Université de Montréal

Conversion d'informations de protection dans les réseaux optiques

par

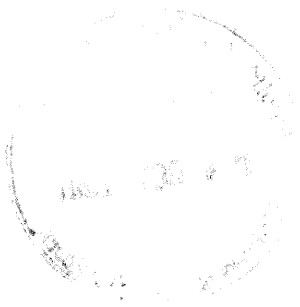
Éric Lesage

Département d'informatique et
de recherche opérationnelle

Faculté des arts et sciences

Mémoire présenté à la Faculté des études supérieures
en vue de l'obtention du grade de
Maître ès sciences (M. Sc.)
en informatique

Juillet 2001



Page d'identification du jury

Université de Montréal
Faculté des études supérieures

Ce mémoire intitulé :

Conversion d'informations de protection dans les réseaux optiques

présenté par :

Éric Lesage

a été évalué par un jury composé des personnes suivantes :

Neil Stewart
(président-rapporteur)

Marc Feeley
(membre du jury)

Esma Aimeur
(directrice de maîtrise)

Rachida Dssouli
(codirectrice de maîtrise)

Mémoire accepté le : 22 août 2001

Sommaire

Notre travail se situe dans le cadre des réseaux de transport de données de grande capacité fonctionnant sur fibres optiques.

Il consiste à permettre à un système de gestion de réseaux de transport d'exporter des informations via une interface normalisée récemment définie par le TeleManagement Forum, nommée Multi-Technology Network Management 2.0 (MTNM 2.0). Plus spécifiquement, nous convertissons à partir d'une représentation interne vers la représentation normalisée les informations ayant trait aux systèmes de protection, c'est-à-dire les systèmes qui assurent la survivabilité du réseau de transport en cas de pannes.

La représentation interne utilise des concepts provenant des normes de l'International Telecommunications Union que nous expliquons dans la première partie de ce travail. Ces concepts permettent de modéliser une représentation fonctionnelle d'un réseau de transport de façon indépendante de la technologie sous-jacente. Nous appliquons cette représentation aux réseaux basés sur la technologie Synchronous Digital Hierarchy.

Un des buts visés est de convertir ces concepts et les relations sémantiques exprimées par cette représentation interne (par des règles de connexion entre et dans les différentes couches-réseaux) vers les objets de la norme MTNM, appelés *groupes de protection.*

Un autre but de ce travail est de découvrir le fonctionnement d'une deuxième interface interne qui n'était pas documentée, afin d'enrichir les informations que nous pouvons publier dans l'interface normalisée. Pour ce faire, nous avons procédé par ingénierie inverse. Cette partie du travail fut rendue difficile étant donné le manque d'équipements requis pour conduire notre enquête.

Table des matières

Liste des figures

Liste des tableaux

Lexique

Acronyme	Description	Première utilisation
ADM	Add/Drop Multiplexer	25
APS	Automatic Protection Switching	28
ATM	Asynchronous Transfer Mode	37
AU	Administrative Unit	27
AUG	Administrative Unit Group	27
BER	Bit Error Rate	38
BLSR	Bilateral Line Switched Ring	59
COCGNB	Collector - Graphical Network Browser	68
CORBA	Common Object Request Broker Architecture	54
CTP	Connection Termination Point	36
DCN	Data Control Network	30
DD	Design Document	72
DN	Distinguished Name	57
DWDM	Dense Wave Division Multiplexing	38
DXC	Digital Cross-Connect	33
EML	Element Management Layer	54
EMS	Element Management System	54
ETSI	European Telecommunications Standard Institute	23
FMMOA	Fault Manager - Managed Object Agent	68
GNB	Graphical Network Browser	67
HOP	High Order Path	40
HOVC	High Order Virtual Container	27
IA	Indirect Adapter	36

IDL	Interface Description Language	59
ISDN	Integrated Services Digital Network	23
ITU	International Telecommunication Union	16
LOP	Low Order Path	40
LOVC	Low Order Virtual Container	27
MD	Mediation Device	67
MOA	Managed Object Agent	67
MS	Multiplexing Section	25
MSSPRING	Multiplex Section Shared Protection Ring	45
MSP	Multiplex Section Protection	42
MSOH	Multiplexing Section OverHead	28
MTNM	Multi-Technology Network Management	16
MVCI-N	Multi-Vendor CORBA Interface - Northbound	67
NE	Network Element	53
NEF	Network Element Function	52
NML	Network Management Layer	54
NMS	Network Management System	54
OC	Optical Carrier	23
OSF	Operations Systems Function	52
OMG	Object Management Group	111
PDH	Plesiochronous Digital Hierarchy	22
PMS	Physical Media Section	78
POH	Path OverHead	26
PTP	Physical Termination Point	56
RDN	Relative Distinguished Name	57
RS	Regenerative Section	25
RSOH	Regenerative Section OverHead	28
SDH	Synchronous Digital Hierarchy	15
SNCP	SubNetwork Connection Protection	62
SOH	Section OverHead	27
SONET	Synchronous Optical Networks	23
STM	Synchronous Transport Module	27
TF	Transormation Function	52

TM	Trail Manager	67
TMMOA	Trail Manager - Managed Object Agent	67
TMF	TeleManagement Forum	51
TMN	Telecommunication Management Network	51
TP	Termination Point	35
TTP	Trail Termination Point	36
TU	Tributary Unit	26
TUG	Tributary Unit Group	26
VC	Virtual Container	26
WS	Wireless Section	79
WSF	WorkStation Function	52
XDR	External Data Representation	67

À mes parents

Chapitre 1

Introduction

Les communications par fibres optiques sont de plus en plus importantes; on le sait, les réseaux de données basés sur cette technologie sont maintenant omni-présents. Comme toute technologie, la fibre optique n'est pas sans faille : il peut se produire des pannes d'équipement, des coupures de fibres dues à des catastrophes naturelles, à un rongeur ou à la négligence humaine. Afin d'accroître la fiabilité de ces réseaux dont l'augmentation en capacité est le reflet de leur augmentation en importance, des mécanismes de protection on été conçus.

Ces mécanismes de protection procèdent de deux façons : premièrement, de l'équipement est ajouté pour servir à transporter des données de façon redondante. C'est la méthode la plus simple, mais aussi la plus coûteuse de s'assurer que l'intégrité des données ne sera pas compromise par une panne isolée.

Deuxièmement, il s'agit de trouver des méthodes pour partager cette infrastructure supplémentaire entre plusieurs systèmes afin de réduire les coûts de son déploiement.

1.1 Objectifs

Le travail que nous avons effectué est réalisé dans le cadre des réseaux optiques de type SDH (Synchronous Digital Hierarchy). Les réseaux SDH sont des systèmes de transport; leur but est de permettre le transport à large échelle de données, sans toutefois que l'information y soit

produite ou ultimement consommée; c'est un système qui permet à un opérateur de réseau de fournir des liens statiques (c'est-à-dire des liens d'un point fixe à un autre – la route que prendront les données est fonction de l'origine de celles-ci, pas fonction des données elles-mêmes) à des clients qui désirent transférer de l'information du niveau métropolitain jusqu'à l'intercontinental. Un réseau de transport commute des liaisons : le format des données transportées n'est qu'une suite de bits. Les liaisons sont établies à long terme par l'opérateur du réseau (contrairement à un réseau téléphonique, où l'utilisateur établit lui-même des connexions de courte durée).

Le problème auquel nous nous sommes attaqué dans ce travail a été celui de présenter sous une forme normalisée les informations de protection dans un réseau SDH.

Nous étions en présence d'un système de gestion de réseaux SDH qui a ses propres interfaces internes de représentation des informations et nous désirions convertir ces informations afin de pouvoir fournir une interface normalisée, appelée MTNM 2.0 (Multi-Technology Network Management 2.0), et conçue par le TeleManagement Forum.

Cette interface normalisée permet à divers systèmes de gestion de réseaux, provenant de divers fournisseurs, de communiquer ensemble. Un des aspects normalisés par MTNM 2.0 consiste en l'information de protection.

MTNM 2.0 utilise pour représenter la protection un concept où les ports sont reliés ensemble en des *groupes de protection* de divers types. Notre travail consistait donc à déduire, à partir des interfaces existantes, la présence et les attributs de ces groupes.

Dans le système de gestion de réseaux avec lequel nous avons travaillé, il existe deux interfaces fournissant des informations de protection : la première interface est celle de gestion des voies de communication (*trail management*). (Les voies de communication sont les routes utilisées pour acheminer l'information dans un réseau de son point d'entrée à son point de sortie.) Cette interface est basée sur une méthode de représentation des réseaux de transport créée par l'ITU (International Telecommunication Union).

La seconde interface est celle de gestion des fautes (*fault management*). Les fautes, ou alarmes, indiquent où sont les problèmes à un certain moment donné dans le réseau géré; le modèle de données employé par cette interface ne semble pas être basé sur une norme particulière (d'ailleurs, nous ne connaissons aucune norme de représentation de telles informations). La difficulté avec cette interface est qu'elle n'était pas documentée. Il a donc fallu procéder par

ingénierie inverse pour découvrir son fonctionnement et en soutirer les informations concernant l'état du réseau (nous savions que ces informations étaient présentes puisqu'elles étaient affichées par une autre composante qui utilisait cette interface).

Notre travail a donc été de concevoir et d'implanter une fonction de conversion qui utilise les informations disponibles via ces deux interfaces pour fournir des données de protection sur l'interface normalisée de MTNM 2.0. Ce fut un travail où nous avons appliqué une norme industrielle pour permettre au système de gestion de réseau sur lequel nous avons travaillé d'être interopérable avec d'autres systèmes.

Ce mémoire discute donc de ce travail dans le domaine des télécommunications effectué dans un contexte industriel ; le résultat en est un logiciel qui est utilisé en pratique.

1.2 Contenu

La première partie de ce mémoire consiste en une synthèse des bases théoriques sur lesquelles le travail d'implémentation, décrit dans la deuxième partie, est fondé.

Le chapitre 2 explique le fonctionnement général des réseaux optiques SDH, y compris les différentes étapes de transformation entre l'entrée de données dans le réseau, leur transport, et leur sortie.

Le chapitre 3 introduit un modèle abstrait de représentation des réseaux de transport de données. Ce modèle abstrait n'est pas limité à SDH, mais nous en tirons un cas particulier et expliquons SDH vis-à-vis de ce modèle. Nous expliquons aussi les concepts de protection, tels les différents types de protection ainsi que les caractéristiques fondamentales du type de protection étudié. Nous indiquons comment la protection est représentée dans le modèle abstrait.

Le chapitre 4 décrit la norme MTNM 2.0, c'est-à-dire quelle est sa portée ; pour ce, nous introduisons un modèle abstrait de réseau de gestion et nous indiquons quel rôle MTNM 2.0 y joue.

Le chapitre 5 fait le pont entre la première partie, théorique, et la seconde partie, décrivant le travail effectué. Dans ce chapitre, l'architecture spécifique du système de gestion de réseau utilisé est décrite, autant en elle-même que sur les bases du modèle abstrait du chapitre 4. Ce

chapitre explique l'architecture interne du bloc fonctionnel dans lequel nous avons implanté le système de conversion des informations de protection.

La deuxième partie commence au chapitre 6 ; ce chapitre décrit les principes méthodologiques de développement logiciel utilisés pour parvenir à concevoir et implémenter un système fonctionnel.

Le chapitre 7 décrit les structures de données employées concrètement et fait la relation entre elles et le modèle, très proche, décrit au chapitre 3. Les algorithmes de conversion des informations topologiques provenant de ces structures sont expliqués, ainsi que la façon de gérer les informations de protection introduites au chapitre 4.

Le chapitre 8 décrit les structures de données et les algorithmes employés pour convertir les informations concernant l'état du réseau vers l'information de protection de l'interface abordée au chapitre 4.

Le chapitre 9 traite des changements requis pour faire interagir les systèmes internes de gestion des données décrits aux chapitres 7 et 8 avec l'interface MTNM décrite au chapitre 4.

Enfin, nous concluons sur notre contribution et nous discutons des ouvertures que ce travail offre.

Chapitre 2

Les réseaux optiques

Les *réseaux optiques* sont les systèmes qui acheminent des données à partir de méthodes de transmissions photoniques, contrairement aux systèmes électroniques ou radiatifs.

2.1 Transmission optique de données

La transmission de données par voie optique n'est pas une nouvelle idée. Déjà, en 1881, Alexander Graham Bell brevetait le photophone [Bel1881], appareil qui convertissait une conversation en signaux lumineux voyageant directement dans l'air.

Ce ne fut que bien plus tard que les *fibres optiques* furent développées. Une fibre optique transmet (ou guide) une onde selon le principe de réflexion totale interne. Ce principe est bien connu ; la première démonstration enregistrée est celle d'une présentation de John Tyndall devant l'Institut Royal de l'Angleterre en 1854. Toutefois, ce ne fut pas avant 1950, avec la publication simultanée des travaux de A. C. S. Van Heel et de Hopkins et Kapany ayant trait à la transmission d'images par des fibres de verre, que l'intérêt envers cette discipline prit de l'ampleur [All1973]. N. S. Kapany [Kap1967] relate plusieurs développements successifs importants du domaine :

1956 Invention de l'appellation "fibre optique", description générale des principes et applications.

1958 Fabrication de fibres multiples, dont une des retombées fut de réduire le diamètre minimal

de fibres.

1959 Développement de fibres de verre avec revêtement en verre (qui élimine les pertes de lumières du revêtement en plastique ainsi que les problèmes d'isolation optique des fibres sans revêtement).

1957-1960 Recherche sur les applications des fibres optiques en photographie à haute-vitesse.

1960 Premières fibres optiques hors du spectre visible (dans le domaine infrarouge).

Au même moment, les recherches dans le domaine des *lasers* conduites aux Bell Labs commencent à porter fruits ; le concept, qui y est inventé en 1958, résulte en la création en 1960 du premier laser à action continue ; et en 1971, du premier laser qui fonctionne à température ambiante [Cun1991].

C'est la combinaison des domaines de la fibre optique et des lasers qui permet de penser à l'application au domaine du transport de données : l'onde émise par un laser est modulée pour transporter un signal via la fibre optique, jusqu'à ce qu'un photodétecteur transforme le signal optique sous forme électrique. Toutefois, l'application reste impossible pour plusieurs années étant donné que les fibres développées jusqu'alors causent des pertes trop importantes du signal. Ce n'est qu'en 1970 que trois chercheurs de Corning Glass Works (Kapron, Keck et Maurer) font la première démonstration de transert de données par une fibre offrant une perte de signal relativement faible (20 dB/km) ; depuis, les fibres ont encore été améliorées pour offrir une perte de moins de 0,01 dB/km [Ren1993].

Les fibres optiques ont plusieurs avantages qui favorisent leur utilisation par rapport aux autres médiums de communication électriques (les câbles coaxiaux et les paires torsadées). Il est notamment mentionné dans [Fre1996] :

- Vitesse de transmission plus élevée ; la lumière voyage pratiquement à la même vitesse dans une fibre que dans le vide, tandis qu'un signal électrique se propage de 50 à 80% de cette vitesse.
- Capacité de transmission plus élevée.
- Absence d'interférence électromagnétique générée par le cable, insusceptibilité à ces interférences, aux éclairs ou aux radiations nucléaires. Cela en fait aussi des conducteurs plus résistants à l'espionnage (on doit intercepter directement le signal pour le divertir ou le dupliquer, et une telle manipulation cause une perte de signal et peut être détectée par réflexométrie).

- Absence d'interférences dues à la boucle de mise à terre, à la réflection ou à la diaphonie (*cross-talk*).
- Atténuation de signal moins rapide que pour les câbles électriques, donc moins de répéteurs requis.
- Taux d'erreurs plus faibles que sur les câbles électriques (jusqu'à 3 ordres de grandeur).
- Pas de risque de court-circuit ni de décharges électriques.
- Les fibres sont plus petites et requèrent moins d'espace. Elles sont aussi du dixième du poids des câbles de cuivre.
- Plus grande résistance à la corrosion et aux liquides que les câbles.
- Grande disponibilité des matériaux de base.
- Possibilité d'accroître la capacité en multiplexant plusieurs longueurs d'ondes sur des fibres déjà existantes sans causer d'interférences avec les longueurs d'onde déjà présentes.
- Coûts d'installation et d'entretien plus faibles sur de longues distances. Vie opérationnelle et temps entre les pannes des fibres plus faible que pour les câbles électriques.

Différents types de fibres optiques existent pour répondre à différents besoins. En ce qui concerne la transmission de données, application qui nous intéresse, une des caractéristiques fondamentales est la longueur d'onde utilisée. Trois plages de longueurs d'ondes sont communément employées : 850 nm, 1300 nm et 1500 nm. Plus l'onde est longue, plus la distance à parcourir avant que le signal soit dégradé peut être grande [Sta1988] ; par conséquent, pour les réseaux de longue distance, un taux de données plus élevé peut être atteint avec une longueur d'onde plus grande.

À quoi destine-t-on toute la capacité offerte par les fibres optiques ? Une des premières applications imaginées fut l'agrégation de circuits de communication (*trunking*) [Sat1996]. L'agrégation résulte de besoins économiques d'augmenter la capacité de transmission. Originellement, le réseau téléphonique était basé sur le fait qu'un circuit [1] prenait une paire de fils ; les troncs interurbains requéraient donc des grands circuits multi-paires. Ces câbles étaient coûteux et prenaient beaucoup de place. En 1962, le Bell System mit en service le "T1 Carrier System", qui était la première application à large échelle des techniques numériques de transmission, incluant le multiplexage temporel [Pic1981]. Grâce à ce système, 24 circuits (conversations) ainsi que les informations de signalisation pouvaient être comprimés sur une seule paire de fils.

[1] Un circuit est un chemin emprunté au travers d'un réseau ; dans le réseau téléphonique, un circuit est utilisé pour transporter une conversation.

Plus le temps avance, plus la quantité de données à transmettre augmente. Heureusement, grâce aux progrès techniques, le multiplexage peut être réalisé pour des volumes de plus en plus grands de données.

Le premier "système" inventé pour parvenir à acheminer de grandes quantités de données porte aujourd'hui le nom de PDH (Plesiochronous Digital Hierarchy). Dans ce système, le taux de base est DS-1, qui est le même que le taux employé par le système T1 pour transmettre des données (1.544 Mbps). Pour véhiculer davantage de données, on combine quatre circuits DS-1 ensemble pour former des DS-2, de capacité supérieure. On peut ensuite agglomérer sept DS-2 ensemble pour former des DS-3, etc. [Haj2000]. Ainsi, une hiérarchie de taux de données est définie, du plus petit (DS-1) au plus grand (DS-4).

Les fibres optiques constituent un moyen de multiplexer encore plus de circuits dans un espace encore plus restreint. Nous nous attarderons ici aux réseaux optiques de haute capacité ; en effet, le fait d'utiliser la fibre optique ne dicte pas en soi une quelconque vitesse de transmission ou un protocole de couche physique particulier. En outre, les systèmes tels que Gigabit Ethernet [IEEE2000] peuvent utiliser comme médium de support la fibre optique et ainsi être considérés comme étant des réseaux optiques.

Nous nous intéressons plutôt aux systèmes basés sur les normes SDH (Synchronous Digital Hierarchy) [ITUg707]. Le système SDH est une évolution par rapport au plus ancien système PDH. Le système SDH a ceci de particulier qu'une horloge maître est utilisée pour la synchronisation de tout le réseau.

Le rôle des deux systèmes est similaire : il consiste en l'agrégation et la commutation d'unités de transmission. Par exemple, un opérateur de réseau pourrait désirer fournir des lignes de faible capacité (par exemple, DS-1) à différents clients qui emprunteront ce réseau pour acheminer des données. Or, à l'intérieur du réseau de l'opérateur, ce dernier ne maintiendra pas des lignes à faible capacité d'un bout à l'autre : les données seront agglomérées sur des fibres permettant de transmettre à plus haut débit. Par exemple, il est envisageable que plusieurs clients voudront transmettre des informations de Montréal à New York. L'opérateur de réseau installera alors un lien haute-capacité entre ces deux villes et agglomérera toutes les lignes à faible capacité de ses clients sur ce lien unique, ce qui lui permettra de réaliser des économies d'entretien et de gestion. Les unités de données du client seront commutées de leur point d'entrée du réseau jusqu'à leur point de sortie.

2.2 Introduction à SDH

La norme SDH fut développée par la compagnie Bellcore sous la désignation SONET (Synchronous Optical Networks). Depuis qu'elle a été adoptée au niveau international par l'organisme de normalisation ITU-T, deux façons de l'utiliser se sont répandues : la façon nord-américaine (provenant de SONET), basée sur l'unité de transmission OC-1 (Optical Carrier 1), qui correspond à 810 DS-1, et la façon européenne (promulguée par l'ETSI, European Telecommunications Standards Institute), basée sur l'unité de transmission STM-1 (Synchronous Transport Module 1), qui correspond à 2430 DS-1.

Le Tableau 2.1 donne les débits de transmission spécifiques aux systèmes SDH. Comme on peut le constater, la notation OC-x (ou STS-x) est équivalente à la notation STM-y, où $x = y/3$. Dans le reste de ce document, nous n'allons utiliser que la terminologie européenne.

TAB. 2.1 – Débit des modules SDH

Désignation SDH	Désignation SONET	Débit (kb/s)
STM-1	OC-3	155 520
STM-4	OC-12	622 080
STM-16	OC-48	2 488 320
STM-64	OC-192	9 953 280
STM-256	OC-768	39 813 120

L'on peut se demander quelle est l'origine de ces vitesses de transmission autrement inexpliquées. L'origine de la hiérarchie prend son fondement dans le réseau téléphonique. Au départ, le réseau analogique a été conçu pour transmettre des fréquences entre 400 et 3400 Hertz, permettant une bande passante de 3000 Hz. Lors de la conversion au numérique (dans le système T1 mentionné précédemment), une largeur de bande légèrement supérieure de 4000 Hz a été préférée. Le théorème d'échantillonnage de Nyquist [Hal1994] spécifie qu'une telle largeur de bande demande un taux d'échantillonnage deux fois plus élevé, donc de 8000 échantillons par seconde. La quantification est effectuée à raison de 8 bits par échantillon, ce qui permet 256 niveaux. Par conséquent, le signal résultant, une fois numérisé, aura un débit de $8000 \times 8 = 64000 bits/seconde$, ce qui correspond au taux DS-0.

Le taux DS-0 correspond également à la largeur de bande d'un canal ISDN (Integrated

Services Digital Network). [ITUi120]. En fait, un canal ISDN consiste simplement à permettre l'utilisation d'un canal vocal pour la transmission directe de données par le client (les informations de contrôle de ce canal occupent toutefois un canal à part).

L'agrégation à plus haut débit de canaux DS-0 se fait différemment selon qu'on se trouve en Amérique du Nord ou en Europe. En Amérique du Nord, 24 canaux sont multiplexés ensemble en temps (Time Division Multiplexing) dans chaque trame DS-1, et un bit de synchronisation est ajouté au début de chaque trame. Une trame DS-1 couvre la même unité de temps qu'un échantillon de canal DS-0, soit 1/8000e de seconde ; dans cette période, on doit donc transmettre 1 bit de synchronisation plus 24 fois 8 bits de données, pour un total de 1544 kilobits par seconde. En Europe, on agglomère ensemble 30 canaux, on en ajoute un artificiellement pour la synchronisation et un autre pour la signalisation. Cela fait un total de 32 canaux à 8 bits de données 8000 fois par seconde, pour un total de 2048 kilobits par seconde, soit un canal E1.

Ces deux types de canaux (DS-1 et E-1) sont les bases des systèmes SDH et PDH.

Dans le système PDH, pour arriver aux débits supérieurs, comme nous l'avons déjà mentionné, on combine à nouveau ces agrégats en des débits de plus en plus élevés. Étant donné que dans ce système la synchronisation n'est pas centralisée, on doit ajouter à chaque niveau des bits de plus qui serviront comme bits de justification. Le rôle de ces bits est de compenser pour les écarts locaux de synchronisation. La hiérarchie PDH est donnée dans le Tableau 2.2.

La difficulté dans ce système vient du fait que, pour extraire une partie des données contenues dans un agrégat, il faut complètement démultiplexer l'agrégat jusqu'au niveau qui nous intéresse. Les autres désavantages, mentionnés en outre dans [Sat1996], sont l'utilisation du multiplexage par bits au lieu d'octets (ce qui est plus lourd), des structures de trames différentes à chaque niveau de la hiérarchie, peu de possibilités de gestion avancée en raison du manque de place pour les informations de gestion dans les trames, et l'inexistance d'une norme mondiale.

Par exemple, supposons qu'on transporte des données de Montréal à Toronto en passant par Ottawa, et que la plupart des données transitent par un lien à 140 mégabits par seconde commençant à Montréal et finissant à Toronto. Si un client désire obtenir une DS-1 de Montréal à Ottawa uniquement, il faudra démultiplexer totalement le lien de 140 mégabits jusqu'au niveau DS-1, et ensuite remultiplexer les connexions restantes jusqu'au niveau de 140 mégabits par seconde.

TAB. 2.2 – Débits du PDH

Désignation du taux	Débit (kb/s)
DS-0	64
DS-1	1 544
DS-2	6 312
DS-3	44 736
DS-4E	139 264
DS-4	274 176
E-1	2 048
E-2	8 448
E-3	34 368
E-4	139 264
E-5	565 148

Ce genre d'opération est très coûteux en termes d'équipement (l'ensemble de l'équipement qui sert à ce genre de traitement se nomme un Add/Drop Multiplexer (ADM)), mais aussi en termes de gestion, qui doit être plus ou moins manuelle.

Le système SDH s'évertue à résoudre ce problème. D'abord, comme nous l'avons déjà mentionné, la synchronisation est centralisée, éliminant en grande partie le besoin de bits de justification. SDH introduit trois niveaux topologiques à la base de son fonctionnement.

Le niveau topologique fondamental consiste en des sections régénératives (Regenerative Section, RS). Une section régénérative correspond à une étendue continue de fibre et peut être terminée par un amplificateur de signal, par exemple. Une RS est quelquefois appelée simplement "section" dans la littérature [Hal1994].

Vient ensuite la section de multiplexage (Multiplexing Section, MS). Une section de multiplexage est un agencement de plusieurs sections de régénération consécutives où les données transportées par le signal ne sont pas modifiées. Une MS peut être terminée à chaque bout par un ADM, par exemple. Une MS est quelquefois appelée "ligne" dans la littérature [Hal1994].

Finalement, on introduit le chemin (*path*). Un chemin est une suite de lignes qu'emprunteront des données à partir de leur introduction dans le réseau SDH jusqu'à leur sortie.

2.3 Interface PDH / SDH

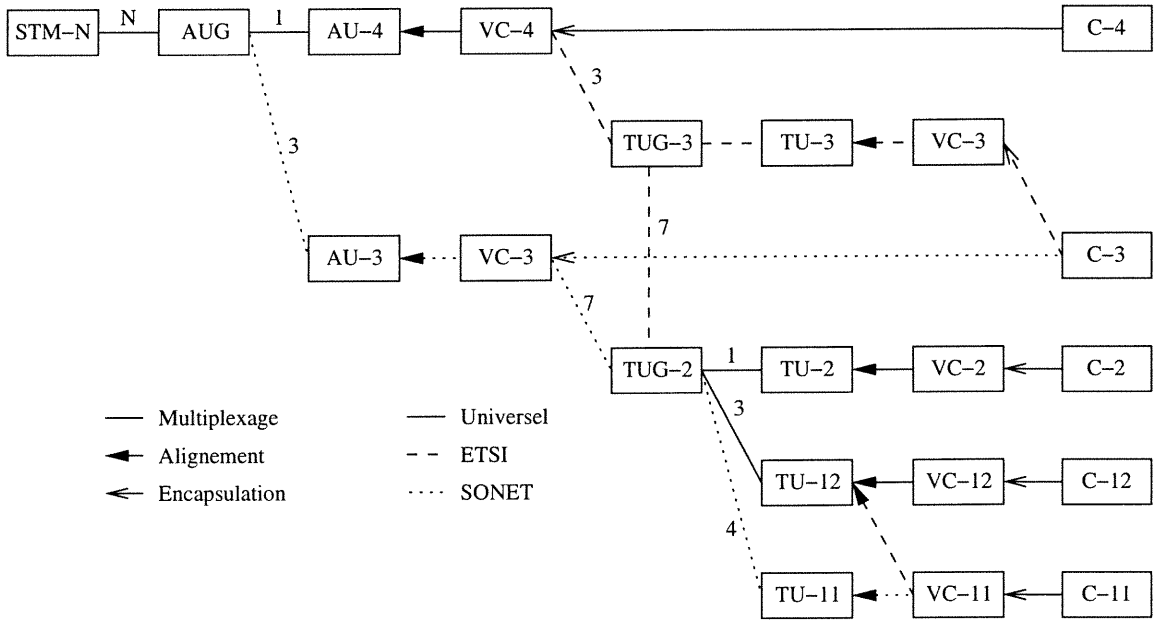


FIG. 2.1 – Options d'intégration PDH et SDH

Comme on le voit à la Figure 2.1, les données transportées par un réseau SDH viennent souvent de réseaux ou de liens formatés selon les normes PDH. Il doit donc se faire une adaptation entre les deux types de réseaux.

La solution employée consiste à utiliser des *conteneurs*. Plusieurs d'entre eux sont définis (de C-1 à C-4) ; ceux-ci encapsulent toute l'information contenue dans une trame PDH à partir de leur entrée et jusqu'à leur sortie du réseau SDH.

Pour gérer un conteneur, on ajoute des informations ancillaires attachées à ce conteneur pour tout son chemin (Path Overhead, POH). Le conteneur et ces informations attachées constituent ensemble un *conteneur virtuel* (VC). Un VC constitue une unité de base indécomposable dans un réseau SDH ; c'est cette unité qui sera commutée d'un bout à l'autre du réseau.

Les VC peuvent être combinés de diverses façons, comme l'indique la Figure 2.1. Les VC sont d'abord encapsulés en unités tributaires (TU) ; les TU sont les unités de base pour être combinées en TUG (Tributary Unit Groups).

Circuit PDH	Conteneur SDH
DS-1	C-11
E-1	C-12
DS-2	C-2
DS-3	C-3
E-4	C-3
DS4-E	C-4
E-5	C-4

TAB. 2.3 – Relation entre circuits PDH et conteneurs SDH

Les TUG finissent par être ré-encapsulés à l'intérieur de VC-3 ou de VC-4. Ces derniers VC sont les VC de haut niveau (à noter qu'un VC-3 peut être autant un VC de haut que de bas niveau). De tels VC sont encapsulés à l'intérieur de AU (Administrative Units), lesquels sont groupés ensemble en AUG (Administrative Unit Groups). Enfin, à un AUG on ajoute des informations ancillaires de section (SOH, Section OverHead) pour former un module (STM-1).

Le problème de la synchronisation des données provenant du PDH vers l'horloge SDH est résolu grâce aux pointeurs contenus dans les TU et les AU. On permet à un VC de ne pas commencer exactement à un endroit fixe du TU ou de l'AU. Cependant, à des endroits fixes dans ceux-ci se trouvent des pointeurs qui spécifient l'endroit exact dans la trame où commencent les VC. De tels pointeurs permettent donc de compenser de légères variations dans le temps dans l'alignement des données. Cela implique toutefois qu'une certaine partie d'un module SDH soit allouée en espace inutilisé pour permettre une certaine flexibilité dans l'alignement.

La distinction entre AU et TU est mince et plutôt terminologique ; un TU soutient la fonction d'adaptation de VC de bas niveau (LOVC, Lower Order VC) vers un VC de haut niveau (HOVC, Higher Order VC), tandis qu'un AU soutient la fonction d'adaptation de VC de haut niveau vers un module. Dans chacun des cas, l'adaptation consiste à ajouter des pointeurs à un endroit fixe pour indiquer l'emplacement réel du VC. Les groupes (TUG et AUG) constituent quant à eux des structures de multiplexage. La façon précise de faire est décrite dans la norme [ITUg707].

Cette encapsulation à outrance peut a priori sembler être une difficulté équivalente au système PDH. Ce n'est pas le cas toutefois, car même si on monte plus haut dans la hiérarchie SDH, l'unité de données est toujours le VC-3 ou le VC-4. En théorie, on peut donc faire un ADM d'un

seul conteneur VC-1 en provenance d'un signal STM-64. En pratique, une telle configuration est rarement utile.

Les modules STM-1 sont combinés ensemble pour faire des connexions allant de STM-1 à STM-256 et au-delà. Contrairement à PDH toutefois, il n'y a pas de bits de justification introduits par les niveaux supérieurs. Ainsi, un lien STM-256 a une capacité exactement 256 fois plus grande qu'un lien STM-1 ; cela est possible étant donné la synchronisation centralisée (le fait que tous les éléments du réseau soient coordonnés par la même horloge) de SDH, son grand avantage par rapport à PDH.

Nous avons mentionné ci-dessus qu'un module était composé d'un AUG et d'un SOH. Le SOH est quant à lui divisé en deux. La première partie est le RSOH (Regenerator Section OverHead) ; cette partie est spécifique à une section de régénération ; elle contient des octets d'indication de début de trame, un canal de communication (*orderwire*) vocal (pour permettre aux techniciens aux deux extrémités de la section de communiquer), un octet pour transmettre un numéro d'identification de section, un octet pour indiquer un calcul de parité sur la trame précédente, un canal de communication de données et un canal réservé à l'utilisateur.

La deuxième partie est le MSOH (Multiplex Section OverHead). Cette partie contient des informations qui servent à gérer une section de multiplexage (ligne) au complet ; elle n'est pas modifiée par les générateurs s'y trouvant. Elle comporte, elle aussi, un canal de communication vocal et un octet pour indiquer un calcul de parité (effectué sur toute la trame excepté sur le RSOH). En plus, elle contient un canal qui indique l'état de la synchronisation (pour la propagation adéquate des informations temporelles) et un canal qui indique la détection d'erreurs. Les données les plus intéressantes sont cependant les informations transmises dans les octets nommés K1 et K2. Ces octets sont réservés pour les informations concernant la commutation automatique de protection (APS, Automatic Protection Switching). C'est grâce au protocole APS que les systèmes de protection présentés au chapitre 3 peuvent être implémentés.

2.4 Conteneurs non équipés

Les conteneurs VC4 ne sont pas nécessairement utilisés pour transférer des informations en provenance d'un réseau PDH. Aussi, il existe un indicateur qui permet de spécifier que le VC contient des données brutes, au lieu d'être sous-structuré. On dit alors que le VC est non équipé.

Il existe aussi des conteneurs génériquement appelés VC4-Xc, où X est un entier qui indique le nombre de trames que dure un VC. Normalement, un VC est multiplexé sur une seule trame ; un VC4-Xc a donc une capacité X fois plus grande. Pour l'instant, VC4-Xc est défini pour X = 4, 16 et 64.

2.5 Résumé

Dans ce chapitre, nous avons présenté les réseaux optiques. Nous avons énuméré les avantages des fibres optiques par rapport aux autres conducteurs ; nous avons expliqué comment leur utilisation était pertinente dans les réseaux de transports.

Nous avons expliqué que le principe qui permettait aux réseaux de transport d'offrir un avantage économique était l'agrégation des données ; les deux systèmes qui ont été employés à cette fin furent d'abord le PDH, qui tire son origine du "T1 Carrier System", et par la suite le SDH, qui nous intéresse plus particulièrement.

Nous avons montré comment le système SDH fonctionne et permet la commutation de conteneurs virtuels, lesquels servent à transporter l'information d'un bout à l'autre du réseau SDH. Nous avons expliqué comment les anciens systèmes PDH pouvaient être intégrés en tant que tributaires (ou réseaux d'accès) des réseaux SDH.

Dans le prochain chapitre, nous parlerons d'un modèle abstrait pour représenter les réseaux de transport. Nous expliquons aussi en détail les concepts de protection, qui sont essentiels pour assurer la fiabilité des réseaux de transport. Enfin, nous appliquons le modèle fonctionnel au système SDH en particulier, et démontrons comment ce même modèle permet de représenter les fonctions de protection offertes par SDH.

Chapitre 3

Concepts de protection et de modélisation réseau

Dans un système SDH, la commutation est configurée à distance, par télécommande logicielle. Cela peut être fait de plusieurs façons, en outre en utilisant un Data Control Network (DCN) pour contrôler les éléments du réseau.

Comme nous l'avons vu, les unités commutées dans le système SDH sont les VC. Or, les VC de haut niveau contiennent bien souvent des VC de bas niveau. Il ne faut pas qu'on puisse envoyer un VC de haut niveau à un endroit et envoyer ailleurs des VC de bas niveau contenus dans le VC de haut niveau.

Pour résoudre ce problème, on pourrait forcer l'utilisateur à configurer la commutation des VC de bas niveau directement, dès qu'un VC de haut niveau contient un TUG. Une telle approche serait toutefois très lourde en termes d'entretien et de gestion.

Au lieu de cela, un modèle fonctionnel a été conçu et normalisé dans [ITUg805]. Pour comprendre ce modèle, il faut d'abord quelques définitions tirées de cette norme.

3.1 Modèle fonctionnel d'un réseau de transport

Un *réseau* est l'ensemble des entités qui ensemble permettent de fournir des services de communication. Un *réseau de transport* comporte trois types de *composantes architecturales*, soient : les composantes topologiques, les entités de transport et les fonctions de traitement de transport. Le but d'un réseau de transport est de transmettre de l'information caractéristique (c'est-à-dire les données visant à être communiquées, par rapport aux informations portant sur la gestion même du réseau).

Deux phénomènes peuvent se produire dans un réseau de transport : l'information peut être traitée (c'est-à-dire que son contenu est modifié selon une transformation ou fonction mathématique) et l'information peut être transportée (c'est-à-dire déplacée entre des endroits différents). Pour ce faire, le modèle fournit, respectivement, les fonctions de traitement de transport et les entités de transport. (Notons que deux phénomènes particuliers ne peuvent pas se produire dans les réseaux de transport, soient la production et la consommation d'information caractéristique ; ces phénomènes peuvent se produire dans d'autres types de réseau, les réseaux d'accès.)

Une *corrélation* (*binding*) est une relation directe entre une fonction de traitement de transport ou entité de transport et une autre fonction de traitement de transport ou entité de transport qui représente un état de connectivité statique (qui ne peut pas être modifié directement par des commandes de gestion de réseau).

Un *point de référence* est une composante architecturale qui est formée en corrélant (bind) les entrées et sorties de fonctions de traitement de transport et/ou d'entités de transport. En d'autres termes, un point de référence représente l'existence d'une corrélation, soit une association permettant le transfert de données.

Les points de référence peuvent être regroupés ensemble selon leurs caractéristiques. Ces regroupements sont des composantes topologiques.

3.1.1 Composantes topologiques

Une *composante topologique* est une composante architecturale qui sert à décrire le réseau de transport en termes de relations topologiques entre des ensembles de points dans le même

réseau en couches. Il existe quatre composantes topologiques : le réseau en couches lui-même (l'entité englobante), le sous-réseau, le lien et le groupe d'accès (Figure 3.1).

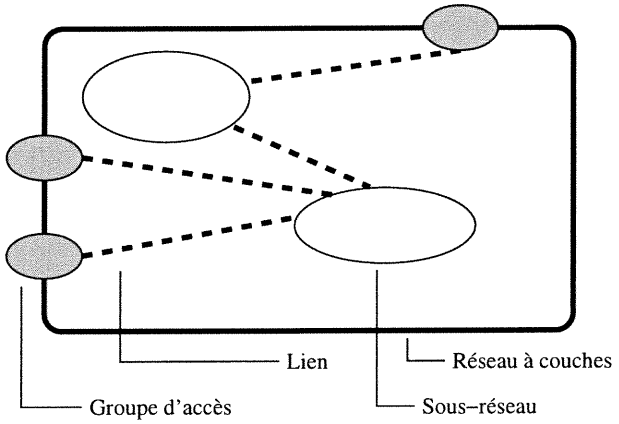


FIG. 3.1 – Composantes topologiques G.805

Un *réseau en couches* est défini par l'ensemble complet des groupes d'accès du même genre qui peuvent faire l'objet d'associations dans le but de transférer de l'information. Autrement dit, un réseau en couches est un peu comme une "boîte noire" dont les entrées et sorties sont ses groupes d'accès. Si l'on regarde à l'intérieur de cette boîte noire, on constate qu'un réseau en couches est construit d'entités de transport et de fonction de traitement de transport qui décrivent la génération, le transport et la terminaison d'information caractéristique. Les points de référence entre ces entités et fonctions forment les groupes d'accès, les liens et les sous-réseaux à l'intérieur du réseau en couches.

Les réseaux en couches ont, entre eux, des relations client/serveur : un réseau en couches client peut utiliser un réseau serveur pour assurer le transport d'information (dans ce cas, le client doit toujours ajouter de l'information permettant de vérifier la qualité du transfert d'information effectuée par le serveur; par exemple, en ajoutant une somme de contrôle avant de transmettre l'information). Les relations client/serveur servent à stratifier les réseaux; c'est de cette stratification que vient le terme "couche" ; les réseaux en couches sont appelés ainsi parce qu'ils sont organisés en couches, l'un sur l'autre. Deux réseaux en couches, l'un client, l'un serveur, sont illustrés à la Figure 3.3.

Un *groupe d'accès* est un groupe de fonctions de terminaison de voie (voir ci-dessous) qui sont connectées sur le même sous-réseau ou lien. Aucune règle ne dicte précisément comment les

groupes d'accès doivent être composés ; dans le cas dégénéré, chaque groupe ne pourrait contenir qu'une fonction de terminaison de voie ; en général, on les rassemble de telle façon à ce que la représentation du réseau en soit simplifiée. Cela implique en général que les fonctions se trouvant au même endroit sont réunies ensemble.

Étant donné qu'un groupe d'accès est formé de fonctions de terminaison de voie, c'est un point du réseau où un client peut se connecter et qui permet ainsi au réseau possédant le groupe d'accès de devenir le serveur du réseau qui s'y connecte.

Un *sous-réseau* est une composante topologique qui permet d'effectuer le routage d'information caractéristique. Un sous-réseau existe à l'intérieur d'un réseau en couches et est défini par l'ensemble des ports à sa frontière. Un sous-réseau peut à son tour être décomposé en sous-réseaux plus petits, jusqu'au cas limite, représenté par une matrice de connexions.

Sur le plan pratique, un sous-réseau est une portion d'un réseau en couches qui est délimitée surtout pour des raisons administratives. Par exemple, si un réseau en couches est partagé entre plusieurs compagnies, chaque portion gérée par une compagnie en particulier pourrait être considérée comme un sous-réseau. À son tour, chaque portion géographique du réseau d'une de ces compagnies pourrait être un sous-sous-réseau ; à une limite extrême, nous pourrions modéliser chacun des éléments du réseau (qu'il soit un ADM, un DXC (Digital Cross-Connect) ou un régénérateur) en tant que très petit sous-réseau. En résumé, un sous-réseau sert à partitionner un réseau ou sous-réseau englobant.

Un *lien* est une composante topologique qui décrit une relation fixe entre un port d'un sous-réseau ou un groupe d'accès d'un réseau et un autre port d'un sous-réseau ou un groupe d'accès d'un réseau. Un lien représente le fait que deux autres composantes topologiques sont physiquement connectées ; il est caractérisé par une certaine capacité de transport.

3.1.2 Entités de transport

Une *entité de transport* est une composante architecturale qui transfère des données de ses entrées vers ses sorties dans un réseau en couches. Les entités de transport sont les connexions et les voies. Il existe trois types de connexions : par lien, par sous-réseau et par réseau ; dans chaque cas, une connexion représente le transport d'information via la composante topologique en question. Les connexions sont délimitées par des points de connexion. Plus particulièrement,

une connexion sur un lien est délimitée par des points de connexions appelés ports ; une connexion sur un réseau est délimitée par des points de connexions de terminaisons.

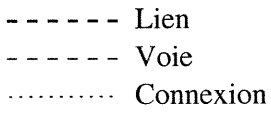


FIG. 3.2 – Représentation des connexions et des liens

Une *voie* représente le transfert adapté et surveillé d'information caractéristique (cela signifie qu'un certain contrôle est effectué pour vérifier la qualité des informations transférées sur une voie). Une voie est délimitée par deux points d'accès et est formée en liant à chaque bout de la voie deux fonctions de terminaison de voies à une connexion sur un réseau.

Lorsqu'une voie est formée, le réseau sous-jacent devient un serveur au réseau englobant, qui est son client (voir exemple à la Figure 3.3).

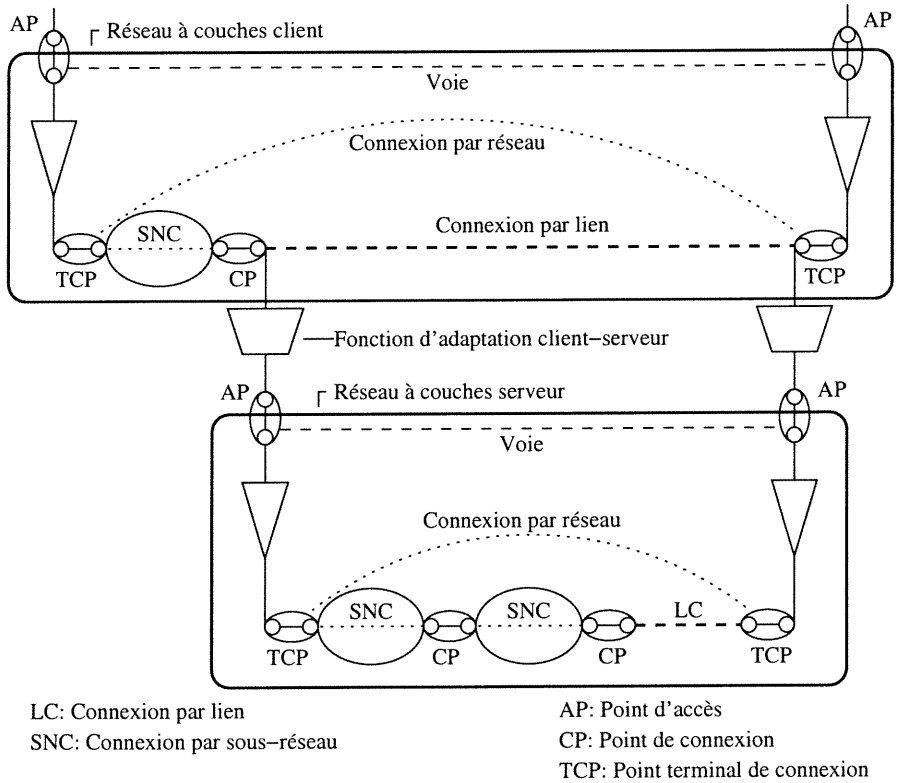


FIG. 3.3 – Entités de transport et relation client-serveur

3.1.3 Fonctions de traitement de transport

Une *fonction de traitement de transport* est une composante architecturale définie par le traitement d'information qui est effectué entre ses entrées et ses sorties. Il faut que l'entrée ou la sortie soit à l'intérieur du réseau en couches ; la sortie ou l'entrée correspondante peut, quant à elle, se situer dans le réseau de gestion.

Deux fonctions de traitement de transport sont définies : la fonction d'adaptation et la fonction de terminaison de voie.

La *fonction d'adaptation* consiste en le traitement qui doit être fait pour convertir l'information caractéristique du réseau en couches client vers une forme appropriée pour le transport sur le réseau en couches serveur, et vice-versa.

La *fonction de terminaison* de voie consiste en le traitement qui doit être fait pour convertir une information caractéristique adaptée à partir d'un réseau en couches client, y ajouter l'information nécessaire à la surveillance de la voie et présenter le tout à la sortie, et vice-versa.

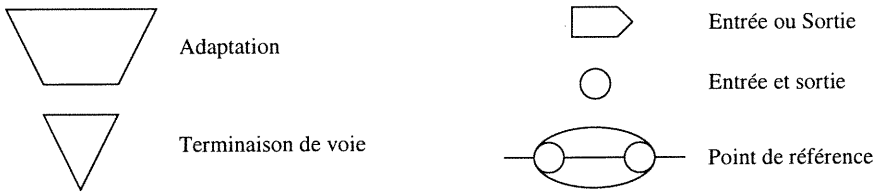


FIG. 3.4 – Fonctions de traitement et point de référence

3.1.4 Points de terminaison et adaptateurs indirects

Le modèle informationnel de réseau de l'ITU [M.3100] spécifie des entités qui généralisent les concepts discutés précédemment. Ces entités plus concrètes sont définies afin de pouvoir les utiliser comme objets sur lesquels une interface de gestion pourra se baser pour transférer de l'information. Certaines de ces entités ont une corrélation directe avec le modèle des réseaux de transport. Ce sont les points de terminaison et les adaptateurs indirects (Figure 3.6).

- Un *point de terminaison* (TP) est un objet qui termine une entité de transport. Les attributs des TP déterminent s'ils sont compatibles entre eux, c.-à-d. s'il est possible d'établir

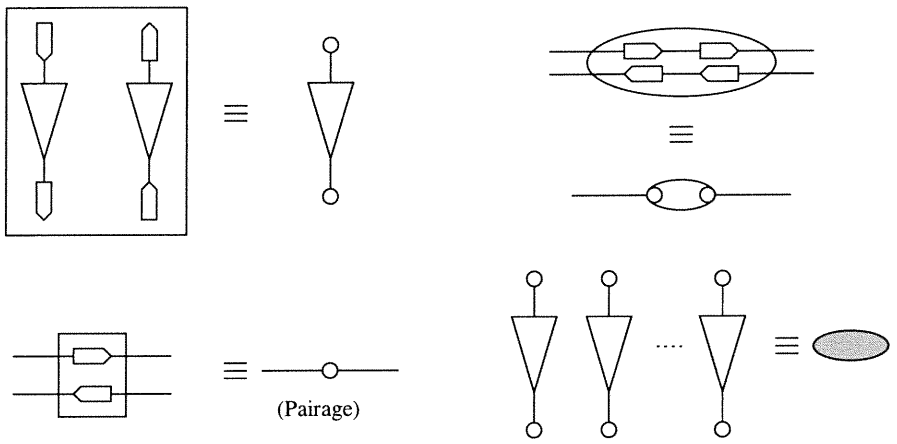


FIG. 3.5 – Équivalences de notation

une connexion entre eux. Les points de terminaisons possèdent de plus des "pointeurs de connectivité" qui indiquent, en pointant vers d'autres TP, d'où les informations caractéristiques viennent et vers où elles se dirigent. Un TP bidirectionnel va comporter quatre pointeurs, soit la source et la destination dans la première direction, et la destination et la source dans le seconde.

- Un *point de terminaison de connexion* (CTP) est un TP qui termine une connexion sur un lien. Un CTP peut être connecté vers le client à un autre CTP ou à un TTP, et peut être connecté vers le serveur à un TTP.
- Un *point de terminaison de voie* (TTP) est un TP qui termine une voie; dans un réseau en couches, un TTP est représenté par un point d'accès, lequel délimite autant la relation client/serveur que la voie. Un TTP peut être connecté vers le client à un ou plusieurs CTPs et vers le serveur aussi à un ou plusieurs (autres) CTPs.

Ce qu'englobent les points de terminaisons est illustré à la Figure 3.6.

La recommandation [ITUg774] introduit également le concept d'*adaptateur indirect* (IA). Un adaptateur indirect est un mécanisme qui permet d'introduire un niveau hiérarchique supplémentaire pour indiquer le multiplexage au niveau des entités de groupes (AUG et TUG) de SDH. Ce genre de multiplexage, en effet, peut se passer d'une fonction d'adaptation propre, puisque aucune facilité de surveillance supplémentaire n'existe pour justifier sa présence. Un adapteur indirect est donc un raffinement d'une fonction d'adaptation. La figure 3.7 illustre des adapteurs indirects dans une couche HOP pour une certaine configurations de TUG qui

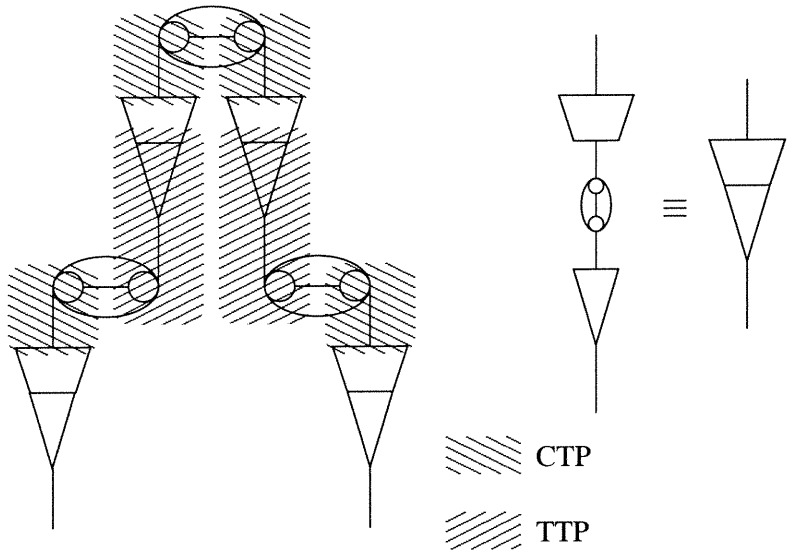


FIG. 3.6 – Points de terminaison

permettent le multiplexage de différents TU. Dans cette figure, les IA sont représentés par les lignes horizontales.

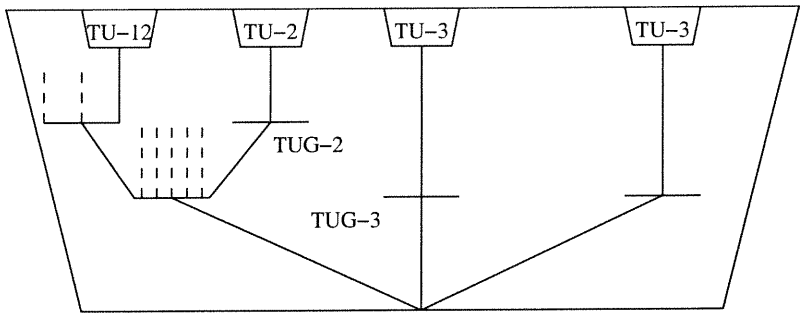


FIG. 3.7 – Adapteurs indirects

3.2 Représentation de SDH avec le modèle fonctionnel

Le modèle expliqué à la section 3.1 est purement abstrait et peut servir à représenter un réseau de transport de façon générique, peu importe la technonologie sous-jacente, qu'il s'agisse de PDH, SDH, ATM (Asynchronous Transfer Mode) [ITUi326], etc. Notons que le modèle représente un réseau à un instant donné ; une reconfiguration du réseau risque de changer les connexions et

même les points de connexions disponibles.

La norme [ITUg803] a pour but d'appliquer le modèle de [ITUg805] au cas spécifique d'un réseau SDH. Pour y parvenir, il emploie et combine les blocs fonctionnels du SDH spécifiés dans [ITUg783], [ITUg783c]. En guise d'introduction à cette section, la Figure 3.8 montre comment on modéliserait un *port de terminaison physique* (PTP, *Physical Termination Point*) STM-4 pour y permettre la commutation de conteneurs VC-12.

Le premier réseau est celui de la section optique. La section optique représente l'onde physique qui sert à la transmission des données. Cette section nous intéresse peu et nous la considérons abstraitement. Il peut exister des configurations plus complexes, où plusieurs signaux optiques servent au transport dans la même fibre. En outre, c'est ainsi que procèdent les systèmes DWDM (Dense Wave Division Multiplexing).

La fonction de terminaison de voie de cette couche a pour rôle d'émettre et de recevoir le signal optique. La fonction d'adaptation recouvre l'alignement et repère les trames STM-n du signal.

Le réseau suivant est celui de la section de régénération. La section de régénération représente les liens point à point entre deux générateurs ou régénérateurs de signal. Les données calculées à ce niveau sont celles de la partie RSOH de la trame SDH.

La fonction de terminaison de voie de cette couche a pour rôle de générer ou de lire le RSOH et de calculer les informations de contrôle d'erreur (BER, Bit Error Rate) ; de plus, elle a pour rôle de brouiller ou de débrouiller le signal (le signal SDH est brouillé selon une fonction simple pour éviter un trop faible nombre de transitions dans le signal optique). La fonction d'adaptation a pour rôle d'effectuer le multiplexage et le démultiplexage de l'information caractéristique de la couche MS-n dans la trame STM-n.

Le réseau suivant est celui de la section de multiplexage. La section de multiplexage représente les liens point à point entre deux générateurs de signal. Les données calculées à ce niveau sont celles de la partie MSOH de la trame SDH.

La fonction de terminaison de voie de cette couche a pour rôle de générer ou de lire le MSOH et de calculer les informations de contrôle d'erreur (BER) ; la fonction doit générer ou lire les signaux indiquant des problèmes (Alarm Indication Signal et Remote Defect Indication).

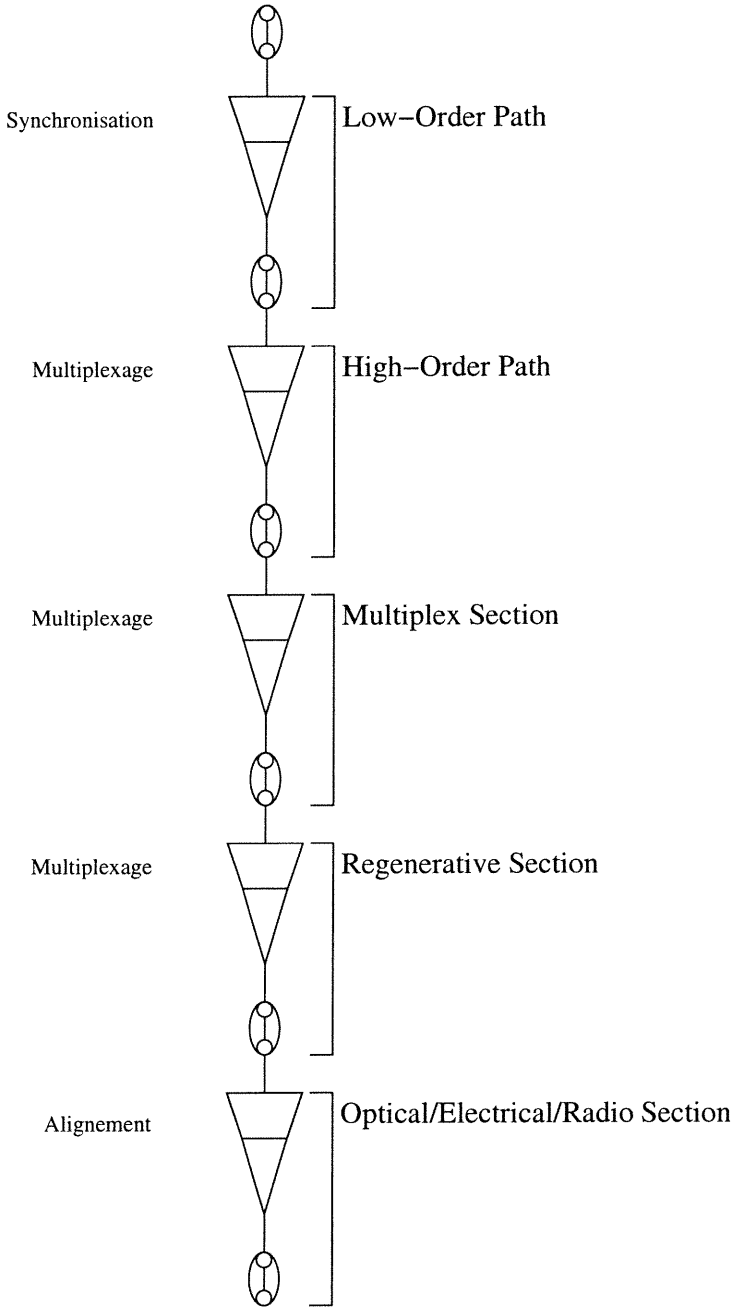


FIG. 3.8 – Réseaux en couches présents dans SDH selon G.803

La fonction d'adaptation de cette couche effectue l'alignement des données nécessaire pour tenir compte du décalage des HOVC dans les AU ; ce décalage est causé soit par la transition entre les systèmes SDH et PDH, soit par des variations intrinsèques au système SDH. C'est donc ici que les HOVC sont assemblés en ou désassemblés à partir d'AU et ensuite (à l'aide d'une adaptation indirecte) en AUG. Ceux-ci sont multiplexés ou démultiplexés dans la charge utile d'un module STM-N.

Le réseau suivant est celui du chemin de haut niveau ; ce chemin est celui qu'empruntent les HOVC en traversant le réseau SDH.

Notons que si la commutation se fait à ce niveau dans un sous-réseau, il n'y aura pas de fonction de terminaison de voie supérieure, car il n'y a pas de voie à terminer ; c'est donc possiblement la dernière couche du sous-réseau. C'est aussi le cas si le HOVC est non équipé.

Autrement, la fonction de terminaison de voie de cette couche a pour rôle de générer ou de lire le POH et de calculer les informations de contrôle d'erreur (BER) associées au chemin.

La fonction d'adaptation joue un rôle similaire à celle au niveau HOP : elle effectue l'alignement des données, mais nécessaire cette fois pour compenser le décalage des LOVC dans les TU. C'est donc à cet endroit que les LOVC sont assemblés en ou désassemblés à partir de TU et ensuite (à l'aide d'une adaptation indirecte) en TUG. Ceux-ci sont multiplexés ou démultiplexés dans la charge utile d'un HOVC (soit le HOC).

Le réseau suivant est celui du chemin de bas niveau ; ce chemin est celui qu'empruntent les LOVC de bout en bout du réseau SDH. Cette couche constitue la dernière étape spécifique à SDH.

La fonction de terminaison de voie de cette couche a le même rôle que celle de la couche précédente. La fonction d'adaptation, quant à elle, a pour rôle d'organiser les données provenant d'un tributaire PDH dans un conteneur SDH, et vice-versa. Cette adaptation peut impliquer la justification des bits et la synchronisation du signal (par bit ou par octet).

Notons que pour une configuration où les HOVC sont, par exemple, des VC-3, et les LOVC des VC-11, il y aurait trois CTP au niveau HOP (High Order Path) ; chaque CTP serait connecté vers le haut à un TTP ; chacun de ceux-ci serait connecté à 21 CTP du niveau LOP (Low Order Path), pour un total de 63 CTP au niveau LOP. Chacun des 63 CTP est connecté à un autre CTP du même sous-réseau pour en permettre la commutation (Figure 3.9).

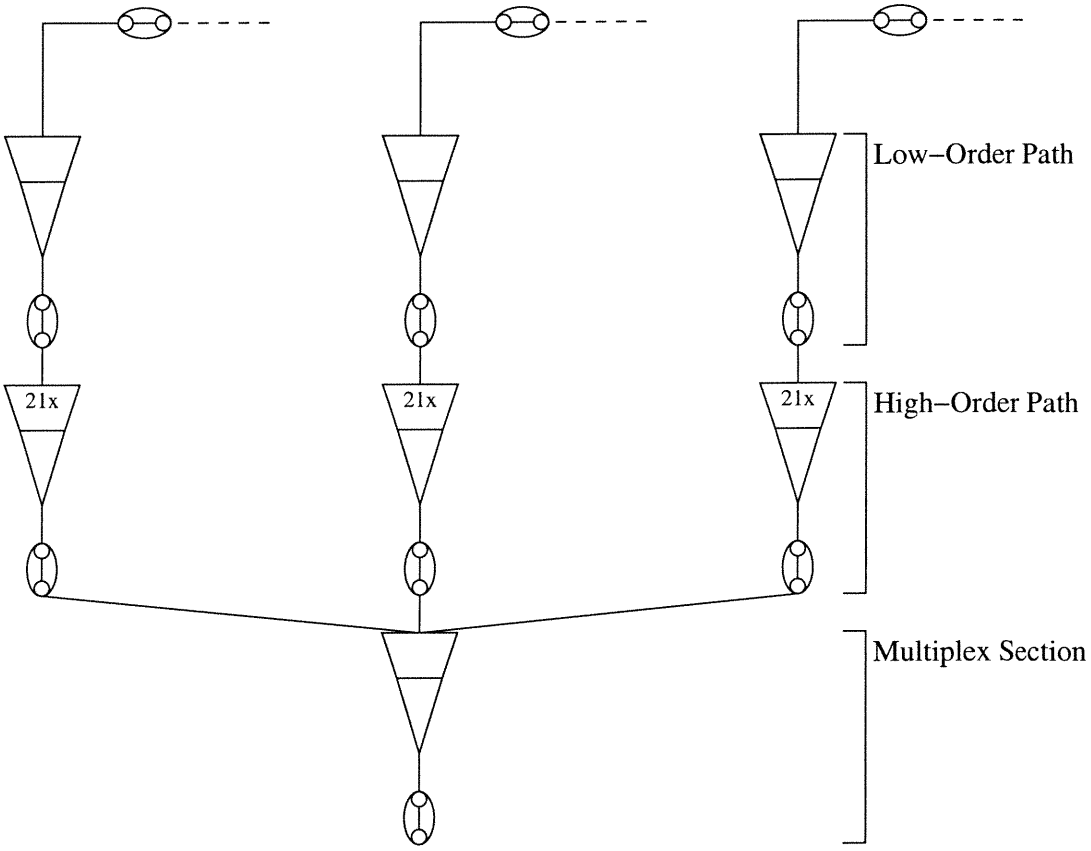


FIG. 3.9 – Exemple de multiplexage des HOP et LOP

Deux genres de systèmes de commutation sont généralement reconnus : l'ADM (Add-Drop Multiplexer) et le DXC (Digital Cross-Connect). Le plus général de ceux-ci est le DXC ; dans ce système, l'élément réseau qui sert de commutateur est représenté comme un sous-réseau ; les PTP de ce sous-réseau sont tous semblables (autrement dit, il n'y a pas de hiérarchie de transport prédéfinie), et la commutation est réalisée aux niveaux HOP et LOP.

Un système plus spécialisé est celui d'ADM ; dans celui-là, deux genres de PTPs existent : les tributaires et les agrégats. On ne peut pas (sauf exception) réaliser une connexion directement entre deux agrégats ; les tributaires servent exclusivement à extraire ou à insérer des données dans les agrégats (qui sont de plus grand débit) ; ces appareils sont effectivement des concentrateurs de données.

Par exemple, soit un ADM simple dont l'agrégat est un STM-4 et qui possède quatre tributaires STM-1. Si on réalise la commutation au niveau HOP, ce système aurait besoin de quatre tributaires pour charger ou décharger entièrement la ligne STM-4.

3.3 Concepts de protection

L'objectif de la protection est d'assurer une certaine redondance dans le réseau et ainsi d'augmenter la probabilité d'un acheminement des informations en cas de panne des équipements de support.

Divers types de protection existent :

- Protection d'une connexion de sous-réseau (SubNetwork Connection Protection) : il s'agit d'offrir de la redondance pour une connexion au travers d'un sous-réseau.
- Protection d'équipement (Equipment Protection) : ce genre de protection affecte l'équipement physique et ne peut pas toujours être modélisé dans le réseau de transport. Il s'agit de faire en sorte que l'équipement qui sert au fonctionnement des éléments réseau (par exemple, cartes de contrôle, générateurs électriques, etc.) soit redondant.
- Protection de la section de multiplexage (MSP, Multiplex Section Protection) : il s'agit de la protection d'une voie au niveau MS en fournissant des connexions redondantes pour la supporter.

Nous nous intéressons uniquement dans ce travail à la protection MSP. Dans ce cas, les

événements qui entraînent la commutation de la protection sont détectés au niveau MSP. De tels événements incluent un BER (Bit-Error Rate) trop élevé, une perte de signal ou un signal incorrect.

3.3.1 Topologies de protection MSP

Différents types de topologies de protection existent, pour satisfaire à divers besoins et/ou critères de rendement économique. Ce sont les systèmes linéaires et les systèmes à anneaux (décrits dans [ITUg841]).

Le système le plus simple est le système de protection linéaire MSP 1+1 (Figure 3.10). Dans ce système, deux sections multiplexes sont employées : la section principale transporte les données en temps normal, tandis que la section en attente est prête à prendre le relais en cas de problème sur la section principale. Cette solution est donc deux fois plus coûteuse qu'un système non protégé.

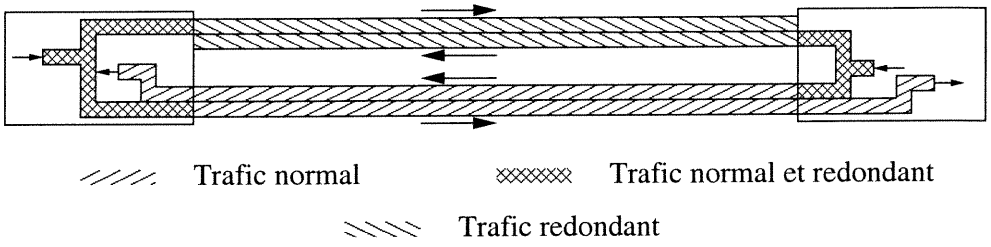


FIG. 3.10 – Protection MSP 1+1, état normal

En cas de coupure de fibres (représentée par × dans les figures suivantes), le trafic normal peut être coupé ; ce sera alors le trafic redondant, transporté par la section en attente, qui sera utilisé (Figure 3.11).

Un système légèrement plus complexe est le système de protection linéaire MSP 1:N (Figure 3.12. Dans ce système, N canaux servent à transporter des données ; un canal est réservé pour le cas où un de ces N canaux tomberait en panne. De plus, ce canal peut transporter du trafic "supplémentaire". Ce trafic supplémentaire n'est pas protégé.

En cas de panne de l'un des N canaux, le trafic supplémentaire est abandonné au profit du trafic du canal tombé en panne (Figure 3.13).

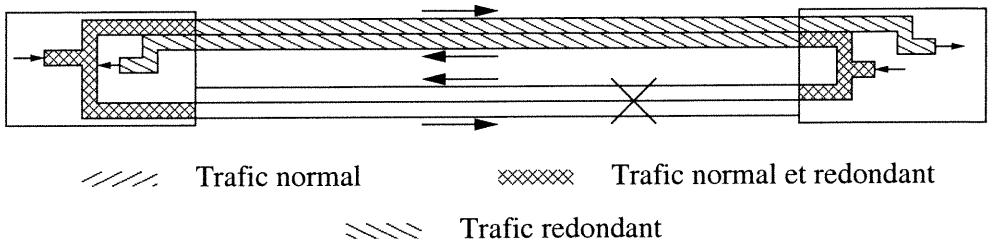


FIG. 3.11 – Protection MSP 1+1, commutation en cours

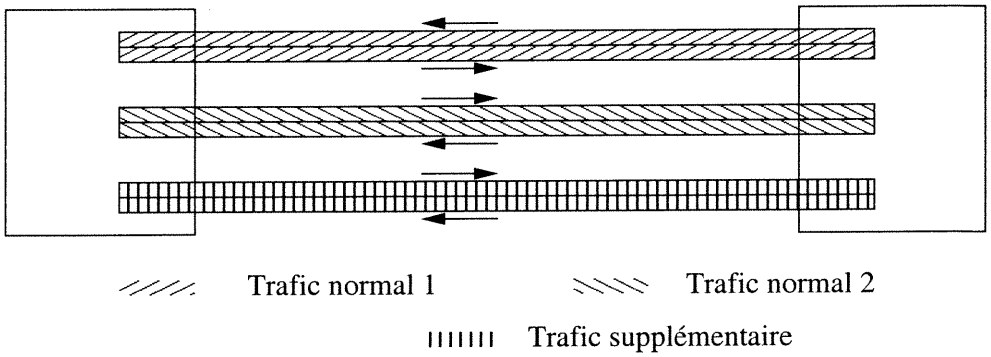


FIG. 3.12 – Protection MSP 1:N (avec N de 2), état normal

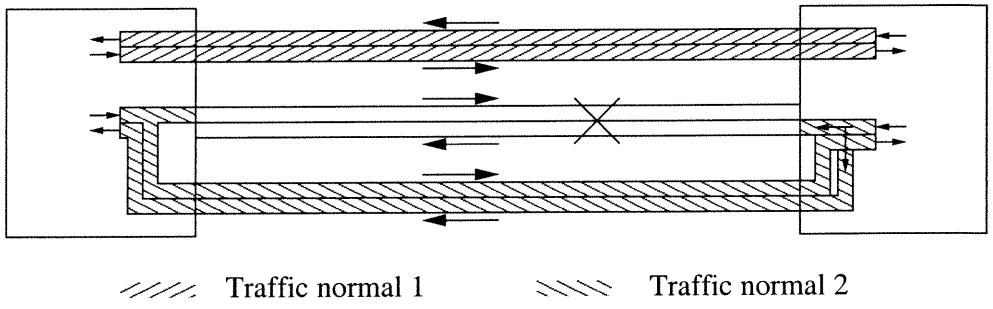


FIG. 3.13 – Protection MSP 1:N (avec N de 2), commutation en cours

Un autre type de topologie consiste à utiliser un système à anneaux ; ce genre de système est appelé Multiplex Section Shared Protection Ring ou MSSPRING. Dans un système à anneaux, la moitié des unités administratives sont affectées au trafic normal ; l'autre moitié est soit en attente, soit affectée à du trafic supplémentaire. En pratique, deux types d'anneaux existent : les anneaux à deux fibres et les anneaux à quatre fibres. Le genre d'événement qui peut survenir dépend du type d'anneau.

Notons que lorsqu'on fait transiter du trafic par un anneau MSSPRING, on peut choisir les canaux particuliers à utiliser de la source à la destination. Ces canaux ne seront utilisés que sur les segments de l'anneau entre les deux endroits (donc, on peut réutiliser les mêmes canaux sur des segments non chevauchants de l'anneau).

Le plus simple système de protection à anneaux comporte deux fibres. Le trafic circulant sur une fibre va dans la direction opposée à celui circulant sur l'autre. Le trafic transporté par chaque fibre est divisé en canaux. Chaque AUG constitue un canal ; ainsi, une fibre STM-4 possède 4 canaux. La moitié des canaux sont utilisés pour le trafic normal ; les canaux de l'autre moitié servent à la protection (Figure 3.14).

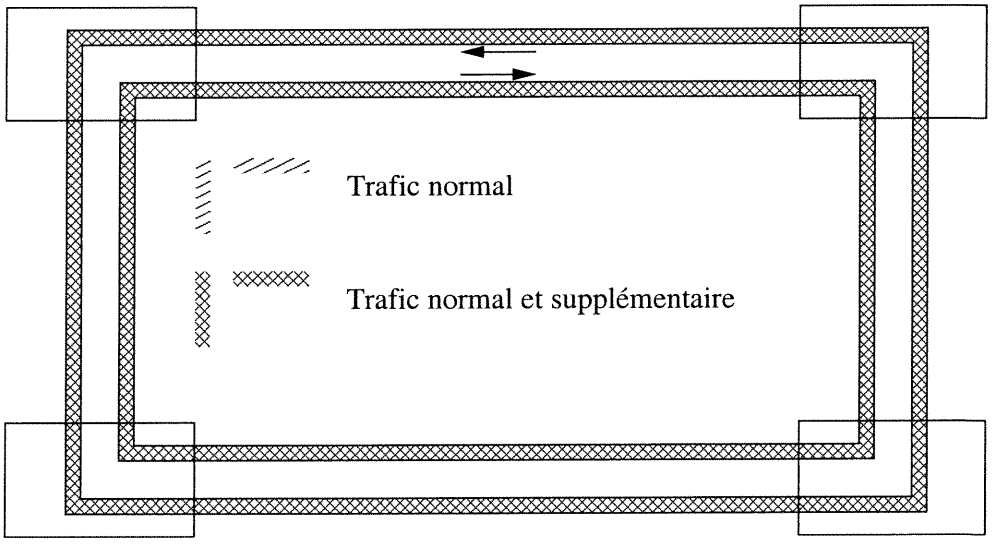


FIG. 3.14 – Protection MSSPRING à deux fibres, état normal

Lorsqu'il y a une coupure dans un anneau à deux fibres (Figure 3.15), un événement de commutation d'anneau survient ; le trafic sur les canaux normaux est redirigé vers les canaux de protection de l'autre fibre. Cela permet de faire une nouvelle boucle. Peu importe où survient

la coupure, tous les segments vont maintenant faire circuler, sur leurs canaux protégés, le trafic redirigé à l'endroit de la coupure. C'est pour cette raison qu'il s'agit de protection partagée (Shared Protection) : les canaux de protection servent de façon partagée à n'importe quel élément qui doit y rediriger du trafic.

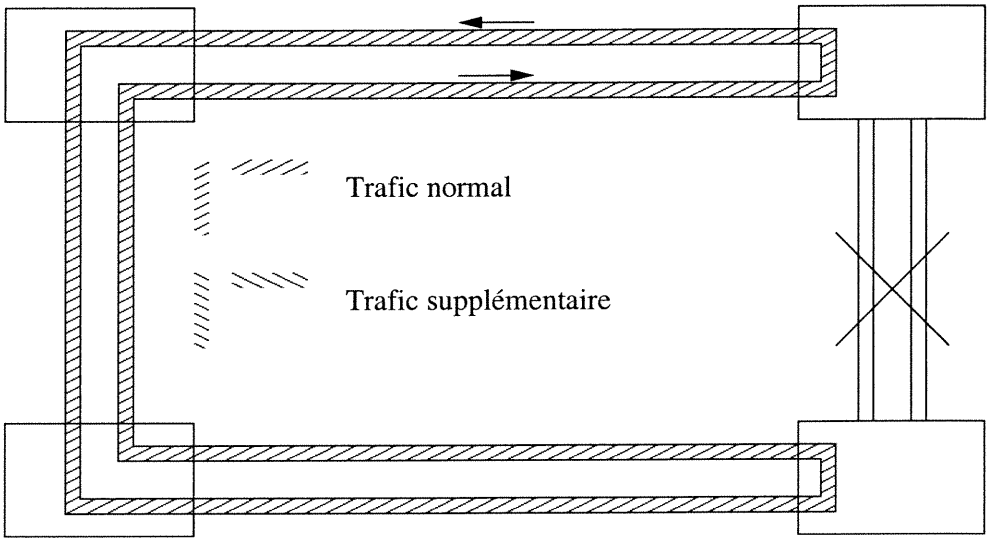


FIG. 3.15 – Protection MSSPRING à deux fibres, commutation d'anneau

Un système plus compliqué, mais plus fiable, est un anneau à quatre fibres. Dans ce genre d'anneau, deux fibres allant en sens inverse l'une de l'autre transportent le trafic normal, et les deux autres fibres (allant également en sens inverse l'une de l'autre) transportent le trafic supplémentaire (Figure 3.16). Comme sur un anneau à deux fibres, le trafic est séparé en canaux. Deux événements peuvent survenir : une commutation de segment et une commutation d'anneau.

Une commutation de segment survient lorsque la paire de fibres transportant le trafic normal devient inopérante. Lorsque cette condition est détectée, le trafic est redirigé vers la paire de fibres de protection, comme l'illustre la Figure 3.17.

Ce genre de commutation peut survenir de façon indépendante pour chaque segment. Toutefois, en cas de problème affectant les deux paires de fibres sur le même segment, une commutation d'anneau est effectuée. Le principe est le même que lorsqu'il n'y a que deux fibres et est illustré à la figure 3.18. Une commutation d'anneau est plus prioritaire qu'une commutation de segment et annule ces dernières, quitte à ce que la connectivité soit imparfaite.

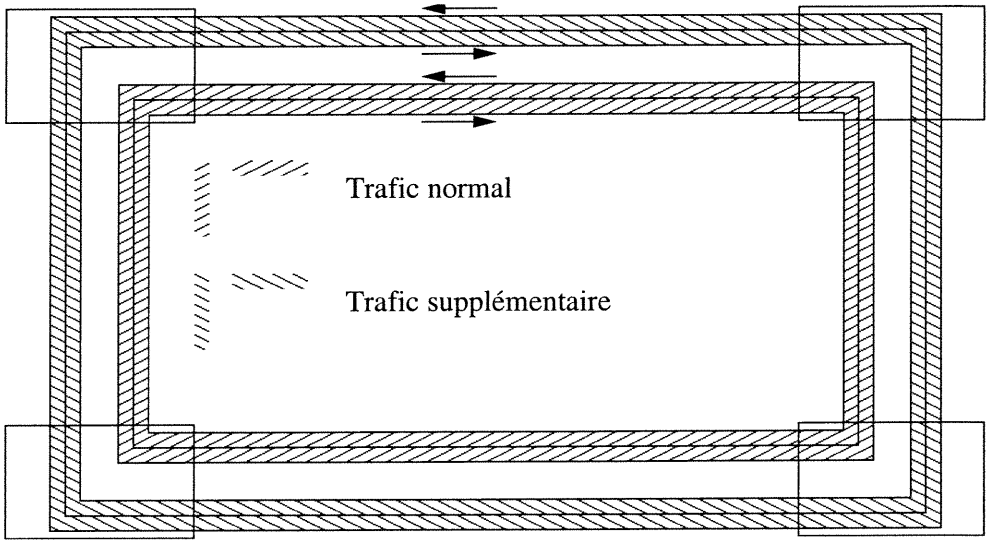


FIG. 3.16 – Protection MSSPRING à quatre fibres, état normal

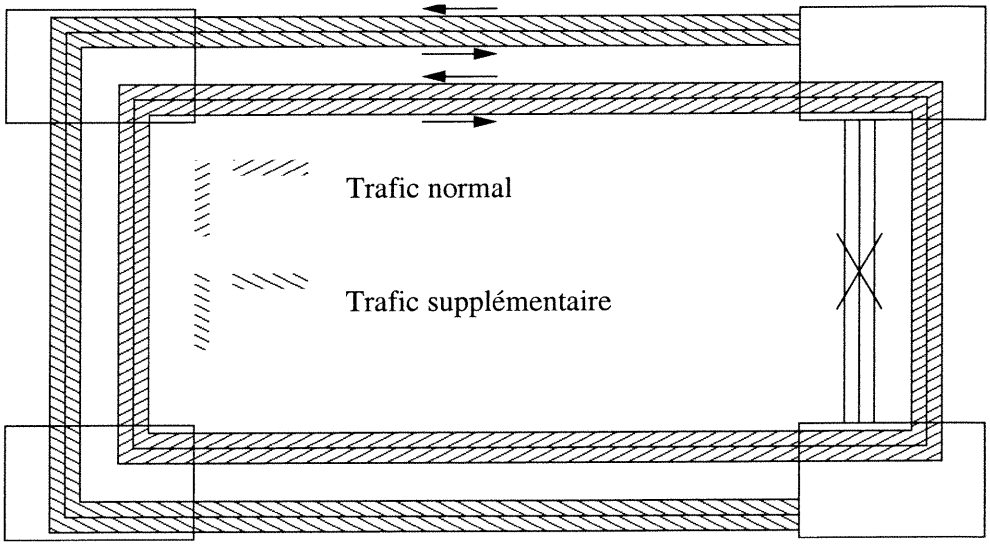


FIG. 3.17 – Protection MSSPRING à quatre fibres, commutation de segment

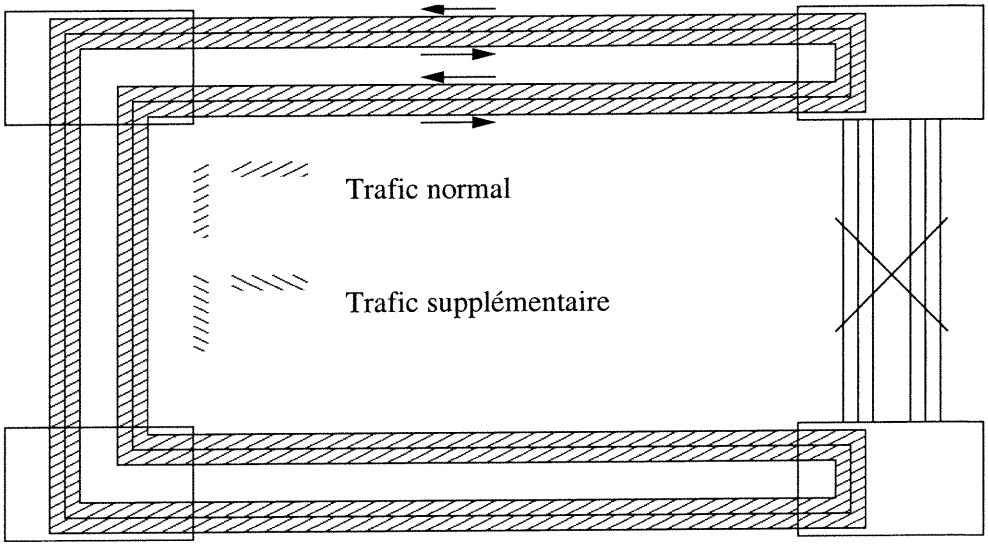


FIG. 3.18 – Protection MSSPRING à quatre fibres, commutation d'anneau

3.3.2 Réversibilité de la protection

Normalement, les désignations des sections (normale et de protection) sont fixes. Cependant, un groupe de protection peut être non réversible. Cela signifie que les sections normales et servant à la protection sont interchangées lorsqu'il y a une commutation de protection.

Autrement dit, dans une configuration non réversible, un retour à la normale de la section régulière n'entraînera pas son utilisation tant que la section de protection ne sera pas hors d'état elle-même. Le trafic supplémentaire est donc abandonné tant qu'une nouvelle commutation n'a pas lieu et que le trafic normal revient à son endroit habituel. Pour cette raison, ce genre de configuration est rarement utilisé avec du trafic supplémentaire.

3.3.3 Mode de commutation

Le mode de commutation d'un groupe de protection est soit unidirectionnel, soit bidirectionnel. Un système unidirectionnel peut effectuer des commutations dans une seule direction à la fois. Ainsi, si une coupure de section ne touche qu'une des deux sections, le trafic sera redirigé vers la section de protection seulement dans la direction affectée.

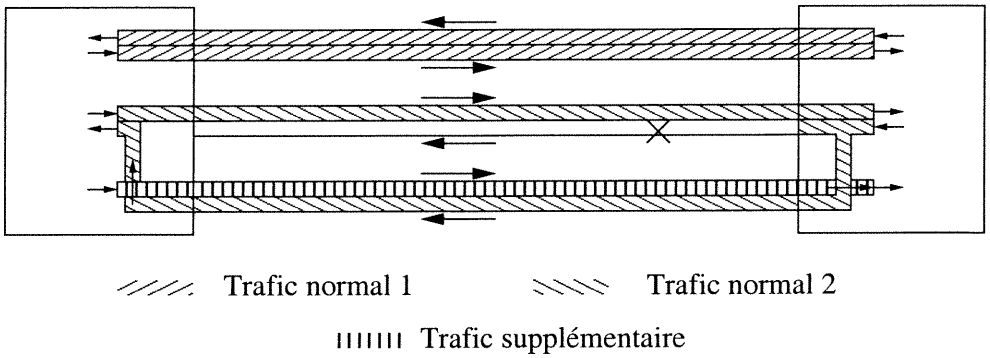


FIG. 3.19 – Commutation unidirectionnelle dans un système MSP 1:N (avec N de 2)

3.4 Modélisation de la protection du niveau MS

Afin de présenter le système MSP dans le cadre du modèle de la norme G.803, il faut prendre la couche MS et y introduire par expansion une sous-couche s'occupant de la protection (la sous-couche MSP). Pour ce faire, on précise la fonction de terminaison de piste pour y incorporer les nouveaux éléments, comme l'indique la Figure 3.20.

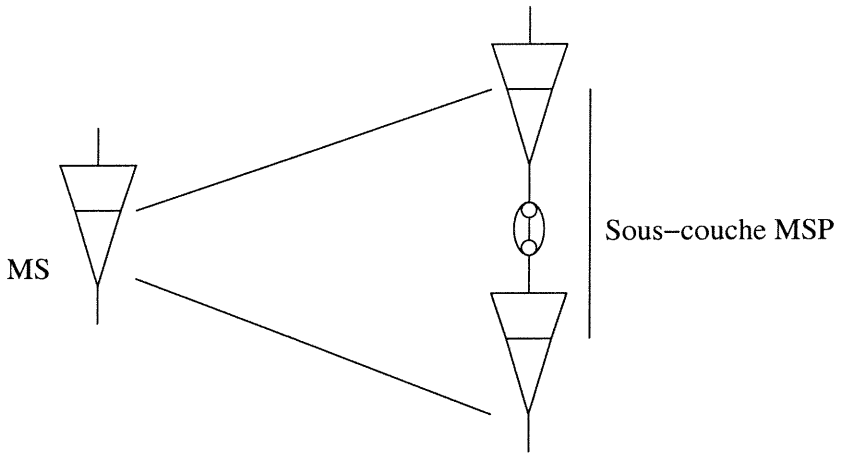


FIG. 3.20 – Sous-couche MSP

Par la suite, on introduit un sélecteur au niveau du TCP de la couche MSP pour choisir entre deux fonctions d'adaptation MSP laquelle sera utilisée pour recevoir et/ou transmettre l'information caractéristique (Figure 3.21).

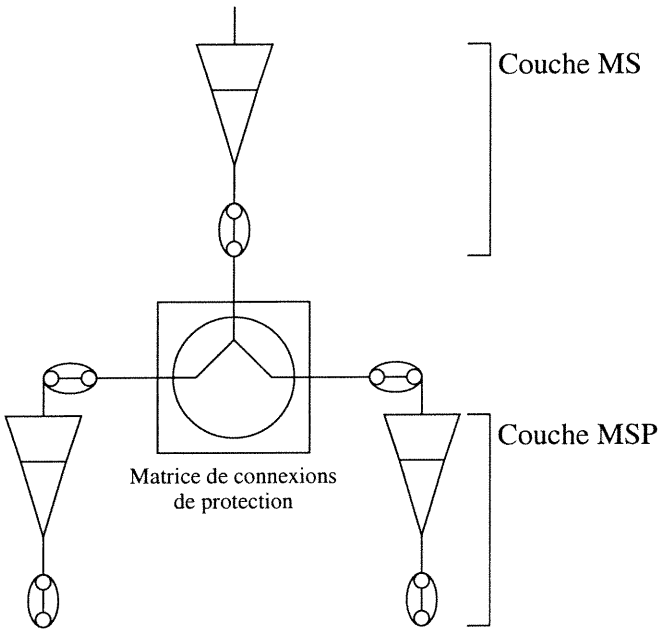


FIG. 3.21 – Sélecteur de sous-couche MSP

3.5 Résumé

Dans ce chapitre, nous avons décrit le modèle fonctionnel des réseaux de transport normalisé dans G.805. Ce modèle permet de décrire une variété de réseaux de transports en utilisant les principes de partition (pour créer des sous-réseaux) et de stratification (pour chaque réseau en couches).

Nous avons vu comment modéliser un port physique typique d'un système SDH en fonction de ce modèle : les différentes sections et chemins d'un système SDH sont représentés par des réseaux : OS, RS, MS, HOP et LOP.

Enfin, nous avons vu les concepts de protection : les différents types de protection (MSP 1+1, MSP 1:1, MSSPRING à deux et quatre fibres), les propriétés d'un système de protection (réversibilité et mode de commutation) et la modélisation de la protection dans le modèle fonctionnel grâce à une couche MSP sous la couche de MS.

Dans le prochain chapitre, nous voyons comment les réseaux de transport et particulièrement les systèmes de protection sont modélisés par la norme MTNM 2.0.

Chapitre 4

La norme MTNM 2.0

Le TMF (TeleManagement Forum) est un regroupement industriel visant à développer des normes de gestion de réseau ; sa publication numéro 814 est intitulée Multi-Technology Network Management 2.0, ou MTNM 2.0 [MTNM2]. La version précédente, MTNM 1.0 [MTNM1], était la publication TMF 509.

Pour comprendre la portée de la norme TMF 814, il faut d'abord expliquer ce qu'est un TMN.

4.1 Le système TMN

La norme MTNM 2.0 est une interface définie pour la gestion d'un réseau de gestion des télécommunications (Telecommunication Management Network, TMN). Un TMN est constitué de tous les équipements servant à la gestion d'un réseau de transport de communications.

La norme M.3010 [ITUm3010] définit un TMN comme étant une architecture de gestion, incluant la planification, l'approvisionnement, l'installation, l'entretien, l'opération et l'administration d'équipements, de réseaux et de services de télécommunications.

L'objectif du modèle TMN est d'uniformiser la gestion d'un réseau, malgré la diversité des configurations possibles et des équipements employés, en introduisant des modèles de gestion génériques et des interfaces normalisées. Les modèles permettent de séparer les différents pro-

cessus de gestion afin de distribuer adéquatement les rôles aux différents systèmes de gestion à l'intérieur et entre les entreprises.

Le modèle TMN fait appel à des blocs fonctionnels qui permettent de distinguer les différents rôles de ses composantes. M.3010 définit quatre blocs fonctionnels :

- Un NEF (*Network Element Function*) représente un élément de réseau (NE) aux fins de sa gestion ; il s'agit de la composante du NE qui se trouve dans un TMN et l'y représente.
- Un OSF (*Operations Systems Function*) est un système de traitement des informations ayant trait à la gestion des télécommunications dans le but de surveiller, contrôler et coordonner les fonctions de télécommunication ou même les fonctions de gestion des télécommunications en elles-mêmes.
- Un WSF (*WorkStation Function*) symbolise un système qui fournit une interface de gestion pour un utilisateur.
- Un TF (*Transformation Function*) représente un système qui permet de connecter deux autres entités qui n'ont pas de mécanisme de communication commun. Au moins l'un de ces systèmes doit être un bloc fonctionnel à l'intérieur du TMN. L'autre système est soit (i) un bloc fonctionnel à l'intérieur du TMN, (ii) un bloc fonctionnel à l'intérieur d'un autre TMN, ou (iii) une entité extérieure à un TMN.

Tout comme dans [ITUg805], les blocs fonctionnels du TMN communiquent entre eux via des interfaces construites sur des points de référence. (Une interface a pour désignation une lettre majuscule qui est la même que la lettre minuscule désignant le point de référence où elle se trouve).

Un point de référence est donc l'interaction en termes d'entrées et de sorties entre deux blocs fonctionnels. Cinq classes de points de référence sont définies ; les trois premières sont à l'intérieur du TMN ; les deux dernières à l'extérieur.

f Classe entre un OSF et un WSF

q Classe entre un OSF, un TF et un NEF

x Classe entre deux TMNs ou entre l'OSF d'un TMN et son équivalent dans un autre réseau

g Classe entre un WSF et son utilisateur

m Classe entre un TF et une entité gérée à l'extérieur d'un TMN

Les interfaces de communication sur un point de références sont notées par la même lettre

que leur classe mais en majuscule.

Afin de simplifier la modélisation, les fonctions de gestion ont été séparées en plusieurs couches selon le modèle LLA (Logical Layered Architecture). Ce modèle spécifie cinq couches de gestion (Figure 4.1).

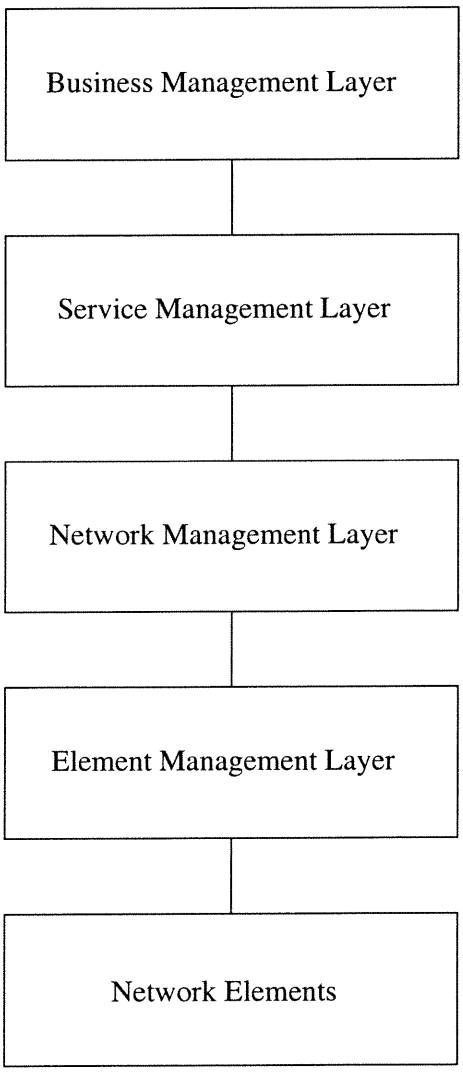


FIG. 4.1 – TMN Logical Layered Architecture

La couche la plus basse est celle des *éléments de réseau* (NE, , Network Element). Ces éléments sont en général de l'équipement de télécommunication, ou des éléments qui supportent un réseau de télécommunication. Fonctionnellement, la composante du NE qui se trouve dans

un TMN est en fait le NEF.

Ces éléments sont gérés par la couche supérieure, la couche de gestion des éléments (EML, Element Management Layer). Les éléments sont gérés en groupe ou de façon individuelle ; trois rôles sont accordés à cette couche : (i) le contrôle et la coordination des NEs dans leurs fonctions individuelles, (ii) le contrôle et la coordination des NEs dans leurs fonctions atteintes collectivement, et (iii) la maintenance de statistiques, journaux et autres données concernant les NEs. Un système de gestion de réseau de ce niveau est appelé EMS (Element Management System).

L'EML est composé de plusieurs blocs E-OSF. Chacun de ces blocs pourrait représenter un système gérant quelques-uns des NEs du réseau, par exemple. Voici des exemples de fonctions remplies par l'EML ; pour (i), coordonner le chargement d'une nouvelle version du logiciel de contrôle d'un NE ; pour (ii), créer une configuration d'anneau entre trois NEs reliés ensemble ; pour (iii), centraliser un journal d'accès aux systèmes.

La couche supérieure est la couche de gestion réseau (NML, Network Management Layer). Ce niveau est composé de blocs N-OSF ; son objectif est la gestion du réseau en son entier ; c'est là que s'exécute l'entretien, le contrôle et la coordination du réseau complet, peu importe son étendue. C'est à cet endroit que la configuration du réseau sera adaptée pour répondre aux besoins des clients. Un système de gestion de réseau implanté à ce niveau est appelé NMS (Network Management System).

D'une certaine façon, la seule différence entre la couche EML et la couche NML est l'échelle de gestion ; un EML gère quelques éléments, tandis qu'un NML gère, au travers de l'EML, tous les éléments du réseau.

4.2 Situation de TMF 814

La norme TMF 814 est défini comme étant une interface entre les couches EML et NML du Telecommunications Management Network.

Le service rendu par une implémentation de cette norme est effectué par un bus de communication CORBA [OMG1998] (Common Object Request Broker Architecture).

C'est donc une interface rendue sur un point de référence q, et par conséquent une interface

de type Q.

4.3 Principes généraux de fonctionnement

Le principe général derrière tout système basé sur CORBA est de fournir des objets d'une façon similaire, peu importe la localisation du fournisseur d'objet par rapport à leurs utilisateurs. Ainsi, les objets pourront, si nécessaire, transiter sur un réseau de communication (dans ce cas-ci, sur le réseau TMN) pour être utilisés par le client.

Le système CORBA fournit également des services de localisation et d'enregistrement pour qu'un client puisse trouver initialement les objets peu importe l'endroit où se trouve le serveur.

Un serveur TMF814 doit publier un ensemble d'objets qui permettent de démarrer et de gérer une session de communication entre un NMS et un EMS. Ces objets possèdent des méthodes pour obtenir les autres objets qui font partie de l'interface. Les autres objets de l'interface sont des "Managers" et permettent d'obtenir des informations de l'EML du type représenté par le manager en question :

EMSMgr_I : Permet d'obtenir des informations sur l'EMS (Element Management System) et des informations générales sur le réseau qu'il gère.

MultiLayerSubnetworkMgr_I : Permet d'obtenir des informations sur les sous-réseaux que reconnaît l'EMS.

ManagedElementMgr_I : Permet d'obtenir des renseignements sur les éléments gérés dans le domaine de l'EMS (normalement, les NEs sous sa responsabilité).

EquipmentInventoryMgr_I : Permet d'obtenir des précisions sur les composantes faisant partie des éléments gérés.

GuiCutThroughMgr_I : Contrôle la redirection d'interfaces de contrôle de plus bas niveau (EMS ou NEs) vers la station de l'utilisateur.

PerformanceManagementMgr_I : Donne des renseignements statistiques sur le trafic qui commute par le réseau.

ProtectionMgr_I : Permet d'obtenir des renseignements sur la configuration de la protection dans le réseau.

TrafficDescriptorMgr_I : Permet d'obtenir des informations concernant les attributs des ports ; par exemple, la largeur de bande disponible ou la qualité de service offerte.

MaintenanceOperationsMgr_I : Permet d'effectuer des opérations de test et d'entretien des éléments du réseau.

Les Managers sont des objets CORBA, c'est-à-dire qu'ils sont l'implantation d'une interface standardisée par la norme MTNM 2.0. L'interface est écrite en IDL, ce qui permet de réaliser l'implantation dans tout langage pour lequel il existe une corrélation entre l'IDL et ce langage, que celui-ci utilise les concepts de programmation orientée objet ou non.

Les Managers n'ont pas d'attributs qui leurs sont propres. Ils exportent uniquement des méthodes qui permettent d'accéder à des structures qui représentent chaque élément du réseau. À leur tour, ces structures ne sont que des agglomérats de données ; ce ne sont pas des objets à part entière, avec des méthodes ou un cycle de vie bien défini ; le fait que les éléments du réseau soient représentés par de simples structures de données constitue un modèle appelé "coarse-grained" ; ce genre de modèle requiert moins de ressources informatiques (mémoire, temps de processeur) que si ces structures étaient des objets de premier ordre (avec une interface complète). Les structures suivantes sont définies :

EMS : Représente l'EMS en soi.

MultiLayerSubnetwork : Représente un sous-réseau d'un réseau en couches.

SubnetworkConnection : Représente une connexion de bout en bout entre deux ports physiques (PTPs) de deux éléments gérés appartenant au même sous-réseau.

TopologicalLink : Représente un lien topologique entre deux points de terminaison d'un réseau.

TrafficDescriptor : Un descripteur de trafic est une collection d'attributs servant à décrire les caractéristiques de bande passante et de qualité de service sur un point de terminaison.

ManagedElement : Représente un élément du réseau géré par l'EMS.

AID : Il n'y a pas d'objet AID. Ce nom spécial représente la source d'une alarme qui n'est pas associée particulièrement à un des autres objets sous l'élément géré.

PTP (Physical Termination Point) : Est un TP qui représente un port physique d'un ManagedElement.

CTP (Connection Termination Point) : Peut correspondre soit à un CTP semblables à ceux définis dans la norme M.3100, ou un aggrégat de CTP et de TTP de cette norme.

TPPool : Est un groupe de CTPs et de PTPs, utilisé pour modéliser l'administration partitive des interfaces ATM.

PGP (Protection Group) : Est un ensemble de PTPs qui assurent un service de protection. Il s'agit de l'élément auquel on s'intéresse le plus dans ce travail.

EquipmentHolder : Représente une ressource physique gérée par l'EMS ; les équipements définis sont les casiers, les chassis, les fentes et les sous-fentes des éléments du réseau. Les EquipmentHolder peuvent contenir des Equipment ou d'autres EquipmentHolder récursivement.

Equipment : Représente une ressource physique finale, telle qu'une carte réseau.

Les structures exportées par les Managers sont organisées de façon hiérarchique : chaque structure possède un nom unique organisé à la manière d'un nom distingable (DN, Distinguished Name) dans un répertoire. Un DN est une suite de noms relatifs (RDN, Relative Distinguished Name) ; chacun de ces noms relatifs possède un type et une valeur. Le premier nom relatif de tout objet est celui du EMS. Les noms suivants varient selon le type d'objet. La figure 4.2 indique la hiérarchie selon le type de structure.

Par exemple, un groupe de protection (structure que nous expliquons ci-dessous) quelconque pourrait avoir le nom suivant, formé de trois noms distingués relatifs :
Type : "EMS", Valeur : "Company/EMSName" ;
Type : "ManagedElement", Valeur : "4234" ;
Type : "PGP", Valeur : "MSP1+1/2.2.3.3.2".

Ainsi, un nom de groupe de protection incorpore toujours le nom du ManagedElement sous lequel il se trouve.

Les types sont standardisés ; les valeurs sont sujettes à des conventions selon leur type.

Dans le cadre de ce travail, c'est le ProtectionMgr_I qui nous intéresse davantage. Le ProtectionMgr_I possède plusieurs méthodes qui sont toutes reliées au traitement des groupes de protection. Les groupes de protection sont décrits dans la section suivante.

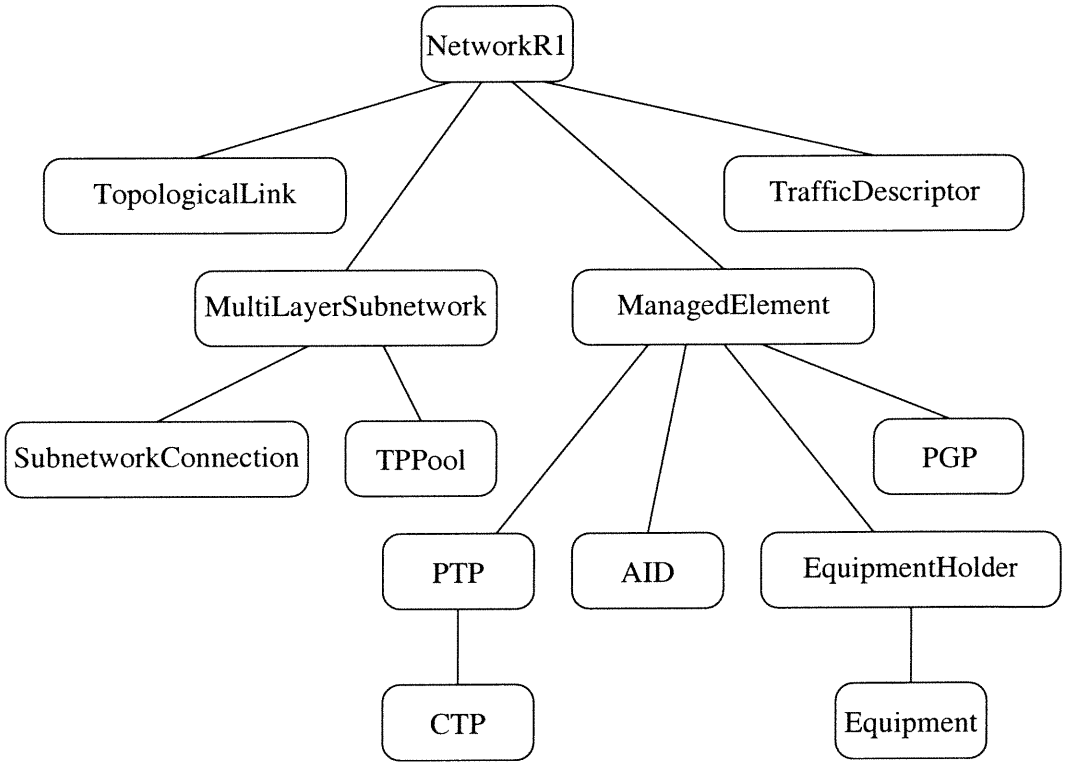


FIG. 4.2 – Hiérarchie des objets

4.3.1 Groupes de protection

Les groupes de protection de MTNM 2.0 sont définis comme les ensembles de ports physiques sur un élément de réseau qui fournissent un des services de protection mentionnés au chapitre précédent.

Plus précisément, il s'agit de l'ensemble des PTPs dont la section MSP est impliquée dans la même structure de protection sur un élément réseau particulier.

Voici un extrait de l'IDL (Interface Description Language) de TMF814 qui décrit la structure de groupe de protection (les déclarations IDL constituent un mécanisme utilisé par CORBA pour permettre l'échange d'information entre les divers intervenants) :

```
struct ProtectionGroup_T
{
  globaldefs::NamingAttributes_T name;
  string userLabel;
  string nativeEMSName;
  string owner;
  ProtectionGroupType_T protectionType;
  ProtectionSchemeState_T protectionSchemeState;
  ReversionMode_T reversionMode;
  transmissionParameters::LayerRate_T rate;
  globaldefs::NamingAttributesList_T pgpTPList;
  globaldefs::NVSList_T pgpParameters;
};
```

L'attribut `name` a déjà été expliqué. Les attributs `userLabel`, `nativeEMSName` et `owner` sont des chaînes de caractères libres. Les chaînes `userLabel` et `owner` sont modifiables par le NMS (Network Management System) et peuvent contenir ce qu'il désire. La chaîne `nativeEMSName` doit normalement porter le nom par lequel l'EMS affiche, dans sa propre interface utilisateur, s'il en a une, le groupe de protection.

Excepté son nom, l'attribut le plus important d'un groupe de protection est son type (protectionType dans l'interface). MTNM 2.0 supporte quatre types de groupes de protection :

`PGT_MSP_ONE_PLUS_ONE` : Il s'agit simplement d'un groupe MSP 1+1. Un tel groupe implique deux PTPs. Ce type sera dorénavant appelé "MSP 1+1".

`PGT_MSP_ONE_FOR_N` : Il s'agit d'un groupe MSP 1:N. Un tel groupe implique $N+1$ PTPs. Ce type sera dorénavant appelé "MSP 1:N. (Ou, dans les cas particuliers, MSP 1:1, par exemple.)

`PGT_2_FIBER_BLSR` : Il s'agit d'un groupe MSSPRING à deux fibres ; BLSR signifie "Bilateral

Line Switched Ring", un terme alternatif pour désigner MSP SPRING. Un tel groupe implique deux PTPs. Ce type sera dorénavant appelé 2F-BLSR.

`PGT_4_FIBER_BLSR` : Il s'agit d'un groupe MSSPRING à quatre fibres. Un tel groupe implique quatre PTPs. Ce type sera dorénavant appelé 4F-BLSR.

Les groupes 4F-BLSR ont ceci de particulier qu'ils comportent quatre PTPs, mais que ces deux PTPs sont regroupés en deux sous-groupes MSP. Ces sous-groupes seront tous deux soit de type MSP 1+1 (s'il n'y a pas de trafic supplémentaire), soit de type MSP 1:1 (s'il y a du trafic supplémentaire). Chacun de ces sous-groupes représente deux PTPs qui sont sur le même côté (nommés est et ouest) de l'anneau.

L'attribut `protectionSchemeState` a un domaine de trois valeurs acceptables. La valeur normale est `PSS_AUTOMATIC`; cela indique que le mécanisme de commutation automatique est actif; autrement, la valeur peut être `PSS_FROZEN` pour indiquer l'état contraire (cela peut arriver si le mécanisme a été inhibé manuellement); la valeur peut enfin être `PSS_NA` si l'état actuel ne peut être déterminé ou s'il ne correspond pas clairement à un des deux autres états (par exemple, si le mécanisme de commutation automatique n'est inhibé que d'un côté dans un groupe BLSR à quatre fibres).

L'attribut `reversionMode` a une des trois valeurs `"RM_REVERTIVE"`, `"RM_NONREVERTIVE"` ou `"RM_UNKNOWN"`. L'attribut indique si, à la suite d'une commutation de protection, le système va revenir à la configuration originale une fois que les conditions ayant mené à l'état protégé sont revenues à la normale.

L'attribut `rate` indique quel genre de trafic est transporté par le groupe de protection. Une longue série de valeurs est disponible pour indiquer ce qui fait l'objet de la protection.

L'attribut `pgpTPList` contient une liste des noms de PTPs qui font partie du groupe de protection. Les PTPs sont nécessairement situés sur le même ManagedElement que le ProtectionGroup en soi.

Enfin, `pgpParameters` contient une liste de paires nom–valeur pour des attributs supplémentaires. Huit paramètres sont définis :

- Le paramètre `SwitchMode` peut avoir deux valeurs, `"SingleEnded"` et `"DualEnded"`. L'attribut indique si la protection est appliquée symétriquement.

- Le paramètre `SPRINGProtocol` a une des deux valeurs, `"Standard"` ou `"TransOceanic"`, selon la méthode de commutation utilisée par le groupe. Ce paramètre n'est défini que dans le cas des groupes BLSR.
- Le paramètre `SPRINGNodeId` est une chaîne contenant un nombre de 0 à 15 ou la valeur "Unknown" et représente le numéro du noeud aux fins de son identification dans le système de protection automatique ; cet identificateur est utilisé par le protocole employé sur les octets K de la trame SDH.
- Le paramètre `nonPreEmptibleTraffic` indique s'il existe une classe de trafic supplémentaire qui peut être maintenue malgré le fait que le maintien optimal du trafic normal requerrait une commutation qui causerait une perte de ce trafic supplémentaire.
- Le paramètre `wtrTime` indique le temps d'attente en secondes avant de revenir d'un état où la protection est active à un état non commuté. Si la valeur n'est pas disponible, la valeur de l'attribut devra être -1. Cet attribut doit nécessairement contenir -1 si le mode de commutation n'est pas réversible ou pas fourni.
- Le paramètre `HoldOffTime` indique combien de temps un problème doit durer avant qu'il puisse causer une commutation. La valeur est en millisecondes (peut aussi être "Infinite" ou "Unknown").
- Les paramètres `LODNumSwitches` `LODDuration` indiquent que si un groupe de protection subit un certain nombre de commutations à l'intérieur d'un temps déterminé, le groupe sera verrouillé (ce qui empêchera toute commutation tant que le verouillage persistera).

Comme on le voit, cette structure n'indique pas dans quel état se trouve un groupe de protection : on ne peut y découvrir si la commutation de protection est active ou non, et le cas échéant, pour quelle raison.

Le rôle de rapporter l'état actuel est rempli par une deuxième structure, SwitchData, que voici en IDL :

```
struct SwitchData_T
{
  ProtectionType_T protectionType;
  SwitchReason_T switchReason;
  transmissionParameters::LayerRate_T layerRate;
  globaldefs::NamingAttributes_T groupName;
  globaldefs::NamingAttributes_T protectedTP;
  globaldefs::NamingAttributes_T switchAwayFromTP;
  globaldefs::NamingAttributes_T switchToTP;
};
```

L'attribut `protectionType` indique si les renseignements ont trait à la protection MSP (pour un groupe de protection) ou bien à la protection SNCP (SubNetwork Connection Protection). Dans le cadre de ce travail, nous n'abordons pas la protection de connexions de sous-réseaux.

Les attributs `layerRate` et `groupName` sont recopiés des attributs `rate` et `name` du ProtectionGroup, respectivement.

L'attribut `switchReason` est plus intéressant : il indique s'il y a une commutation de protection en cours et pour quelle raison. L'attribut peut prendre une des valeurs suivantes :

`SR_NA` : Indique qu'il n'y a pas de commutation de protection en cours.

`SR_AUTOMATIC_SWITCH` : Indique qu'une commutation automatique a lieu mais que la raison précise n'est pas connue.

`SR_RESTORED` : Indique que la commutation a cessé et que la situation est revenue à la normale.

`SR_SIGNAL_FAIL` : Indique qu'une commutation de protection a été déclenchée parce que le signal principal n'est plus reçu.

`SR_SIGNAL_MISMATCH` : Indique que les données ne sont pas formatées comme on s'y attendait. Par exemple, un signal dont les trames ne peuvent être correctement identifiées.

`SR_SIGNAL_DEGRADE` : Indique que la qualité du signal principal est passée en dessous d'un seuil acceptable (Bit Error Rate trop élevé).

`SR_AUTOMATIC_SWITCH` : Indique qu'une commutation automatique a lieu mais que la raison précise n'est pas connue.

`SR_MANUAL` : Indique que la commutation de protection a lieu à la suite d'une demande formulée par l'opérateur du réseau.

L'attribut `protectedTP` est le nom du PTP dans le groupe qui transporte normalement le trafic et qui fait donc l'objet d'une protection.

L'attribut `switchAwayFromTP` indique, lors d'une commutation, quel PTP transportait le trafic avant la commutation.

Finalement, l'attribut `switchToTP` indique quel PTP sert dorénavant à transporter le trafic.

4.3.2 Interface du ProtectionMgr_I

C'est grâce aux méthodes du ProtectionMgr_I que le client peut obtenir des informations sur les groupes de protection qui existent au niveau de l'EMS. Cette interface est en lecture seule : on peut obtenir des informations de protection, mais on ne peut pas les créer, les modifier, ni les détruire.

Les méthodes les plus intéressantes sont :

getAllProtectionGroups : Permet d'obtenir toutes les structures représentant des groupes de protection en-dessous d'un nom de ManagedElement particulier.

getProtectionGroup : Permet d'obtenir la structure de groupe de protection correspondant au nom passé en paramètre.

retrieveSwitchData : Sert à retourner l'état actuel d'utilisation d'un groupe de protection.

performProtectionCommand : Permet au NMS de demander à l'EMS de causer ou d'arrêter une commutation manuelle ou un verrouillage sur un groupe de protection.

D'autres méthodes sont disponibles, qui retournent les noms des TPs qui transportent du trafic protégé (`getAllProtectedTPNames`), non protégé (`getAllNUTTPNames`) ou non préemptible (`getAllPreemptibleTPNames`), selon le cas.

4.3.3 Interface de notification

L'interface de notification utilisée dans le cadre de ce travail n'est pas spécifiée par la norme MTNM 2.0, mais est celle provenant de MTNM 1.0. En plus de publier les objets de gestion de session, le serveur doit publier un objet Observable. L'objet Observable sert d'interface à laquelle des objets Observer du ou des clients peuvent s'enregistrer. En s'enregistrant, le client spécifie quel genre de notification il désire recevoir. Neuf genres de notifications existent :

Object Creation : Émise lorsqu'un nouvel objet est crée ; par exemple, si un nouveau ManagedElement est activé sur le réseau.

Object Deletion : Émise lorsqu'un objet est enlevé du réseau.

State Change : Émise lorsqu'une valeur d'état d'un objet change et indique quelle valeur de l'objet a changé.

Attribute Value Change : Émise lorsqu'un attribut d'un objet change et indique quelle valeur a changé.

Route Change : Émise lorsqu'un chemin est modifié dans un sous-réseau sous la responsabilité de l'EMS.

Protection Switch : Émise lorsqu'une commutation de protection débute ou cesse en indiquant quel PTP transportait le trafic et quel PTP le transporte maintenant ; en fait, toutes les informations de la structure `SwitchData_T` sont reproduites.

Alarm : Émise lorsqu'une alarme (ponctuelle ou temporelle) se produit sur un des objets représentés par l'interface.

Threshold Crossing Alert : Émise lorsqu'un seuil est atteint sur une certaine condition (par exemple, le nombre de commutations de protection depuis 15 minutes).

File Transfer Status : Émise lorsqu'un transfert de fichier (par exemple, un fichier journal de l'EMS) est prêt à commencer.

4.4 Résumé

Ce chapitre a introduit la norme MTNM 2.0 ; nous avons pris un moment pour introduire les concepts généraux de gestion réseau exprimés par le modèle TMN de l'ITU afin de mieux situer le rôle et la portée de MTNM 2.0.

Nous avons ensuite parlé des caractéristiques générales de la norme. Celle-ci permet que les parties désirant utiliser l'interface qu'elle définit communiquent entre eux via un bus CORBA. Le serveur de l'interface met à la disposition du client divers Managers qui permettront d'obtenir de l'information sur un aspect du réseau géré par le serveur. Ces managers, à leur tour, exportent des structures représentant de façon abstraite les éléments du réseau.

Les éléments qui nous intéressent particulièrement sont les groupes de protection qui représentent un ensemble de ports physiques pour lesquels un système de protection est mis en place.

L'autre aspect pertinent de la norme est le système de notification ; pour ce travail, il a été décidé d'employer les événements de MTNM 2.0 et de les coupler au système de notification plus simple de MTNM 1.0.

Chapitre 5

Architecture système

Le travail effectué dans ce projet faisait partie de l'implémentation d'un serveur pour une interface MTNM 2.0. Cette implémentation avait pour but de fournir à une plate-forme NMS (voir page 54) existante une interface par laquelle cette plate-forme pouvait être subordonné à un NMS différent via une interface normalisée. Le rôle du premier NMS était donc transformé en EMS (*Element Management System*).

L'architecture de la plate-forme NMS existante est illustrée à la Figure 5.1.

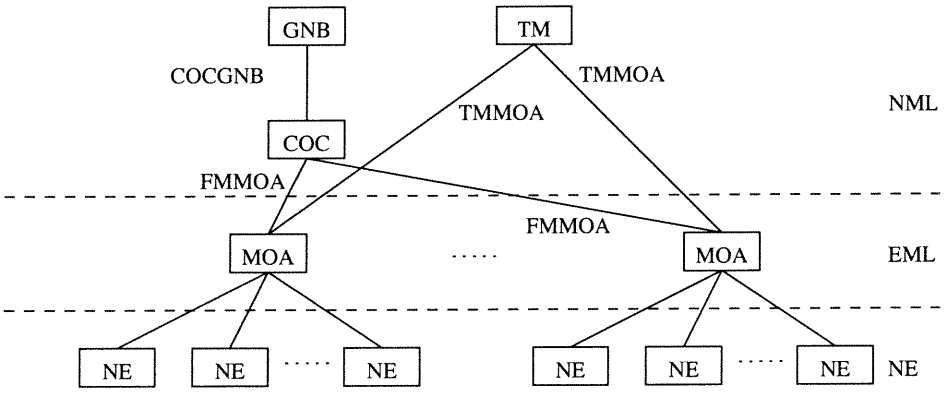


FIG. 5.1 – Système de gestion de réseau existant

L'architecture du système combiné est montrée à la Figure 5.2, et son abstraction en composantes d'un TMN, à la Figure 5.3.

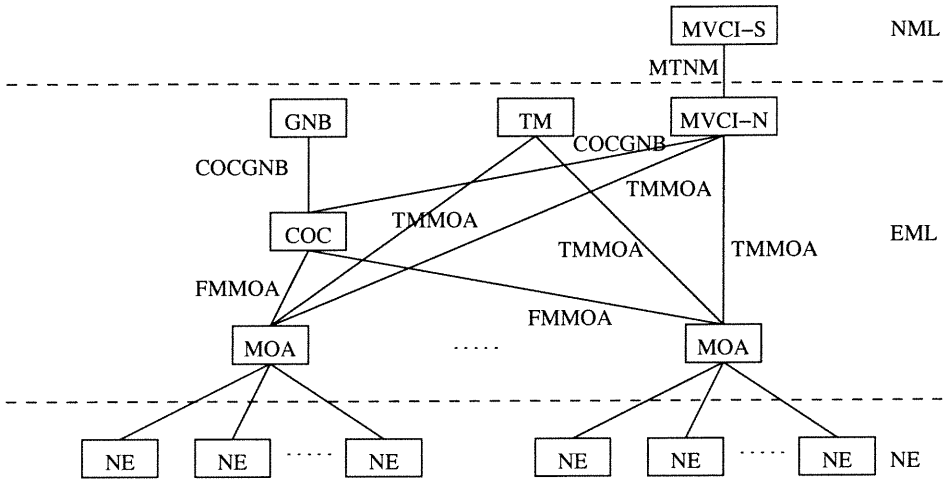


FIG. 5.2 – Système de gestion de réseau subordonné à un autre

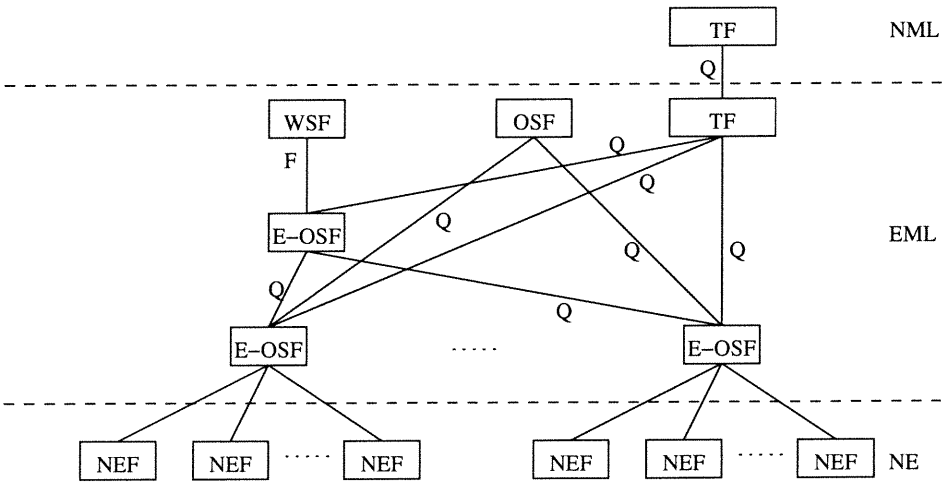


FIG. 5.3 – Système de gestion de réseau subordonné, modèle abstrait

5.1 Description des composantes

Les composantes les plus fondamentales sont les NEs. Ceux-ci sont gérés par des MOA, les *Managed Objet Agents*. Les MOAs remplissent les fonctions attribuées aux E-OSF dans le TMN.

Vient ensuite le COC ; cette composante a normalement un rôle de N-OSF et permet de centraliser toutes les données sur la gestion des éléments gérés. En outre, à ce point central peuvent se connecter les interfaces utilisateurs.

Le GNB (Graphical Network Browser) est une interface de ce genre et remplit donc la fonction de WSF. Le GNB permet à un utilisateur de constater l'état du réseau et de générer des ordres pour modifier la façon dont le service est rendu.

Le TM (Trail Manager) est aussi un système d'opération (OSF) ; il permet de gérer de bout en bout les connexions établies dans le réseau pour transporter les données.

Enfin MVCI-N est le Multi-Vendor CORBA Interface Northbound (l'apellation Northbound indique qu'il s'agit d'un serveur pour une couche située plus "haut"). Cette interface procure un serveur basé sur la norme MTNM en transformant les informations provenant des interfaces internes. C'est donc un bloc fonctionnel de type TF (Transformation Function) dans le TMN. Plus particulièrement, il s'agit d'un mécanisme de médiation (MD, Mediation Device) entre les blocs fonctionnels dont les mécanismes de communication sont incompatibles. Puisque ce mécanisme opère sur une interface Q, on le désignera comme QMD.

5.2 Description des interfaces

Toutes les interfaces décrites dans cette section fonctionnent entre les diverses composantes du TMN. Par conséquent, ce sont des interfaces de type Q. L'interface MTNM est l'interface de communication standard MTNM 2.0 (excepté pour quelques divergences, tel que, par exemple, le système de notification provenant de MTNM 1.0).

L'interface TMMOA est une interface interne au NMS ; il s'agit d'une interface basée sur XDR [Sun1995] (External Data Representation), dont le rôle est de gérer et de présenter les connexions dans le sous-réseau sous le contrôle de son serveur. Le nom de cette interface signifie Trail Manager - Managed Object Agent, d'après le nom des deux participants normaux.

L'interface FMMOA est aussi une interface interne au NMS basée sur XDR. Son rôle est plus varié : les trois composantes les plus importantes de sa fonctionnalité sont (i) de rapporter l'existence et l'état des NEs, (ii) d'indiquer comment les NEs sont connectés entre eux et (iii) de rapporter les alarmes indiquant des problèmes dans le réseau pour centralisation et analyse ; ces informations sont les alarmes, les commutations de protection, etc.

L'interface COCGNB est également une interface interne au NMS basée sur XDR. Le nom de cette interface signifie Collector - Graphical Network Browser. Son rôle consiste à transmettre les informations de fautes au GNB une fois qu'elles ont été centralisées et digérées par le COC.

5.3 Architecture interne de MVCI Northbound

La composante MVCI-N est construite selon une architecture modulaire à trois niveaux basée sur le flot des données dans le système (Figure 5.4).

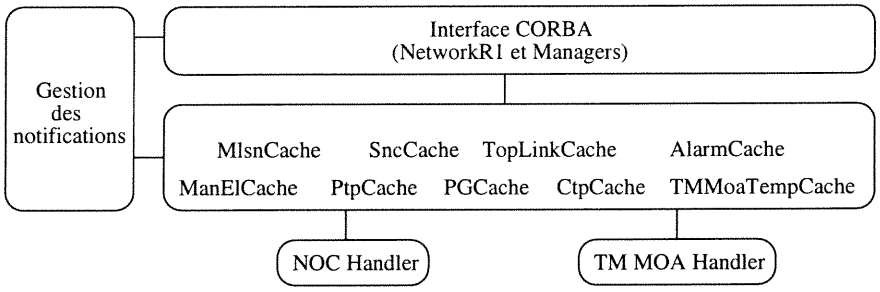


FIG. 5.4 – Système MVCI Northbound

Au niveau le plus bas se trouvent les gestionnaires d'interfaces. Ces gestionnaires ont pour rôle de maintenir une communication avec les autres éléments de l'EMS via les interfaces COCGNB et TMMOA. Leurs fonctions sont donc les suivantes :

- Déterminer avec quelles autres entités de l'EML communiquer
- Ouvrir une session de communication avec ces entités
- Obtenir les informations qui décrivent la topologie et l'état général du réseau sous le contrôle de ces entités au moment de l'initialisation de MVCI
- Rétablir la communication en cas de perte due à un problème temporaire
- Envoyer aux parties de MVCI-N intéressées les notifications d'événements en provenance

des entités

- Agir en tant qu'agent des parties de MVCI-N qui désirent de l'information via ces interfaces pour demander l'information et acheminer la réponse à la partie intéressée

La couche médiane est celle des caches. Les caches ont pour rôle de maintenir en mémoire une représentation fidèle du réseau. Pour ce faire, elles utilisent les services des gestionnaires d'interfaces et communiquent entre elles via un système de notification interne (voir section suivante).

AlarmCache : Est responsable de la gestion globale des alarmes, c'est-à-dire de rapporter les alarmes associés aux conditions anormales dans tout le réseau géré par l'EMS.

CtpCache : Est responsable des objets CTP (Connection Termination Point)

ManElCache : Est responsable des objets ManagedElement.

MlsnCache : Est responsable des objets MultiLayerSubnetwork. Les MLSN supportés par MVCI-N sont assez simples : il existe un et un seul sous-réseau pour chaque ManagedElement. Aucun autre sous-réseau ne peut exister.

PgCache : Est responsable des objets PGP (groupes de protection).

PtpCache : Est responsable des objets PTP (Physical Termination Point).

SncCache : Est responsable des objets SubnetworkConnection ; cette cache a de particulier qu'elle peut accepter des commandes en provenance de l'interface CORBA pour créer ou détruire des connexions, contrairement à toutes les autres caches qui sont en lecture seulement.

TMMoaTempCache : Est responsable de stocker des objets internes, les templates. Ces objets sont des entités provenant de l'interface TMMOA et modélisent en détail les capacités de chaque port sur les éléments gérés.

TopLinkCache : Est responsable des objets TopologicalLink. MVCI-N ne supporte pas ces objets pour l'instant.

Enfin, la couche supérieure est celle des Managers de l'interface MTNM, basés sur CORBA. Cette couche fournit l'interface de communication avec les clients de MTNM. Les requêtes qui y sont envoyées sont analysées et validées à ce niveau, et les données pertinentes sont extraites des caches et ensuite envoyées aux clients.

5.3.1 Système de notification

Comme nous en avons déjà discuté, les clients peuvent s'inscrire via l'interface Observable pour être alertés des changements dans le réseau. Chaque cache est responsable d'émettre des notifications quand survient un événement approprié pour le genre d'objet qu'elle contient.

Ce système est étendu à l'interne afin que les caches puissent s'enregistrer entre elles afin d'obtenir des notifications des autres caches. Par exemple, lorsqu'un changement survient dans la ManElCache, la PtpCache en est informée. Cela est utile, par exemple, au moment de la mise en service d'un nouvel élément géré; la PtpCache est avertie d'un tel événement et réagit en demandant au MOA, via l'interface TMMOA, la liste des ports sur le NE nouvellement affecté.

5.3.2 Implémentation

Au niveau logiciel, MVCI-N est implémenté en utilisant le langage de programmation C++ [ISO14882], [Str2000]. C'est un système multiprogrammé avec données partagées (*multithreads*) : chaque requête CORBA est gérée par son propre processus léger (*thread*), et chaque gestionnaire d'interface roule également dans son processus léger. Enfin, chaque message reçu par les gestionnaires est également traité dans son propre processus.

Chapitre 6

Méthodologie de développement

Dans le cadre de ce travail, nous avons utilisé un cycle de développement en linéaire séquentiel assez standard ([Pre1997]), passant par quatre étapes : analyse, conception, implémentation et véfification (test).

En bref, l'analyse consiste à réunir assez d'information pour comprendre et résoudre le problème; la conception consiste à formuler une solution; l'implémentation consiste à coder cette solution dans le langage de programmation choisi, et la vérification (test) vise à s'assurer que la solution implémentée résoud correctement le problème.

6.1 Exécution des tâches

Plus spécifiquement, l'exécution des tâches suivait un cadre semi-formel.

L'analyse est l'activité la moins formelle ; il s'agit de comprendre le problème et de mener les recherches nécessaires pour effectuer la tâche. Pour un produit, cela peut impliquer la création de documents spécifiant les caractéristiques attendues, par exemple. Par contre, pour des tâches de développement semblables à celles que nous avons effectuées, il s'agit plutôt d'acquérir la connaissance nécessaire pour produire un design adéquat.

Dans notre cas, il s'agissait de faire le travail pour bien comprendre les spécifications des interfaces et des autres parties du logiciel MVCI. Du travail d'ingénierie inverse a aussi été

requis.

La conception consiste à élaborer une solution au problème posé. Il s'agit d'abord d'un processus cognitif, dont le résultat est mis sur papier dans un document de design (DD). Ce document, une fois achevé, est révisé en groupe afin de s'assurer qu'il est complet et précis. Si des problèmes sont découverts ou si des détails manquent, le document est modifié et révisé itérativement. Le but ultime d'un DD est de fournir assez de détails afin que l'étape suivante, l'implémentation, soit assez prévisible et puisse être réalisée sans diffulté, possiblement par une autre personne que le concepteur.

L'implémentation est l'étape où le design est codifié dans le langage de programmation. C'est aussi l'étape où un design de plus bas niveau est réalisé ; c'est la tâche de l'implémenteur de trouver une façon efficace de réaliser la conception. L'implémenteur doit faire part de ses choix en inscrivant des commentaires aux endroits appropriés du code qu'il écrit. Une fois l'implémentation réalisée, une inspection de code est effectuée. L'inspection de code est similaire à la révision du document de design : le groupe de développement lit le code afin d'y repérer les problèmes potentiels, et ce, de façon itérative jusqu'à ce qu'il soit jugé suffisamment correct pour être intégré à la version officielle du projet.

Trois classes de problèmes peuvent survenir : (i) explications insuffisantes : il s'agit du cas où les commentaires pour décrire le code sont jugés déficients ; (ii) problèmes non opérationnels : il s'agit du cas où le code n'est pas organisé de façon optimale quant à la facilité de lecture, la simplicité, la facilité d'entretien ou la performance ; (iii) problèmes opérationnels : il s'agit du cas où le code ne remplit pas les objectifs du design, ou est autrement fautif.

La dernière étape de développement consiste en l'exécution de tests. Il s'agit d'une étape de contrôle de qualité : le produit est soumis à des tests plus ou moins formels, qui permettent de trouver des problèmes qui n'auraient pas été repérés lors des étapes de développement précédentes. La découverte d'un problème entraîne la création d'un rapport de suivi, qui détaille les étapes prises pour résoudre le problème, si celui-ci est jugé suffisamment grave. Le cas échéant, un correctif est apporté ; le code requis pour corriger le problème est également soumis à une inspection.

6.2 Étendue des travaux

Nous avons participé au développement du système de protection de MVCI-N. Le système a été développé en deux phases étant donné les contraintes liées aux équipements réseau.

6.2.1 Phase I - Informations topologiques

Le but de la phase I fut essentiellement la planification générale du système et l'utilisation des informations disponibles via l'interface TMMOA pour découvrir la présence de groupes de protection.

Le Tableau 6.2.1 indique les différents aspects de la phase I et dans quelle mesure nous les avons nous-même réalisés.

TAB. 6.1 – Activités de la phase I

Activité	Analyse	Conception	Implémentation	Test
Cache des groupes de protection	0 %	10 %	100 %	20 %
Interface du ProtectionMgr_I	10 %	100 %	35 %	0 %
Intégration au système de notification	50 %	100 %	100 %	0 %

La cache des groupes de protection (PGCache) garde en mémoire toutes les informations ayant trait aux groupes de protection sur le réseau. Son design était presque terminé lorsque nous avons commencé à travailler sur le projet. Nous avons fait une révision du design et nous sommes chargé de l'implantation.

Le premier problème de cette partie fut de trouver une façon efficace de mettre à jour la cache lorsqu'un changement survient dans la topologie du réseau. En effet, la cache doit être cohérente et nous voulons éviter des algorithmes qui nous forceraient à recalculer trop d'informations.

Ce problème vient du fait que les groupes de protection sont, comme il a déjà été expliqué, des structures qui sont propres à MTNM 2.0. Les interfaces sous-jacentes (TMMOA et COCGNB), quant à elles, n'emploient pas ce concept. Il faut donc calculer la présence (détecter) des groupes de protection. Ce calcul n'est pas trivial. En fait, l'information en provenance de ces interfaces décrit les PTP. Il faut donc essayer de trouver, en fonction des attributs et caractéristiques de

ces PTP, lesquels vont ensemble.

Une grande partie des caractéristiques de ces PTPs est contenue dans des structures appelées templates. Le rôle des templates est de permettre la modélisation des capacités et caractéristiques des PTP de façon générique. Les templates sont basés sur le modèle G.805 et fonctionnent donc eux aussi par couches. Les structures utilisées pour représenter les templates sont assez complexes afin d'être suffisamment flexibles pour représenter divers types de PTP. En conséquence, le code requis pour trouver les informations qui indiquent la présence d'un groupe de protection est assez complexe. Trouver une façon élégante et efficace pour repérer ces groupes de protection fut le deuxième problème majeur de cette phase.

L'interface du ProtectionMgr_I permet d'interroger MVCI-N sur les données contenus dans la cache de protection. Quelques méthodes ont été implantées par une autre personne selon le design que j'ai écrit. Ce design n'était pas particulièrement difficile ; il fallait seulement faire attention aux exceptions C++ et aux fuites de mémoires (*leaks*).

Enfin, l'intégration au système de notification permet aux clients d'être avertis lorsque des changements surviennent dans la cache des groupes de protection.

Le travail effectué dans la phase I est décrit au chapitre suivant.

6.2.2 Phase II - Informations statutaires

Le but de la phase II était d'intégrer les informations sur l'état du réseau aux informations topologiques afin de pouvoir rapporter fidèlement l'information déclarée dans les structures SwitchData. L'information appropriée provenait de l'interface COCGNB.

Cette interface interne n'était pas documentée ; il a donc fallu effectuer un travail d'ingénierie inverse afin de découvrir son fonctionnement et la valeur sémantique des structures de données. Ce travail d'analyse fut d'une assez grande envergure ; il nous a permis en outre de déterminer que l'information que nous recherchions n'était disponible par cette interface que pour les ports des agrégats des NEs, et non pour les ports tributaires.

Cette phase ne comportait que cette activité ; nous avons réalisé les tâches d'analyse, de conception et d'implémentation. Quelques tests minimaux ont aussi été réalisés.

L'analyse dans cette phase est la partie qui a requis le plus de temps ; en effet, nous ne

possédions aucune information à propos de l'interface COCGNB si ce n'est que le module GNB, client de cette interface, avait accès aux informations que nous désirions obtenir et qu'une autre partie de MVCI-N se servait de cette interface pour énumérer les éléments du réseau.

Nous avons donc dû d'abord analyser l'interface, c'est-à-dire trouver quelles étaient les primitives de communication et surtout quelles en étaient leur valeur sémantique. L'interface COCGNB étant de type XDR, il a été facile de trouver quelles étaient les primitives de communication : le format des données est en effet contenu dans un fichier qu'utilisent toutes les entités qui communiquent par cette interface. Toutefois, ce fichier n'est pas commenté et nous avons dû faire des expériences avec des éléments réseau pour tenter de voir dans quelles circonstances chacun des messages est employé.

Grâce à ce travail d'analyse, nous avons découvert l'existence de deux structures importantes pour les données de protection : les résumés de protection et les plans de connectivité. Lors de la conception, nous avons écrit un plan pour intégrer les informations données par cette structure aux caches de données de MVCI-N. Le problème principal rencontré lors de cette étape avait à voir avec l'ordre d'initialisation : la solution la plus simple aurait été de mettre toutes les données en provenance de l'interface COCGNB dans la cache des groupes de protection. Or ce n'était pas possible car l'initialisation de cette cache n'est pas complétée au moment où les premières données en provenance de l'interface COCGNB peuvent arriver.

Le travail effectué lors de cette phase est discuté au chapitre 8.

Chapitre 7

Conversion d'informations topologiques

La conversion d'informations topologiques est, en somme, le repérage des groupes de protection et de leurs attributs en fonction des informations provenant de l'interface TMMOA. Cela signifie que selon la façon dont les différents PTPs sont connectés ensemble, et quelles règles régissent leurs interactions, on peut repérer des conditions desquelles on peut déduire l'existence des groupes de protection.

Par exemple, si on a deux PTPs et qu'on cherche à savoir s'ils peuvent former un groupe MSP 1+1, on peut vérifier certaines choses, comme par exemple, si leur débit est le même... si leur débit n'est pas le même, automatiquement les PTPs sont trop différents pour former un groupe de protection.

L'interface TMMOA est normalement destinée au Trail Manager. Le but de ce logiciel est de provisionner des voies de bout en bout du réseau. Pour ce faire, il doit connaître quels sont les ports, quels CTP existent et où, quels CTP sont compatibles entre eux et où sont les connexions actuelles. Par conséquent, cette interface doit fournir de façon très précise comment les connexions peuvent être établies et quelles règles régissent l'établissement de connexions – il ne faudrait pas que l'utilisateur puisse créer des connexions impossible à réaliser ou qui compromettraient l'intégrité du transfert de données dans le réseau.

L'interface TMMOA permet donc d'obtenir toutes ces informations via divers messages échangés entre le serveur MOA et le client (dans ce cas-ci, MVCI-N).

7.1 Templates

L'interface TMMOA utilise des structures de données appelées templates pour décrire les points de terminaison physiques. À tout moment, chaque PTP est associé à un template qui lui donne des caractéristiques ; un template est donc une façon de modéliser une configuration de PTP. (Une reconfiguration de la fonction d'un PTP peut causer l'utilisation d'un template différent ; par contre, la création ou la rupture de connexions sur les CTPs de ce PTP n'a aucun impact sur le template.)

Le but ultime d'un template est de réduire la quantité d'information qui transite sur le réseau de gestion : en effet, sur un réseau comportant quelques milliers de PTPs, il est raisonnable de penser que ceux-ci partagent des caractéristiques semblables. Par exemple, un ADM possède très souvent des dizaines de ports tributaires identiques ; ceux-ci seront tous concentrés vers un lien à haute capacité. Ces liens à haute capacité pourraient, par exemple, être ceux d'agrégats d'un anneau MSSPRING. Dans ce cas, tous les ADM de l'anneau auront des agrégats similaires.

En conséquence, il est avantageux de centraliser les parties qui ont tendance à se répéter, rôle rempli par les templates.

De plus, chaque PTP peut être membre d'une ou de plusieurs instances de *groupes de CTPs.* (Un PTP possède une liste d'instances de groupes, et dès qu'un des CTPs modélisés sur ce PTP fait partie de l'instance, celle-ci est ajoutée à la liste du PTP). Les groupes de CTPs permettent de centraliser les caractéristiques communes à plusieurs CTPs et les règles d'interaction entre ces CTPs et les autres.

Dans MVCI, les templates sont stockés dans la TMMoaTempCache ; au moment où un nouvel MOA est ajouté au réseau, ses templates sont importés dans cette cache et deviennent disponibles pour consultation ; les templates sont des structures XDR.

Deux sortes de templates existent : les templates de PTP et les templates de groupes CTP. Les templates de PTP décrivent un PTP ; les templates de groupes CTP ont pour objet un ensemble de CTP et décrivent des caractéristiques communes à ceux-ci. Notons que le même

template de groupe CTP peut être utilisé pour décrire plusieurs groupes de CTPs disjoints. Pour distinguer les différents groupes utilisant le même template, chaque groupe possède un numéro d'instance unique. Ainsi, lorsqu'un PTP est membre d'une instance de groupe de CTP, le numéro d'instance du groupe est spécifié tout comme le numéro du groupe, lequel indique quel template de groupe utiliser. Le rôle des templates de groupes CTP est de permettre la création de groupes ayant les mêmes caractéristiques (indiquées par le template) mais dont les membres sont disjoints (selon leur numéro d'instance).

7.1.1 Templates de points de terminaison physiques

Les templates de PTP sont identifiés par un numéro (unique), un nom décrivant la fonction du PTP et un numéro de version.

Un template indique si un PTP est bidirectionnel, une source ou une destination seulement. Enfin, une liste de couches (dans un tableau indexé) est fournie. Ces couches correspondent à celles du modèle énoncé au chapitre 2 ; cependant, il existe une différence importante qui permet de faciliter leur interprétation : au lieu de partir du CTP, de décrire les fonctions de terminaison et d'adaptation, et de finir au TTP, les couches des templates commencent au TTP, décrivent les fonctions de terminaison et d'adaptatation, et finissent au CTP du réseau de la couche suivante.

La première couche d'un template décrivant un PTP commence au niveau PMS (Physical Media Section). Il s'agit d'un niveau qui décrit les caractéristiques physiques du conducteur ; par exemple, la fibre qui transporte l'onde optique.

Le niveau suivant dépend de la technologie employée ; pour les PTP de type SDH, ce sera OS, RS, MS et ainsi de suite.

Enfin, une autre différence dans les templates est qu'il n'y a pas de sous-couche ; toutes les couches sont au niveau principal. Ainsi, la couche MSP est décrite similairement à la couche MS.

7.1.2 Description des couches

Les couches peuvent être spécifiées dans n'importe quel ordre dans un template ; cependant, il est usuel de commencer par le bas (couche physique) et de remonter la chaîne des TCP. Chaque

couche contient cinq parties permettant de la décrire :

Détails de type : Cette partie identifie le type de couche (PMS, OS, RS, etc.), un qualificateur général s'y rapportant (par exemple, FIBER pour PMS, STM-4 pour RS, etc.) et une liste optionnelle d'attributs décrivant la couche (par exemple, la longueur d'onde pour une couche OS, la fréquence pour une couche WS (Wireless Section), etc.)

Règles d'adaptation : Cette partie indique avec quoi le TTP est compatible, c.-à-d., à quoi il peut être adapté. C'est dans cette partie que l'adaptation indirecte est définie s'il y a lieu. Une règle d'adaptation contient un facteur multiplicateur (qui indique le nombre de CTPs) et une liste de possibilités pour réaliser l'adaptation. Ces possibilités peuvent être inclusives (les possibilités sont reliées par une règle "et") ou exclusives (les possibilités sont reliées par une règle "ou"). Deux types de possibilités existent : la première possibilité est de fournir un TP compatible en donnant ses détails de type (ce TP sera soit un CTP d'une couche ayant les mêmes détails, soit un TTP offrant une possibilité d'adaptation similaire). La deuxième possibilité est de fournir un niveau d'indirection en indiquant une autre règle d'adaptation parmi celles fournies (cette possibilité permet de représenter les adaptateurs indirects présentés à la section 3.1.4).

Par exemple, l'adaptation d'un réseau MS STM-64 vers un réseau HP VC4 implique deux règles : la première règle est de type AUGmap, laquelle est un adaptateur indirect pour représenter le multiplexage d'AUG (voir Figure 2.1). Cette règle spécifie un facteur multiplicateur de 64, ce qui signifie que 64 instances de la règle d'adaptation suivante existent. La règle d'adaptation suivante spécifie, bien entendu, un AUG qui sera composé d'un conteneur HP VC4.

Si nous avions affaire à un PTP un peu plus évolué qui en plus supporte des connexions HP VC3, la deuxième règle aurait eu une mention "ou" pour indiquer un choix avec une troisième règle, laquelle aurait spécifié un AUG composé de trois conteneurs HP VC3.

Règles d'assemblage : Cette partie indique comment agréger des CTPs ensemble pour former d'autres CTPs. Aucun template n'utilise actuellement les règles d'assemblage.

Règles de connexion du TTP : Cette partie indique comment connecter le TTP vers un CTP d'une couche serveur. Ces règles sont données à la section 7.1.4.

Règles de connexion des CTPs : Cette partie indique comment connecter les CTP vers des TTP du client ou à des CTPs du même niveau.

7.1.3 Groupes CTP

Les groupes CTP sont employés pour grouper des CTP similaires ensemble. Ces groupes permettent de réduire la taille des templates de PTPs en centralisant certaines caractéristiques communes. Un template de groupe contient deux éléments : son identification (un numéro, un nom et un numéro de version) et une liste de règles de connexion.

Les PTPs possèdent tous une liste d'instances de groupes CTPs. Lorsqu'une instance se trouve dans cette liste, cela signifie qu'un moins un des CTPs de l'une des couches du template du PTP fait partie de cette instance de groupe. Les templates du PTP ne spécifient que le numéro du template du groupe CTP. Un tel template donne les caractéristiques de l'instance dont le numéro est spécifié par le PTP. L'existence d'instances permet d'avoir plusieurs groupes distincts qui ont des caractéristiques similaires (données par leur template).

7.1.4 Règles de connexion

Les règles de connexion indiquent comment connecter les points de connexion entre eux : autrement dit, comment les TTPs doivent entrer en interaction avec les CTPs et vice-versa. Plus abstraitement, ils spécifient la sémantique des points de référence du modèle fonctionnel. Les règles ont un type ainsi qu'une liste de paramètres ; chaque unité dans la liste de paramètres représente une façon correcte d'appliquer la règle.

Les types de règles de connexion les plus importants sont :

connects_externally : Est utilisé pour indiquer qu'il s'agit d'une connexion physique en utilisant le médium de connexion ; il est utilisé dans le cas du TTP du niveau PMS.

must_connect_to : Est utilisé pour indiquer qu'une connexion à un TP du même PTP selon les paramètres indiqués existe de façon obligatoire.

must_be_connected : Est utilisé pour indiquer qu'une connexion existe de façon obligatoire selon les paramètres spécifiés ; ce type de règle est utilisé pour les TTP par exemple.

must_not_connect_to : Est utilisé pour indiquer qu'une connexion selon les paramètres indiqués est interdite.

may_connect_to : Est utilisé pour indiquer qu'une connexion selon les paramètres indiqués est permise sans être obligatoire.

may_connect_to_any_ctp_in_group : Est utilisé pour indiquer qu'une connexion avec les membres du groupe indiqué dans les paramètres est permise. Si aucun groupe n'est indiqué dans une des unités de paramètres, cela signifie qu'il faut utiliser les instances de groupes auxquelles le PTP appartient.

use_ctp_group_rules : Est utilisé pour indiquer que les règles sont spécifiée dans le groupe CTP mentionné dans les paramètres ; si aucun groupe n'est indiqué, cela signifie qu'il faut utiliser les instances de groupes auxquelles le PTP appartient.

may_connect_to_any_ctp_in_group_on_one_to_one_basis : Est utilisé dans un groupe CTP pour indiquer que les membres du groupes peuvent être interconnectés entre eux un à un.

Les paramètres les plus importants qui peuvent être spécifiés pour chaque connexion sont :

- Une liste de groupes de CTP à utiliser selon la sémantique de la règle ;
- Une liste de noms de CTP à utiliser selon la sémantique de la règle (on ne peut utiliser des noms de CTP pour la règle use_ctp_group_rules, par exemple) ;
- Le type de connexion : comment les TPs qui sont reliés ensemble doivent interagir ;
- Les points de connexion : quel rôle ce TP peut prendre dans les connexions établies selon la règle choisie et le type mentionné.

La prochaine section détaille quelques types de connexions.

7.1.5 Types et points de connexions

Les types de connexions confèrent une valeur sémantique à l'établissement d'une connexion, c'est-à-dire les rôles que prennent les différentes parties. Une connexion peut en effet impliquer deux, trois ou quatre TPs. Les TPs qui font partie d'une connexion peuvent prendre un rôle parmi deux, trois ou quatre rôles. Ces rôles sont définis par des points de connexion, lesquels sont nommés A, B, A' et B'. Les points B et B' sont habituellement (mais pas nécessairement) du côté de la couche client ; les points A et A' sont habituellement du côté de la couche serveur. Cinq types de connexions existent :

- Les connexions les plus simples sont les connections de type "unprotected". Ces connexions impliquent deux TPs, A et B ; l'information caractéristique est transmise de B à A et vice-versa.

- Les connexions du type "protected" ont trois points : A, A' et B. L'information caractéristique est continuellement transmise de B à A et A'. Par contre, un sélecteur détermine laquelle des sources (A ou A') devrait être utilisée pour fournir l'information à B.
- Les connexions du type "open scissors" ont quatre points. Dans cette configuration, un sélecteur détermine laquelle des sources (A ou A') devrait être utilisée pour fournir l'information au point B, et laquelle des sources (B ou B') devrait être utilisée pour fournir l'information au point A. La source A alimente toujours le point A', et la source B alimente toujours le point B'.
- Les connexions du type "closed scissors" ont également quatre points. Cette configuration est semblable aux "open scissors", mais elle va dans les deux sens, et l'information sera envoyée aux deux destinations. Ainsi, un sélecteur détermine laquelle des sources (A ou A') devrait être utilisée pour fournir l'information aux clients B et B', et laquelle des sources (B ou B') devrait être utilisée pour fournir l'information aux serveurs A et A'.
- Les connexions du type "stubbed scissors" ont aussi quatre points. Ce genre de connexion est similaire aux "closed scissors", mais deux des points sont connectés à des entités nulles (soient A et B ou A' et B'). Ce genre de connexion est utile pour modéliser les connexions protégées 1:N avec trafic supplémentaire : lorsque le sélecteur choisit un point nul, la connection est effectivement désactivée.

7.2 Patrons à rechercher

Le but de cette partie du travail est de repérer les groupes de protection parmi l'information topologique fournie par l'interface. L'information qui concerne les PTPs (lesquels forment les groupes de protection) se trouvant dans les templates, il faut donc analyser ceux-ci afin d'y repérer des groupes de protection.

Les templates étant des structures extrêmement flexibles, il faudra limiter les patrons à rechercher aux constructions réellement déclarées par les MOAs employés.

7.2.1 Patron MSP 1+1

Un groupe MSP 1+1 peut se déduire en présence de deux PTPs. Le premier PTP transporte normalement le trafic et a les caractéristiques suivantes :

- Présence d'une couche de type MSP avec qualificatif NORMAL
- Absence d'une couche de type MSP avec qualificatif EXTRA
- Présence d'une règle "must be connected" au TTP de la couche MSP, dont les paramètres spécifient un type de connexion "protected" et le point de connexion B
- Présence d'une règle "must connect to" au CTP de la couche MS, dont les paramètres spécifient un type de connexion "protected" et le point de connexion A
- Présence dans la couche MS d'une adaptation vers la couche MSP

Le second PTP transporte le trafic en situation de commutation et a les caractéristiques suivantes :

- Absence de TTP MSP
- Présence d'une règle "must connect to" au CTP de la couche MSP, dont les paramètres spécifient un type de connexion protected et le point de connexion A'

Le qualificatif des sections MS doit être le même pour les deux PTPs. Enfin, les PTPs doivent faire partie de la même instance de groupe CTP ; ce groupe doit spécifier la règle de connexion "may connect to any ctp in group". Le tout peut être résumé par la Figure 7.1.

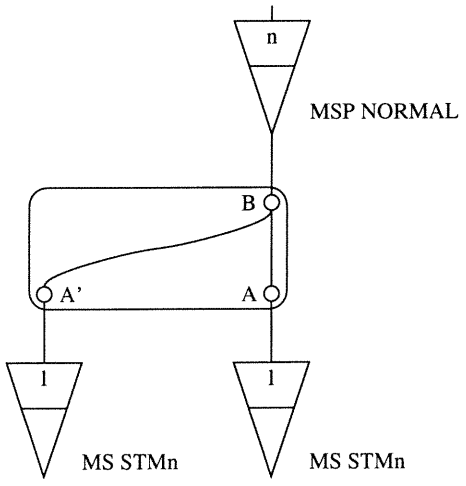


FIG. 7.1 – Patron MSP 1+1

7.2.2 Patron MSP 1:1

Il n'a malheureusement pas été possible de trouver un patron utilisé pour les groupes MSP 1:1. Cela n'implique pas qu'il ne peut y en avoir ; seulement que les MOA utilisés ne déclaraient pas avec suffisamment de précision les relations entre les PTPs pour pouvoir déduire la présence d'un tel groupe.

Autrement, le patron aurait été semblable à celui de la Figure 7.2.

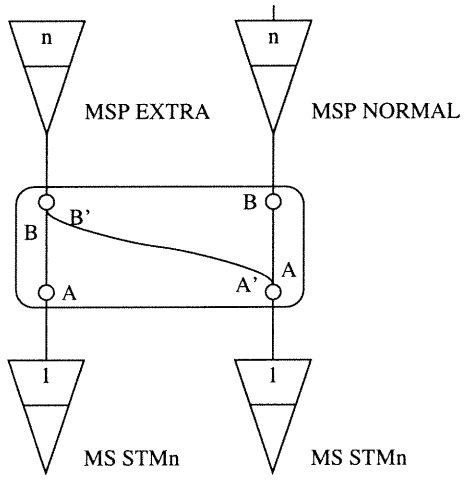


FIG. 7.2 – Patron MSP 1:1

Le premier PTP d'un tel groupe a les caractéristiques suivantes :

- Présence d'une couche de type MSP avec qualificatif NORMAL
- Absence d'une couche de type MSP avec qualificatif EXTRA
- Présence d'une règle "must connect to" au CTP de la couche MS, dont les paramètres spécifient un type de connexion "stubbed scissors" et le point de connexion A
- Présence d'une règle "must be connected" au CTP de la couche MS, dont les paramètres spécifient un type de connexion "stubbed scissors" et le point de connexion A'
- Présence d'une règle "must connect to" au TTP de la couche MSP, dont les paramètres spécifient un type de connexion "stubbed scissors" et le point de connexion B

Le deuxième PTP d'un tel groupe a les caractéristiques suivantes :

- Présence d'une couche de type MSP avec qualificatif NORMAL

- Absence d'une couche de type MSP avec qualificatif NORMAL
- Présence d'une règle "must connect to" au CTP de la couche MS, dont les paramètres spécifient un type de connexion "stubbed scissors" et le point de connexion A
- Présence d'une règle "must be connected" au TTP de la couche MSP, dont les paramètres spécifient un type de connexion "stubbed scissors" et le point de connexion B'
- Présence d'une règle "must connect to" au TTP de la couche MSP, dont les paramètres spécifient un type de connexion "stubbed scissors" et le point de connexion B

7.2.3 Patron 2F-BLSR sans trafic supplémentaire

Un groupe 2F-BLSR sans trafic supplémentaire peut se déduire en présence de deux PTPs. Les deux PTPs sont symétriques et possèdent les caractéristiques suivantes :

- Présence d'une couche de type MSP avec qualificatif NORMAL
- Absence d'une couche de type MSP avec qualificatif EXTRA
- Présence d'une adaptation à la couche MSP NORMAL qui n'utilise que la moitié des canaux de la couche MS (par exemple, 2 HP-VC4 pour un MS STM-4)
- Un groupe CTP doit être utilisé pour les connexions permises au CTP du niveau MSP. Ce groupe CTP doit être utilisé avec (ou son template doit mentionner) la règle "may connect to ctp on one to one basis in group"

Les deux PTPs doivent avoir le même qualificatif dans la couche MS. Le tout est résumé par la Figure 7.3.

7.2.4 Patron 2F-BLSR avec trafic supplémentaire

La présence d'un groupe 2F-BLSR avec trafic supplémentaire peut être déduite lorsque les PTPs ont les mêmes caractéristiques que ceux d'un groupe 2F-BLSR sans trafic supplémentaire ; cependant :

- On ajoute une couche de type MSP avec qualificatif EXTRA.
- Tout comme la couche MSP NORMAL, la couche MSP EXTRA utilise une moitié des canaux de la couche MS.

Le tout peut être résumé par la Figure 7.4.

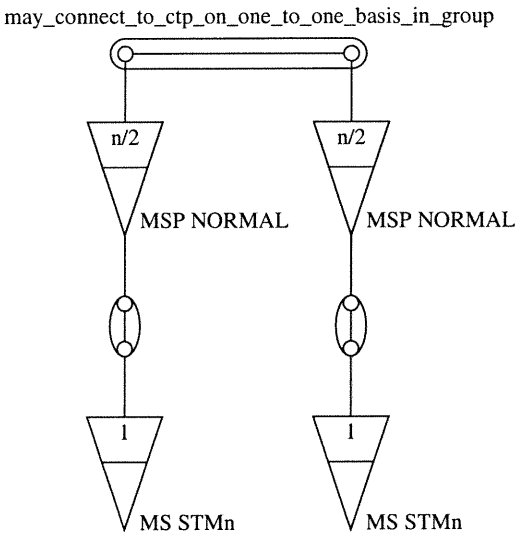


FIG. 7.3 – Patron 2F-BLSR sans trafic supplémentaire

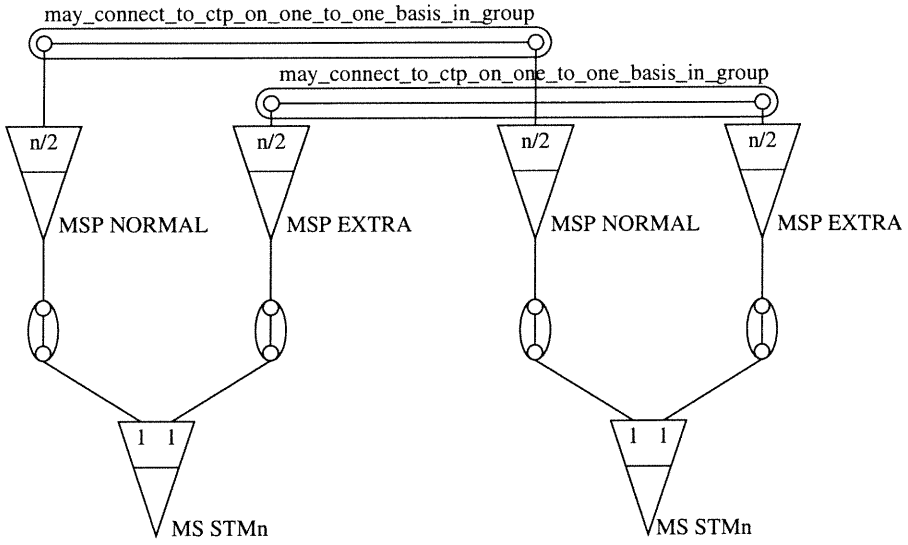


FIG. 7.4 – Patron 2F-BLSR avec trafic supplémentaire

7.2.5 Patron 4F-BLSR

La présence d'un groupe 4F-BLSR, avec ou sans trafic supplémentaire, est déduite en trouvant un groupe CTP commun mentionné dans les règles de connexion des CTP des couches MSP NORMAL des deux PTPs servant au trafic normal de groupes de protection MSP. Ce groupe CTP doit être utilisé avec ou doit mentionner une règle "may connect to ctp on one to one basis in group", comme l'illustre la Figure 7.5.

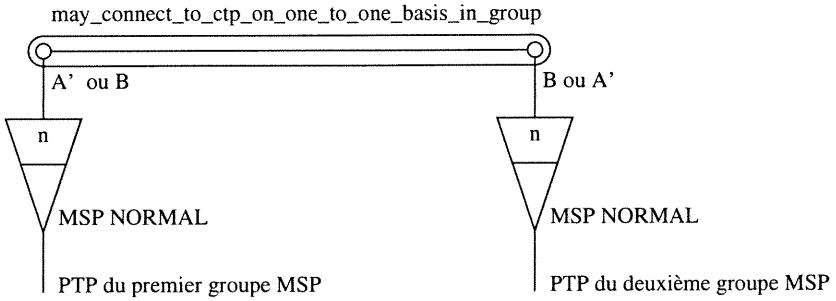


FIG. 7.5 – Patron 4F-BLSR

7.3 Algorithme général de recherche

Trois situations peuvent provoquer la recherche de groupes de protection : (i) l'ajout d'un nouvel élément géré, (ii) l'ajout d'un nouveau PTP sur un élément géré existant et (iii) la modification d'un PTP existant.

Dans le premier cas, on considère qu'on part de zéro et qu'on veut chercher tous les groupes de protection sur l'élément géré. Dans le deuxième cas, on désire chercher les seuls groupes de protection dont le nouveau PTP peut faire partie; dans le troisième cas, on vérifie que la modification est significative; si oui, on considère nécessaire de recalculer les groupes de protection sur tout le NE, étant donné les répercussions potentiellement importantes que peut causer le changement d'un PTP.

Avant d'expliquer l'algorithme de recherche, il importe de savoir que tout PTP ne peut faire partie, au maximum, que de deux groupes de protection : d'abord, un PTP ne peut faire partie que d'un groupe MSP (soit 1+1, soit 1:1) ou 2F-BLSR.

Ensuite, s'il fait partie d'un groupe MSP, il peut aussi faire partie d'un groupe 4F-BLSR. Cela est dû au fait qu'un groupe 4F-BLSR est composé de deux sous-groupes MSP, un pour chaque côté de la boucle (tel que mentionné à la section 4.3.1).

L'algorithme implémenté se déroule comme suit :

Soit E un ensemble de PTPs ; E contient initialement le [cas (ii)] ou les [cas (i) et (iii)] PTPs pour lesquels il faut tenter de trouver des groupes de protection. Une des conséquences des conditions de recherche énumérées ci-dessus est que tous les éléments de E sont situés sur le même élément géré.

Soit R un ensemble de PTPs ; R contient les PTPs candidats pour former des groupes de protection avec ceux de l'ensemble E ; initialement, tous les PTPs sur l'élément géré où se trouvent les PTP de E. Cela entraîne, pour les cas (i) et (iii), que $R = E$, et pour (ii), que $E \subseteq R$.

Soit $f(a, D_a, b)$ une fonction qui retourne vrai si les PTPs a et b forment un groupe de protection, faux autrement. D_a représente des propriétés calculées de a aidant au calcul de f.

Soit $k(g, h)$ une fonction qui retourne si les groupes de protection MSP g et h forment un groupe de protection 4F-BLSR.

Soit M l'ensemble des groupes MSP sur l'élément géré dans la cache au moment de démarrer l'algorithme.

1. Si $E = \emptyset \vee |R| < 2$, terminer la recherche.
2. Prendre un x quelconque dans E ; x existe puisque E n'est pas vide ; calculer certaines données aidant le calcul, les mettre dans D_x.
3. $E \leftarrow E \setminus \{x\}$; $R \leftarrow R \setminus \{x\}$
4. $\forall y \in R$, calculer D_y et vérifier si $f(x, y, D_x, D_y)$ est vrai. Dès qu'un y tel que $f(x, y, D_x)$ est trouvé, passer à l'étape 4(a) ; sinon, passer directement à l'étape 5.
   (a) Ajouter le groupe formé de x et y à la cache ; soit g ce groupe.
   (b) $E \leftarrow E \setminus \{y\}$; $R \leftarrow R \setminus \{y\}$
   (c) Si le groupe g est de type MSP, alors passer à 4(a)(i) ; sinon, passer directement à l'étape 5.

i. $M \leftarrow M \cup \{g\}$.

ii. $\forall h \in M$, vérifier di $k(g,h)$. Dès qu'un h tel que $k(g,h)$ est trouvé, ajouter ce groupe à la cache, et retourner à l'étape 1.

5. Retourner à l'étape 1. $\Diamond$

7.3.1 Remarques

1. Il s'agit de la condition d'arrêt ; R a été implanté comme une liste. Si $|R|$ est inférieur à deux, la formation de groupes de protection est impossible puisque le nombre minimal de PTPs distincts dans un groupe de protection est de deux. L'un de ces PTPs étant $x \in E$, et puisque $E \subseteq R$, $x \in R$, $|R|$ doit être supérieur à un pour avoir le nombre minimal de PTPs distincts pour former un groupe de protection.

2. La façon la plus simple de choisir x est de prendre le premier dans la liste. On calcule certaines propriétés de x que nous passerons à f pour faciliter la tâche de f. On effectue cette étape maintenant afin de ne pas répéter le calcul à chaque appel de f avec le même x.

3. Chaque tour de boucle a pour but de trouver tous les PTPs dont x fait partie ; on enlève x de E afin que l'algorithme progresse ("liveness property") ; cela garantit que la recherche ne sera pas effectuée deux fois pour le même PTP. De plus, on peut enlever x de R, car s'il ne fait pas partie d'un groupe de protection à ce tour de boucle, il n'en fera pas partie plus tard, et il est donc inutile de le garder dans la liste R des PTPs susceptibles de former des groupes de protection. Notons que puisque nous enlevons x des deux ensembles, nous maintenons l'invariant $E \subseteq R$.

4. On essaie de former un groupe avec x et les éléments de R un à un. Si on ne trouve pas, alors il n'y a pas de groupe à deux éléments pouvant être formés avec x ; on retourne alors à l'étape 1 pour essayer un nouveau x. Si on réussit, c'est qu'on a trouvé un groupe MSP 1+1, 1:1 ou 2F-BLSR avec un des y. Lorsqu'on trouve un tel groupe, on peut immédiatement cesser de chercher dans les y sur lesquels on n'a pas encore itéré, car x ne peut faire partie d'un seul groupe MSP ou 2F-BLSR à la fois ; cela implique que f va retourner vrai pour un seul y.

   (a) Le groupe trouvé est ajouté à la cache.

(b) Comme il ne peut y avoir qu'un seul tel groupe dans lequel se retrouve x ou y, on peut maintenant enlever y de E et de R ; en effet, aucun autre groupe avec l'un de ces éléments ne pouvant exister, il est inutile de le considérer comme candidat. Notons que y existe nécessairement dans R, mais pas nécessairement dans E.

(c) Si le groupe trouvé est de type MSP, alors ses éléments peuvent entrer dans la composition du type de groupe non-considéré jusqu'à maintenant, un 4F-BLSR. En effet, un 4F-BLSR est formé de quatre PTPs répartis en deux sous-groupes MSP distincts. Dans ce cas, on continue la procédure ; autrement, on retourne à la prochaine itération de boucle (étape 1).

i. Le groupe MSP venant d'être trouvé est ajouté au groupe M. Le groupe M continue donc de contenir tous les groupes MSP sur l'élément géré.

ii. On vérifie que le nouveau groupe MSP, combiné à un groupe MSP de M, puisse former un nouveau groupe 4F-BLSR. Le cas échéant, ce nouveau groupe est ajouté à la cache.

7.3.2 Définition de D_a

Dans le préambule à l'algorithme, nous avons déclaré D comme étant un ensemble de propriétés appliquées à un PTP ; nous utilisons cette notation pour dénoter qu'on calcule ce D avant que la boucle de l'algorithme commence, car la boucle n'a aucun impact sur les données du PTP. Il est donc inutile de calculer ces propriétés à chaque fois.

D peut être considéré comme un tuple exposant diverses propriétés du PTP a. Les propriétés suivantes dont d'abord calculées :

- Index de la couche MS dans le template du PTP
- Index de la couche MSP NORMAL dans le template du PTP
- Index de la couche MSP EXTRA dans le template du PTP
- Capacité de faire partie d'un groupe MSP 1+1
- Capacité de faire partie d'un groupe MSP 1:1
- Capacité de faire partie d'un groupe 2F-BLSR

La capacité de faire partie d'un groupe MSP 1+1 est vraie si les conditions suivantes sont remplies :

- Présence d'une couche MS
- Absence d'une couche MSP EXTRA

La capacité de faire partie d'un groupe MSP 1:1 est vraie si les conditions suivantes sont remplies :

- Présence d'une couche MS
- Présence d'une couche MSP NORMAL ou d'une couche MSP EXTRA, mais pas des deux

La capacité de faire partie d'un groupe 2F-BLSR est vraie si les conditions suivantes sont remplies :

- Présence d'une couche MS
- Présence d'une couche MSP NORMAL
- Présence d'une adaptation indirecte AUG et de l'utilisation de la moitié des canaux MS dans la couche MSP NORMAL
- Présence d'une adaptation indirecte AUG et de l'utilisation de la moitié des canaux MS dans la couche MSP EXTRA, si celle-ci existe

7.3.3 Définition de f

Nous avons déclaré f dans l'algorithme de recherche des groupes de protection ; cette fonction est utilisée dans le coeur de la boucle de recherche.

La fonction f prend quatre paramètres : x, y, D_x et D_y, où x et y sont deux PTPs et D_x et D_y les données précalculées de ces PTPs. La fonction a pour but de déterminer si les deux PTPs forment ensemble un groupe de protection de type MSP 1+1, MSP 1:1 ou 2F-BLSR. La fonction va entamer le calcul nécessaire pour déterminer si les PTPs peuvent former un groupe de chaque type seulement si D_x et D_y indiquent que l'appartenance au type de groupe testé est possible. En séparant ainsi les calculs impliquant un seul PTP (D) et ceux impliquant deux PTP (f), on gagne du temps de recherche.

Enfin, dès qu'on trouve que x et y sont membres d'un groupe, on ne s'acharne pas à tester s'ils sont membres d'un groupe d'un autre type, étant donné qu'il est impossible qu'un PTP fasse partie de deux groupes des types recherchés par f.

La recherche d'un groupe MSP 1:1, tel que déjà mentionné, ne se fait pas en cherchant un

patron ; les paires de PTPs qui peuvent former un tel groupe sont préprogrammées statiquement.

La recherche d'un groupe MSP 1+1 ou 2F-BLSR s'effectue de façon similaire. Par conséquent, certaines parties de la recherche sont paramétrisées et pourront servir à plusieurs endroits semblables. Ce processus est expliqué à la section suivante.

7.3.4 Segmentation de la recherche

Notons tout d'abord que tous les types de groupes de protection (MSP 1+1, 2F-BLSR, 4F-BLSR) ont des paires de PTPs qui sont, à un niveau ou un autre, reliés par une règle portant sur un groupe CTP. En réalité, cela est modélisé par l'utilisation d'une règle "use ctp group rules" dans les templates. Donc, la première étape sera d'essayer de trouver une liste de paramètres d'une règle de connexion des CTPs dans le template du bon niveau. Le niveau à utiliser sera MS pour les PTPs d'un MSP 1+1, MSP NORMAL pour les PTPs normaux d'un 4F-BLSR, et MSP NORMAL dans le cas 2F-BLSR (sans trafic supplémentaire). Pour 2F-BLSR avec trafic supplémentaire, il faudra chercher d'abord MSP NORMAL, et ensuite recommencer la recherche pour MSP EXTRA.

De plus, dans les cas MSP 1+1 et 4F-BLSR, la liste de paramètres doit spécifier que les connexions de type "protected" sont permises. Trois possiblités existent pour qu'une telle condition soit remplie :

- La liste de paramètres spécifie explicitement le type "protected" pour les connexions.
- La liste de paramètres ne spécifie explicitement aucun type de connexion (cela doit être interprété comme les autorisant tous).
- Il n'y a pas de liste de paramètres pour la règle choisie ; dans ce cas, tous les paramètres possibles sont autorisés.

En résumé, cette étape itère sur les listes de paramètres applicables dans les règles de connexion applicables sur les deux PTPs, de façon à essayer toutes les combinaisons jusqu'à ce qu'un groupe de protection soit trouvé, à une exception près : si une règle dans un des deux PTPs ne spécifie pas de listes de paramètres, c'est que cette règle ne peut être contrainte par des paramètres ; c'est la règle de son type la plus permissive possible. Par conséquent, dans ce cas, si aucun groupe de protection n'a été trouvé avec cette règle, il est inutile d'essayer de chercher une combinaison avec la même règle de l'autre PTP. Nous avons codifié ce raccourci

afin d'optimiser le code.

Une fois deux listes de paramètres (une par PTP) trouvées [ou une règle appropriée sans liste de paramètres (un des PTPs) et une liste de paramètres (l'autre PTP), ou deux règles appropriées sans listes de paramètres (une par PTP)], l'algorithme de recherche se poursuit ; s'il échoue, deux autres listes de paramètres seront essayées, jusqu'à ce que toutes les combinaisons aient été essayées.

Pour donner un ordre de grandeur à cette étape, disons que la plupart des niveaux ont une ou deux règles de connexion, et que celles-ci ont chacune en moyenne une liste de paramètres. Donc généralement, en pire cas, 4 $(2 \cdot 2)$ combinaisons seront essayées.

La deuxième étape de l'algorithme sera de considérer les instances de groupe CTPs. Pour chacun des PTPs, la liste de paramètres choisie peut spécifier une liste de groupes CTPs qui sont acceptables pour la règle "use ctp group rules". Comme chaque PTP a une liste d'intances de groupe, il faudra restreindre les instances acceptables à celles du genre spécifié dans la liste de paramètres. Tout comme pour les types de connexions, trois possibilités doivent être considérées à ce niveau :

- Les groupes CTP sont spécifiés explicitement (par numéro de template) : seules les instances de ceux-là peuvent alors être utilisées.
- Aucun groupe CTP n'est spécifié explicitement : toutes les instances peuvent alors tous être utilisées.
- Il n'y avait pas de liste de paramètres pour cette règle : toutes les instances de groupe CTPs peuvent alors être utilisées.

Ainsi, nous aurons un ensemble d'instances de groupe acceptables pour chacun des deux PTPs en fonction des deux listes de paramètres choisies.

Afin de donner un ordre de grandeur à la cette étape, notons qu'en moyenne les PTPs sont membres d'une ou deux instances de groupe CTP. Pour chaque combinaison d'instances de groupes CTPs, le reste de l'algorithme est exécuté.

La troisième étape est la première qui variera significativement en fonction du type de groupe de protection recherché. À chaque invocation de cette étape, une instance spécifique de groupe CTP a été sélectionnée pour chaque PTP.

Selon le type de groupe recherché, les sous-étapes suivantes seront effectuées :

- Vérification du type du groupe CTP (tous les types de groupes)
- Vérification des points de connexion (MSP 1+1 et 4F-BLSR)
- Vérification des connexions interniveaux (MSP 1+1 seulement)

La vérification du groupe CTP commence par déterminer si les deux instances de groupes CTPs sélectionnés sont en fait les mêmes pour les deux PTPs. Cette condition doit être remplie afin de pouvoir déclarer que les deux PTPs ont au moins une instance acceptable en commun. Le cas échéant, le groupe doit posséder une règle du type recherché :

- "may connect to any ctp in group" pour MSP 1+1
- "may connect to ctp on one to one basis" pour 2F-BLSR et 4F-BLSR

La règle recherchée doit se trouver au bon niveau ; pour MSP 1+1, il s'agit du niveau MSP NORMAL. Pour les autres types de groupes de protection, elle doit se trouver à l'une des couches communes aux règles d'adaptation des CTPs des deux PTPs ; habituellement, mais pas forcément, HP VC4.

La vérification des points de connexion est assez simple ; on consulte la liste de paramètres. Pour chacun des PTPs, les trois mêmes situations que pour les types de connexions peuvent se produire (spécification explicite, pas de spécification, pas de liste de paramètres).

Dans le cas MSP 1+1, le PTP sur lequel transite normalement le trafic doit permettre l'utilisation du point A ; l'autre PTP doit permettre l'utilisation du point A'. Dans le cas 4F-BLSR, un des deux PTPs doit permettre l'utilisation du point A', et l'autre le point B.

Enfin, dans la recherche de groupes MSP 1+1, on doit faire la vérification des connexions interniveaux. Cela englobe les points suivants :

- Présence d'une règle de connexion CTP de type "must connect to" dans le template du niveau MS du PTP normal
- Vérification que cette règle autorise les connexions protégées et l'utilisation du point de connexion A

Si toutes les conditions énumérées à la troisième étape sont remplies, nous pouvons alors déclarer que nous avons un groupe de protection du type recherché et mettre fin à la recherche d'autres combinaisons de groupes.

7.4 Autres attributs

Le seul autre attribut qui puisse être obtenu à partir des informations topologiques est le `SPRINGNodeId`. Cet attribut est calculé directement dans chaque PTP, au moment de sa construction. L'attribut correspondant dans le groupe de protection est copié tel quel du premier PTP du groupe.

Le calcul au moment de la construction du PTP est assez simple : cet attribut est présent dans les templates dans la partie attribut/valeur de l'identification des niveaux. Il faut simplement trouver une valeur correspondant à l'attribut `"PortApsId"` au niveau MSP. Puisque le premier PTP du groupe est le PTP transportant normalement le trafic, un niveau MSP existe toujours ; donc si cet attribut est défini, il sera trouvé.

L'attribut SPRINGProtocol est toujours standard, puisqu'aucun des éléments gérés qui sont supportés par les MOAs que MVCI-N accepte ne fonctionnent avec le protocole "TransOceanic".

7.5 Processus de modification

Trois situations peuvent se présenter où l'information provenant de l'interface TMMOA indique un changement et doit être tenue en compte lors du calcul des groupes de protection.

L'ajout d'un nouveau PTP : Ce cas est expliqué à la section traitant de l'algorithme de recherche. On tente alors de trouver un nouveau groupe de protection dont l'un des PTPs serait le PTP nouvellement ajouté.

Modification d'un PTP : Ce cas est aussi expliqué à la section traitant de l'algorithme de recherche. On n'agit que si l'ancienne et/ou la nouvelle version possède un niveau MSP dans son template, ou encore une règle d'adaptation vers le niveau MSP dans son template de niveau MS.

Retrait d'un PTP : Ce cas est traité très facilement : on efface simplement tous les groupes qui ont pour membre le PTP retiré.

Pour effacer les groupes de protection dont un PTP fait partie, il faut d'abord les trouver. Cela se fait simplement en itérant au travers du contenu de la PGCache. Dès qu'on trouve un groupe de protection dont le PTP est membre, on peut arrêter la recherche ; un PTP ne peut

en effet, comme nous l'avons déjà mentionné, être membre que de deux groupes ; de plus, ces deux groupes ont toujours un pointeur l'un vers l'autre ; ainsi, dès qu'un de ces deux groupes est trouvé, on peut utiliser les pointeurs internes pour trouver l'autre, le cas échéant.

Finalement, notons que lorsqu'on efface un groupe MSP, il faut s'assurer que cet effacement est effectué seulement une fois que le groupe 4F-BLSR parent, s'il existe, n'existe plus : en effet, il ne faudrait pas que la cache contienne pendant un cours instant un groupe 4F-BLSR dont les groupes enfants ne sont pas accessibles : cela pourrait créer une incohérence au niveau de l'énumération des éléments demandée, par exemple, par un client externe. En fin de compte, l'effacement doit se faire dans l'ordre inverse de l'ajout (un 4F-BLSR n'est détecté par l'algorithme seulement une fois que ses sous-groupes MSP sont créés).

Chapitre 8

Conversion d'informations statutaires

Les informations en provenance de l'interface TMMOA ont permis de découvrir la présence des groupes de protection, tel que discuté au chapitre 7. Cependant, aucune information ne peut être obtenue quant à l'état de ces groupes : l'état actuel est-il normal ou commuté ? Quelle est la cause d'une commutation, le cas échéant ? Les informations provenant de l'interface TMMOA ne permettent pas de le déterminer.

En bref, la plupart des attributs des groupes de protection restent indéfinis si l'on ne se fie qu'aux renseignements fournis par l'interface TMMOA. Les informations manquantes proviennent d'une autre interface, COCGNB, qui est utilisée normalement pour transmettre les informations relatives aux fautes – ou problèmes – présents dans le réseau. L'interface COCGNB en effet, est normalement utilisée entre le collecteur d'informations de fautes (COC) et l'interface utilisateur pour afficher l'état du réseau (*Graphical Network Browser*) ; cette interface, tout comme l'interface TMMOA, est basée sur le protocole XDR ; il s'agit d'une interface client-serveur, où MVCI-N est le client.

Deux structures importantes sont définies par cette interface : les résumés d'état de protection et les plans de connectivité.

8.1 Procédures d'ingénierie inverse

Comme il a été mentionné au chapitre 6, nous avons procédé par ingénierie inverse pour découvrir le fonctionnement interne de l'interface COCGNB. Nous avions a notre disposition un fichier avec les structures de données transmises entre le client et le serveur de l'interface. Toutefois, la façon dont ces structures de données étaient employées et la valeur sémantique de celles-ci n'était pas indiquée.

La première étape réalisée à cette fin fut d'intercepter les messages en tant que client de l'interface. Comme MVCI-N avait déjà un module chargé de recevoir les messages (étant donné que l'interface COCGNB est aussi utilisée par une autre partie de MVCI-N pour recevoir les alarmes et énumérer les éléments du réseau), ce ne fut pas très compliqué. Il fallait simplement imprimer les messages qui semblaient avoir trait à la protection.

Une fois le client instrumenté de cette façon, il a fallu déterminer dans quelles circonstances ces messages étaient générés par le module COC (voir l'architecture du système, chapitre 5). Cependant, les messages de changements d'état de protection surviennent, bien entendu, lorsque l'état des PTPs faisant partie des groupes de protection change ; comme ces PTPs se trouvent sur les NEs, il fallait en conséquence faire changer l'état des ports sur les NEs pour qu'un message approprié soit généré : le NE avertit son MOA qu'un changement s'est produit ; à son tour, le MOA donne cette information au COC ; celui-ci, présumément, transfère ensuite l'information pertinente au client de l'interface COCGNB.

On serait porté à déduire du paragraphe précédent que le seul moyen de causer des messages ayant trait à la protection est d'affecter directement le NE ; comme les systèmes de protection existent pour maintenir la capacité du réseau à transporter des données lorsque son intégrité est compromise, il faudrait donc physiquement affecter un réseau test pour générer des messages de protection.

Cette solution fut d'ailleurs la première retenue : nous envisagions de débrancher des fibres optiques entre les NEs d'un réseau-test et d'observer les messages générés sur l'interface COC-GNB en réaction.

Malheureusement, nous n'avons pu exécuter ce plan : il nous a en effet été impossible d'obtenir un accès physique sur les NEs, lequel était requis pour effectuer ce genre de tests. Nous avons donc entrepris d'enquêter sur d'autres moyens de générer ces messages : nous avons réussi à

obtenir des accès à distance sur certains NEs et les MOAs qui les gèrent : chaque NE, en effet, peut être contrôlé à distance via une interface à ligne de commande. De plus, les MOAs possèdent une interface un peu plus conviviale qui permet de contrôler globalement la portion de réseau gérée par chacun d'eux.

Au lieu de causer des commutations de protection à cause d'une coupure de fibre simulée, nous nous sommes concentrés d'abord à causer des commutations manuelles : cela signifie que l'on demande au NE de faire comme s'il y avait eu un problème sur le réseau sans toutefois que celui-ci n'ait été physiquement altéré. Cette stratégie a porté fruits : nous nous sommes rendu compte que les messages qui transitaient sur l'interface COCGNB était similaires peu importe si la commutation était automatique ou manuelle.

Nous avons trouvé trois façons de causer des commutations de protection manuelles : directement via l'interface du NE, par les MOAs (via un module de contrôle de protection) et par le GNB, le client normal de l'interface COCGNB. La plupart du temps, nous avons utilisé l'interface des MOAs : c'est celle-là qui nous donnait un aperçu suffisamment précis de l'état de protection et dont la terminologie et les concepts semblaient davantage correspondre aux structures de COCGNB ; l'interface offerte par le GNB quant à elle synthétisait trop les renseignements de protection ; bien que cela soit utile dans la gestion de tous les jours d'un réseau de transport, c'est moins pratique lorsque vient le temps de faire de l'ingénierie inverse.

En causant des commutations de protection et en modifiant les autres attributs des systèmes de protection, nous avons pu empiriquement déterminer comment fonctionnait l'interface COCGNB pour les informations de protection. Nous avons même découvert comment en déduire les informations requises pour créer les liens topologiques ; cependant, nous n'en faisons pas usage dans ce travail.

8.2 Terminologie

Les informations pertinentes de l'interface COCGNB sont basées sur deux concepts :

Équipement : Un équipement est une carte dans un élément réseau ; cette carte est une unité remplaçable dans l'élément, sur laquelle se trouvent des ports, lesquels correspondent à des PTPs. Ce concept est similaire à celui d'équipement de MTNM 2.0.

Facilité : Une facilité est un concept légèrement plus abstrait. Il s'agit d'un service de communication rendu par un équipement ; plus précisément, il s'agit d'un service de transport (soit la fonction d'aggrégat dans un ADM). Puisque dans les NEs les cartes sur lesquelles les agrégats se trouvent n'ont qu'un port bidirectionnel (ou deux ports unidirectionnels), on peut associer une facilité à un PTP.

8.3 Attributs des NEs et résumés d'état de protection

Un NE comporte plusieurs attributs qui ont trait à la protection ; ces attributs ne sont pertinents que pour les facilités du NE ; autrement dit, ils ne s'appliquent pas aux groupes de protection des ports tributaires, mais seulement aux groupes de protection formés sur les aggrégats. Ces attributs, tout comme les autres propriétés d'un NE, sont transmis par le serveur de l'interface (COC) lorsqu'un nouvel NE prend part au réseau, ou sur demande du client.

Ces attributs sont :

switch mode : Cet attribut indique si la commutation est exécutée de façon bilatérale ou unilatérale. Une valeur "inconnue" existe aussi.

revertive indicator : Cet attribut indique si la commutation est réversive ou non. Une valeur "inconnue" existe aussi.

protection type : Cet attribut indique quel type de groupe de protection les facilités utilisent. Ce type peut être MSP 1+1, MSP 1:1, 2F-BLSR, 4F-BLSR (voir chapitre 3). Une valeur "inconnue" existe également.

Les résumés d'état de protection sont des structures qui indiquent l'état actuel d'une facilité. Ces résumés sont transmis par le serveur de l'interface (COC) en même temps que les autres propriétés d'un NE ou dès que se produit un changement.

Étant donné qu'elles ont trait a des facilités, les informations contenues dans un résumé d'état de protection pourraient être stockées dans les objets PTP (dans la PtpCache). Cependant, cela n'est pas possible avec l'architecture de MVCI-N, car les objets PTP ne sont pas créés au moment où l'information initiale concernant les NEs est obtenue. En effet, l'information concernant les NEs provenant de l'interface COCGNB est utilisée pour énumérer initialement les PTPs. Cependant, les informations pour créer chacun des PTPs viennent plutôt de l'interface

TMMOA. Autrement dit, on détermine partiellement qu'un PTP existe grâce aux informations provenant de COCGNB ; on détermine ensuite ses caractéristiques en interrogeant le MOA via l'interface TMMOA. Cela signifie que les PTPs ne sont pas créés tant que les informations du MOA n'auront pas été colligées.

À prime abord, on pourrait penser qu'une solution à ce problème pourrait être simplement de créer le PTP avec les informations partielles provenant du COC (les résumés de protection) pour ensuite les compléter avec les renseignements provenant du TMMOA. Cette solution a toutefois été rejetée pour plusieurs raisons. D'abord, cela impliquerait l'existence dans la cache des PTPs d'éléments incomplets. Il faudrait donc prévoir deux mécanismes d'accès à la cache : un mécanisme interne qui retourne les objets incomplets, et un mécanisme externe (par exemple, pour interrogation par un client MTNM) qui ne les retourne pas ; il aurait fallu modifier le système de notification pour ne pas avertir les entités externes de l'existence des objets incomplets ; il aurait enfin fallu prévoir un mécanisme pour s'assurer qu'éventuellement tous les objets deviennent complet. Ce mécanisme aurait été complexe étant donné la nature *multi-threads* de MVCI-N. Enfin, l'architecture actuelle impose une indexation des éléments dans les caches en fonction de leurs noms (DN) dans l'interface MTNM ; les résumés de protection en provenance de l'interface COCGNB ne permettent pas de déterminer ce nom. En bref, cette solution aurait été trop lourde.

Nous avons plutôt résolu ce problème en choisissant de garder toute l'information provenant de l'interface COCGNB en tant qu'attributs des NEs.

Un résumé d'état de protection inclut les informations suivantes :

- Identificateur de facilité unique pour le NE
- Identification textuelle du nom de l'équipement sur lequel la facilité se trouve
- Débit de l'équipement (STM-1, STM-4, etc.)
- État de commutation manuelle pour le segment
- État de commutation manuelle
- État de commutation automatique pour le segment
- État de commutation automatique
- État de commutation forcée pour le segment
- État de commutation forcée
- État de verrou de commutation pour le segment

- État de verrou de commutation
- État de verrou global de commutation de protection

Ce ne sont pas tous les champs qui sont significatifs pour tous les types de groupes de protection ; d'ailleurs, une des difficultés rencontrées lors de l'étape d'ingénierie inverse fut de déterminer lesquels ont une valeur et dans quelles circonstances. Nous voyons qu'il y a deux séries de champs : les champs généraux et les champs spécifiques aux informations de segment. Les champs spécifiques aux informations de segment ne sont valides que pour les groupes 4F-BLSR.

- Pour les groupes MSP 1+1 et MSP 1:1, les champs ayant trait aux commutations ou au verrouillage pour le segment n'ont pas de signification : la commutation entre les deux NEs participants (au niveau MS) est rapportée dans les champs généraux.
- Pour les groupes 2F-BLSR, les champs ayant trait aux commutations ou au verrouillage pour le segment n'ont pas de signification : le seul genre de commutation qui existe sont des commutations d'anneau, et ce genre est rapporté dans les champs généraux.
- Pour les groupes 4F-BLSR, tous les champs ont une signification. Les commutations d'anneau sont rapportées dans les champs généraux, tandis que les commutations de segment sont rapportées dans les champs ayant trait aux segments.

Pour chacun des éléments dans la liste des composantes d'un résumé de protection, deux informations sont rapportées : la localisation et l'état de commutation.

La localisation indique si le verrou ou la commutation a été fait à cause d'une requête locale (sur le même NE) ou bien à cause d'une requête distante (sur un autre NE).

L'état de verrouillage ou de commutation indique si l'état est normal, s'il y a commutation ou verrou actif, s'il y a désir de commutation ou de verrouillage mais que la condition est bloquée face à une situation prioritaire (état *pending*), ou s'il y a commutation due au délai (*wait-to-restore time*) avant le retour à la normale.

On peut constater que l'état de connectivité est rapporté pour chacune des facilités ; nous devrons donc recouper cette information pour obtenir l'état global d'un groupe de protection. Cela est expliqué en détail à la section 8.5.

Grace aux informations rapportées, nous pouvons donc déterminer quel est le PTP actif à l'intérieur du groupe (`switchAwayFromTP` et `switchToTP` dans les données de `SwitchData_T`) ; voir à ce sujet le chapitre 9.

8.4 Plans de connectivité

Les plans de connectivité constituent la deuxième structure d'importance pour la protection dans l'interface COCGNB. Ces plans indiquent comment les NEs sont connectés entre eux ; toutefois, cette information en tant que tel n'a aucune utilité dans cette recherche. Ce qui nous intéresse, c'est simplement un des attributs que ces plans contiennent pour chacune des facilités faisant partie d'une connexion : son wtrTime.

Ainsi, bien qu'ils soient utiles pour découvrir les liens topologiques, ces plans n'ont qu'un intérêt accessoire dans dans le cadre de la protection ; on pourrait même dire que cet intérêt est dû à une incohérence de l'interface.

Quoi qu'il en soit, les plans de connectivité sont constitués d'une liste d'éléments du réseau. Les cinq champs qui nous intéressent dans un élément de plan de connectivité sont les suivants :

span id : numéro du MOA qui gère l'élément réseau

ne id : numéro attribué par ce MOA à l'élément réseau

prev cpg : nom de l'équipement utilisé pour relier cet élément réseau à l'élément mentionné juste avant dans le plan

next cpg : nom de l'équipement utilisé pour relier cet élément réseau à l'élément mentionné juste après dans le plan

wtr time : temps d'attente avant le rétablissement de l'état normal

Le temps d'attente est associé aux deux facilités utilisant les équipements mentionnés dans le plan. Les facilités auxquelles on n'aura pu associer aucun wtrTime (parce qu'elles utilisent des équipements qui ne sont pas mentionnés dans aucun plan de connectivité) ont un wtrTime de -1, valeur signifiant "inconnu".

Les plans de connectivité sont identifiées par un numéro unique.

8.5 Gestion des modifications

Les plans de connectivité peuvent être modifiés par des actions de gestion de réseau visant à changer la configuration des éléments. De plus, ils peuvent aussi être retirés complètement

(et dans ce cas, seul leur numéro unique est mentionné). À cause de cela, les plans doivent être gardés en mémoire au niveau de MVCI-N : en effet, si un plan venait à être retiré, il faudrait que les wtrTime des facilités qui en faisaient partie retournent à la valeur -1, ce qui ne peut se faire que si nous avons stocké une liste des équipements faisant partie du plan.

De même, lorsqu'un plan est modifié, il est possible que certains éléments ne fassent plus partie de la nouvelle version ; il faut aussi remettre à -1 les wtrTime des facilités affectées.

L'algorithme de modification que nous avons conçu est, somme toute, très simple. Soit P l'ensemble des facilités qui utilisent des équipements dans l'ancienne version du plan ; Q l'ensemble des facilités dans la nouvelle version ; soit N un ensemble d'éléments réseau initialement vide.

1. Mettre une copie des éléments réseau sur lesquels les membres de P se trouvent dans N.
2. Dans les copies des éléments réseau (dans N), remettre tous les wtrTime des facilités membres de P à -1.
3. Ajouter une copie des éléments réseau sur lesquels les membres de Q se trouvent à N, s'ils ne sont pas déjà dans N.
4. Dans les copies des éléments réseau (dans N), changer les wtrTime des éléments de Q à la valeur indiquée dans le nouveau plan.
5. Substituer en mémoire l'ancien plan pour la nouveau plan.
6. Remplacer les anciennes versions des éléments réseau (stockés dans la cache) par ceux de l'ensemble N. $\diamondsuit$

Cette procédure est assez générale pour pouvoir traiter de façon similaire l'ajout, la modification et le retrait d'un plan : dans le cas d'un ajout, P est vide ; dans le cas d'un effacement, Q est vide. Par ailleurs, le fait de grouper dans un ensemble les objets représentant les éléments réseau et d'incorporer les changements agglomérés seulement à la fin réduit le nombre de modifications à effectuer dans la cache, ce qui est économique en temps de traitement. En effet, les caches sont en lecture seule : à chaque fois qu'on désire modifier un objet, on doit invoquer une méthode spéciale qui va s'occuper de générer les notifications de changement (voir la section 5.3.1). Cette méthode va remplacer l'objet dans la cache avec une nouvelle version. Notre méthode évite donc d'invoquer inutilement ce processus.

8.6 Appariement des informations

Puisque les groupes de protection sont stockés dans la PGCache mais que les facilités sont des attributs des ManagedElements, lesquels sont stockés dans la ManElCache, il est nécessaire de prévoir un mécanisme pour importer les renseignements appropriés dans les groupes de protection. Ce mécanisme doit tenir compte du fait que les PTPs et les groupes de protection sont créés après les ManagedElements (raison pour laquelle on ne peut pas stocker les attributs des facilités dans les PTPs eux-mêmes).

Le mécanisme que nous avons retenu fonctionne à l'inverse. Lorsqu'un nouveau groupe de protection est créé, on vérifie si ses PTPs sont des aggrégats. On effectue cette vérification en essayant de trouver une facilité dont le nom correspond à celui du PTP.

Cette vérification s'effectue de la manière suivante : à partir des caractéristiques du PTP, on peut déduire quel est le débit de l'équipement sur lequel il se trouve et quel est le nom textuel de cet équipement. Nous essayons de trouver dans la liste des résumés de protection du NE sur lequel se trouve le PTP un élément ayant les caractéristiques correspondantes. (Nous pouvons dire qu'il est assez fortuit que nous puissions déduire des caractéristiques du PTP ces informations et que celles-ci puissent identifier une facilité de façon unique ; si cela n'avait pas été le cas, une autre méthode aurait dû être trouvée.)

Cette identification de nom d'équipement à partir des informations du PTP n'est pas entièrement triviale : en effet, nous avons, grâce aux tests effectués durant l'analyse, rencontré des PTPs qui rapportaient plus d'un nom d'équipement ; en fait, dans ce cas, un groupe de PTPs rapportent tous le même nom d'équipement. Il n'y a pas d'information qui nous indique à quel équipement quel PTP se rapporte précisément.

Qui plus est, ce cas n'est pas exceptionnel : souvent la situation survient lorsque deux PTPs sont configurés dans un groupe MSP. La solution que nous avons choisie consiste à utiliser un autre attribut du PTP, celui-ci toujours unique, la *localisation universelle*. L'algorithme obtient de la cache des PTPs tous ceux sur le même élément géré qui ont la même liste d'équipements que celui dont on cherche le nom d'équipement. Cette liste de PTP est triée en fonction de la localisation universelle ; l'index dans cette liste triée du PTP dont on cherche dans le nom d'équipement sert à indiquer quel nom d'équipement utiliser dans la liste d'équipements que le PTP déclare.

Cette solution se fonde sur l'hypothèse que la liste des noms d'équipements partagée par le nom du PTP est triée dans l'ordre des localisations universelles ; les tests que nous avons fait semblent confirmer cette hypothèse, bien qu'aucune documentation officielle puisse le confirmer hors de tout doute. Pour cette raison, nous avons ajouté dans le code des instructions pour vérifier la cohérence des données, lorsque possible. Par exemple, une des assertions que l'on peut faire est que la liste d'équipements a la même longueur que le nombre de PTPs qui la partagent. Dans le cas contraire, la méthode échoue et un message diagnostic est écrit dans un fichier journal.

Remarquons que la solution possède aussi le désavantage de demander des recherches dans la cache des PTPs, procédure assez coûteuse. Nous avons donc ajouté une heuristique pour guider la recherche : à force d'expérimenter, nous avons remarqué, tel que mentionné ci-dessus, que ce cas de partage de liste d'équipements survenait surtout pour les groupes MSP (qui ont deux éléments). Au moment de vérifier si les PTPs sont associés à des facilités, on connaît quels PTPs font partie du groupe de protection. On fait donc cette vérification par paire ; dans le cas où le premier PTP spécifie une liste d'équipement, on vérifie immédiatement, sans passer par la cache, si l'autre PTP du groupe MSP contient la même liste d'équipements. Le cas échéant, on peut considérer que les équipements dans la liste partagée sont ceux employés pour les PTP du groupe MSP sans avoir eu à chercher dans la cache ; on assigne simplement un équipement par PTP.

Si on trouve des facilités pour chacun des PTPs membres du groupe de protection, on importe les attributs représentant le `switchMode`, `reversionMode` à partir des attributs du NE, on calcule s'il y a une commutation en cours (attributs `switchToTP` et `switchAwayFromTP`) et enfin si la commutation est verrouillée ou non (`protectionSchemeState`).

Le calcul de commutation est effectué grâce aux informations provenant des résumés de protection. Ces informations indiquent pour chacun des PTPs participant à un groupe de protection quel est son état ; il faut donc regrouper ces informations pour donner l'état global du groupe. Nous avons réussi à condenser cet état en une seule variable qui garde en mémoire l'état de commutation d'un groupe de protection. Le champ `activeTP` de l'objet ProtectionGroup gardé dans la cache indique quel PTP transporte le trafic selon l'analyse effectuée. La façon précise de faire celle-ci dépend du type de groupe de protection.

Pour les groupes MSP, la première chose à faire est de déterminer si les agrégats du NE sont

configurés en tant que groupes 4F-BLSR. Le cas échéant, les données devront être prises de la série de champs de segments. Sinon, la série de champs régulière devra être utilisée (car la série de champs de segments contient dans ce cas des données indéfinies, voir la section 8.3).

On considère qu'il y a une commutation si le résumé du PTP normal (celui qui transporte normalement le trafic) indique qu'une commutation automatique, manuelle ou forcée a lieu (dans l'état "completed" ou "wait to restore"). Le champt `activeTP` indique quel PTP transporte le trafic. S'il n'y a pas de commutation, ce sera le premier PTP (le PTP normal) ; s'il y a une commutation, ce sera l'autre.

Pour les groupes 2F et 4F-BLSR, on doit vérifier sur les PTPs de chaque côté pour une commutation d'anneau (on utilise les PTPs normaux pour les groupes 4F-BLSR). On considère qu'il y a une commutation si un des deux résumés indique dans ses champs généraux qu'une commutation automatique, manuelle ou forcée a lieu. Le champ `activeTP` de l'object ProtectionGroup indique s'il y a une commutation d'anneau d'après cette analyse. S'il n'y a pas de commutation, on met le champ à 0 (nul). S'il y a une commutation d'anneau, le champ pointe vers le PTP du côté opposé à la coupure de fibres (dans le cas des anneaux 4F-BLSR, on utilise les PTPs de protection).

Nous pouvons aussi déterminer quelle est, le cas échéant, la cause d'une commutation :

- S'il n'y a pas de commutation, `switchReason` est `SR_NA`.
- S'il y a une commutation automatique, `switchReason` est `SR_SIGNAL_FAIL` (il n'y a pas moyen par l'interface COCGNB de trouver une cause plus précise telle que `SR_SIGNAL_MISMATCH`).
- S'il y a une commutation forcée ou manuelle, `switchReason` est `SR_MANUAL`.

Enfin, l'attribut `protectionSchemeState`, lequel indique s'il y a un verrouillage de protection, est calculé de la manière suivante :

Pour un groupe MSP on vérifie si une commutation forcée ou un verrouillage de protection simple est enclenché sur l'un des deux PTPs (les verrouillages de protection globaux n'existent pas pour les groupes MSP). S'il y a un verrouillage ou une commutation forcée, l'état sera `PSS_FROZEN` ; sinon (pour le cas normal), il sera `PSS_AUTOMATIC`.

Pour un groupe 2F-BLSR, on vérifie si une commutation forcée ou un verrouillage de protection global est enclenché sur l'un des deux PTPs ; le cas échéant, l'état sera `PSS_FROZEN`. S'il y

a un verrouillage de commutation simple (non global) sur les deux PTPs, l'état sera également `PSS_FROZEN`. Par contre, s'il y a un verrouillage de commutation simple sur un seul PTP, cela signifie qu'il peut toujours y avoir des commutations sur l'autre PTP ; l'état sera alors NA (pas totalement `PSS_FROZEN`, pas totalement `PSS_AUTOMATIC`).

Pour un groupe 4F-BLSR, on utilise les mêmes critères que pour un groupe 2F-BLSR ; les verrouillages de commutation simples sont rapportés sur les PTPs normaux de chaque côté, tandis que les verrouillages de commutation globaux sont rapportés sur les PTPs de protection.

Lorsqu'un attribut ou une facilité change, le mécanisme de notification est utilisé pour avertir la PGCache. Pour les changements simples comme une modification du `reversionMode` ou du `switchMode`, on n'a qu'à vérifier (de la façon exposée au paragraphe précédent) lesquels des groupes de protection sont touchés puis à les modifier. Pour les changements plus complexes, on recalcule tous les groupes de protection sur le ManagedElement qui a changé.

Chapitre 9

Gestion de l'interface externe

Les clients qui désirent communiquer avec MVCI-N doivent passer par le bus de communications CORBA, tel que discuté au chapitre 5. Dans ce chapitre, nous discutons des étapes nécessaires à l'intégration des systèmes de bas niveau (caches de MVCI-N) avec les modules chargés de fournir l'interface CORBA pour les clients utilisateurs de MVCI-N.

9.1 Gestionnaire de protection CORBA

Comme nous l'avons mentionné précédemment, le ProtectionMgr_I est chargé de fournir aux client les opérations ayant trait à la protection.

Le gestionnaire de protection en tant que tel est un objet CORBA (avec une interface IDL) qui ne sert que de façade ; il n'a aucune fonctionnalité en soi excepté celle de fournir une interface constituée des méthodes énumérées précédemment. Il est construit au moment de l'initialisation de MVCI-N, et une référence vers l'objet est gardée par le système de gestion de session.

MVCI-N supporte trois opérations via le ProtectionMgr_I : obtenir des structures CORBA représentant tous les ProtectionGroup sous un ManagedElement, obtenir un ProtectionGroup en particulier et obtenir les données de commutation d'un ProtectionGroup en particulier.

Ces trois opérations sont assez simples à implémenter étant donné les fonctions offertes par la PGCache. Celle-ci permet en effet d'obtenir tous les objets dont le Distinguished Name com-

mence par un ManagedElement en particulier (ce qui permet d'implanter la première opération), et aussi de trouver un objet particulier par son nom (ce qui, combiné aux deux méthodes de conversion des ProtectionGroup, permet l'implémentation des deux dernières opérations).

Les deux dernières opérations retournent de l'information concernant un groupe de protection en particulier. Ces fonctions s'appuient sur deux méthodes des objets ProtectionGroup, soient `ConvertToIdl` et `ConvertSwitchDataToIdl`. La première retourne une structure CORBA représentant un ProtectionGroup CORBA ; la seconde, une ou plusieurs structures pour les informations de commutation d'un groupe de protection.

La structure retournée par `ConvertToIdl` est un ProtectionGroup_T ; cette structure est présentée au chapitre 4. Cette structure contient des renseignements essentiellement statiques : ceux-ci ne changent pas avec le temps à moins qu'ait eu lieu une reconfiguration des PTPs du groupe de protection.

La ou les structures retournées par `ConvertSwitchDataToIdl` sont des SwitchData_T ; cette structure est également présentée au chapitre 4. Contrairement aux ProtectionGroup_T, elle contient des informations qui varient avec le temps ; chaque fois qu'une commutation de protection survient, les données de cette structure sont sujettes à changement. Dépendemment du type de groupe de protection, une ou deux structures sont retournées. Notons que comme c'est l'information courante qui est retournée dans ces structures, `switchAwayFromTP` est toujours nul et `switchToTP` indique quel TP transporte les données en ce moment. Les valeurs de `protectedTP` et `switchToTP` sont déterminées selon l'algorithme suivant :

Pour un groupe MSP 1+1 ou 1:1, la liste des PTPs (`pgpTPList`) contient toujours deux membres : le premier est le PTP normal (protégé), le second le PTP de protection (qui transporte le traffic supplémentaire pour MSP 1:1). L'attribut `activeTP` indique duquel des deux PTPs du groupe proviennent les données à traiter. Par conséquent :

- `protectedTP` porte toujours le nom du premier PTP dans la liste `pgpTPList`.
- `switchToTP` porte le nom du PTP pointé par `activeTP`.

Pour un groupe 2F-BLSR, deux structures sont retournées ; une pour chaque côté de l'anneau ; ces côtés sont appelés Est et Ouest par convention. Le champ interne `activeTP` contient habituellement, lorsqu'il n'y a pas de commutation qui affecte le groupe, la valeur 0 (nul). S'il y a une commutation, le champ pointe vers le PTP du côté sain, c'est-à-dire le côté opposé à

celui où se situe la coupure (Figure 3.15). Voici ce que contient la structure du côté Est. (La structure du côté Ouest contient des informations similaires ; il suffit d'interchanger les termes.)

- `protectedTP` porte toujours le nom du PTP du côté Est.
- `switchToTP` porte le nom du PTP du côté Est s'il n'y a pas de commutation en cours (si `activeTP` vaut 0) ou le nom du PTP sain (vers lequel pointe `activeTP`) s'il y a une commutation.

Pour les groupes 4F-BLSR, le fonctionnement de la méthode est semblable au cas 2F-BLSR. Deux structures sont retournées, une par côté. La seule différence réside en le fait qu'il y a deux PTPs par côté ; pour le côté Est, `switchToTP` porte le nom du PTP normal du côté Est s'il n'y a pas de commutation d'anneau. S'il y a une commutation d'anneau, `activeTP` et `switchToTP` portent le nom du PTP de protection du côté opposé à la coupure (Figure 3.18).

9.2 Notifications CORBA

Un système de notification basé sur Observer/Observable est disponible dans MVCI-N ; ce système est en fait celui de MTNM 1.0 (TMF509). La norme MTNM 2.0 utilise plutôt le service de notification CORBA construit par l'OMG (Object Management Group).

Le système de notification est basé sur un Event Manager ; celui-ci reçoit un avertissement dès qu'un objet géré par l'une des caches y est ajouté, modifié ou effacé. Dans le cas d'un objet créé, le nouvel objet est transmis au Event Manager. Pour un objet effacé, la dernière version avant l'effacement est transmise. Pour un objet qui subit une modification, les versions avant et après la modification lui sont transmises, ce qui permet aux procédures chargées de générer les notifications de comparer l'ancien et le nouvel états.

Si un nouvel objet est créé, une notification de type Object Creation est émise. Si un objet est effacé, ce sera alors une notification de type Object Deletion qui sera envoyée. Si un objet est modifié, la situation est alors plus compliquée. Dans le cas qui nous intéresse, soit celui des groupes de protection, deux cas peuvent se produire.

Dans le premier cas, le changement peut signaler une commutation (ou un retour à la situation normale). Dans ce cas, une notification de type Protection Switch doit être émise. Une telle notification inclut les mêmes renseignements que ceux dans la structure `SwitchData_T` (voir cha-

pitre 4) ; s'y retrouve aussi l'heure où l'événement s'est produit. Une commutation est détectée lorsque l'object ProtectionGroup gardé dans la cache voit son *TP actif* être modifié.

Les valeurs des champs `protectedTP`, `switchAwayFromTp` et `switchToTP` sont déterminées en fonction d'un algorithme (semblable à celui pour `retrieveSwitchData`) qui se base sur la valeur du TP actif (`activeTP`).

Pour un groupe MSP 1+1 ou 1:1, la façon de calculer les valeurs `protectedTP` et `switchToTP` est identique : il suffit d'appliquer la méthode donnée pour `retrieveSwitchData` avec le nouvel état du groupe de protection. L'attribut `switchAwayFromTP` quant à lui est le nom du PTP vers lequel pointe `activeTP` dans l'ancien état du groupe de protection. Ce sera nécessairement le PTP qui n'est pas pointé par `activeTP` dans le nouvel état du groupe de protection.

Pour un groupe 2F-BLSR, on observe la valeur de `activeTP` dans la nouvelle version du groupe de protection. Si la valeur est différente de 0, cela signifie qu'il y a commutation en cours (sinon, c'est un retour à l'état normal). S'il y a commutation en cours :

- S'il y a commutation, `protectedTP` est le nom du PTP qui est du côté de la coupure, soit celui vers lequel `activeTP` ne pointe pas.
- `switchToTP` est le nom du PTP vers lequel `activeTP` pointe, soit le PTP du côté opposé à la coupure.
- `switchAwayFromTP` est le même PTP que `protectedTP`, donc le nom du PTP vers lequel `activeTP` ne pointe pas.

S'il n'y a plus de commutation :

- `protectedTP` est le nom du PTP qui état du côté de la coupure.
- `switchToTP` est le même PTP que `protectedTP`, soit celui du côté où il y avait une coupure.
- `switchAwayFromTP` est le nom du PTP vers lequel `activeTP` pointait, soit celui du côté opposé à la coupure.

Les groupes 4F-BLSR sont traités similairement aux groupes 2F-BLSR. La différence réside dans le fait qu'il y a quatre PTPs. Les mêmes règles sur les côtés s'appliquent avec les précisions suivantes :

- `protectedTP` est toujours le PTP normal sur le côté mentionné.
- `activeTP` est soit 0 (s'il n'y a pas de commutation), soit un des PTPs de protection (s'il

y a commutation).

- Par conséquent, lorsqu'une commutation survient, `switchToTP` pointe vers un PTP de protection (celui opposé au côté où la coupure est survenue), tandis que `switchAwayFromTP` pointe vers un PTP normal (celui du côté opposé où la coupure est survenue).
- Lorsque la situation revient à la normale, `switchAwayFromTP` pointe vers un PTP de protection (celui opposé au côté où la coupure est survenue), tandis que `switchToTP` pointe vers un PTP normal (celui du côté où la coupure est survenue).

Le deuxième cas possible lorsque survient un changement dans un groupe de protection est celui où il s'agit d'un simple AVC (Attribute Value Change). Cela signifie qu'un des attributs du ProtectionGroup a changé sans toutefois que l'état de commutation soit différent.

Les notifications émises sont transmises au gestionnaire de notifications ; celui-ci vérifie quels clients enregistrés à l'interface Observable sont intéressés à quelles notifications (en fonction de filtres demandés par les clients). Les notifications ne sont envoyées qu'aux client intéressés. Par exemple, certains filtres spécifient que seuls les notifications de commutations ayant trait à des PTPs de types précis devraient parvenir au client ; il font donc que le gestionnaire de notification analyse chacun des messages pour s'assurer que la condition est remplie.

Chapitre 10

Conclusion

L'objectif de la recherche était d'ajouter une interface à un système de gestion de réseau afin que celui-ci puisse être compatible à la norme MTNM 2.0. Pour ce faire, nous avons participé à l'implantation d'une composante fonctionnelle qui, en quelque sorte, "traduisait" les informations en provenance des interfaces existantes vers l'interface conforme à MTNM 2.0.

Notre travail consistait à examiner les informations ayant trait à la protection du réseau. Le travail effectué comporte trois parties :

- Implémenter de façon efficace la recherche de patrons permettant d'identifier des groupes de protection (modèle TMF) à partir des règles décrivant les capacités des PTPs et CTPs (modèle ITU).
- Analyser une interface non documentée décrivant l'état des PTPs et trouver comment y extraire l'information permettant de déterminer les attributs et l'état des groupes de protection ; implémenter la méthode trouvée.
- Faire le design et guider l'implémentation des méthodes d'accès ayant trait à la protection dans l'interface MTNM.

La première partie a été réalisée avec succès ; cependant, la quantité de travail requise a été plus grande que prévu, en raison des structures de données passablement compliquées requises pour modéliser les capacités des points de terminaison (ces structures doivent être assez flexibles pour modéliser une grande variété de points de terminaison ; en conséquence, elles comportent une grande quantité de règles, de listes de possibilités et de sous-structures imbriquées).

La deuxième partie a également été réalisée avec succès ; cette partie a été compliquée par le manque d'équipement servant à tester la façon dont l'interface était utilisée. Néanmoins, nous avons pu quand même déchiffrer son fonctionnement. Malheureusement, nous avons par le fait même découvert que cette interface donnait moins de renseignements que nous l'escomptions : elle ne fonctionne qu'avec certains MOAs et ne donne aucun renseignement sur les tributaires.

La troisième et dernière partie est la moins importante et la moins complexe ; elle fut réalisée sans aucune difficulté.

En somme, le travail accompli dans le cadre de cette maîtrise a bien répondu aux objectifs visés. Nous constatons que le modèle utilisé pour représenter les PTPs, soit les templates, a une bonne base descriptive fondée dans les normes de modélisation des réseaux de transport. Toutefois, il ne semble pas y avoir de base théorique sur la partie associative, c'est-à-dire sur les règles qui permettent de connecter les CTPs et PTPs entre eux, et d'attribuer une sémantique à ces connexions.

Le travail que nous avons fait pourra servir de base à d'autre travaux. Par exemple, l'investigation que nous avons entreprise pour découvrir le fonctionnement de l'interface décrivant l'état des PTPs nous a permis de constater que l'interface est également appropriée pour d'autres informations exportées par MTNM 2.0, nommément les liens topologiques.

Également, certaines améliorations pourraient être faites dans le système de notification : en ce moment, le système plutôt *ad hoc* de MTNM 1.0 est employé. Le système utilisé dans MTNM 2.0 est plus général et est fondé sur une base plus reconnue.

Enfin, dans le cadre de ce travail, nous nous sommes limité aux réseaux SDH ; nous estimons que la prochaine étape dans l'évolution de la gestion des réseaux de transport sera effectuée au niveau photonique ; il sera intéressant d'améliorer ce travail pour qu'il fonctionne avec cette classe de réseaux et les concepts de protection qui leur sont rattachés.

Bibliographie

[All1973] Allan, W. B. *Fibre Optics : Theory and Practice.* London : Plenum, 1973.

[Bel1881] Bell, Alexander Graham and Summer Tainter. *Photophonic Receiver.* United States Patent No 241909, 1881.

[Cun1991] Cunningham, Donna C. *Photonics.* AT&T Bell Laboratories, 1991.

[Fre1996] Freer, John. *Computer Communications and Networks.* Second Edition. London : UCL Press, 1996.

[Haj2000] Hajbandeh, Roohollah. "T1, T3, and SONET Networks" in *Broadband Communications.* Boca Raton : CRC Press LLC (Auerbach), 2000.

[Hal1994] Halsall, Fred. *Data Communications, Computer Networks and Open Systems.* Fourth Edition. Readings, MA : Addison-Wesley, 1995.

[IEEE2000] IEEE 802.3, 2000 Edition, Gigabit Ethernet in *IEEE Standard for Information technology–Telecommunications and information exchange between systems–Local and metropolitan area networks–Specific requirements–Part 3 : Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications.*

[ISO14882] ISO/IEC 14882 :1998. *Programming languages – C++.*

[ITUg707] ITU-T Recommendation G.707 (1996), *Network node interface for the synchronous digital hierarchy (SDH).*

[ITUg774] ITU-T Recommendation G.774 (2001), *Synchronous digital hierarchy (SDH) management information model for the network element view.* Pre-published edition.

[ITUg783] ITU-T Recommendation G.783 (2000), *Characteristics of synchronous digital hierarchy (SDH) equipment functional blocks.* Pre-published edition.

[ITUg783c] ITU-T Recommendation G.783 Corr.1 (2001), *Corrigendum 1 to Recommendation G.783.* Pre-published edition.

[ITUg803] ITU-T Recommendation G.803 (1997), *Architecture of transport networks based on the synchronous digital hierarchy (SDH).*

[ITUg805] ITU-T Recommendation G.805 (1995), *Generic Functional Architecture of Transport Networks.*

[ITUg841] ITU-T Recommendation G.841 (1998), *Types and characteristics of SDH network protection architectures.*

[ITUi120] ITU-T Recommendation I.120 (1993), *Integrated services digital networks (ISDNs).*

[ITUi326] ITU-T Recommendation I.326 (1995), *Functional architecture of transport networks based on ATM.*

[ITUm3010] ITU-T Recommendation M.3010 (2000), *Principles for a telecommunications management network.*

[Kap1967] Kapany, N. S. *Fiber Optics : Principles and Applications.* New York : Academic Press, 1967.

[MTNM1] TeleManagement Forum Publication TMF509. *NML-EML Interface Business Aggrement for Management of SONET/SDH Transport Networks.*

[MTNM2] TeleManagement Forum Publication TMF814. *Multi Technology Network Management Solution Set - IDL Version 2.0.*

[OMG1998] Object Management Group. *The Common Object Request Broker Architecture and Specification.* Revision 2.3, 1998.

[Pic1981] Pickens, R. Andrew. "Wideband Transmission Media III : Guided Transmission : Wireline, Coaxial Cable, and Fiber Optics" in *Computer Communications.* Englewood Cliffs (NJ) : Prentice Hall, 1983.

[Pre1997] Pressman, Roger S. *Software Engineering : A Practitioner's Approach.* Fourth Edition. New-York : McGraw-Hill, 1997.

[Ren1993] Rens, Jean-Guy. *L'empire invisible : histoire des télécommunications au Canada – De 1956 à nos jours.* Sainte-Foy : Presses de l'Université du Québec, 1993.

[Sat1996] Sato, Ken-Ichi. *Advances in Transport Network Technologies : Photonic Networks, ATM, and SDH.* Norwood, MA : Artech House, 1996.

[Sta1988] Stallings, William. *Data and Computer Communications.* Second Edition. New York : Macmillan, 1988.

[Str2000] Stroustrup, Bjarne. *The C++ Programming Language.* Special Edition. Readings, MA : Addison-Wesley, 2000.

[Sun1995] IETF Network Working Group Request for Comments 1832 (1995). Srinivasan, R. (Sun Microsystems). *XDR : External Data Representation Standard.*

Remerciements

Je tiens d'abord à remercier mes parents, Jacinthe et Robert. Ils ont toujours été présents et attentifs, non seulement au cours de ma maîtrise, mais aussi tout au long de mes études. Ils ont su conserver leur patience maintes fois éprouvée au cours de cette période.

Je veux également remercier mes directrices de recherche, Esma Aïmeur et Rachida Dssouli, qui m'ont appuyé au cours de mes démarches et ont toujours su exiger un travail de qualité de ma part.

Je désire remercier mes superviseurs, François Bédard et David Ardman, qui ont su me donner la flexibilité et la liberté d'action nécessaires pour faire un travail pertinent à tous points de vue.

Enfin, je désire saluer mes camarades de classe, particulièrement Étienne Bergeron et Jean-François Gagné, lesquels ont partagé stress, travaux, consommations alcoolisées et bons moments au cours de cette période (pas nécessairement dans cet ordre).

Ce travail a été rendu possible en outre grâce au support financier du Conseil de recherches en sciences naturelles et en génie du Canada (CRSNG).