Groupe de travail Réseau
Request for Comments : 4821
Catégorie : Sur la voie de la normalisation
Traduction Claude Brière de L'Isle

M. Mathis, PSC
J. Heffner, PSC
mars 2007

Découverte de la MTU de chemin de couche de mise en paquet

Résumé
Le présent document décrit une méthode robuste pour la découverte de la MTU de chemin (PMTUD, *Path MTU Discovery*) qui s'appuie sur TCP ou une autre couche de mise en paquets pour sonder un chemin Internet avec des paquets de taille progressivement croissante. Cette méthode est décrite comme une extension des RFC 1191 et RFC 1981, qui spécifient la découverte de MTU de chemin fondée sur ICMP pour IP versions 4 et 6, respectivement.

Table des matières

1. Introduction

Le présent document décrit une méthode pour la découverte de la MTU de chemin de couche de mise en paquets (PLPMTUD, *Packetization Layer Path MTU Discovery*) qui est une extension des méthodes existantes de découverte de la MTU de chemin décrite dans les [RFC1191] et [RFC1981]. En l'absence de messages ICMP, la MTU appropriée est déterminée en commençant par de petits paquets et en sondant avec des paquets successivement plus grands. Le corps de l'algorithme est mis en œuvre par dessus IP, dans la couche transport (par exemple, TCP) ou un autre "protocole de mise en paquets" qui est chargé de déterminer les limites de paquet.

Le présent document ne met pas à jour la RFC 1191 ni la RFC 1981 ; cependant, comme il prend en charge un fonctionnement correct sans ICMP, il relâche implicitement certaines des exigences des algorithmes spécifiés dans ces documents.

Les méthodes décrites dans le présent document s'appuient sur des caractéristiques des protocoles existants. Elles s'appliquent à de nombreux protocoles de transport sur IPv4 et IPv6. Elles n'exigent pas de coopération des couches inférieures (excepté qu'elles sont cohérentes quant aux tailles de paquet acceptables) ou des homologues. Comme les méthodes s'appliquent seulement aux envoyeurs, les variantes de mise en œuvre ne causeront pas de problèmes d'interopérabilité.

Dans un souci de clarté, on préfère la terminologie de TCP et IPv6. Dans la section de terminologie, on présente aussi les termes et concepts analogues de IPv4 pour la terminologie IPv6. Dans quelques situations, on décrit les détails spécifiques qui sont différents entre IPv4 et IPv6.

Les mots clés "DOIT", "NE DOIT PAS", "EXIGE", "DEVRA", "NE DEVRA PAS", "DEVRAIT", "NE DEVRAIT PAS", "RECOMMANDE", "PEUT", et "FACULTATIF" en majuscules dans ce document sont à interpréter comme décrit dans le BCP 14, [RFC2119].

Le présent document est un produit du groupe de travail PMTUD *(Path MTU Discovery)* de l'IETF et il emprunte beaucoup aux RFC 1191 et RFC 1981 pour la terminologie, les idées, et une partie du texte.

2. Vue d'ensemble

La découverte de la MTU de chemin de couche de mise en paquet (PLPMTUD, *Packetization Layer Path MTU Discovery*) est une méthode pour que TCP ou d'autres protocoles de mise en paquets découvrent dynamiquement la MTU d'un chemin en le sondant avec des paquets progressivement plus gros. Elle est la plus efficace quand elle est utilisée en conjonction avec le mécanisme de découverte de la MTU de chemin fondée sur ICMP spécifiée dans les RFC 1191 et RFC 1981, mais résout beaucoup des problèmes de robustesse des techniques classiques car elle ne dépend pas de la livraison de messages ICMP.

Cette méthode est applicable à TCP et autres protocoles de niveau transport ou application qui sont chargés de choisir les limites de paquet (par exemple, les tailles de segment) et ont une structure d'accusé de réception qui livre à l'envoyeur des indications précises et en temps utile sur les paquets perdus.

La stratégie générale est que la couche de mise en paquet trouve une MTU de chemin appropriée en sondant le chemin avec des paquets de taille progressivement croissante. Si un paquet sonde est bien livré, la MTU de chemin effective est montée à la taille de la sonde.

La perte isolée d'un paquet sonde (avec ou sans message ICMP Paquet trop gros) est traitée comme une indication d'une limite de MTU, et non comme un indicateur d'encombrement. Dans ce seul cas, il est permis au protocole de mise en paquet de retransmettre les données manquantes sans ajuster la fenêtre d'encombrement.

Si il y a une fin de temporisation ou si des paquets supplémentaires sont perdus durant le processus de sondage, le sondage est considéré ne pas être concluant (par exemple, la sonde perdue n'indique pas nécessairement qu'elle dépassait la MTU de chemin). De plus, les pertes sont traitées comme toute autre indication d'encombrement : l'ajustement de fenêtre ou de taux est obligatoire selon les normes de contrôle d'encombrement pertinentes [RFC2914]. Le sondage peut reprendre après un délai qui est déterminé par la nature de la défaillance détectée.

La PLPMTUD utilise une technique de recherche pour trouver la MTU de chemin. Chaque sonde concluante réduit la

gamme de recherche de MTU, soit en élevant la limite inférieure suite à une sonde réussie, soit en abaissant la limite supérieure suite à l'échec de la sonde, convergeant vers la vraie MTU de chemin. Pour la plupart des couches de transport, la recherche devrait être arrêtée une fois que la gamme est assez resserrée pour que le bénéfice d'une MTU de chemin plus précise soit plus petit que les frais généraux de recherche pour la trouver.

L'échec d'une sonde le plus probable (et le moins sérieux) est dû à des pertes en relation avec de l'encombrement sur la liaison lors du sondage. Dans ce cas, il est approprié de réessayer une sonde de la même taille aussitôt que la couche de mise en paquet s'est pleinement adaptée à l'encombrement et a récupéré des pertes. Dans d'autres cas, des pertes supplémentaires ou des fins de temporisation indiquent des problèmes de la liaison ou de la couche de mise en paquet. Dans ces situations, il est désirable d'utiliser de plus longs délais selon la sévérité de l'erreur.

Un processus de vérification facultatif peut être utilisé pour détecter les situations où l'élévation de la MTU augmente le taux de perte de paquets. Par exemple, si une liaison parcourt plusieurs canaux physiques qui ont des MTU non cohérentes, il est possible qu'une sonde soit livrée même si elle est trop grande pour certains de canaux physiques. Dans ce cas, élever la MTU de chemin à la taille de la sonde peut causer de sévères pertes de paquet et des performances insondables. Après l'élévation de la MTU, la nouvelle taille de MTU peut être vérifiée en surveillant le taux de perte.

La PMTUD de couche de mise en paquet (PLPMTUD, *Packetization Layer Path MTU Discovery*) introduit une certaine souplesse dans la mise en œuvre de la découverte classique de la MTU de chemin. Elle peut être configurée pour effectuer juste la récupération de trous noirs ICMP pour accroître la robustesse de la découverte de la MTU de chemin classique, ou à l'autre extrême, tous les traitements ICMP peuvent être désactivés et la PLPMTUD peut complètement remplacer la découverte classique de MTU de chemin.

La découverte classique de MTU de chemin est sujette à des défaillances de protocole (ruptures de connexion) si des messages ICMP Paquet trop gros (PTB, *Packet Too Big*) ne sont pas délivrés ou traités pour une raison quelconque [RFC2923]. Avec la PLPMTUD, la découverte classique de MTU de chemin peut être modifiée pour inclure des vérifications de cohérence supplémentaires sans augmenter le risque de coupure de connexion due à des défaillances parasites des vérifications supplémentaires. De tels changements à la découverte classique de MTU de chemin sortent du domaine d'application, du présent document.

Dans le cas limite, tous les messages PTB ICMP pourraient être inconditionnellement ignorés, et la PLPMTUD peut être utilisée comme seule méthode pour découvrir la MTU de chemin. Dans cette configuration, la PLPMTUD est parallèle au contrôle d'encombrement. Un protocole de transport de bout en bout ajuste les propriétés du flux de données (taille de fenêtre ou taille de paquet) tout en utilisant les pertes de paquet pour déduire la justesse des ajustements. Cette technique semble être plus cohérente philosophiquement avec le principe de bout en bout de l'Internet que de s'appuyer sur des messages ICMP contenant des en-têtes transcrits de plusieurs couches de protocole.

La plupart des difficultés de mise en œuvre de la PLPMTUD surviennent parce qu'elle a besoin d'être mise en œuvre dans plusieurs endroits différents au sein d'un seul nœud. En général, chaque protocole de mise en paquets a besoin d'avoir sa propre mise en œuvre de PLPMTUD. De plus, le mécanisme naturel de partage des informations de MTU de chemin entre des connexions concurrentes ou suivantes est une antémémoire d'informations de chemin dans la couche IP. Les divers protocoles de mise en paquet ont besoin d'un moyen pour accéder et mettre à jour l'antémémoire partagée dans la couche IP. Le présent mémoire décrit la PLPMTUD en termes de sous systèmes principaux sans pleinement décrire comment ils sont assemblés dans une mise en œuvre complète.

La vaste majorité des détails de mise en œuvre décrits dans ce document sont des recommandations fondées sur l'expérience des versions antérieures de la découverte de la MTU de chemin. Ces recommandations sont motivées par le désir de maximiser la robustesse de la PLPMTUD en présence de conditions de réseau moins idéales que celles qui existent sur le terrain.

Le présent document ne contient pas une description complète d'une mise en œuvre. Il met seulement en scène les détails qui n'affectent pas l'interopérabilité avec les autres mises en œuvre et ont de forts critères d'optimisation imposés de l'extérieur (par exemple, la recherche de la MTU et des heuristiques de mise en antémémoire). D'autres détails sont explicitement inclus parce que il y a une solution évidente de mise en œuvre de remplacement qui ne fonctionne pas bien dans certains cas (éventuellement subtils).

La Section 3 donne un glossaire complet des termes.

La Section 4 décrit les détails de la PLPMTUD qui affectent l'interopérabilité avec d'autres normes ou protocoles Internet.

La Section 5 décrit comment partager la PLPMTUD en couches, et comment gérer l'antémémoire d'informations de chemin

dans la couche IP.

La Section 6 décrit les propriétés générales de couche de mise en paquets et les caractéristiques nécessaires pour mettre en œuvre la PLPMTUD.

La Section 7 décrit comment utiliser les sondes pour rechercher la MTU de chemin.

La Section 8 recommande l'utilisation de la fragmentation IPv4 dans une configuration qui imite la fonctionnalité IPv6, pour minimiser les futurs problèmes de migration à IPv6.

La Section 9 décrit une interface de programmation pour la mise en œuvre de la PLPMTUD dans des applications qui choisissent leurs propres limites de paquet et pour des outils capables de diagnostiquer les problèmes de chemin qui interfèrent avec la découverte de la MTU de chemin.

La Section 10 discute les détails de mise en œuvre des protocoles spécifiques, incluant TCP.

3. Terminologie

On utilise les termes suivants dans le présent document :

IP : IPv4 [RFC0791] ou IPv6 [RFC2460].

Nœud : appareil qui met en œuvre IP.

Couche supérieure : couche de protocole immédiatement au dessus de IP. Des exemples sont des protocoles de transport comme TCP et UDP, des protocoles de contrôle comme ICMP, des protocoles d'acheminement comme OSPF, et des protocoles Internet ou de couche inférieure "tunnelés" sur (c'est-à-dire, encapsulés dans) IP comme IPX, AppleTalk, ou IP lui-même.

Liaison : facilité ou support de communication sur lequel les nœuds peuvent communiquer à la couche de liaison, c'est-à-dire, la couche immédiatement en dessous de IP. Des exemples sont les Ethernets (simples ou pontés) les liaisons PPP, X.25, le relais de trame, ou les réseaux en mode de transfert asynchrone (ATM, *Asynchronous Transfer Mode*) et les "tunnels" de couche Internet (ou supérieure) comme les tunnels sur IPv4 ou IPv6. Occasionnellement, on utilise le terme légèrement plus général de "couche inférieure" pour ce concept.

Interface : rattachement d'un nœud à une liaison.

Adresse : identifiant de couche IP pour une interface ou ensemble d'interfaces.

Paquet : un en-tête IP plus sa charge utile.

MTU *(Maximum Transmission Unit)* : unité de transmission maximum, taille en octets du plus grand paquet IP, incluant l'en-tête IP et la charge utile, qui peut être transmise sur une liaison ou un chemin. Noter que cela pourrait être appelé de façon plus appropriée la MTU IP, pour être cohérent avec la façon dont les autres organisations de normalisation utilisent l'acronyme MTU.

MTU de liaison : l'unité maximum de transmission, c'est-à-dire, la taille maximum de paquet IP en octets, qui peut être convoyée en une fois sur une liaison. Il faut savoir que cette définition est différente de la définition utilisée par les autres organisations de normalisation. Pour les documents de l'IETF, la MTU de liaison est uniformément définie comme la MTU IP sur la liaison. Cela inclut l'en-tête IP, mais exclut les en-têtes de couche de liaison et tout autre tramage qui ne fait pas partie de IP ou de la charge utile IP. Il faut savoir que les autres organisations de normalisation définissent généralement la MTU de liaison comme incluant les en-têtes de couche de liaison.

Chemin : ensemble des liaisons traversées par un paquet entre un nœud de source et un nœud de destination.

MTU de chemin, ou PMTU : MTU de liaison minimum de toutes les liaisons dans un chemin entre un nœud de source et un nœud de destination.

Découverte de MTU de chemin classique : processus décrit dans les RFC 1191 et RFC 1981, dans lequel les nœuds

s'appuient sur les messages ICMP Paquet trop gros (PTB) pour apprendre la MTU d'un chemin.

Couche de mise en paquets : couche de la pile de réseau qui segmente les données en paquets.

PMTU effective : valeur estimée courante de la PMTU utilisée par une couche de mise en paquets pour la segmentation.

PLPMTUD *(Packetization Layer Path MTU Discovery)* : découverte de MTU de chemin de couche de mise en paquets. C'est la méthode décrite dans le présent document, qui est une extension de la découverte de PMTU classique.

Message PTB *(Packet Too Big)* : message ICMP qui rapporte qu'un paquet IP est trop gros pour être transmis. C'est le terme IPv6 qui correspond au message ICMP IPv4 "Fragmentation nécessaire et bit DF établi".

Flux : contexte dans lequel les algorithmes de découverte de MTU peuvent être impliqués. Ceci est naturellement une instance d'un protocole de mise en paquets, par exemple, un côté d'une connexion TCP.

MSS *(Maximum Segment Size)* : taille maximum de segment de TCP [RFC0793], c'est la taille maximum de charge utile disponible à la couche TCP. C'est normalement la MTU de chemin moins la taille des en-têtes IP et TCP.

Paquet sonde : paquet utilisé pour essayer une MTU plus grande sur un chemin.

Taille de sonde : taille d'un paquet utilisé pour sonder si une MTU plus grande est possible, incluant les en-têtes IP.

Trou de sonde : données de charge utile qui vont être perdues et ont besoin d'être retransmises si la sonde n'est pas livrée.

Fenêtre en tête : toutes données non acquittées dans un flux au moment de l'envoi d'une sonde.

Fenêtre en queue : toutes données dans un flux envoyées après une sonde, mais avant l'accusé de réception de la sonde.

Stratégie de recherche : heuristique utilisée pour choisir des tailles successives de sonde pour converger sur la MTU de chemin appropriée, comme décrit au paragraphe 7.3.

Temporisation point : temporisation où aucun des paquets transmis après un certain événement n'est acquitté par le receveur, incluant toute retransmission. Ceci est pris comme l'indication d'une condition de défaillance dans le réseau, comme un changement d'acheminement sur une liaison avec une plus petite MTU. Ceci est décrit plus en détails au paragraphe 7.7.

4. Exigences

Toutes les liaisons DOIVENT appliquer leur MTU : les liaisons qui pourraient livrer de façon non déterministe les paquets qui font plus que leur MTU DOIVENT éliminer de tels paquets.

Dans un lointain passé, il y avait un petit nombre d'appareils du réseau qui n'appliquaient pas la MTU, mais ne pouvaient pas livrer de façon fiable les paquets surdimensionnés. Par exemple, certains anciens répéteurs Ethernet transmettaient des paquets de taille arbitraire, mais ne pouvaient pas le faire de façon fiable du fait de la stabilité finie de l'horloge de données du matériel. C'est la seule exigence que la PLPMTUD fait peser sur les couches inférieures. Il est important que cette exigence soit explicite pour anticiper la future normalisation ou le futur déploiement de technologies qui pourraient être incompatibles avec la PLPMTUD.

Tous les hôtes DEVRAIENT utiliser la fragmentation IPv4 dans un mode qui imite la fonctionnalité IPv6. Toute fragmentation DEVRAIT être faite chez l'hôte, et tous les paquets IPv4, y compris les fragments, DEVRAIENT avoir le bit DF établi afin qu'ils ne soient pas fragmentés (à nouveau) dans le réseau. Voir la Section 8.

Les exigences ci-dessous ne s'appliquent qu'aux mises en œuvre qui incluent la PLPMTUD.

Pour utiliser la PLPMTUD, une couche de mise en paquets DOIT avoir un mécanisme de rapport de pertes qui fournit à l'envoyeur des indications précises et en temps utile des paquets perdus dans le réseau.

Les algorithmes normaux de contrôle d'encombrement DOIVENT rester en fonction dans toutes les conditions sauf quand seulement un paquet de sonde isolé est détecté comme perdu. Dans ce cas seulement, la réduction normale d'encombrement

(fenêtre ou taux de données) DEVRAIT être supprimée. Si d'autres pertes de données sont détectées, le contrôle d'encombrement standard DOIT avoir lieu.

La suppression du contrôle d'encombrement DOIT être limitée en débit afin qu'elle survienne moins fréquemment que le pire cas de taux de pertes pour le contrôle d'encombrement TCP à un débit de données comparable sur le même chemin, (c'est-à-dire, moins que le taux de pertes "TCP-friendly" [tcp-friendly]). Ceci DEVRAIT être appliqué en exigeant une voie moyenne minimum entre un ajustement d'encombrement supprimé (dû à un échec de sonde) et la prochaine tentative de sonde, qui est égale à un délai aller-retour pour chaque paquet permis par la fenêtre d'encombrement. Ceci est discuté plus en détails au paragraphe 7.6.2.

Chaque fois que la MTU est relevée, les variables d'état d'encombrement DOIVENT être rééchelonnées afin de ne pas relever la taille de fenêtre en octets (ou le taux de données en octets par seconde).

Chaque fois que la MTU est réduite (par exemple, lors du traitement de messages PTB ICMP) la variable d'état d'encombrement DEVRAIT être rééchelonnée afin de ne pas relever la taille de fenêtre en paquets.

Si la PLPMTUD met à jour la MTU pour un chemin particulier, toutes les sessions de couche de mise en paquets qui partagent la représentation de chemin (comme décrit au paragraphe 5.2) DEVRAIENT être notifiées d'utiliser la nouvelle MTU et faire les ajustements de contrôle d'encombrement requis.

Toutes les mises en œuvre DOIVENT inclure des mécanismes pour que les applications transmettent sélectivement les paquets plus grands que la MTU de chemin courante effective, mais plus petits que la MTU de liaison du premier bond. Ceci est nécessaire pour mettre en œuvre la PLPMTUD en utilisant un protocole sans connexion dans une application et pour mettre en œuvre des outils de diagnostic qui ne s'appuient pas sur la mise en œuvre de la découverte de la MTU de chemin du système d'exploitation. Voir les détails à la Section 9.

Les mises en œuvre PEUVENT utiliser des heuristiques différentes pour choisir la MTU de chemin initiale effective pour chaque protocole. Les protocoles sans connexion et les protocoles qui ne prennent pas en charge la PLPMTUD DEVRAIENT avoir leur propre valeur par défaut pour la MTU de chemin initiale effective, qui peut être réglée à une valeur plus prudente (plus petite) que la valeur initiale utilisée par TCP et les autres protocoles qui conviennent bien pour la PLPMTUD. Il DEVRAIT y avoir des limites par protocole et par chemin sur la MTU de chemin initiale effective (eff_pmtu) et la limite supérieure de recherche (search_high). Voir les détails au paragraphe 7.2.

5. Mise en couches

La découverte de la MTU de chemin de couche de mise en paquets est très facilement mise en œuvre en partageant ses fonctions entre les couches. La couche IP est le meilleur endroit pour conserver l'état partagé, collecter les messages ICMP, garder trace des tailles d'en-têtes IP, et gérer les informations de MTU fournies par les interfaces de couche de liaison. Cependant, les procédures qu'utilise la PLPMTUD pour les sondages et la vérification de la MTU de chemin sont très étroitement couplées aux caractéristiques des couches de mise en paquet, comme la récupération des données et les automates à états de contrôle d'encombrement.

Noter que cette approche de mise en couches est une extension directe de l'avis des spécifications actuelles de PMTUD dans les RFC 1191 et RFC 1981.

5.1 Prise en compte des tailles d'en-tête

La façon dont la PLPMTUD opère à travers plusieurs couches exige un mécanisme pour tenir compte des tailles d'en-têtes à toutes les couches entre IP et la couche de mise en paquets (incluse). Quand elle transmet des paquets qui ne sont pas de sondage, il est suffisant que la couche de mise en paquets s'assure d'une limite supérieure sur la taille finale de paquet IP, de façon à ne pas excéder la MTU de chemin courante effective. Toutes les couches de mise en paquets qui participent à la découverte de MTU de chemin classique ont déjà cette exigence. Quand elle fait un sondage, la couche de mise en paquets DOIT déterminer la taille finale du paquet de sonde incluant les en-têtes IP. Cette exigence est spécifique de la PLPMTUD, et la satisfaire peut exiger une communication inter couches supplémentaire dans les mises en œuvre existantes.

5.2 Mémorisation des informations de PMTU

Le présent mémoire utilise le concept de "flux" pour définir la portée des algorithmes de découverte de la MTU de chemin.

Pour de nombreuses mises en œuvre, un flux va naturellement correspondre à une instance de chaque protocole (c'est-à-dire, chaque connexion ou session). Dans de telles mises en œuvre, les algorithmes décrits dans ce document sont effectués dans chaque session pour chaque protocole. La PMTU observée (eff_pmtu au paragraphe 7.1) PEUT être partagée entre différents flux avec une représentation de chemin commune.

Autrement, la PLPMTUD pourrait être mise en œuvre de façon à ce que son état complet soit associé à la représentation de chemin. Une telle mise en œuvre pourrait utiliser plusieurs connexions ou sessions pour chaque séquence de sondage. Cette approche va probablement converger beaucoup plus rapidement dans certains environnements, comme lorsque une application utilise de nombreuses petites connexions, dont chacune est trop courte pour achever le processus complet de découverte de la MTU de chemin.

Au sein d'une seule mise en œuvre, différents protocoles peuvent utiliser l'une ou l'autre de ces deux approches. Du fait des différences spécifiques du protocole des contraintes de génération des sondes (paragraphe 6.2) et de l'algorithme de recherche de la MTU (paragraphe 7.3) il peut n'être pas faisable que différents protocoles de couche de mise en paquets partagent l'état de PLPMTUD. Cela suggère qu'il est possible que certains protocoles partagent l'état de sondage, mais que d'autres protocoles puissent seulement partager la PMTU observée. Dans ce cas, les différents protocoles vont avoir des propriétés de convergence de PMTU différentes.

La couche IP DEVRAIT être utilisée pour mémoriser la valeur de PMTU mise en antémémoire et d'autre état partagé comme les valeurs de MTU rapportées par les messages PTB ICMP. Idéalement, cet état partagé devrait être associé à un chemin spécifique traversé par les paquets échangés entre les nœuds de source et de destination. Cependant, dans la plupart des cas un nœud ne va pas avoir assez d'informations pour identifier complètement et précisément un tel chemin. Un nœud doit plutôt associer une valeur de PMTU à une représentation locale d'un chemin. Il appartient à la mise en œuvre de choisir la représentation locale d'un chemin.

Une mise en œuvre PEUT utiliser l'adresse de destination comme représentation locale d'un chemin. La valeur de PMTU associée à une destination serait la PMTU minimum apprise sur l'ensemble de tous les chemins en usage vers cette destination. L'ensemble des chemins en usage vers une destination particulière est supposé être petit, consistant dans de nombreux cas en un seul chemin. Cette approche va résulter en l'utilisation de paquets de taille optimale par destination, et s'intègre bien dans le modèle conceptuel d'un hôte comme décrit dans la [RFC2461] : une valeur de PMTU pourrait être mémorisée avec l'entrée correspondante dans l'antémémoire de destination. Comme les traducteurs d'adresse réseau (NAT, *Network Address Translator*) et autre formes de boîtiers de médiation peuvent afficher des PMTU différentes simultanément à une seule adresse IP, la valeur minimum DEVRAIT être mémorisée.

Les numéros de réseau ou de sous réseau NE DOIVENT PAS être utilisés comme représentations d'un chemin, parce que il n'y a pas de mécanisme général pour déterminer le gabarit de réseau chez l'hôte distant.

Pour les paquets en acheminement de source (c'est-à-dire, les paquets qui contiennent un en-tête d'acheminement IPv6, ou les options IPv4 acheminement lâche et enregistrement de chemin (LSRR, *Loose Source and Record Route*) ou routage strict et enregistrement du chemin (SSRR, *Strict Source and Record Route*)) l'acheminement de source PEUT de plus qualifier la représentation locale d'un chemin. Une mise en œuvre PEUT utiliser les informations de route de source dans la représentation locale d'un chemin.

Si des flux IPv6 sont en usage, une mise en œuvre PEUT utiliser le triplet d'étiquette de flux, adresse de source et adresse de destination [RFC2460], [RFC3697]comme représentation locale d'un chemin. Une telle approche pourrait théoriquement résulter en l'utilisation de paquets de taille optimale sur la base du flux, fournissant une plus fine granularité que les valeurs de MTU tenues sur la base de la destination.

5.3 Prise en compte de IPsec

Le présent document ne fait pas d'hypothèse sur le placement de la sécurité IP (IPsec) [RFC2401], qui se tient logiquement entre la couche IP et la couche de mise en paquets. Une mise en œuvre de PLPMTUD peut traiter IPsec soit au titre de la couche IP soit au titre de la couche de mise en paquets, pour autant que la prise en compte soit cohérente dans la mise en œuvre. Si IPsec est traité comme faisant partie de la couche IP, alors chaque association de sécurité à un nœud distant peut devoir être traité comme un chemin séparé. Si IPsec est traité comme faisant partie de la couche de mise en paquets, la taille de l'en-tête IPsec DOIT être incluse dans les calculs de taille d'en-tête de couche de mise en paquets.

5.4 Diffusion groupée

Dans le cas d'une adresse de destination de diffusion groupée, des copies d'un paquet peuvent traverser de nombreux chemins différents pour atteindre de nombreux nœuds différents. La représentation locale du "chemin" pour une destination de diffusion groupée doit en fait représenter un ensemble de chemins potentiellement grand.

Au minimum, une mise en œuvre PEUT conserver une seule valeur de MTU à utiliser pour tous les paquets en diffusion groupée originaires du nœud. Cette MTU DEVRAIT être suffisamment petite car il est attendu qu'elle soit inférieure à la MTU de chemin de tous les chemins qui constituent l'arborescence de diffusion groupée. Si une MTU de chemin de moins que la MTU de diffusion groupée configurée est apprise via des moyens en envoi individuel, la MTU de diffusion groupée PEUT être réduite à cette valeur. Cette approche va probablement résulter en l'utilisation de paquets plus petits que nécessaire pour de nombreux chemins.

Si l'application qui utilise la diffusion groupée obtient des rapports de livraison complets (ce qui est peu probable parce que cette exigence a de mauvaises propriétés d'adaptabilité) la PLPMTUD PEUT être mise en œuvre dans des protocoles de diffusion groupée tels que la plus petite MTU de chemin apprise dans un groupe devienne la MTU effective pour ce groupe.

6. Propriétés communes de mise en paquet

Cette Section décrit les propriétés et caractéristiques générales de couche de mise en paquets nécessaires pour mettre en œuvre la PLPMTUD. Elle décrit aussi des problèmes de mise en œuvre qui sont communs à toutes les couches de mise en paquet.

6.1 Mécanisme de détection de perte

Il est important que la couche de mise en paquets ait un mécanisme bien rythmé et robuste pour détecter et rapporter les pertes. La PLPMTUD fait des ajustements de MTU sur la base des pertes détectées. Tout délai ou imprécision des notifications de pertes va probablement résulter en décisions incorrectes de MTU ou en convergence lente. Il est important que le mécanisme puisse distinguer de façon sûre la perte isolée de juste une sonde des autres pertes dans les fenêtre de tête et de queue de la sonde.

Le mieux est que les protocoles de mise en paquets utilisent un mécanisme explicite de détection de pertes comme un tableau des résultats d'accusé de réception sélectif (SACK, *Selective Acknowledgment*) [RFC3517] ou un vecteur d'accusés de réception *(ACK Vector)* [RFC4340] pour distinguer les pertes réelles des données réordonnées, bien que des mécanismes implicites tels que le comptage des accusés de réception dupliqués de style TCP Reno soit suffisant.

La PLPMTUD peut aussi être mise en œuvre dans des protocoles qui s'appuient sur des temporisateurs comme principal mécanisme pour la récupération de pertes ; cependant, des temporisateurs NE DEVRAIENT PAS être utilisés comme mécanisme principal pour les indications de pertes sauf si il n'y a pas d'autre solution de remplacement.

6.2. Génération de sondes

Plusieurs façons sont possibles pour altérer les couches de mise en paquets pour générer des sondes. Les différentes techniques subissent des frais généraux différents dans trois domaines : la difficulté de générer le paquet sonde (en termes de complexité de mise en œuvre de couche de mise en paquets et de mouvements de données supplémentaires) la possibilité de capacité réseau supplémentaire consommée par les sondes, et les frais généraux de récupération de défaillances de sondes (à la fois dans le réseau et dans le protocole).

Certains protocoles pourraient être étendus pour permettre un bourrage arbitraire avec des données factices. Cela simplifie considérablement la mise en œuvre parce que le sondage peut être effectué sans participation des couches supérieures et si la sonde échoue, les données manquantes (le "trou de sonde") sont assurées de tenir dans la MTU courante quand elles sont retransmises. Ceci est probablement la méthode la plus appropriée pour les protocoles qui prennent en charge des options de longueur arbitraire ou le multiplexage au sein du protocole lui-même.

De nombreux protocoles de couche de mise en paquets peuvent porter de purs messages de contrôle (sans aucune donnée provenant des couches de protocole supérieures) qui peuvent être bourrés à des longueurs arbitraires. Par exemple, le tronçon SCTP PAD peut être utilisé de cette manière (voir le paragraphe 10.2). Cette approche présente l'avantage que rien

n'a besoin d'être retransmis si la sonde est perdue.

Ces techniques ne fonctionnent pas pour TCP, parce que il n'y a pas de champ de longueur séparé ou autre mécanisme pour différencier entre bourrage et données de charge utile réelles. Avec TCP, la seule approche est d'envoyer des données de charge utile supplémentaires dans un segment surdimensionné. Il y a au moins deux variantes de cette approche, discutées au paragraphe 10.1.

Dans quelques cas, il peut n'y avoir aucun mécanisme raisonnable pour générer des sondes au sein du protocole de couche de mise en paquets lui-même. En dernier recours, il peut être possible de s'appuyer sur un protocole auxiliaire, comme un ECHO ICMP ("ping") pour envoyer des paquets sondes. Voir au paragraphe 10.3 une discussion de cette approche.

7. Méthode de sondage

Cette Section décrit les détails de la méthode de sondage de la MTU, incluant comment envoyer les sondes et traiter les indications d'erreur nécessaires pour chercher la MTU de chemin.

7.1 Gammes de tailles de paquet

Le présent document décrit la méthode de sondage utilisant trois variables d'état :

search_low : la plus petite taille utile de sonde, moins un. Le réseau est supposé être capable de livrer des paquets de taille search_low.

search_high : la plus grande taille utile de sonde. Les paquets de taille search_high sont supposés être trop grands pour que le réseau les livre.

eff_pmtu : la PMTU effective pour ce flux. C'est le plus grand paquet non sonde permis par la PLPMTUD pour le chemin.

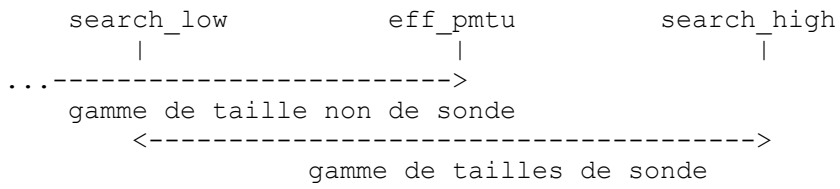

Figure 1

Quand elle transmet des non sondes, la couche de mise en paquets DEVRAIT créer des paquets d'une taille inférieure ou égale à eff_pmtu.

Quand elle transmet des sondes, la couche de mise en paquets DOIT choisir une taille de sonde supérieure à search_low et inférieure ou égale à search_high.

Quand elle sonde vers l'aval, eff_pmtu est toujours égale à search_low. Dans les autres états, comme les conditions initiales, après le traitement d'un message ICMP PTB ou à la suite d'une PLPMTUD sur un autre flux partageant la même représentation de chemin, eff_pmtu peut être différente de search_low. Normalement, eff_pmtu va être supérieure ou égale à search_low et inférieure à search_high. Il est généralement attendu mais pas exigé que la taille de la sonde soit supérieure à eff_pmtu.

Pour les conditions initiales quand il n'y a pas d'informations sur le chemin, eff_pmtu peut être supérieure à search_low. La valeur initiale de search_low DEVRAIT être assez faible, mais les performances peuvent être meilleures si eff_pmtu commence à une valeur supérieure, moins prudente. Voir le paragraphe 7.2.

Si eff_pmtu est supérieure à search_low, il est explicitement permis d'envoyer des paquets non de sonde plus grands que search_low. Quand un tel paquet est acquitté, il est effectivement une "sonde implicite" et search_low DEVRAIT être relevé à la taille du paquet acquitté. Cependant, si une "sonde implicite" est perdue, elle NE DOIT PAS être traitée comme un échec de sonde comme le serait une vraie sonde. Si eff_pmtu est trop grande, cette condition va seulement être détectée avec des messages PTB ICMP ou la découverte de trous noirs (voir au paragraphe 7.7).

7.2 Choix des valeurs initiales

La valeur initiale pour search_high DEVRAIT être le plus grand paquet possible qui pourrait être accepté par le flux. Ce peut être limité par la MTU de l'interface locale, par un mécanisme explicite du protocole tel que l'option TCP MSS, ou par une limite intrinsèque comme la taille d'un champ Longueur du protocole. De plus, la valeur initiale pour search_high PEUT être limitée par une option de configuration pour empêcher des sondes au dessus d'une taille maximum. Search_high va probablement être la même que la MTU de chemin initiale telle que calculée par l'algorithme classique de découverte de MTU de chemin.

Il est RECOMMANDÉ que search_low soit initialement réglé à une taille de MTU qui va probablement fonctionner sur une très large gamme d'environnements. Étant données les technologies d'aujourd'hui, une valeur de 1024 octets est probablement assez sûre. La valeur initiale pour search_low DEVRAIT être configurable.

Un fonctionnement approprié de la découverte de la MTU de chemin est critique pour le fonctionnement robuste et efficace de l'Internet. Tout changement majeur (comme décrit dans ce document) a un potentiel de perturbation très important si il cause des changements inattendus du comportement des protocoles. Le choix d'une valeur initiale pour eff_pmtu détermine dans quelle mesure un comportement de mise en œuvre de PLPMTUD ressemble à la PMTUD classique dans les cas où la méthode classique est suffisante.

Une configuration prudente serait de régler eff_pmtu à search_high, et de s'appuyer sur les messages PTB ICMP pour régler la eff_pmtu en baisse comme approprié. Dans cette configuration, la PMTUD classique est pleinement fonctionnelle et la PLPMTUD est seulement invoquée pour récupérer de trous noirs ICMP par la procédure décrite au paragraphe 7.7.

Dans certains cas, où il est connu que la PMTUD classique va probablement échouer (par exemple, si des messages PTB ICMP sont administrativement désactivés pour des raisons de sécurité) utiliser une eff_pmtu initiale petite évitera les coûteuses fins de temporisation exigées pour la détection de trous noirs. Le compromis est qu'utiliser une eff_pmtu initiale plus petite que nécessaire peut causer une réduction des performances.

Noter que la eff_pmtu initiale peut être toute valeur dans la gamme de search_low à search_high. Une eff_pmtu initiale de 1400 octets pourrait être un bon compromis parce qu'elle serait sûre pour presque tous les tunnels sur tous les appareils de réseautage courants, et quand même proche de la MTU optimale pour la majorité des chemins dans l'Internet d'aujourd'hui. Cela pourrait être amélioré en utilisant des statistiques d'autres flux récents : par exemple, la eff_pmtu initiale pour un flux pourrait être réglée à la médiane de la taille de sonde pour toutes les sondes récentes réussies.

Comme le coût de la PLPMTUD est dominé par les frais généraux spécifiques de protocole de la génération et du traitement des sondes, il est probablement souhaitable que chaque protocole ait sa propre heuristique pour choisir la eff_pmtu initiale. Il est particulièrement important que les protocoles sans connexion et autres protocoles qui ne peuvent pas recevoir de claires indications des trous noirs ICMP utilisent des valeurs initiales prudentes (plus petites) pour eff_pmtu, comme décrit au paragraphe 10.3.

Il DEVRAIT y avoir des options de configuration par protocole et par chemin pour outrepasser les valeurs initiales pour eff_pmtu et les autres variables d'état de PLPMTUD.

7.3 Choix de taille de sonde

La sonde peut avoir une taille quelconque dans la "gamme de taille de sonde" décrite ci-dessus. Cependant, un certain nombre de facteurs affectent le choix d'une taille appropriée. Une stratégie simple pourrait être de faire une recherche binaire en divisant par deux la gamme de taille de sonde à chaque sonde. Cependant, pour certains protocoles, comme TCP, les échecs de sonde sont plus coûteux que ceux qui réussissent, car les données d'une sonde qui échoue vont devoir être retransmises. Pour ces protocoles, une stratégie qui augmente la taille des sondes en plus petits incréments pourrait avoir de moindres frais généraux. Pour de nombreux protocoles, à la couche de mise en paquets et au dessus, l'avantage de tailles de MTU croissantes peut suivre une fonction en escalier telle qu'il n'est pas avantageux de sonder du tout dans certaines régions.

Pour une optimisation, il peut être approprié de sonder à certaines tailles de MTU courantes ou attendues, par exemple, 1500 octets pour l'Ethernet standard, ou 1500 octets moins les tailles d'en-tête pour les protocoles de tunnel.

Certains protocoles peuvent utiliser d'autres mécanismes pour choisir les tailles de sondes. Par exemple, les protocoles qui

ont certaines tailles de bloc naturel de données pourraient simplement assembler des messages à partir d'un certain nombre de blocs jusqu'à ce que la taille totale soit inférieure à search_high, et si possible supérieure à search_low.

Chaque couche de mise en paquets DOIT déterminer quand le sondage a convergé, c'est-à-dire, quand la gamme de tailles de sondes est assez petite pour que d'autres sondes ne vaillent plus leur coût. Quand le sondage a convergé, un temporisateur DEVRAIT être établi. Quand le temporisateur arrive à expiration, search_high devrait être remis à sa valeur initiale (décrite ci-dessus) afin que le sondage puisse reprendre. Donc, si le chemin change, augmenter la MTU de chemin, puis le flux va finalement en tirer parti. La valeur de ce temporisateur NE DOIT PAS être de moins de 5 minutes et il est recommandé qu'elles soit de 10 minutes, selon la RFC 1981.

7.4 Pré-conditions de sondage

Avant d'envoyer une sonde, le flux DOIT satisfaire au moins les conditions suivantes :
- o Il n'y a pas de sonde ou perte en instance.
- o Si la dernière sonde a échoué ou était non concluante, alors la temporisation de sonde a expiré (paragraphe 7.6.2).
- o La fenêtre disponible est supérieure à la taille de sonde.
- o Pour un protocole qui utilise des données dans la bande pour le sondage, assez de données sont disponibles pour envoyer la sonde.

De plus, les algorithmes de détection à temps de pertes dans la plupart des protocoles ont des pré-conditions qui DEVRAIENT être satisfaites avant d'envoyer une sonde. Par exemple, la retransmission rapide de TCP n'est pas assez robuste sauf si il y a suffisamment de segments qui suivent une sonde ; c'est-à-dire que l'envoyeur DEVRAIT avoir assez de données en file d'attente et suffisamment de fenêtre de réception pour envoyer la sonde plus au moins Tcprexmtthresh [RFC2760] segments supplémentaires. Cette restriction peut interdire le sondage dans certains états de protocole, comme trop proches de la fin d'une connexion, ou quand la fenêtre est trop petite.

Les protocoles PEUVENT retarder l'envoi de non sondes afin d'accumuler assez de données pour satisfaire les pré-conditions de sondage. L'algorithme d'envoi retardé DEVRAIT utiliser une technique d'auto adaptation pour limiter de façon appropriée le temps de retard des données. Par exemple, les ACK de retour peuvent être utilisés pour empêcher la fenêtre de manquer de plus que la quantité de données nécessaires pour le sondage.

7.5 Procédure de sondage

Une fois qu'une taille de sonde dans la gamme appropriée a été choisie, et que les pré-conditions ci-dessus sont satisfaites, la couche de mise en paquets PEUT faire un sondage. Pour ce faire, elle crée un paquet sonde tel que sa taille, incluant les en-têtes IP les plus externes, soit égale à la taille de la sonde. Après l'envoi de la sonde, elle attend une réponse, qui va avoir un des résultats suivants :

Succès : la sonde est acquittée comme ayant été reçue par l'hôte distant.

Échec : un mécanisme de protocole indique que la sonde a été perdue, mais aucun paquet dans la fenêtre de tête ou de queue n'a été perdu.

Échec de temporisation : un mécanisme de protocole indique que la sonde a été perdue, et aucun paquet dans la fenêtre de tête n'a été perdu, mais il est incapable de déterminer si des paquets dans la fenêtre de queue ont été perdus. Par exemple, la perte est détectée par une fin de temporisation, et la retransmission retour-n est utilisée.

Non concluant : la sonde a été perdue en plus d'autres paquets dans les fenêtres de tête ou de queue.

7.6 Réponse au résultat du sondage

Quand une sonde s'est achevée, le résultat DEVRAIT être traité comme suit, catégorisé par le type de résultat de la sonde.

7.6.1 Succès du sondage

Quand la sonde est livrée, c'est l'indication que la MTU de chemin est au moins aussi grande que la taille de la sonde. On règle search_low à la taille de sonde. Si la taille de sonde est supérieure à eff_pmtu, on élève eff_pmtu à la taille de sonde. La taille de sonde pourrait être inférieure à eff_pmtu si le flux n'a pas utilisé la pleine MTU du chemin parce que il est

soumis à d'autres limitations, comme les données disponibles dans une session interactive.

Noter que si les paquets d'un flux sont acheminés via plusieurs chemins, ou sur un chemin avec une MTU non déterministe, la livraison d'un seul paquet sonde n'indique pas que tous les paquets de cette taille vont être livrés. Pour être robuste dans ce cas, la couche de mise en paquets DEVRAIT faire une vérification de la MTU, comme décrit au paragraphe 7.8.

7.6.2 Échec du sondage

Quand seule la sonde est perdue, c'est traité comme l'indication que la MTU de chemin est plus petite que la taille de la sonde. Dans ce seul cas, la perte NE DEVRAIT PAS être interprétée comme un signal d'encombrement.

En l'absence d'autre indication, on règle search_high à la taille de la sonde moins un. La eff_pmtu pourrait être supérieure à la taille de sonde si le flux n'avait pas utilisé la MTU complète du chemin parce qu'il est soumis à quelque autre limitation, comme les données disponibles dans une session interactive. Si eff_pmtu est supérieure à la taille de sonde, eff_pmtu DOIT être réduite à pas plus que search_high, et DEVRAIT être réduite à search_low, car eff_pmtu a été déterminée comme étant invalide, comme après une temporisation point (voir au paragraphe 7.7).

Si un message ICMP PTB est reçu qui correspond au paquet sonde, alors search_high et eff_pmtu PEUVENT être réglées d'après la valeur de MTU indiquée dans le message. Noter que le message ICMP peut être reçu soit avant, soit après l'indication de perte du protocole.

Un événement d'échec de sonde est la situation dans laquelle la couche de mise en paquets DEVRAIT ignorer la perte comme signal d'encombrement. Parce qu'il y a un petit risque que la suppression du contrôle d'encombrement puisse avoir des conséquences imprévues (même pour une perte isolée) il est EXIGÉ que les événements d'échec de sonde soient moins fréquents que la période normale pour les pertes dans le contrôle d'encombrement standard. Spécifiquement, après un événement d'échec de sonde et la suppression du contrôle d'encombrement, la PLPMTUD NE DOIT PAS sonder à nouveau jusqu'à ce qu'un intervalle supérieur à l'intervalle attendu s'écoule entre les événements de contrôle d'encombrement. Voir les détails à la Section 4. La plus simple estimation de l'intervalle jusqu'au prochain événement d'encombrement est le même nombre d'allers-retours que la fenêtre courante d'encombrement en paquets.

7.6.3 Échec de fin de temporisation de sondage

Si la perte a été détectée avec une fin de temporisation et réparée avec n retransmissions de retour, la réduction de la fenêtre d'encombrement va alors être nécessaire. Le prix relativement élevé dans ce cas d'un échec de sonde peut mériter un intervalle de temps plus long jusqu'à la prochaine sonde. Un intervalle de temps de cinq fois le cas d'échec sans fin de temporisation (paragraphe 7.6.2) est RECOMMANDÉ.

7.6.4 Sondage non concluant

La présence d'autres pertes proches de la perte d'une sonde peut indiquer que la perte de la sonde était due à l'encombrement plutôt qu'à une limitation de MTU. Dans ce cas, les variables d'état eff_pmtu, search_low, et search_high NE DEVRAIENT PAS être mises à jour, et la même taille de sonde DEVRAIT être tentée à nouveau aussitôt que les préconditions de sondage sont satisfaites (c'est-à-dire, une fois que la couche de mise en paquets n'a plus de pertes non récupérées en instance). À ce point, il est particulièrement approprié de re-sonder car la fenêtre d'encombrement du flux va être à son point le plus bas, minimisant la probabilité de pertes dues à l'encombrement.

7.7 Fin de temporisation persistante

Dans toutes les conditions, une temporisation point (aussi appelée une "temporisation persistante" dans d'autres documents) DEVRAIT être prise comme l'indication d'un événement de perturbation significative dans le réseau, comme une défaillance de routeur ou un changement d'acheminement sur un chemin avec une plus petite MTU. Pour TCP, ceci se produit quand expire le seuil de temporisateur R1 décrit par la [RFC1122] .

Si il y a une temporisation persistante et qu'il n'y a pas de message ICMP indiquant une raison (PTB, réseau inaccessible, etc., ou si le message ICMP a été ignoré pour une raison quelconque) la première action RECOMMANDÉE de récupération est de traiter cela comme un trou noir ICMP détecté comme défini dans la [RFC2923].

La réponse à un trou noir détecté dépend des valeurs courantes de search_low et eff_pmtu. Si eff_pmtu est supérieur à

search_low, on règle eff_pmtu à search_low. Autrement, on règle eff_pmtu et search_low à la valeur initiale de search_low. Sur des fins de temporisation successives supplémentaires, search_low et eff_pmtu DEVRAIENT être diminuées par deux, avec une limite inférieure de 68 octets pour IPv4 et 1280 octets pour IPv6. Des limites encore inférieures PEUVENT être permises pour prendre en charge une opération limitée sur des liaisons avec des MTU qui sont inférieures à ce qui est permis par les spécifications IP.

7.8 Vérification de la MTU

Il est possible à un flux de traverser simultanément plusieurs chemins, mais une mise en œuvre va seulement être capable de garder une seule représentation de chemin pour le flux. Si les chemins ont des MTU différentes, mémoriser la MTU minimum de tous les chemins dans la représentation de chemin du flux va résulter en un comportement correct. Si des messages PTB ICMP sont livrés, alors la PMTUD classique va fonctionner correctement dans cette situation.

Si la livraison ICMP échoue, cassant la PMTUD classique, la connexion va seulement s'appuyer sur la PLPMTUD. Dans ce cas, la PLPMTUD peut échouer aussi car elle suppose qu'un flux traverse un chemin avec une seul MTU. Une sonde de taille supérieure au minimum mais inférieure au maximum des MTU de chemin peut réussir. Cependant, en relevant la PMTU effective du flux, le taux de pertes va augmenter significativement. Le flux peut encore faire des progrès, mais le taux de pertes résultant va probablement être inacceptable. Par exemple, quand on utilise un round-robin de marquage en bandes bidirectionnel, 50 % des paquets de pleine taille vont être éliminés.

Le marquage en bande de cette manière est souvent indésirable pour le fonctionnement pour d'autres raisons (par exemple, à cause de la réorganisation des paquets) et est généralement évité en hachant chaque flux sur un seul chemin. Cependant, pour accroître la robustesse, une mise en œuvre DEVRAIT appliquer certaines formes de vérification de la MTU, comme celle que si l'augmentation de eff_pmtu résulte en une forte augmentation du taux de pertes, elle revient à l'utilisation d'une MTU inférieure.

Une stratégie RECOMMANDÉE serait de sauvegarder la valeur de eff_pmtu avant de l'augmenter. Ensuite, si le taux de pertes passe au dessus d'un seuil pendant un certain temps (par exemple, le taux de pertes est supérieur de 10 % sur plusieurs intervalles de temporisations de retransmission) alors la nouvelle MTU est considérée comme incorrecte. La valeur sauvegardée de eff_pmtu DEVRAIT être restaurée, et search_high réduit de la même manière que dans une défaillance de sonde. Les mises en œuvre de PLPMTUD DEVRAIENT appliquer la vérification de MTU.

8. Fragmentation d'hôte

Les couches de mise en paquet DEVRAIENT éviter d'envoyer des messages qui vont exiger la fragmentation [Kent87], [RFC4963]. Cependant, empêcher entièrement la fragmentation n'est pas toujours possible. Certaines couches de mise en paquet, comme une application UDP en dehors du noyau, peut être dans l'incapacité de changer la taille des messages qu'elle envoie, résultant en des tailles de datagrammes qui excèdent la MTU de chemin.

IPv4 permettait à de telles applications d'envoyer des paquets sans le bit DF établi. Les paquets surdimensionnés sans le bit DF établi vont être fragmentés dans le réseau ou par l'hôte envoyeur quand ils rencontrent une liaison avec une MTU plus petite que le paquet. Dans certains cas, les paquets pourraient être fragmentés plus d'une fois si il y a des liaisons en cascade avec des MTU progressivement plus petites. Cette approche N'EST PAS RECOMMANDÉE.

Il est RECOMMANDÉ que les mises en œuvre de IPv4 utilisent une stratégie qui imite la fonctionnalité de IPv6. Quand une application envoie des datagrammes supérieurs à la MTU de chemin effective, ils DEVRAIENT être fragmentés à la MTU de chemin dans la couche IP de l'hôte même si ils sont plus petits que la MTU de la première liaison, directement rattachée à l'hôte. Le bit DF DEVRAIT être établi sur les fragments, afin qu'ils ne soient pas fragmentés à nouveau dans le réseau. Cette technique va minimiser la probabilité que les applications s'appuient sur la fragmentation IPv4 d'une façon qui ne peut pas être mise en œuvre dans IPv6. Au moins un système d'exploitation majeur utilise déjà cette stratégie. La Section 9 décrit des exceptions à cette règle quand l'application envoie des paquets surdimensionnés à des fins de sondage ou de diagnostic.

Comme les protocoles qui ne mettent pas en œuvre la PLPMTUD sont quand même soumis à des problèmes à cause des trous noirs ICMP, il peut être souhaitable de limiter ces protocoles aux MTU "sûres" susceptibles de fonctionner sur tout chemin (par exemple, 1280 octets) et de permettre à tout protocole qui met en œuvre la PLPMTUD de fonctionner sur toute la gamme prise en charge par la couche inférieure.

Noter que la fragmentation IP divise les données en paquets, donc elle est une couche de mise en paquets minimaliste. Cependant, elle n'a pas de mécanisme pour détecter les paquets perdus, de sorte qu'elle ne peut pas prendre en charge une mise en œuvre native de PLPMTUD. La PLPMTUD fondée sur la fragmentation exige l'adjonction d'un protocole, comme décrit au paragraphe 10.3.

9. Sondage d'application

Toutes les mises en œuvre DOIVENT inclure un mécanisme par lequel les applications qui utilisent des protocoles sans connexion peuvent envoyer leurs propres sondes. Ceci est nécessaire pour mettre en œuvre la PLPMTUD dans un protocole d'application comme décrit au paragraphe 10.4 ou pour mettre en œuvre des outils de diagnostic pour des problèmes de débogage avec la PMTUD. Il DOIT y avoir un mécanisme permettant à une application d'envoyer des datagrammes plus grands que eff_pmtu, l'estimation par les systèmes d'exploitation de la MTU de chemin, sans être fragmentés. Si ce sont des paquets IPv4, ils DOIVENT avoir le bit DF établi.

Pour l'instant, la plupart des systèmes d'exploitation prennent en charge deux modes pour l'envoi de datagrammes : un qui fragmente en silence les paquets qui sont trop gros, et un autre qui rejette les paquets qui sont trop gros. Aucun de ces modes ne convient pour la mise en œuvre de la PLPMTUD dans une application ou le diagnostic de problèmes avec la découverte de la MTU de chemin. Un troisième mode est EXIGÉ où le datagramme est envoyé même si il est plus grand que l'estimation actuelle de la MTU de chemin.

La mise en œuvre de la PLPMTUD dans une application exige aussi un mécanisme par lequel elle puisse informer le système d'exploitation du résultat de la sonde, comme décrit au paragraphe 7.6, ou mettre directement à jour search_low, search_high, et eff_pmtu, comme décrit au paragraphe 7.1.

Les applications de diagnostic sont utiles pour trouver les problèmes de PMTUD, comme ceux qui pourraient être causés par un routeur défectueux qui retourne les messages PTB ICMP avec des informations de taille incorrectes. De tels problèmes peuvent être très rapidement localisés avec un outil qui peut envoyer des sondes de toute taille spécifiée, et collecter et afficher tous les messages PTB ICMP retournés.

10. Couches spécifiques de mise en paquet

Tous les protocoles de couche de mise en paquets doivent considérer toutes les questions discutées à la Section 6. Pour de nombreux protocoles, ces questions sont traitées directement. Cette section discute les détails spécifiques de la mise en œuvre de la PLPMTUD avec un couple de protocoles. On espère que les descriptions données ici seront une illustration suffisante pour que les mises en œuvre les adaptent aux autres protocoles.

10.1 Méthode de sondage utilisant TCP

TCP n'a pas de mécanisme pour distinguer les données dans la bande du bourrage. Donc, TCP doit générer des sondes en segmentant les données de façon appropriée. Il y a deux approches de la segmentation : avec chevauchement et sans chevauchement.

Dans la méthode sans chevauchement, les données sont segmentées de telle façon que la sonde et tous les segments suivants ne contiennent pas de données qui se chevauchent. Si la sonde est perdue, le "trou de sonde" va être une taille de sonde complète moins les en-têtes. Les données dans le trou de sonde vont devoir être retransmises avec plusieurs segments plus petits.

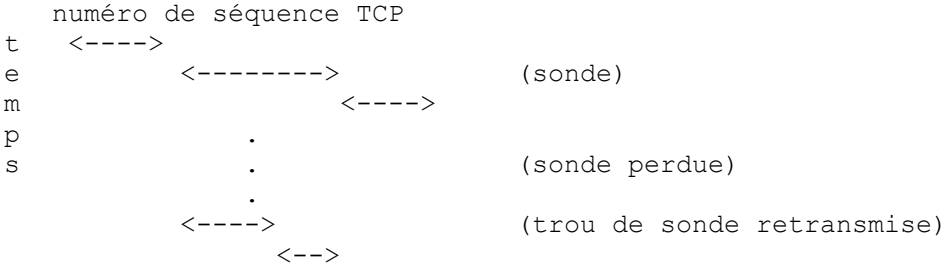

Figure 2

Une autre approche est d'envoyer les données suivantes qui chevauchent la sonde de telle sorte que le trou de sonde soit de longueur égale à la MSS courante. Dans le cas d'une sonde réussie, cela ajoute des frais généraux parce que les données sont envoyées deux fois, mais il faudra retransmettre seulement un segment après une sonde perdue. Quand une sonde réussit, il va probablement y avoir des accusés de réception dupliqués générés à cause des données dupliquées envoyées. Il est important que ces accusés de réception dupliqués ne déclenchent pas de retransmission rapide. À ce titre, une mise en œuvre qui utilise cette approche DEVRAIT limiter la taille de sonde à trois fois la MSS courante (causant au plus deux accusés de réception dupliqués) ou ajuster de façon appropriée son seuil d'accusés de réception dupliqués pour les données immédiatement après une sonde réussie.

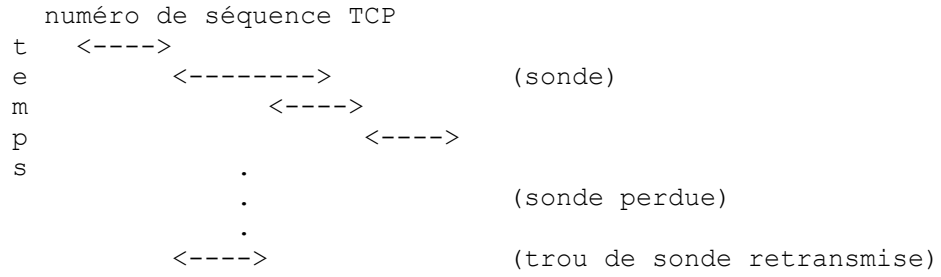

Figure 3

Le choix de la méthode de segmentation à utiliser devrait se fonder sur ce qui est le plus simple et le plus efficace pour une mise en œuvre de TCP donnée.

10.2 Méthode de sondage utilisant SCTP

Dans le protocole de transmission des commandes de flux (SCTP, *Stream Control Transmission Protocol*) [RFC2960], l'application écrit des messages à SCTP, qui divise les données en plus petits "tronçons" convenables pour la transmission à travers le réseau. Chaque tronçon reçoit un numéro de séquence de transmission (TSN, *Transmission Sequence Number*). Une fois qu'un TSN a été transmis, SCTP ne peut pas changer la taille du tronçon. La prise en charge de SCTP multi chemins exige normalement que SCTP choisisse une taille de tronçon telle que ses messages tiennent dans la plus petite PMTU de tous les chemins. Bien que ce ne soit pas exigé, les mises en œuvre peuvent grouper plusieurs tronçons de données pour faire de plus gros paquets IP à envoyer sur des chemins qui ont une plus grande PMTU. Noter que SCTP doit sonder indépendamment la PMTU sur chaque chemin vers l'homologue.

La méthode RECOMMANDÉE pour générer les sondes est d'ajouter un tronçon consistant seulement en bourrage à un message SCTP. Le tronçon PAD défini dans la [RFC4820] DEVRAIT être rattaché à un tronçon HEARTBEAT (HB) de longueur minimum pour construire un paquet sonde. Cette méthode est pleinement compatible avec toutes les mises en œuvre courantes de SCTP.

SCTP PEUT aussi sonder avec une méthode similaire à celle de TCP décrite ci-dessus, en utilisant des données en ligne. Utiliser une telle méthode présente l'avantage que les sondes réussies n'ont pas de frais généraux supplémentaires ; cependant, les sondes défaillantes vont exiger la retransmission des données, ce qui peut impacter les performances du flux.

10.3 Méthode de sondage pour la fragmentation IP

Il y a peu de protocoles et applications qui envoient normalement de grands datagrammes et s'appuient sur la fragmentation IP pour les livrer. On sait depuis longtemps que cela a des conséquences indésirables [Kent87]. Plus récemment, il a été mis en lumière que la fragmentation IPv4 n'est pas suffisamment robuste pour une utilisation généralisée dans l'Internet d'aujourd'hui. Le champ d'identification IP de 16 bits n'est pas assez grand pour empêcher de fréquentes mauvaises associations de fragments IP, et les sommes de contrôle TCP et UDP sont insuffisantes pour empêcher les données corrompues résultantes d'être livrées aux couches de protocole supérieures [RFC4963].

Comme mentionné à la Section 8, les protocoles de datagrammes (comme UDP) peuvent s'appuyer sur la fragmentation IP comme couche de mise en paquets. Cependant, utiliser la fragmentation IP pour mettre en œuvre la PLPMTUD est problématique parce que la couche IP n'a pas de mécanisme pour déterminer si les paquets sont finalement livrés au nœud distant, sans participation directe par l'application.

Pour prendre en charge la fragmentation IP comme couche de mise en paquets sous une application non modifiée, une mise

en œuvre DEVRAIT s'appuyer sur le partage de MTU de chemin décrit au paragraphe 5.2, plus un protocole adjoint pour sonder la MTU de chemin. Un certain nombre de protocoles peuvent être utilisés à cette fin, comme ECHO et ECHO REPLY de ICMP, ou des datagrammes UDP de style "traceroute" qui déclenchent des messages ICMP. L'utilisation de ECHO et ECHO REPLY ICMP va sonder les chemins à la fois vers l'amont et vers l'aval, de sorte que l'envoyer va seulement être capable de tirer parti du minimum des deux. D'autres méthodes qui sondent seulement le chemin vers l'aval sont préférées si disponibles.

Toutes ces approches ont un certain nombre de problèmes potentiels de robustesse. Les défaillances les plus probables sont dues à des pertes sans relation avec la MTU (par exemple, des nœuds qui éliminent certains types de protocole). Ces pertes sans lien avec la MTU peuvent empêcher la PLPMTUD de relever la MTU, forçant la fragmentation IP à utiliser une MTU plus petite que nécessaire. Comme ces défaillances ne vont probablement pas causer de problèmes d'interopérabilité, elles sont relativement bénignes.

Cependant, d'autres modes de défaillance plus sérieux existent bien, comme ceux qui pourraient être causés par des boîtiers de médiation ou des routeurs de couche supérieure qui choisissent des chemins différents pour des types de protocoles ou sessions différents. Dans de tels environnements, des protocoles adjoints peuvent légitimement rencontrer une MTU de chemin différente de celle du protocole principal. Si le protocole adjoint trouve une MTU supérieure à celle du protocole principal, la PLPMTUD peut choisir une MTU qui n'est pas utilisable par le protocole principal. Bien que ce soit un problème potentiellement sérieux, cette sorte de situation sera probablement vue comme incorrecte par un grand nombre d'observateurs, et il y aura donc un fort mouvement pour la corriger.

Comme les protocoles sans connexion pourraient ne pas garder assez d'état pour diagnostiquer effectivement les trous noirs de MTU, il serait plus robuste de pencher du côté de l'utilisation d'une MTU initiale trop petite (par exemple, 1 k octets ou moins) avant de sonder un chemin pour en mesurer la MTU. Pour cette raison, les mises en œuvre qui utilisent la fragmentation IP DEVRAIENT utiliser une eff_pmtu initiale, choisie comme décrit au paragraphe 7.2, sauf à utiliser un contrôle global distinct pour la eff_mtu initiale par défaut pour les protocoles sans connexion.

Les protocoles sans connexion introduisent aussi aussi un problème supplémentaire de conservation de l'antémémoire d'informations de chemin : il n'y a pas d'événement correspondant à l'établissement et la suppression de connexion à utiliser pour gérer l'antémémoire elle-même. Une approche naturelle serait de garder une entrée d'antémémoire immuable pour le "chemin par défaut", qui aurait une eff_pmtu fixée à la valeur initiale pour les protocoles sans connexion. Le protocole adjoint de découverte de MTU de chemin serait invoqué une fois que le nombre de datagrammes fragmentés pour une destination particulière atteint un seuil configurable (par exemple, 5 datagrammes). Une nouvelle entrée d'antémémoire de chemins serait créée quand le protocole adjoint met à jour eff_pmtu, et serait supprimée sur la base d'un temporisateur ou d'un algorithme de remplacement d'antémémoire "le plus ancien utilisé".

10.4 Méthode de sondage utilisant des applications

Les inconvénients de s'appuyer sur la fragmentation IP et un protocole adjoint pour effectuer la découverte de la MTU de chemin peuvent être surmontés par la mise en œuvre de la découverte de la MTU de chemin au sein de l'application elle-même, en utilisant le propre protocole de l'application. L'application doit avoir une méthode convenable pour générer les sondes et avoir un mécanisme précis et bien rythmé pour déterminer si les sondes sont perdues.

Idéalement, le protocole d'application comporte une légère fonction d'écho qui confirme la livraison du message, plus un mécanisme pour le bourrage des messages à la taille de sonde désirée, de façon qu'il n'est pas fait écho du bourrage. Cette combinaison (autrement dit le HB SCTP plus le PAD) est RECOMMANDÉE parce que une application peut mesurer séparément la MTU de chaque direction sur un chemin avec des MTU asymétriques.

Pour les protocoles qui ne peuvent pas mettre en œuvre la PLPMTUD avec "écho plus bourrage", il y a souvent des méthodes de remplacement pour générer les sondes. Par exemple, le protocole peut avoir un écho de longueur variable qui mesure effectivement la MTU minimum des deux chemins d'aller et de retour, ou il peut y avoir un moyen d'ajouter le bourrage aux messages réguliers qui portent des données réelles d'application. Il peut aussi y avoir d'autres moyens pour segmenter les données d'application pour générer des sondes, ou en dernier recours, il peut être faisable d'étendre le protocole avec de nouveaux types de message spécifiques pour la prise en charge de la découverte de la MTU.

Noter que si il est nécessaire d'ajouter de nouveaux types de message pour la prise en charge de la PLPMTUD, l'approche la plus générale est d'ajouter les messages ECHO et PAD, ce qui permet la plus grande latitude possible quant à la façon dont une mise en œuvre spécifique de l'application de PLPMTUD interagit avec les autres applications et protocoles sur le même système d'extrémité.

Toutes les techniques de sondage d'application exigent la capacité d'envoyer des messages plus grands que la eff_pmtu courante décrite à la Section 9.

11. Considérations sur la sécurité

Dans toutes les conditions, les procédures de PLPMTUD décrites dans le présent document sont au moins aussi sûres que les procédures courantes de découverte de la MTU de chemin standard décrites dans les RFC 1191 et RFC 1981.

Comme la PLPMTUD est conçue pour un fonctionnement robuste sans aucun message ICMP ou autre provenant du réseau, elle peut être configurée à ignorer tous les messages ICMP, soit globalement, soit par application. Dans une telle configuration, elle ne peut pas être attaquée sauf si l'attaquant peut identifier les paquets sonde et causer leur perte. Attaquer la PLPMTUD réduit les performances, mais pas autant que d'attaquer le contrôle d'encombrement en causant la perte de paquets arbitraires. Un tel attaquant pourrait faire beaucoup plus de dommages en perturbant complètement des protocoles spécifiques, comme le DNS.

Comme les protocoles de mise en paquets peuvent partager l'état avec chaque autre, si un protocole de mise en paquets (en particulier une application) était hostile aux autres protocoles sur le même hôte, il pourrait dégrader les performances dans les autres protocoles en réduisant la MTU effective. Si un protocole de mise en paquets n'est pas de confiance, il ne devrait pas être autorisé à écrire sur l'état partagé.

10. Références

10.1 Références normatives

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

[RFC0793] J. Postel (éd.), "Protocole de commande de transmission – Spécification du protocole du programme Internet DARPA", STD 7, septembre 1981.

[RFC1191] J. Mogul et S. Deering, "Découverte de la MTU de chemin", novembre 1990.

[RFC1981] J. McCann, S. Deering, J. Mogul, "Découverte de la MTU de chemin pour IP version 6", août 1996. *(D.S. ; Remplacé par* [RFC8201], STD87)

[RFC2119] S. Bradner, "Mots clés à utiliser dans les RFC pour indiquer les niveaux d'exigence", BCP 14, mars 1997. *(MàJ par* RFC8174)

[RFC2460] S. Deering et R. Hinden, "Spécification du protocole Internet, version 6 (IPv6)", décembre 1998. *(MàJ par* 5095, 6564 *; D.S ; Remplacée par* RFC8200, STD 86*)*

[RFC2960] R. Stewart et autres, "Protocole de transmission de commandes de flux", octobre 2000. (*Obsolète, voir* RFC4960) *(P.S.)*

[RFC3697] J. Rajahalme et autres, "Spécification d'étiquette de flux IPv6", mars 2004. (*Obsolète, voir* RFC6437) *(P.S.)*

[RFC4820] M. Tuexen et autres, "Tronçon de bourrage et paramètres pour le protocole de transmission de contrôle de flux (SCTP)", mars 2007. *(P.S.)*

12.2 Références pour information

[Kent87] Kent, C. and J. Mogul, "Fragmentation considered harmful", Proc. SIGCOMM '87 vol. 17, No. 5, octobre 1987.

[RFC1122] R. Braden, "Exigences pour les hôtes Internet – couches de communication", STD 3, octobre 1989. *(MàJ par* RFC6633, 8029)

[RFC2401] S. Kent et R. Atkinson, "Architecture de sécurité pour le protocole Internet", novembre 1998. (*Obsolète, voir* RFC4301)

[RFC2461] T. Narten, E. Nordmark, W. Simpson, "Découverte de voisins pour IP version 6 (IPv6)", décembre 1998. (*Obsolète, voir* RFC4861) *(D.S.)*

[RFC2760] M. Allman et autres, "Recherches en cours sur TCP concernant les satellites", février 2000. *(Information)*

[RFC2914] S. Floyd, "Principes du contrôle d'encombrement", BCP 41, septembre 2000.

[RFC2923] K. Lahey, "Problèmes de TCP avec la découverte de MTU de chemin", septembre 2000. *(Information)*

[RFC3517] E. Blanton et autres, "Algorithme de récupération de perte fondé sur l'accusé de réception sélectif prudent (SACK) pour TCP", avril 2003. (*Remplacée par* RFC6675) *(P.S.)*

[RFC4340] E. Kohler et autres, "Protocole de contrôle d'encombrement de datagrammes (DCCP)", mars 2006. *(P.S.) (MàJ par* 6773)

[RFC4963] J. Heffner et autres, "Erreur de réassemblage IPv4 à hauts débits de données", juillet 2007. *(Information)*

[tcp-friendly] Mahdavi, J. et S. Floyd, "TCP-Friendly Unicast Rate-Based Flow Control", Note technique envoyée à la liste de diffusion end2end-interest, janvier 1997, <http://www.psc.edu/networking/papers/tcp_friendly.html>.

Appendice A. Remerciements

De nombreuses idées et même du texte viennent directlement des RFC 1191 et RFC 1981.

De nombreuses personnes ont fait des contributions significatives à ce document, incluant : Randall Stewart pour les texte sur SCTP, Michael Richardson pour du matériel provenant de projet Internet antérieurs sur les tunnels qui ignorent DF, Stanislav Shalunov pour l'idée que la pure PLPMTUD est parallèle au contrôle d'encombrement, et Matt Zekauskas pour maintenir le cap durant les réunions. Merci à ceux qui ont effectué les premières mises en œuvre : Kevin Lahey, John Heffner, et Rao Shoaib, qui ont fourni de retours concrets sur les faiblesses des versions antérieures. Merci aussi à toutes les personnes qui ont fait des commentaires constructifs dans les réunions du groupe de travail et sur sa liste de diffusion. Il est sûr que nous avons oublié de nombreuses personnes méritantes.

Matt Mathis et John Heffner ont soutenu ce travail par un don de Cisco Systems, Inc.

Adresse des auteurs

Matt Mathis
Pittsburgh Supercomputing Center
4400 Fifth Avenue
Pittsburgh, PA 15213
USA
téléphone: 412-268-3319
mél : mathis@psc.edu

John W. Heffner
Pittsburgh Supercomputing Center
4400 Fifth Avenue
Pittsburgh, PA 15213
USA
téléphone : 412-268-2329
mél : jheffner@psc.edu

Déclaration complète de droits de reproduction

Remerciement
Le financement de la fonction d'édition des RFC est actuellement fourni par la Internet Society.