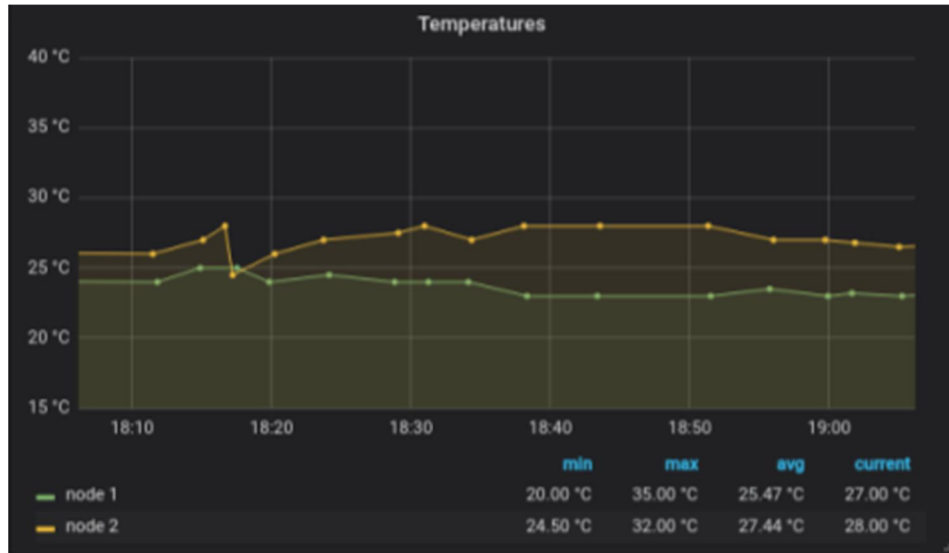


Figure 58 - Visualisation des métriques

7.5 Contrainte d'énergie dans les systèmes de surveillance de la qualité de l'air

La contrainte d'énergie dans les systèmes TNGAPMS varie en fonction du type considéré. Dans les systèmes de type VSN (Vehicle Sensor Network), la contrainte d'énergie est inexistante car chaque nœud de capteurs est alimenté par la batterie du véhicule sur lequel il est installé. De même, dans

les systèmes CSN (Community Sensor Network), la contrainte énergétique n'est pas très importante car l'utilisateur qui transporte le nœud de capteur peut remplacer ou recharger sa batterie autant que de fois que nécessaire. Dans les systèmes SSN (Static Sensor Network) où la position des nœuds permet de les connecter au réseau électrique ou lorsqu'il est possible de les alimenter par des panneaux solaires, la question de la gestion de l'énergie ne se pose pas. Cela peut être le cas lorsque les capteurs sont déployés dans des marchés, des industries, des maisons d'habitations ou à des carrefours. En effet, il est généralement possible dans ces lieux de connecter les nœuds sur la source d'électricité des lampadaires ou des feux de signalisation, ou sur des prises électriques des bâtiments ou des maisons sous surveillance. Lorsque la position des nœuds et le coût d'installation le permettent, il est possible d'installer des panneaux solaires pour recharger les batteries qui alimentent les nœuds. Cependant, lorsque la zone de déploiement ne permet pas d'accéder au réseau électrique et qu'il n'est pas possible d'alimenter les capteurs par des panneaux, les nœuds de capteurs doivent alors être alimentés par des batteries dont la quantité d'énergie est limitée. Grâce à l'évolution technologique, la capacité des batteries en général et des batteries 9V en particulier, ne cesse de croître. On trouve ainsi sur des sites de vente en ligne comme Amazon et Ebay, des batteries 9V de différentes capacités parmi lesquelles 200 mAh, 500 mAh, 600 mAh, 950 mAh, 1100 mAh et 2000 mAh. Un nœud de capteur qui consomme en moyenne 1 mW peut fonctionner pendant environ 2 mois avec une batterie 9V de 200 mAh, 7 mois avec une batterie 9V de 600 mAh, 1 an avec une batterie 9V de 1100 mAh et 2 ans avec une batterie 9V de 2000 mAh. Ainsi, un système de surveillance de la qualité de l'air peut fonctionner pendant 2 ans sans besoin de remplacer les batteries si les nœuds de capteurs consomment 1 mW et sont alimentés par des batteries 9V de 2000 mAh. Cette puissance de fonctionnement est tout à fait réaliste comme le montre l'expérience décrite ci-dessous.

Considérons un nœud de capteur composé d'un microcontrôleur ATmega328, d'un modem LoRa SX1276, et d'un capteur de température DS18B20. Mesurons la puissance moyenne consommée par ce capteur lorsqu'il est utilisé dans le système de surveillance proposé dans la section précédente (ce nœud ne transmet que des mesures de température). Pour cela, nous définissons quatre modes de consommation : SLEEP, CPU, TX, RX. Dans le mode SLEEP, la radio et le microcontrôleur sont en veille. En mode CPU, la radio est en veille et le microcontrôleur est actif. En mode TX, la radio est en mode transmission (+20 dBm) et le microcontrôleur en mode actif. Enfin, en mode RX, la radio est en mode réception et le microcontrôleur en mode actif. Nous ne prenons pas en compte la consommation de capteur de température car elle est négligeable par rapport aux autres éléments. Le tableau 4 montre la puissance consommée dans chaque mode, sachant que le voltage est de 5V.

Tableau 4 – Modes de fonctionnement d'un nœud de capteur

Mode	Puissance (en mW)
SLEEP	0.0335
CPU	5.501
TX	605.5
RX	65.5

Afin d'obtenir la puissance moyenne du capteur, nous initialisons une variable représentant l'énergie disponible (E). Durant le fonctionnement du nœud de capteur dans le système de surveillance, nous mesurons le temps de passage dans chaque mode. A la sortie d'un mode, nous multiplions le temps de séjour par la puissance correspondante à ce mode pour calculer l'énergie qui est consommée. Cette énergie est déduite de l'énergie initiale et le nœud de capteur continue de fonctionner si l'énergie résiduelle est non nulle. Lorsque l'énergie devient nulle après une durée de fonctionnement T , on peut alors calculer la puissance moyenne $P = E/T$. Nous réalisons l'expérience pour les deux statuts définis par le système à savoir statut Actif et statut Veille. Le tableau 5 montre les puissances correspondantes à différentes valeurs des paramètres DATA_PER et NODE_STATE_PER.

Tableau 5 – Consommation d'un nœud de capteur et ratio d'énergie

Paramètres du système		Puissance capteur Actif (P_a)	Puissance capteur Veille (P_v)	$r = \frac{P_v}{P_a}$	$1/r$
DATA_PER	NODE_STATE_PER				
1 min	2 min	1.3 mW	0.59 mW	0.45	2.2
1 min	5 min	1.16 mW	0.24 mW	0.2	4.8
2 min	5 min	0.63 mW	0.24 mW	0.38	2.6
2 min	10 min	0.6 mW	0.16 mW	0.27	3.75
5 min	10 min	0.32 mW	0.16 mW	0.5	2

Le statut par défaut d'un nœud de capteur dans le système de surveillance est le statut actif. Nous observons sur le tableau 5 que la puissance consommée par un nœud de capteur actif est d'environ 1mW lorsqu'il envoie les mesures chaque minute. Lorsque cette période d'envoi des mesures augmente, la puissance consommée devient inférieure à 1 mW. Par exemple, la puissance du nœud de capteur est de 0.6 mW environ lorsqu'une mesure est envoyée toutes les 2 minutes et chute à 0.3 mW lorsque le nœud de capteur envoie les données toutes les 5 minutes. Ces valeurs de puissance varient faiblement si on ajoute à ce nœud des capteurs de gaz basse consommation tel que le capteur de monoxyde de carbone Figaro CMM5042 [98] qui consomme 200 μA .

Bien que les capacités des batteries actuelles permettent de faire les nœuds de capteurs des systèmes SSN pour des périodes allant de plusieurs mois à quelques années, il est possible d'augmenter davantage l'autonomie de ces systèmes en utilisant un algorithme du MLCP. Pour cela, le nombre et la position des nœuds de capteurs déployés doivent être tel que chaque point à surveiller soit couverte par plus d'un nœud de capteurs. Lorsque les capteurs sont redondants, un algorithme du MLCP sélectionne les capteurs qui fonctionnent dans le mode Actif tandis que les autres sont en mode Veille. Considérons une expérience où l'on utilise le nœud de capteur de température décrit plus haut pour surveiller la température d'une citerne à l'aide de notre système de surveillance. Ce nœud de capteur peut fonctionner pendant un an et demi lorsqu'il transmet des mesures toutes les minutes ($P_a = 1.3 \text{ mW}$). Au lieu de placer un seul nœud de capteur dans la citerne, plaçons-en deux. Grâce à l'algorithme d'optimisation d'énergie (du MLCP), un nœud de capteur est mis par le système en mode Actif avec une consommation $P_a = 1.3 \text{ mW}$ tandis que l'autre est mis en mode

veille ($P_v = 0.59 \text{ mW}$) et envoie un message à propos de son état toutes les 2 minutes. Lorsque le premier nœud arrive à court d'énergie (un an et demi plus tard), le second est placé en mode actif et envoie les mesures toutes les minutes pendant environ 10 mois avant d'être à court d'énergie. De cette façon, le système fonctionne pendant 2 ans et 4 mois avec deux nœuds de capteurs au lieu d'un an et demi avec un seul nœud de capteur. Augmenter l'autonomie du système est très utile lorsque la zone de déploiement est difficile d'accès ou dangereuse (présence de gaz toxique) et que l'on souhaite réduire le plus possible le nombre d'opérations sur le terrain. Cependant, la durée de l'autonomie obtenue grâce à cette méthode d'optimisation est limitée. En effet, la puissance consommée par un capteur en mode veille limite la durée de sa batterie. Ainsi, cette méthode d'optimisation permet de multiplier la durée de vie du système d'un facteur au plus égal à $\frac{1}{r} = \frac{P_v}{P_a}$. Dans l'exemple de la citerne, même en plaçant un très grand nombre de nœuds dans la citerne, il n'est pas possible que le système soit autonome en énergie pendant plus de 3.3 ans (car $\frac{1}{r} = 2.2$). Supposons que l'on place 5 capteurs dans la citerne, le premier est actif pendant 1 an et demi, le second pendant 10 mois, le troisième pendant 6 mois, le quatrième pendant 2 mois et demi et le cinquième pendant 1 mois et 10 jours, pour un total de surveillance de 3.15 ans sans remplacement de batteries. La dernière colonne du tableau 5 donne le facteur de gain maximal en termes d'autonomie d'énergie que cette méthode d'optimisation permet d'atteindre en fonction de différentes valeurs des paramètres du système. Outre l'augmentation de l'autonomie, la redondance des nœuds de capteurs permet aussi d'augmenter la robustesse du système. En cas de panne, un nœud de capteur cesse de fonctionner bien que sa batterie ne soit pas épuisée. Lorsque cette situation se produit, l'algorithme du MLCP permet au système de continuer à fonctionner jusqu'à ce que tous les nœuds redondants cessent de fonctionner à leur tour. Par exemple, si le seul nœud dans la citerne tombe en panne après 2 mois, le système cesse de fonctionner de suite. Par contre, s'il y a un second capteur, le système lui passe le relais et continue de fonctionner pendant 1 an et 5 mois supplémentaires sans intervention humaine.

L'autonomie apportée par l'utilisation d'un tel algorithme d'optimisation de l'énergie permet de réduire les coûts car une fois le réseau déployé, la fréquence des opérations de maintenance du système est réduite.

7.6 Conclusion

Dans ce chapitre, nous nous sommes intéressés aux réseaux de capteurs sans fil appliqués à la surveillance de la pollution atmosphérique. Les systèmes de surveillance de la qualité de l'air basés sur les réseaux de capteurs sans fil sont moins coûteux que les systèmes de surveillance conventionnels et sont de ce fait adaptés aux pays en voie de développement en général et au Cameroun en particulier. Bien que la précision de ces systèmes soit plus faible que celles des systèmes conventionnels, ils permettent toutefois de collecter des mesures qui sont très utiles pour la lutte contre la pollution de l'air. Plusieurs systèmes de surveillance de la qualité de l'air ont été proposés dans la littérature. En général, ces systèmes prélèvent les mesures des polluants dans l'air et les envoient sur des serveurs connectés à Internet. Ces données stockées sur une base de données sont affichées sous forme de graphiques ou de cartes. Nous avons implémenté un système de surveillance selon ce schéma général. Les nœuds de capteurs sont réalisés à base de carte Arduino et d'un modem LoRa qui permet une communication longue distance entre les nœuds de capteurs

et la station de base. La station de base est réalisée à l'aide d'un Raspberry Pi associé à un modem LoRa et dotée d'une carte WiFi. Le serveur situé au centre de surveillance intègre une base de données PostgreSQL et une application web Grafana qui permet de visualiser les données et d'envoyer des courriers électroniques en cas d'alerte. En plus de surveiller le niveau de monoxyde de carbone dans l'air, notre système permet de lutter contre les incendies grâce à la détection de gaz inflammable GPL, de fumée et de température. De plus, notre système intègre un algorithme du MLCP afin d'augmenter l'autonomie énergétique du réseau de capteur lorsque les nœuds de capteurs sont redondants. Le facteur de gain en autonomie dépend de la consommation des nœuds et est limité par le ratio entre la puissance d'un capteur actif et celle d'un capteur en veille. L'utilisation de l'algorithme MLCP permet en outre d'augmenter la robustesse du système en cas de panne d'un nœud ou de plusieurs nœuds de capteurs.

Chapitre 8 Conclusion générale

8.1 Rappel des enjeux

L'objectif de cette thèse était de proposer des solutions permettant de prolonger la durée de vie d'un réseau de capteurs sans fil où les nœuds sont statiques et déployés pour couvrir un ensemble de points fixes appelés cibles. Les nœuds du réseau communiquent directement (en un saut) avec la station de base grâce à une technologie longue portée et de faible consommation. De ce fait, les contraintes de connexité et de routage ne sont pas prises en compte. Par ailleurs, les nœuds doivent être déployés de façon redondante, c'est-à-dire en nombre supérieur par rapport au nombre de cibles. L'approche veille/activité est celle que nous avons considérée. Elle consiste à mettre un sous-ensemble de nœuds dans un mode de faible consommation (mode veille), tandis que les autres sont dans un mode de fonctionnement normal (mode actif). Ainsi, il était question dans cette thèse de proposer des méthodes pour former des sous-ensembles de nœuds de capteurs qui sont activés successivement pendant des durées définies tandis que les autres nœuds sont en veille de sorte à maximiser la durée pendant laquelle chaque cible sera couverte par au moins un nœud capteur. Ce problème NP-difficile est connu sous le nom de MLCP (Maximum Lifetime Coverage Problem en anglais). Contrairement aux heuristiques de la littérature qui s'intéressent uniquement aux instances du MLCP où les nœuds ont la même durée de vie, notre objectif était de proposer des heuristiques pour toute instance du MLCP, indépendamment des durées de vie des nœuds de capteurs. Par ailleurs, les travaux antérieurs considèrent que la consommation des nœuds placés en veille est négligeable par rapport à celle des nœuds actifs. Ce qui n'est pas une hypothèse réaliste. Un autre objectif était de proposer une heuristique pour le MLCP qui tient compte de la consommation des nœuds en veille. Ces heuristiques, de même que celles de la littérature ne donnent pas de solutions satisfaisantes pour les instances du MLCP où le réseau a la forme d'un anneau. Il était question de proposer une méthode permettant de trouver des solutions plus proches de l'optimum pour cette famille d'instances. Enfin, il était aussi question dans ce travail d'implémenter un système de surveillance de la qualité de l'air et de détection d'incendie basé sur les réseaux de capteurs et d'y intégrer les solutions du MLCP afin d'améliorer l'autonomie énergétique de ce système.

8.2 Contributions

En considérant une hypothèse plus réaliste sur les durées de vie des capteurs, à savoir que chaque capteur a sa durée de vie propre, nous avons proposé une nouvelle approche de résolution du MLCP appelée approche réactive. Contrairement à l'approche classique qui consiste à construire les couvrants en une étape et à les activer dans une seconde étape, l'approche réactive ne construit un nouveau couvrant que lorsque le couvrant actif est défectueux. Cette approche a l'avantage d'être plus adaptée à l'environnement réel notamment en cas de panne d'un capteur. En s'appuyant sur l'approche réactive, nous avons proposé deux nouveaux algorithmes de construction des couvrants: l'algorithme adaptatif et l'algorithme blacklist. L'idée de base de l'algorithme adaptatif est de

construire un nouveau couvrant en remplaçant le capteur défectueux dans le couvrant précédent. L'algorithme blacklist quant à lui utilise une liste noire afin de minimiser le nombre de capteurs qui couvrent une cible critique dans un couvrant. Les simulations réalisées permettent de constater que ces nouveaux algorithmes ont une performance supérieure à celles des anciens algorithmes de la littérature. Bien que les deux nouveaux algorithmes trouvent la solution optimale dans 75% des cas simulés, l'algorithme blacklist est meilleur que l'algorithme adaptatif.

Après avoir montré qu'il n'est pas réaliste de supposer que l'énergie consommée par les capteurs en veille est nulle, nous avons introduit la consommation en veille dans le MLCP. Nous considérons que le ratio entre l'énergie consommée par un capteur en veille et l'énergie consommée par un capteur actif est $r = \frac{P_v}{P_a}$ où P_v et P_a sont les puissances moyennes respectives d'un capteur en veille et d'un capteur actif, tel que $r \in [0; 1]$. Nous avons énoncé une proposition pour déterminer une borne supérieure de la durée de vie optimale du réseau. Cette proposition stipule que la durée de couverture maximale d'une cible est atteinte lorsque les capteurs qui la couvrent sont activés dans l'ordre croissant de leurs énergies. Nous avons proposé un nouvel algorithme basé sur cette proposition: l'algorithme Energy-blacklist. L'algorithme Energy-blacklist tient compte de l'énergie consommée par les capteurs en veille dans la construction des couvrants. Cet algorithme est invoqué à l'étape 1 de l'approche réactive qui dans ce cas, collecte les énergies résiduelles de tous les capteurs (actifs et en veille). L'algorithme Energy-blacklist utilise aussi l'idée de la liste noire. Les résultats des simulations ont montré que l'algorithme Energy-blacklist est meilleur que les autres algorithmes lorsque la consommation des capteurs en veille est non nulle. De plus, l'algorithme Energy-blacklist donne un ratio à l'optimum supérieur à 0.9 dans 75% quels que soient l'énergie consommée en veille, le nombre de capteurs du réseau et la dispersion des énergies de ces capteurs.

Nous avons étudié une famille d'instances pour laquelle aucune des heuristiques proposées ne trouve des solutions proches de l'optimum. Il s'agit de la classe des réseaux en anneau impair. Après avoir défini cette famille d'instances, nous avons démontré que lorsque les nœuds ont la même énergie initiale, toute solution optimale d'une instance de cette famille ne laisse aucune énergie résiduelle sur les nœuds. En se basant sur cette propriété et sur les automorphismes de graphe, nous avons proposé une méthode analytique qui trouve la solution optimale pour les réseaux en anneau de 3 capteurs et 3 cibles. Nous avons conjecturé que cette solution est optimale pour toute instance de la famille des réseaux en anneau impair. Nous avons ensuite montré que cette solution est meilleure que celle donnée par les algorithmes basés sur l'approche réactive et par les algorithmes de la littérature. Puisque cette méthode analytique ne s'applique qu'aux instances où les capteurs ont la même énergie initiale, nous avons proposé une nouvelle heuristique appelée OddRingNetwork. Cette heuristique s'appuie sur la méthode analytique proposée et permet d'obtenir une solution pour toute instance de la famille des réseaux en anneau impair quelles que soient les énergies initiales des capteurs.

Enfin, nous avons implémenté un système de surveillance de la qualité de l'air et de détection d'incendie. Ce système prélève les mesures des polluants dans l'air et les envoie sur des serveurs connectés à Internet. Ces données stockées sur une base de données sont affichées sous forme de graphiques. Les nœuds de capteurs sont réalisés à base de carte Arduino et d'un modem LoRa qui

permet une communication longue distance entre les nœuds de capteurs et la station de base. Les éléments mesurés par un nœud de capteur sont le monoxyde de carbone, la température, la fumée et le gaz GPL. La station de base est réalisée à l'aide d'un Raspberry Pi associé à un modem LoRa et dotée d'une carte WiFi. Le serveur situé au centre de surveillance intègre une base de données PostgreSQL et une application web Grafana qui permet de visualiser les données et d'envoyer des courriers électroniques en cas d'alerte. Ce système intègre un algorithme du MLCP qui permet d'augmenter la robustesse et l'autonomie énergétique du réseau de capteurs lorsque les nœuds de capteurs sont redondants. Le gain en autonomie dépend de la consommation des nœuds et est limité par le ratio entre la puissance d'un capteur actif et celle d'un capteur en veille.

8.3 Perspectives

Une perspective importante de ce travail serait de trouver des méthodes pour réduire la consommation des nœuds de capteurs en veille. Comme nous l'avons noté, le gain en autonomie qu'un algorithme du MLCP permet d'obtenir est limité par le ratio entre la consommation du nœud actif et celle du nœud en veille. Ainsi, plus la consommation du nœud en veille est faible, plus l'autonomie du système peut être améliorée. Sachant que la seule activité consommatrice d'énergie d'un nœud en veille est la communication radio, déterminer une stratégie pour réduire le nombre de communications tout en garantissant la disponibilité du nœud est une question importante. Une idée serait de varier la fréquence de communication d'un nœud en veille en fonction de la probabilité que le capteur actif cesse de fonctionner. De même, les capteurs en veille pourraient avoir des fréquences de communication différentes correspondant à des niveaux de veille différents. Ainsi, un capteur en veille émettrait toutes les 10 minutes tandis que les autres émettraient toutes les 20 voire 30 minutes par exemple. Les nœuds de capteurs de plus grande fréquence de communication auraient alors une priorité plus élevée que les autres lors de la construction des couvrants.

Dans ce travail, nous nous sommes intéressés aux applications qui requièrent une couverture simple. Une autre perspective serait d'adapter les heuristiques proposées afin de les utiliser dans le cas de la couverture multiple (ou k -couverture), car ces heuristiques ne peuvent pas être utilisées tel quel dans ce contexte. En effet, lorsque chaque cible doit être couverte continuellement par k nœuds de capteurs, construire k couvrants disjoints et les activer en même temps ne constitue pas une solution optimale. Il serait donc important d'étendre ces heuristiques pour le cas général de la couverture multiple.

Bibliographie

- [1] I. F. Akyildiz, Weilian Su, Y. Sankarasubramaniam, et E. Cayirci, « A survey on sensor networks », *IEEE Communications Magazine*, vol. 40, n° 8, p. 102-114, août 2002.
- [2] J. Yick, B. Mukherjee, et D. Ghosal, « Wireless sensor network survey », *Computer Networks*, vol. 52, n° 12, p. 2292-2330, août 2008.
- [3] C. Alcaraz, P. Najera, J. Lopez, et R. Roman, « Wireless Sensor Networks and the Internet of Things: Do We Need a Complete Integration? », p. 8.
- [4] M. R. Palattella et al., « Internet of Things in the 5G Era: Enablers, Architecture, and Business Models », *IEEE Journal on Selected Areas in Communications*, vol. 34, n° 3, p. 510-527, mars 2016.
- [5] A. Whitmore, A. Agarwal, et L. Da Xu, « The Internet of Things—A survey of topics and trends », *Information Systems Frontiers*, vol. 17, n° 2, p. 261-274, avr. 2015.
- [6] R. Piyare et S. R. Lee, « Towards Internet of Things (IOTS): Integration of Wireless Sensor Network to Cloud Services for Data Collection and Sharing », *International journal of Computer Networks & Communications*, vol. 5, n° 5, p. 59-72, sept. 2013.
- [7] M. Swan, « Sensor Mania! The Internet of Things, Wearable Computing, Objective Metrics, and the Quantified Self 2.0 », *Journal of Sensor and Actuator Networks*, vol. 1, n° 3, p. 217-253, nov. 2012.
- [8] H. Karl et A. Willig, *Protocols and Architectures for Wireless Sensor Networks*. Wiley, 2006.
- [9] D. Puccinelli et M. Haenggi, « Wireless sensor networks: applications and challenges of ubiquitous sensing », *IEEE Circuits and Systems Magazine*, vol. 5, n° 3, p. 19-31, 2005.
- [10] T. Arampatzis, J. Lygeros, et S. Manesis, « A Survey of Applications of Wireless Sensors and Wireless Sensor Networks », in *Proceedings of the 2005 IEEE International Symposium on, Mediterrean Conference on Control and Automation Intelligent Control, 2005.*, Limassol, Cyprus, 2005, p. 719-724.
- [11] C. F. García-Hernández, P. H. Ibargüengoytia-González, J. García-Hernández, et J. A. Pérez-Díaz, « Wireless Sensor Networks and Applications: a Survey », p. 10, 2007.
- [12] S. Sharma, R. K. Bansal, et S. Bansal, « Issues and Challenges in Wireless Sensor Networks », in *2013 International Conference on Machine Intelligence and Research Advancement*, Katra, India, 2013, p. 58-62.
- [13] I. Dietrich et F. Dressler, « On the lifetime of wireless sensor networks », *ACM Transactions on Sensor Networks*, vol. 5, n° 1, p. 1-39, févr. 2009.
- [14] G. Anastasi, M. Conti, M. Di Francesco, et A. Passarella, « Energy conservation in wireless sensor networks: A survey », *Ad Hoc Networks*, vol. 7, n° 3, p. 537-568, mai 2009.
- [15] A. A. Aziz, Y. A. Sekercioglu, P. Fitzpatrick, et M. Ivanovich, « A Survey on Distributed Topology Control Techniques for Extending the Lifetime of Battery Powered Wireless Sensor Networks », *IEEE Communications Surveys & Tutorials*, vol. 15, n° 1, p. 121-144, 2013.
- [16] M. Younis, I. F. Senturk, K. Akkaya, S. Lee, et F. Senel, « Topology management techniques for tolerating node failures in wireless sensor networks: A survey », *Computer Networks*, vol. 58, p. 254-283, janv. 2014.
- [17] « A Survey on Topology Control in Wireless Sensor Networks: Taxonomy, Comparative Study, and Open Issues », *Proceedings of the IEEE*, vol. 101, n° 12, p. 2538-2557, déc. 2013.
- [18] M. T. Thai, F. Wang, D. H. Du, et X. Jia, « Coverage problems in wireless sensor networks: designs and analysis », *International Journal of Sensor Networks*, vol. 3, n° 3, p. 191, 2008.
- [19] G. Fan et S. Jin, « Coverage Problem in Wireless Sensor Network: A Survey », *Journal of Networks*, vol. 5, n° 9, p. 1033-1040, sept. 2010.
- [20] M. Cardei, M. T. Thai, Y. Li, et W. Wu, « Energy-efficient target coverage in wireless sensor networks », in *Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies*, Miami, FL, USA, 2005, p. 1976-1984.
- [21] V. V. Vazirani, *Approximation algorithms*. Springer-Verlag, 2003.
- [22] M. Cardei et D.-Z. Du, « Improving Wireless Sensor Network Lifetime through Power Aware Organization », *Wireless Networks*, vol. 11, n° 3, p. 333-340, mai 2005.
- [23] Chih-Chung Lai, Chuan-Kang Ting, et Ren-Song Ko, « An effective genetic algorithm to improve wireless sensor network lifetime for large-scale surveillance applications », in *2007 IEEE Congress on Evolutionary Computation*, Singapore, 2007, p. 3531-3538.
- [24] Y. Li, X. Hu, et J. Zhang, « A new genetic algorithm for the set k-cover problem in wireless sensor networks », in *2009 IEEE International Conference on Systems, Man and Cybernetics*, San Antonio, TX, USA, 2009, p. 1405-1410.
- [25] S. Slijepcevic et M. Potkonjak, « Power efficient organization of wireless sensor networks », présenté à IEEE International Conference on Communications, Helsinki, Finland, 2001, p. 472-476.
- [26] D. Zorbas, D. Glynos, P. Kotzanikolaou, et C. Douligeris, « Solving coverage problems in wireless sensor networks using cover sets », *Ad Hoc Networks*, vol. 8, n° 4, p. 400-415, juin 2010.
- [27] P. Baronti, P. Pillai, V. W. C. Chook, S. Chessa, A. Gotta, et Y. F. Hu, « Wireless sensor networks: A survey on the state of the art and the 802.15.4 and ZigBee standards », *Computer Communications*, vol. 30, n° 7, p. 1655-1695, mai 2007.
- [28] U. Raza, P. Kulkarni, et M. Sooriyabandara, « Low Power Wide Area Networks: An overview », *IEEE Communications Surveys & Tutorials*, vol. 19, n° 2, p. 855-873, janv. 2017.

- [29] M. Centenaro, L. Vangelista, A. Zanella, et M. Zorzi, « Long-Range Communications in Unlicensed Bands: the Rising Stars in the IoT and Smart City Scenarios », *IEEE Wireless Communications*, vol. 23, oct. 2016.
- [30] B. Vejlggaard, M. Lauridsen, H. Nguyen, I. Z. Kovacs, P. Mogensen, et M. Sorensen, « Interference Impact on Coverage and Capacity for Low Power Wide Area IoT Networks », in *2017 IEEE Wireless Communications and Networking Conference (WCNC)*, San Francisco, CA, USA, 2017, p. 1-6.
- [31] B. Vejlggaard, M. Lauridsen, H. Nguyen, I. Z. Kovacs, P. Mogensen, et M. Sorensen, « Coverage and Capacity Analysis of Sigfox, LoRa, GPRS, and NB-IoT », in *2017 IEEE 85th Vehicular Technology Conference (VTC Spring)*, Sydney, NSW, 2017, p. 1-5.
- [32] N. Sornin, M. Luis, T. Eirich, T. Kramp, et O. Hersent, « LoRa Specification 1.0 ». LoRa Alliance, 2015.
- [33] A.-I. Pop, U. Raza, P. Kulkarni, et M. Sooriyabandara, « Does Bidirectional Traffic Do More Harm Than Good in LoRaWAN Based LPWA Networks? », *arXiv:1704.04174 [cs]*, avr. 2017.
- [34] O. Georgiou et U. Raza, « Low Power Wide Area Network Analysis: Can LoRa Scale? », *IEEE Wireless Communications Letters*, vol. 6, n° 2, p. 162-165, avr. 2017.
- [35] C. Alippi, M. D. Francesco, M. Roveri, et M. D. Francesco, « Energy Management in Wireless Sensor Networks with Energy-hungry Sensors », *IEEE Instrumentation and measurement Magazine*, vol. 6, n° 2, p. 66-23, mars 2009.
- [36] A. Kansal, J. Hsu, S. Zahedi, et M. B. Srivastava, « Power management in energy harvesting sensor networks », *ACM Transactions on Embedded Computing Systems*, vol. 6, n° 4, p. 32-es, sept. 2007.
- [37] W. K. G. Seah, Z. A. Eu, et H.-P. Tan, « Wireless sensor networks powered by ambient energy harvesting (WSN-HEAP) - Survey and challenges », in *2009 1st International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace & Electronic Systems Technology*, Aalborg, Denmark, 2009, p. 1-5.
- [38] S. Sudevalayam et P. Kulkarni, « Energy Harvesting Sensor Nodes: Survey and Implications », *IEEE Communications Surveys & Tutorials*, vol. 13, n° 3, p. 443-461, 2011.
- [39] V. Potdar, A. Sharif, et E. Chang, « Wireless Sensor Networks: A Survey », in *2009 International Conference on Advanced Information Networking and Applications Workshops*, Bradford, United Kingdom, 2009, p. 636-641.
- [40] « Arduino Uno Rev3 Datasheet ». [En ligne]. Disponible sur: <https://store.arduino.cc/usa/arduino-uno-rev3>. [Consulté le: 07-sept-2018].
- [41] « SX1276/77/78/79 - 137 MHz to 1020 MHz Low Power Long Range Transceiver Datasheet ». [En ligne]. Disponible sur: https://www.semtech.com/uploads/documents/DS_SX1276-7-8-9_W_APP_V5.pdf. [Consulté le: 07-sept-2018].
- [42] H. Alemdar et C. Ersoy, « Wireless sensor networks for healthcare: A survey », *Computer Networks*, vol. 54, n° 15, p. 2688-2710, oct. 2010.
- [43] G. V. Crosby, « Wireless Body Area Networks for Healthcare: A Survey », *International Journal of Ad hoc, Sensor & Ubiquitous Computing*, vol. 3, n° 3, p. 1-26, juin 2012.
- [44] Y. Hao et R. Foster, « Wireless body sensor networks for health-monitoring applications », *Physiological Measurement*, vol. 29, n° 11, p. R27-R56, nov. 2008.
- [45] P. Corke, T. Wark, R. Jurdak, W. Hu, P. Valencia, et D. Moore, « Environmental Wireless Sensor Networks », *Proceedings of the IEEE*, vol. 7, n° 3, p. 1903-1917, oct. 2010.
- [46] L. M. Oliveira et J. J. Rodrigues, « Wireless Sensor Networks: a Survey on Environmental Monitoring », *Journal of Communications*, vol. 6, n° 2, avr. 2011.
- [47] F. G. Costa, J. Ueyama, T. Braun, G. Pessin, F. S. Osorio, et P. A. Vargas, « The use of unmanned aerial vehicles and wireless sensor network in agricultural applications », in *2012 IEEE International Geoscience and Remote Sensing Symposium*, Munich, Germany, 2012, p. 5045-5048.
- [48] E. Nash, P. Korduan, et R. Bill, « Applications of open geospatial web services in precision agriculture: a review », *Precision Agriculture*, vol. 10, n° 6, p. 546-560, déc. 2009.
- [49] G. Vellidis, M. Tucker, C. Perry, C. Kvien, et C. Bednarz, « A real-time wireless smart sensor array for scheduling irrigation », *Computers and Electronics in Agriculture*, vol. 61, n° 1, p. 44-50, avr. 2008.
- [50] T. Wark et al., « Transforming Agriculture through Pervasive Wireless Sensor Networks », *IEEE Pervasive Computing*, vol. 6, n° 2, p. 50-57, avr. 2007.
- [51] N. Wang, N. Zhang, et M. Wang, « Wireless sensors in agriculture and food industry—Recent development and future perspective », *Computers and Electronics in Agriculture*, vol. 50, n° 1, p. 1-14, janv. 2006.
- [52] A. Festag, A. Hessler, et R. Baldessari, « Vehicle -to-vehicle and road-side sensor communication for enhanced road safety », in *Proceedings of ITS World Congress and Exhibition*, New York, USA, 2008, p. 12.
- [53] C. Tripp Barba, K. Ornelas Aguirre, et M. Aguilar Igartua, « Performance evaluation of a hybrid sensor and vehicular network to improve road safety », in *Proceedings of the 7th ACM workshop on Performance evaluation of wireless ad hoc, sensor, and ubiquitous networks - PE-WASUN '10*, Bodrum, Turkey, 2010, p. 71.
- [54] A. Gaur, B. Scotney, G. Parr, et S. McClean, « Smart City Architecture and its Applications Based on IoT », *Procedia Computer Science*, vol. 52, p. 1089-1094, 2015.
- [55] R. DOMGA K., R. STANICA, M. TCHUENTE, et F. VALOIS, « WARIM : Wireless Sensor Network Architecture for a Reliable Intersection Monitoring », présenté à IEEE 17th International Conference on Intelligent Transportation, Quindao, China, 2014.
- [56] R. DOMGA K., R. STANICA, M. TCHUENTE, et F. VALOIS, « Sensor deployment in wireless sensor networks with linear topology using virtual node concept », *Wireless Networks*, p. 1-16, 2019.
- [57] V. Tang, Y. Zheng, et J. Cao, « An Intelligent Car Park Management System based on Wireless Sensor Networks », in *2006 First International Symposium on Pervasive Computing and Applications*, Urumqi, China, 2006, p. 65-70.
- [58] J. Chinrungrueng, U. Sunantachaikul, et S. Triamlumlerd, « Smart Parking: An Application of Optical Wireless Sensor Network », in *2007 International Symposium on Applications and the Internet Workshops*, Hiroshima, Japan, 2007, p. 66-66.

- [59] D. A. Goel et R. S. Mishra, « Remote Data Acquisition Using Wireless - Scada System », *International Journal of Engineering*, p. 9.
- [60] H. Kim, « Security and Vulnerability of SCADA Systems over IP-Based Wireless Sensor Networks », *International Journal of Distributed Sensor Networks*, vol. 8, n° 11, p. 268478, nov. 2012.
- [61] V. C. Gungor et G. P. Hancke, « Industrial Wireless Sensor Networks: Challenges, Design Principles, and Technical Approaches », *IEEE Transactions on Industrial Electronics*, vol. 56, n° 10, p. 4258-4265, oct. 2009.
- [62] F. Salvadori et al., « Monitoring in Industrial Systems Using Wireless Sensor Network With Dynamic Power Management », *IEEE Transactions on Instrumentation and Measurement*, vol. 58, n° 9, p. 3104-3111, sept. 2009.
- [63] M. S. BenSaleh, S. M. Qasim, A. M. Obeid, et A. Garcia-Ortiz, « A review on wireless sensor network for water pipeline monitoring applications », in *2013 International Conference on Collaboration Technologies and Systems (CTS)*, San Diego, CA, USA, 2013, p. 128-131.
- [64] S. Yoon, W. Ye, J. Heidemann, B. Littlefield, et C. Shahabi, « SWATs: Wireless sensor networks for steamflood and waterflood pipeline monitoring », *IEEE Network*, vol. 25, n° 1, p. 50-56, janv. 2011.
- [65] M. Agalya, S. Nancy, R. Selvarasu, et U. Scholar, « Home automation system using wireless sensor networks », vol. 13, n° 2, p. 5.
- [66] K. Gill, S.-H. Yang, F. Yao, et X. Lu, « A zigbee-based home automation system », *IEEE Transactions on Consumer Electronics*, vol. 55, n° 2, p. 422-430, mai 2009.
- [67] Jeongyeup Paek, K. Chintalapudi, R. Govindan, J. Caffrey, et S. Masri, « A Wireless Sensor Network for Structural Health Monitoring: Performance and Experience », in *The Second IEEE Workshop on Embedded Networked Sensors, 2005. EmNetS-II.*, Sydney, Australia, 2005, p. 1-10.
- [68] K. Chintalapudi et al., « Monitoring civil structures with a wireless sensor network », *IEEE Internet Computing*, vol. 10, n° 2, p. 26-34, mars 2006.
- [69] S. N. Pakzad, G. L. Fenves, S. Kim, et D. E. Culler, « Design and Implementation of Scalable Wireless Sensor Network for Structural Monitoring », *Journal of Infrastructure Systems*, vol. 14, n° 1, p. 89-101, mars 2008.
- [70] I. Khoufi, P. Minet, A. Laouiti, et S. Mahfoudh, « Survey of Deployment Algorithms in Wireless Sensor Networks: Coverage and Connectivity Issues and Challenges », *International Journal of Autonomous and Adaptive Communications Systems*, vol. 10, n° 4, p. 341-390, 2017.
- [71] C. Zhu, C. Zheng, L. Shu, et G. Han, « A survey on coverage and connectivity issues in wireless sensor networks », *Journal of Network and Computer Applications*, vol. 35, n° 2, p. 619-632, mars 2012.
- [72] S. Mini et K. Pujari, « Phase transition analysis of target coverage problem in wireless sensor networks », *IEEE Sensors Journal*, vol. 13, n° 7, p. 2742-2749, avr. 2013.
- [73] Ravishankar Rao, S. Vrudhula, et D. N. Rakhmatov, « Battery modeling for energy-aware system design », *Computer*, vol. 36, n° 12, p. 77-87, déc. 2003.
- [74] « Datasheet Energizer E91 ». [En ligne]. Disponible sur: <http://data.energizer.com/pdfs/e91.pdf>. [Consulté le: 07-sept-2018].
- [75] J. Rahme, N. Fourty, K. Al Agha, et A. Van den Bossche, « A recursive battery model for nodes lifetime estimation in wireless sensor networks », in *2010 IEEE Wireless Communication and Networking Conference*, Sydney, NSW, Australia, 2010, p. 1-6.
- [76] M. Handy et D. Timmermann, « Simulation of Mobile Wireless Networks with Accurate Modelling of Non-linear Battery Effects », in *proceedings of Applied simulation and modeling (ASM)*.
- [77] A. Dunkels, F. -osterlind, N. Tsiftes, et Z. He, « Software-based On-line Energy Estimation for Sensor Nodes », in *Proceedings of the 4th workshop on Embedded networked sensors*, Cork, Ireland, 2007, p. 28-32.
- [78] E. Nataf et O. Festor, « Accurate Online Estimation of Battery Lifetime for Wireless Sensors Network », in *SENSORNETS - 2nd International conference on sensor networks*, Barcelone, Spain, 2013.
- [79] D. Tchuani Tchakonte, E. Simeu, et M. Tchuente, « Adaptive healing procedure for lifetime improvement in Wireless Sensor Networks », in *2015 IEEE 21st International On-Line Testing Symposium (IOLTS)*, Halkidiki, Greece, 2015, p. 59-64.
- [80] D. Tchuani Tchakonte, E. Simeu, et M. Tchuente, « Lifetime optimization of wireless sensor networks with sleep mode energy consumption of sensor nodes », *Wireless Networks*, 2018.
- [81] Directive européenne 2000/60/CE du 23 octobre 2000. 2000.
- [82] LOI n° 96-1236 du 30 décembre 1996 sur l'air et l'utilisation rationnelle de l'énergie. 1996.
- [83] « Site officiel de l'Organisation mondiale de la Santé ». [En ligne]. Disponible sur: <http://www.who.int/fr>. [Consulté le: 07-sept-2018].
- [84] Décret N°2011/2582/PM du 23 août 2011-08-25. 2011.
- [85] W. Ying Yi, K. Ming Lo, T. Mak, K. Sak Leung, L. Leung, et M. Ling Meng, « A Survey of Wireless Sensor Network Based Air Pollution Monitoring Systems », *Sensors*, vol. 15, n° 12, p. 31392-31427, 2015.
- [86] P. Bultynck, C. Reliquet, et M. [editors Felicio, « Initiative sur la qualite de l'air dans les villes d'Afrique sub-saharienne : rapport d'avancement 1998-2002 », The World Bank, 28450, janv. 2003.
- [87] M. Pavani et R. P. Trinatha, « Urban Air Pollution Monitoring Using Wireless Sensor Networks: A Comprehensive Review », *International Journal of Communication Networks and Information Security*, vol. 9, n° 3, p. 439-449, déc. 2017.
- [88] T. Ahuja, V. Jain, et S. Gupta, « Smart Pollution Monitoring for Instituting Aware Traveling », *International Journal of Computer Applications*, vol. 145, n° 9, p. 4-11, juill. 2016.
- [89] S. Abraham et X. Li, « A Cost-effective Wireless Sensor Network System for Indoor Air Quality Monitoring Applications », *Procedia Computer Science*, vol. 34, p. 165-171, 2014.
- [90] S. Mansour, N. Nasser, L. Karim, et A. Ali, « Wireless Sensor Network-based air quality monitoring system », in *2014 International Conference on Computing, Networking and Communications (ICNC)*, Honolulu, HI, USA, 2014, p. 545-550.

- [91] A. Kadri, E. Yaacoub, M. Mushtaha, et A. Abu-Dayya, « Wireless sensor network for real-time air pollution monitoring », in *2013 1st International Conference on Communications, Signal Processing, and their Applications (ICCSPA)*, Sharjah, 2013, p. 1-5.
- [92] J. Kang et J. Young Kim, « Portable RF-Sensor System for the Monitoring of Air Pollution and Water Contamination », *Journal of Analytical Methods in Chemistry*, vol. 2012, p. 5, 2012.
- [93] S. Devarakonda, P. Sevusu, H. Liu, R. Liu, L. Iftode, et B. Nath, « Real-time air quality monitoring through mobile sensing in metropolitan areas », in *Proceedings of the 2nd ACM SIGKDD International Workshop on Urban Computing - UrbComp '13*, Chicago, Illinois, 2013, p. 1.
- [94] « DS18B20 Datasheet ». [En ligne]. Disponible sur: <https://datasheets.maximintegrated.com/en/ds/DS18B20.pdf>. [Consulté le: 07-sept-2018].
- [95] « Raspberry Pi Datasheet ». [En ligne]. Disponible sur: https://www.raspberrypi.org/documentation/hardware/computemodule/datasheets/rpi_DATA_CM_1p0.pdf. [Consulté le: 07-sept-2018].
- [96] « PostgreSQL: The world's most advanced open source database ». [En ligne]. Disponible sur: <https://www.postgresql.org/>. [Consulté le: 07-sept-2018].
- [97] « Grafana - The open platform for analytics and monitoring ». [En ligne]. Disponible sur: <https://grafana.com/>. [Consulté le: 07-sept-2018].
- [98] « Figaro CMM5042 Datasheet ». [En ligne]. Disponible sur: [https://www.figaro.co.jp/en/product/docs/cmm5042_product%20infomation\(en\)_rev02.pdf](https://www.figaro.co.jp/en/product/docs/cmm5042_product%20infomation(en)_rev02.pdf). [Consulté le: 07-sept-2018].

Annexe

D. Tchuani Tchakonte, E. Simeu, et M. Tchuente, « Lifetime optimization of wireless sensor networks with sleep mode energy consumption of sensor nodes », *Wireless Networks*, 2018, URL: <https://doi.org/10.1007/s11276-018-1783-3>.



Lifetime optimization of wireless sensor networks with sleep mode energy consumption of sensor nodes

Diane Tchuani Tchakonté^{1,2,3,4} · Emmanuel Simeu¹ · Maurice Tchuente^{2,3}

© Springer Science+Business Media, LLC, part of Springer Nature 2018

Abstract

The energy constraint is a major issue in wireless sensor networks since battery cells that supply sensor nodes have a limited amount of energy and are neither replaceable nor rechargeable in most cases. A common assumption in previous work is that the energy consumed by sensors in sleep mode is negligible. With this hypothesis, the usual approach is to iteratively consider subsets of nodes that cover all the targets. These subsets, also called cover sets, are then put in the active mode whereas the others are in the low-power or sleep mode. The scheduling of the appropriate cover sets in order to maximize the network lifetime is a challenging problem known to be NP-hard. The consideration of non-zero energy consumption of sensor nodes in sleep mode is more realistic but significantly increases the complexity of the problem. In this paper, we address this question by proposing a greedy algorithm that gives priority to sensors with lowest energy, and uses a blacklist to limit the number of sensors covering critical targets. Simulations show that this algorithm outperforms the previously published solutions. We then propose for regular arrays, an analytical approach which shows that, for any optimal solution, all sensors' remaining energies are zero. This theoretical approach sheds a new light on ring connected arrays of odd size, that are known to be rather tricky when non-disjoint cover sets are considered.

Keywords Wireless sensor network · Energy consumption · Lifetime maximization · Cover set · Regular array

1 Introduction

Wireless sensor networks (WSNs) consist of tiny sensor nodes with embedded microcontrollers, low power radios, battery cells and sensors which are used to monitor environmental conditions such as temperature, pressure, humidity, and vibration [1].

Target coverage applications are the category of WSN applications where several discrete points of interest called targets need to be continuously monitored by sensor nodes [2]. Since sensor nodes have a limited amount of energy, it is critical to efficiently manage their energy consumption in order to extend the network lifetime that is defined as the time span during which all the targets are continuously covered. A common approach to tackle this issue is to alternate the sensor node functioning modes between active and sleep modes. The scheduling of appropriate subsets of sleep/active sensor nodes in order to maximize the network lifetime is called maximum lifetime coverage problem (MLCP).

WSNs belong to the class of low-power wide area network (LPWAN) technology that is specialized for interconnecting devices with low-bandwidth connectivity, focusing on range and power efficiency [3]. LPWAN technology is perfectly suited for connecting devices that need to send small amount of data over a long range, while preserving a long battery life. Many applications use LPWAN technology, including smart city, street lighting control, precision agriculture, leak detection, and environment monitoring. Several solutions

✉ Diane Tchuani Tchakonté
diane.tchuani@gmail.com

Emmanuel Simeu
emmanuel.simeu@univ-grenoble-alpes.fr

Maurice Tchuente
maurice.tchuente@gmail.com

¹ Univ. Grenoble Alpes, CNRS, Grenoble INP (Institute of Engineering Univ. Grenoble Alpes), TIMA, 38000 Grenoble, France

² IRD, Unité de Modélisation Mathématique et Informatique des Systèmes Complexes (UMMISCO), Bondy, France

³ Université de Yaoundé I, Yaoundé, Cameroon

⁴ PKFokam Institute of Excellence, Yaoundé, Cameroon

have been developed for LPWAN technology, including LoRaWAN, Ingenu, Weightless, and Sigfox. LoRaWAN is arguable the most adopted because it is open and considered more flexible.

LoRa is the physical layer used in LoRaWAN [4]. It features low power operation (10–120 mA in transmit mode between +7 and +20 dBm versus 12 mA in receive mode), low data rate (300–5000 bps) and long communication range (2–5 km in urban areas and 15 km in suburban areas). LoRaWAN networks are organized in a star-of-stars topology, in which gateway nodes relay messages between end-devices (e.g. sensor nodes) and a central network server. End-devices send data to gateways over a single wireless hop and gateways are connected to the network server through a non-LoRaWAN network (e.g. IP over cellular or Ethernet). Communication is bi-directional, although uplink communication from end-devices to the network server is strongly favored. LoRaWAN defines three types of devices (class A, B and C) with different capabilities. Class A is the class of LoRaWAN devices with the lowest power consumption [4]. Therefore, the sensor nodes of a WSN are of class A. In LoRaWAN, a class A device listens for a response after sending a frame. Thus, downlink transmission (from the base station to the sensor node) is only allowed after a successful uplink transmission. This implies that in sleep mode, a sensor node must periodically send a message to the base station in order to allow the base station to send it a message if there is any.

Let us consider a target coverage application using LoRaWAN, where a sensor in active mode periodically transmits samples. Considering that the energy consumption of sensor nodes is dominated by communication costs, let us observe the values of the ratio between the energy consumed in sleep mode and the energy consumed in active mode according to the values of the downlink communication latency of a sensor in sleep mode. Table 1 shows that the ratio between the energy consumed in sleep mode and the energy consumed in active mode is not negligible.

With the assumption that the energy consumed in sleep mode is negligible, the battery lifetime of the sensor s_i is

$b_i = E_i/P_a$, where E_i is the initial energy of the sensor s_i and P_a is the power consumed by a sensor in active mode. But when we consider that a sensor consumes a power P_v in sleep mode, the battery lifetime b_i now depends not only on the initial energy but also on the time spent in sleep mode. Let r be the ratio between the energy in sleep mode and the energy in active mode, then $r = P_v/P_a$ and $r \in]0; 1]$. If the energy consumed in sleep mode is negligible, then the ratio r tends to zero otherwise the ratio r needs to be taken into consideration. In this paper, we address the MLCP when the energy consumed in sleep mode is not negligible.

The rest of this paper is organized as follows. We introduce previous work on the MLCP in Sect. 2. In Sect. 3, a procedure to identify critical targets in the context where the energy consumed in sleep mode is not null is presented. Section 4 presents the greedy algorithm proposed for the MLCP. Simulation results are presented and discussed in Sect. 5. In Sect. 6, we propose a solution for ring connected networks, that solution generalizes a well-known tricky example in the literature. We conclude in Sect. 7.

2 Previous work

The MLCP has been studied extensively in the literature. This problem is defined as follows. Let $S = \{s_1, \dots, s_n\}$ be a set of sensor nodes and $T = \{t_1, \dots, t_m\}$ be a set of targets. Every sensor s_i has a battery lifetime b_i . Sensor s_i covers target t_j if the distance between t_j and s_i does not exceed the sensing range of s_i . Hereafter we denote $T_{[s]}$ the set of targets covered by a sensor s , and $S_{[t]}$ the set of sensors covering a target t .

A cover set C is a set of sensor nodes that jointly cover all the targets. The activation time Lt of a cover set C is the time span during which the sensor nodes of the set C are in active mode while all the others are in sleep mode. The activation time Lt cannot exceed the smallest available battery lifetime of the sensor nodes belonging to C (i.e. $Lt \leq \min_{s_i \in C} b_i$). The MLCP is the problem of finding a succession of cover sets with their respective activation times, which has the maximum network lifetime under the constraint that no sensor s_i has an aggregated activation time longer than b_i [5]. The MLCP is known to be NP-complete [6].

Many approaches have been proposed in the literature in order to find approximate solutions for the MLCP with the common assumption that the energy consumed in sleep mode is negligible.

Combinatorial optimization methods have been used to develop approximation algorithms based on the maximum disjoint cover sets (MDSC) problem. For instance, Mini and Pujari [5] designed a simplex-like algorithm while Cardei and Du [7] proposed to transform the MDSC problem into a maximum flow problem. On the other hand,

Table 1 Approximate values of the ratio between the energy consumed in sleep mode and the energy consumed in active mode

Sample transmission period of a sensor in active mode (min)	Downlink communication latency of a sensor in sleep mode (min)	Ratio between energy consumed in sleep mode and energy consumed in active mode
1	5	$r = 0.2$
1	10	$r = 0.1$
5	10	$r = 0.5$
5	20	$r = 0.25$

two genetic algorithms have been proposed by Lai et al. [8] and Li et al. [9]. Combinatorial optimization methods produce a collection of unordered cover sets. This is not satisfactory for networks where sensors have non-zero energy consumption in sleep mode. Indeed, when this hypothesis is taken into account, the lifetime of a network depends heavily on the order in which the sensors are activated, as we will see in the next section. One may attempt to solve this problem by looking for new formulations that take into account the order of cover sets generated by these combinatorial optimization methods, but this seems very difficult.

A second idea is to approximate the MLCP by iteratively generating cover sets of minimum sizes. One is then faced with the minimum cover set (MCS) problem that is known to be NP-hard problem [10]. Moreover, this approach does not work well as illustrated by the example of Fig. 1 where all sensors have the same battery life. Indeed, it may produce the single cover set $C = \{s_1, s_2\}$ whereas the optimal solution consists of the two covers $C_1 = \{s_1, s_4\}$ and $C_2 = \{s_2, s_3\}$.

The third class of algorithms proceeds iteratively to construct disjoint or non-disjoint cover sets. Almost all of them use the notion of critical target combined with a greedy heuristic for the MLCP [6, 11–13]. A target is said to be critical if its maximal coverage time when taken individually, is minimal. Indeed, the targets with the smallest maximal coverage time place an upper bound on the network lifetime. Thus, it is important to identify critical targets and manage very carefully the sensors that cover them. Cardei et al. [6] proposed the Greedy-MSC algorithm that iteratively chooses a critical target and selects among the sensors covering it, the one that covers the highest number of targets not yet covered by the cover set under construction. Tchuani et al. [13] proposed a greedy algorithm called adaptive algorithm which iteratively selects the sensor covering the highest number of critical targets not yet covered by the cover set under construction. For the iterative construction of a cover set C ,

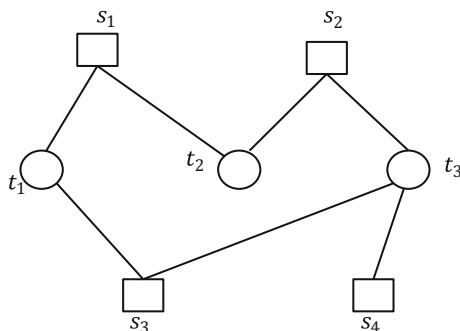


Fig. 1 An instance of the MLCP

Zorbas et al. [12] compute at each step of the static-CCF algorithm, and for each sensor s^* , a score that depends on the size of $T_{[s^*]}$, the number of elements of $T_{[s^*]}$ covered by C , and a parameter that expresses the relation between s^* and the critical targets. Contrary to the three previous algorithms, the greedy algorithm proposed by Slijepcevic and Potkonjak [11] defines the notion of critical targets not in a static way regarding the whole set of targets, but in a dynamic way regarding the set of targets not yet covered. At each step, it then selects a critical target t^* , computes a score for each sensor of $S_{[t^*]}$ and chooses among them, the one with the highest score. The algorithm called dynamic-CCF introduced by Glynos et al. [12] takes the basic idea of static-CCF algorithm and combines it with the concept of dynamic critical target.

In this paper, we introduce a new iterative method based on the concept of blacklist which is a binary version of the relation between a sensor and the critical targets as introduced in [12], and we combine it with the sensors' energy levels.

3 Identifying critical targets

The coverage time of critical targets taken individually, places an upper bound on the network lifetime. With the assumption that the energy consumed in sleep mode is negligible and all the sensors have the same energy, critical targets are the ones covered by the smallest number of sensors. But, in the context where sensors have non-zero energy consumption in sleep mode, the computation of the maximal coverage time of a target is not trivial.

Proposition 1 *Let t be a target covered by p sensors $s_0, s_1, \dots, s_{p-1}$ with respective energies $E_0, E_1, \dots, E_{p-1}$. The coverage time D_t of t is maximal when the p sensors are activated according to the ascending order of their energies.*

Proof Let us first consider the case of two sensors s_0 and s_1 of energies E_0 and E_1 such that $E_0 \leq E_1$. Let Δ_a be the coverage time when the sensors are activated in ascending order of their energies and let Δ_d be the coverage time if they are activated in decreasing order of their energies. Then, $\Delta_a = \frac{E_0(1-r)+E_1}{P_a}$ while $\Delta_d = \frac{E_1(1-r)+E_0}{P_a}$ if $E_0 > r \cdot E_1$, and $\Delta_d = \frac{E_1}{P_a}$ if $E_0 \leq r \cdot E_1$. It easily checked that $\Delta_a > \Delta_d$ and the result holds.

For $p > 2$, let us consider the sequence $s_0, s_1, \dots, s_{i-2}, s_{i+1}, \dots, s_{p-1}$ and two sensors s and s' with respective energies E and E' . Let us assume that $E < E'$. We have just seen that the activation time of the sequence (s, s') is greater than the activation time of the sequence (s', s) . Let us insert the two sensors in the sequence,

respectively in ascending and descending order of their energies. Let Seq_1 be the sequence $s_0, s_1, \dots, s_{i-2}, s, s', s_{i+1}, \dots, s_{p-1}$ and Seq_2 be the sequence $s_0, s_1, \dots, s_{i-2}, s', s, s_{i+1}, \dots, s_{p-1}$. Let D_1 and D_2 be the time when sensor s_{i+1} is activated respectively in Seq_1 and Seq_2. Since $D_1 > D_2$, then at time D_2 , Seq_1 has one more sensor s^* of energy $(D_1 - D_2) * P_a$ than Seq_2. Therefore, Seq_1 is better than Seq_2. This shows that in an optimal activation sequence, two consecutive sensors are in the order of increasing energies, and consequently, the sensors are globally in ascending order of their energies. $\square$

We are now ready to compute the maximal coverage time D_t of a target covered by p sensors $s_0, s_1, \dots, s_{p-1}$ with energies $E_0, E_1, \dots, E_{p-1}$. As seen above, we can assume that $E_0 \leq E_1 \leq \dots \leq E_{p-1}$. Let us first note that there is only one active sensor at a time. Let t'_i be the time for which sensor s_i is in active mode. It is easy to check that: $t'_0 = t_0$ and $t'_i = t_i - r(t'_0 + t'_1 + \dots + t'_{i-1})$ for $0 < i < p$, where $t_i = E_i/P_a$. The formula for D_t is the sum $t'_0 + t'_1 + \dots + t'_{p-1}$. A target is critical if this sum is minimal.

4 Proposed algorithm

The algorithm that we propose for the MLCP is based on two main strategies.

The first idea is to limit in every cover set, the number of sensors that cover the critical targets. This idea has been first introduced by Slijepcevic and Potkonjak [11] and then by Zorbas et al. [12]. The common approach used in these two works is to assign scores to sensors. But the formulas to compute these scores include some parameters whose values need to be appropriately set depending on the instance of the problem. A different approach is to define a blacklist. The blacklist contains all the sensors not yet selected and that cover a critical target already covered by the cover set under construction. A sensor is selected from the blacklist only when none of the sensors out of the blacklist covers the remaining targets.

The second idea of this algorithm is to give priority to sensors with minimal energy. According to Proposition 1, the coverage time of a target is maximal when the sensors covering that target are activated in the ascending order of their initial energies.

The new algorithm is described next. The algorithm first identifies the set of critical targets. Then, the algorithm iteratively constructs the cover sets till a target is no more covered by an active sensor. Before the construction of a new cover set, the algorithm computes the remaining energy of every sensor to make sure that every target is covered by at least one active sensor. The computation of the remaining energy of a sensor takes into consideration the energy

consumed both in active and sleep modes. The construction of a new cover set starts by creating an empty blacklist. Then the algorithm iteratively selects a sensor with minimal energy that is not in the blacklist and that covers at least one new target. If there are many sensors with minimal energy, the algorithm selects the one covering the highest number of new targets. If the remaining targets are covered only by sensors in the blacklist, then the sensor with minimal energy is selected from the blacklist. If there are many candidates, the sensor covering the lowest number of already covered critical targets is selected. When a new sensor is selected, the blacklist is updated. The construction of the new cover set ends when it covers all the targets. The next step is to search for redundant sensors and remove them from the cover set. The activation duration of the new cover set is set to the minimal lifetime of its sensors. The time complexity of the algorithm is polynomial.

Algorithm

Begin

Identify the set of critical targets.

Set $k = 1$ (first cover set)

For every sensor s_i , set $E_{i,1} = E_i$, (where E_i is the initial energy of the sensor s_i)

While every target is covered by a sensor s_i such that

$E_{i,k} > 0$ **Do**

Initialize the blacklist to the empty set

While all targets are not covered by cover set C_k **Do**

Select the sensor s_{min} not in the *blacklist* whose energy $E_{i,k}$ is minimal among those that cover at least one target not yet covered by the cover set C_k .

If no sensor is found **then**

Select the sensor s_{min} from the blacklist according to the same criterion.

Endif

Insert in the blacklist all the sensors that cover the same critical targets as s_{min} .

Insert the sensor s_{min} in the cover set C_k .

EndWhile

Delete redundant sensors from C_k if there are any.

Set the activation duration C_k as:

$$Lt_k = \frac{1}{P_a} * \min_{s_i \in C_k} (E_{i,k}).$$

Update the remaining energy of every sensor s_i so that

$E_{i,k+1} = \max(0, E_{i,k} - P_a \cdot Lt_k)$ if the sensor belongs to C_k and $E_{i,k+1} = \max(0, E_{i,k} - r \cdot P_a \cdot Lt_k)$ otherwise.

$k = k + 1$

EndWhile

Return the collection of cover sets $C_1, C_2, \dots$ with the activation durations $Lt_1, Lt_2, \dots$

End

5 Simulation results

In this section, we illustrate the performance of the new algorithm and we compare it to some greedy algorithms for the MLCP. These algorithms include Greedy-MSC [6], static-CCF [12], dynamic CCF [12], adaptive algorithm [13], and MCS-based algorithm generating minimum size cover sets by using the classical greedy algorithm [10].

The metric used to measure the performance of each algorithm is the ratio between the network lifetime given by the algorithm and the upper bound of the network lifetime that is the maximal coverage time of the critical targets (see Sect. 3).

We considered networks with the following characteristics:

- Sensors:
 - 200 sensors.
 - Sensing range: 10.
 - Uniform distribution over the area of coverage.
 - Energies: taken randomly according to the Gaussian law of mean 5085 mW h (that is the energy of common AA batteries) and standard deviations 100 and 500.
- Targets:
 - 50 targets.
 - Uniform distribution over the area of coverage.
- Area of coverage: 50×50 squares.
- Ratio r between power in sleep mode and power in active mode: from 0 to 0.9.

For each experiment, 100 different instances are generated. We execute the algorithm on each of these instances and we compute the ratio between the solution obtained and the upper bound of the network lifetime. We consider the series of the 100 ratios and we determine the first quartile, the minimum, the median, the third quartile and the maximum of the series. These values are then used to draw box plots represented on the figures below.

Figure 2 shows the performances of the algorithms for $r = 0$ and standard deviation of the energy distribution is equal to 100. We observe that most of the algorithms find a ratio very close to 1 in more than 75% of cases while Greedy-MSC gives a ratio very close to 1 in more than 50% of cases. Since the standard deviation is small compared to the mean of the Gaussian law, the sensors have almost the same initial energy. So, the algorithms perform well when the energy in sleep mode is negligible and the sensors have similar energies.

Figure 3 shows the performances of the algorithms for $r = 0.1$ and standard deviation is equal to 100. All the

algorithms give a ratio to the optimum greater than 0.95 in about more than 75% of cases and a ratio very close to 1 in more than 50% of cases. Hence, all the algorithms perform well when the energy consumed in sleep mode is not negligible and the sensor's energies are similar.

Figure 4 compares the performances of algorithms for $r = 0$ and standard deviation of the energy distribution is equal to 500. In this case, the sensors' energies are more scattered than previously. We observe that adaptive algorithm, dynamic-CCF algorithm, MCS-based algorithm and the new algorithm give a ratio to the optimum very close to 1 in more than 75% of cases, while Static-CCF and Greedy-MSC algorithms give a ratio greater than 0.9 in more than 75% of cases. We can therefore note that all these algorithms perform well when the energy consumed in sleep mode is negligible, no matter the dispersion of the sensors' energies.

Figure 5 compares the performances of the algorithms for $r = 0.1$ and standard deviation of the energy distribution is 500. We can observe that the performances of the algorithms, except the new algorithm, decrease. Indeed, dynamic-CCF, static-CCF and Greedy-MSC give a ratio less than 0.9 in about 50% of cases. Adaptive algorithm and MSC-based algorithm give a ratio less than 0.95 in about 50% of cases. The new algorithm gives a ratio greater than 0.95 in more than 75% of cases and a ratio equal to 1 in about 25% of cases. This shows that when the energy consumed in sleep mode is not negligible and the sensors' energies are quite different, the new algorithm performs better than the others.

Figures 6 and 7 present the performances of the new algorithms for $r = 0.2, 0.3, \dots, 0.9$ and standard deviation of the energy distribution equals to 100 and 500 respectively. We observe that when the sensors' energies are similar (standard deviation equal to 100), the new algorithm gives a ratio very close to 1 in more than 75% of cases for $r \leq 0.4$ and a ratio very close or equal to 1 in all cases for $r > 0.4$. When the sensor's energies are more scattered (standard deviation equal to 500), the new algorithm finds a ratio greater than 0.95 in more than 75% of cases for $r \leq 0.4$ and a ratio very close or equal to 1 in all

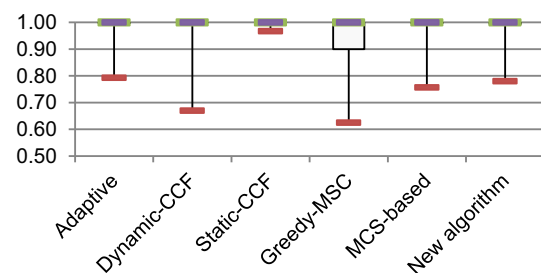


Fig. 2 Performance of algorithms when $r = 0$ and standard deviation is 100

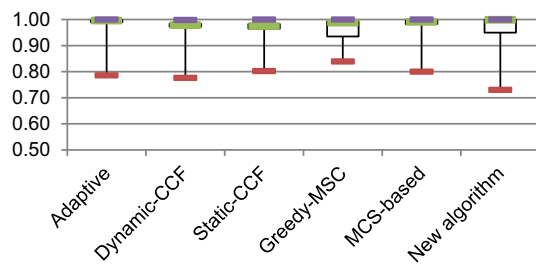


Fig. 3 Performance of algorithms when $r = 0.1$ and standard deviation is 100

cases for $r > 0.4$. This shows that the new algorithm performs very well when $r \neq 0$ no matter the dispersion of the sensors' energies.

6 A solution for general ring connected networks

In the literature, there is a well-known example that is a hard instance of the MLCP when the energy consumed in sleep mode is negligible [5, 6, 12]. This instance, shown in Fig. 8, consists of three targets (t_1 , t_2 , and t_3) and three sensors (s_1 , s_2 , and s_3). Sensor s_1 covers targets t_1 and t_2 , sensor s_2 covers t_2 and t_3 , while sensor s_3 covers t_1 and t_3 . Let $E = 1$ be the energy of each sensor and $P_a = 1$ be the power in active mode.

With the assumption that the energy consumed in sleep mode is negligible ($r = 0$), the solution made of three non-disjoint cover sets $C_1 = \{s_1, s_2\}$, $C_2 = \{s_2, s_3\}$, and $C_3 = \{s_1, s_3\}$ with activation time $\frac{1}{2}$ each, gives the optimal network lifetime $\frac{3}{2}$. Indeed, the optimal solution consumes two units of energy (for two active sensors) during at least 1 unit of time, i.e. before the first sensor runs out of energy. Then, it remains globally at most 1 unit of energy to activate two sensors. This second phase lasts for at most $\frac{1}{2}$ unit of time. Therefore, the lifetime $\frac{3}{2}$ obtained by the above solution is optimal.

This instance of the MLCP belongs to the family $\mathcal{F}$ of ring connected networks illustrated in Fig. 9 and defined as follows.

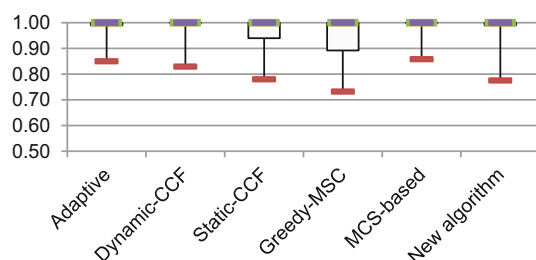


Fig. 4 Performance of algorithms when $r = 0$ and standard deviation is 500

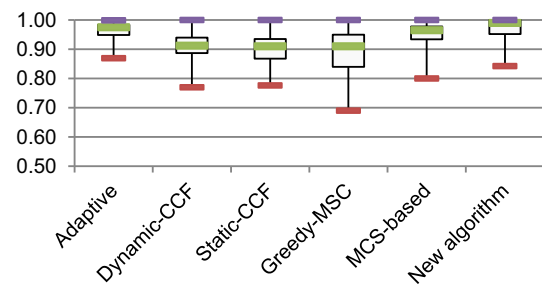


Fig. 5 Performance of algorithms when $r = 0.1$ and standard deviation is 500

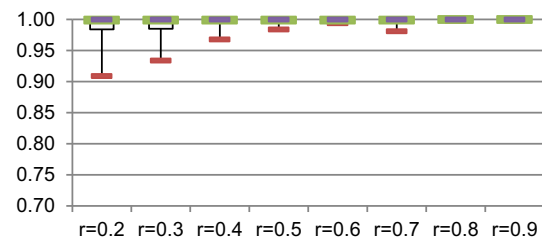


Fig. 6 Performance of the new algorithm when standard deviation is 100

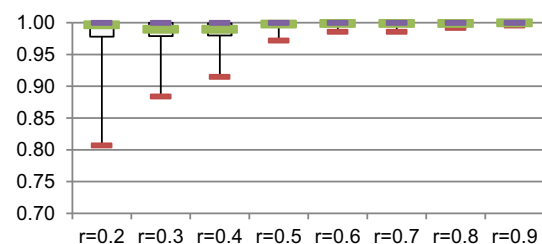


Fig. 7 Performance of the new algorithm when standard deviation is 500

Definition 1 An instance $\mathcal{F}n$ of $\mathcal{F}$ is a bipartite ring-connected network where n sensor nodes $s_1, \dots, s_n$ and n targets $t_1, \dots, t_n$ alternate as illustrated in Fig. 9. In such a network, each sensor covers exactly two targets and each target is covered by exactly two sensors.

The previous example of Fig. 8 is the instance of the family $\mathcal{F}$ where $n = 3$. As said earlier in this section, some existing greedy algorithms [6, 12] produce an optimal solution consisting of three non disjoint cover sets, and where the parameter w specifying the maximum number of cover sets a sensor can belong to, is set to 2. However, the authors don't explain how the appropriate value of this parameter w was found. In the following, we propose a systematic approach to find a solution for every instance of $\mathcal{F}$ when energy consumption in sleep mode is negligible.

Let us first note that the problem is trivial for $n = 2p$, because the optimal solution consists of two disjoint covers involving sensors with even indices and sensors with odd indices. For every instance of the family $\mathcal{F}$ with

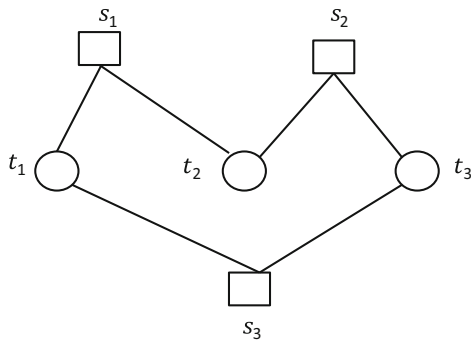


Fig. 8 A hard instance of the MLCP

$n = 2p + 1$ sensors, there are n non-disjoint cover sets of minimum size and every sensor belongs to $p + 1$ of them. If all the n cover sets are activated, then every sensor belongs to $p + 1$ covers. We then set $w = p + 1$ and each cover set will use sensors with energy $\frac{E}{p+1}$. Therefore, the activation duration of each cover set is $\frac{1}{p+1} * \frac{E}{P_a}$ and the network lifetime of the solution is $\frac{n}{p+1} * \frac{E}{P_a}$.

Let us now consider the more general case of regular networks.

Let $G = (S, T, U)$ be a sensor network with S as set of sensors, T as set of targets, and $(s, t) \in U$ if sensor s covers target t .

Definition 2 An automorphism of G is a permutation σ of $S \cup T$, such that $(s, t) \in U$ if and only if $(\sigma(s), \sigma(t)) \in U$. In other words, an automorphism is a structure-preserving permutation of vertices.

Definition 3 G is said to be regular if every sensor may be mapped by an automorphism to any other sensor. In other words, for any couple (s, s') of sensors, there is an automorphism σ such that $\sigma(s) = s'$.

$\mathcal{F}$ is an example of a family of regular networks.

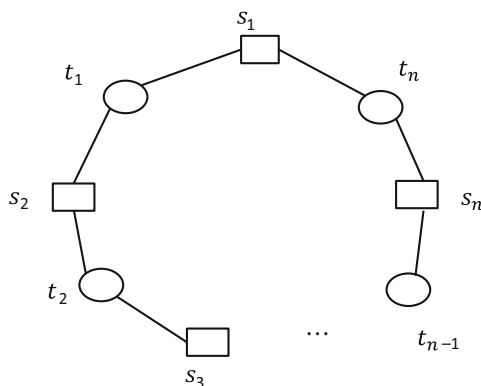


Fig. 9 An instance $\mathcal{F}_n$ of family $\mathcal{F}$

Definition 4 The residue at sensor s_i corresponding to an algorithm for the MLCP, is the amount of energy left in s_i after the execution of the algorithm.

We can now state some fundamental properties of MLCP solutions for regular networks.

Proposition 2 Every optimal solution for the MLCP on a regular network has zero residue.

Proof We will use logic and absurdity to prove this result. So, let us assume that an optimal Algorithm A with maximal lifetime Δ_{opt} on a regular array G , has a non-zero residue e at node s^* . Let us consider the automorphisms $\sigma^{(j)}, j = 1, \dots, n$ such that $\sigma^{(j)}(s^*) = s_j$, for $j = 1, \dots, n$. Let $A^{(1)}, A^{(2)}, \dots, A^{(n)}$ be the algorithms obtained from A by applying the permutations $\sigma^{(j)}, j = 1, \dots, n$, after dividing the energy level of each sensor by $w = n$. Then, each $A^{(j)}$ has residue $\frac{e}{n}$ at node s_j because $s_j = \sigma^{(j)}(s^*)$ and A has a residue e at node s^* . We can therefore form a new cover set and the network lifetime of the new solution is $\Delta_{opt} + \frac{e}{P_a * n}$. This leads to a contradiction since Δ_{opt} is the maximal network lifetime. This shows that every optimal solution of a regular network has zero residue. $\square$

Remark Let us consider an algorithm A defined on a regular network, associated with a cover set C , and for which there is exactly one sensor s^* with non-zero residue. As above, we can generate, using appropriate automorphisms, $|C|$ cover sets of lifetime $\frac{1}{|C|} * \frac{E}{P_a}$ such that each new cover set C_i produces a residue on a specific sensor of C . These cover sets leave on all the sensors of C , residues of value $\frac{E}{|C|}$ which together form a new cover set. Therefore, we have a zero residue solution of network lifetime $\frac{|C|+1}{|C|} * \frac{E}{P_a}$.

In the case where the energy consumed in sleep mode is not negligible, the following proposition gives the optimal solution for the instance of family $\mathcal{F}$ where $n = 3$.

Proposition 3 The optimal solution of the instance of family $\mathcal{F}$ when $n = 3$, is of network lifetime $\frac{3-r}{2} * \frac{E}{P_a}$ and consists of the cover sets $C_1 = \{s_1, s_2\}$, $C_2 = \{s_2, s_3\}$, and $C_3 = \{s_1, s_3\}$, with respective activation durations $Lt_1 = \frac{1}{2} * \frac{E}{P_a}$, $Lt_2 = \frac{1}{2} * \frac{E}{P_a}$, and $Lt_3 = \frac{1-r}{2} * \frac{E}{P_a}$.

Proof Let us denote x the time span during which two sensors are active and one is in sleep mode, and y the time span during which two sensors are active and one is out of energy. The objective is to maximize the network lifetime $x + y$ under the following constraints:

- $x \geq \frac{E}{P_a}$ since we need at least $\frac{E}{P_a}$ units of time to deplete a battery.

- $2xP_a + xrP_a + 2yP_a \leq 3E$ since the total energy available in the three sensors is $3E$.

The solution of this linear programming problem can be obtained graphically and is $x = \frac{E}{P_a}, y = \frac{E}{2P_a}(1 - r)$. This corresponds to the network lifetime $\frac{3-r}{2} * \frac{E}{P_a}$. $\square$

In the following, we propose a solution for every instance $\mathcal{F}n$, $n = 2p + 1$, of family $\mathcal{F}$ when the energy consumed in sleep mode is negligible or not (i.e. $0 \leq r \leq 1$).

Let us consider the sequence $C_1, C_2 = \sigma(C_1), \dots, C_n = \sigma(C_{n-1})$, where C_1 is the minimal cover set $\{s_1, s_3, s_5, \dots, s_{n-1}\}$ and σ is the permutation defined by $\sigma(i) = i + 1$, for $i = 1, \dots, n - 1$, and $\sigma(n) = 1$. Note that σ is an automorphism of the ring of order n . The activation duration L_{t_j} of the cover set C_j (for $j = 1, \dots, n$) is $L_{t_j} = \gamma_j * \frac{E}{P_a}$, where $\gamma_j \in \mathbb{R}^*$.

We consider the subclass of solutions where the battery of a sensor is exhausted after its last activation in a cover set. For each sensor s_i , let us denote K^i the index of the last cover set to which it belongs. With these hypotheses, the total energy consumed by sensor s_i is $P_a * \sum_{j=1}^{K^i} L_{t_j} * (\delta_{ij} + r * \bar{\delta}_{ij})$, where $\delta_{ij} = 1$ if $s_i \in C_j$ and $\delta_{ij} = 0$ otherwise. Indeed, the energy consumed by sensor s_i during the activation of C_j is equal to $P_a * L_{t_j}$ if sensor s_i belongs to C_j and is equal to $r * P_a * L_{t_j}$ otherwise.

Assuming that every sensor s_i has zero residue after belonging to the last cover set, it is easy to check that we obtain the following system of equations:

$$\begin{cases} \sum_{j=1}^{K^1} \gamma_j * (\delta_{1j} + r * \bar{\delta}_{1j}) = 1 \\ \sum_{j=1}^{K^2} \gamma_j * (\delta_{2j} + r * \bar{\delta}_{2j}) = 1 \\ \vdots \\ \sum_{j=1}^{K^n} \gamma_j * (\delta_{nj} + r * \bar{\delta}_{nj}) = 1 \end{cases} \quad (1)$$

The solution $(\gamma_1, \gamma_2, \dots, \gamma_n)$ of the system of Eq. (1) is: $\gamma_i = \frac{1}{(p+1)+r*(p-1)}$ for $i = 1, \dots, n - 1$, and $\gamma_n = \frac{1-r}{(p+1)+r*(p-1)}$.

Let $\alpha = \frac{1}{(p+1)+r*(p-1)}$. Each sensor belongs to $p + 1$ cover sets. The last cover set containing a sensor is either C_{n-1} or C_n . If the last cover set containing a sensor is C_{n-1} , then, after the activation of C_{n-1} , its battery has consumed $\alpha((p + 1) + r * (p - 1)) * E = E$. As a consequence, its battery is exhausted after its last activation. If the last cover set containing a sensor is C_n , then after the activation of C_n , its battery has consumed $\alpha(p + (1 - r) + r * p) * E = E$, hence its battery is exhausted after its last activation.

The network lifetime of this solution is $\Delta = \frac{n-r}{(p+1)+r*(p-1)} * \frac{E}{P_a}$. Note that this generalizes the result shown in Proposition 3 for $n = 3$. This solution is optimal

for $n = 3$, and we conjecture that it is optimal for every instance $\mathcal{F}n$, $n = 2p + 1$, of family $\mathcal{F}$.

Let us consider another solution also based on graph automorphisms. In this solution, we assume that the battery of a sensor is exhausted after the activation of the last cover set. Therefore, we obtain the following system of equations:

$$\begin{cases} \sum_{j=1}^n \gamma_j * (\delta_{1j} + r * \bar{\delta}_{1j}) = 1 \\ \sum_{j=1}^n \gamma_j * (\delta_{2j} + r * \bar{\delta}_{2j}) = 1 \\ \vdots \\ \sum_{j=1}^n \gamma_j * (\delta_{nj} + r * \bar{\delta}_{nj}) = 1 \end{cases} \quad (2)$$

The solution $(\gamma_1, \gamma_2, \dots, \gamma_n)$ of the system of Eq. (2) is: $\gamma_i = \frac{1}{(p+1)+r*p}$ for $i = 1, \dots, n$. Indeed, each sensor belongs to $p + 1$ cover sets and does not belong to p cover sets. The network lifetime of this solution is $\Delta' = \frac{n}{(p+1)+r*p} * \frac{E}{P_a}$.

We observe that $\Delta = \Delta'$ for $r = 0$, while $\Delta > \Delta'$ for $0 < r < 1$. This shows that the two solutions return the same network lifetime when the energy in sleep mode is negligible whereas the former solution is better than the latter solution when the energy in sleep mode is not negligible. This result is logical because the total time that the sensors spent in sleep mode is higher in the second solution than in the former solution. Thus, when the energy in sleep mode is not negligible, this extra time spent in sleep mode reduces the network lifetime returned by the second solution.

Remark Let $\mathcal{F}n$ be an instance of family $\mathcal{F}$ with the optimal network lifetime Δ_{opt} . Let us consider the instance I of the MLCP obtained from $\mathcal{F}n$ by adding a new sensor that covers one target. We can generate two disjoint cover sets and get the network lifetime $\Delta'' = (2 - r) * \frac{E}{P_a}$ for the instance I . When we compare the network lifetime $\Delta = \frac{n-r}{(p+1)+r*(p-1)} * \frac{E}{P_a}$ of the instance $\mathcal{F}n$ to the network lifetime Δ'' of the instance I , we note that $\frac{\Delta''}{\Delta} \leq 1 + \frac{1-r}{3-r}$. Since $\Delta \leq \Delta_{opt}$, then the ratio between the network lifetime Δ'' of the instance I and the optimal network lifetime Δ_{opt} of the instance $\mathcal{F}n$ is such that $\frac{\Delta''}{\Delta_{opt}} \leq 1 + \frac{1-r}{3-r}$.

7 Conclusion

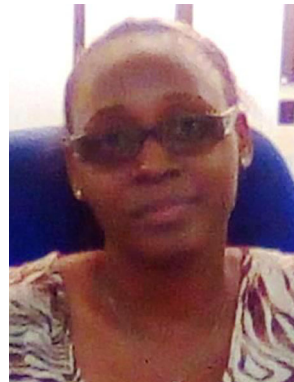
In this paper, we addressed the maximum lifetime coverage problem (MLCP) that aims at determining a sleep/active schedule for sensors in order to maximize the time span when all the targets are continuously covered by the network. Contrary to previous work that assumed that energy consumed in sleep mode is negligible, we take that energy into consideration. We presented two contributions. The

first contribution is a greedy algorithm that iteratively selects the sensor with minimal energy while trying to limit the number of sensors covering the critical targets through the use of a blacklist. The second contribution is a formal framework, based on graph automorphisms and linear algebra, for ring connected networks that generalize a well-known tricky example of the literature. Simulations show that the greedy algorithm performs very well whether the energy consumed in sleep mode is negligible or not and no matter the dispersion of the sensors' energies, contrary to existing algorithms.

Funding Funding was provided by Univ. Grenoble Alpes, CNRS, Grenoble INP (Institute of Engineering Univ. Grenoble Alpes), TIMA, 38000 Grenoble, France, Agence Universitaire de la Francophonie (AUF), African Center of Excellence in Information and Communication Technologies (CETIC), PKFokam Institute of Excellence, Yaoundé, Cameroon and IRD UMI 209 UMMISCO, 32, Avenue Henri Varagnat 93143 Bondy cedex FRANCE.

References

1. Akyildiz, I., Su, W., Sankarasubramaniam, Y., & Cayirci, E. (2002). Wireless sensor networks: A survey. *Computer Networks*, 38(4), 393–422.
2. Thai, M. T., Wang, F., Du, H., & Jia, X. (2008). Coverage problems in wireless sensor networks: Designs and analysis. *International Journal of Sensor Networks*, 3(3), 191–200.
3. Adelantado, F., Vilajosana, X., Tuset-Peiro, P., Martinez, B., Melià-Seguí, J., & Watteyne, T. (2017). Understanding the limits of LoRaWAN. *IEEE Communications Magazine*, 55(9), 34–40.
4. Sornin, N., Luis, M., Eirich, T., Kramp, T., & Hersent, O. (2015). LoRa specification 1.0, Lora Alliance Standard specification. www.lora-alliance.org. Accessed December 19, 2016.
5. Mini, S., & Pujari, K. (2013). Phase transition analysis of target coverage problem in wireless sensor networks. *IEEE Sensors Journal*, 13(7), 2742–2749.
6. Cardei, M., Thai, M. T., Li, Y., & Wu, W. (2005). Energy-efficient target coverage in wireless sensor networks. *Proceedings of IEEE INFOCOM*, 05(3), 1976–1984.
7. Cardei, M., & Du, D. Z. (2005). Improving wireless sensor network lifetime through power aware organization. *ACM Wireless Networks*, 11(3), 333–340.
8. Lai, C. C., Ting, C. K., & Ko, R. S. (2007). An effective genetic algorithm to improve wireless sensor network lifetime for large-scale surveillance applications. In *Proceedings of IEEE congress on evolutionary computation* (pp. 3531–3538).
9. Li, Y., Hu, X., & Zhang, J. (2009). A new genetic algorithm for the set k-cover problem in wireless sensor networks. In *Proceedings of IEEE international conference on systems, man, and cybernetics* (pp. 1405–1410).
10. Vazirani Vijay, V. (2003). *Approximation algorithms*. Berlin: Springer.
11. Slijepcevic, S., & Potkonjak, M. (2001). Power efficient organization of wireless sensor networks. In *Proceedings of IEEE international conference on communications* (pp. 472–476).
12. Zorbas, D., Glynos, D., Kotzanikolaou, P., & Douligeris, C. (2010). Solving coverage problems in wireless sensor networks using cover sets. *Ad Hoc Networks*, 8(4), 400–415.
13. Tchuani, D. T., Simeu, E., & Tchuenté, M. (2015). Adaptive healing procedure for lifetime improvement in wireless sensor networks. In *Proceedings of IEEE international on-line testing symposium* (pp. 59–64).



Diane Tchuani Tchakonté received her Bachelor degree in Computer Science in 2007 and her Master degree in 2011 from the University of Yaounde I, Cameroon. She is currently a Ph.D. student both in the Department of Computer Science at the University of Yaounde I, Cameroon and in the TIMA Laboratory at the Université Grenoble Alpes, France. She received a scholarship for 3 years of study from Agence Universitaire de la

Francophonie (AUF). She is a student member of the African Center of Excellence information and Communication Technologies (CETIC) and the IRD UMI 209 UMMISCO (Unité de Modélisation Mathématique et Informatique des Systèmes Complexes), that also support her studies. Her research theme is energy management in wireless sensor networks. She is currently an assistant lecturer in the Department of Computing and software Engineering at the University PKFokam Institute of Excellence, Cameroon.



Emmanuel Simeu received the electrical engineer diploma from Hassania School of engineering of Casablanca in 1987, Master degree in Automatic control and signal processing and the Ph.D. in Automatic Control and System Theory from the INPG (Institut National Polytechnique de Grenoble) in 1989 and 1992 respectively. He received the diploma of accreditation to supervise research (HDR) in physic sciences from Joseph

Fourier University of Grenoble in September 2005. From September 1989 to July 1992 he was a researcher in France Telecom Research Centre (CNET). He was associate professor in automatic control at ISAR (Automatic and Robotics Institute of Valence) from September 1992 to August 1995. He joined the Grenoble Joseph Fourier University (UJF) in September 1995 as associate professor in the fields of reliability and automatic control. He is currently a researcher at TIMA laboratory of Grenoble where he is the leader of RMS (Reliable Mixed Signal Systems) research group. He is the head of the Risk Management Department of Ecole Polytechnique of Univ Grenoble Alpes (UGA). He also leads the Computer Integrated Manufacturing (CIM) platform of UGA. Emmanuel Simeu has been involved in nonlinear systems modelling and control. His current main field of research includes heterogeneous system modelling, reliability of embedded systems, diagnosis and control of analogue/RF circuits, statistical modelling and machine learning. He has served on the organization committees and the Technical Program Committees of several international conferences, including IEEE European Test Symposium (ETS), IEEE Latin America Test Symposium (LATS), IEEE On-Line Test Symposium (IOLTS), IEEE International Mixed-Signal, Sensors, and Systems Testing Workshop

(IMS3TW, IFIP/IEEE International Conference on Very Large Scale Integration (VLSI-SoC).



Maurice Tchuente is the Board Chairman, University of Yaounde II and former Pro-Chancellor, University of Buea in Cameroon. Professor of computer science, University of Yaounde I, Member of the Pan African University Council, the Scientific Council of IRD, France and Chairman of the Scientific Council of the international laboratory LIRIMA. He is docteur d'état es sciences mathématiques, Université Joseph Fourier, Grenoble and

HDR in Computer Science, INPG, Grenoble. Among his former

positions, he was coordinator (interim rector), Pan African University in 2012, Directeur de Recherche, IRD, France from 2008 to 2010, and chairman of the National Agency for Information and Communication Technologies, Rector of the universities of Dschang (1996–1998), Ngaoundere (1998–2000) and Douala (2000–2002), and minister of higher education (2002–2004). His research topics are parallel algorithms, dynamics of automata networks, epidemiological modelling and data mining. He is the co-author of more than one hundred research papers and author of a book entitled “Parallel Computation on Regular Arrays” published by Manchester University Press. He has supervised more than 20 Ph.D. thesis. He is a founding member of CARI, an African Network for Research in mathematics and Computer Science, and his research team is a component of UMMISCO (Unité de Modélisation Mathématique et Informatique des Systèmes Complexes). He received the Boutros Boutros Ghali Award of the United Nations University in 1995 and the CNR Rao Prize for Scientific Research, TWAS, 2007.