

Groupe de travail Réseau
Request for Comments : 893
Catégorie : Information
Traduction Claude Brière de L'Isle

Samuel J. Leffler
Michael J. Karels
University of California at Berkeley
avril 1984

Encapsulations d'en-queue

Statut de ce mémoire

La présente RFC discute des motivations de l'utilisation des "encapsulations d'en-queue" sur les réseaux de zone locale et décrit la mise en œuvre d'une telle encapsulation sur divers supports. Le présent document est seulement pour information. Ce n'est pas un protocole officiel de la communauté ARPA Internet.

Résumé

Une encapsulation de niveau liaison qui assure les propriétés d'alignement nécessaires pour l'utilisation efficace des facilités matérielles de mémoire virtuelle a été décrite. Ce format d'encapsulation est utilisé sur de nombreux systèmes et est une facilité standard dans le 4.2BSD UNIX. L'encapsulation fournit un mécanisme efficace pour des hôtes qui coopèrent sur un réseau pour obtenir des améliorations significatives des performances. L'utilisation de cette technique d'encapsulation exige actuellement une coopération uniforme de la part de tous les hôtes d'un réseau ; un mécanisme de négociation hôte par hôte peut heureusement être ajouté pour permettre aux hôtes qui y consentent d'utiliser l'encapsulation dans un environnement non uniforme.

1. Introduction

Une encapsulation d'en-queue est un format de paquet de niveau liaison employé par le 4.2BSD UNIX (entre autres). Une encapsulation d'en-queue, ou "en-queue", peut être générée par un système dans certaines conditions pour essayer de minimiser le nombre et la taille des opérations de copie de mémoire à mémoire effectuées par un hôte receveur lorsque il traite un paquet de données. Les en-queues sont strictement un format de paquet de niveau liaison et ne sont pas visibles (lorsque ils sont correctement mis en œuvre) dans tout traitement de protocole de niveau supérieur. La présente note cite les motivations qui sous-tendent l'encapsulation d'en-queue et décrit les formats de paquet d'encapsulation d'en-queue actuellement utilisés sur l'Ethernet expérimental à 3 Mbit/s, à 10 Mbit/s, et sur les réseaux en anneau V2LNI à 10 Mbit/s [1].

L'utilisation de l'encapsulation d'en-queue a été suggérée par Greg Chesson, et l'encapsulation décrite ici a été conçue par Bill Joy.

2. Motivation

Les en-queues sont motivés par la redondance qui peut être subie durant le traitement de protocole lorsque une ou plusieurs copies de mémoire à mémoire doivent être effectuées. La copie peut être requise à de nombreux niveaux du traitement, depuis le déplacement des données entre le support réseau et la mémoire de l'hôte jusqu'au passage des données entre le système d'exploitation et les espaces d'adresse d'utilisateur. Une mise en œuvre optimale de réseau s'attendrait à ne subir aucune opération de copie entre la livraison d'un paquet de données dans la mémoire de l'hôte et la présentation des données appropriées au processus receveur. Alors que de nombreux paquets ne peuvent pas être traités sans un certain nombre d'opérations de copie, lorsque l'ordinateur hôte fournit la prise en charge d'une gestion convenable de mémoire, il est souvent possible d'éviter de copier par une simple manipulation appropriée du matériel de mémoire virtuelle.

Dans un environnement de mémoire virtuelle transposée par page, deux prérequis sont habituellement nécessaires pour réaliser l'objectif de zéro opération de copie durant le traitement de paquet. Les données destinées à un agent receveur doivent être alignées sur une frontière de page et doivent avoir une taille qui soit un multiple de la taille de page du matériel (ou bourrée jusqu'à une limite de page). Cette dernière restriction suppose que la protection de la mémoire virtuelle soit maintenue au niveau de la page ; différentes architectures peuvent altérer ces prérequis.

Les données à transmettre à travers un réseau peuvent aisément être segmentées à la taille appropriée, mais, sauf si les informations d'en-tête de protocole d'encapsulation sont de taille fixée, l'alignement sur une frontière de page est virtuellement impossible. La taille des informations d'en-tête de protocole peut varier à cause de l'utilisation de plusieurs protocoles (ayant chacun un en-tête différent) ou elle peut varier en taille par accord (par exemple, lorsque des informations

facultatives sont incluses dans l'en-tête). Pour assurer l'alignement sur la page, ces informations d'en-tête qui préfixent les données destinées au receveur doivent être réduites à une taille fixée ; c'est normalement le cas au niveau liaison d'un réseau. En prenant toutes (si possible) les informations d'en-tête de longueur variable et en les déplaçant après le segment de données, un hôte envoyeur peut "faire de son mieux" pour permettre à l'hôte receveur l'opportunité de recevoir les données sur une frontière de page. Ce réarrangement des données au niveau liaison pour forcer les informations d'en-tête de longueur variable à se mettre "en queue" des données est la substance de l'encapsulation d'en-queue.

Il y a plusieurs hypothèses implicites dans l'argument ci-dessus.

1. L'hôte receveur doit vouloir accepter les en-queues. Comme c'est une encapsulation de niveau liaison, sauf si est effectuée une négociation d'hôte à hôte (de préférence au niveau liaison pour éviter de violer les principes de mise en couche) seuls certains hôtes seront capables de converser, ou leur communication peut être significativement perturbée si les paquets d'en-queue sont mélangés avec d'autres paquets.
2. Le coût de réception des données sur une frontière de page devrait être comparable à celui de réception des données sur une frontière non alignée sur la page. Si la redondance pour assurer l'alignement approprié est trop élevé, les économies faites en évitant les opérations de copie peuvent n'en valoir pas la peine.
3. La taille des informations d'en-tête de longueur variable devrait être significativement inférieure à celle du segment de données transmis. Il est possible de déplacer les informations d'en-queue sans les copier physiquement, mais souvent, les contraintes de mise en œuvre et les caractéristiques du matériel de réseau sous-jacent empêchent simplement de retransposer le ou les en-têtes.
4. La redondance de la copie de mémoire à mémoire qu'on s'attend à subir chez le receveur doit être assez significative pour compenser la complexité supplémentaire dans le logiciel de l'hôte aussi bien envoyeur que receveur.

Le premier point est bien connu et est le motif de la présente note.

On avait pensé à la négociation de l'utilisation des en-queues sur la base de l'hôte en utilisant une variante du protocole de résolution d'adresse [2] (augmentant en fait le protocole) mais à présent tous les systèmes qui utilisent des en-queues exigent que les hôtes qui partagent un support réseau acceptent uniformément les en-queues ou ne les transmettent jamais. (Cette dernière solution est facilement réalisée à l'amorçage dans les 4.2BSD sans modifier le code source du système d'exploitation.)

Le second point est (à notre connaissance) insignifiant. Bien qu'un hôte puisse ne pas être capable de tirer parti de l'alignement et des propriétés de taille d'un paquet d'en-queue, il ne devrait néanmoins jamais l'empêcher.

En ce qui concerne le troisième point, supposons que les informations d'en-tête d'en-queue soient copiées et non retransposées, et considérons la redondance de l'en-tête dans le protocole TCP/IP comme un exemple représentatif [3]. Si on suppose que les en-têtes de protocole TCP et IP font tous deux partie des informations d'en-tête de longueur variable, alors le plus petit paquet d'en-queue (généralisé par un VAX) aurait 512 octets de données et 40+ octets d'informations d'en-tête (plus l'en-tête d'en-queue décrit plus loin). Bien que l'en-tête d'en-queue puisse avoir des options IP et/ou TCP incluses, cela devrait normalement être rare (on pourrait s'attendre à ce que la plupart des options TCP, par exemple, soient incluses dans l'échange initial d'établissement de connexion et certainement beaucoup plus petites que 512 octets. Si le segment de données est supérieur, le ratio diminue et le gain attendu de moins de copies à l'extrémité réceptrice augmente. Étant donnée la redondance relative d'une opération de copie de mémoire à mémoire et du fait d'une manipulation de transposition de page (incluant l'invalidation d'une mémoire tampon de traduction) l'avantage est évident.

Le quatrième point est à notre avis en fait un non problème. Dans notre mise en œuvre, le supplément de code requis pour prendre en charge l'encapsulation d'en-queue se monte à environ une douzaine de lignes de code dans chaque "pilote d'interface réseau" de niveau liaison. L'amélioration de performances qui en résulte fait plus que compenser cet investissement mineur en logiciel.

On doit reconnaître que modifier le format de niveau réseau (et liaison normal) d'un paquet de la manière décrite force l'hôte receveur à mettre en mémoire tampon le paquet entier avant de le traiter. Une mise en œuvre habile peut analyser les en-têtes de protocole lorsque arrive le paquet pour trouver la taille réelle (ou le type de paquet de niveau réseau) d'un message entrant. Cela permet à ces mises en œuvre d'éviter de préallouer des mémoires tampon de la taille maximale pour les paquets entrants qu'elle peut reconnaître comme inacceptables. Les mises en œuvre qui analysent le format de niveau réseau au vol violent les principes de mise en couche qui ont été exaltés dans la conception pendant un certain temps (mais souvent violés dans la mise en œuvre). Le problème qu'il y a à retarder la reconnaissance du type de niveau liaison est une critique valide. Dans le cas de matériels réseau qui prennent en charge DMA, le paquet entier est cependant toujours reçu avant que ne commence le traitement.

3. Formats de paquet d'encapsulation d'en-queue

Dans cette section, on décrit les formats de paquet de niveau liaison utilisés sur l'Ethernet expérimental à 3 Mbit/s, et les réseaux Ethernet à 10 Mbit/s ainsi que le réseau en anneau V2LNI à 10 Mbit/s. Les formats utilisés dans chaque cas diffèrent seulement dans les valeurs du champ de format et de type utilisées dans chacun des en-têtes de réseau de zone locale.

Le format d'un paquet d'en-tête est montré dans le diagramme suivant.



LH :

C'est l'en-tête de réseau local de taille fixe. Pour un Ethernet à 10 Mbit/s, c'est l'en-tête Ethernet de 16 octets. Le champ Type dans l'en-tête indique le type de paquet (en-queue) et la longueur du segment de données.

Pour l'Ethernet à 10 Mbit/s, les types sont entre 1001 et 1010 (hexadécimal, soit 4096 et 4112 en décimal). Le type est calculé comme 1000 (en hexadécimal) plus le nombre de pages de données de 512 octets. Un maximum de 16 pages de données peuvent être transmises dans un seul paquet d'en-queue (8192 octets).

Données :

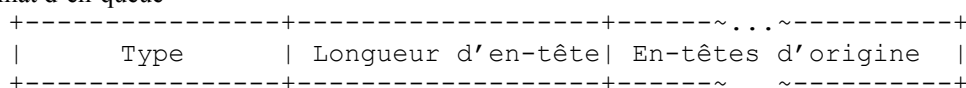
C'est la portion "données" du paquet. Ce sont normalement seulement les données à livrer au processus receveur (c'est-à-dire qu'elle ne contient pas d'information d'en-tête TCP ou IP). La taille des données est toujours un multiple de 512 octets.

TH :

C'est "l'en-queue". C'est en fait une composition des en-têtes du protocole original et d'un préfixe d'en-queue de taille fixe qui définit le type et la taille des données d'en-queue. Le format d'un en-queue est montré ci-dessous.

Les caractères (^) indiquent les frontières de page sur lesquelles l'hôte receveur va placer son entrée de mémoire tampon pour un alignement optimal lors de la réception d'un paquet d'en-queue. Le sous-programme de réception de niveau liaison est capable de localiser l'en-queue en utilisant la taille indiquée dans le champ de type de l'en-tête de niveau liaison. Le sous-programme receveur est supposé éliminer l'en-tête de niveau liaison et le préfixe d'en-queue, et de retransposer le segment de données d'en-queue sur le devant du paquet pour régénérer le format original du paquet de niveau réseau.

Format d'en-queue



Type : 16 bits

Le champ Type code le type original de niveau liaison du paquet transmis. C'est la valeur qui serait normalement placée dans l'en-tête de niveau liaison si un en-queue n'était pas généré.

Longueur d'en-tête : 16 bits

C'est le champ de longueur d'en-tête du segment de données de l'en-queue. Cela spécifie la longueur en octets des données de l'en-tête qui suit.

En-têtes d'origine : <longueur variable>

Ce sont les informations sur les en-têtes qui se trouvent logiquement devant le segment de données. Ce sont normalement les en-têtes de protocole de niveau réseau et transport.

Références

- [1] "The Ethernet - A Local Area Network", Version 1.0, Digital Equipment Corporation, Intel Corporation, Xerox Corporation, septembre 1980.
- [2] [RFC0826] D. Plummer, "Protocole de [résolution d'adresses Ethernet](#) : conversion des adresses de protocole réseau en adresses Ethernet à 48 bits pour transmission sur un matériel Ethernet", STD 37, novembre 1982.

- [3] [RFC0791] J. Postel, éd., "Protocole Internet - Spécification du [protocole du programme Internet](#)", STD 5, septembre 1981.