RESEAUX A HAUT DEBIT : INTERFACES

Auteur : Thierry USO

Version : 1.3

Date : 1 Aout 2001

Table des matières

CHAPITRE 1 INTRODUCTION 1–1

CHAPITRE 2 CONCEPTS 2–1

CHAPITRE 3 ADAPTATEUR 3–1

3.1 INTERCONNEXION DE L'ADAPTATEUR 3–1

3.2 REPARTITION DES OPERATIONS 3–3

CHAPITRE 4 LOGICIEL 4–1

4.1 DYSFONCTIONNEMENTS 4–1

4.2 OPTIMISATIONS 4–4

ANNEXE A BIBLIOGRAPHIE A–1

GLOSSAIRE Gloss.–1

FIGURES

2–1 Blocs fonctionnels d'une interface réseau 2–3

3–1 Interconnexion des adaptateurs 3–2

4–1 Livelocks en réception 4–3

4–2 Prédiction d'entête par le récepteur TCP 4–6

TABLEAUX

3–1 Quelques adaptateurs FDDI 3–3

3–2 Impact du DMA sur la performance d'un adaptateur FDDI 3–4

4–1 Mesure des livelocks 4–2

4–2 Influence de la taille des messages sur le récepteur UDP 4–5

4–3 Influence de la taille de la fenêtre sur le débit de TCP 4–5

Chapitre 1

INTRODUCTION

Ce document explique comment les constructeurs informatiques conçoivent des interfaces de réseau à haut débit performantes.

Le chapitre 2 présente les concepts nécessaires à la compréhension du fonctionnement d'une interface. Le chapitre 3 aborde les aspects matériels (architecture des adaptateurs) et le chapitre 4 les aspects logiciels (techniques d'optimisation).

Un glossaire à la fin du document explicite les nombreux acronymes utilisés.

Chapitre 2

CONCEPTS

Le traitement d'une PDU en réception ou en émission s'effectue dans 3 blocs fonctionnels distincts :

- l'application
  L'application manipule ses données dans l'espace mémoire utilisateur et communique avec le réseau au travers d'un ensemble de primitives. Ces primitives implémentées dans le système d'exploitation permettent l'allocation d'un port de communication, la lecture et l'écriture de données à travers ce port (sockets BSD, XTI OSF...).
- le système d'exploitation
  Le système d'exploitation manipule ses données dans l'espace mémoire système et communique avec l'adaptateur réseau par un driver.
- l'adaptateur réseau
  L'adaptateur réseau manipule ses données dans son espace d'E/S et se charge de recevoir et d'émettre des PDUs sur le réseau.

La réception d'une PDU dans l'espace d'E/S de l'adaptateur entraine plusieurs opérations successives :

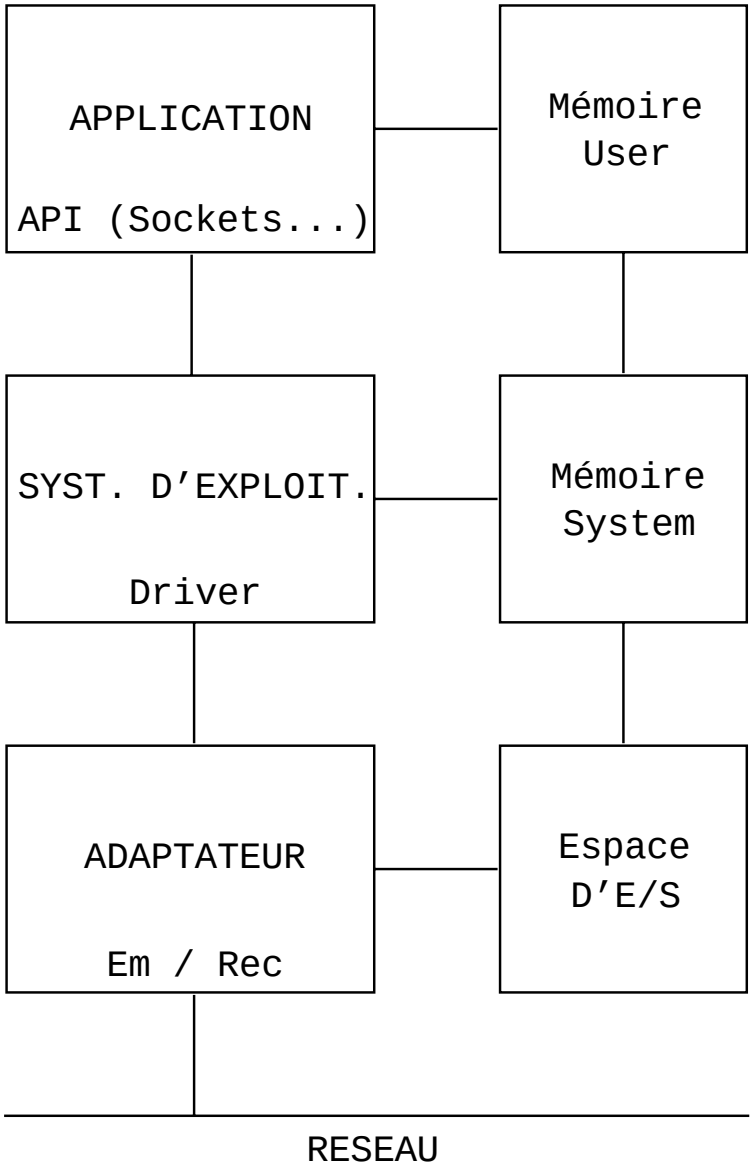
1. le déclanchement d'une interruption pour avertir le système d'exploitation
2. l'allocation de buffers dans l'espace mémoire du système
3. la copie de la PDU dans ces buffers
4. le traitement de l'entête
5. le set/reset de timers
6. le calcul du checksum
7. le changement d'état de l'automate du protocole
8. la copie des buffers vers l'espace mémoire utilisateur
9. le changement de contexte système d'exploitation-process applicatif

Certaines des opérations ci-dessus sont indépendantes du ou des protocoles mis en oeuvre par le réseau (interruption, copie, changement de contexte...). La même remarque peut être faite pour l'émission d'une PDU.

Clark D. et al. (1989) ont montré que dans la plupart des implémentations antérieures aux années 90 le temps passé dans les opérations indépendantes du ou des protocoles représente 80% du temps de traitement de la PDU. Parmi ces opérations, celles mettant en jeu la mémoire (allocation de buffers, copies, alignement correct des données...) se révèlent les plus coûteuses (40-50%).

Une interface de réseau à haut débit doit être capable de traiter plusieurs centaines de milliers de PDUs par seconde durant un burst. Une implémentation 80-20 ne peut pas répondre à ces besoins et conduit à des performances catastrophiques (network I/O bottleneck). Par conséquent, les constructeurs informatiques ont dû remettre en cause la conception des adaptateurs mais aussi du logiciel.

Figure 2–1: Blocs fonctionnels d’une interface réseau



Chapitre 3

ADAPTATEUR

3.1 INTERCONNEXION DE L'ADAPTATEUR

Le 1er choix auquel est confronté le concepteur d'adaptateur est la façon dont l'adaptateur s'interconnecte avec la mémoire et le processeur du système. La figure 3–1 illustre les 3 solutions possibles d'interconnexion de l'adaptateur :

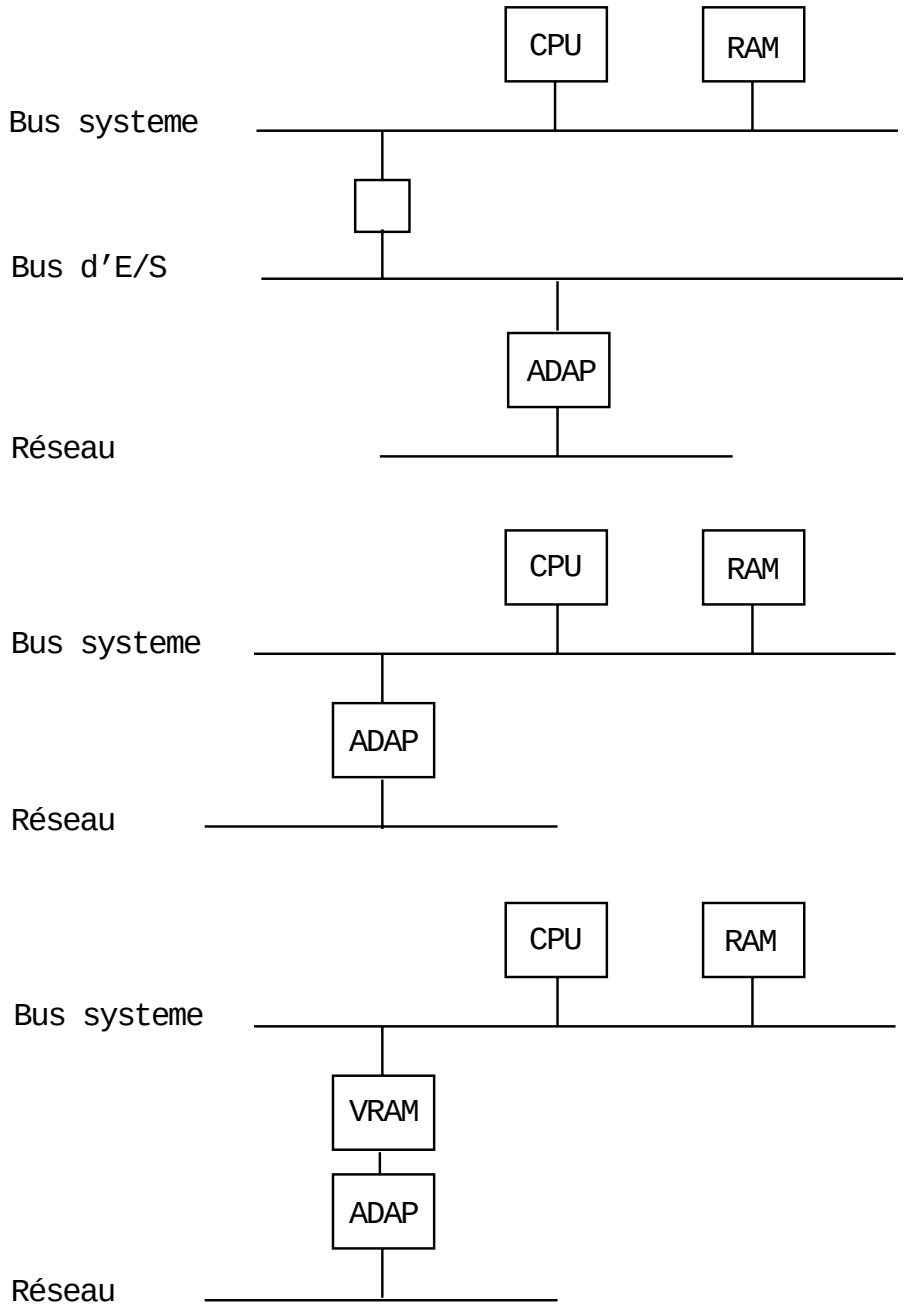
- le bus d'E/S
- le bus système
- la mémoire à double accès de type VRAM

La solution via le bus d'E/S a été longtemps privilégiée par les concepteurs. En effet, elle minimise les coûts de développement puisque certains bus d'E/S couvrent de larges gammes de systèmes. Cependant, la difficulté réside dans l'optimisation de la copie des données entre l'espace d'E/S de l'adaptateur et la mémoire système et cela d'autant plus qu'il est nécessaire de traverser un contrôleur d'E/S.

La solution via le bus système est moins viable économiquement que la solution précédente car chaque bus système est spécifique d'un nombre limité de systèmes. Elle présente cependant quelques avantages techniques par rapport à la solution précédente. L'overhead dû à la traversée d'un contrôleur d'E/S disparait. L'espace d'E/S de l'adaptateur peut être défini comme de la mémoire système éliminant la nécessité d'effectuer la copie des données entre l'espace d'E/S de l'adaptateur et la mémoire système.

La solution via de la mémoire VRAM est la plus coûteuse et n'est utilisée que dans quelques cas. L'espace d'E/S de l'adaptateur est alors défini comme de la mémoire système éliminant la nécessité d'effectuer la copie des données entre l'espace d'E/S de l'adaptateur et la mémoire système.

Figure 3–1: Interconnexion des adaptateurs



3.2 REPARTITION DES OPERATIONS

Le 2ème choix auquel est confronté le concepteur d'adaptateur est la détermination des opérations à faire exécuter respectivement par l'adaptateur et le processeur du système. Les adaptateurs sont classés en 3 catégories selon la répartition de ces opérations :

- On-board processing
- On-board processing partiel
- Programmed I/O

Le On-board processing consiste à doter l'adaptateur d'un processeur et de faire exécuter par celui-ci l'ensemble des fonctions liées au réseau. Le processeur du système est ainsi déchargé des opérations telles que le calcul du checksum ou le traitement de l'entête. De plus, la plupart des opérations indépendantes du ou des protocoles n'ont plus de raison d'être (interruption...). Cette approche n'a pas donné jusqu'à présent de produits économiquement viables. En effet, l'adaptateur n'est performant que si son processeur est aussi rapide que celui du système.

Le On-board processing partiel consiste à doter l'adaptateur d'un peu d'intelligence sous la forme d'un DMA et/ou d'un petit processeur dédié à quelques fonctions. Cette approche est en général couplée à une interconnexion via le bus d'E/S. Le DMA a alors pour rôle d'optimiser la copie des données entre l'espace d'E/S de l'adaptateur et la mémoire système en déchargeant le processeur du système de cette tâche. Le petit processeur dédié effectue par exemple le setup du DMA et traite les fonctions SMT dans le cas d'un adaptateur FDDI.

Le Programmed I/O consiste à laisser le processeur du système exécuter l'ensemble des fonctions liées au réseau. Pour que l'adaptateur soit néanmoins performant, son espace d'E/S doit être défini comme de la mémoire système. Par conséquent, cette approche est en général couplée à une interconnexion via le bus système ou la mémoire VRAM.

Le tableau 3–1 décrit les choix des concepteurs de quelques adaptateurs FDDI performants (environ 90 Mbit/s en débit maximum au niveau UDP).

Tableau 3–1: Quelques adaptateurs FDDI

Système	Interconnexion	Processing
HP9000-7xx	VRAM	Programmed I/O
DEC3000-AXP	TurboChannel	DMA + 68000
DEC2100-AXP	PCI	DMA + 68000

Un des concepteurs de l'adaptateur FDDI du DEC3000-AXP (RAMAKRISHNAN K.K., 1993) montre que les performances obtenues dépendent à la fois du type de bus d'E/S et du type de DMA.

Le bus d'E/S doit être très rapide d'où le choix du TurboChannel qui a une capacité de 800 Mbit/s (hors arbitrage du bus).

Le DMA doit être "intelligent" (tableau 3–2). En émission, les données à émettre se présentent sous forme de buffers discontinus dans la mémoire système. Ces données doivent être assemblées pour former la PDU. L'assemblage peut être effectué soit par des mouvements dans la mémoire système préalablement au transfert DMA (solution DMA), soit par le DMA lui-même (solution DMA+). En réception, les transferts DMA ne doivent pas pénaliser trop lourdement le processeur du système; le nombre de cycles du bus bloqués doit rester faible. De plus, le système d'exploitation doit continuer à assurer la cohérence des caches malgré les transferts DMA.

Tableau 3–2: Impact du DMA sur la performance d'un adaptateur FDDI

	Rcv DL	**Rcv UDP**	**Tx DL**	**Tx UDP**
Prg I/O (Mbit/s)	41	27	58	34
DMA (Mbit/s)	110	49	61	35
DMA+ (Mbit/s)	110	49	122	48

Chapitre 4

LOGICIEL

4.1 DYSFONCTIONNEMENTS

Une interface ne peut être performante que si elle est dotée à la fois d'un adaptateur bien conçu et d'un logiciel optimisé. Lorsque l'adaptateur est trop rapide par rapport au traitement logiciel des PDUs, RAMAKRISHNAN (1993) met en évidence 2 types de dysfonctionnements :

- famine en émission
- livelocks en réception

La famine en émission s'explique par le fait que la réception est prioritaire par rapport à l'émission. La réception d'un burst de PDUs peut ainsi bloquer temporairement l'émission d'une PDU.

La figure 4–1 illustre les livelocks en réception. Le traitement d'une PDU en réception implique plusieurs interruptions du processeur du système. La 1ère interruption est matérielle (adaptateur) et elle ne peut pas être préemptée. Les autres interruptions sont logicielles et ont des niveaux de priorité inférieurs. Si la durée de traitement d'une PDU en réception est trop longue par rapport au délai inter-arrivée des PDUs appartenant à un même burst, les interruptions matérielles bloquent temporairement le traitement des PDUs. Une PDU dans un burst peut ainsi voir son temps de traitement multiplié par 2 ou 3 par rapport à une PDU isolée.

LOEB et al. (2001) ont quantifié les livelocks dans l'environnement de test GigaEthernet suivant :

- Server IBM Netfinity
  Processeur Intel Pentium Xeon 450 à 700 MHz
  Microsoft NT 4.0 et TCP/IP

- Adaptateur IBM Netfinity GigaEthernet SX
  Interconnexion via bus PCI (64 bits, 33 MHz) à 2 Gbit/s
  Branchement sur commutateur Intel 510T
- Réception de trames de 1500 octets
  Trames émises par un ou plusieurs postes clients

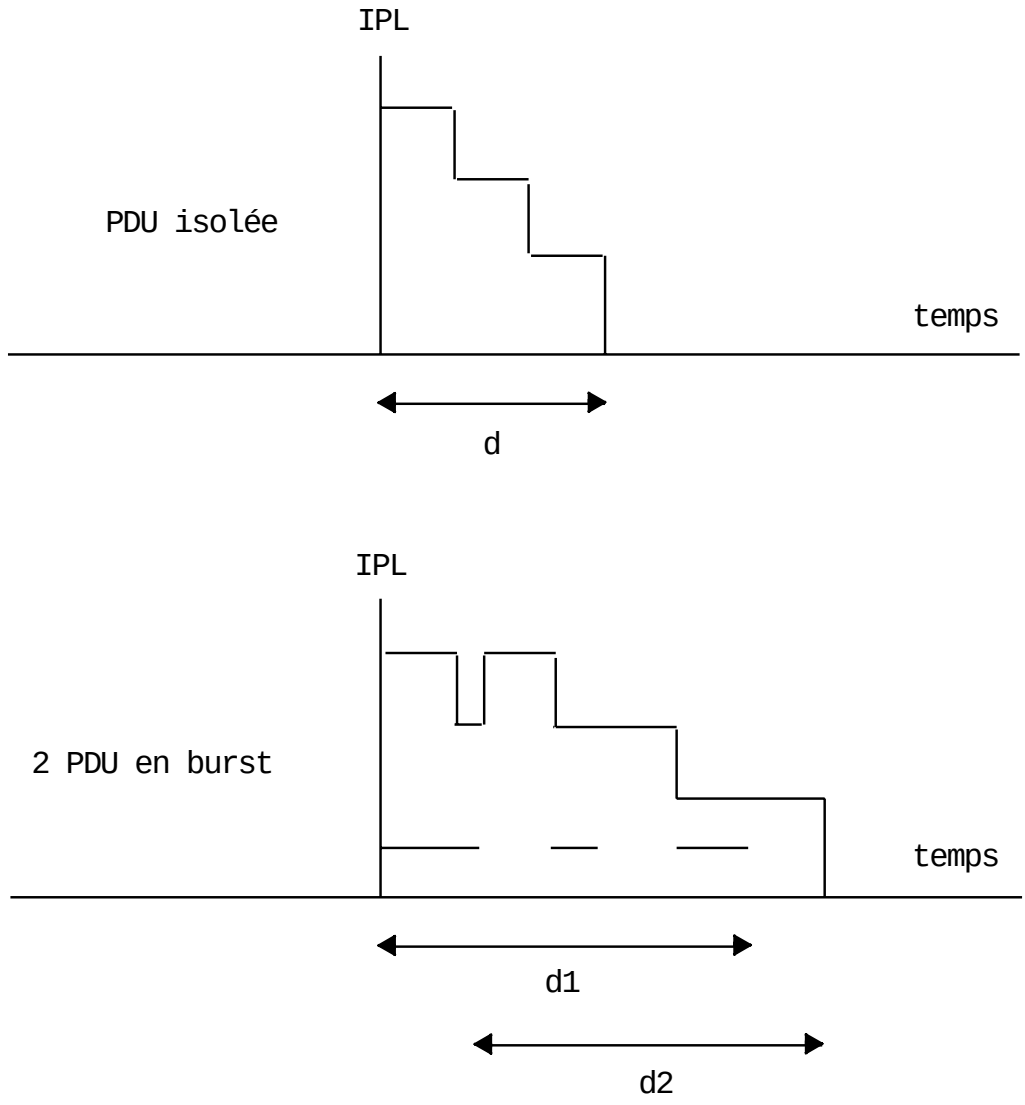
La copie entre l'espace d'E/S et la mémoire du système est optimisée par un mécanisme de type DMA utilisant l'instruction PCI MWI. Le tableau 4–1 résume les résultats obtenus.

Tableau 4–1: Mesure des livelocks

Processeur	Débit max	Livelock
400 MHz	500 Mbit/s	210 trames
550 MHz	660 Mbit/s	6 trames
700 MHz	754 Mbit/s	1 trame

Ces résultats mettent en évidence la relation étroite entre le débit max et l'importance des livelocks. La disparition des livelocks se traduit par une augmentation significative du débit max. La disparition des livelocks est ici la conséquence de l'augmentation de la fréquence du processeur du système.

Figure 4–1: Livelocks en réception



4.2 OPTIMISATIONS

Pour éviter les dysfonctionnements décrits au paragraphe précédent, les concepteurs d'interface optimisent :

- le calcul des checksums
- le traitement des entêtes
- la copie entre les espaces mémoire système et utilisateur
- les paramètres des protocoles

L'optimisation du calcul des checksums repose sur une bonne utilisation des possibilités offertes par le processeur. Dans le cas d'un processeur Alpha (CHANG et al., 1993), il est pertinent que :

- les opérandes soient des quadwords (64 bits)
- les données soient manipulées dans les caches
- le calcul soit pipeliné
- les retenues soient prises en compte à la fin

Dans les protocoles de transport connectés (TCP, OSI TP4...), une fois la connexion établie, la plupart des champs de l'entête des TPDUs qui se suivent sont prédictibles avec un taux d'erreur faible (figure 4–2). Or, la prédiction de l'entête réduit le nombre d'instructions à effectuer en émission et en réception (JACOBSON V., 1990). En conséquence, la prédiction est souvent utilisée dans les implémentations performantes de TCP.

Les copies entre les espaces mémoire système et utilisateur peuvent être très fortement réduites grâce à une technique de gestion de la mémoire virtuelle appelée Copy-on-write. Lorsque l'application et le système désirent partager une page mémoire, celle-ci est marquée Copy-on-write dans leurs tables de pages respectives. La page n'est effectivement copiée que si l'application ou le système désire la modifier. Le Copy-on-write est pour l'instant peu utilisé. En effet, il nécessite une refonte profonde du code logiciel. De plus, son effet sur la performance dépend des applications.

CHANG et al. (1993) montrent que certains paramètres des protocoles ont une forte influence sur la performance de l'interface réseau du DEC3000-AXP. Le débit et le taux de perte du récepteur UDP sont fonction de la taille des messages applicatifs (tableau 4–2). Le débit TCP est fonction de la taille de la fenêtre d'émission (tableau 4–3).

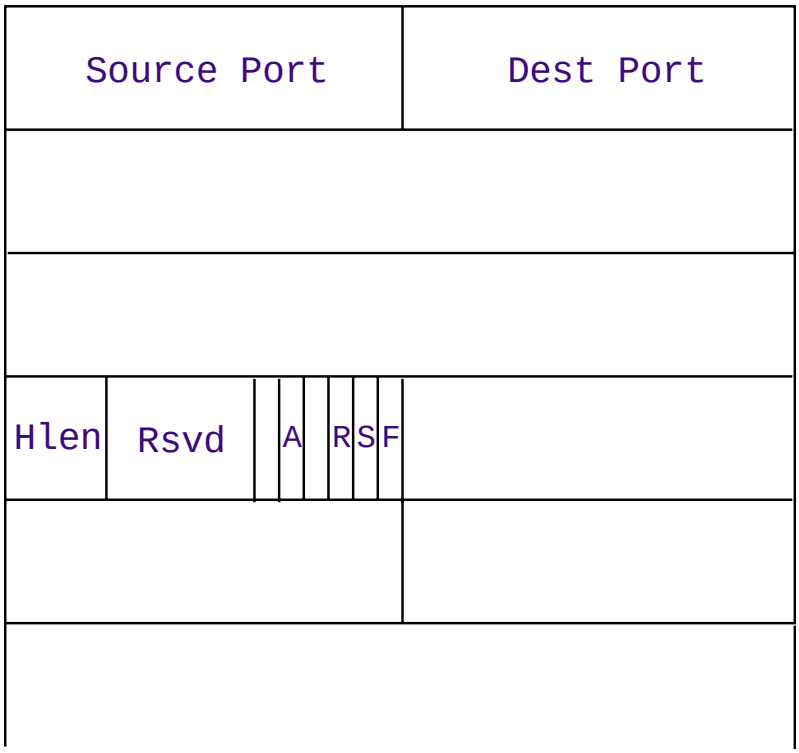
Tableau 4–2: Influence de la taille des messages sur le récepteur UDP

Message (octets)	Débit max (Mbit/s)	Taux de perte (%)
128	0.64	83.1
1024	23.77	46.9
4096	96.91	1.1
8192	97.01	0.6

Tableau 4–3: Influence de la taille de la fenêtre sur le débit de TCP

Fenêtre d'émission (Ko)	Débit max (Mbit/s)
16	40
32	73
64	78
150	95

Figure 4–2: Prédiction d’entête par le récepteur TCP



Annexe A

BIBLIOGRAPHIE

BANKS D. et PRUDENCE M. (1993) IEEE JSAC 10(1) pp. 191-202
A high performance network architecture for a PA-RISC workstation

CHANG C-H. et al. (1993) Digital Technical Journal 5(1) pp. 1-19
High-performance TCP/IP and UDP/IP networking in DEC OSF1 for Alpha AXP

CLARK D.D. et al. (1989) IEEE Communications Magazine 27(6) pp. 23-29
An analysis of TCP processing overhead

JACOBSON V. (1990) ACM Computer Communication Review 20(1) pp.13-15
4BSD header prediction

JACOBSON V. et al. (1991) IETF RFC 1323
TCP extensions for high performance

JAIN R. et ROUTHIER S.A. (1986) JSAC 4(6) pp.1162-1165
Packet trains: measurements and a new model for computer network traffic

LOEB M.L. et al. (2001) IEEE Network 15(2) pp. 42-47
Gigabit Ethernet PCI adapter performance

PARTRIDGE G. (1994) Ed. Addison-Wesley
Gigabit Networking

RAMAKRISHNAN K.K. (1993) IEEE JSAC 11(2) pp. 203-219
Performance considerations in designing network interfaces

Glossaire

DL Data Link