



HAL
open science

Protocole de diffusion fiable pour réseaux locaux à diffusion

Amar Khider

► **To cite this version:**

Amar Khider. Protocole de diffusion fiable pour réseaux locaux à diffusion. Réseaux et télécommunications [cs.NI]. Institut National Polytechnique de Grenoble - INPG, 1983. Français. NNT : . tel-00010662

HAL Id: tel-00010662

<https://theses.hal.science/tel-00010662>

Submitted on 17 Oct 2005

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

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

THESE

présentée à

l'Institut National Polytechnique de Grenoble

pour obtenir le grade de

DOCTEUR INGENIEUR

Informatique

par

Amar KHIDER



**PROCOLE DE DIFFUSION FIABLE POUR
RESEAUX LOCAUX A DIFFUSION.**



Thèse soutenue le 19 mai devant la commission d'examen:

S. KRAKOWIAK	: Président
V. QUINT	} Examineurs
J. MOSSIERE	
G. VANDOME	



INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE

Année universitaire 1979-1980

Président : M. Philippe TRAYNARD
Vice-Présidents : M. Georges LESPINARD
M. René PAUTHENET

PROFESSEURS DES UNIVERSITES

MM.	ANCEAU François	Informatique fondamentale et appliquée
	BENOIT Jean	Radioélectricité
	BESSON Jean	Chimie Minérale
	BLIMAN Samuel	Electronique
	BLOCH Daniel	Physique du Solide - Cristallographie
	BOIS Philippe	Mécanique
	BONNETAIN Lucien	Génie Chimique
	BONNIER Etienne	Métallurgie
	BOUVARD Maurice	Génie Mécanique
	BRISSONNEAU Pierre	Physique des Matériaux
	BUYLE-BODIN Maurice	Electronique
	CHARTIER Germain	Electronique
	CHERADAME Hervé	Chimie Physique Macromoléculaires
Mme	CHERUY Arlette	Automatique
MM.	CHIAVERINA Jean	Biologie, Biochimie, Agronomie
	COHEN Joseph	Electronique
	COUMES André	Electronique
	DURAND Francis	Métallurgie
	DURAND Jean-Louis	Physique Nucléaire et Corpusculaire
	FELICI Noël	Electrotechnique
	FOULARD Claude	Automatique
	GUYOT Pierre	Métallurgie Physique
	IVANES Marcel	Electrotechnique
	JOUBERT Jean-Claude	Physique du Solide - Cristallographie
	LACOUME Jean-Louis	Géographie - Traitement du Signal
	LANCIA Roland	Electronique - Automatique
	LESIEUR Marcel	Mécanique
	LESPINARD Georges	Mécanique
	LONGEQUEUE Jean-Pierre	Physique Nucléaire Corpusculaire
	MOREAU René	Mécanique
	MORET Roger	Physique Nucléaire Corpusculaire
	PARIAUD Jean-Charles	Chimie - Physique
	PAUTHENET René	Physique du Solide - Cristallographie
	PERRET René	Automatique

.../...

MM.	PERRET Robert	Electrotechnique
	PIAU Jean-Michel	Mécanique
	PIERRARD Jean-Marie	Mécanique
	POLOUJADOFF Michel	Electrotechnique
	POUPOT Christian	Electronique - Automatique
	RAMEAU Jean-Jacques	Chimie
	ROBERT André	Chimie Appliquée et des matériaux
	ROBERT François	Analyse numérique
	SABONNADIÈRE Jean-Claude	Electrotechnique
Mme	SAUCIER Gabrielle	Informatique fondamentale et appliquée
M.	SOHM Jean-Claude	Chimie - Physique
Mme	SCHLENKER Claire	Physique du Solide - Cristallographie
MM.	TRAYNARD Philippe	Chimie - Physique
	VEILLON Gérard	Informatique fondamentale et appliquée
	ZADWORNY François	Electronique

CHERCHEURS DU C.N.R.S. (Directeur et Maître de Recherche)

M.	FRUCHART Robert	Directeur de Recherche
MM.	ANSARA Ibrahim	Maître de Recherche
	BRONOEL Guy	Maître de Recherche
	CARRE René	Maître de Recherche
	DAVID René	Maître de Recherche
	DRIOLE Jean	Maître de Recherche
	KAMARINOS Georges	Maître de Recherche
	KLEITZ Michel	Maître de Recherche
	LANDAU Ioan-Doré	Maître de Recherche
	MERMET Jean	Maître de Recherche
	MUNIER Jacques	Maître de Recherche

Personnalités habilitées à diriger des travaux de recherche (décision du Conseil Scientifique)

E.N.S.E.E.G.

MM.	ALLIBERT Michel
	BERNARD Claude
	CAILLET Marcel
Mme	CHATILLON Catherine
MM.	COULON Michel
	HAMMOU Abdelkader
	JOUD Jean-Charles
	RAVAINE Denis
	SAINFORT

C.E.N.G.

MM. SARRAZIN Pierre
 SOUQUET Jean-Louis
 TOUZAIN Philippe
 URBAIN Georges

Laboratoire des Ultra-Réfractaires ODEILLO

E.N.S.M.E.E.

MM. BISCONDI Michel
 BOOS Jean-Yves
 GUILHOT Bernard
 KOBILANSKI André
 LALAUZE René
 LANCELOT François
 LE COZE Jean
 LESBATS Pierre
 SOUSTELLE Michel
 THEVENOT François
 THOMAS Gérard
 TRAN MINH Canh
 DRIVER Julian
 RIEU Jean

E.N.S.E.R.G.

MM. BOREL Joseph
 CHEHIKIAN Alain
 VIKTOROVITCH Pierre

E.N.S.I.E.G.

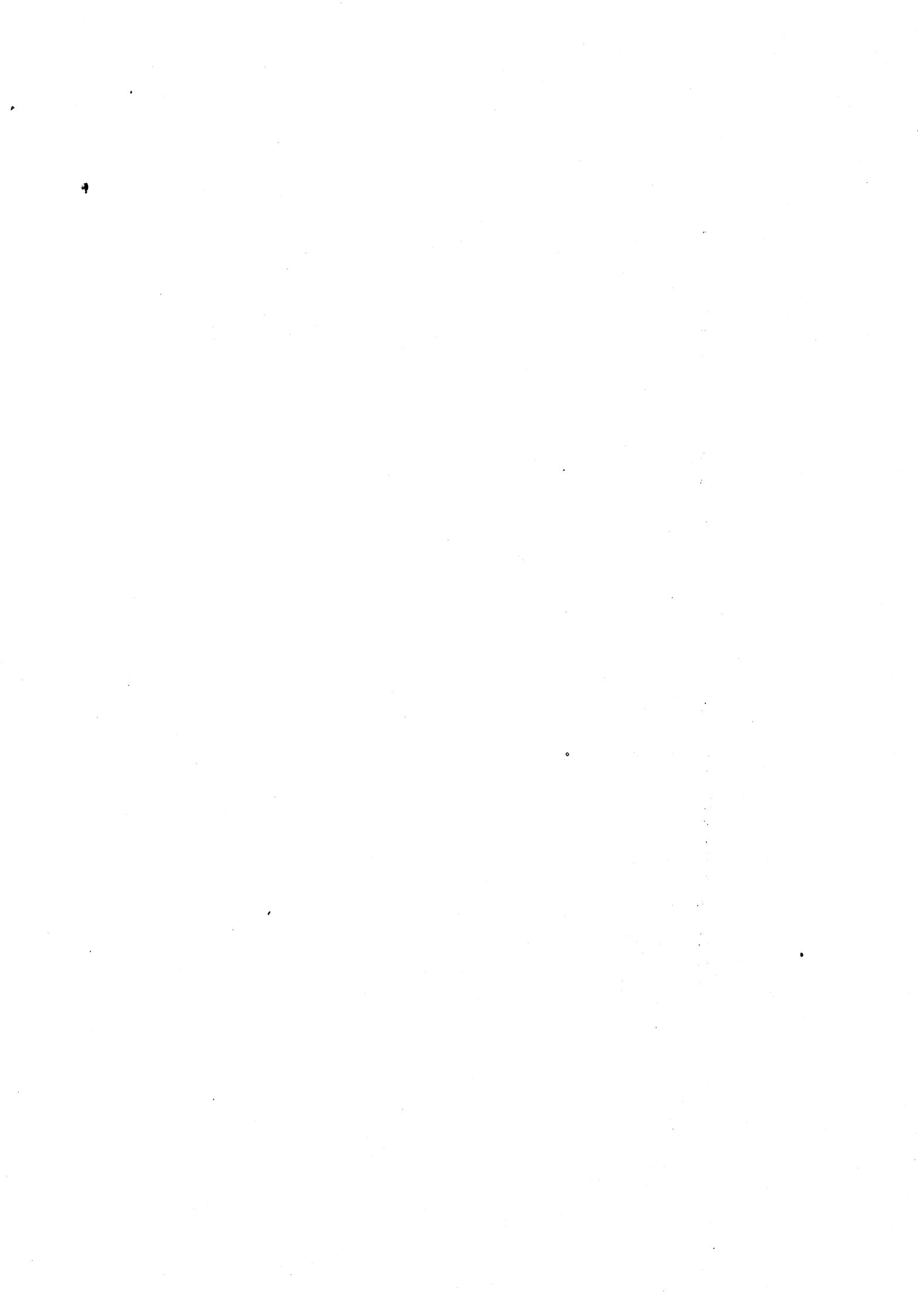
MM. BORNARD Guy
 DESCHIZEAUX Pierre
 GLANGEAUD François
 JAUSSAUD Pierre
 Mme JOURDAIN Geneviève
 MM. LEJEUNE Gérard
 PERARD Jacques

E.N.S.H.G.

M. DELHAYE Jean-Marc

E.N.S.I.M.A.G.

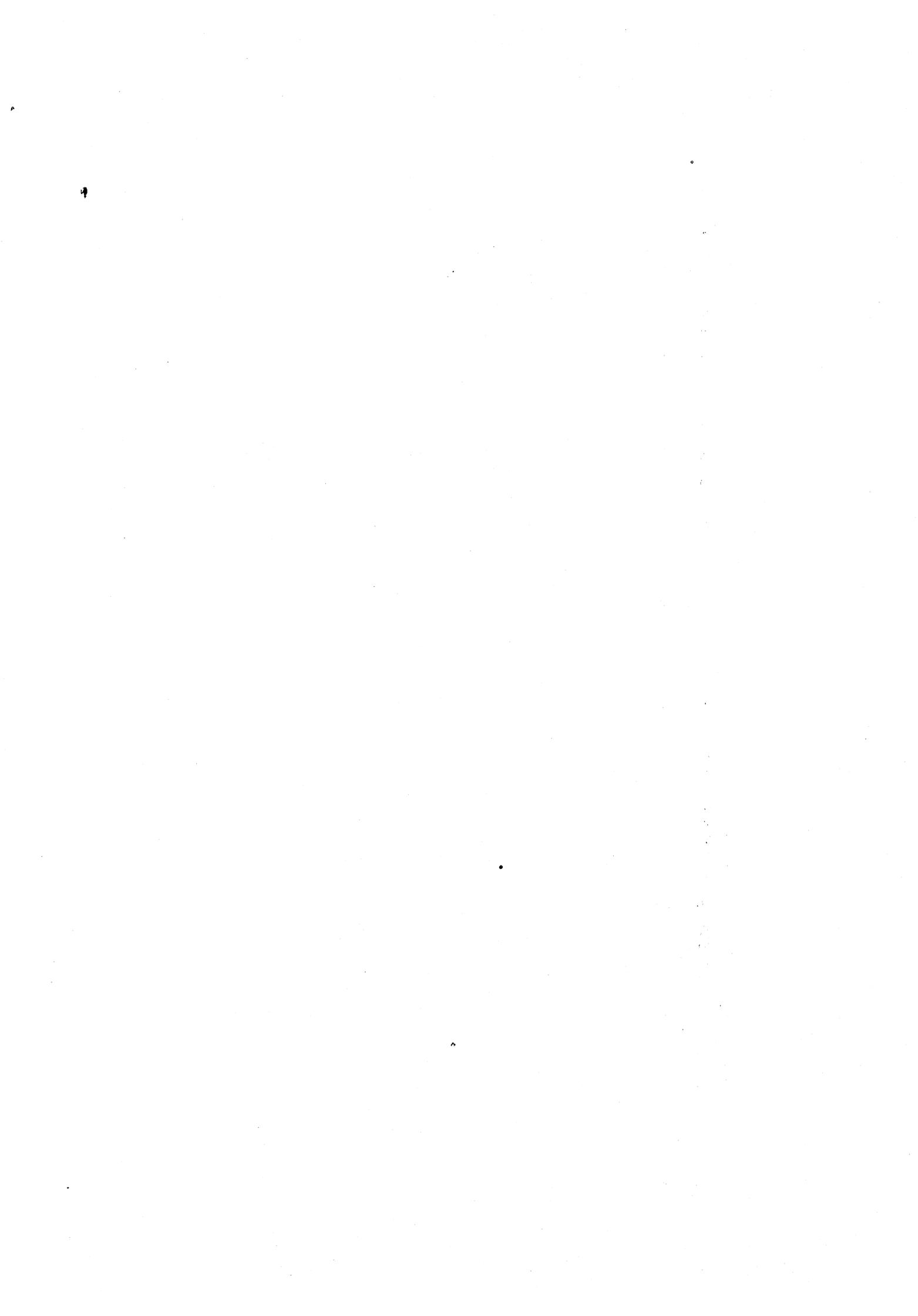
MM. COURTIN Jacques
 LATOMBE Jean-Claude
 LUCAS Michel
 VERDILLON André



Résumé:

Dans cette thèse, nous tentons, dans le cadre des réseaux locaux à diffusion, d'allier la performance des techniques de diffusion à la fiabilité des techniques de transport bi-points. A la suite d'une brève synthèse sur les réseaux locaux, nous présentons un protocole de conversation permettant d'utiliser la propriété de diffusion dans les protocoles d'applications réparties. La définition des services offerts à l'utilisateur par ce protocole est suivie de l'étude et de la comparaison de deux approches de détection et de récupération des erreurs. Les autres mécanismes mis en oeuvre pour assurer ces services sont décrits. Une comparaison du protocole de conversation avec le protocole de connexion bi-points est effectuée et les algorithmes proposés sont simulés. Des éléments d'implémentation d'une station de transport à diffusion fiable sont présentés.

Mots clés : diffusion, conversation, réseau local, détection d'erreur, station de transport



0. INTRODUCTION

0.1 Besoin de diffuser	3
0.2 Utilisation actuelle des réseaux à diffusion	5
0.3 Objectif	5
0.4 Plan de l'étude	6

1. RESEAUX A DIFFUSION.

1.1 Généralités	11
1.2 Les réseaux locaux	
1.2.1 Supports	12
1.2.2 Topologies	13
1.2.3 Méthodes d'accès	14
1.2.4 Performances	14
1.3 Présentation du réseau DANUBE	
1.3.1 Architecture	15
1.3.2 Emission des paquets	16
1.3.3 Réception des paquets	17

2. PROTOCOLE DE CONVERSATION.

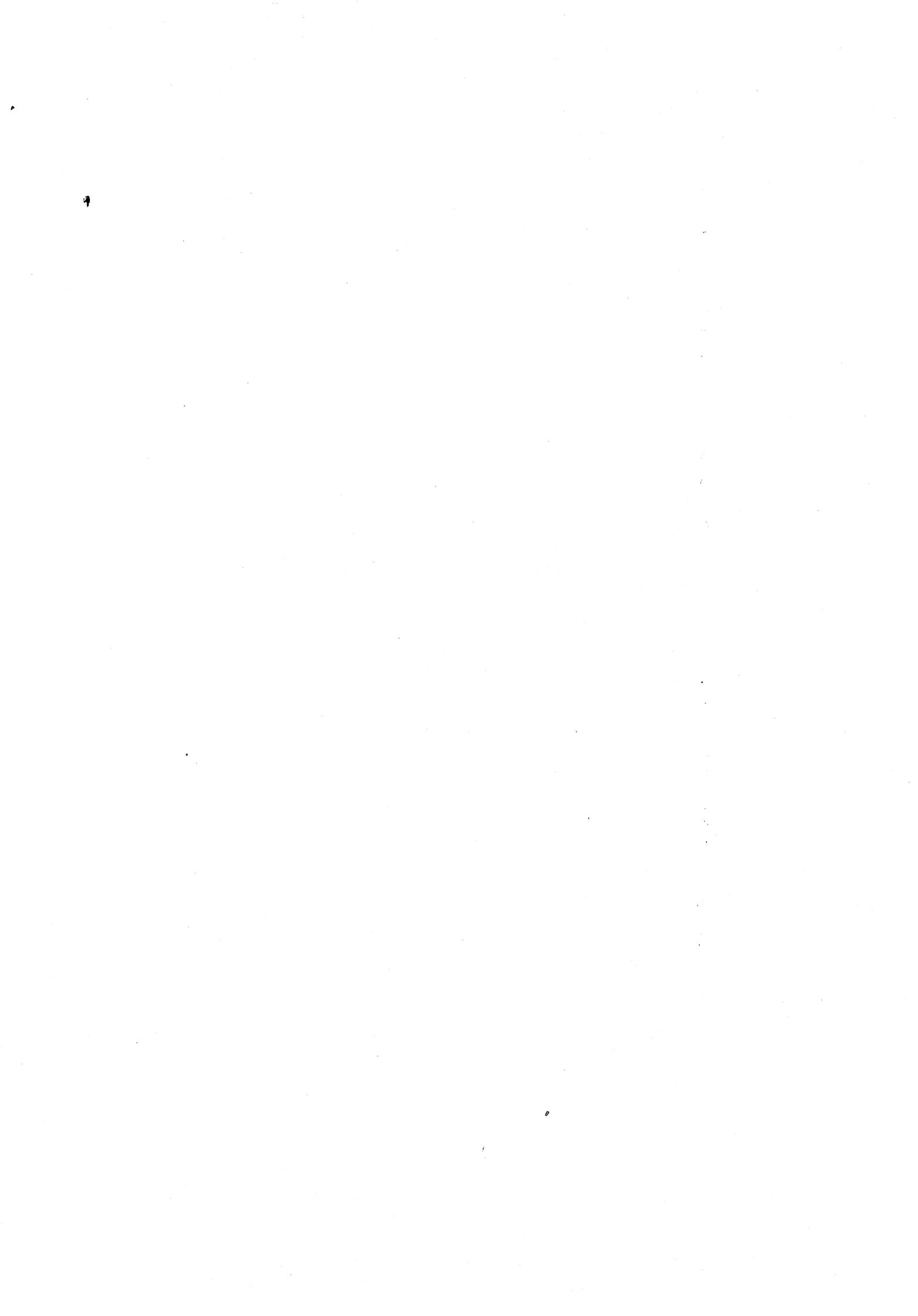
2.1 Notions intuitives	21
2.2 Adressage des membres d'une conversation	21
2.3 Services d'une conversation	
2.3.1 Les services de transfert	23
2.3.2 Les services de connexion	24
2.3.3 Listage des membres d'une conversation	26

3. DETECTION ET RECUPERATION DES ERREURS

3.1 Les erreurs	
3.1.1 Définition	29

3.1.2	Technique de contrôle	30
3.2	Utilisation de l'ordre de passage des paquets sur le médium	31
3.2.1	Numérotation des paquets	31
3.2.2	Détection des erreurs par diffusion de la liste des paquets reçus	33
3.2.3	Récupération des erreurs	46
3.2.4	Problèmes posés	52
3.3	Attribution décentralisée d'un droit de parole	57
3.3.1	Principe	57
3.3.2	Détection et récupération des erreurs	57
3.3.3	Avantages et inconvénients	59
4.	MISE EN OEUVRE DU PROIOCOLE	
4.1	Gestion du droit de parole	63
4.2	Connexion	
4.2.1	Ouverture de la conversation	64
4.2.2	Entrée d'un abonné dans la conversation	66
4.3	Déconnexion	
4.3.1	Sortie d'un membre de la conversation.....	66
4.3.2	Fermeture d'une conversation	67
4.4	Emission de messages	67
4.5	Réception	68
4.6	Listage des membres	68
4.7	Controle de flux	69
4.8	Detection des pannes	70
5.	EVALUATION DU PROIOCOLE	
5.1	Comparaison avec le protocole point a point	75

5.1.1	Diffusion d'un fichier	76
5.1.2	Fonctionnement en mode interactif.....	80
5.1.3	Conversation	82
5.2	Simulation du protocole	87
5.2.1	Description du modèle	87
5.2.2	Les applications	90
5.2.3	Les paramètres	91
5.2.4	Les résultats	91
6.	CONCLUSION	98
ANNEXE		
A.	Format des paquets echanges	103
B.	Interface de transport	106
C.	La station de transport	
C.1	Description des contextes	110
C.2	Schema de fonctionnement	114
C.3	Algorithmes des procedures	119
C.4	Algorithmes des processus	123
	BIBLIOGRAPHIE	131



Je tiens à remercier :

Monsieur Sacha KRAKOWIAK, professeur à l'USMG, pour avoir bien voulu me faire l'honneur de présider le jury de cette thèse,

Monsieur Jacques MOSSIERE, professeur à l'INPG, qui a accepté de juger ce travail,

Monsieur Vincent QUINT, responsable de l'équipe réseau local du laboratoire IMAG, pour l'intérêt qu'il a porté à ce travail,

Monsieur Gerard VANDOME, du centre scientifique CII-HB, qui m'a beaucoup aidé dans ce travail depuis son arrivée à GRENOBLE.

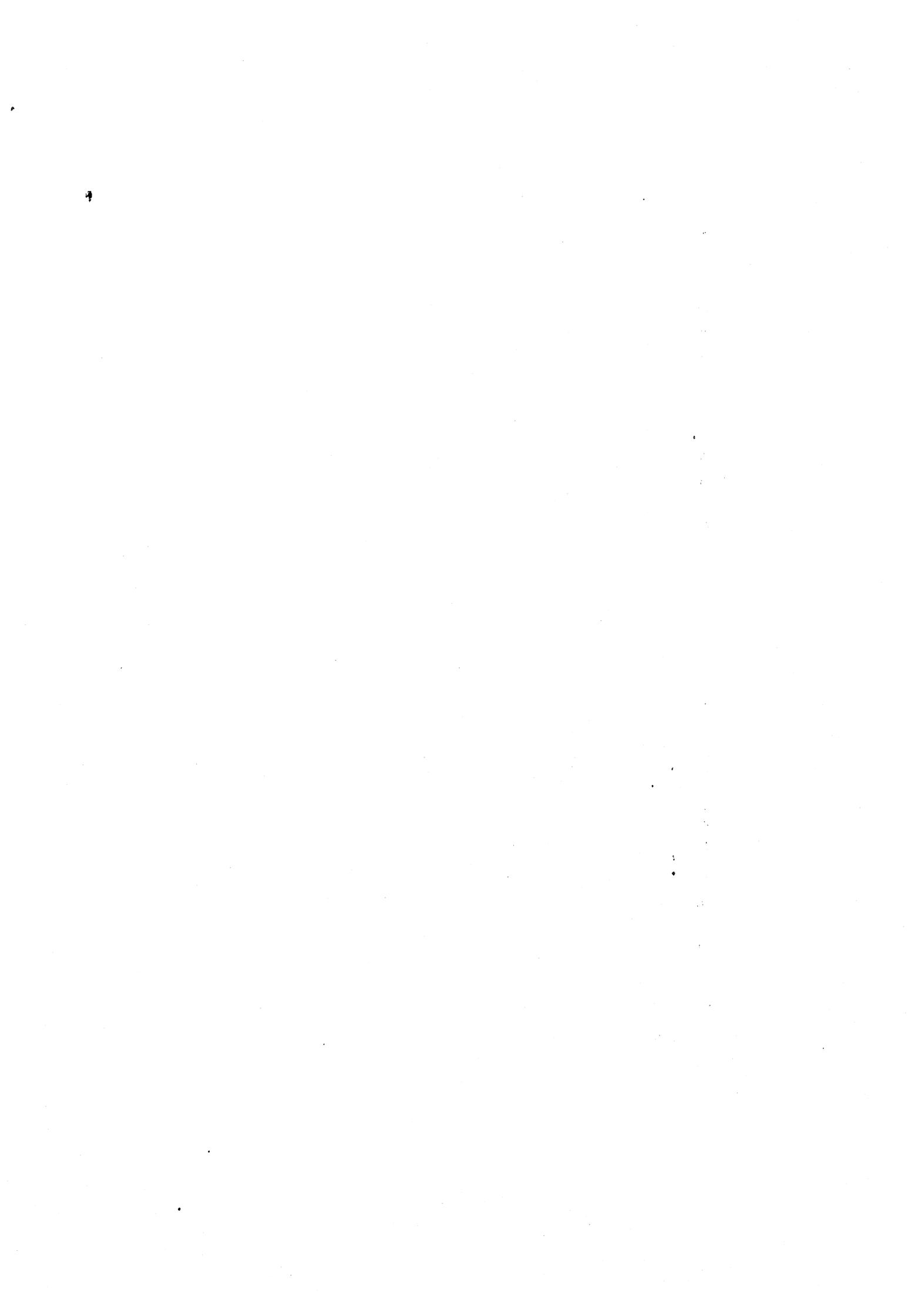
Je remercie aussi Monsieur Paul DECITRE qui a dirigé mes travaux jusqu'à son départ, ses conseils et remarques m'ont permis de réaliser ce travail.

Je tiens à remercier aussi Madame Irene VATTON, Mademoiselle Cecile ROISIN et Messieurs Guy SERGEANT, Jacky ESTUBLIER, Xavier Rousset DE PINA et Michel VERAN pour les remarques et les suggestions dont a bénéficié ce rapport.

Enfin, je remercie toutes les personnes qui ont permis la réalisation matérielle de ce rapport.



0. INTRODUCTION



0.1 BESOIN DE DIFFUSER.

Des études menées sur la coopération cohérente de plusieurs systèmes transactionnels pour exécuter des traitements interactifs [SCO.81] ont permis de définir des protocoles d'applications impliquant plus de deux participants. Ces protocoles font largement appel à des primitives de diffusion de messages au sein d'un ensemble de participants.

D'autres travaux ont mis en évidence un besoin de primitives de diffusion fiable au sein d'une communauté de processus. Parmi ces travaux, on peut citer :

- Le projet MICROBE [DIE.80] qui étudie les problèmes posés par les bases de données relationnelles distribuées sur des micro-ordinateurs,
- Le projet SER [SER.80] qui propose un système d'exécution de logiciels distribués,
- Le projet PAPHYRUS [LUI.80] qui couvrent les problèmes de messageries électroniques sur un ensemble de mini-ordinateurs.

De manière générale, toutes les applications faisant coopérer plusieurs entités (la diffusion de fichier, le transactionnel coopérant, les bases de données réparties, ...) ont besoin de primitives de diffusion fiable. Elles sont toutes multipoints et ont besoin de communications simultanées entre plus de deux processus à la fois.

Initialement, les réseaux de transport étaient développés sur des moyens de communications physiques point à point et offrent de ce fait des services qui répondent bien aux besoins d'applications telles que la connexion d'un appareil à un programme, les transferts de fichiers.

Cette situation n'a pas évolué malgré l'arrivée des réseaux à diffusion. Les applications multipoints qui les utilisent continuent à être développées de la même manière que sur des architectures de type point à point alors que les protocoles de ces applications se simplifieraient avantageusement s'il était possible de remplacer des séries d'envois point à point par une émission en diffusion.

Non seulement l'envoi répétitif de messages identiques consomme inutilement des ressources, mais il introduit un délai important lors de l'envoi d'une commande de changement d'état dans un système distribué. Ce délai augmente la probabilité de pannes critiques du réseau où une partie seulement des systèmes connectés a reçu un message de contrôle alors que l'autre reste dans l'expectative.

Les réseaux à diffusion n'ont pas bouleversé cette situation pour deux raisons principales :

- L'utilisation de services de communication à base de diffusion remet en cause les développements déjà effectués dans les logiciels. Cela ne peut être envisageable que pour développer de nouvelles applications.
- La Mise en oeuvre de la diffusion fiable pose deux problèmes techniques :
 - . Parmi l'ensemble des processus en cours, comment atteindre simultanément ceux qui participent à une même application sans déranger les autres,
 - . Comment garantir que tous les destinataires d'un message l'ont reçu et asservir le débit des émetteurs au rythme du plus lent des récepteurs.

0.2 UTILISATION ACTUELLE DES RESEAUX A DIFFUSION.

En raison de leur haut débit et de leur faible coût, les réseaux à diffusion ne cessent d'être de plus en plus utilisés pour la connexion d'équipements informatiques. La concurrence en ce domaine laisse penser que ceux-ci vont s'imposer.

Actuellement deux modes de communications sont utilisés sur ces réseaux :

- Un mode de communication point à point qui est équivalent au mode de transmission utilisé dans les réseaux maillés. Ce mode permet l'établissement d'une liaison entre deux entités et assure la détection, la récupération des erreurs et le contrôle de flux. Mais il ne permet pas l'adressage de groupe d'entités car il nécessite des réponses de la part de chaque destinataire.
- Un mode de communication sans connexion qui permet l'adressage de groupe d'entités et donne ainsi la possibilité d'utiliser la diffusion. Par contre, il n'offre pas de garantie de réception et n'assure pas de contrôle de flux.

Dans la pratique, la plupart des applications implémentées sur les réseaux à diffusion ne bénéficient pas des potentialités offertes par la diffusion. Elles utilisent en général des protocoles de communication point à point du fait que le mode diffusion n'offre aucune garantie quant à la bonne réception des données.

0.3 OBJECTIF

Le but de notre travail est d'apporter, dans le cadre des réseaux locaux à diffusion qui utilisent la technique CSMA/CD comme méthode d'accès, une solution aux problèmes techniques qui freinent l'usage de la diffusion dans la définition des protocoles

d'applications distribuées. En nous appuyant sur les possibilités offertes par le réseau local à diffusion DANUBE, nous nous sommes intéressés :

- D'une part à la définition d'un protocole permettant à 2 ou plusieurs entités coopérantes de se rencontrer et d'échanger des messages en diffusion avec contrôle d'erreur et contrôle de flux.

Le contrôle d'erreur du type de celui que l'on rencontre dans les protocoles de transport point à point étant incompatible au cas d'échange entre plus de deux entités, la recherche d'un mécanisme adéquat constitue une grande partie de ce travail.

- D'autre part à l'évaluation de ce protocole en le comparant au protocole de transport point à point et en le simulant en vue de détecter des cas de famine, de boucle infini etc...

0.4 PLAN DE L'ETUDE.

- Le premier chapitre est consacré aux réseaux à diffusion. Nous y abordons de manière générale les caractéristiques des réseaux à diffusion (supports utilisés, partage de la voie, ...) et de façon plus détaillée les réseaux locaux suivies de la description du réseau expérimental DANUBE.

- Dans le chapitre 2 nous présentons le protocole de conversation. Nous y trouvons:

- . la description du mécanisme d'adressage des membres d'une conversation basé sur des noms réseaux,

- . la définition précise des services offerts par une conversation.

- Dans le chapitre 3, sont présentées deux techniques de détection et de récupération des erreurs. Toutes les deux utilisent la numérotation des paquets comme moyen de détection des erreurs.

La première suppose l'unicité du médium de transmission et est basée sur l'ordre de passage des paquets sur ce médium. La deuxième technique est basée sur un mécanisme d'attribution décentralisée d'un droit de parole qui rend impossible l'attribution du même numéro à plusieurs paquets, et permet donc de simplifier la récupération des erreurs.

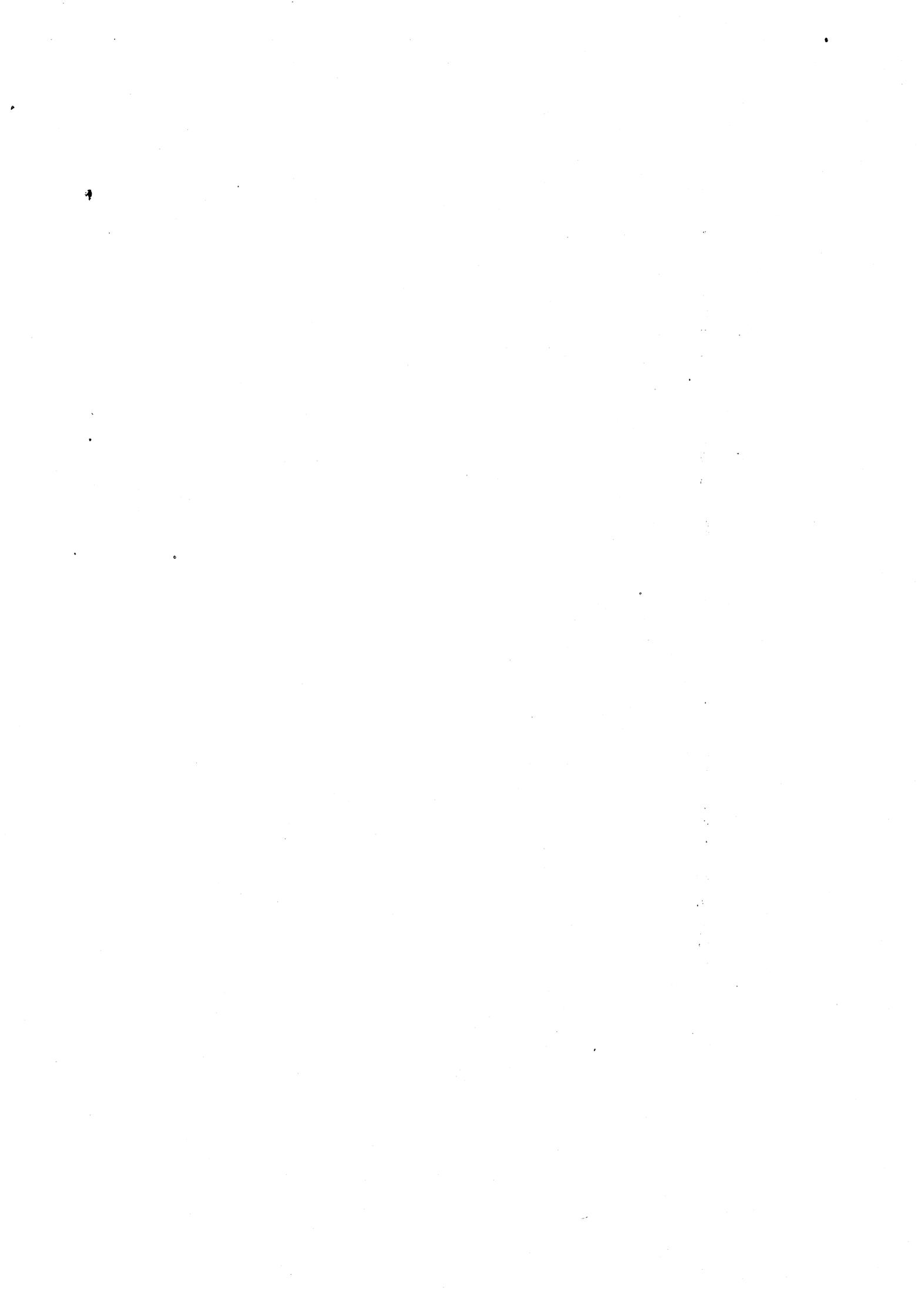
- Le chapitre 4 est consacré à la mise en oeuvre du protocole en utilisant la deuxième technique de détection et récupération des erreurs. Nous y traitons:

le protocole de gestion de la parole,
la mise en oeuvre des différents services (connexion, déconnexion, émission et réception des messages),
le contrôle de flux,
la détection des pannes et reprises.

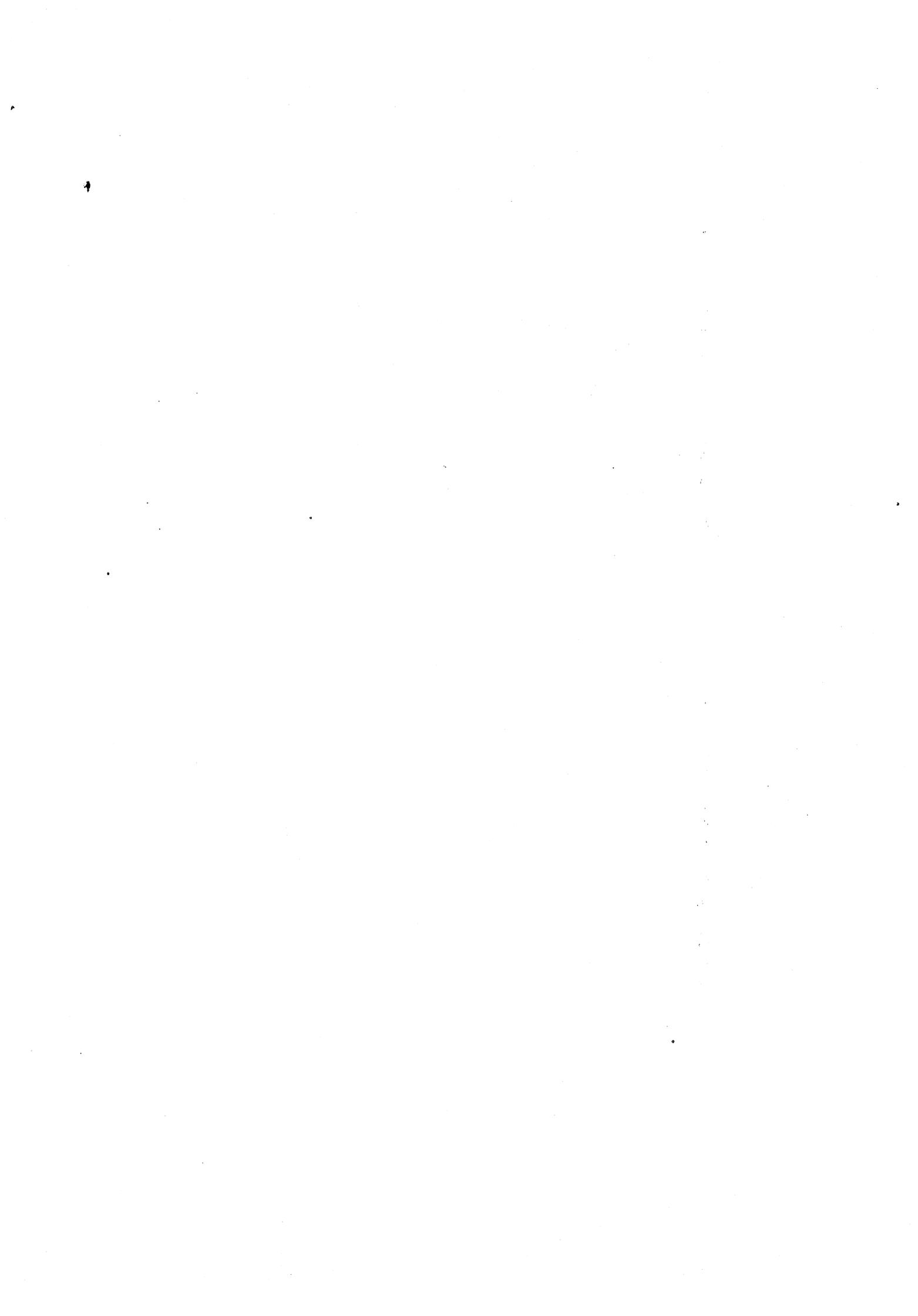
- le chapitre 5 est consacré à une évaluation du protocole.

La première partie de ce chapitre est consacré à une étude comparative avec les liaisons point à point,

La deuxième partie à la simulation du protocole en vue de détecter les cas de mauvais fonctionnement (famine, boucle infinie, ...)



1. RESEAUX A DIFFUSION.



1.1 GENERALITES.

On entend par réseau à diffusion un moyen ou un système de transmission à correspondants multiples dans lequel lorsqu'un correspondant émet un signal, tous les autres le reçoivent. Les réseaux à diffusion [JAC.78] [DAV.79] permettent en général un haut débit avec un faible taux d'erreur.

Les supports les plus couramment utilisés sont les suivants:

- Transmission par radio terrestre.
- Transmission par satellite.
- Transmission sur cable coaxial.
- Transmission sur fibre optique.

L'ensemble des correspondants utilise le même support de transmission. Si à un moment donné, un correspondant est le seul à émettre, son message est reçu par l'ensemble des destinataires. Par contre si deux ou plusieurs correspondants décident simultanément d'émettre chacun un paquet, ils se brouillent mutuellement. On dit que les deux correspondants sont entrés en concurrence pour l'utilisation du support de transmission ou que les deux messages sont entrés en collision. Pour éviter les collisions, de nombreuses méthodes de partage de la voie sont utilisées. Ces méthodes dépendent du type de la voie et des objectifs recherchés. Le partage peut être réalisé de manière statique ou dynamique [COR.81].

Dans le cas d'un partage statique, chaque correspondant sait quand il peut utiliser la voie sans provoquer de collisions. On rencontre deux techniques qui sont:

- Découpage de la bande passante en plusieurs plages de fréquences pour permettre le multiplexage sur la même voie de plusieurs communications (FDM),

- allocation d'une tranche de temps de longueur constante à chaque correspondant (TDM).

A l'intérieur du partage dynamique, on distingue deux cas :

- Partage par compétition (cas auquel se limite notre étude) : la méthode repose sur l'absence d'arbitre ou de protocole de coopération. Chaque correspondant envoie librement ses messages sans se préoccuper des autres correspondants. Pour éviter partiellement les collisions, un émetteur écoute la voie et attend qu'elle soit silencieuse pour émettre. Cette méthode appelée CSMA (Carrier Sense Multiple Access) est utilisée dans la plupart des réseaux locaux. Si deux émetteurs écoutent en même temps et émettent dès que la voie est libre, il y a collision. Les messages en collision sont altérés et l'émetteur le détecte car il reçoit les informations transmises de la même manière que les autres correspondants et les compare bit à bit avec celles qu'il a émises.

- Partage par élection : Tout émetteur demande la voie par émission d'un message de requête et attend d'être élu. Cette élection peut se faire par :

- . Consultation : les émetteurs sont interrogés dans un ordre fixe,
- . Selection : on exécute un algorithme pour élire un émetteur.

1.2 LES RESEAUX LOCAUX

1.2.1 Supports

En général, les réseaux locaux [CLA.78] ont été créés pour permettre, à des vitesses élevées, l'interconnexion de systèmes informatiques séparés par de courtes distances (quelques

kilometres). Ils offrent pratiquement tous un mode de transmission par paquets. Les supports couramment utilisés sont :

- Le cable coaxial: plus approprié dans les cas d'applications qui nécessitent un partage intensif des ressources réparties sur le réseau.
- La fibre optique: préférée au coaxial dans les cas d'échange à très haute vitesse (débit important d'informations).

1.2.2 Topologies.

Deux topologies dominant: le bus et l'anneau

- Le bus.

La majorité des réseaux locaux utilisent une architecture bus. Une des réalisations marquantes en ce domaine est le réseau ETHERNET [MET.76]. Un certain nombre de systèmes informatiques sont interconnectés par le cable et se partagent des ressources. Pour avoir un temps de réponse global acceptable, le nombre de stations dépasse rarement 100 à 200. Avec cette architecture, de nouveaux tronçons de cable peuvent être ajoutés à ceux qui existent déjà grâce à des répéteurs. L'ajout ou la suppression d'un noeud se fait sans perturber le fonctionnement du réseau par simple branchement d'un boitier de raccordement sur le cable.

- L'anneau.

Les réseaux en anneau sont entièrement fermés. Un des exemples de réalisation de réseau de ce type est l'anneau de Cambridge [HER.81]. Les premiers réseaux présentaient l'inconvénient majeur de tomber hors service à chaque panne de noeud. Depuis lors, grâce à des dispositifs de sécurité (matériels + logiciels), les noeuds

peuvent se déconnecter sans entraver le fonctionnement du réseau. Cette architecture présentent toujours l'inconvénient de l'arrêt de tout le réseau à chaque fois que l'on veut rajouter un noeud.

1.2.3 Méthodes d'accès.

L'architecture d'un réseau local est très liée à la méthode d'accès.

Les réseaux de type bus utilisent généralement la méthode dite CSMA/CD (Carrier Sense Multiple Access with Collision Detection). Dans cette méthode, l'interface réseau de chaque station raccordée au bus écoute en permanence et dès qu'il y'a absence de signal sur le cable, elle commence sa transmission. Les collisions sont détectées en comparant l'information transmise à celle reçue.

Les réseaux de type anneau utilisent la méthode de droit à transmettre ou jeton (token passing). Un message, dit jeton, passe d'un noeud à l'autre toujours dans le même sens. Chaque station qui detient le jeton, le passe à la station suivante dès qu'elle a fini sa transmission ou consommé son temps de transmission,

1.2.4 Performances.

Vis à vis des "réseaux longue distance", les performances des réseaux locaux sont de loin plus importantes et ce principalement sur 2 points :

- Le débit (bande passante):

- . quelques Kilobits par seconde pour les réseaux longue distance,
- . quelques mégabits pour les réseaux locaux.

- Le délai de propagation: Il est de l'ordre du dixième de seconde pour les réseaux longue distance et de la micro-seconde pour les réseaux locaux.

La comparaison entre les différents réseaux locaux existants doit se situer au niveau des topologies, des vitesses de transmission, des méthodes d'accès ainsi qu'au niveau de tous les aspects logiciels.

1.3 PRESENTATION DU RESEAU DANUBE

1.3.1 Architecture.

Le réseau local DANUBE [DAN.79] est un réseau expérimental dérivé du réseau ETHERNET. Il est bâti en architecture bus autour d'un câble coaxial sur lequel sont connectés des communicateurs programmables. Le réseau peut s'étendre sur une longueur de 1 Km avec un débit de 1 Mbits par seconde. L'accès au câble se fait en CSMA/CD. Chaque communicateur peut émettre dès que le câble est "silencieux" (absence de signal).

Chaque site du réseau comprend un ordinateur et un commutateur. Tous les sites sont connectés à un même câble qui constitue le médium de transmission.

Le commutateur se compose d'un coupleur, d'une "boîte intelligente" dotée d'un microprocesseur et de mémoires.

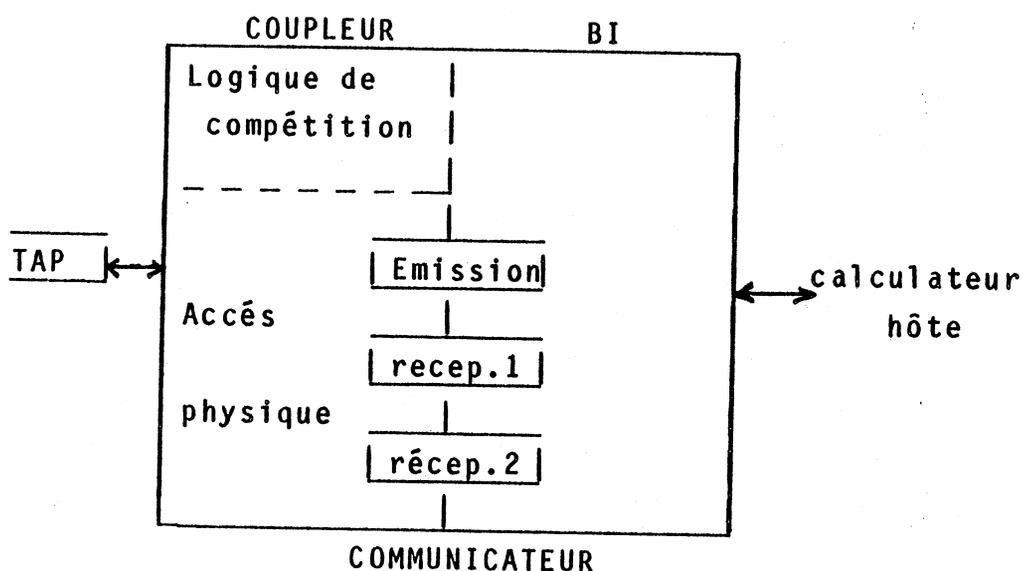
Trois mémoires tampons 256 octets chacune sont partagées entre la boîte intelligente et le coupleur.

- une mémoire tampon d'émission,
- deux mémoires tampons de réception.

Le commutateur est relié au câble par un adaptateur dont le rôle est :

- . Emission / réception
- . Détection de l'occupation du câble
- . Détection des collisions

Le coupleur gère l'accès physique (sérialisation, désérialisation, délimitation des trames format HDLC et le calcul et la vérification des erreurs de transmission) et la logique de compétition (calcul des délais de transmission après collision et retransmission).



Dans la boîte intelligente "BI" nous trouvons la procédure de gestion de la ligne communicateur-calculateur.

1.3.2 Emission des paquets.

Un paquet à émettre est remis par le calculateur hôte à la boîte intelligente qui, à son tour, le range dans le tampon émission, puis le signale au coupleur en lui communiquant la longueur. C'est le coupleur qui prend l'initiative de la

transmission du paquet sur le cable une fois que celui-ci se trouve "silencieux".

L'émission des paquets se fait avec écho, c'est à dire que le coupleur reçoit ce qu'il a transmis. Il détecte une collision en comparant l'information transmise avec l'information reçue. Quand la collision est détectée le coupleur émet un signal de brouillage pour avertir les autres coupleurs de la non validité du paquet. Le paquet est retransmis au bout d'un intervalle de temps déterminé aléatoirement (logique de contention).

Si plusieurs stations décident d'émettre un paquet dans un même intervalle de temps, l'ordre de transmission est imprévisible pour 2 raisons :

- . Le cable est occupé par la transmission de l'un des paquets, les autres doivent attendre que le cable se libère.
- . Les paquets transmis entrent en collision, ils devront être retransmis plus tard. Pendant ce temps des paquets d'autres stations peuvent être transmis.

A partir de l'instant où un paquet est émis (confié au coupleur) on ignore au bout de combien de temps il sera transmis (envoyé sur le cable). La boîte intelligente est avertie par un signal dès que le paquet est correctement transmis. Elle peut à nouveau remplir le tampon.

1.3.3 Réception des paquets.

En fin de réception d'un paquet, le coupleur prévient par un signal la boîte intelligente si ce paquet est reçu sans erreur de transmission et est adressé a ce site. La boîte intelligente peut alors transférer dans ses propres buffers le contenu de la mémoire tampon où est reçu le paquet. L'assignation des mémoires tampon en réception est réalisée de manière cyclique par le coupleur.

Tout message reçu alors que les 2 mémoires tampon sont pleines n'est pas pris en compte .

2. PROTOCOLE DE CONVERSATION.



2.1 NOTIONS INTUITIVES

Le but du protocole de diffusion fiable est d'offrir des services de transport adaptés aux applications nécessitant la diffusion. Ces services ne sont offerts ni par les protocoles de transport normalisés ni par les protocoles de transport utilisés dans les réseaux locaux actuels.

Ces services doivent être des services de base à partir desquels, il est facile de construire les protocoles d'applications. Ces applications, contrairement aux applications classiques nécessitent la mise en relation simultanée de plus de deux entités coopérantes.

On a introduit la notion de conversation définie comme la mise en relation, par échange de messages, d'un nombre quelconque d'entités coopérantes. Cela permet d'éviter la complexité de la solution qui consiste à établir autant de connexions point à point qu'il y a de couples d'entités devant échanger des messages.

Deux notions intuitives seront donc utilisées pour définir les services offerts par une station de transport à diffusion :

- la notion de conversation
- la notion d'entités communicantes que nous appellerons abonnés de la conversation (un abonné sera désigné par le terme de membre s'il participe déjà à la conversation).

2.2 ADRESSAGE DES MEMBRES D'UNE CONVERSATION.

Il s'agit de définir un mécanisme de reconnaissance mutuelle des entités communicantes, c'est à dire l'information permettant à un membre d'une conversation de désigner un autre membre ou l'ensemble des autres membres.

Pour devenir membre d'une conversation, un abonné de la station de transport se présente sous son nom d'abonné (chaîne de caractères). Ce nom permet d'identifier l'origine des messages qu'il a émis dans la conversation et la destination des messages qui lui sont transmis.

On peut définir l'adressage des membres d'une conversation de deux manières :

- Chaque message émis peut contenir la liste des noms des membres destinataires, ainsi chaque station de transport peut savoir si un message passant sur le réseau est destiné à l'un des membres qu'elle gère.

- On peut, au contraire donner un identificateur à chaque conversation (CVID) et, si chaque station de transport maintient la liste des CVID des conversations auxquelles elle participe, elle pourra reconnaître ce qui lui est destiné dans le trafic en diffusion.

Nous avons choisi cette deuxième solution car elle permet :

- De garder des champs d'adressage courts,
- La connexion et la déconnexion dynamique des participants,
- L'émission vers un groupe dont on peut ignorer la composition exacte.

Nous supposons ici que les noms réseaux sont alloués à l'extérieur de la couche transport.

Pour désigner les membres d'une conversation à l'aide d'un identificateur, il faut disposer d'un mécanisme d'adressage

spécifique. Nous utilisons ici les possibilités offertes par le réseau expérimental DANUBE [DAN.79]. Chaque site de ce réseau dispose d'un communicateur permettant un adressage à 2 niveaux. Le premier niveau est composé de deux numéros. Un numéro propre au site compris entre 0 et 254 et un numéro général (diffusion) égal à 255. Le deuxième niveau est composé d'une table contenant une liste de noms. Un paquet est pris en compte par le communicateur s'il porte l'un des deux numéros.

Les paquets en diffusion portent le numéro 255 et une adresse secondaire appelée "nom réseau" sous la forme d'une chaîne de caractères. Ils sont filtrés par le communicateur en fonction de la liste des noms. Si le nom réseau d'un paquet se trouve dans la table des noms, il est accepté sinon il est ignoré. Cette table est tenue à jour en fonction des demandes qui sont faites par le niveau transport.

2.3 SERVICES D'UNE CONVERSATION

Les abonnés ont à leur disposition deux types de services :

- L'initialisation de la conversation et la manière dont les abonnés peuvent entrer et sortir d'une conversation; ce sont les services de connexion,
- l'émission et la réception des données; ce sont les services de transfert de données.

2.3.1 Les services de transfert

Les messages sont échangés soit en diffusion (de l'un des membres vers tous les autres membres) soit en aparté (de l'un des membres vers un autre membre). Un message est une chaîne de caractères de longueur variable fournie par un membre. La longueur maximale d'un message est un paramètre commun à toutes les stations à diffusion. La couche transport garantit :

- L'égalité de tous les membres dans leur droit de parole,
- que tous les messages émis sur une conversation seront délivrés à tous les membres dans les cas d'envois en diffusion ou à son unique destinataire dans les cas d'envois en aparté,
- l'ordre de réception des messages est le même chez tous les membres de la conversation,
- que les messages émis par un même membre seront reçus dans l'ordre de leur émission,
- L'asservissement du flux des émetteurs au débit des récepteurs,
- Les erreurs non récupérables provoqueront la déconnexion des membres correspondants de la conversation.

Le service aparté permet à un membre d'envoyer un message à destination d'un autre membre de la conversation. Ce message est ordonné par rapport aux autres messages émis dans la conversation. Ce service permet d'éviter l'ouverture d'une conversation entre 2 membres, solution qui peut être préférable dans le cas où le volume d'informations échangées est important et indépendant par rapport aux messages de la conversation.

Un service télégramme peut être facilement rajouté à ces services de transfert.

2.3.2 Les services de connexion.

Les services de connexion permettent aux abonnés d'une station de transport :

- de déclarer les noms de conversations auxquelles ils désirent participer,

- de se déclarer eux mêmes en tant que membres,
- de quitter ces conversations.

Connexion.

Un abonné de la station de transport peut être connecté à une conversation à partir du moment où il est capable de fournir le nom de celle-ci. Il n'y a pas d'autres contrôles d'accès.

Nous supposons qu'un abonné est capable d'indiquer à la station de transport s'il est le premier à utiliser un nom de conversation. Ainsi il y a toujours une station particulière qui initialise une conversation. Cette station n'a aucun privilège particulier par la suite.

La connexion effective d'un abonné de la station de transport à une conversation se fait en deux étapes. Un abonné qui s'intéresse à une conversation demande à la station de transport de "l'ouvrir". Il peut ensuite demander la liste des membres de la conversation et décider de le devenir à son tour.

Un nouveau membre de la conversation ne recevra que les messages émis après son entrée effective.

Tout membre est averti des arrivées et des départs des autres membres au fur et à mesure qu'ils se produisent.

Déconnexion.

Il existe deux types de déconnexion: la déconnexion volontaire d'un membre et la déconnexion involontaire provoquée par une panne.

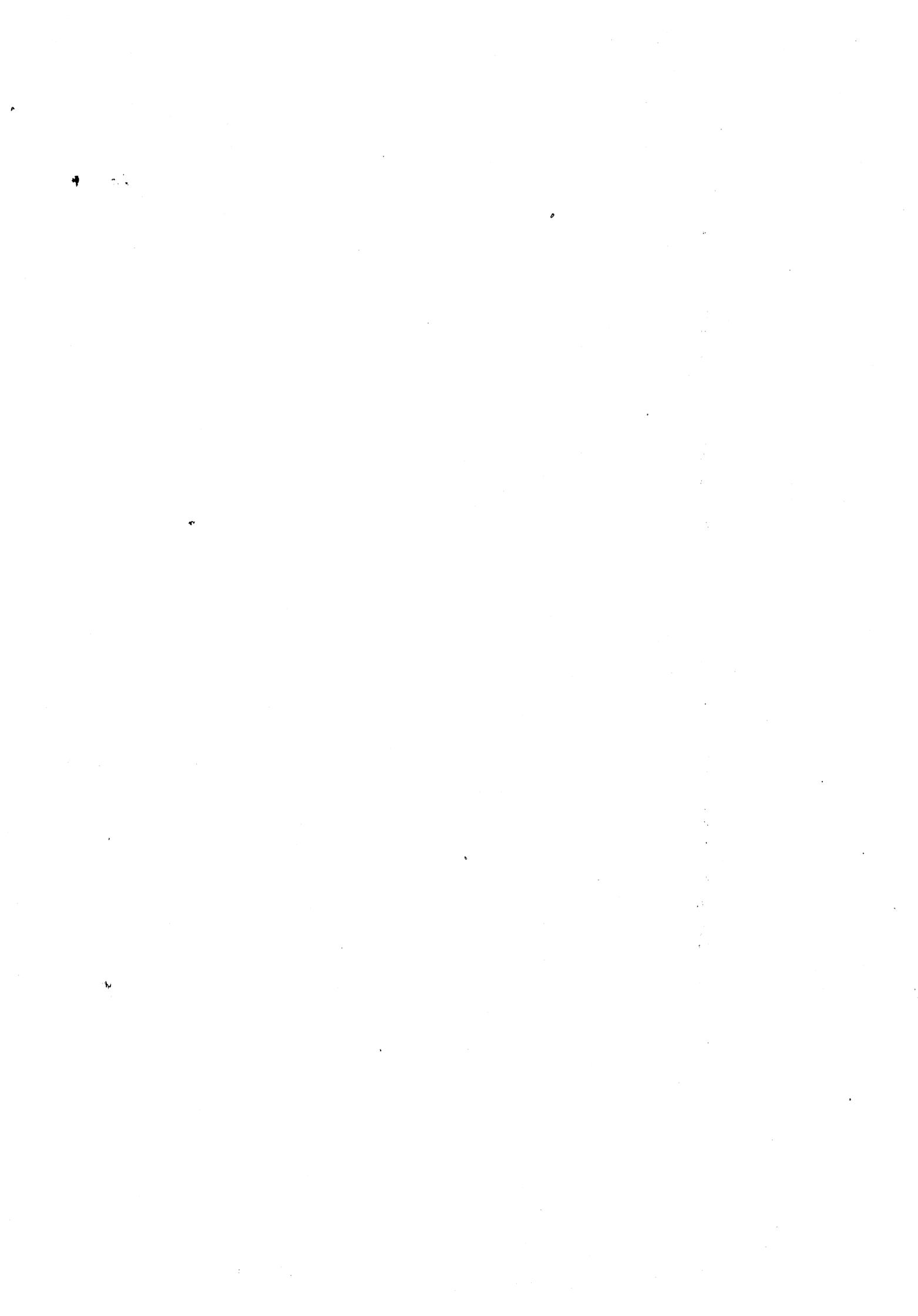
- un membre peut volontairement demander de quitter une conversation. Son départ est signalé aux membres restants.

- Certains membres peuvent devenir inaccessibles suite à une panne, les membres restants sont avertis de la sortie involontaire de certains membres sans indiquer leur nom. Un membre voulant connaître la liste exacte des membres restants doit utiliser le service de listage.

2.3.3 Listage des membres d'une conversation.

Un abonné de la station de transport ou un membre de la conversation peut à tout moment demander à la station de transport la liste exacte des membres de la conversation. Cette information peut permettre à l'application de construire des mécanismes de rendez-vous complexes.

3. DETECTION ET RECUPERATION DES ERREURS



3.1. LES ERREURS.

3.1.1 Définition.

Les paquets à émettre comprennent un nombre entier d'octets et sont soumis par la couche transport au niveau ligne. Le niveau ligne enveloppe chaque paquet dans une trame HDLC avant de le transmettre. Il assure aussi la réception des paquets et la détection des erreurs de transmission. Un paquet reçu par le niveau ligne ne sera pas remis à la couche transport si une des deux conditions suivantes est remplie :

- Une erreur de transmission est détectée à la réception du paquet,
- l'interface n'a pas de tampon libre au moment où est arrivé le paquet.

On ne considère dans la suite de cette présentation que des erreurs de ces 2 types. On suppose que les autres mauvais fonctionnements sont signalés par le matériel et impliquent l'arrêt de la transmission. Ces types d'erreur vont se traduire par l'un des deux cas suivants :

- . Le paquet transmis est reçu par l'émetteur et n'est pas reçu par certaines autres stations destinataires.
- . Le paquet transmis, est reçu par l'émetteur et n'est reçu par aucune des stations destinataires.

Une station reçoit toujours les paquets qu'elle a émis dans le même ordre que toutes les autres stations.

3.1.2 Technique de contrôle.

La principale difficulté rencontrée dans la définition d'un protocole de transport basé sur la diffusion est certainement l'efficacité du contrôle d'erreur.

Pour des liaisons point à point, le contrôle d'erreur consiste habituellement à acquitter chaque paquet correctement reçu. Des mécanismes permettent une factorisation des acquittements et donc une diminution du nombre d'acquittements circulant effectivement sur le réseau. On peut craindre malgré cela, que dans le cas d'une conversation mettant en jeu N stations le nombre d'acquittements circulant sur le réseau devienne trop important ($N-1$ acquittements par paquet émis).

Aussi nous avons choisi une technique de contrôle d'erreur s'appuyant sur l'envoi d'acquittements négatifs. Plus précisément, une station ayant détecté qu'elle n'a pas reçu un certain paquet émet un acquittement négatif à destination de son émetteur. Les stations silencieuses émettront sur temporisateur des acquittements pour signaler leur présence. L'avantage d'une telle méthode est qu'elle garantit en l'absence d'erreur une faible surcharge du réseau. Par contre, pour qu'elle puisse être mise en oeuvre efficacement, il faut garantir qu'une station soit capable de détecter rapidement la perte d'un paquet.

Le moyen retenu consiste à numéroter, par conversation, les paquets avant leur émission de telle manière que chaque station puisse détecter les paquets manquants dans cette conversation. Nous allons dans ce qui suit, tout en utilisant la numérotation des paquets, étudier 2 méthodes de détection des erreurs. La première s'appuie sur l'ordre de passage des paquets sur le médium et le second sur un mécanisme d'attribution décentralisée de la parole.

3.2. UTILISATION DE L'ORDRE DE PASSAGE DES PAQUETS SUR LE MEDIUM.

Le mécanisme de détection des erreurs dont la description suit, repose sur l'unicité du médium de transmission. Les décisions d'émettre sont décentralisées, chaque station de transport soumet à l'interface réseau le paquet qu'elle désire émettre dès qu'il est prêt. L'interface transmet le paquet dès que le support est libre.

De l'unicité du médium de transmission découlent les propriétés suivantes:

- . Tous les paquets sont diffusés sur le même médium.
- . Les paquets ne peuvent pas se croiser sur le médium ni être dupliqués

Il en résulte que :

- L'ordre de réception des paquets est le même pour toutes les stations,
- Les paquets émis par une même station seront reçus dans l'ordre où ils ont été émis,
- L'émetteur reçoit les paquets qu'il a diffusés de la même manière que les autres destinataires.

Les paquets transmis, en l'absence d'erreur, doivent être reçus par toutes les stations dans le même ordre.

3.2.1 Numérotation des paquets

Le principe de la numérotation consiste pour la station émettrice du paquet à affecter au paquet un numéro se rapprochant le plus possible de son rang de passage sur le câble. Elle

maintient donc un compteur R des paquets reçus et affecte à chaque paquet qu'elle émet un numéro NM égal à R+1.

Le fait que les stations émettent de manière indépendante, sans connaissance précise de l'état du médium, implique que la numérotation pourra présenter les anomalies suivantes :

- existence de plusieurs numéros identiques

En effet soit ST_j et ST_k deux stations. A l'instant t, alors que R_j et R_k valent tous deux n, ces deux stations décident d'émettre.

Si N_j et N_k désignent respectivement les numéros affectés par ST_j et ST_k on aura :

$$N_j = N_k = n+1$$

- existence de trous

Supposons que le cas décrit précédemment se soit produit et que q paquets de même numéro aient été reçus par toutes les stations présentes. Le mécanisme fera que les numéros n+1 à n+q-1 ne seront pas affectés.

- La suite des numéros circulant sur le cable n'est pas croissante

En effet un paquet émis par une station peut être retardé dans un coupleur et passer sur le cable à la suite de paquets émis après lui par d'autres stations. L'ordre de transmission n'est donc pas toujours égal à l'ordre d'émission des paquets.

Ces anomalies font que les numéros seuls ne suffisent pas pour détecter tous les cas d'erreurs. Pour éliminer le cas de plusieurs paquets de mêmes numéros, nous les désignons de manière unique par le couple (NM = numéro du paquet , SE = numéro de la station émettrice).

3.2.2 Détection des erreurs par diffusion de la liste des paquets reçus.

3.2.2.1 Description de la liste.

Sur chaque station, on maintient par conversation une file F qui fournit l'historique des paquets de données reçus par la station dans cette conversation. Chaque rang de cette file contient l'identité (NM,SE) d'un paquet reçu.

En absence d'erreur :

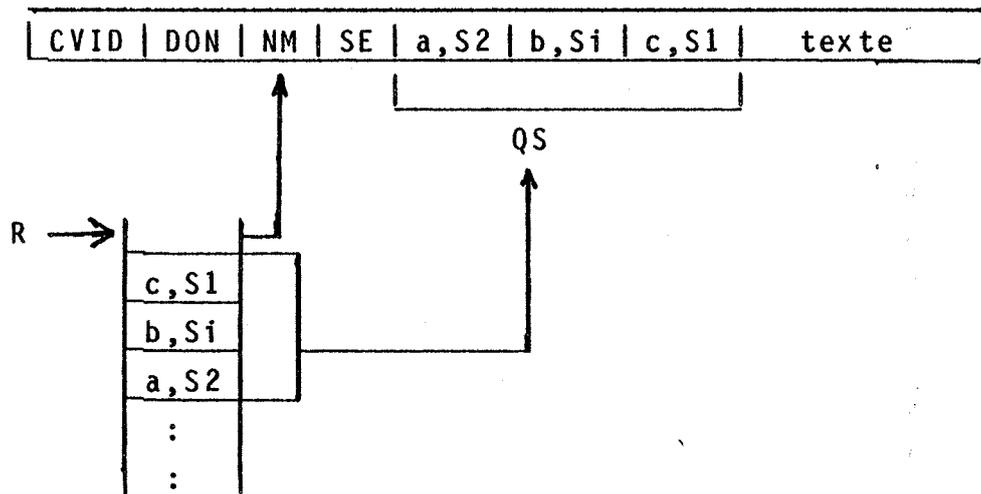
- Le rang d'un paquet dans la file correspond à son ordre de passage sur le support par rapport aux paquets de données transmis dans cette conversation,
- Les contenus de toutes ces files sont identiques.

L'inégalité d'une des files par rapport aux autres files à un instant donné signifie la présence d'une perte de paquet par la station. Pour détecter cette inégalité, nous diffusons dans chaque paquet de données un état partiel de la file de la station émettrice. Cette état noté QS contiendra la liste des derniers paquets reçus par la station.

Nous trouverons dans l'entête des paquets données les informations suivantes:

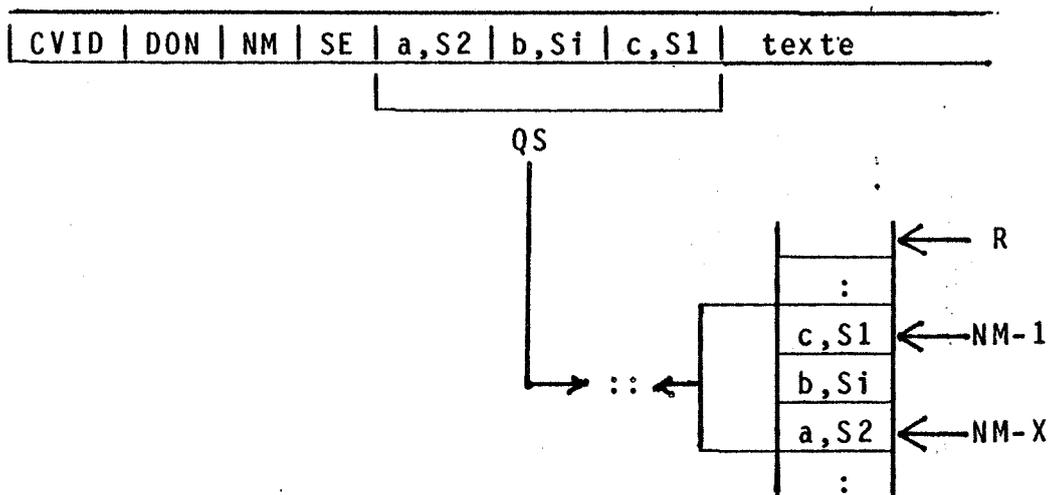
- CVID = Nom de la conversation,
- DON = Paquet de type données,
- NM = Numéro du paquet,
- SE = Station émettrice du paquet,
- QS = identités des derniers paquets reçus par la station émettrice.

Paquet à émettre.



F de la station émettrice

paquet reçu



F de la station
réceptrice

A la réception d'un paquet de données, nous procéderons à la comparaison du QS qui accompagne le paquet reçu avec la partie correspondante dans la file du récepteur en vue de détecter d'éventuelles pertes de paquets par l'une des stations.

Il s'agit de déterminer le nombre d'identités de paquets que doit comporter le QS pour assurer la détection de tous les cas d'erreur.

Examinons le cas suivant où A, B et C sont 3 stations et F_a , F_b , F_c leurs files associées.

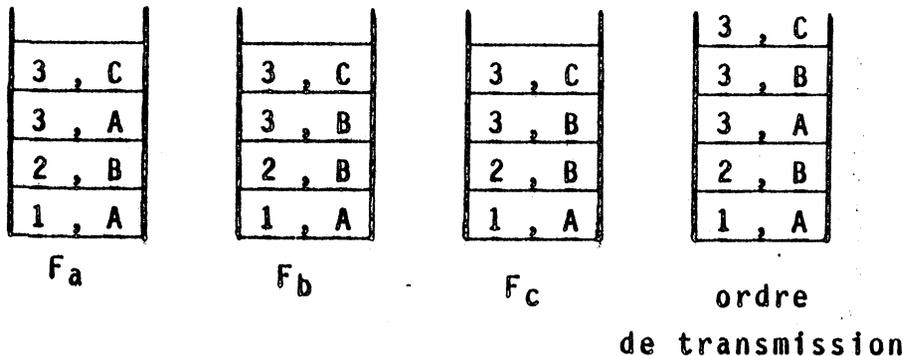
Au temps t où toutes les stations ont reçu les paquets (1,A) et (2,B), elles décident simultanément d'émettre chacune un paquet.

A émet le paquet (3,A) qui sera perdu par les stations B et C,

B émet le paquet (3,B) qui sera perdu par la station A,

C émet le paquet (3,C) qui sera reçu par toutes les stations

En supposant que les 3 paquets sont transmis dans cet ordre, en fin de réception du dernier paquet (3,C), chaque station a perdu un paquet et le nouvel état des files est le suivant :



Si le QS ne décrit que le dernier paquet reçu par la station émettrice, le prochain paquet à émettre sera numéroté 5 et son QS contiendra (3,C) quelque soit son émetteur. A sa réception, les stations procéderont à la comparaison du contenu de leur file au rang $NM-1$ c'est à dire 4 avec le contenu du QS. Elles trouveront toutes (3,C) au rang 4 et aucune d'elles ne détectera la perte de (3,B) par A et celle de (3,A) par B et C.

Pour être certain que la réception, par toutes les stations du paquet émis avec $NM=5$, permette de détecter toutes les erreurs, il faut que le QS décrive les 3 derniers paquets reçus par son émetteur.

De manière générale si N stations décident simultanément d'émettre chacune un paquet, pour s'assurer que toutes les stations ont reçu les N paquets, il faut que le QS du paquet émis après leur transmission contienne les identificateurs des N derniers paquets reçus par la station émettrice.

Théoriquement N peut être égal au nombre de stations dans le réseau. Cela implique que :

- . Une erreur peut être détectée très tard, ce qui retarde la validation des messages et pose à la station de transport le problème de capacité de stockage,
- . Le QS, étant transmis en même temps que le texte, risque d'occuper trop d'espace dans le paquet au détriment des données,
- . Le temps de déroulement de l'algorithme de détection des erreurs peut être assez long à cause du nombre de comparaisons du à la taille de QS.

Pour pouvoir accélérer la détection des erreurs et limiter la taille du QS, tous les paquets reçus avec un numéro NM tel que :

$NM < R - RA$ seront ignorés par l'ensemble des stations

On appellera RA le retard admis.

En l'absence d'erreur toutes les stations ignorent les mêmes paquets. Les stations émettrices de paquets ignorés doivent les

réémettre après renumérotation. En présence d'erreurs, des stations peuvent, jusqu'à la détection de l'erreur, conserver à tort un paquet ignoré par d'autres stations. A la détection de l'erreur ce paquet sera rejeté.

On transmettra donc dans le QS du paquet les identités des RA derniers paquets reçus par l'émetteur.

3.2.2.2 Détection d'une erreur simple

La perte d'un paquet est appelée erreur simple si tous les paquets transmis après l'erreur et avant sa détection sont correctement reçus par toutes les stations.

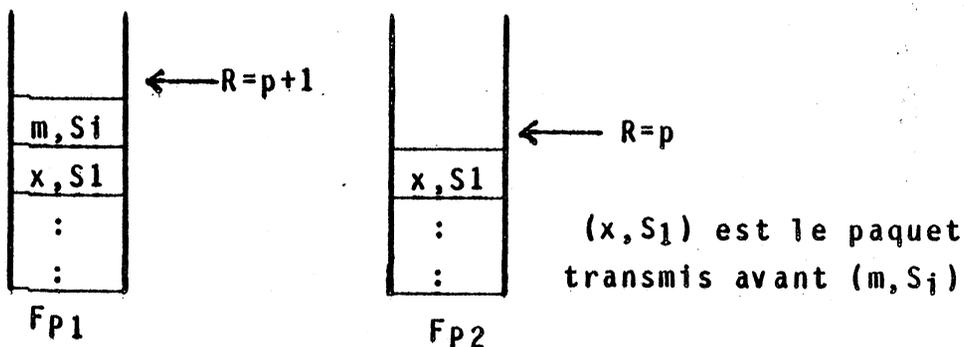
Soit (m, S_i) l'identificateur du paquet perdu.

Après la transmission de (m, S_i) , l'ensemble des stations participant à la conversation se trouve divisé en 2 parties.

P1 : L'ensemble des stations ayant reçu (m, S_i)

P2 : L'ensemble des stations n'ayant pas reçu (m, S_i) .

Les contenus des files des deux ensembles sont les suivants :



La valeur de R (rang de la lière entrée libre dans les files) est égale à $p+1$ pour les stations appartenant à $P1$ et à p pour les stations appartenant à $P2$.

Dans un premier temps pour simplifier la présentation, nous supposons qu'il n'y a pas de compétition, c'est à dire qu'aucune station ne tente de numéroter un paquet en même temps qu'une autre station ou pendant la transmission d'un autre paquet.

La différence entre les files des stations appartenant à P1 et à P2 se situe au rang p. Pour que l'erreur soit détectée il faut qu'un paquet dont le QS décrit le rang p, passe d'un ensemble à l'autre. Le premier paquet dont le QS décrit ce rang est le paquet numéroté p+1. Celui-ci est correctement reçu par toutes les stations (hypothèse: erreur simple).

Soit S_k la station émettrice de ce paquet.

a) La station S_k appartient à l'ensemble P1

$$QS = \begin{array}{|c|} \hline m, S_i \\ \hline x, S_1 \\ \hline : \\ \hline \end{array}$$

A la réception de ce paquet ($p+1, S_k$) toutes les stations appartenant à P2 détectent qu'elles n'ont pas reçu (m, S_j):

b) La station S_k appartient à P2

L'affectation par la station S_k du numéro p+1 signifie qu'une station appartenant à P2 a émis un paquet (m, S_j).

$$QS = \begin{array}{|c|} \hline m, S_j \\ \hline x, S_1 \\ \hline : \\ \hline \end{array}$$

A la réception de ce paquet, toutes les stations appartenant à P1 détectent que l'émetteur n'a pas reçu le paquet (m, S_j).

Supposons maintenant qu'il y a eu compétition c'est à dire que plusieurs paquets de numéros inférieurs ou égaux à p sont reçus. Le premier paquet dont le QS décrit le rang p de son émetteur est porteur d'un numéro NM tel que :

$$p + 1 \leq NM \leq p + RA$$

Soit (NM, S_k) ce paquet. Il est correctement reçu par toutes les stations (erreur simple).

- S_k appartient à P1

La station S_k a reçu le paquet (m, S_j) au rang p .

$$QS = \begin{array}{|c|} \hline : \\ \hline m, S_j \\ \hline x, S_l \\ \hline \end{array}$$

A la réception de ce paquet les stations appartenant à P2 détectent qu'elles n'ont pas reçu (m, S_j) . Car celui-ci appartient au QS du paquet reçu et n'appartient pas à leurs files.

- S_k appartient à P2

La station S_k n'ayant pas reçu le paquet (m, S_j) , l'identité de ce paquet n'apparaît pas dans le QS du paquet qu'elle vient d'émettre.

$$QS = \begin{array}{|c|} \hline : \\ \hline x, S_l \\ \hline : \\ \hline \end{array}$$

Les stations appartenant à P1 détectent qu'il manque (m, S_j) à l'émetteur.

Nous venons de montrer, à l'aide d'une étude exhaustive des différents cas possibles, que les erreurs simples sont toujours détectées quelque soit le cas de figure qui se présente. Une erreur simple est détectée au plus tard après la réception de RA paquets par la station qui a l'erreur.

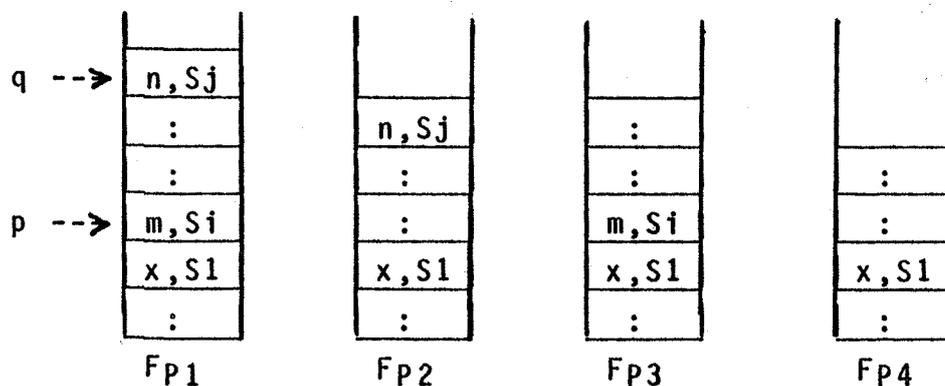
3.2.2.3 Détection d'une erreur double

On appelle erreur double la perte de 2 paquets telle que :

- . La perte du 2ème paquet intervienne avant la détection de la perte du 1er paquet.
- . Tous les paquets transmis après la 2ème erreur jusqu'à la détection de l'une des 2 erreurs soient correctement reçus par toutes les stations.

Soient (m, S_i) et (n, S_j) les 2 paquets perdus. Après la transmission de (n, S_j) , l'ensemble des stations participant à la conversation sera divisé au maximum en 4 groupes.

- P1 : les stations ayant reçu (m, S_i) et (n, S_j)
- P2 : les stations ayant perdu (m, S_i)
- P3 : les stations ayant perdu (n, S_j)
- P4 : les stations ayant perdu (m, S_i) et (n, S_j)



Soient p et q les rangs de passage sur le support de transmission des paquets (m, S_i) et (n, S_j) .

Tous les paquets transmis après (n, S_j) seront correctement reçus par toutes les stations jusqu'à la détection de la perte de l'un des 2 paquets (hyp. erreur double). Le premier paquet permettant de détecter la perte de l'un ou des deux paquets porte un numéro $NM > p$

Ce paquet est transmis après les deux erreurs et est reçu par toutes les stations. Appelons le (NM, S_k) .

A. NM est inférieur ou égal au rang de la 2ème erreur : $NM \leq q$

Le paquet (NM, S_k) a été numéroté avant que (n, S_j) ne soit transmis. Le QS qui l'accompagne, décrit le contenu au rang p de la file de la station émettrice.

a) Si (NM, S_k) est émis par une station appartenant à P_1 ou P_3 toutes les stations appartenant à P_2 et P_4 détectent qu'elles n'ont pas reçu le paquet (m, S_i) . En effet le QS décrit un paquet (m, S_i) qui n'existe pas dans les files des stations de P_2 et P_4 .

b) Si (NM, S_k) est émis par une station appartenant à P_2 ou P_4 toutes les stations appartenant à P_1 et P_3 détectent que la station émettrice n'a pas reçu (m, S_i) car celui-ci figure dans leurs files et est absent dans le QS.

Après récupération de la première erreur on se retrouve dans le cas d'une erreur simple.

B. NM est supérieur au rang de la 2ème erreur : $NM > q$

(NM, S_k) a été numéroté après la transmission des 2 paquets perdus.

1ier cas : $NM - p \leq RA$

Le QS de ce paquet décrit, entre autres, les contenus aux rangs p et q de la file de la station émettrice S_k .

a) (NM, S_k) est émis par une station appartenant à P1.

Le QS contient (m, S_i) et (n, S_j)

- Les stations appartenant à P2 détectent qu'elles n'ont pas reçu (m, S_i)
- Les stations appartenant à P3 détectent qu'elles n'ont pas reçu (n, S_j)
- Les stations appartenant à P4 détectent qu'elles n'ont pas reçu (m, S_i) et (n, S_j) .

b) (NM, S_k) est émis par une station appartenant à P2

Le QS décrit un autre paquet à la place de (m, S_i) et (n, S_j) y figure en retrait d'un rang.

- Les stations appartenant à P1 et P3 détectent que l'émetteur a perdu (m, S_i) .
- En tenant compte du paquet perdu (m .à. j de la file) les stations appartenant à P3 et P4 détectent qu'elles n'ont pas reçu (n, S_j) .

c) (NM, S_k) est émis par une station appartenant à P3

Le QS décrit (m, S_i) au bon emplacement mais ne décrit pas (n, S_j) .

- Les stations appartenant à P2 et P4 détectent qu'elles n'ont pas reçu (m, S_i) .

- Les stations appartenant à P1 et P2 détectent que l'émetteur n'a pas reçu (n, S_j) .

d) (NM, S_k) est émis par une station appartenant à P4

La station S_k a perdu les deux paquets, le QS qui accompagne ce paquet ne décrit aucun d'eux.

- Les stations appartenant à P1 et P3 détectent que l'émetteur n'a pas reçu (m, S_j) .
- Les stations appartenant à P1 et P2 détectent de la même manière que l'émetteur n'a pas reçu (n, S_j) .

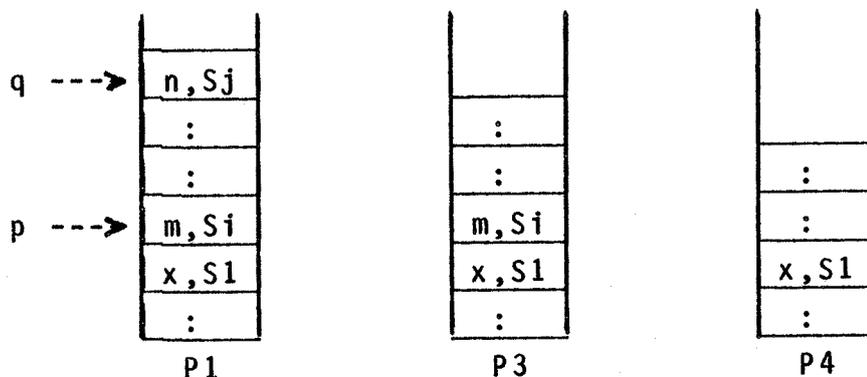
Nous constatons que si le QS du paquet permettant de détecter la première erreur décrit le rang de passage sur le support de transmission de la deuxième erreur, la réception de celui-ci permet de détecter simultanément les 2 erreurs.

2ieme cas : $NM - p > RA$

Le paquet permettant de détecter la perte de (m, S_j) est le deuxième paquet perdu et que l'ensemble des stations P2 est vide.

Si P2 était non vide alors la réception de (n, S_j) (NM, S_k) par les stations appartenant à P1 et P2 aurait permis de détecter la perte de (m, S_j) quelque soit son émetteur P1 ou P2 et après récupération on se serait retrouvé dans le cas d'une erreur simple (perte de (n, S_j) par certaines stations).

La situation avant l'émission de (NM, S_k) était donc la suivante :



Pour porter un numéro supérieur à q , ce paquet ne peut provenir que d'une station appartenant à P1. Les stations appartenant à P3 et P4 doivent d'abord affecter les numéros $q-1$ et q avant le numéro $q+1$ et la réception d'un de ces paquets aurait permis la détection de la première erreur.

La distance p à NM étant supérieure à RA, le QS de ce paquet décrit le rang q mais ne décrit plus le rang p . A sa réception les stations appartenant à P3 et P4 détectent la perte du paquet (n, S_j) . Lors de la réémission de (n, S_j) , les stations appartenant à P4 s'apercevront grâce au QS l'accompagnant qu'elles n'ont pas reçu (m, S_i) .

De la même manière que les erreurs simples, nous venons de montrer que les erreurs doubles sont toujours détectées.

3.2.2.4 Détection des erreurs d'ordre RA.

Définition.

On appelle erreur d'ordre RA, la perte de RA paquets et la réception de tous les suivants jusqu'à la détection de l'une des erreurs.

Détection.

A la perte du $RA^{\text{ième}}$ paquet, l'ensemble des stations est dévisé en $RA+1$ parties de la façon suivante :

P_0 : Les stations n'ayant perdu aucun paquet.

P_1 : Les stations ayant perdu un paquet.

P_2 : Les stations ayant perdu 2 paquets.

:

:

P_{RA} : Les stations ayant perdu RA paquets.

Compte tenu de la complexité due au nombre de combinaisons, nous allons pour simplifier nous intéresser à la détection d'une erreur parmi les RA .

Soit (n, S_j) un des paquets perdus, on peut classer les stations en 2 ensembles

E_1 : Les stations ayant reçu le paquet (n, S_j) .

E_2 : Les stations ayant perdu le paquet (n, S_j) .

Soit (NM, S_j) le paquet émis après la perte des RA paquets.

a) S_j est une station appartenant à E_1 :

A la réception du paquet (NM, S_j) toutes les stations appartenant à l'ensemble E_2 trouveront dans le QS l'identité d'un paquet qu'elles n'ont pas reçu.

b) S_j est une station appartenant à E_2 :

A sa réception par les stations appartenant à E_1 , elles détecteront que l'émetteur n'a pas reçu un paquet sauf si (NM, S_j) est émis par une station appartenant à P_{RA} . Cette station a perdu

les RA paquets, les paquets décrits par le QS du paquet sont antérieur aux RA erreurs. Dans ce cas il faut attendre la réception du prochain paquet émis.

La récupération de l'une des erreurs nous ramène à un niveau d'erreur d'ordre inférieur. On peut procéder ainsi jusqu'à la détection des RA erreurs. Une station ayant perdu au delà de RA paquets consécutifs se voit obliger de quitter la conversation.

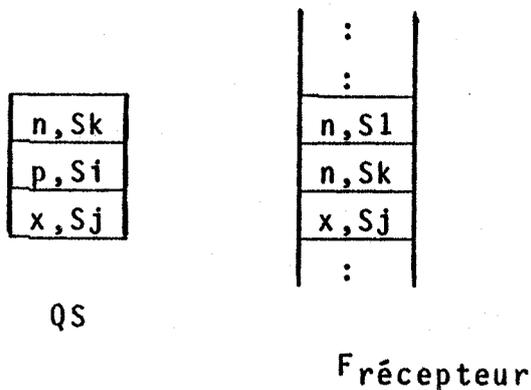
3.2.3 RECUPERATION DES ERREURS

3.2.3.1 Description générale.

A la réception d'un nouveau paquet de type données, chaque station procède à la comparaison du QS accompagnant le paquet avec la partie correspondante à sa file de réception. Au cours de cette comparaison 2 types d'erreur peuvent être détectées par une station.

a) Perte d'un paquet par la station réceptrice

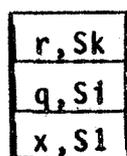
A la réception d'un paquet, la station détecte qu'elle n'a pas reçu un paquet transmis antérieurement. Le QS du paquet reçu décrit un paquet qui ne figure pas dans la file du récepteur.



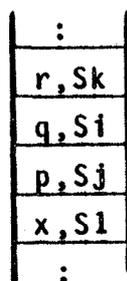
Pour récupérer le paquet perdu, la station va émettre une demande de réémission du paquet ("Selective Reject") à destination de la station émettrice de celui-ci (S_j). Afin que les paquets suivants ne provoquent pas la détection de la même erreur, la station va se mettre en attente de la réémission du paquet perdu ("Repeat Message"). Les paquets reçus entre temps seront mis en file d'attente. La demande de réémission du paquet est répétée sur temporisateur, la station quitte la conversation après plusieurs tentatives sans succès.

b) Perte d'un paquet par le station émettrice

La station détecte que l'émettrice du paquet reçu a perdu un paquet antérieur. Il existe dans la file du récepteur entre les rangs NM-RA et NM-1 un paquet (p, S_j) non décrit par le QS qui accompagne le paquet reçu.



QS



Récepteur

Tous les stations qui ont reçu (p, S_j), détectent l'erreur à la réception de ce paquet. Pour éviter que plusieurs stations ne réagissent en même temps, on propose que seul l'émetteur qui par définition a reçu son propre paquet, manifeste la présence de l'erreur en le réémettant. Ce paquet sera réémis à l'aide de la commande RPM.

Deux types de paquets sont donc utilisés pour la récupération des erreurs.

- Selective reject (SREJ) : pour demander la réémission d'un paquet

- Repeat message (RPM) : pour réémettre un paquet

Ces deux paquets ne sont pas soumis au contrôle d'erreur. La demande de réémission est répétée sur temporisateur jusqu'à réception du paquet ou un nombre maximum de répétitions.

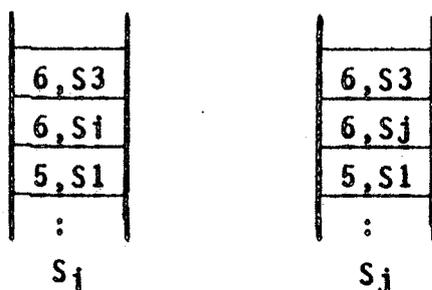
3.2.3.2 Récupération d'une erreur chez le récepteur.

Dans le cas où une station S_j détecte après réception d'un paquet en provenance de la station S_i qu'il lui manque un paquet numéroté p et émis par S_k , elle va émettre à destination de S_k (station émettrice du paquet perdu) une demande de réémission du paquet en précisant:

- Le numéro du paquet perdu : $NM = p$
- Le station destinataire : $SD = S_k$
- Le station émettrice : $SE = S_j$
- L'identité du paquet reçu au rang où devrait l'être le paquet perdu : PR

A la réception de SREJ, la station SD recherche dans sa file F entre les rangs NM et $NM+RA$, le paquet (NM,SD) qu'elle a par hypothèse reçu pour déterminer son rang. Dans le cas où une station est incapable de déterminer l'ordre de passage d'un paquet sur le support de transmission, elle ne le réémet pas. Cette situation peut se produire dans le cas d'une erreur double portant sur 2 paquets transmis consécutivement.

Exemple :



(6, S_i) et (6, S_j) sont tous deux d'origines différentes qui ont été transmis consécutivement sur le support. La station S_j n'a pas reçu (6, S_i) et la station S_i n'a pas reçu (6, S_j). L'examen des files des stations S_i et S_j ne permet pas de déterminer l'ordre de passage des deux paquets sur le support. La consultation d'une station qui a reçu les deux paquets pour rétablir cet ordre peut être coûteuse et son existence n'est pas toujours certaine.

La première station qui détecte l'erreur émet une demande de réémission vers l'autre. Le champ PR permet à la station réceptrice de cette demande de réémission de déceler cette situation et de s'abstenir de réémettre le paquet.

Après plusieurs demandes de réémission du paquet sans réponse, la station quitte la conversation.

3.2.3.3 Récupération d'une erreur chez l'émetteur.

Après avoir perdu un paquet (m, S_i), une station S_j émet le paquet (n, S_j). Tous les récepteurs de ce paquet qui ont reçu (m, S_i) détecteront l'erreur. La station S_i émettrice du paquet perdu le réémet dans un paquet de type RPM dans lequel elle précise :

SE : numéro de station

NM : Numéro du paquet réémis identique au numéro d'origine

SD : Station destinataire du RPM

QS : Liste des RA paquets reçus avant le paquet réémis.

Le QS servira à la station réceptrice du RPM à détecter des erreurs qui auraient pu l'être si ce paquet avait été reçu normalement. L'erreur est récupérée si la station émettrice du paquet perdu existe toujours et a reçu le paquet permettant de la détecter.

a) La station émettrice n'a pas reçu ce paquet.

Cela va augmenter le nombre d'erreurs.

b) Le station émettrice du paquet (m, S_j) est en panne:

Tant que S_j continue à être la seule station à émettre, toutes les stations qui n'ont pas cette erreur détecteront la perte d'un paquet par celle-ci même si le rang de passage sur le cable du paquet perdu ne fait plus partie du QS. N'étant pas émettrice du paquet perdu, aucune station ne réagira. la station S_j détectera l'erreur à la réception d'un paquet en provenance d'une des stations ayant reçu le paquet qu'elle a perdu et quittera la conversation pour impossibilité de récupérer l'erreur.

3.2.3.4 Déconnexion involontaire des stations.

Une station sera amenée à quitter la conversation si un des cas suivants se produit :

a) Impossibilité de récupérer un paquet perdu

. Le station émettrice du paquet perdu a quitté la conversation.

. Le station émettrice du paquet s'abstient de le réémettre (cause désaccord sur les précédents paquets).

- . Paquet non disponible chez son émetteur.

- b) Délivrance de messages dans un ordre différent de celui de leur passage sur le cable.

- c) Perte de plus de RA paquets consécutifs : On ne récupère que les erreurs détectées parmi les RA derniers paquets transmis.

3.2.4 PROBLEMES POSES.

3.2.4.1 Taille des messages limitée.

Chaque station émet son paquet dès qu'il est prêt. Ce fonctionnement pose des problèmes dans le cas où la taille des messages à émettre tient sur plusieurs paquets. Les stations risquent de recevoir dans un ordre quelconque des fragments de messages d'origines différentes. Pour reconstituer ces messages les stations doivent prévoir autant de messages en cours de reconstitution qu'il y a de stations participant à la conversation. Ce mécanisme nous contraint à limiter la taille d'un message à celle d'un paquet.

3.2.4.2 Critère de délivrance non déterminé.

Les messages doivent être remis à tous les membres de la conversation dans le même ordre par chaque station. Cet ordre est celui de leur passage sur le support de transmission. Le mécanisme que nous venons de décrire garantit la détection de toutes les erreurs. Il assure dans la majorité des cas leur récupération et signale dans les autres cas à l'application la présence d'une erreur non récupérée et la station quitte la conversation. Il ne précise aucun délai quant à la détection ou à la récupération de ces erreurs. Cette lacune pose le problème du délai de délivrance des messages par la station à l'application.

La seule démarche sûre consiste, avant de délivrer un paquet, à attendre la réception d'au moins un paquet émis après la transmission de celui-ci en provenance de chacune des stations qui participent à la conversation, afin de s'assurer que la station n'a perdu aucun paquet transmis avant celui-ci.

Cette solution pose évidemment des problèmes :

- Le délai de délivrance des paquets est trop long. Il faut attendre la réception d'au moins N paquets (N = nombre de stations -1).
- Pour pouvoir stocker les paquets reçus entre temps, la station de transport doit disposer d'un nombre de tampons supérieur au nombre de stations participant à la conversation.

Pour pallier en partie ces problèmes, nous pouvons opter pour une solution intermédiaire qui permet de diminuer le délai de délivrance mais présente le risque qu'une station quitte la conversation après s'être aperçu qu'elle a délivré des paquets trop tôt à l'application. La station suppose qu'un paquet est reçu sans erreur si aucun des RA paquets numérotés reçus après celui-ci n'a permis la détection d'une erreur. Le paquet est alors délivré à l'application. Si une erreur portant sur un paquet transmis antérieurement au dernier délivré, est détectée, la station quitte la conversation et avertit l'application de ce problème.

Nous avons procédé à la simulation de cet algorithme dans les conditions de fonctionnement suivantes :

- les caractéristiques du réseau sont celles de DANUBE [LAB.80]
- nombre de stations qui participent à la conversation égal à 20,
- débit de la conversation important, chaque station a constamment un paquet à émettre,
- valeur de RA = 3,
- erreurs
 - . paquets perdus pour manque de tampon de réception
 - . erreurs de transmission 10^{-2}

- critère de dilivrance des paquets : réception de RA paquets.

En imposant aux stations de transport de n'émettre un second paquet qu'après reçu le précédent et de traiter en priorité les paquets reçus par rapport à l'émission, on assiste à une autorégulation du trafic.

Les résultats montrent que :

- une station réémet très peu de paquets à cause du retard (1.4% en moyenne),
- toutes les erreurs sont détectées et récupérées,
- aucune station n'a délivré de paquets trop tôt.

On estime donc que les sorties involontaires des stations ne peuvent produire que dans des cas de très fortes compétitions ou de problèmes matériels.

Cependant, cette solution reste satisfaisante tant que la conversation est bien "alimentée" (il y a de nouveaux paquets qui circulent), les paquets peuvent être délivrés aux applications. Un problème se pose dans les cas d'applications de type interactif. Une station diffuse une question et attend une réponse. Pour délivrer à l'application le texte de cette question, les différentes stations attendent la réception de RA paquets. Nous nous trouvons alors dans une situation de blocage. Pour sortir de cette situation, les stations silencieuses doivent émettre des paquets additifs vides dont le rôle est de faire avancer les files de réception des stations. Ces paquets constituent une charge supplémentaire et inutile.

Nous pouvons déminuer le délai et garantir des livraisons sans erreur en supprimant la compétition. Il s'agit d'obtenir une numérotation des paquets strictement séquentielle. Nous citons trois manières de réaliser cette numérotation.

. Réduction du retard admis RA à 1:

Les stations ignorent les paquets reçus avec :
rang émission < rang réception.

L'inconvénient est que, en cas de trafic intense, un même paquet peut être ignoré et donc réémis plusieurs fois par son émetteur.

. Numérotation des paquets par l'interface réseau :

L'interface dispose de l'état du support de transmission. En supposant qu'il soit possible de la modifier de sorte qu'elle puisse garder le numéro R du dernier paquet reçu et affecter un numéro NM au paquet à émettre, on peut obtenir ainsi une numérotation séquentielle des paquets. L'interface réseau doit alors avant de transmettre un nouveau paquet lui affecter un numéro $NM=R+1$, ce qui revient à lui affecter un numéro qui correspond à son ordre de passage sur le support. En cas de collision, le numéro sera réévalué en fonction du nombre de paquets reçus entre temps. En absence d'erreur tous les paquets portent des numéros consécutifs.

. Attribution aux stations de transport d'un droit de numéroté

Mécanisme qui garantit qu'une seule station numérote à un instant donné et que nous étudierons par la suite.

3.2.4.3 Découpage du réseau en 2 parties non détecté.

Il se peut que, pour des raisons matérielles, le réseau se trouve partagé en 2 ou plusieurs ensembles de stations qui évoluent en parallèle. Ce type de fonctionnement n'est pas détecté avec le contrôle d'erreur que nous venons de présenter. En faisant émettre des paquets d'entretien sur temporisateur par les stations silencieuses, les stations de chaque ensemble considèreront que ceux de l'autre sont en panne suite à un silence prolongé. Ce

mauvais fonctionnement persiste tant qu'aucun paquet ne passe d'un ensemble à l'autre. Ce problème se résoud facilement dans le cas de fonctionnement avec attribution du droit de parole car il n'y aura qu'un seul ensemble qui continuera à évoluer.

3.2.4.4 Difficulté d'adaptation à un réseau à diffusion + une passerelle.

Le contrôle d'erreur que nous préconisons est basé sur l'unicité du médium de transmission, il n'est plus valable à partir de l'instant où les stations qui participent à la même conversation ne font pas partie d'un même réseau (exemple : connexion via TRANSPAC de deux réseaux locaux). L'ordre de réception des paquets peut ne pas être identique au niveau de toutes les stations. Ce problème est résolu dans le cas de fonctionnement avec attribution aux stations d'un droit de parole.

3.3 ATTRIBUTION DECENTRALISEE D'UN DROIT DE PAROLE.

3.3.1 Principe.

Les problèmes rencontrés dans la proposition précédente découlent en général de la présence de plusieurs paquets de même numéro à un instant donné. Le contrôle d'erreur que nous proposons ici est basé sur une numérotation séquentielle des paquets. Pour permettre cette numérotation tout en conservant la multiplicité des émetteurs, on a introduit un mécanisme d'attribution décentralisée d'un droit de parole.

Ce mécanisme garantit qu'une seule station numérote à un instant donné (suppression de la compétition). A l'ouverture de la conversation, c'est la station initiatrice qui possède ce droit de parole. Avant d'émettre un paquet numéroté, tout émetteur doit réclamer ce droit de parole par une demande de parole. La demande de parole est un paquet non numéroté qui peut être émis à tout moment en diffusion. La station qui détient le droit de parole doit le passer à la première station qui l'a demandé dès qu'elle a fini d'émettre le dernier fragment du message courant. La dernière station émettrice garde le droit de parole jusqu'à ce qu'une autre station le lui réclame.

3.3.2 Détection et récupération des erreurs.

On appelle erreur un paquet numéroté qui est envoyé par une station et n'est pas reçu par une ou plusieurs stations destinatrices.

Détection.

Les erreurs sont toujours détectées par les stations destinatrices. La détection est basée sur une numérotation

séquentielle des paquets. Si une station reçoit le paquet de numéro NM sans avoir reçu le paquet numéroté NM-1, c'est qu'elle a perdu le paquet NM-1.

Ce mécanisme permet de détecter l'absence d'un paquet à la réception du suivant. En cas d'arrêt momentané du trafic, la perte du dernier paquet transmis sera tout de même détectée grâce à la répétition, sur temporisateur par la station qui a la parole, du dernier paquet émis. La réception d'un paquet dont le numéro est inférieur au numéro du paquet attendu est ignoré par toutes les stations de transport.

Récupération.

Quand une station a détecté l'absence d'un ou plusieurs paquets numérotés, elle diffuse, sur la conversation, une demande de réémission du paquet portant comme numéro du paquet réclamé le plus petit des numéros de paquets manquants. La demande de réémission est un paquet non numéroté émis sans contrôle d'erreur afin de ne pas perturber la numérotation des paquets soumis au contrôle d'erreur ni la poursuite de l'émission normale de ces paquets. Elle est répétée sur temporisateur jusqu'à la réception du paquet réclamé. Le paquet réclamé est renvoyé par son émetteur. Les paquets répétés sont strictement identiques à leur première version.

La station s'abstient de demander la parole tant qu'elle n'a pas reçu le paquet réclamé. Elle met en file d'attente par ordre croissant de numéros les paquets numérotés reçus entre temps. Ceux-ci seront traités après la réception du paquet attendu. Par contre les paquets non numérotés sont traités normalement. Une station qui reçoit un paquet numéroté en double l'ignore.

Chaque paquet numéroté et émis est conservé par sa station émettrice jusqu'à ce qu'elle soit certaine qu'il est reçu par

toutes les stations. Pour savoir quand un paquet numéroté est bien reçu, les stations émettrices gardent la position du dernier paquet acquitté par chacune des stations participants à la conversation. Cette position est mise à jour à la réception de paquets en provenance des différentes stations. En cas de non trafic, les stations émettent périodiquement des paquets d'entretien. Ces paquets permettent aux stations de redonner du crédit, acquitter une suite de paquets et signaler leur présence.

3.3.3 Avantages et inconvénients.

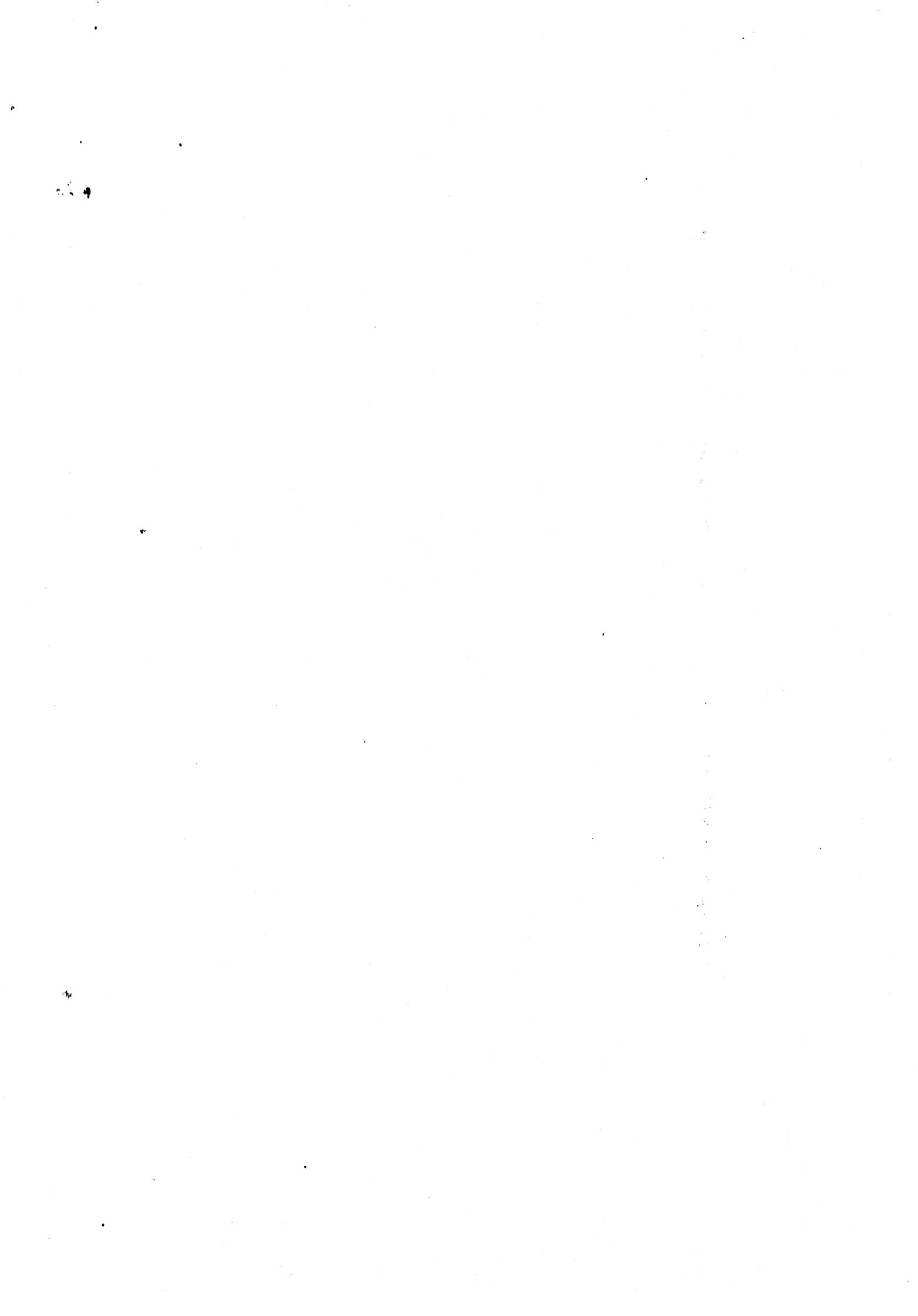
Par rapport à la proposition précédente, cette méthode présente plusieurs avantages :

- Simplicité du mécanisme de détection et de récupération des erreurs.
 - . Une erreur est toujours détectée par la station qui a perdu le paquet à la réception du premier paquet transmis après celui-ci.
 - . Les paquets perdus sont toujours récupérés dans l'ordre.
- Simplicité du critère de délivrance des messages.
- Il y' a moins de risque de déconnexion involontaire d'une station.
- En cas de découpage du réseau en 2 parties, seul l'ensemble des stations auquel appartient la station qui détient à ce moment le droit de parole continuera à évoluer.
- Le mécanisme s'adapte facilement à des applications où les stations concernées appartiennent à réseaux locaux différents reliés par un réseau classique

L'inconvénient de cette méthode est le ralentissement du débit des applications. En effet l'émission par une station d'un paquet numéroté nécessite l'émission d'un paquet de demande de parole et l'attente de l'attribution par une autre station de ce droit. Cet inconvénient apparaît cependant minime lorsqu' on compare le débit réelle des applications à celui des supports de transmission utilisés.

Nous allons, dans ce qui suit, décrire la mise en oeuvre des différents services de ce protocole en se basant sur ce mécanisme de numérotation et de détection et de récupération des erreurs.

4. MISE EN OEUVRE DU PROTOCOLE



4.1 GESTION DU DROIT DE PAROLE.

Chaque fois qu'une station désire émettre un paquet numéroté, elle demande, si elle ne le possède pas déjà, le droit de parole. Pour assurer une distribution équitable du droit de parole, il faut l'attribuer dans l'ordre des demandes. Pour éviter de conserver la liste de toutes les demandes en attente, on diffuse les demandes de parole, et chaque station ayant demandé la parole conserve le numéro de site de la station qui a demandé la parole après elle. Les demandes de parole se font sans contrôle d'erreur, ce qui peut perturber leur prise en compte. Pour garantir qu'elles soient vues dans le même ordre par toutes les stations, on les numérote à l'aide d'une estampille [LAM.78] et on les répète sur temporisateur. Les estampilles générées (numéro de la demande en poids fort, numéro de la station en poids faible) sont uniques si les numéros de stations sont uniques. Cette unicité garantit qu'à une demande correspond une seule affectation.

Chaque station conserve le numéro de la dernière demande de parole NDD. Ce numéro est porté par tous les paquets soumis au contrôle d'erreur (numérotés) sauf les paquets d'attribution de parole qui porte à la place le numéro de parole attribué NPA. Pour chacun de ces paquets reçus, la station compare la valeur du NDD local à la valeur portée par le paquet reçu. La valeur du NDD local est augmentée pour rattraper celle portée par le paquet dans le cas où celle-ci lui est supérieure. La valeur du NDD local est aussi augmentée de 1 à l'émission d'une demande de parole.

Pour obtenir le droit de parole, une station émet un paquet de demande de parole avec une estampille NPD (NDD+1). Elle surveille les nouvelles demandes pour déterminer son successeur. Pour chaque nouvelle demande de parole, elle compare son NPD à celui de la demande reçue. Si l'estampille de la demande reçue est supérieure à la sienne, la station émettrice devient son successeur. Les demandes suivantes sont ignorées à moins qu'elles ne portent une

estampille comprise entre celle de la station et celle de son successeur auquel cas la station émettrice de la demande devient le nouveau successeur. Si une station s'aperçoit que la parole est attribuée à une station ayant une estampille supérieure à la sienne, c'est que sa demande a été perdue, elle doit la réémettre avec une nouvelle estampille.

Ce mécanisme assure que, quelque soit les cas de pertes ou de compétition, une station parviendra toujours à obtenir le droit de parole et qu'une demande n'est satisfaite qu'une seule fois. L'ordre des demandes est respecté sauf en cas de perte et dans certains cas de compétition (demandes simultanées).

Pour diminuer le nombre de paquets nécessaires à un passage de parole, on met dans le paquet de demande le texte qu'on désire émettre. Ce texte est émis dans le paquet d'attribution de parole par la station qui cède le droit de parole pour le compte de la station qui va le recevoir. La station qui reçoit la parole émet tout de suite les paquets suivants sans avoir à réémettre le paquet ayant servi à effectuer la demande de parole.

4.2 CONNEXION.

L'entrée d'un abonné de la station de transport dans une conversation se fait en deux étapes.

4.2.1 Ouverture de la conversation.

L'abonné demande à la station de transport d'ouvrir la conversation en spécifiant s'il est le premier à utiliser le nom de cette conversation.

- Si l'abonné déclare être le premier à demander l'ouverture de la conversation, la station de transport crée le contexte de la

conversation. Elle a le droit de parole et affecte le premier numéro de paquet (NM=0) et le premier numéro de parole (NP=0). La conversation est ouverte. L'entrée d'autres abonnés locaux et l'échange de messages entre des membres locaux sont autorisés. La réception d'un paquet de demande de parole correspond à la demande d'entrée d'une deuxième station de transport dans la conversation. Elle numérote et renvoie ce paquet avec attribution de parole.

- Si l'abonné n'est pas le premier à utiliser ce nom de conversation, la station de transport signale son entrée dans la conversation à moins qu'elle ne le soit déjà pour un autre abonné. Cette entrée va se traduire par la création du contexte de la conversation et par l'émission d'un paquet de demande de parole avec le texte égal (E-S) et le numéro de parole quelconque (0 par exemple). La station gère le protocole de gestion de parole; soit elle reçoit la parole si aucune station n'est en attente de parole, soit elle se voit "sauter" son tour auquel cas elle doit réémettre sa demande avec un numéro correcte. L'entrée de la station dans la conversation est effective à partir de l'instant où sa demande d'entrée est émise (numérotée) par la station qui lui attribue la parole. Ce paquet d'attribution de parole est accompagné de la liste des stations qui participent à la conversation avec leurs crédits respectifs et le nombre de membres. Ce paquet permettra à la station de transport de mettre à jour le contexte conversation en y introduisant la liste des stations et de déclarer la conversation ouverte.

En cas de non réponse à des demandes répétées, la station conclura à l'absence de correspondant et la conversation sera abandonnée.

Le crédit d'une station de transport n'est pris en compte par les autres stations qu'à partir de l'instant où elle gère au moins un membre de la conversation.

4.2.2 Entrée d'un abonné dans la conversation.

En cas de réponse positive à la demande d'ouverture, l'abonné est connu de la conversation mais il n'est encore membre de celle-ci. Dans cet état l'abonné peut demander la liste des membres de la conversation et décider à le devenir à son tour. Si l'abonné désire devenir membre de la conversation, il le signale à la station de transport. Dans sa demande d'entrée, l'abonné précise son crédit (nombre de messages qu'il peut recevoir). Si ce crédit est inférieur au crédit en cours de la conversation, la station de transport refuse son entrée afin de ne pas réduire le débit de l'application. Si le crédit est suffisant, la station émet un paquet numéroté entrée membre (E-M). Sa réception par les différentes stations provoquera l'avertissement de tous les membres de la conversation. Ce paquet peut contenir un texte fourni par l'abonné qui sera délivré à tous les membres. Le nouveau membre ne recevra que les messages émis après son entrée effective.

4.3 DECONNEXION.

4.3.1 Sortie d'un membre de la conversation.

Un membre d'une conversation peut à tout moment demander à quitter celle-ci. La station de transport avertit alors tous les autres membres en diffusant un paquet numéroté (S-M) qui peut contenir un texte fourni par le membre sortant et qui sera délivré aux membres restants. Le membre est supprimé de la liste des membres de la conversation et n'a plus accès aux services de celle-ci.

En cas de panne d'une station, il n'y a pas d'évènement sortie de membre pour les membres qu'elle gère. Les membres de la conversation restants sont avertis de la panne d'une station. Pour

connaître la liste exacte des membres restants, il faut utiliser le service de listage.

4.3.2 Fermeture d'une conversation.

Lorsqu'une station de transport n'a plus de membre à gérer pour une conversation, elle émet un paquet de sortie de station (S-S) pour avertir les autres stations. Si la station n'est pas la dernière à quitter la conversation, elle répète son paquet jusqu'à la réception d'une demande de parole. Elle attribue la parole et lorsque tous les paquets numérotés qu'elle a émis sont acquittés par l'ensemble des stations restantes, elle détruit le contexte conversation.

4.4 EMISSION DE MESSAGES.

Un membre de la conversation peut à tout moment émettre un message dans celle-ci soit en diffusion soit en aparté

Au niveau de la station de transport, les messages sont transportés dans des paquets de données avec le code DONN pour la diffusion ou DONA pour les émissions en aparté. Un message qui ne tient pas dans un seul paquet est découpé en fragments envoyés dans des paquets successifs. Pour indiquer qu'un paquet de données est suivi d'un autre paquet de données d'un même message, la station émettrice positionne un indicateur "fin msg". L'indicateur du paquet qui transporte le dernier fragment d'un message est mis à un.

La station de transport demande le droit de parole avant de transmettre le premier paquet du message. La parole ne peut pas changer de station entre deux fragments d'un même message.

4.5 RECEPTION.

Au moment de son entrée dans la conversation, un membre fournit le nom d'une procédure déclarée par lui. La station de transport déroute le processus abonné vers cette procédure pour soit lui délivrer un fragment d'un message ou d'une liste reçu soit lui signaler l'entrée ou la sortie d'un membre. Les fragments d'un même message sont reçus dans l'ordre par tous les récepteurs alors que ceux d'une liste peuvent être entrecoupés par des messages car ils ne proviennent pas de la même station. Pour signaler aux récepteurs le fin d'une liste ou d'un message, la station de transport positionne un indicateur qui sera testé par ces récepteurs lors du retrait d'un fragment.

4.6 LISTAGE DES MEMBRES.

Un abonné connu de la conversation (après avoir demandé l'ouverture) ou membre de la conversation peut à tout moment demander la liste exacte des membres de la conversation. Pour obtenir cette liste, la station diffuse un paquet numéroté DLST auquel chacune des stations réceptrices va répondre en envoyant un paquet numéroté RLST contenant la liste des membres de cette conversation qu'elle gère. La station délivre à l'abonné la liste des membres de la conversation par fragments au fur et à mesure qu'ils arrivent. Pour lui signaler qu'un fragment constitue le dernier élément d'une réponse, elle positionne de la même manière que pour les messages un indicateur.

Les stations peuvent utiliser le mécanisme de fragmentation et réassemblage pour envoyer une liste qui ne tient pas dans un seul paquet.

4.7 CONTROLE DE FLUX.

Le contrôle de flux se base sur un mécanisme de crédit.

Chaque membre de la conversation fournit un crédit (nombre de messages qu'il est prêt à recevoir) au moment de son entrée dans la conversation. La station de transport décrémente ce crédit de 1 à chaque fois qu'elle lui délivre un message. Les membres d'une conversation disposent d'une primitive pour augmenter tout moment leur crédit.

La station tient à jour pour chaque conversation deux crédits :

- Le crédit en réception : Ce crédit est égal au minimum des crédits fournis par les membres locaux de cette conversation. Il est décrémente de 1 chaque fois que la station reçoit un paquet numéroté dans cette conversation. Dans le cas où la station n'a pas encore de membres locaux à gérer pour, le crédit en réception de celle-ci est considéré illimité.

- Le crédit en émission : Chaque station indique à l'ensemble des autres stations le nombre de paquets supplémentaires CR qu'elle peut recevoir dans cette conversation à partir du dernier reçu NR. Le crédit en émission d'une station pour une conversation est égal au minimum des crédits fournis par les différentes stations qui participent à celle-ci et de son crédit en réception. La station qui a la parole ne peut émettre un paquet que si ce crédit est positif. Le crédit d'une station est transmis aux autres soit dans un paquet numéroté soit dans un paquet d'entretien. Chaque station émet sur temporisateur un paquet d'entretien dont lequel elle indique pour chaque conversation qu'elle gère le numéro du dernier paquet reçu et le nombre de paquets qu'elle est prête à recevoir.

Le crédit spécifié dans un paquet d'attribution de parole appartient à la station qui attribue la parole et non à celle qui la reçoit.

Un paquet portant un numéro supérieur au crédit autorisé dénote une violation du protocole et entraîne la sortie de la conversation des stations ayant détectée cette violation.

4.8 DETECTION DES PANNES.

On entend par panne la disparition d'une station. Cette disparition peut être due à une panne du réseau provoquant l'isolement de la station ou une panne de la station elle-même.

La panne d'une station implique la sortie involontaire de certains membres de la conversation. Le protocole de conversation doit permettre à l'application de s'arrêter lorsque les membres encore présents ne répondent plus à ses besoins. Avec le mécanisme d'acquiescement négatif, les stations silencieuses n'émettent rien en absence d'erreur. Il faut donc produire périodiquement des paquets d'entretien pour permettre à celles-ci de manifester leur présence et de redonner un crédit. Chaque station vérifie périodiquement qu'elle a reçu au moins un paquet d'entretien de chacune des autres stations.

La panne d'une station est détectée par les autres stations par l'absence de paquets en provenance de celle-ci. Une panne de station peut conduire à la perte du droit de parole.

- La station tombe en panne après avoir émis une demande de parole. La station qui lui attribue le droit de parole, répète sur temporisateur le paquet attribution de parole. Elle considère que son successeur est en panne après plusieurs répétitions et elle reprend le droit de parole.

- La station qui a la parole tombe en panne avant de l'attribuer à son successeur. Ce cas nécessite la concertation des stations restantes pour l'attribution d'un nouveau droit de parole. Une

manière de réattribuer la parole de manière unique serait la suivante : lorsqu'une station détecte qu'elle répète sans succès sa demande de parole, elle émet une demande avec attribution en acquittant le dernier paquet reçu. Elle réémet plusieurs fois ce paquet sur temporisateur. Si entre temps, elle reçoit des demandes attributions concurrentes, elle résout la compétition de la manière suivante : la station acquittant le paquet de numéro maximum avec un numéro de site maximum l'emporte.

Chaque station avertit les membres de la conversation qu'elle gère lorsqu'une panne de station se produit.

5. EVALUATION DU PROTOCOLE

5.1 COMPARAISON AVEC LE PROTOCOLE POINT A POINT

Cette comparaison sera basée sur l'évaluation du nombre de paquets transmis sur le réseau pour un même service et dans les mêmes conditions de fonctionnement par la station de transport à diffusion et la station de transport point à point. Nous effectuerons cette comparaison avec 3 types de services différents :

- Diffusion d'un fichier
- Interactif (Question / réponse)
- Conversation (système réparti)

Nous allons prendre comme station de transport point à point une station qui utilise comme protocole le protocole PAR (Positive Acknowledgement Retransmission) [SUN.75] que nous rappelons brièvement ici.

La station émettrice numérote les paquets qu'elle émet et arme une horloge de garde avec un délai D , à l'émission de chaque paquet. La station réceptrice acquitte chaque paquet en renvoyant le numéro du paquet acquitté à condition que le paquet et les précédents aient été reçus sans erreur, sinon elle n'envoie pas d'acquiescement. La station émettrice réémet le paquet s'il n'est pas acquitté au bout du délai de garde. La station réceptrice alloue des crédits à l'émettrice, si ce crédit est nul, il y a arrêt d'émission jusqu'à la réception du nouveau crédit.

Pour simplifier, nous supposons que :

- il n'y a pas d'erreur de transmission,
- les connexions sont déjà établies, c'est à dire que nous ne faisons pas intervenir les paquets nécessaires à leur établissement. En point à point, le nombre de connexions est fonction du nombre d'abonnés et du type d'application. Par con-

tre dans le cas d'utilisation d'une station à diffusion, les abonnés qui participent à une même application sont regroupés au sein d'une même conversation et sont désignés à travers le nom de celle-ci quelque soit leur nombre et le type d'application.

- toutes les abonnés participants ont le même crédit,
- les acquittements et les crédits sont transmis dans les mêmes paquets.

Nous appellerons : S le nombre de stations,
C le crédit des abonnés,
A le nombre d'abonnés par station.

5.1.1 Diffusion d'un fichier.

- Utilisation d'une station point à point.

La diffusion d'un fichier à l'aide d'une station point à point consiste à émettre une copie vers chaque abonné destinataire. Ce qui nécessite au préalable l'établissement d'une connexion entre l'abonné émetteur et chacun des abonnés récepteurs.

le nombre total de connexions est donc égal à $A(S-1)+A-1 = AS-1$

Le transfert d'un fichier de taille T paquets vers un abonné destinataire nécessite la transmission de p paquets tels que :

$$p = T+T/C$$

T = nombre de paquets de données

T/C = nombre de paquets d'acquittements émis par la station destinatrice

Pour transmettre une copie du fichier vers chaque abonné destinataire, il faut répéter autant de fois l'opération précéden-

te qu'il y en a de destinataires , c'est à dire (AS-1) fois. Le nombre total P1 de paquets transmis est donc :

$$P1 = (AS-1)p = (AS-1)(T+T/C)$$

- Utilisation d'une station à diffusion.

A l'aide de cette station chaque paquet émis sur le réseau est reçu par toutes les autres stations. Pour transmettre un paquet à l'ensemble des membres de la conversation, il suffit que chaque station qui participe à cette conversation ait reçu ce paquet. C'est elle qui se chargera de le remettre aux membres qu'elle gère.

Dans le cas d'une diffusion de fichier, nous avons une station émettrice et S-1 stations réceptrices. Les stations réceptrices n'utiliseront le réseau que pour émettre des paquets d'entretien pour renouveler le crédit.

La diffusion d'un fichier de taille T vers tous les membres de la conversation nécessite :

- L'émission de T paquets de données par la station émettrice,
- Chacune des S-1 stations réceptrices émet un paquet d'acquittement toutes les fois qu'elle a reçu C paquets de données. Nous aurons au total (S-1)T/C paquets d'acquittement.

Au total, la diffusion du fichier nécessite :

$$P2 = T+(S-1)T/C \text{ paquets.}$$

Nous avons supposé que la station émettrice détient le droit de parole sinon il faut prévoir un paquet supplémentaire de demande de droit de parole.

- Remarque.

En point à point le nombre de paquets de données et le nombre de paquets d'acquittement transmis augmentent avec le nombre de récepteurs quelque leur répartition géographique, alors qu'en diffusion le nombre de paquets de données est fixe et égal à la taille du fichier quelque soit le nombre de récepteurs et le nombre de paquets d'acquittement augmente avec le nombre de stations participantes. Ce qui signifie que le gain en nombre de transmis en utilisant le protocole de conversation serait plus important dans les cas de fonctionnement où :

- le crédit des récepteurs est important
- un grand nombre de récepteurs est réparti sur peu de stations.

La figure 1 montre le nombre de paquets transmis, en fonction du nombre de stations, pour diffuser un fichier de 100 paquets de données suivant qu'on utilise le point à point ou la diffusion. Sur chaque courbe sont indiqués le nombre d'abonnés par station ($A=1, 2$) et leur crédit C . Un crédit de 1 correspond à un acquittement par paquet et un crédit infini au cas où il n'y a pas de trafic d'acquittement.

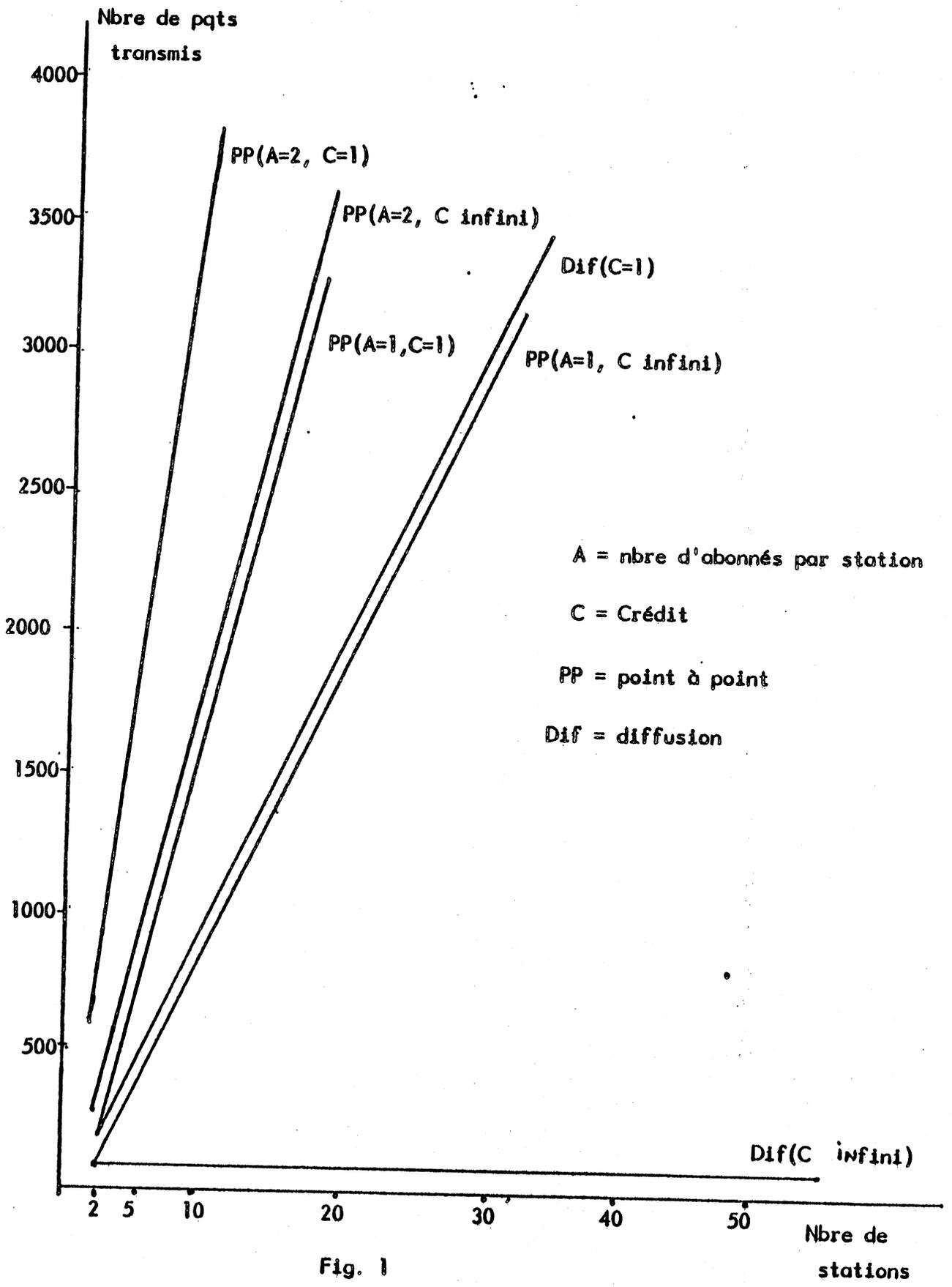


Fig. 1

Diffusion de fichier

5.1.2 Fonctionnement en interactif.

Nous supposons qu'un abonné (membre) quelconque parmi les AS diffuse une question à tous les autres et attend leurs réponses. Il s'agit d'évaluer le nombre de paquets transmis sur le réseau jusqu'à la réception par cet abonné (membre) de toutes les réponses. Nous supposons, pour simplifier, que la question et toutes les réponses tiennent chacune sur un paquet.

- Utilisation d'une station de transport point à point.

L'abonné émet une copie de la question à chacun des autres abonnés. Cette opération nécessite la transmission de AS-1 paquets.

A la réception de la question, chacun des abonnés récepteurs émet une réponse. L'émission par la station de transport du paquet contenant la réponse de l'abonné permet d'acquitter la réception du paquet question. La réception de toutes les réponses équivaut à la transmission de AS-1 paquets.

Au total l'abonné recevra toutes les réponses au bout d'un échange de P1 paquets tels que :

$$P1 = 2(AS-1)$$

- Utilisation d'une station de transport à diffusion.

A l'aide d'une station à diffusion, la diffusion de la question et la réception de toutes les réponses se dérouleront comme suit :

- Sur demande du membre, la station de transport diffuse une demande de parole (1 paquet),
- le paquet d'attribution de parole entraîne la réception de la question par toutes les stations et sa délivrance à tous les autres membres (1 paquet),

- le membre qui a posé la question doit recevoir $A(S-1)$ réponses en provenance des membres distants et $A-1$ réponses en provenance des membres se trouvant sur la même station.

Le nombre total de réponses est donc égal :

$$AS-A+A-1 = AS-1$$

- chaque réponse nécessite la transmission de 2 paquets (1 demande de parole et 1 attribution de parole qui contient la réponse).

Le membre recevra toutes les réponses après un échange de $P2$ paquets tels que :

$$P2 = 2+2(AS-1) = 2AS$$

- Remarque.

Nous constatons sans tenir compte des paquets d'acquittement que, dans ce type de fonctionnement, la station de transport point à point requiert un nombre de paquets inférieur à celui de la station de transport à diffusion.

Cependant, la différence n'est que de deux paquets et l'utilisation d'une station de transport à diffusion évite la gestion de $AS-1$ connexions. De plus chacune des stations est au courant de la réponse des autres; ce qui n'est pas le cas du point à point. Si nous voulons obtenir le même résultat à l'aide d'une station de transport point à point, il faudrait $AS(AS-1)/2$ connexions d'où un nombre de paquets transmis beaucoup plus important.

Le nombre de paquets d'acquittement est fonction du nombre de paquets numérotés. Parmi les $2AS$ paquets transmis, AS sont numérotés, chaque station en a émis A qui ont entre autre servi à acquitter AC paquets et en a reçu $AS-A$. Le nombre d'acquittement

émis par chaque station au cours de cette opération est donc égal à $(AS-A-AC)/C$. L'ensemble des stations en ont émis :

$AS(S-1-C)/C$ paquets d'acquiescement

Ce nombre n'est significatif que si $S > C+1$

5.1.3 Conversation.

Nous nous plaçons dans le cas où chaque abonné doit transmettre un même message à tous les autres et nous nous intéressons au nombre de paquets transmis sur le réseau à l'aide de chaque station après que chaque abonné ait transmis son message. Nous supposons que les décisions d'émettre sont réparties dans le temps, c'est qu'il n'y a pas de concurrence pour l'obtention du droit de parole.

- Utilisation d'une station point à point.

La diffusion d'un message par un abonné se traduit par l'émission d'une copie de ce message à chacun des AS-1 destinataires. Si le crédit C est égal à 1, chaque destinataire émet un paquet d'acquiescement (AS-1).

Pour que chacun des AS abonnés diffuse son message, il faut un échange de P1 paquets tels que :

$$\begin{aligned} P1 &= 2AS(AS-1) \text{ si } C=1 \\ &= AS(AS-1) \text{ si } C \text{ est infini} \end{aligned}$$

- Utilisation d'une station à diffusion.

Pour chaque station, la diffusion du message d'un des membres locaux de la conversation qu'elle gère se déroule ainsi :

- Emission d'une demande de parole (1 paquet),
- la réception du droit de parole correspond à la diffusion du message (1 paquet).

Chaque station va donc émettre $2A$ paquets. Sachant qu'il y a S stations, il faut pour faire le tour de "table", sans tenir compte des paquets d'acquittement, P_2 paquets tels que :

$$P_2 = 2AS$$

Le nombre de paquets d'acquittement transmis au cours de cette opération est fonction du nombre de paquets numérotés (données) transmis. Parmi les $2AS$ paquets transmis, AS sont numérotés, chaque station en a émis A qui ont entre autre servi à acquitter AC paquets et en a reçu $AS-A$. Le nombre d'acquittement émis par chaque station est donc égal à $(AS-A-AC)/C$. L'ensemble des stations en ont émis :

$$AS(S-1-C)/C \text{ paquets d'acquittement}$$

Ce nombre est égal à 0 si $S \leq C+1$.

- Remarque.

Les courbes des figures 2 et 3 montrent suivant les valeurs de A ($=1, 2$) et C ($=1, \text{infini}$) le nombre de paquets transmis par chacune des méthodes en fonction du nombre de stations et du nombre d'abonnés concernés par station dans le cas d'une application de type conversation.

Nous constatons que le gain en nombre de paquets transmis en utilisant le protocole à diffusion augmente avec le nombre de stations. Avec un abonné par station, ce gain est sensible dès que le nombre de stations dépasse 2. Avec plus d'un abonné par station, le gain est plus important et se fait sentir même avec deux stations.

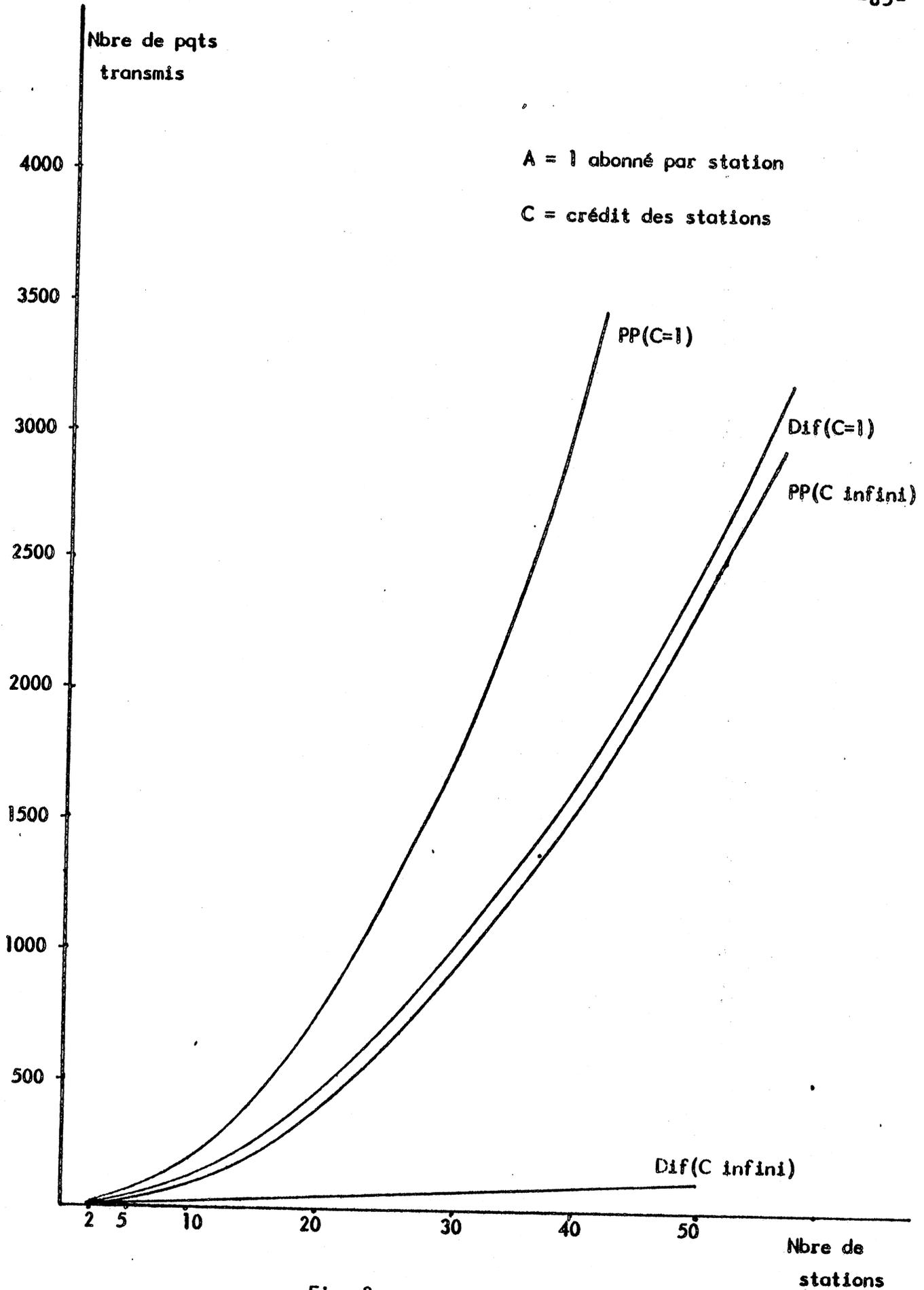


Fig. 2

Conversation (A=1)

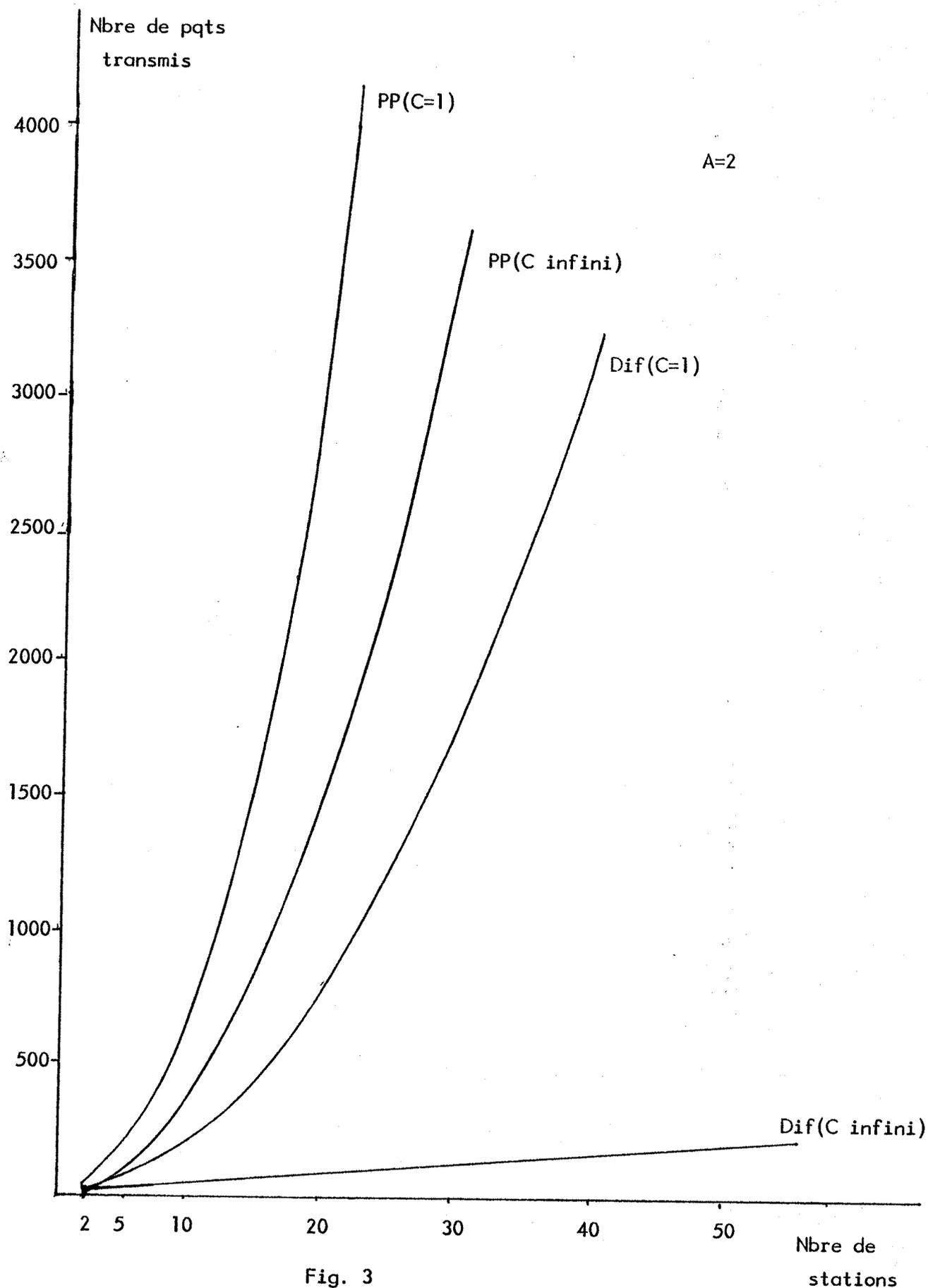


Fig. 3

Conversation (A=2)

5.2 SIMULATION DU PROTOCOLE.

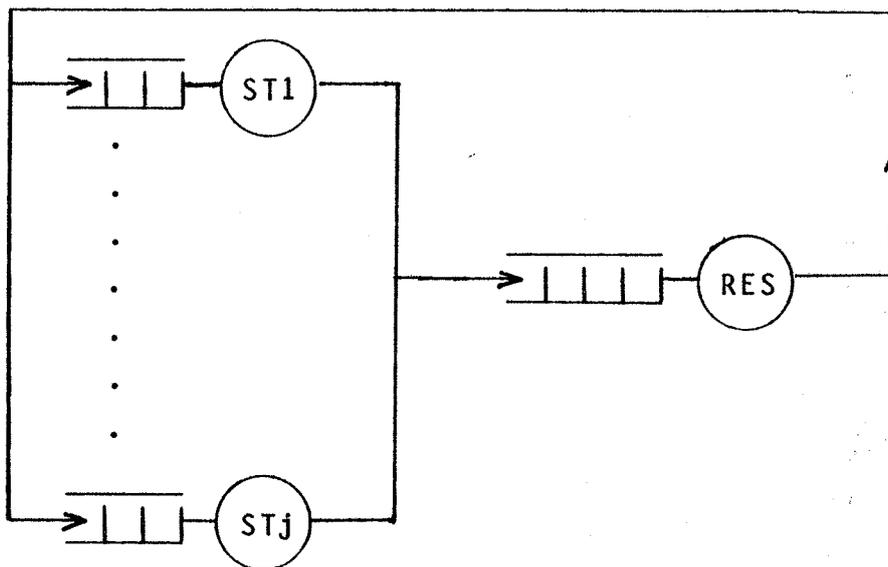
L'objectif, ici est de décrire sous la forme d'un modèle et de simuler le protocole en vue de détecter des cas de mauvais de fonctionnement et de déterminer les débits de deux types d'applications (diffusion de fichier et conversation).

5.2.1 Description du modèle

Dans la description du modèle interviennent deux aspects qui sont :

- la configuration qui est matérialisée par
 - . le nombre et les caractéristiques des calculateurs impliqués
 - . les caractéristiques du support de transmission
- La charge qui est représentée par le comportement de l'application mise en oeuvre.

Le modèle a été représenté en terme de réseau de files d'attente décrit et résolu à l'aide du logiciel QNAP [VER.79]



Chaque station de transport est représentée par une station ST_j (serveur) et le support de transmission est représenté par la station RES.

a) Comportement des stations de transport.

Le service de chaque station de transport est représenté par l'ensemble des processus décrit en annexe que nous résumons brièvement ici.

- Une station ne peut émettre un paquet numéroté que si elle détient le droit de parole et le crédit de la conversation est positif. Le droit de parole est réclamé par un paquet de demande de parole non numéroté.
- Le protocole de gestion de parole permet à une station de déterminer son successeur (station à qui elle attribuera la parole).
- Un paquet perdu est réclamé par une demande de réémission.

- Chaque station garde les paquets numérotés qu'elle a émis (sauf les attributions de parole) jusqu'à ce qu'ils soient acquittés par l'ensemble des stations. Si le nombre de ces paquets atteint un seuil, la station ne peut pas émettre d'autres paquets jusqu'à l'acquiescement par l'ensemble des stations d'au moins un paquet.
- Trois types de paquets peuvent être émis par une station :
 - Des paquets de demande de parole
 - des paquets de données (+ attributions de parole)
 - des paquets d'entretien
- Les acquiescements et les crédits d'une station sont transportés soit dans des paquets numérotés soit dans des paquets d'entretien (ACK).

Etat initial :

Nous supposons qu'il y a une seule conversation en cours et que les connexions sont déjà établies.

Le droit de parole est détenu par une station quelconque

Chaque station dispose d'un crédit (Nombre de paquets numérotés qu'elle peut recevoir)

b) Comportement de la station RES.

La station RES représente le support de transmission. Dans la simulation, nous avons utilisé le câble DANUBE dont le débit est de 1 Mbits/seconde. Nous avons supposé que le taux de compétition est négligeable. Le rôle du câble est de transporter les différents paquets émis par les stations de transport. Le service de cette station est représenté par un temps de service (retard) égal au temps de transmission d'un paquet.

5.2.2 Les applications

Nous avons simulé deux types d'applications qui peuvent tirer profit de la diffusion.

a) Diffusion de fichier

Parmi l'ensemble des stations de transport connectées, une station diffuse des paquets de données et toutes les autres ne font que recevoir ces paquets. Ces stations réceptrices diffusent périodiquement des paquets d'entretien pour renouveler leur crédit et/ou acquitter une suite de paquets.

Dans cette application, deux types de paquets sont échangés entre les stations : des paquets de données générés par la station émettrice et des paquets d'entretien générés par les stations réceptrices.

b) Conversation

On entend conversation toutes les applications où il y a plusieurs émetteurs et plusieurs récepteurs telles que le transactionnel coopératif et les systèmes distribués. Le droit de parole change souvent de station. Dans les simulations, nous avons supposé que la parole change de station après chaque fin de message.

Trois types de paquets sont échangés entre les stations pour cette application :

- Les paquets de demande de parole,
- les paquets de données (+ attribution de parole),
- les paquets d'entretien.

5.2.3 Paramètres de simulation

Les paramètres de simulation sont les suivants :

- Nombre de stations S : 2, 5, 10, 20, 30, 50
- Longueur d'un message l : 1, 5 paquets de données
- Longueur d'un paquet de données : 250 caractères
- Longueur d'un paquet d'entretien : 40 caractères
- Temps de transmission d'un pqt de données : 2×10^{-3} secondes
- Temps de transmission d'un pqt d'entretien : 3×10^{-4} secondes
- Délai d'émission des paquets d'entretien : 10^{-2} , 1 secondes
- Taux d'erreur : 10^{-1} (1 paquet sur 10 est perdu par certaines stations, ce taux est choisi volontairement grand afin de passer en revue le maximum de cas possibles)
- Crédit des stations C : 1, 10 paquets
- Seuil des paquets non acquittés = Crédit de la conversation

5.2.4 Résultats

a) Evaluation qualitative

La première série d'expérimentations que nous avons réalisé, consiste à évaluer la fiabilité des différents algorithmes permettant la mise en oeuvre des services offerts par ce protocole.

La première constatation concernant les résultats obtenus est que l'envoi des paquets d'entretien uniquement sur temporisateur posait un problème au niveau de l'écoulement du trafic.

- Si le délai d'envoi de ces paquets est long, la station qui détient le droit de parole se trouve souvent bloquée durant un certain temps pour manque de crédit sur la conversation. Ce temps de blocage est plus important lorsque le crédit de la conversation est petit.

- Par contre si ce délai est court et que le nombre de stations qui participent à l'application est important, nous constatons un ralentissement du débit de l'application. Les paquets d'entretien s'accaparent la bande passante.

Pour remédier à ce phénomène, nous avons modifié la stratégie d'envoi de ces paquets d'entretien. Chaque station tient à jour son crédit et celui de la conversation. Elle décrémente son crédit de 1 chaque fois qu'elle reçoit, d'une autre station, un paquet numéroté. Elle peut ainsi déterminer si le crédit qu'elle fournit est consommé. Lorsque c'est le cas, elle doit le renouveler en le transmettant soit dans un paquet d'entretien soit dans un paquet de données si elle détient le droit de parole. Les stations silencieuses continuent cependant à émettre sur temporisateur des paquets d'entretien pour manifester leur présence. Dans ce cas on peut fixer le délai à une valeur plus grande (2 secondes).

La simulation du protocole moyennant cette modification a montré dans les deux cas d'applications, un meilleur écoulement du trafic. En diffusion de fichier, le trafic se déroule par saccade (jet de paquets de données suivi d'un jet de paquets d'entretien) lorsque tous les récepteurs ont fourni le même crédit. Ce phénomène disparaît lorsque les capacités des récepteurs sont différentes.

Dans les deux cas d'applications, nous n'avons pas déceler de situation de "deadlock" ni de bouclage infini dus aux algorithmes. Le trafic s'écoule au rythme des récepteurs. Toutes les erreurs sont détectées et récupérées immédiatement.

Dans le cas de la conversation, le droit de parole est réparti de manière équitable et dans l'ordre des demandes entre les différentes stations. Nous jugeons que, même en cas de forte compétition, les stations arrivent toujours à obtenir le droit de parole, il n'y a pas de risque de famine.

b) Evaluation quantitative

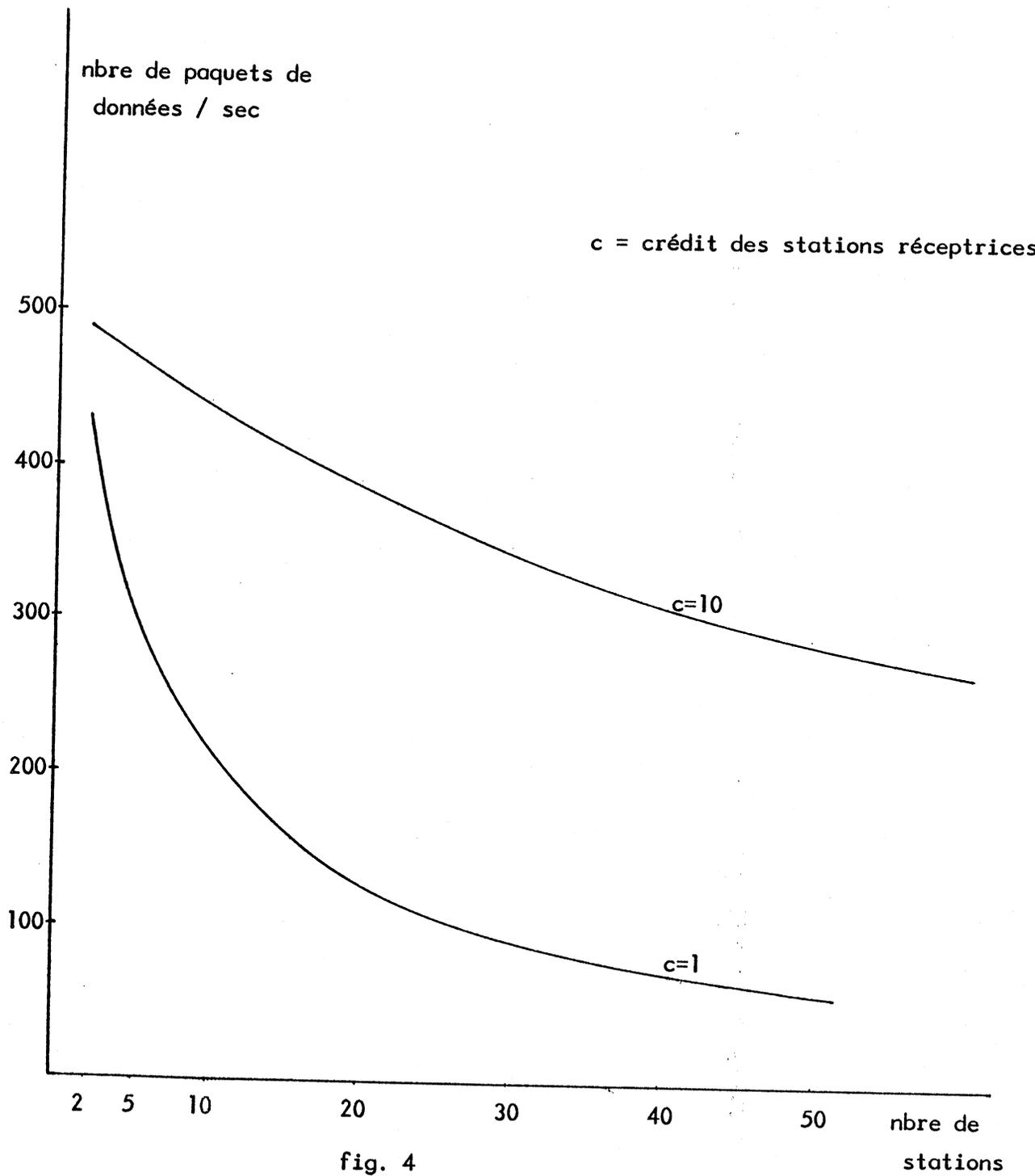
La seconde serie d'expérimentations auxquelles nous avons procédé a pour but d'évaluer les débits maximum des 2 types d'applications. Nous avons supposé que :

- les erreurs de transmission sont négligeables (taux d'erreur=0),
- il y'a une seule conversation en cours,
- il y'a un seul membre de la conversation par station.

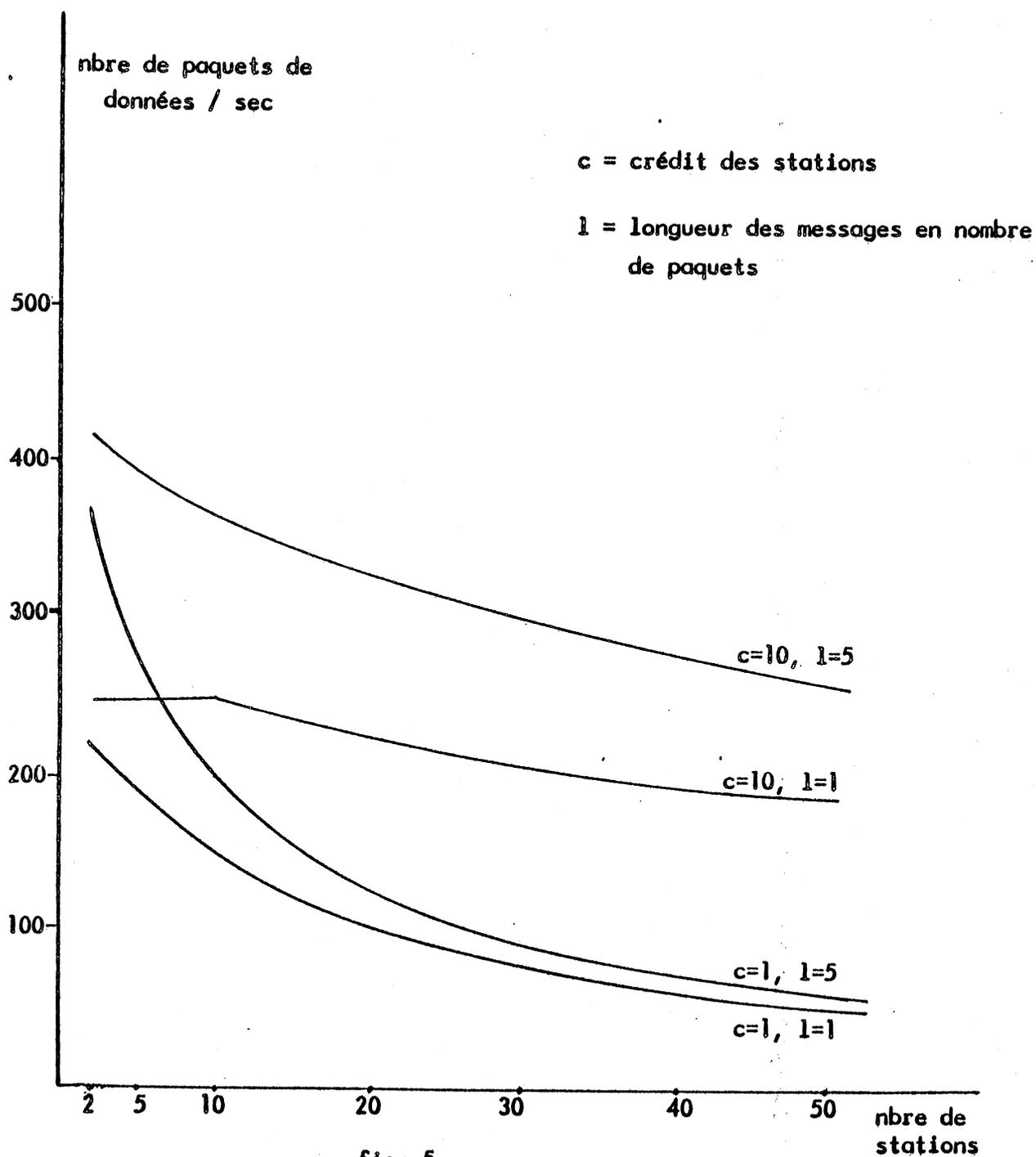
Les courbes présentées (fig. 4 et 5) résumant les résultats obtenus. Elles expriment le nombre de paquets de données transmis par seconde en fonction du nombre de stations.

En plus du nombre de stations, nous constatons que le débit dans le cas d'une diffusion de fichier (fig. 4) dépend de la capacité des récepteurs. Un crédit assez grand implique une déminution de paquets d'entretien en provenance des récepteurs au profit des paquets de données.

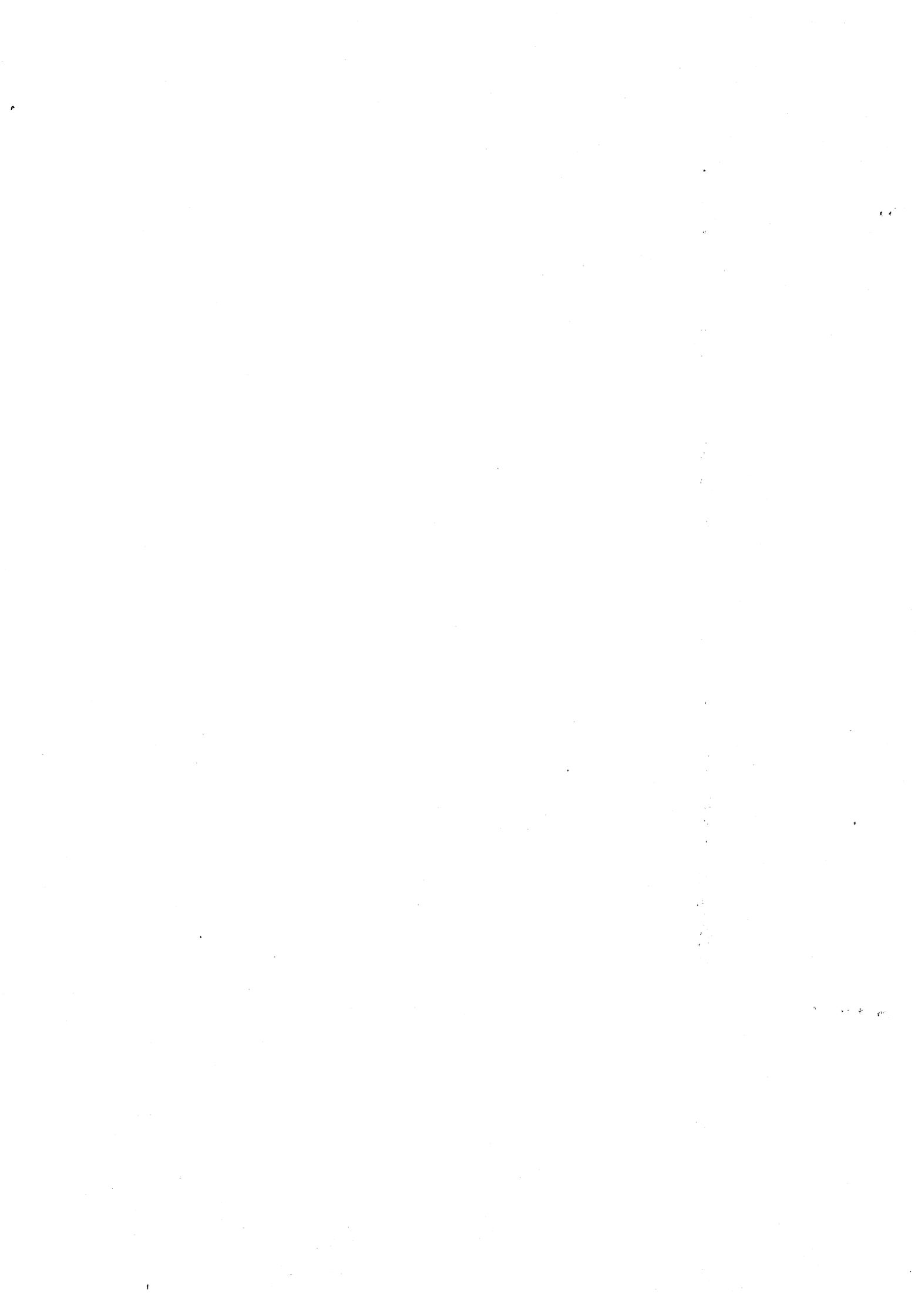
Dans le cas d'une conversation, le débit de l'application (fig. 5) dépend de deux facteurs qui sont : le crédit de la conversation et la longueur des messages échangés. Les messages longs diminuent la proportion des paquets de demande de parole.



Diffusion de fichier



6. CONCLUSION



Dans cette thèse, nous avons tenté d'allier la performance des techniques de diffusion à la fiabilité des techniques de transport point à point. Notre travail a donc porté sur la définition de mécanismes permettant de d'offrir aux utilisateurs des réseaux locaux un service de diffusion fiable.

Après une synthèse sur les réseaux locaux et une brève présentation du réseau DANUBE, nous avons défini un protocole de conversation permettant d'utiliser la propriété de diffusion dans les applications implémentées sur les réseaux locaux. Nous avons d'abord décrit ce protocole du point de vue services offerts à l'utilisateur (connexion, transfert, ...) puis de façon détaillée les mécanismes mis en oeuvre pour assurer ces services.

Deux méthodes de détection et de récupération des erreurs ont été étudiées et comparées. Les deux méthodes reposent sur la numérotation des paquets. La première s'appuie sur l'utilisation de l'ordre de passage des paquets sur le support de transmission et la seconde sur un algorithme décentralisé qui assure sur la répartition d'un droit de parole entre les différentes stations et permet ainsi d'éviter des paquets de numéros identiques et de simplifier les mécanismes de détection et de récupération des erreurs.

La mise en oeuvre du protocole en retenant la seconde méthode est décrite. Les problèmes de contrôle de flux et de détection des pannes de stations sont étudiés.

La comparaison de ce protocole avec le protocole de connexion point à point montre bien les gains en performance que son utilisation peut apporter dans les applications de types transactionnels coopérants, diffusion de fichiers, messagerie,

Ce protocole est enfin simulé. Deux séries d'expériences ont été menées. La première série a permis de tester la validité des algo-

rithmes proposés, les résultats obtenus ne font apparaître aucun cas de mauvais fonctionnement. La deuxième série donne une idée des débits maximum que peut assurer ce protocole sur un réseau ayant les caractéristiques du réseau DANUBE.

Les solutions que nous avons proposées reposent sur quelques possibilités offertes par le réseau local diffusion DANUBE. Elles peuvent en fait s'appliquer à la plupart des réseaux à diffusion car elles ne nécessitent qu'un service minimum de la part du réseau. Les seuls services demandés sont : tramage des données, émission et réception des trames et abandon des trames erronées.

Un aspect intéressant de ce protocole est que le protocole point à point est un cas particulier du protocole multipoints. Aucune des hypothèses sur le support de transmission n'est invalide quand on utilise une ligne de télécommunication classique. Il est donc possible de n'implémenter qu'un seul protocole de transport pour offrir à la fois des services point à point et des services multipoints.

Ce travail n'étant qu'une ébauche de solution, il laisse toutes les voies ouvertes. Nous pensons que dans l'immédiat les points suivants constituent une suite logique :

- Nous avons eu recours à la simulation pour vérifier le bon fonctionnement de ce protocole, son implémentation sur un réseau réel permettrait de le tester en vraie grandeur sur des applications effectives et de l'évaluer plus complètement.

- le protocole que nous avons développé s'adresse aux réseaux locaux à diffusion. Il serait intéressant vu l'évolution des réseaux satellites d'étudier les possibilités d'adaptation de ce protocole à ces derniers.

A N N E X E

A. FORMAT DES PAQUETS ECHANGES.

Le format des paquets échangés entre les différentes stations est inspiré de celui des paquets de la station de transport actuelle du réseau DANUBE [ROU.79]. Les paquets se composent d'une partie spécifique à chacun d'eux et d'une partie commune.

La partie commune comporte :

- le numéro de la station destinataire (SD), égal à 255 car tous les paquets sont en diffusion,
- le numéro de la station émettrice (SE),
- le nom réseau de la conversation (N. CV : longueur suivie du nom),
- le nom réseau de membre émetteur (N. EM),
- le numéro du paquet (NM) ou 255 pour les paquets non numérotés,
- 3 indicateurs (MDA) : Fin du message, Demande et Attribution de parole,
- le Code du paquet .

La partie spécifique de chaque code est décrite ci-dessous :

SD	SE	N. CV	N. EM	NM	MDA	Code	NR	CR	SS	NP	
FF	DONN	texte
FF	DONA	dest. texte
FF	0..	E-M	param
FF	0..	S-M	param
FF	0	..	0..	E-S	S1/NR/CR, S2/NR/CR,
FF	0	..	0..	S-S	
FF	0..	DLST	
FF	0	RLST	dest Nom1, Nom2,
FF	0	..	000	SREJ	ND				
FF	..	DIFFU	0	..	000	ACK	CV1/NR/CR, CV2/NR/CR,				

DIFFU : Nom réseau (DIFFUSION) se trouvant sur tous les communicateur, il permet d'adresser toutes les stations.

DONN : Code des paquets de données transmis en diffusion.

NR : Numéro du dernier paquet reçu.

CR : Crédit, nombre de paquets que la station est prête à recevoir dans cette conversation à partir de NR.

SS : Le numéro de la station auquel est attribuée la parole, significatif si A=1.

NP : Numéro de parole demandée si D=1, attribuée si A=1, de la dernière demande si D et A sont égaux à 0.

texte: Données transmises dans le paquet.

DONA : Code des paquets de données transmis en aparté.

Dest : Nom réseau du membre destinataire du paquet.

E-M : Code paquet indiquant l'arrivée (entrée) d'un membre.

S-M : Code paquet indiquant le départ (sortie) d'un membre.

Param: Texte fourni par l'abonné lors de son entrée ou sa sortie de la conversation et délivré aux membres de la conversation.

E-S : Code paquet indiquant l'entrée d'une station dans la conversation.

S-M : Code paquet indiquant la sortie d'une station de la conversation.

Si/NR/CR : Liste des stations qui participent déjà à la conversation avec le dernier paquet reçu NR et leur crédit CR dans cette conversation.

DLST : Code paquet de la demande de la liste des membres de la conversation.

RLST : Code paquet de la réponse à une demande de liste.

dest. : Nom du membre destinataire de la liste.

Nom1, Nom2, .. : Noms réseaux des membres gérés par la station émettrice de RLST.

SREJ : Code paquet de demande de réémission d'un paquet.
ND est le numéro du paquet à réémettre.

ACK : Code des paquets d'entretien

CVI/NR/CR : liste des conversations gérées par la station avec le dernier paquet reçu NR et le crédit CR de chacune d'elles

B. INTERFACE DE TRANSPORT.

Cette interface définit pour les processus abonnés ou membres la manière d'utiliser le service conversation.

1. Entrée d'un abonné dans une conversation.

L'entrée d'un abonné dans une conversation se passe en 2 étapes. Dans un premier temps, il demande à la station de transport d'ouvrir cette conversation en fournissant le nom de celle-ci et en précisant s'il est le premier à le demander.

OUVRIR (Abonné, Nomcv, [premier], proc.récep, Statut)

Nomcv : Ce paramètre indique le nom réseau de la conversation.

Abonné : Nom réseau de l'abonné.

premier: Paramètre précisant si l'abonné est le premier à demander l'ouverture cette conversation.

proc.récep : procédure déclarée par l'abonné, elle permet à la station de transport de le dérouter vers celle-ci pour lui signaler l'un des événements suivants : entrée d'un membre, sortie d'un membre, disparition d'un ensemble de membres (panne station) et arrivée d'un message ou d'une liste. Cette procédure doit être déclarée de la manière suivante :

```
PROCEDURE recept(type, émet, nbr, lg, ptexte, fin)
VAR
  [déclaration des variables]
DEBUT
  :
  :
  nbr:=nbr-1;
FIN
```

type : type de l'évènement

- . e-m : entrée d'un nouveau membre.
- . s-m : sortie d'un membre de la conversation.
- . s-s : sortie accidentelle d'une station.
- . donn: réception d'un paquet de données émis en diffusion.
- . dona: réception d'un paquet de données émis en aparté.
- . rlst: réception d'une liste des membres de la conversation.

émet : Nom réseau du membre émetteur.

nbr : indique le nombre de récepteurs. Il est décrémenté de 1 par chaque récepteur et permet d'indiquer à la station de transport si ce paquet a été retiré par tous ses destinataires.

lg : longueur du texte reçu.

ptexte : pointeur du texte constituant le paquet.

fin : indique la fin d'un message.

statut : Ce paramètre contient en retour le statut de l'opération

- . ouverture réussie,
- . ouverture manquée (nom incorrect, ressources insuffisantes, ...).

Dans le cas où l'opération d'ouverture s'est passée correctement, l'abonné est connu de la conversation mais n'est pas encore membre.

Il peut demander la liste des membres de la conversation en utilisant la primitive LISTE .

LISTE (Abonné, Nomcv, statut)

statut : Statut de l'opération

- . Demande acceptée
- . Nomcv ou Abonné incorrect.

Les éléments de cette liste lui seront délivrés au fur et à mesure qu'ils arrivent. La station positionnera l'indicateur fin lors de la remise de la dernière réponse.

Il peut aussi demander à devenir membre de la conversation à l'aide de la primitive ENTRER.

ENTRER (Abonné, Nomcv, Crédit, [lg, ptexte], statut)

Crédit : nombre de paquets que le nouveau membre est prêt à recevoir.

ptexte : Pointeur d'un texte non obligatoire délivré à tous les membres de la conversation.

lg : Longueur du texte.

Si l'opération d'entrée s'est bien passée, l'abonné est membre de la conversation .

2. Emission de messages.

Une fois devenu membre de la conversation, un abonné peut émettre à tout moment des messages en diffusion à l'aide de la primitive DIFFUSER ou en aparté à l'aide de la primitive EMETTRE dans cette conversation.

DIFFUSER (Abonné, Nomcv, lg, ptexte, statut)

lg : longueur du texte à émettre.

ptexte : Adresse de la chaîne de caractères constituant le message.

EMETTRE (abonné, Nomcv, dest., lg, ptexte, statut)

dest. : Nom réseau du membre destinataire.

3. Crédit.

Un membre de la conversation indique à l'aide de la primitive CREDIT le nombre de paquets qu'il est prêt à recevoir après le dernier reçu.

CREDIT(Abonné, Nomcv, val.crédit, statut)

val.crédit : nombre de paquets supplémentaires que le membre peut recevoir.

4. Sortie d'un membre de la conversation.

Un abonné qui désire quitter la conversation, le signale à sa station de transport à l'aide de la primitive SORTIR. La station avertit les autres membres du départ de celui-ci.

SORTIR (Abonné, Nomcv, [lg, ptexte], statut)

lg, ptexte : longueur et adresse d'un texte non obligatoire délivré aux membres restants.

5. Fermeture de la conversation

Chaque abonné doit, de la même manière qu'il a demandé l'ouverture d'une conversation, demander à la station de transport la fermeture de celle-ci à l'aide de la primitive FERMER.

FERMER (Abonné, Nomcv, statut)

Cette primitive entraîne la sortie de la station de transport de la conversation si l'abonné est le dernier.

C. LA STATION DE TRANSPORT

C.1 DESCRIPTION DES CONTEXTES

A l'ouverture d'une conversation par un abonné, la station de transport met à jour son propre contexte et crée, s'il, ne l'est pas déjà, un contexte conversation qui ne sera détruit qu'à la fermeture de la conversation.

Contexte station de transport

- . Numéro de site de la station (s)
- . liste des abonnés (pointeur sur le 1^{er} contexte)
- . liste des stations distantes (pointeur sur le 1^{er} contexte)
- . Liste des conversations (pointeur sur le 1^{er} ctxt)
- . nbre de conversations pour lesquelles la station a le droit de parole (nbpa)
- . Tampons utilisés par les différents processus
- . réveil n° 1 (sert à détecter les pannes stations)
- . val compteur (comptabilise les top du réveil 1)
- . réveil n° 2 (permet l'émission des paquets d'entretien)

Contexte abonné

- . pointeur sur le contexte suivant
- . Nom réseau de l'abonné

Contexte station distante

- . Pointeur sur le contexte suivant
- . Numéro station
- . Indicateur présence

Contexte conversation

- . Pointeur sur le contexte suivant
- . Nom réseau de la conversation (cv)
- . Etat
- . Numéro paquet attendu (nat)
- . Parole
 - Numéro de la dernière demande (ndd)
 - Numéro de la demande (npd)
 - Texte de la demande
- . Successeur
 - Numéro de station (suc)
 - Numéro de parole (nps)
 - Texte
- . Crédit local [MIN(crédits des membres locaux)]
- . Crédit en émission [MIN(crédit local, MIN(crédits des stations participantes)]
- . Indicateurs (vrai/faux)
 - Ouverture en cours
 - Sortie en cours
 - Finmsg
 - Demande de parole DP
 - Erreur
- . Membres locaux
 - Nombre
 - Liste (pointeur sur le premier ctxt membre)
- . Stations distantes
 - Nombre
 - Liste (pointeur sur le premier ctxt station)
- . Files des paquets en attente
 - d'émission
 - d'acquiescement
 - de contrôle d'erreur
 - de retrait par les membres
- . Réémission

- Numéro du réveil
- Tampon

Pour gérer les membres locaux d'une conversation, la station de transport crée pour chacun d'eux un contexte qui est rattaché au contexte de la conversation.

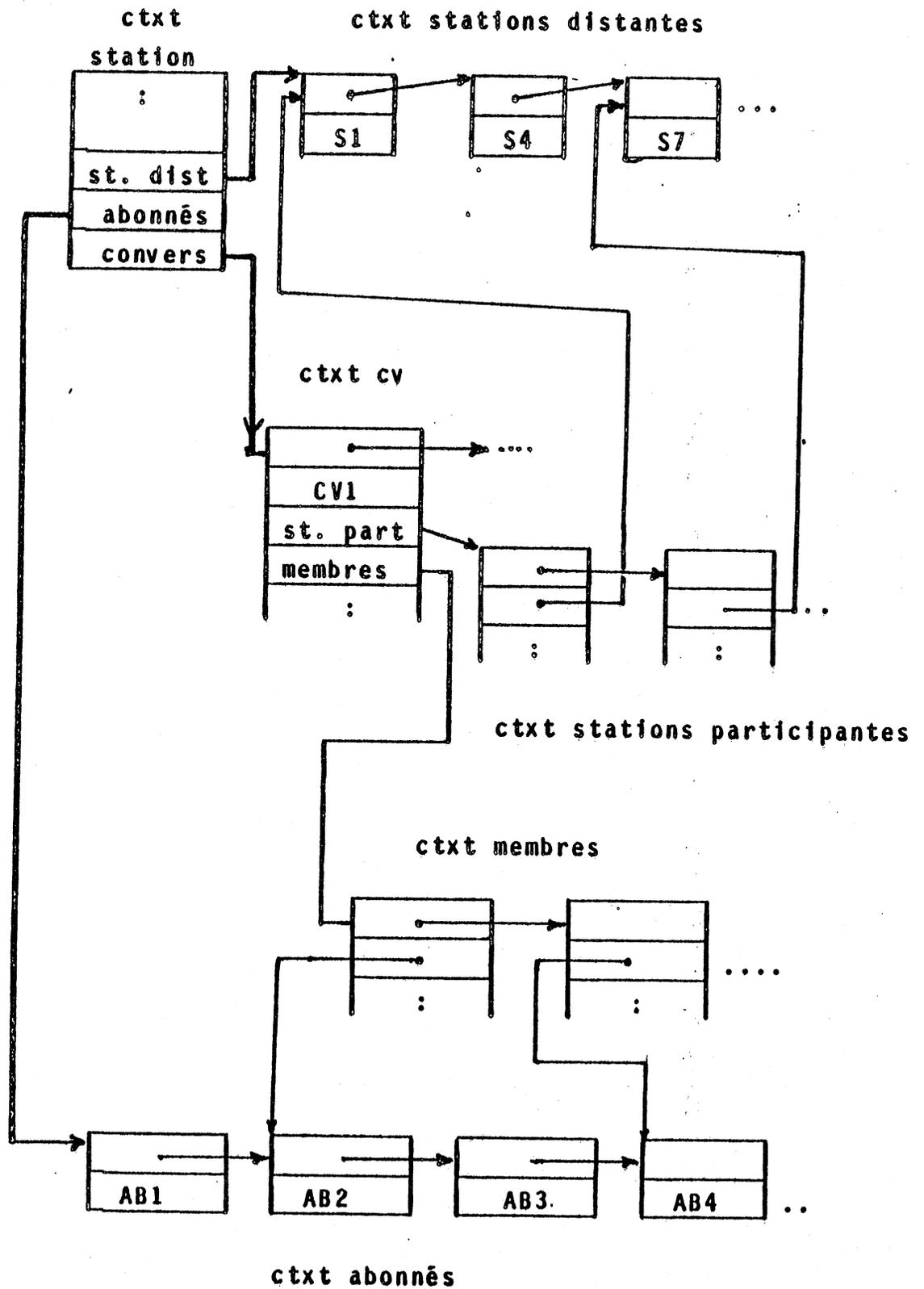
Contexte membre

- . pointeur sur le contexte suivant
- . ponteur sur le contexte abonné
- . Etat (en cours, membre)
- . Crédit en réception
- . Procédure réception
- . Liste en cours
- . Nbre de réponses reçues

Pour contrôler le bon fonctionnement de la conversation, la station de transport crée pour chaque station distante qui participe à cette conversation un contexte station distante qui est aussi rattaché au contexte conversation.

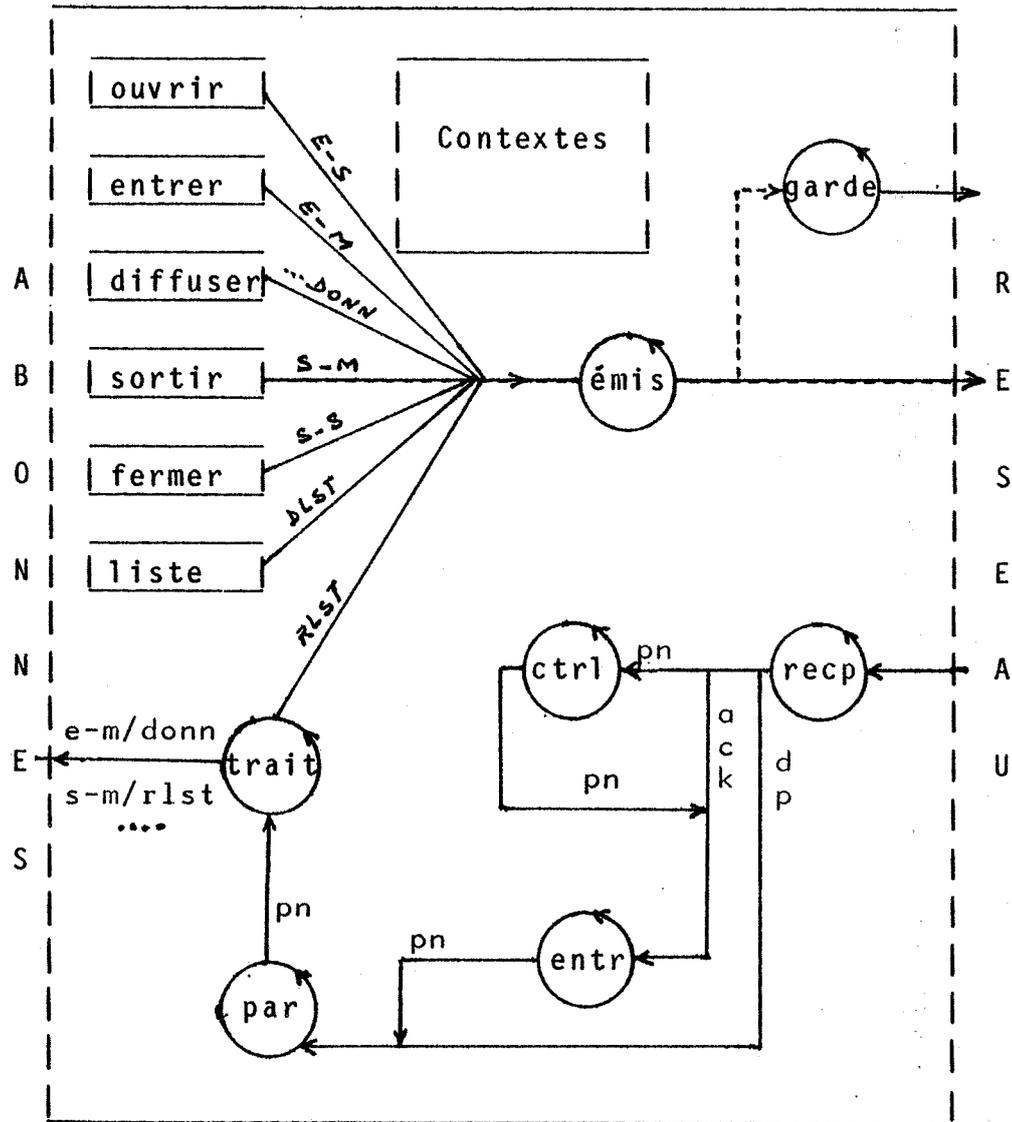
Contexte station participante

- . Pointeur sur le contexte suivant
- . Pointeur sur le contexte station distante
- . Crédit en réception
- . Numéro dernier paquet acquitté



Enchaînement des contextes.

C.2 SCHEMA DE FONCTIONNEMENT



dp : demande de parole

pn : paquet numéroté

: Procédures appelées par les abonnés

: processus de la station de transport

- émis** : Processus de gestion des émissions. Il consulte les files des paquets en attente d'émission des différentes conversations en cours et suivant l'état de la conversation, il émet un paquet numéroté ou une demande de parole.
- récp** : Processus récepteur, il reçoit les paquets en provenance du réseau et les dirige suivant leur nature vers les différents processus.
- ctrl** : processus de contrôle d'erreur, il traite tous les paquets numérotés (PN).
- par** : Processus de gestion de parole, il reçoit les paquets numérotés et les demandes de parole. Il garde le numéro de la dernière demande de parole et éventuellement le numéro et le texte du successeur.
- entr** : Processus entretien, il gère les crédits, les acquittements et la présence des stations.
- trait** : Processus de traitement des paquets suivant leur code.
- garde** : Processus de garde, il permet la réémission sur temporisateur de certains paquets. A chaque conversation sont associés un réveil et un compteur d'échéances.

A chaque processus est associé un tampon dans lequel est déposé le paquet à analyser. Le nom de la conversation concernée se trouve dans le paquet lui même.

Etats d'une conversation au niveau d'une station :

N : état initial et final.

PAR : la station détient le droit de parole pour cette conversation.

ATDP : la station a le droit de parole, mais pas de paquets numérotés à émettre. Elle attend soit des paquets à émettre soit une demande de parole.

ECO : la station n'a pas le droit de parole, elle ne peut émettre que des paquets non numérotés dans cette conversation.

ATCF : la station vient d'attribuer la parole à son successeur et attend la confirmation (réception d'un paquet quelconque numéroté).

ATCR : en attente de crédit, les émissions de paquets numérotés sont bloquées.

Liste des évènements

Nous ne citons ici que les évènements susceptibles de modifier l'état de la conversation. Les évènements pris en compte sont classés en 3 catégories.

- Liste des évènements de l'interface de transport

OV1 : demande d'ouverture de la conversation avec l'option premier.

OV0 : demande d'ouverture de la conversation sans l'option premier.

EPQ : paquet numéroté à émettre, évènement généré par l'une des primitives (ENTREE, SORTIE, DIFFUSER, EMETTRE, FERMER).

CRA : augmentation du crédit d'un membre.

- Liste des évènements liés à la communication entre stations de transport.

Entre les différentes stations circulent des paquets. Lorsqu'un paquet est reçu, c'est à dire pris en compte, il provoque un évènement.

DP : arrivée d'un paquet de demande de parole.

AP : arrivée d'un paquet attribution de parole à la station.

PN : réception d'un paquet numéroté (attribution de parole incluse).

AK : réception d'un paquet d'entretien

- Liste des évènements générés par la station de transport.

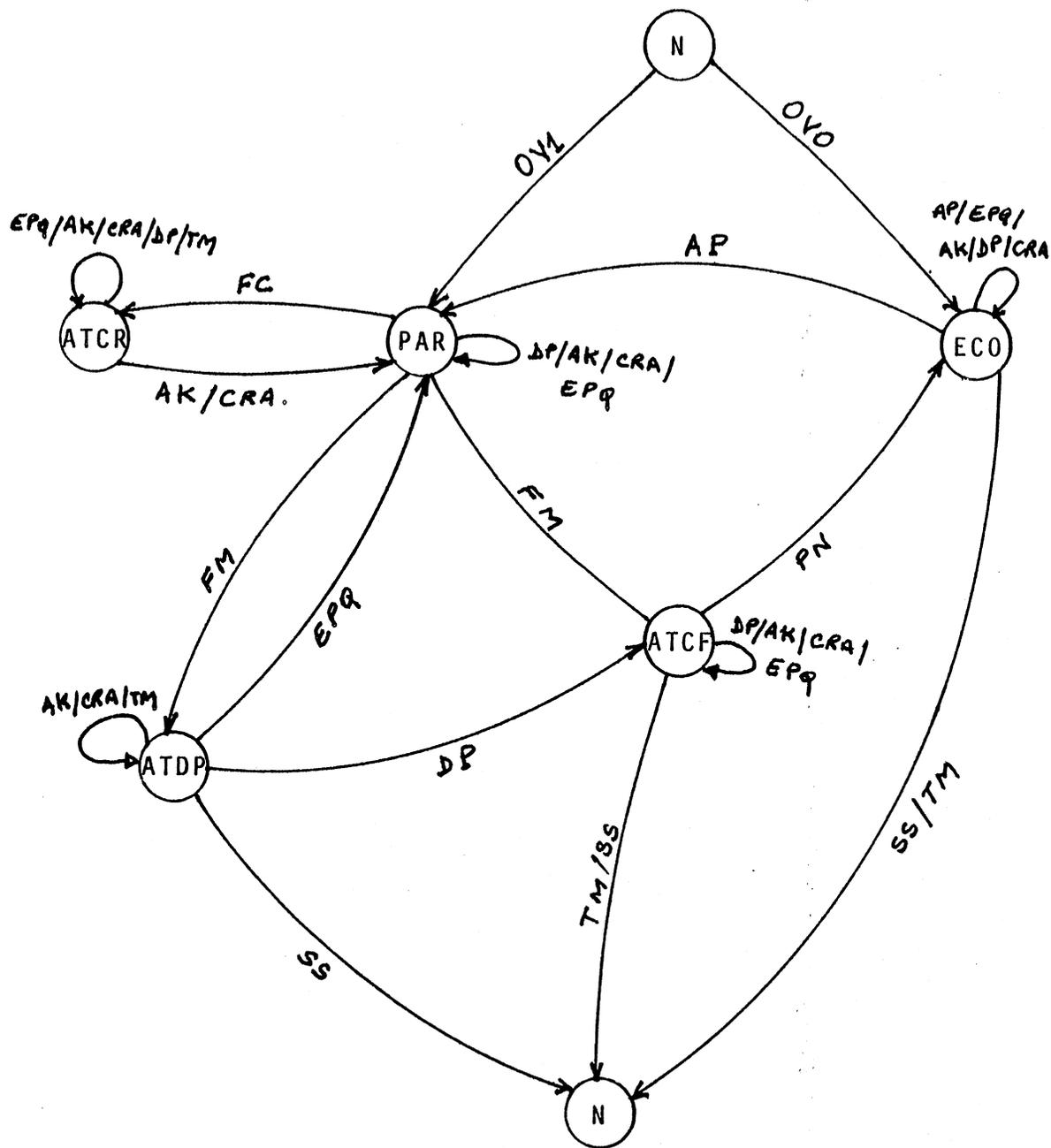
FM : le paquet émis constitue la fin d'un message.

FC : fin crédit de la conversation.

SS : la station quitte la conversation.

TM : fin d'un délai

automate d'états finis pour une conversation dans une station.



C.3 ALGORITHMES DES PROCEDURES

```
PROCEDURE OUVRIR(abonné, nomcv, [premier], proc.recep, statut)
DEBUT
  SI ouverture ou sortie en cours ALORS
    suspendre le proc. abonné
  FSI
  SI cv inexistante ALORS
    allouer un bloc mémoire pour le contexte cv
    remplir le contexte (nat = 0, ndd = 0, suc = 0)
  SI premier ALORS
    état = parole
    nbpa = nbpa + 1
    préparer un paquet e-s
    ranger le paquet dans la file des paquets en attente d'émission
  SINON
    ouverture en cours; état = écoute
    préparer un paquet e-s
    le ranger dans la file des paquets en attente d'émission
    suspendre le processus abonné en attente de l'attribution du
    droit de parole ou de fin délai
  SI fin délai ALORS
    libérer la mémoire
    retour avec statut "ouverture manquée"
  FSI
  FSI
  allouer un bloc mémoire pour le contexte membre
  remplir ce contexte (nom membre, état = en cours, proc. récept)
  retour avec statut "ouverture réussie"
FIN
```

PROCEDURE ENTRER(abonné, nomcv, crédit, [lg, ptexte], statut)
DEBUT

SI abonné ou cv inconnu ALORS
retour avec statut "entrée refusée"

FSI

préparer un paquet e-m

le ranger dans la file des paquets en attente d'émission
mettre à jour le ctxt membre (crédit)

suspendre le processus abonné

retour avec statut "entrée acceptée"

FIN

PROCEDURE LISTE(Abonné, nomcv, retour)

DEBUT

SI abonne inconnu de la cv ALORS

retour avec statut "refusé"

SINON

préparer un paquet DLST

le ranger dans la file des paquet en attente
d'émission

FSI

FIN

PROCEDURE CREDIT(membre, nomcv, val.credit, statut)

DEBUT

SI membre ou cv inconnu ALORS

retour avec statut "refusé"

FSI

rajouter val.crédit au crédit restant (ctxt membre)

retour

FIN

PROCEDURE DIFFUSER(membre, nomcv, lg, ptexte, statut)

DEBUT

SI cv ou membre inexistant ALORS

retour avec statut "refusé"

FSI

fragmenter le texte en paquets de code donn

ranger ces paquets dans la file émission

retour

FIN

PROCEDURE EMETTRE(membre, nomcv, destinataire, lg, ptexte, statut)

DEBUT

identique à DIFFUSER sauf que le code paquet est dona et qu'il faut
spécifier dans le paquet le nom du membre destinataire.

FIN

PROCEDURE SORTIR(membre, nomcv, statut)

DEBUT

SI membre ou cv inexistant ALORS

retour avec statut " cv ou membre inexistant"

FSI

préparer un paquet s-m

le ranger dans la file émission

modifier le ctxt membre (état = en cours)

retour

FIN

```
PROCEDURE FERMER(abonné, nomcv, statut)
DEBUT
  SI ctxt abonné ou cv inexistant ALORS
    retour statut "inexistant"
  FSI
  SI état.abonné = membre ALORS
    préparer un paquet s-m
    le ranger dans la file émission
  FSI
  détruire le ctxt abonné
  SI dernier abonné ALORS
    préparer un paquet s-s
    le ranger dans la file émission
  FSI
  retour
FIN
```

C.4 ALGORITHMES DES PROCESSUS

PROCESSUS EMISSION

CYCLE

POUR chaque conversation en cours

FAIRE

SI état = atdp ALORS

SI (suc > 0) ou (file non vide) ALORS

désarmer le réveil

SI suc = 0 ALORS finmsg = faux FSI

état = parole

FSI

FSI

SI état = parole ALORS

SI crédit en émission = 0 ALORS

recalculer le crédit local

recalculer le crédit distant

crédit en émission = MIN(cr local, cr distant)

FSI

SI crédit en émission = 0 ALORS

déposer le paquet précédent dans le tampon de réémission de
de la conversation

nbre d'échéances = 0

armer le réveil de la conversation

état = atcr

SINON

crédit en émission = crédit en émission - 1

SI finmsg = faux ALORS

extraire le premier paquet de la file

le compléter (nm = nat, np = ndd, nr = nat, ss = s,
cr = crédit local)

nat = nat+1

SI (code = e-m) et (crédit.ab < crédit en émission) ALORS

crédit en émission = crédit.abonné

FSI

déposer une copie du paquet dans le tampon du processus
traitement

déposer une copie dans la file des paquets en attente
d'acquiescement

émettre le paquet vers le réseau

SI pqt = finmsg ALORS finmsg = vrai FSI

SINON

SI suc = 0 ALORS

recalculer le crédit local

préparer un paquet attribution de parole

(nm = nat, np = npa, nr = nat, cr = crédit local,
ss = suc)

nat = nat+1

déposer une copie dans le tampon du proc. traitement

déposer une copie dans le tampon de réémission de la
conversation

armer le réveil de la conversation

émettre le paquet vers le réseau

nbre d'échéances = 0

suc = 0

```
nbpa = nbpa - 1
SI nbpa = 0 ALORS
    armer réveil station n° 2
FSI
état = atcf
SINON
SI file émission vide ALORS
    déposer le paquet précédent dans le tampon de
    réémission conversation
    armer le réveil de la conversation
    nbre d'échéances = 0
    état = atdp
FSI
FSI
FSI
FSI
SI (état = écoute) et (file non vide) et (demande de parole
= faux) et (erreur = faux) ALORS
    demande de parole = vrai
    ndd = ndd+1
    npd = ndd
    préparer un paquet de demande de parole (np = npd)
    déposer une copie dans le tampon réémission de la conversation
    SI paquet = finmsg ALORS finmsg = vrai FSI
    émettre le paquet vers le réseau
    état = atpa
FSI
FIN
FCYCLE
```

PROCESSUS RECEPTEUR
CYCLE

```
Attendre le prochain paquet en provenance du réseau
SI (se = s) ou (station non participante et code ≠e-s) ALORS
    ignorer le paquet
FSI
SI ouverture en cours ALORS
    SI paquet numéroté ALORS
        déposer le paquet dans le tampon du processus gestion parole
    FSI
    SINON
    SI code = srej et pqt émis par la station ALORS
        réémettre le paquet
    SI code = ack ALORS déposer le pqt dans le tampon processus
        entretien
    SI d = 1 ALORS déposer le pqt dans le tampon du proc. gestion
        de parole
    SI pqt numéroté ALORS déposer le pqt dans le tampon du
        processus contrôle erreur
FSI
FCYCLE
```

PROCESSUS CONTROLE ERREUR

CYCLE

attendre qu'il y ait un paquet dans le tampon

SI $nm < nat$ ALORS

ignorer le paquet

SINON

SI état = atcf ALORS

désarmer le réveil de la conversation

état = écoute

FSI

SI $nm > nat$ ALORS

ranger le paquet dans la file des paquets en attente de contrôle

SI erreur = faux ALORS

erreur = vrai

SI demande parole = vrai ALORS

désarmer le réveil de la conversation

FSI

préparer un paquet srej

déposer une copie dans le tampon de réémission de la conversation

émittre le paquet vers le réseau

armer le réveil de la conversation

nbre d'échéances = 0

FSI

SINON

SI erreur = vrai ALORS

erreur = faux

désarmer le réveil de la conversation

SI demande de parole = vrai ALORS

reconstituer le paquet demande de parole

déposer une copie dans le tampon de réémission de la conversation

émittre le paquet vers le réseau

armer le réveil de la conversation

nbre d'échéances = 0

FSI

FSI

nat = nat+1

déposer le paquet dans le tampon du processus entretien

SI file des paquets en attente d'acquittement non vide ALORS

déposer le premier dans le tampon du processus contrôle d'erreur

FSI

FSI

FSI

FCYCLE

PROCESSUS GESTION DE PAROLE

CYCLE

attendre qu'il y ait un paquet dans le tampon
demande de parole

SI (état = parole ou atdp ou atcr) ou (écoute et demande de
parole = vrai) ALORS

SI (np > npd et suc = 0) ou (estamp.s < estamp.se < estamp.suc)
ALORS

suc = se, npa = np
garder texte successeur

FSI

FSI

attribution de parole

SI demande parole = vrai ALORS

SI SS = S ALORS

nbpa = nbpa + 1

SI nbpa = 1 ALORS

désarmer le réveil n° 2

FSI

demande de parole = faux

SI ouverture en cours = vrai ALORS

ouverture en cours = faux

FSI

désarmer le réveil de la conversation

SINON

SI np > npd ALORS

SI np > ndd alors ndd = np FSI

ndd = ndd+1

npd = ndd

SI erreur = faux ALORS

désarmer le réveil de la conversation

préparer une autre demande de parole

déposer une copie dans le tampon de réémission

émettre le paquet vers le réseau

armer le réveil de la conversation

nbre d'échéances = 0

FSI

FSI

FSI

FSI

paquet numéroté (y compris attribution de parole)

SI np > ndd ALORS ndd = np FSI

SI ouverture en cours = faux ALORS

déposer le paquet dans le tampon du processus
traitement

FSI

FCYCLE

PROCESSUS ENTRETIEN
CYCLE

attendre qu'il y ait un paquet dans le tampon ou un réveil

Paquet

présence.se = val. compteur

SI code = ack ALORS

POUR chaque conversation

FAIRE

mettre à jour le contexte station se (num. dernier paquet,
crédit)

SI état = atcr ALORS

recalculer le crédit en émission

SI crédit > 0 ALORS état = parole FSI

FSI

FIN

SINON

mettre à jour le contexte station(se) de la conversation

crédit local = crédit local - 1

SI (crédit local = 0) et (ss ≠ s) et MIN(crédits des membres
locaux > 0) ALORS

crédit local = MIN(crédits des membres locaux)

désarmer le réveil station n° 2

recalculer tous les crédits locaux des autres conversations

constituer un paquet ACK

émettre le paquet vers le réseau

SI nbpa = 0 ALORS

réarmer le réveil station n° 2

FSI

FSI

FSI

Réveil n°1

val. compteur = val. compteur + 1

POUR chaque conversation

FAIRE

POUR chaque station

FAIRE

SI val. compteur - val.présence seuil ALORS

détruire contexte station

nbre stations = nbre stations - 1

avertir les membres locaux de la conversation

FSI

FIN

SI état = atcr ALORS

recalculer le crédit en émission

SI crédit > 0 ALORS état = parole FSI

FSI

FIN

Réveil n°2

calculer le crédit local de toutes les conversations

constituer et émettre vers le réseau un paquet ACK

FCYCLE

PROCESSUS TRAITEMENT

CYCLE

Attendre qu'il y ait un paquet dans le tampon

SI code = e-s ALORS

SI ss = s ALORS

mettre à jour le contexte conversation
créer et remplir les différents contextes stations
participantes

SINON

créer le contexte station ss
nbre de stations = nbre de stations + 1

FSI

FSI

SI code = s-s ALORS

SI ss = s ALORS

sortie en cours = vrai
SI nbre de stations = 0 ALORS
détruire le contexte conversation

FSI

SINON

nbre stations = nbre de stations - 1
détruire contexte station

FSI

FSI

SI code = d1st ALORS

préparer un paquet r1st

le ranger dans la file des paquets en attente d'émission

FSI

SI (code = e-m) et (ss = s) ALORS

mettre à jour le contexte membre
(état.ab = membre)

réveiller le processus abonné suspendu dans la procédure ENTRER

FSI

SI code = donn / dona / e-m / s-m ALORS

ranger le texte avec son code, son émetteur et le nbre de récep-
teurs dans la file des paquets reçus

activer les procédures de réception des abonnés concernés

décrémenter de 1 les crédits des abonnés concernés

supprimer de la file les paquets retirés par tous les
destinataires

FSI

SI (attr. parole) et (ss = s) ALORS

modifier le paquet (se = s, a = 0)

le ranger dans la file des paquets en attente d'acquittement

calculer le crédit en émission

état=parole

FSI

FCYCLE

PROCESSUS DE GARDE

CYCLE

atteindre la prochaine échéance

Réveil N° 1

réveiller le processus traitement
réarmer le réveil

réveil N° 2

réveiller le processus traitement

Réveil d'une conversation

SI état = atcr ou atdp ALORS

réémettre le paquet

SINON

nbre d'échéances + 1

SI nbre d'échéances < seuil ALORS

réémettre le paquet

réarmer le réveil

SINON

SI état = écoute ALORS

SI ouverture en cours = vrai ALORS

réveiller le processus abonné suspendu
dans la procédure OUVRIR (fin délai)

SINON

SI erreur = vrai ALORS

avertir les membres locaux (erreur irrécupérable)
détruire le contexte conversation

SINON

avertir les membres locaux (perte du droit de parole)
détruire le contexte conversation

FSI

FSI

SINON

avertir les membres locaux "panne station"

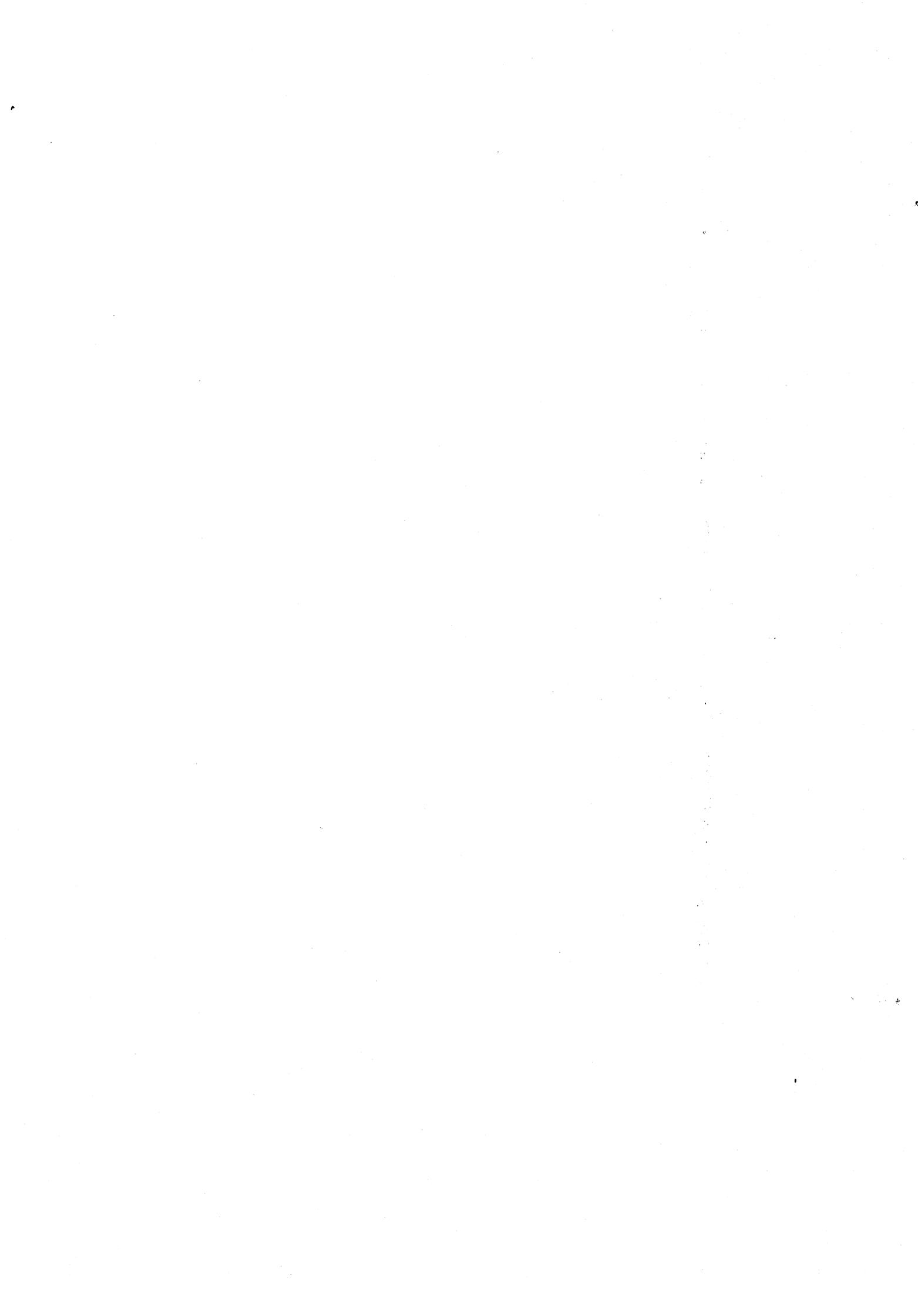
état=parole

FSI

FSI

FSI

FCYCLE



BIBLIOGRAGHIE

- SUN.75 C. A. Sunshine
"Interprocess communication protocol for computer network"
Technical report n. 105, Standford university decembre 1975
- MET.76 R. Metcalfe, D. Boggs
"ETHERNET : Distributed packet switching for local computer network"
Comm. of ACM, juillet 1976
- CLA.78 D. D. Clark, K. T. Pogran
"An introduction to local area network"
Preceedings of IEEE vol. 66 Novembre 1978
- JAC.78 I. M. Jacobs, R. Binder, E. V. Hoversten
"General purpose packet satellite networks"
Proceding of IEEE, vol 66 n 11, Novembre 78
- LAM.78 L. Lamport
"Time clocks and the ordering of events in a distributed system"
Comm. of ACM, juillet 1978
- DAV.79 Davies, Barber, Prince and Solomonies
"Computer networks and their protocols"
(Chap. 5), 1979
- QUI.79 V. Quint, N. Naffah
"Projet pilote KAYAK"
Protocole de transport pour réseaux locaux"
INRIA, 1979
- NAF.79 N. Naffah
"Projet pilote KAYAK"
Description du réseau expérimental DANUBE"
INRIA, Novembre 1979

- ROU.79 X. Rousset de Pina, I. Vattton
"Interface de la station de transport sur MICRO-1",
Note interne IMAG, Université de GRENOBLE 1979
- VER.79 M. Veran
"QNAP : Queuning Network Analysis Package"
CII-HB, Grenoble, 1979
- SCO.80 SCOT Group
"A concurrency control algorithm in a distributed
environment"
CII-HB, RR n° 12, octobre 1980
- DIE.80 R. Diehl, F. Fernandez, L. Ferrat, N. Gia Taon
" MICROBE : Système de bases de données relationnelles ré-
parties sur un réseau local de micro-ordinateurs"
Congrès AFCET informatique, Novembre 1980
- QUI.80 V. Quint, H. Richey, G. Sergeant
"PAPYRUS : Proposition pour un système de messagerie et sa
mise en oeuvre sur un réseau local"
Congrès AFCET informatique, Novembre 1980
- LAB.80 J. Labetoulle
"Etude analytique du réseau DANUBE"
INRIA, Re1 2.518 Mars 1980
- SER.80 G. Sergeant, L. Treille
"SER : a system for distributed execution based on
decentralized control techniques"
ICCC 80 Atlanta, octobre 1980
- WAL.80 D. W. Wall
"Selective broadcast in packet switched"
Juin 1980

- COR.81 Cornafion**
"Systèmes informatiques répartis : Concepts et techniques"
1981
- HER.81 A. Herbert**
"The user interface to Cambridge model distributed system"
Proc. 2nd int. conf. on distributed system PARIS, Avril
1981
- NAD.81 J. L. Grangé, C. Huitema, H. Zimmermann**
"Utilisation informatique des satellites de
télécommunication"
Projet pilote NADIR, INRIA, decembre 1981
- DEC.82 P. Decitre, J. Estublier, A. Khider, X. Rousset de Pina, I.
Vatton**
"An efficient error detection mechanism for a multicast
service on DANUBE network"
Local computer networks IFIP TC6, Florence 1982
- DEC.83 P. Decitre, A. Khider, G. Sergeant, I. Vatton, G. Vandome**
"Protocole de diffusion fiable pour réseaux locaux"
RR n° 348 IMAG, janvier 1983
- DEC.83 P. Decitre, A. Khider, G. Vandome**
"Protocole de diffusion fiable pour le réseau local DANUBE"
RR n° 368 IMAG, mars 1983
- LOT.82 C. Lotito**
"Interconnexion de système de messageries"
Memoire d'ingenieur, CNAM, decembre 1982
- SAN.82 V. Sanchez**
"Specification et réalisation d'un protocole de station de
transport offrant un service à diffusion fiable sur le
réseau local DANUBE"
Rapport de DEA ENSIMAG, septembre 1982

AUTORISATION DE SOUTENANCE

VU les dispositions de l'article 3 de l'arrêté du 16 avril 1974,

VU les rapports de présentation de Messieurs

. KRAKOWIAK, Professeur

. VANDOME, Ingénieur

Monsieur KHIDER Amar

est autorisé à présenter une thèse en soutenance pour l'obtention du diplôme de
DOCTEUR-INGENIEUR, spécialité "Informatique".

Fait à Grenoble, le 03 mai 1983

Le Président de l'I.N.P.-G.

D. BLOCH
Président
de l'Institut National Polytechnique
de Grenoble

RO. le Vice-Président.



