Groupe de travail sur les réseaux
Request for Comments : 2453
Remplace : 1723, 1388
STD : 56
Catégorie : Norme homologuée

G. Malkin
Bay Networks

Novembre 1998

RIP version 2

Statut de ce document

Ce document spécifie un protocole Internet standard à destination de la communauté Internet, et invite à toute discussion ou suggestion visant à son d'amélioration. Référez-vous s.v.p. à l'édition actuelle du « Internet Official Protocol Standards » (STD 1) pour connaître l'état de standardisation et le statut de ce protocole. La distribution de ce document est libre.
Cette traduction française a été effectuée par Frédéric Delanoy.
Vous pouvez le contacter ici.

Droits d'auteur

Résumé

Ce document spécifie une extension du Protocole d'Information de Routage (RIP), comme défini dans [1], qui a pour but d'étendre la quantité d'information utile transportée dans les messages RIP et d'ajouter des mesures de sécurité.

Un document associé définira les objets SNMP MIB pour RIP-2 [2]. Un document supplémentaire définira des améliorations de sécurité cryptographiques pour RIP-2 [3].

Remerciements

Je voudrais remercier le groupe de travail RIP de l'IETF pour leur aide à l'amélioration du protocole RIP-2. La plupart du texte relatif aux discussions sur les protocoles à vecteurs de distance et certaines des descriptions du fonctionnement de RIP ont été empruntés au « Protocole d'Information de Routage » de C. Hedrick [1]. Une partie de l'édition finale de ce document a été réalisée par Scott Bradner.

Table des matières

1 Justification

Avec l'avènement de OSPF et IS-IS, certaines personnes pensent que RIP est obsolète. Bien que les protocoles de routage plus récents sont largement supérieurs à RIP, celui-ci présente tout de même certains avantages. Premièrement, dans un petit réseau, RIP n'engendre qu'une très petite surcharge en termes de bande passante utilisée et de temps de configuration et de gestion. RIP est également très facile à implémenter, en particulier par rapport aux IGPs plus récents.

De plus, il y a beaucoup, beaucoup plus d'implémentations de RIP dans la nature que OSPF et IS-IS combinés. Cela va probablement rester le cas pour encore quelques années.

Étant donné que RIP sera utile dans de nombreux environnements pendant encore quelque temps, il est raisonnable d'accroître son utilité. C'est particulièrement vrai car le gain est beaucoup plus grand que le coût de changement.

2 RIP actuel

Le message RIP-1 actuel contient le minimum d'information nécessaire aux routeurs pour acheminer des messages dans un réseau. Il dispose également d'une grande quantité d'espace inutilisé, qu'il doit à ses origines.

Le protocole RIP-1 actuel ne prend pas en compte les systèmes autonomes et les interactions IGP/EGP, le découpage en sous-réseaux [11], et l'authentification puisque ces implémentations postdatent RIP-1. L'absence de prise en charge des masques de sous-réseaux est un problème particulièrement important pour les routeurs car ils ont besoin d'un masque de sous-réseau pour savoir comment déterminer une route. Si une route RIP-1 est une route de réseau (tous les bits non-réseau sont à 0), le masque de sous-réseau est égal au masque de réseau. Néanmoins, si certains des bits non-réseau sont positionnés, le routeur ne peut déterminer le masque de sous-réseau. Pire encore, le routeur ne peut déterminer si la route RIP-1 est une route de sous-réseau ou une route d'hôte. Actuellement, certains routeurs choisissent simplement le masque de sous-réseau de l'interface depuis laquelle ils ont appris la route et déterminent le type de route à partir de cela.

3 Protocole de base

3.1 Introduction

RIP est un protocole de routage basé sur l'algorithme de Bellman-Ford (ou à vecteurs de distance). Cet algorithme a été utilisé pour des calculs de routage dans les réseaux informatiques depuis les débuts d'ARPANET. Les formats de paquets particuliers et le protocole décrits ici sont basés sur le programme « `routed` », qui est inclus dans la distribution Unix de Berkeley. Dans un réseau international comme Internet, il est très improbable qu'un unique protocole de routage soit utilisé dans l'entièreté du réseau. Le réseau sera plutôt organisé comme une collection de « systèmes autonomes »[1], chacun d'entre eux étant en général administré par une seule entité. Chaque AS aura sa propre technologie de routage, qui peut

[1] Autonomous Systems, AS

être différente pour des systèmes autonomes distincts. Le protocole de routage utilisé à l'intérieur d'un système autonome est référencé en tant que protocole interne à des passerelles, ou « IGP ». Un protocole séparé, appelé « EGP » (protocole externe à des passerelles [2], est utilisé pour transférer des informations de routage entre les différents systèmes autonomes. RIP a été conçu pour fonctionner en tant qu'IGP dans des systèmes autonomes de taille modérée. Pour obtenir des informations sur les situations où RIP est supposé convenir, voyez Braden et Postel [6].

RIP utilise un algorithme d'une classe d'algorithmes connue sous le nom d'« algorithmes à vecteurs de distance ». La première description de cette classe d'algorithmes connue par l'auteur est présentée dans Ford et Fulkerson [8]. De ce fait, ils sont parfois connus sous le nom d'algorithmes de Ford-Fulkerson. Le terme Bellman-Ford est également utilisé, et provient du fait que la formulation est basée sur l'équation de Bellman [4]. La présentation qui en est faite dans ce document est basée étroitement sur [5]. Ce document contient la spécification d'un protocole. Pour une introduction aux mathématiques des algorithmes de routage, voyez [1]. Les algorithmes de base décrits dans ce protocole ont déjà été utilisés dans le routage informatique depuis 1969 dans ARPANET. Néanmoins, l'ancêtre spécifique de ce protocole se trouve dans les protocoles réseau de Xerox. Les protocoles PUP [7] utilisaient le Gateway Information Protocol [3] pour échanger de l'information de routage. Une version quelque peu mise à jour de ce protocole a été adoptée pour l'architecture de systèmes de réseau Xerox (Xerox Network Systems, XNS), sous le nom de « Routing Information Protocol » [9]. Le `routed` de Berkeley est en grande partie identique au Routing Information Protocol, les adresses XNS étant remplacées par un format d'adresse plus général capable d'utiliser IPv4 et d'autres types d'adresses, et où les mises à jour de routage sont limitées à une toutes les 30 secondes. Du fait de cette similitude, le terme « Routing Information Protocol »(ou simplement RIP) est utilisé pour se référer à la fois au protocole XNS et au protocole utilisé par `routed`.

RIP est destiné à être utilisé à l'intérieur de l'Internet basé sur IP. Internet est organisé en un grand nombre de réseaux connectés par des passerelles à visée spéciale connues sous le nom de « routeurs ». Les réseaux peuvent être soit des liaisons point-à-point, soit des réseaux plus complexes comme Ethernet ou un réseau à jeton (*token ring*). Les hôtes et les routeurs se voient remettre des datagrammes IP adressés à un hôte quelconque. Le routage est la méthode par laquelle l'hôte ou le routeur décide de l'endroit où envoyer le datagramme. Il peut être possible d'envoyer le datagramme directement à la destination, si cette destination est située sur l'un des réseaux directement connectés à l'hôte ou au routeur. Néanmoins, le cas intéressant se produit quand la destination n'est pas directement accessible. Dans ce cas, l'hôte ou le routeur essaie d'envoyer le datagramme à un routeur qui est plus proche de la destination. Le but d'un protocole de routage est très simple : fournir l'information nécessaire pour effectuer un routage.

3.2 Limitations du protocole

Ce protocole ne résout pas tous les problèmes de routage imaginables. Comme mentionné plus haut, il est principalement destiné à une utilisation en tant qu'IGP, dans des réseaux de taille modérée. De plus, les limitations spécifiques suivantes devraient être mentionnées :

[2] Exterior Gateway Protocol
[3] Protocole d'information sur les passerelles

– Le protocole est limité aux réseaux dont le plus long chemin (le diamètre du réseau) implique 15 sauts [4]. Les concepteurs croient que la conception du protocole de base n'est pas appropriée pour les réseaux plus larges. Notez que cette affirmation suppose qu'un coût unitaire est associé à chaque réseau. C'est la façon dont RIP est normalement configuré. Si l'administrateur système choisit d'utiliser des coûts plus élevés, la limite supérieure de 15 peut facilement devenir un problème.
– Le protocole dépend du « comptage à l'infini » pour résoudre certaines situations inhabituelles. (Cela sera expliqué dans la section 3.4) Si le système de réseaux comporte plusieurs centaines de réseaux, et qu'une boucle de routage les impliquant tous survient, la résolution de la boucle requerrait soit beaucoup de temps (si la fréquence des mises à jour de routage est limitée), soit beaucoup de bande passante (si les mises à jour sont envoyées à chaque fois que des changements sont détectés). Une telle boucle consommerait une grande quantité de bande passante du réseau avant que la boucle ne soit corrigée. Nous croyons que, dans les cas réalistes, cela ne sera pas un problème sauf pour les lignes lentes (à bas débit). Même dans ce cas, le problème sera plutôt inhabituel, puisque différentes précautions sont prises, qui devraient éviter ces problèmes dans la plupart des cas.
– Ce protocole utilise des « métriques » fixes pour comparer des routes alternatives. Cela n'est pas approprié pour les situations où les routes doivent être choisies en fonction de paramètres temps-réel comme un délai, une fiabilité ou une charge mesurés. Les extensions évidentes permettant des métriques de ce type vont probablement introduire des instabilités que le protocole n'est pas censé pouvoir traiter.

3.3 Organisation de ce document

Le corps principal de ce document est organisé en deux parties, qui occupent les deux prochaines sections :

– Un développement conceptuel et une justifications des algorithmes à vecteurs de distance en général.
– La description réelle du protocole.

Chacune de ces deux sections peut largement mériter un document à elle seule. La section 3.4 essaie de donner une présentation informelle des fondements mathématiques de l'algorithme. Notez que la présentation suit une méthode en « spirale ». Un algorithme initial, assez simple, est décrit. Ensuite, des raffinements y sont ajoutés dans les sections successives. La section 3.5 est la description réelle du protocole . À part en cas de référence spécifique à la section 3.4, il devrait être possible d'implémenter RIP entièrement à partir des spécifications fournies dans la section 3.5.

3.4 Algorithmes à vecteurs de distance

Le routage est la tâche consistant à trouver un chemin d'un émetteur à une destination souhaitée. Dans le « modèle Internet » IP, il se réduit essentiellement à trouver une série de routeurs entre le réseau source et le réseau destination. Aussi longtemps qu'un message reste sur un réseau ou sous-réseau unique, tout problème d'acheminement est résolu par une technologie qui est spécifique au réseau. Par exemple, Ethernet et ARPANET définissent tous

[4] en anglais, « hops »

deux un moyen par lequel tout émetteur peut parler à toute destination spécifiée à l'intérieur de ce propre réseau. Le routage IP entre en jeu essentiellement quand les messages doivent aller d'un émetteur d'un tel réseau vers une destination située sur un réseau différent. Dans ce cas, le message doit traverser un ou plusieurs routeurs connectant les réseaux. Si les réseaux ne sont pas adjacents, le message peut traverser plusieurs réseaux intermédiaires, et les routeurs les connectant. Une fois que le message arrive sur un routeur situé sur le même réseau que la destination, la propre technologie de ce réseau est utilisée pour atteindre la destination. Tout au long de cette section, le terme « réseau » est utilisé de façon générique pour couvrir un seul réseau à diffusion [5] (p.ex. Ethernet), une ligne point-à-point, ou ARPANET. Le point critique est qu'un réseau est traité comme une simple entité par IP. Soit aucune décision de redirection n'est nécessaire (comme pour une ligne point-à-point), soit cet acheminement est effectué d'une manière transparente pour IP, permettant à IP de traiter le réseau entier comme un système unique complètement connecté (comme pour un réseau Ethernet ou ARPANET). Notez que le terme « réseau » est utilisé d'une façon quelque peu différente dans les discussions concernant l'adressage IP. Nous utilisons ici le terme « réseau » pour nous référer aux sous-réseaux dans le cas où un adressage des sous-réseaux est utilisé.

Un certain nombre d'approches différentes pour la découverte de routes entre réseaux sont possibles. Une manière utile de catégoriser ces approches est de se baser sur le type d'information que les routeurs doivent s'échanger afin d'être capables de trouver des routes. Les algorithmes à vecteurs de distance sont basés sur l'échange d'une petite quantité d'information. Chaque entité (routeur ou hôte) qui participe au protocole de routage est supposée conserver de l'information sur toutes les destinations du système. Généralement, l'information concernant toutes les entités connectées au réseau est résumée par une seule entité, qui décrit la route vers toutes les destinations de ce réseau. Ce résumé est possible car, en ce qui concerne IP, le routage à l'intérieur d'un réseau est invisible. Chaque entrée de la base de données de routage inclut le prochain routeur auquel les datagrammes destinés à l'entité doivent être envoyés. De plus, elle inclut une « métrique » mesurant la distance totale menant à l'entité. La distance est un concept quelque peu généralisé, qui peut couvrir le délai d'acheminement des messages vers l'entité, son coût d'émission en $, etc. Les algorithmes à vecteurs de distance tirent leur nom du fait qu'il est possible de calculer des routes optimales quand la seule information échangée est la liste de ces distances. En outre, l'information n'est échangée qu'entre entités adjacentes, c.-à-d. des entités partageant un réseau commun.

Bien que le routage soit la plupart du temps basée sur de l'information concernant les réseaux, il est parfois nécessaire de garder une trace des routes menant aux hôtes individuels. Le protocole RIP ne fait aucune distinction formelle entre réseaux et hôtes. Il décrit simplement l'échange d'information concernant des destinations, qui peuvent être soit des réseaux, soit des hôtes. (Notez néanmoins qu'un implémenteur peut choisir de ne pas supporter les routes menant à des hôtes. Voyez la section 3.7) En fait, les développements mathématiques se conçoivent le plus à propos en termes de routes menant d'un hôte ou routeur à un autre. Quand on considère l'algorithme en termes abstraits, il vaut mieux se représenter une entrée de routage pour un réseau comme une abréviation des entrées de routage pour toutes les entités connectées à ce réseau. Ce type d'abréviation n'a de sens que parce que nous considérons que les réseaux n'ont pas de structure interne visible au niveau IP. Par conséquent, nous affectons généralement la même distance à chaque entité d'un réseau donné.

[5] broadcast

Nous disions plus haut que chaque entité conserve une base de données de routage comprenant une entrée pour chaque destination possible du système. Une implémentation réelle va probablement devoir conserver les informations suivantes sur chaque destination :

adresse	dans les implémentations IP de ces algorithmes, cela sera l'adresse IP de l'hôte ou du réseau.
routeur	le premier routeur sur la route menant à la destination.
interface	le réseau physique qui doit être utilisé pour atteindre le premier routeur.
métrique	un nombre indiquant la distance vers la destination.
temporisateur	la durée écoulée depuis la dernière mise à jour de l'entrée

De plus, divers drapeaux et d'autres informations internes seront probablement inclus. Cette base de données est initialisée par une description des entités qui sont directement connectées au système. Elle est mise à jour en fonction des informations reçues dans les messages provenant des routeurs voisins.

L'information la plus importante échangée entre hôtes et routeurs est véhiculée par les messages de mise à jour. Chaque entité qui participe au processus de routage envoie des messages de mise à jour qui décrivent la base de données de routage comme elle existe actuellement dans cette entité. Il est possible de maintenir des routes optimales pour le système entier en utilisant uniquement les informations obtenues depuis les entités voisines. L'algorithme utilisé pour cela sera décrit dans la section suivante.

Comme nous l'avons mentionné plus haut, le but du routage est de déterminer un chemin pour amener des datagrammes à leur destination finale. Les algorithmes à vecteurs de distance sont basés sur une table, présente dans chaque routeur, fournissant la meilleure route vers chaque destination du système. Bien sûr, pour définir quelle route est la meilleure, nous devons disposer d'un moyen de mesure de sa « bonté ». On la référence sous le nom de « métrique ».

Dans les réseaux simples, il est habituel d'utiliser une métrique qui compte simplement le nombre de routeurs qu'un message doit traverser. Dans des réseaux plus complexes, une métrique est choisie pour représenter le délai total qu'endure le message, son coût d'émission, ou une autre quantité pouvant être minimisée. L'exigence principale est qu'il doit être possible de représenter la métrique comme une somme de « coûts » pour les sauts individuels.

Formellement, s'il est possible de se rendre directement d'une entité i à une entité j (c.-à-d. sans traverser d'autres routeurs intermédiaires), alors un coût $d(i,j)$ est associé au saut entre i et j. Dans le cas normal où toutes les entités d'un réseau donné sont considérées être similaires, $d(i,j)$ est le même pour toutes les destinations d'un réseau donné, et représente le coût d'utilisation de ce réseau. Pour obtenir la métrique d'une route complète, il suffit d'additionner les coûts individuels des sauts composant la route. Dans le cadre de ce document, nous supposons que les coûts sont des entiers positifs.

Soit $D(i,j)$ la métrique de la meilleure route allant de l'entité i à l'entité j. Elle devrait être définie pour chaque paire d'entités. $d(i,j)$ représente les coûts des pas individuels. Formellement, soit $d(i,j)$ le coût du chemin direct allant de l'entité i à l'entité j. Il vaut ∞ si i et j ne sont pas des voisins immédiats. (Notez que $d(i,i)$ égale ∞, c.-à-d. que nous considérons qu'il n'existe pas de connexion directe d'un nœud vers lui-même.) Puisque les coûts s'additionnent, il est facile de montrer que la meilleure métrique doit être décrite par

$$\forall\, i,j \quad : \quad D(i,j) = \begin{cases} 0 & \text{si } i = j \\ \min\limits_{k} \left[d(i,k) + D(k,j) \right] & \text{sinon} \end{cases}$$

et que les meilleures routes débutent en allant de i aux voisins k pour lesquels $d(i,k)+D(k,j)$ possède la valeur minimale. (Cela peut être démontré par induction sur le nombre de pas des routes.) Notez que nous pouvons limiter la deuxième équation aux k qui sont des voisins immédiats de i. Pour les autres, $d(i,k) = \infty$, de sorte que le terme les impliquant ne peut jamais être le minimum.

Il s'avère que l'on peut calculer la métrique par un simple algorithme basé sur ceci : l'entité i contacte ses voisins k pour qu'ils lui envoient leurs estimations des distances vers la destination j. Quand i obtient les estimations de k, il ajoute $d(i,k)$ à chacun des nombres. C'est simplement le coût de traversée du réseau entre i et k. De temps à autre, i compare les valeurs de ses voisins et prend la plus petite.

Une preuve est données dans [2] que cet algorithme convergera vers les estimations correctes de $D(i,j)$ en un temps fini en l'absence de changement de topologie. Les auteurs ne font que peu de suppositions quant à l'ordre dans lequel les entités s'envoient leur information l'une l'autre, ou quand le min est recalculé. En gros, les entités ne peuvent pas simplement arrêter d'envoyer des messages ou de recalculer des métriques, et les réseaux ne peuvent retarder les messages indéfiniment. (Le crash d'une entité de routage est un changement de topologie.) De plus, leur preuve ne fait pas usage d'hypothèse relative aux estimations initiales de $D(i,j)$, à part qu'elles doivent être non négatives. Le fait que ces hypothèses plutôt faibles soient suffisamment bonnes est important. Puisqu'on ne doit pas faire d'hypothèses sur le moment d'envoi des mises à jour, on peut exécuter l'algorithme de façon asynchrone en toute sécurité, c.-à-d. que chaque entité peut envoyer des mises à jour en fonction de sa propre horloge. Les mises à jour peuvent être perdues par le réseau, pour autant qu'elles ne le soient pas toutes. Puisqu'on ne doit pas faire d'hypothèses sur la condition de démarrage, l'algorithme peut gérer les changements. Quand le système change, l'algorithme de routage commence à converger vers un nouvel équilibre, en utilisant l'ancien comme point de départ. Il est important que l'algorithme converge en un temps fini quel que soit le point de départ. Sinon, certains types de changements pourraient mener à un comportement non convergent.

L'exposé de l'algorithme donné plus haut (et la preuve) suppose que chaque entité conserve des copies des estimations provenant de chacun de ses voisins, et calcule de temps à autre un minimum sur tous les voisins. En fait, les implémentations réelles ne font pas nécessairement cela. Elles se rappellent simplement de la meilleure métrique rencontrée jusqu'ici, et l'identité du voisin qui l'a envoyée. Elles remplacent cette information à chaque fois qu'elles rencontrent une meilleure (c.-à-d. plus petite) métrique. Cela leur permet de calculer le minimum de façon incrémentale, sans avoir à stocker les données de tous les voisins.

Il y a une autre différence entre l'algorithme comme décrit dans les textes, et ceux utilisés dans des protocoles réels comme RIP : la description ci-dessus ferait inclure par chaque entité une entrée pour elle-même, en montrant une distance de zéro. En fait, ce n'est généralement pas le cas. Rappelez-vous que toutes les entités présentes sur un réseau sont normalement résumées en une seule entité pour le réseau. Considérez la situation d'un hôte ou d'un routeur G qui est connecté au réseau A. C représente le coût d'utilisation du réseau A (habituellement une métrique de un). (Rappelez-vous que nous supposons que la structure interne d'un réseau n'est pas visible pour IP, et que le coût de déplacement entre deux entités quelconques est par conséquent toujours le même.) En principe, G devrait obtenir un message de chaque autre entité sur le réseau A, en montrant un coût de 0 pour aller de cette entité à elle-même. G calculerait ensuite C + 0 comme la distance la séparant de H. Plutôt que G doive regarder tous ces messages identiques, l'algorithme démarre simplement en créant une entrée pour le

réseau A dans sa table, et en lui affectant une métrique de C. Cette entrée pour le réseau A devrait être perçue comme un résumé des entrées de toutes les entités du réseau A. La seule entité sur A qui ne peut être récapitulée par cette entrée commune est G elle-même, car le coût du voyage entre G et G est 0, et pas C. Mais puisque nous n'avons jamais besoin de ces entrées nulles, nous pouvons nous en passer sans problème en conservant uniquement l'entité pour le réseau A. Notez une autre implication de cette stratégie : puisque les entrées nulles ne sont absolument pas nécessaires, les hôtes ne fonctionnant pas comme routeurs ne doivent pas envoyez de message de mise à jour. À l'évidence, les hôtes qui ne font pas office de routeur (c.-à-d. les hôtes qui ne sont connectés qu'à un seul réseau) ne peuvent disposer d'autre information utile pour contribuer que leur propre entrée $D(i, i) = 0$. Comme ils n'ont qu'une seule interface, il est facile de voir qu'une route vers n'importe quel réseau les traversant ira simplement sur cette interface et en ressortira immédiatement. Par conséquent, le coût d'une telle route sera plus élevé que le meilleur coût d'au moins C. Puisque nous n'avons pas besoin des entrées nulles, les hôtes non-routeurs n'ont aucunement besoin de participer au protocole de routage.

Résumons ce qu'un hôte ou un routeur G fait. Pour chaque destination du système, G conservera une estimation de la métrique courante pour cette destination (c.-à-d. le coût total pour l'atteindre) et l'identité du routeur voisin dont les données ont servi de base au calcul de la métrique. Si la destination est sur un réseau qui est directement connecté à G, alors G utilise simplement une entrée qui montre le coût d'utilisation du réseau, et le fait qu'aucun routeur n'est nécessaire pour atteindre la destination. Il est facile de montrer qu'une fois que le calcul a convergé vers les métriques correctes, le voisin qui est enregistré par cette technique est en fait le premier routeur sur le chemin vers la destination. (S'il y a plusieurs chemins de même coût, il s'agit du premier routeur sur l'un d'entre eux.) La combinaison de la destination, de la métrique et du routeur est typiquement référencé comme une route vers la destination avec cette métrique, en utilisant ce routeur.

La méthode vue jusqu'ici ne permet que de diminuer la métrique, car la métrique existante est conservée jusqu'à ce qu'une autre plus petite apparaisse. Il est possible que l'estimation initiale soit trop basse. Par conséquent, il doit y avoir un moyen d'augmenter la métrique. Il s'avère suffisant d'utiliser la règle suivante : supposez que la route actuelle vers une destination a la métrique D et utilise le routeur G. Si un nouveau jeu d'informations arrive depuis une autre source que G, ne mettez à jour la route que si la nouvelle métrique est meilleure que D. Mais si de nouvelles informations arrivent depuis G elle-même, mettez *toujours* à jour D avec la nouvelle valeur. Il est facile de montrer qu'avec cette règle, le processus de mise à jour incrémentale produit les mêmes routes qu'un calcul se souvenant de la dernière information provenant de tous les voisins et obtient un minimum explicite. (Notez que la discussion suppose à cet instant que la configuration du réseau est statique. Elle ne prend pas en compte la possibilité qu'un système puisse tomber en panne.).

Pour résumer, voici l'algorithme à vecteurs de distance de base comme il a été développé jusqu'à présent. (Notez que ce n'est pas une description du protocole RIP. Il y a encore plusieurs raffinements à ajouter.) La procédure suivante est entreprise par chaque entité qui participe au protocole de routage (ceci doit inclure tous les routeurs du système. Les hôtes qui ne sont pas des routeurs peuvent également participer) :

– conserver une table avec une entrée pour chaque destination possible du système. L'entrée contient la distance D vers la destination, et le premier routeur G sur la route vers

ce réseau. Conceptuellement, il devrait y avoir une entrée pour l'entité elle-même, de métrique 0, mais elle n'est en fait pas incluse.
- Périodiquement, envoyer une mise à jour de routage à chaque voisin. La mise à jour est un groupe de messages contenant toute l'information de la table de routage. Elle contient une entrée pour chaque destination, avec la distance menant à celle-ci.
- Quand une mise à jour de routage arrive depuis un voisin G', ajouter le coût associé au réseau partagé avec G'. (Cela devrait être le réseau par lequel la mise à jour est arrivée.) Appelons la distance résultante D'. Comparer les distances résultantes avec les entrées actuelles de la table de routage. Si la nouvelle distance D' pour N est plus petite que la valeur existante D, adopter la nouvelle route, c.-à-d. modifier l'entrée N de la table pour qu'elle ait une métrique D' et un routeur G'. Si G' est le routeur d'où provenait la route existante, c.-à-d. si G' = G, alors utiliser la nouvelle métrique même si elle est plus grande que l'ancienne.

3.4.1 S'accommoder des changements dans la topologie

La discussion ci-dessus suppose que la topologie du réseau est fixe. En pratique, les routeurs et les lignes tombent souvent en panne et redeviennent actifs. Pour prendre en compte cette possibilité, nous devons modifier légèrement l'algorithme.

La version théorique de l'algorithme impliquait un minimum sur tous les voisins immédiats. Si la topologie change, le jeu de voisins change. Par conséquent, la prochaine fois qu'un calcul sera effectué, le changement sera reflété. Néanmoins, comme mentionné plus haut, les implémentations réelles utilisent une version incrémentale de la minimisation. Seule la meilleure route vers toute destination est conservée. Si le routeur impliqué dans cette route venait à se crasher, ou si la connexion réseau se rompait, le calcul ne refléterait jamais le changement. L'algorithme énoncé jusqu'ici dépend du fait qu'un routeur avertisse ses voisins si ses métriques changent. Si le routeur crashe, il n'a alors aucun moyen de prévenir ses voisins d'un changement.

Afin de traiter les problèmes de ce type, les protocoles à vecteurs de distance doivent prendre certaines dispositions pour invalider des routes. Les détails dépendent du protocole spécifique. Par exemple, dans RIP, chaque routeur qui participe au routage envoie un message de mise à jour à tous ses voisins toutes les 30 secondes. Supposez que la route actuelle pour le réseau N utilise le routeur G. Si nous n'avons pas de nouvelles de G depuis 180 secondes, nous pouvons supposer que soit le routeur a crashé, soit la connexion nous y reliant est devenue indisponible. Ainsi donc, nous marquons la route comme étant invalide. Quand nous entendons un autre voisin qui a une route valide vers N, la route valide remplacera l'invalide. Notez que nous attendons 180 secondes avant d'invalider une route même si nous nous attendons à recevoir des nouvelles de chaque voisin toutes les 30 secondes. Malheureusement, des messages sont occasionnellement perdus par les réseaux. Il n'est donc probablement pas souhaitable d'invalider une route sur base d'un seul message manqué.

Comme nous le verrons ci-dessous, il est utile de disposer d'un moyen d'avertir les voisins qu'il n'y a actuellement pas de route valide vers un réseau donné. RIP, ainsi que plusieurs autres protocoles de cette classe, effectue cela via un message de mise à jour normal, en marquant ce réseau comme étant inaccessible. Une valeur de métrique spécifique est choisie pour indiquer une destination injoignable ; cette valeur de métrique est plus grande que la plus grande métrique valide que l'on s'attend à voir. Dans l'implémentation existante de RIP,

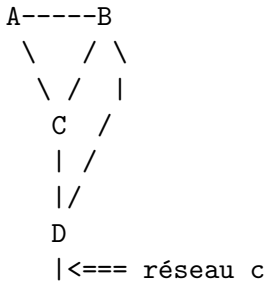
la valeur 16 est utilisée. Cette valeur est habituellement référencée comme l'« infini », car elle est plus grande que la plus grande métrique valide. 16 peut sembler être un nombre étonnamment petit. On l'a choisi petit à ce point pour des raisons que nous verrons sous peu. Dans la plupart des implémentations, la même convention est utilisée en interne pour marquer une route comme étant invalide.

3.4.2 Éviter l'instabilité

L'algorithme présenté jusqu'ici permettra toujours à un hôte ou un routeur de calculer une table de routage correcte. Néanmoins, cela n'est pas encore assez pour le rendre utile en pratique. Les preuves précitées auxquelles on se réfère montrent uniquement que les tables de routage convergeront vers les valeurs correctes en un temps fini. Elles ne garantissent pas que ce temps sera suffisamment court pour être utile, ni ne disent ce qui arrivera aux métriques des réseaux devenus inaccessibles.

Il est assez facile d'étendre les mathématiques pour traiter les routes devenues inaccessibles. La convention suggérée plus haut fera cela. Nous choisissons une grande valeur de métrique pour représenter l'« infini ». Cette valeur doit être suffisamment grande pour qu'aucune métrique réelle ne puisse l'égaler. Pour les besoins de cet exemple, nous utiliserons la valeur 16. Supposons qu'un réseau devienne inaccessible. Tous les routeurs voisins immédiats deviennent obsolètes et fixent la métrique pour ce réseau à 16. Pour les besoins de l'analyse, nous pouvons supposer que tous les routeurs voisins ont obtenu un nouveau matériel les connectant directement au réseau disparu, avec un coût de 16. Puisque c'est la seule connexion au réseau disparu, tous les autres routeurs du système vont converger vers de nouvelles routes qui traversent l'un de ces routeurs. Il est facile de voir qu'une fois que la convergence s'est produite, tous les routeurs auront des métriques d'au moins 16 pour le réseau disparu. Les routeurs situés à un saut des voisins d'origine finiront avec des métriques d'au moins 17 ; ceux situés à deux sauts des voisins d'origine finiront avec des métriques d'au moins 18, etc. Comme ces métriques sont plus grandes que la valeur de métrique maximale, elles sont toutes fixées à 16. Il est évident que le système convergera maintenant vers une métrique de 16 pour le réseau disparu, et ce pour tous les routeurs.

Malheureusement, la question du temps que prendra la convergence n'est pas réductible à une réponse aussi simple. Avant d'aller plus loin, il sera utile de regarder un exemple (emprunté de [2]). Notez, à propos, que ce que nous sommes sur le point de montrer ne se passera pas avec une implémentation correcte de RIP. Nous essayons de montrer pourquoi certaines fonctionnalités sont nécessaires. Notez que les lettres correspondent à des routeurs, et les lignes à des réseaux.



Dans cet exemple, tous les réseaux ont un coût de 1, sauf le lien direct allant de C à D, qui a un coût de 10.

Chaque routeur disposera d'une table montrant une route vers chaque réseau. Néanmoins, à des fins d'illustration, nous ne montrons que les routes menant de chaque routeur au réseau marqué au bout du diagramme.
Voici les chemins d'accès au réseau cible depuis chacun des hôtes/routeurs :

```
D : directement connecté, métrique 1
B : route via D, métrique 2
C : route via B, métrique 3
A : route via B, métrique 3
```

Supposons maintenant que le lien de B à D tombe en panne. Les routes devraient maintenant être ajustées pour utiliser le lien allant de C à D. Malheureusement, cela prendra un moment avant que cela ne se produise. Les changements de routage débutent quand B remarque que la route menant à D n'est plus utilisable. Pour la simplicité, le tableau ci-dessous suppose que tous les routeurs envoient des mises à jour au même moment. Le tableau montre la métrique du réseau cible, comme elle apparaît dans la table de routage de chaque routeur.

Évolution de la métrique du réseau cible au cours du temps													
Hôte	via	coût	via	coût	via	coût	via	coût		via	coût	via	coût
D	dir[a]	1	dir	1	dir	1	dir	1	...	dir	1	dir	1
B	inac[b]		C	4	C	5	C	6		C	11	C	12
C	B	3	A	4	A	5	A	6		A	11	A	12
A	B	3	C	4	C	5	C	6		C	11	C	12

[a] directement connecté
[b] inaccessible

Voici le problème : B est capable de se débarrasser de la route en panne en utilisant un mécanisme de temporisation, mais des vestiges de cette route persistent dans le système pendant une longue période. Initialement, A et C pensent toujours qu'ils peuvent atteindre D via B. Aussi, ils continuent d'envoyer des mises à jour indiquant des métriques de 3. Lors de l'itération suivante, B fera savoir qu'il peut atteindre D via soit A soit C. Bien sûr, il ne le peut pas. Les routes signalées par A et C ne sont maintenant plus envisageables, mais ils n'ont aucun moyen de déjà le savoir. Et même lorsqu'ils découvrent que leurs routes via B se sont volatilisées, ils pensent tous deux qu'il y a une route disponible via l'autre. Finalement, le système converge, comme les mathématiques le soutiennent, mais cela peut prendre un peu de temps avant que cela ne se produise. Le pire cas se présente quand un réseau devient complètement inaccessible depuis une partie du système. Dans ce cas, les métriques vont s'accroître lentement d'une manière semblable à celle vue plus haut qu'elles atteignent finalement l'infini. Pour cette raison, le problème est appelé « comptage à l'infini ».

Vous devriez maintenant voir pourquoi l'« infini » doit être choisi aussi petit que possible. Si un réseau devient complètement inaccessible, il est souhaitable que le comptage à l'infini cesse dès que possible. L'infini doit être suffisamment grand pour qu'aucune route ne soit aussi longue. Mais il ne devrait pas être plus grand que nécessaire. Ainsi, le choix de l'infini est un compromis entre taille du réseau et vitesse de convergence en cas de comptage à l'infini. Les

concepteurs de RIP croyaient que le protocole ne serait probablement pas utilisable dans un réseau d'un diamètre supérieur à 15.

Il y a plusieurs méthodes qui peuvent être employées pour éviter des problèmes comme celui-ci. Celles utilisées par RIP sont appelées « Horizon partagé avec empoisonnement », et « Mises à jour déclenchées ».

3.4.3 Horizon partagé

Notez que certains des problèmes exposés plus haut proviennent du fait que A et C sont engagés dans une partie de tromperie mutuelle. Chacun prétend être capable de rejoindre D via l'autre. Cela peut être évité en faisant un peu plus attention à l'endroit où l'information est envoyée. En particulier, il n'est jamais utile de proclamer l'accessibilité d'un réseau destination au(x) voisin(s) duquel(desquels) on a appris la route. L'« horizon partagé » est un mécanisme destiné à éviter les problèmes causés par l'inclusion de routes dans des mises à jour envoyées à un routeur, alors que ce routeur est lui-même à l'origine de ces informations. Le mécanisme d'« horizon partagé simple » omet les routes apprises depuis un voisin dans les mises à jour envoyées à ce voisin. L'« horizon partagé avec empoisonnement » inclut de telles routes dans les mises à jour, mais fixe leur métrique à l'infini.

Si A pense qu'il peut atteindre D via C, ses messages vers C devraient indiquer que D n'est pas joignable. Si la route vers C est réelle, alors soit C dispose d'une connexion directe vers D, soit d'une connexion passant par un autre routeur quelconque. La route de C peut éventuellement ne pas revenir à A, puisque cela forme une boucle. En indiquant à C que D est inaccessible, A se prémunit simplement contre la possibilité que C puisse être embrouillé et croire qu'il y a une route passant par A. C'est évident pour une ligne point-à-point. Mais considérez le cas où A et C sont connectés par un réseau à diffusion comme Ethernet, et qu'il y a d'autres routeurs sur ce réseau. Si A a une route via C, il devrait indiquer que D est inaccessible quand il parle à un autre routeur de ce réseau. Les autres routeurs du réseau peuvent atteindre C eux-mêmes. Ils n'auront jamais besoin de passer par A pour accéder à C. Si la meilleure route de A passe réellement par C, aucun autre routeur de ce réseau n'a besoin de savoir que A peut atteindre D. C'est heureux, car cela signifie que le même message de mise à jour qui a été utilisé pour C peut être utilisé pour tous les autres routeurs du même réseau. Par conséquent, les messages de mise à jour peuvent être émis par diffusion.

En général, l'horizon partagé avec empoisonnement est plus sûr que l'horizon partagé simple. Si deux routeurs possèdent des routes pointant l'un vers l'autre, l'annonce de routes empoisonnées avec une métrique de 16 cassera la boucle immédiatement. Si les routes empoisonnées ne sont simplement pas annoncées, les routes erronées devront être éliminées par l'attente de l'expiration d'une temporisation. Néanmoins, l'empoisonnement a un désavantage : il accroît la taille des messages de routage. Considérez le cas d'un backbone [6] de campus connectant plusieurs bâtiments différents. Dans chaque bâtiment, il y a un routeur connectant le backbone à un réseau local. Réfléchissez à quelles mises à jour de routage ces routeurs devraient diffuser sur le réseau backbone. Tout ce que le reste du réseau doit réellement savoir sur chaque routeur est l'identité des réseaux locaux qui y sont connectés. En utilisant l'horizon partagé simple, seules ces routes apparaîtront dans les messages de mise à jour envoyés par le routeur au réseau backbone. Si l'horizon partagé avec empoisonnement est utilisé, le routeur

[6] réseau fédérateur

doit mentionner toutes les routes qu'il apprend du backbone, avec une métrique de 16. Si le système est grand, cela peut résulter en un grand message de mise à jour, dont presque toutes les entrées indiquent des réseaux inaccessibles.

Dans un certain sens statique, l'annonce de routes empoisonnées avec une métrique de 16 ne fournit pas d'information supplémentaire. S'il y a beaucoup de routeurs sur un réseau à diffusion, ces entrées supplémentaires peuvent utiliser une bande passante significative. La raison pour laquelle elles sont présentes est d'améliorer le comportement dynamique. Quand la topologie change, mentionner les routes qui ne devraient pas traverser le routeur aussi bien que celles qui le devraient peut accélérer la convergence. Néanmoins, dans certaines situations, les gestionnaires de réseaux peuvent préférer accepter une convergence un peu plus lente afin de minimiser la surcharge due au routage. De ce fait, les implémenteurs peuvent à leur convenance implémenter l'horizon partagé simple plutôt que l'horizon partagé avec empoisonnement, ou peuvent fournir une option de configuration permettant au gestionnaire de réseaux de choisir quel comportement utiliser. Il est également permis d'implémenter des mécanismes hybrides qui annoncent certaines routes empoisonnées avec une métrique de 16 et omettent les autres. Un exemple d'un tel mécanisme serait d'utiliser une métrique de 16 pour les routes empoisonnées pour une certaine période de temps après les changements de routage les impliquant, et après cela les omettre dans les mises à jour.

Le RFC traitant des obligations des routeurs [11] spécifie que toutes les implémentations de RIP doivent utiliser l'horizon partagé et devraient également utiliser l'horizon partagé avec empoisonnement, bien qu'il puisse y avoir un moyen de désactiver l'empoisonnement.

3.4.4 Mises à jour déclenchées

L'horizon partagé avec empoisonnement empêchera toute boucle de routage n'impliquant que deux routeurs. Néanmoins, il est toujours possible d'arriver à des situations où trois routeurs sont engagés dans une partie de tromperie mutuelle. Par exemple, A peut croire qu'il a une route vers B, B vers C, C vers A. L'horizon partagé ne peut arrêter une telle boucle. La boucle ne sera résolue que lorsque la métrique atteindra l'infini, et le réseau impliqué sera ensuite déclaré injoignable. Les mises à jour déclenchées constituent une tentative d'accélérer cette convergence. Pour utiliser des mises à jour déclenchées, nous ajoutons simplement une règle qui dit qu'à chaque fois qu'un routeur change la métrique d'une route, il doit envoyer des messages de mise à jour presque immédiatement, même si ce n'est pas encore le moment d'envoi du message de mise à jour régulier. (Les détails de chronométrage différeront de protocole à protocole. Certains protocoles à vecteurs de distance, RIP compris, spécifient un délai faible, afin d'éviter que des mises à jour déclenchées ne génèrent un trafic réseau excessif.). Notez la façon dont cela se combine avec les règles de calcul de nouvelles métriques. Supposons que la route allant d'un routeur à la destination N emprunte le routeur G. Si une mise à jour provient de G elle-même, le routeur récepteur *doit* croire la nouvelle information, que la nouvelle métrique soit supérieure ou inférieure à l'ancienne. Si le résultat est une modification de la métrique, alors le routeur récepteur enverra des mises à jour déclenchées à tous les hôtes et routeurs qui y sont directement connectés. Ils peuvent alors à leur tour envoyer des mise à jour à leurs voisins. Le résultat est une cascade de mises à jour déclenchées. Il est facile de montrer quels hôtes et routeurs sont impliqués dans la cascade. Supposez qu'un routeur G soutient qu'une route vers la destination N est périmée. G enverra des mises à jour déclenchées à tous ses voisins. Néanmoins, les seuls voisins qui croiront la nouvelle information

sont ceux dont les routes vers N passent par G. Les autres routeurs et hôtes considéreront ceci comme une information sur une nouvelle route moins bonne que celle qu'ils utilisent déjà, et l'ignoreront. Les voisins dont les routes passent par G mettront à jour leurs métriques et enverront des mises à jour déclenchées à tous leurs voisins. À nouveau, seuls les voisins dont les routes les traversent y prêteront attention. Par conséquent, les mises à jour déclenchées se propageront vers l'arrière le long de tous les chemins menant au routeur G, en mettant à jour les métriques vous leur donner la valeur « infini ». Cette propagation s'arrêtera dès qu'elle atteint une partie du réseau dont la route vers la destination N emprunte un autre chemin.

Si le système pouvait rester tranquille lorsque la cascade de mises à jour déclenchées se produit, il serait possible de prouver que le comptage à l'infini ne se produira jamais. Les mauvaises routes seront toujours supprimées immédiatement, et aucune boucle de routage ne pourrait se former.

Malheureusement, la réalité n'est pas aussi idyllique. Pendant que les mises à jour déclenchées sont envoyées, des mises à jour régulières peuvent se produire au même moment. Les routeurs qui n'ont pas encore reçu la mise à jour déclenchée enverront toujours de l'information basée sur la route qui n'existe plus. Il est possible qu'après que la mise à jour déclenchée ait traversé un routeur, il puisse recevoir une mise à jour normale de l'un des routeurs qui n'a pas encore été prévenu. Cela pourrait reconstituer un vestige orphelin de la route défectueuse. Si les mises à jour déclenchées se produisent suffisamment rapidement, c'est très improbable. Néanmoins, le comptage à l'infini est toujours possible.

Le RFC traitant des obligations des routeurs [11] spécifie que toutes les implémentations de RIP doivent implémenter les mises à jour déclenchées pour les routes effacées, et peuvent implémenter les mises à jour déclenchées pour les nouvelles routes ou les changements de route. Les implémentations de RIP doivent également limiter la fréquence à laquelle les mises à jour déclenchées peuvent être transmises. (voyez la section 3.10.1)

3.5 Spécifications du protocole

RIP doit permettre à des hôtes et routeurs d'échanger de l'information pour calculer des routes au travers d'un réseau basé sur IPv4. Tout routeur utilisant RIP est censé disposer d'interfaces vers un ou plusieurs réseaux, ou sinon ce n'est pas vraiment un routeur. Ils sont référencés sous le terme « réseaux directement connectés ». Le protocole se base sur l'accès à certaines informations sur chacun de ces réseaux, la plus importante d'entre elles étant sa métrique. La métrique RIP d'un réseau est un entier compris entre 1 et 15 inclus. Elle est définie d'une manière non spécifiée par ce protocole ; néanmoins, étant donné la limite maximale de longueur de chemin, une valeur de 1 est habituellement utilisée. Les implémentations devraient permettre à l'administrateur système de fixer la métrique de chaque réseau. En plus de la métrique, chaque réseau aura une adresse IPv4 destination et le masque de sous-réseau associé. Ils doivent être spécifiés par l'administrateur système d'une façon non spécifiée par ce protocole.

Tout hôte utilisant RIP est censé disposer d'interfaces vers un ou plusieurs réseaux. Ils sont référencés sous le terme de « réseaux directement connectés ». Le protocole se base sur l'accès à certaines informations sur chacun de ces réseaux, la plus importante d'entre elles étant sa métrique, ou « coût ». La métrique d'un réseau est un entier compris entre 1 et 15 inclus. Elle est définie d'une manière non spécifiée par ce protocole. La plupart des

implémentations existantes utilisent toujours une métrique de 1. De nouvelles implémentations devraient permettre à l'administrateur système de fixer le coût de chaque réseau. En plus du coût, chaque réseau aura un numéro de réseau IPv4 et le masque de sous-réseau associé. Ils doivent être spécifiés par l'administrateur système d'une façon non spécifiée par ce protocole.

Notez que les règles spécifiées dans la section 3.7 supposent qu'il y a un seul masque de sous-réseau s'appliquant à chaque réseau IPv4, et que seuls les masques de sous-réseau des réseaux directement connectés sont connus. Il peut y avoir des systèmes qui utilisent des masques de sous-réseau pour différents sous-réseaux à l'intérieur d'un unique réseau. Il peut également y avoir des exemples où il vaut mieux qu'un système connaisse les masques de sous-réseau des réseaux distants. Néanmoins, si tous les routeurs du réseau n'utilisent pas ces extensions, la distribution des informations de routage comportant plusieurs masques de sous-réseaux doit être limitée afin d'éviter des problèmes d'interopérabilité. Voyez les sections 3.7 et 4.3 pour prendre connaissance des règles gouvernant la « distribution » des sous-réseaux.

Chaque routeur implémentant RIP doit posséder une table de routage. Cette table comprend une entrée pour chaque destination qui est accessible via le système exploitant RIP. Chaque entrée contient au moins les informations suivantes :

- L'adresse IPv4 de la destination.
- Une métrique, qui représente le coût total de transport d'un datagramme de l'hôte à cette destination. Cette métrique est la somme des coûts associés aux réseaux qui seraient traversés pour arriver à la destination.
- L'adresse IPv4 du prochain routeur le long du chemin vers la destination. Si la destination est située sur l'un des réseaux directement connectés, cet élément n'est pas nécessaire.
- Un drapeau pour indiquer que l'information sur la route a changé récemment. Il sera référencé sous le nom de « drapeau de changement de route ».
- Différents temporisateurs associés à la route. Voyez la section 3.8 pour plus de détails à leur sujet.

Les entrées pour les réseaux directement connectés sont définies par le routeur, en utilisant des informations récoltées par des moyens non spécifiés par ce protocole. La métrique d'un réseau directement connecté est définie par le coût de ce réseau. Comme indiqué, 1 est le coût habituel. Dans ce cas, la métrique RIP se réduit à un simple comptage des sauts. Des métriques plus complexes peuvent être utilisées quand il est préférable d'indiquer une priorité de certains réseaux sur d'autres (p.ex. pour faire part de différences de bande passante ou de fiabilité).

Pour supporter les extensions détaillées dans ce document, chaque entrée doit en plus contenir un masque de sous-réseau. Le masque de sous-réseau (ainsi que l'adresse IPv4 de la destination) permet au routeur d'identifier les différents masques de sous-réseau à l'intérieur d'un réseau autant que les masques de sous-réseau de réseaux distants.

Les implémenteurs peuvent également choisir de permettre à l'administrateur système d'entrer des routes additionnelles. Celles-ci seraient plus que probablement des routes vers des hôtes ou réseaux à l'extérieur de la portée du système de routage ; elles sont appelées « routes statiques ». Les entrées pour les destinations autres que celles initiales sont ajoutées et mises à jour par les algorithmes décrits dans les sections suivantes.

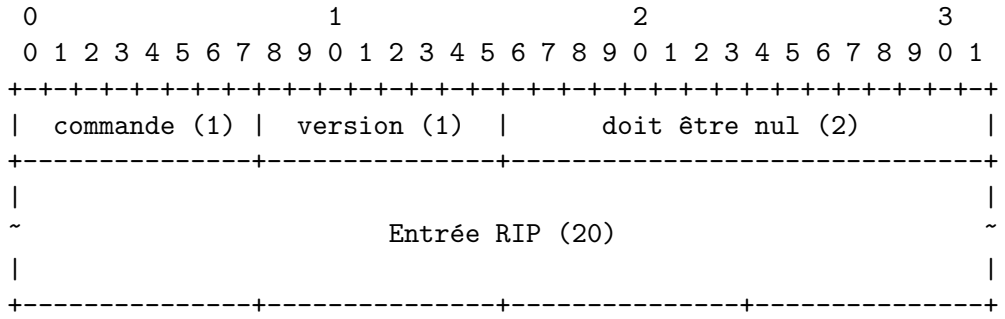
Afin que le protocole puisse fournir une information de routage complète, chaque routeur de l'AS doit participer au protocole. Au cas où de multiples IGPs sont utilisés, il doit y avoir

au moins un routeur qui puisse échanger des informations de routage entre les protocoles.

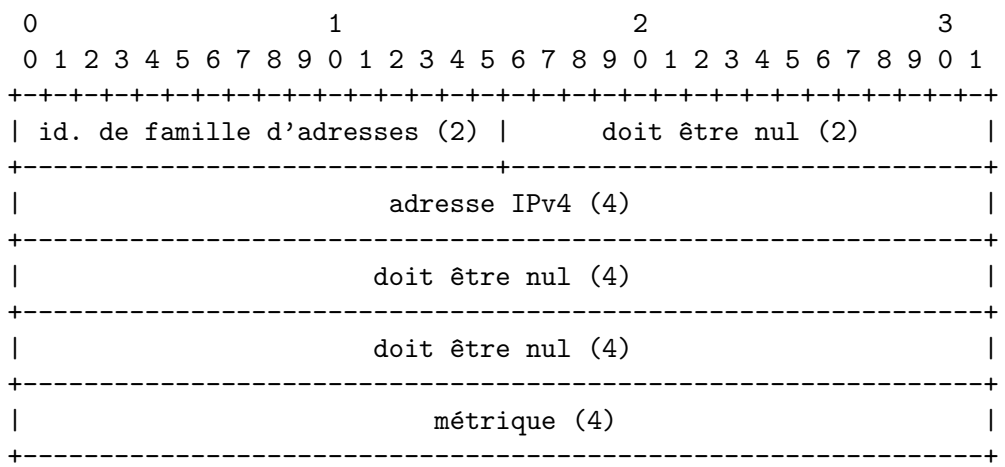
3.6 Formats des messages

RIP est un protocole basé sur UDP. Chaque routeur utilisant RIP dispose d'un processus de routage qui envoie et reçoit des datagrammes sur le n° de port UDP 520, le port RIP-1/RIP-2. Toutes les communications adressées à un processus RIP d'un autre routeur sont envoyées au port RIP. Tous les messages de mise à jour de routage sont envoyés depuis le port RIP. Des messages de mise à jour de routage non sollicités ont pour port source *et* port destination le n° de port RIP. Ceux envoyés en réponse à une requête sont envoyés au port d'où provenait la requête. Des requêtes spécifiques peuvent être envoyées depuis des ports différents du port RIP, mais doivent être dirigées vers le port RIP de la machine cible.

Le format de paquet RIP est :



Il peut y avoir entre 1 et 25 (compris) entrées RIP. Une entrée RIP-1 a le format suivant :



La taille des champs est données en octets. À moins que cela ne soit spécifié différemment, les champs contiennent des entiers binaires, dans l'ordre d'octets réseau, avec l'octet le plus significatif en premier lieu (gros-boutiste). Chaque trait vertical au-dessus de la première ligne représente un bit.

Chaque message contient un en-tête RIP consistant en une commande et un n° de version. Cette section du document décrit la version 1 du protocole ; la section 4 décrit les extensions

de la version 2. Le champ « commande » est utilisé pour spécifier le but de ce message. Les commandes implémentées dans les versions 1 et 2 sont :

	commande	**description**
1	request	Une requête au système répondant indiquant d'envoyer tout ou partie de sa table de routage.
2	response	Un message contenant tout ou partie de la table de routage de l'émetteur. Ce message peut être envoyé en réponse à une requête, ou peut être une mise à jour de routage non sollicitée générée par l'émetteur.

Pour chacun de ces types de messages, dans la version 1, le reste du datagramme contient une liste d'entrées de routage (Route Entries, RTEs). Chaque RTE de cette liste contient un identificateur de famille d'adresses (Address Family Identifier, AFI), une adresse IPv4 destination, et le coût pour rejoindre cette destination (métrique).

L'AFI est le type d'adresse. Pour RIP-1, seul AF_INET (2) est généralement supporté.

Le champ ' métrique ' contient une valeur comprise entre 1 et 15 (inclus), qui spécifie la métrique actuelle pour la destination, ou la valeur 16 (infini), qui indique que la destination est inaccessible.

3.7 Considérations d'adressage

Le routage à vecteurs de distance peut être utilisé pour décrire des routes vers des hôtes individuels ou vers des réseaux. Le protocole RIP permet n'importe laquelle de ces possibilités. Les destinations apparaissant dans les messages *request* et *response* peuvent être des réseaux, des hôtes, ou un code spécial utilisé pour indiquer une adresse par défaut. En général, les types de routes réellement utilisées dépendront de la stratégie de routage utilisée pour le réseau particulier. Beaucoup de réseaux sont configurés de sorte qu'une information de routage pour les hôtes individuels n'est pas nécessaire. Si chaque hôte d'un réseau ou d'un sous-réseau donné est accessible au travers des mêmes routeurs, alors il n'y a aucune raison de mentionner les hôtes individuels dans les tables de routage. Néanmoins, les réseaux qui incluent des lignes point-à-point requièrent parfois que les routeurs gardent une trace des routes vers certains hôtes. La nécessité ou non de cette fonctionnalité dépend de l'adressage et de l'approche du routage utilisés dans le système. Par conséquent, certaines implémentations peuvent choisir de ne pas supporter les routes vers des hôtes. Si les routes vers des hôtes ne sont pas supportées, elles doivent être supprimées quand elles sont reçues dans des messages *response* (voyez la section 3.9.2).

Le format de paquet RIP-1 ne fait pas de distinction entre les différents types d'adresse. Les champs qui sont étiquetés « adresse » peuvent contenir un des éléments suivants :

- adresse d'hôte
- numéro de sous-réseau
- numéro de réseau
- 0 (route par défaut)

Les entités qui utilisent RIP-1 sont supposées utiliser l'information la plus spécifique disponible lors du routage d'un datagramme, c.-à-d. que lors du routage d'un datagramme, son adresse destination doit d'abord être comparée avec la liste des adresses de nœuds. Ensuite, elle doit être examinée pour voir si elle correspond à un numéro de sous-réseau ou de réseau connu. Finalement, si aucun des cas précités ne convient, la route par défaut est utilisée.

Quand un nœud évalue l'information qu'il reçoit via RIP-1, son interprétation d'une adresse dépend de sa connaissance ou non du masque de sous-réseau qui s'applique au réseau. Si c'est le cas, alors il est possible de déterminer la signification de l'adresse. Par exemple, considérons le réseau 128.6. Il a un masque de sous-réseau de 255.255.255.0. Donc, 128.6.0.0 est un numéro de réseau, 128.6.4.0 est un numéro de sous-réseau, et 128.6.4.1 est une adresse de nœud. Néanmoins, si le nœud ne connaît pas le masque de sous-réseau, l'évaluation de l'adresse peut être ambiguë. S'il y a une partie nœud non nulle, il n'y a aucun mécanisme sûr pour déterminer si l'adresse représente un numéro de sous-réseau ou une adresse de nœud. Comme un numéro de sous-réseau serait inutile sans le masque de sous-réseau, les adresses sont supposées représenter des nœuds dans cette situation. Afin d'éviter ce type d'ambiguïté, les nœuds ne doivent pas envoyer de routes de sous-réseaux aux nœuds dont on ne peut présumer qu'ils connaissent le masque de sous-réseau approprié. Normalement, les hôtes ne connaissent les masques de sous-réseau que des réseaux directement connectés. Par conséquent, à moins que des dispositions spéciales n'aient été prises, les routes menant à un sous-réseau ne doivent pas être envoyées à l'extérieur du réseau auquel le sous-réseau fait partie. RIP-2 (voir section 4) élimine l'ambiguïté sous-réseau/hôte en incluant le masque de sous-réseau dans l'entrée de routage.

Ce « filtrage de sous-réseaux » est exécuté par les routeurs à la « frontière » du réseau comportant des sous-réseaux. Ce sont des routeurs qui connectent ce réseau avec d'autres réseaux. À l'intérieur de réseau découpé en sous-réseaux, chaque sous-réseau est traité comme un réseau individuel. Les entrées de routage pour chaque sous-réseau sont passées en revue par RIP. Néanmoins, les routeurs frontière n'envoient aux hôtes des autres réseaux qu'une seule entrée pour le réseau entier. Cela signifie qu'un routeur frontière enverra des informations différentes à des voisins différents. Pour les voisins connectés au réseau composé de sous-réseaux, il génère une liste de tous les sous-réseaux auxquels il est directement connecté, en utilisant le n° de sous-réseau. Pour les voisins connectés à d'autres réseaux, il crée une unique entité pour le réseau entier, en montrant la métrique associée à ce réseau. Cette métrique devrait normalement être la plus petite métrique des sous-réseaux auxquels le routeur est attaché.

De façon similaire, les routeurs frontières ne doivent pas mentionner de route d'hôtes vers des nœuds situés dans l'un des réseaux directement connectés dans les messages envoyés à d'autres réseaux. Ces routes seront synthétisées par l'entrée unique pour le réseau considéré comme un tout.

Le RFC traitant des obligations des routeurs [11] spécifie que toutes les implémentations de RIP devraient supporter les routes d'hôtes et que, si elles ne le font pas, elles doivent alors ignorer toute route d'hôte reçue.

L'adresse spéciale 0.0.0.0 est utilisée pour décrire une route par défaut. Une route par défaut est utilisée quand il n'est pas commode de lister tous les réseaux possibles dans les mises à jour RIP, et quand un ou plusieurs des routeurs proches connectés au système sont préparés à traiter du trafic à destination de réseaux qui ne sont pas listés explicitement. Ces routeurs devraient créer des entrées RIP pour l'adresse 0.0.0.0, tout comme si c'était un réseau auquel ils sont connectés. La mise en œuvre pratique de la création d'entrées 0.0.0.0 par un routeur est laissée aux soins de l'implémenteur. La plupart du temps, l'administrateur système disposera d'un moyen de spécifier quels routeurs devraient créer des entrées pour 0.0.0.0. Néanmoins, d'autres mécanismes sont possibles. Par exemple, un implémenteur pourrait décider que tout routeur parlant BGP devrait être déclaré routeur par défaut. Il peut être utile de permettre

à l'administrateur réseau de choisir la métrique à utiliser pour ces entrées. S'il y a plus d'un routeur par défaut, cela lui permettra d'exprimer une priorité de l'un sur l'autre. Les entrées pour 0.0.0.0 sont traitées par RIP exactement de la même manière qu'un réseau réel ayant cette adresse. Les administrateurs système devraient s'assurer que les routes vers 0.0.0.0 ne se propagent pas plus loin que prévu. Généralement, chaque système autonome a son routeur par défaut préféré. Par conséquent, les routes impliquant 0.0.0.0 ne devraient généralement pas quitter la frontière d'un système autonome. Les mécanismes permettant d'imposer cela ne sont pas spécifiés dans ce document.

3.8 Temporisateurs

Cette section décrit tous les événements déclenchés par des temporisateurs.

Toutes les 30 secondes, le processus RIP est réveillé afin qu'il envoie un message ' response ' non sollicité contenant la table de routage complète (voyez la section 3.9 sur l'horizon partagé) à chaque routeur voisin. Quand il y a beaucoup de routeurs sur un même réseau, ces routeurs ont tendance à se synchroniser entre eux de sorte qu'ils émettent tous des mises à jour au même moment. Cela peut se produire à chaque fois que le temporisateur de 30 secondes est affecté par la charge de travail du système. Il est indésirable que ces messages de mise à jour deviennent synchronisés, car cela peut mener à des collisions inutiles sur les réseaux à diffusion. De ce fait, les implémentations doivent prendre une des deux précautions suivantes :

- Les mise à jour 30-secondes sont déclenchées par une horloge dont le rythme n'est pas affecté par la charge du système ou le temps requis pour s'occuper du temporisateur de mise à jour précédent.
- Le temporisateur 30-secondes est retardé/avancé par l'ajout d'un petit temps aléatoire (+/− 0 à 5 secondes) à chaque fois qu'il est armé. (Les implémenteurs pourraient tenir compte d'une variation plus grande encore à la lumière des récents résultats de recherche [10])

Il y a deux temporisateurs associés à chaque route, une « temporisation » et un « temporisateur de ramassage des déchets »[7]. À l'expiration de la temporisation, la route n'est plus valide. Néanmoins, elle est conservée dans la table pour un court moment, le temps que les voisins soient prévenus que la route à été abandonnée. À l'expiration du temporisateur de ramassage des déchets, la route est finalement supprimée de la table de routage.

La temporisation est initialisée quand une route est établie, et à chaque fois qu'un message de mise à jour est reçu pour la route. Si 180 secondes s'écoulent depuis le dernier moment où la temporisation a été initialisée, la route est considérée avoir dépassé sa période de validité, et le processus de suppression que nous sommes sur le point de décrire est démarré à cet effet.

Les suppressions peuvent se produire pour une des deux raisons suivantes :

1. la temporisation expire
2. la métrique est fixée à 16 du fait de la réception d'une mise à jour depuis le routeur courant

(Voyez la section 3.9.2 pour une discussion sur le traitement des mises à jour provenant d'autres routeurs.) Dans chacun des cas, les événements suivants se produisent :

[7] garbage-collection timer

– Le temporisateur de ramassage des déchets est fixé à 120 secondes.
– La métrique de la route est fixée à 16 (infini). Cela provoque l'abandon de la route.
– Le drapeau de changement de route est défini pour indiquer que cette entrée a été modifiée.
– Le processus de sortie reçoit un signal lui enjoignant de déclencher une réponse.

Jusqu'au moment où le temporisateur de ramassage des déchets expire, la route est incluse dans toutes les mises à jour envoyées par ce routeur. Quand le temporisateur de ramassage des déchets expire, la route est supprimée de la table de routage.

Si une nouvelle route vers ce réseau est établie alors que le temporisateur de ramassage des déchets est en cours de fonctionnement, la nouvelle route remplacera celle qui est sur le point d'être effacée. Dans ce cas, le temporisateur de ramassage des déchets doit être réinitialisé.

Les mises à jour déclenchées utilisent également un petit temporisateur ; néanmoins, ce point sera mieux décrit dans la section 3.9.1.

3.9 Traitement de l'entrée

Cette section décrira le traitement des datagrammes reçus sur le port RIP. Le traitement dépendra de la valeur du champ ' commande '.

Voyez les sections 3.6 et 5.1 pour obtenir des détails concernant le traitement des numéros de version.

3.9.1 Messages request

La commande *request* est utilisée pour demander une réponse contenant tout ou partie de la table de routage d'un routeur. Normalement, les requêtes sont envoyées par diffusion (par transmission multidestinataire [8] pour RIP-2), à partir du port RIP, par des routeurs qui viennent de démarrer et qui cherchent à remplir leur table de routage le plus vite possible. Néanmoins, il peut y avoir des situations (p.ex. le contrôle de routeurs) où la table de routage d'un seul routeur est nécessaire. Dans ce cas, la requête devrait être envoyée depuis un n° de port UDP différent du port RIP. Si une telle requête est reçue, le routeur répond directement à l'adresse et au n° de port du requérant.

La requête est traitée entrée par entrée. S'il n'y a pas d'entrée, aucune réponse n'est fournie. Il y a un cas spécial : s'il y a exactement une entrée dans la requête, avec un identificateur de famille d'adresses de valeur nulle, et une métrique de valeur infinie (c.-à-d. 16), alors il s'agit d'une requête d'envoi de l'entièreté de la table de routage. Dans ce cas, un appel est fait au processus de sortie pour envoyer la table de routage aux adresse/port requis. Mis à part ce cas spécial, le traitement est assez simple. Parcourez la liste des RTEs de la requête une par une. Pour chaque entrée, recherchez la destination dans la base de données de routage du routeur et, s'il y a une route, placez la métrique de cette route dans le champ ' métrique ' de la RTE. S'il n'existe pas de route explicite vers la destination spécifiée, placez l'infini dans le champ ' métrique '. Une fois que toutes les entrées ont été remplies, faites passer la valeur du champ commande de Request à Response et renvoyez le datagramme au requérant.

[8] multicast

Notez qu'il y a une différence dans le traitement de la métrique pour les requêtes spécifiques, et les requêtes sollicitant une table entière. Si la requête demande une table de routage complète, un traitement de sortie normal est effectué. Cela inclut l'horizon partagé (voyez la section 3.4.3 sur l'horizon partagé). Si la requête réclame des entrées spécifiques, elles sont recherchées dans la table de routage et l'information est retournée telle quelle ; aucun traitement d'horizon partagé n'est effectué. La raison d'être de cette distinction est l'espérance d'une utilisation de ces requêtes dans différents contextes. Quand un routeur démarre pour la première fois, il transmet une requête en mode multidestinataire sur chaque réseau connecté en demandant une table de routage complète. On suppose que ces tables de routage complètes vont être employées pour mettre à jour la table de routage du requérant. Pour cette raison, l'horizon partagé doit être utilisé. On suppose de plus qu'une requête pour des réseaux spécifiques n'est faite que par des logiciels de diagnostic, et n'est utilisée pour le routage. Dans ce cas, le requérant voudrait connaître le contenu exact de la base de données de routage, et ne voudrait pas qu'on lui cache ou modifie la moindre information.

3.9.2 Messages response

Une réponse peut être reçue pour plusieurs raisons différentes :
- réponse à une question spécifique
- mise à jour régulière (réponse non sollicitée)
- mise à jour déclenchée provoquée par un changement de route

Le traitement est identique quelle que soit la façon dont la réponse a été générée.

Puisque le traitement d'une réponse peut mettre à jour la table de routage du routeur, la validité de la réponse doit être vérifiée avec soin. La réponse doit être ignorée si elle ne provient pas du port RIP. L'adresse IPv4 source devrait être examinée pour voir si le datagramme provient d'un voisin valide ; la source du datagramme doit se situer sur un réseau directement connecté. Cela vaut également la peine de vérifier si la réponse provient de l'une des adresses propres du routeur. Les interfaces situées sur des réseaux à diffusion peuvent recevoir immédiatement des copies de leurs propres diffusions/transmissions multidestinataires. Si un routeur traite sa propre sortie comme une nouvelle entrée, une certaine confusion en résultera certainement, de sorte que de tels datagrammes doivent être ignorés.

Maintenant que le datagramme a été validé dans son ensemble, traitez ses RTEs une par une. À nouveau, commencez en faisant la validation. Des métriques incorrectes et autres erreurs de format indiquent habituellement des voisins fonctionnant mal, et devraient probablement être portés à l'attention de l'administrateur. Par exemple, si la métrique est plus grande que l'infini, ignorez l'entrée mais enregistrez l'événement. Les tests de validation de base sont :
- l'adresse destination est-elle valide ? (de type unicast [9] ; pas le réseau 0 ni le réseau 127)
- la métrique est-elle valide ? (soit entre 1 et 16 inclus)

Si un des tests échoue, ignorez cette entrée et passez à la suivante. À nouveau, consigner l'erreur est probablement une bonne idée.

Une fois que l'entrée a été validée, mettez à jour la métrique en ajoutant le coût du réseau par lequel le message est arrivé. Si le résultat est supérieur à l'infini, utilisez l'infini, c.-à-d.

$$métrique = \min(métrique + coût, \infty)$$

[9] c.-à-.d l'adresse d'un seul hôte, et pas d'un groupe

Examinez maintenant l'adresse destination pour voir s'il y a déjà une route explicite la rejoignant. Si ce n'est pas le cas, ajoutez cette route à la table de routage, à moins que la métrique soit infinie (ça ne sert à rien d'ajouter une route qui est inutilisable). Ajouter une route à la table de routage consiste à :

- Fixer l'adresse de destination à l'adresse destination de la RTE
- Fixer la métrique à la métrique nouvellement calculée (comme décrit plus haut)
- Fixer l'adresse du saut suivant [10] à l'adresse du routeur à l'origine du datagramme
- Initialiser la temporisation pour la route. Si le temporisateur de ramassage des déchets est en cours de fonctionnement pour cette route, arrêtez-le. (Voyez la section 3.8 pour une discussion sur les temporisateurs.)
- Définir le drapeau de changement de route
- Aviser le processus de sortie de déclencher une mise à jour (voyez la section 3.10.1).

S'il y a une route existante, comparez l'adresse de saut suivant de l'adresse du routeur duquel provenait le datagramme. Si ce datagramme provient du même routeur que la route existante, réinitialisez la temporisation. Ensuite, comparez les métriques. Si le datagramme provient du même routeur que la route existante et que la nouvelle métrique est différente de l'ancienne, ou si la nouvelle métrique est plus petite que l'ancienne, effectuez les actions suivantes :

- Adoptez la route provenant du datagramme (c.-à-d. placez-y la nouvelle métrique, et ajustez l'adresse de saut suivant, si nécessaire).
- Établissez le drapeau de changement de route, et signalez au processus de sortie de déclencher une mise à jour.
- Si la nouvelle métrique est infinie, débutez le processus d'effacement ; sinon, réinitialisez la temporisation.

Si la nouvelle métrique est infinie, le processus d'effacement commence pour la route, qui n'est alors plus utilisée pour router les paquets. Notez que le processus d'effacement n'est entamé que lorsque la métrique est d'abord fixée à l'infini. Si la métrique valait déjà l'infini, alors une nouveau processus de suppression n'est pas démarré.

Si la nouvelle métrique est identique à l'ancienne, le plus simple est de ne rien faire de plus (à part réinitialiser la temporisation, comme spécifié plus haut), mais il existe une heuristique qui pourrait être appliquée. Normalement, il est absurde de remplacer une route si la nouvelle route possède la même métrique que la route existante ; cela provoquerait un va-et-vient de la route, ce qui générerait un nombre intolérable de mises à jour déclenchées. Néanmoins, si la route existante montre des signes de dépassement possible du délai, il peut être préférable de passer immédiatement à une route alternative de même valeur, plutôt que d'attendre que la temporisation se produise. Par conséquent, si la nouvelle métrique est identique à l'ancienne, examinez l'état de la temporisation pour la route existante. Si on est au moins à mi-chemin de l'expiration de sa période de validité, passez à la nouvelle route. Cette heuristique est optionnelle, mais hautement recommandée.

Toute entrée échouant à ces tests est ignorée, car elle n'est pas meilleure que la route actuelle.

[10] next hop

3.10 Traitement de la sortie

Cette section décrit le traitement utilisé pour créer des messages de réponse ('response') contenant tout ou partie de la table de routage. Ce traitement peut être déclenché de l'une des manières suivantes :

- par traitement de l'entrée quand un message 'request' est rencontré (cette réponse est envoyée en mode unicast vers le requérant ; voyez la section 3.9.1).
- par la mise à jour régulière de routage (par diffusion/transmission multidestinataire toutes les 30 secondes).
- par des mises à jour déclenchées (diffusion/transmission multidestinataire à chaque fois qu'une route change).

Quand une réponse est envoyée à tous les voisins (c.-à-d. une mise à jour régulière ou déclenchée), un message 'response' est envoyé au routeur situé à l'autre bout de chaque liaison point-à-point, et est diffusée (transmise en mode multidestinataire pour RIP-2) sur tous les réseaux connectés supportant la diffusion. Ainsi donc, une réponse est préparée pour chaque réseau directement connecté et envoyée à l'adresse appropriée (directe ou diffusion/transmission multidestinataire). Dans la plupart des cas, cela atteint tous les routeurs voisins. Néanmoins, il y a certains cas où cela peut ne pas être assez bien. Cela peut impliquer un réseau qui ne supporte pas la diffusion (p.ex. ARPANET), ou une situation impliquant des routeurs stupides. Dans de telles circonstances, il peut être nécessaire de spécifier une liste réelle de routeurs voisins, et d'envoyer explicitement un datagramme à chacun d'entre eux. Il revient à l'implémenteur de décider si un tel mécanisme est nécessaire, et le cas échéant de définir la façon dont la liste est spécifiée.

3.10.1 Mises à jour déclenchées

Les mises à jour déclenchées requièrent un traitement spécial pour deux raisons. Premièrement, l'expérience montre que les mises à jour déclenchées peuvent provoquer des charges excessives sur des réseaux de capacité limitée ou comportant trop de routeurs. Le protocole requiert donc que les implémenteurs prennent des dispositions pour limiter la fréquence des mises à jour déclenchées. Après l'émission d'une mise à jour déclenchée, un temporisateur devrait être démarré pour un temps aléatoire compris entre 1 et 5 secondes. Si d'autres changements susceptibles de déclencher des mises à jour se produisent avant que le temporisateur n'expire, une simple mise à jour est déclenchée quand le temporisateur expire. Le temporisateur est ensuite fixé à une autre valeur aléatoire comprise entre 1 et 5 secondes. Une mise à jour déclenchée devrait être supprimée si une mise à jour régulière est prévue avant que la mise à jour déclenchée ne soit envoyée.

Deuxièmement, les mises à jour déclenchées n'ont pas besoin d'inclure la table de routage en entier. En principe, seules les routes qui ont été modifiées doivent être incluses. Les messages générés faisant partie d'une mise à jour déclenchée doivent donc au moins inclure les routes dont le drapeau de changement de route est défini. Ils peuvent inclure des routes supplémentaires, à la discrétion de l'implémenteur ; néanmoins, envoyer des mises à jour de routage complètes est fortement déconseillé. Quand une mise à jour déclenchée est traitée, les messages devraient être générés pour chaque réseau directement connecté. Le traitement de l'horizon partagé est effectué lors de la génération de mises à jour déclenchées aussi bien que lors des mises à jour normales (voir la section 3.9). Si, après le traitement de l'horizon

partagé pour un réseau donné, une route modifiée devait apparaître identique sur ce réseau (p.ex. si elle apparaît avec une métrique infinie), la mise à jour route ne doit pas être envoyée. Si aucune route ne doit être envoyée sur ce réseau, la mise à jour peut être omise. Une fois que toutes les mises à jour déclenchées ont été générées, les drapeaux de changement de route devraient être réinitialisés.

Si le traitement de l'entrée est autorisé alors que la sortie est en train d'être générée, un interverrouillage approprié doit être mis en œuvre. Les drapeaux de changement de route ne devraient pas être modifiés à la suite du traitement de l'entrée quand un message de mise à jour déclenchée est en cours de génération.

La seule différence entre une mise à jour déclenchée et les autres messages de mise à jour est la possible omission des routes qui n'ont pas changé. Les mécanismes restants, décrits dans la prochaine section, doivent être appliqués à toutes les mises à jour.

3.10.2 Générer des messages response

Cette section décrit la façon dont un message 'response' est généré pour un réseau directement connecté particulier :

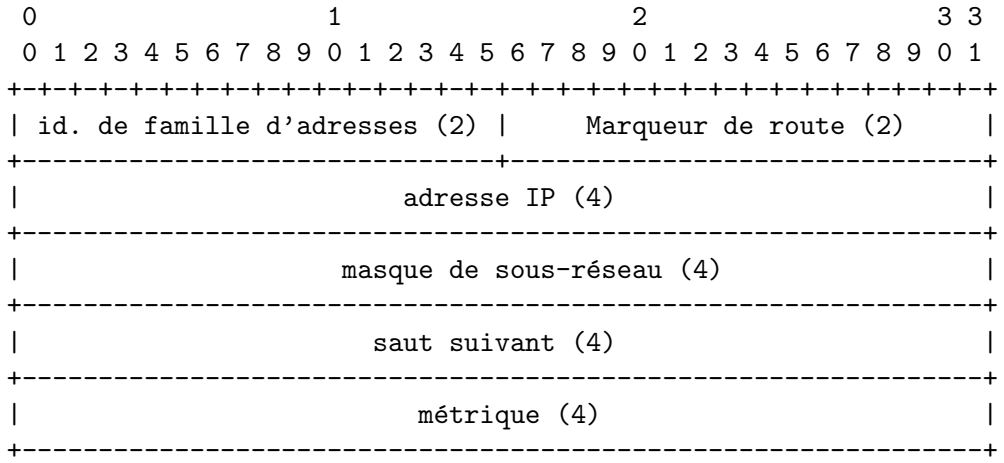
Fixez le n° de version à 1 ou 2. Le dispositif permettant de décider quelle version envoyer est spécifique à l'implémentation ; néanmoins, si c'est la réponse à une requête, la version de la réponse devrait correspondre à celle de la requête. Fixez la commande à 'response'. Fixez les octets marqués « doit être nul » à zéro. Commencez à remplir les RTEs. Rappelez-vous qu'il y a une limite de 25 RTEs par réponse ; s'il y en a plus, envoyez la réponse actuelle et démarrez-en une nouvelle. Il n'y a pas de limite définie relative au nombre de datagrammes qui composent une réponse.

Pour remplir les RTEs, examinez chaque route de la table de routage. Si une mise à jour déclenchée est en cours de génération, seules les entrées dont les drapeaux de changement de route sont spécifiés doivent être incluses. Si, après le traitement de l'horizon partagé, la route ne doit pas être incluse, passez-la. Si la route doit être incluse, alors l'adresse destination et la métrique sont placées dans la RTE. Les routes doivent être incluses dans le datagramme même si leur métrique est infinie.

4 Extensions du protocole

Cette section ne modifie pas le protocole RIP en tant que tel. Il fournit plutôt des extensions du format de message qui permettent aux routeurs de partager des informations supplémentaires importantes.

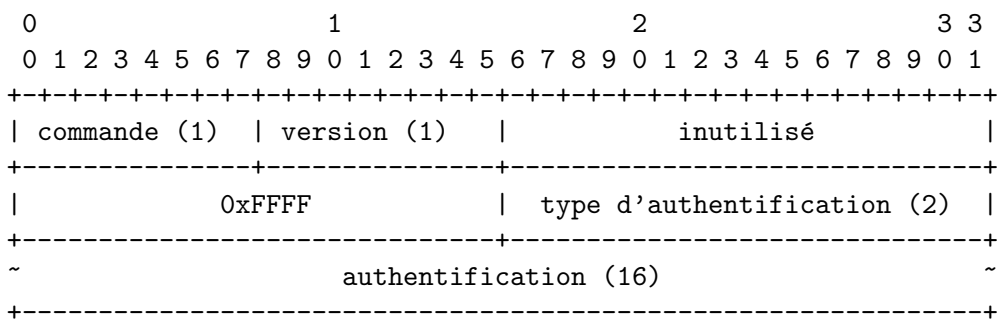
Le même format d'en-tête est utilisé par les messages RIP-1 et RIP-2 (voyez la section 3.6). Le format de l'entrée de routage (RTE) sur 20 octets pour RIP-2 est :



La signification de l'Identificateur de Famille d'Adresses (FAI), de l'adresse IP, et de la métrique est présentée dans la section 3.6. Le champ ' version ' spécifiera le n° de version 2 pour les messages RIP utilisant l'authentification ou transportant des informations dans l'un des champs nouvellement définis.

4.1 Authentification

Puisque l'authentification se fait par message, qu'il n'y a qu'un champ de 2 octets disponible dans l'en-tête du message, et que tout mécanisme d'authentification passable requerra plus de deux octets, le mécanisme d'authentification pour RIP version 2 utilisera l'espace d'une entrée RIP entière. Si l'identificateur de famille d'adresses de la première (et uniquement de la première) entrée du message est 0xFFFF, alors le restant de l'entrée contient l'authentification. Cela signifie qu'il peut y avoir, au plus, 24 entrées RIP dans le reste du message. Si l'authentification n'est pas utilisée, alors aucune entrée comprise dans le message ne devrait avoir un identificateur de famille d'adresses de 0xFFFF. Un message RIP qui contient une entrée d'authentification devrait commencer par le format suivant :



Actuellement, l'unique type d'authentification est un simple mot de passe et est de type 2. Les 16 octets restants contiennent le mot de passe en clair. Si le mot de passe fait moins de 16 octets, il doit être justifié à gauche et complété à droite avec des octets nuls (0x00).

4.2 Marqueur de route

Le champ « marqueur de route » (Route Tag, RT) est un attribut affecté à une route qui doit être préservée et réannoncée avec une route. Le but poursuivi est de fournir une méthode permettant de séparer les routes RIP « internes » (vers des réseaux à l'intérieur du domaine de routage RIP) des routes RIP « externes », qui peuvent avoir été importées depuis un EGP ou un autre IGP.

Les routeurs supportant des protocoles différents de RIP devraient être configurables afin de permettre la configuration du marqueur de route pour les routes importées depuis différentes sources. Par exemple, les routes importées depuis un EGP ou BGP devraient voir leur marqueur de route fixé à une valeur arbitraire, ou au moins au n° du système autonome depuis lequel elles ont été apprises.

D'autres utilisations du marqueur de route sont valides, pour autant que tous les routeurs du domaine RIP l'utilise de façon cohérente. Cela ouvre la voie à un document sur les interactions entre les protocoles BGP et RIP, qui décrirait les méthodes permettant de synchroniser les routage dans un réseau de transit.

4.3 Masque de sous-réseau

Le champ 'masque de sous-réseau' contient le masque de sous-réseau qui est appliqué à l'adresse IP pour produire la partie non hôte de l'adresse. Si ce champ vaut zéro, alors aucun masque de sous-réseau n'a été inclus pour cette entrée.

Sur une interface où un routeur RIP-1 peut recevoir et exploiter les informations d'une entrée de routage RIP-2, les règles suivantes s'appliquent :

1. les informations internes à un réseau ne doivent jamais être annoncées dans un autre réseau
2. une information concernant un sous-réseau plus spécifique ne peut pas être annoncée là où des routeurs RIP-1 pourraient la considérer comme une route d'hôte
3. des routes de sur-réseaux (des routes avec un masque de sous-réseau moins spécifique que le masque de sous-réseau « naturel ») ne peuvent pas être annoncées là où elles pourraient être mal interprétées par des routeurs RIP-1.

4.4 Saut suivant

L'adresse IP du saut suivant immédiat auquel les paquets vers la destination devraient être redirigés. Spécifier une valeur de 0.0.0.0 dans ce champ indique que le routage devrait se faire via le routeur à l'origine de l'annonce RIP. Une adresse spécifiée en tant que saut suivant doit, par la force des choses, être directement accessible au sous-réseau logique sur lequel est effectuée l'annonce.

Le but du champ 'saut suivant' est d'éliminer les paquets routés au travers de sauts supplémentaires dans le système. C'est particulièrement utile quand RIP n'est pas exécuté par tous les routeurs d'un réseau. Un exemple simple est donné dans l'annexe A. Notez que le saut suivant est un champ « consultatif », c.-à-d. que si l'information fournie est ignorée, une route éventuellement sous-optimale, mais néanmoins absolument valide, sera empruntée. Si le prochain saut reçu n'est pas directement accessible, il devrait être traité comme l'est 0.0.0.0.

4.5 Transmission multidestinaire

Afin de réduire une charge non nécessaire pour les hôtes qui n'écoutent pas les messages RIP-2, une adresse IP multidestinataire sera utilisée pour des diffusions périodiques. L'adresse IP multidestinataire est 224.0.0.9. Notez que IGMP n'est pas nécessaire puisque ce sont des messages inter-routeurs qui ne sont pas propagés.

Sur les réseaux NBMA, l'adressage point-à-point classique (*unicast*) peut être utilisé. Néanmoins, si une réponse adressée à l'adresse RIP-2 multidestinataire est reçue, elle devrait être acceptée.

Afin de maintenir une compatibilité descendante, l'utilisation de l'adresse multidestinataire sera configurable, comme décrit dans la section 5.1. Si la transmission multidestinaire est utilisée, elle devrait être utilisé sur toutes les interfaces qui le supportent.

4.6 Requêtes

Si un routeur RIP-2 reçoit une requête RIP-1, il devrait répondre avec une réponse RIP-1. Si le routeur est configuré pour n'envoyer que des messages RIP-2, il ne devrait pas répondre à une requête RIP-1.

5 Compatibilité

Le RFC [1] a fait preuve d'une grande prévoyance dans sa spécification du traitement des numéros de version. Il spécifie que les messages RIP de version 0 doivent être éliminés, que les messages RIP de version 1 doivent être éliminés si n'importe lequel des champs « doit être nul » (Must Be Zero, MBZ) est non-nul, et que les messages RIP de toute version supérieure à 1 ne devraient pas être éliminés simplement à cause du fait qu'un champ MBZ contient une valeur non différente de zéro. Cela signifie que la nouvelle version de RIP est totalement compatible vers l'arrière avec des implémentations existantes de RIP qui adhèrent à cette partie de la spécification.

5.1 Interrupteur de compatibilité

Un interrupteur de compatibilité est nécessaire pour deux raisons. D'abord, il existe des implémentations de RIP-1 qui ne suivent pas le RFC [1] comme décrit ci-dessus. Ensuite, l'utilisation de la transmission multidestinaire empêcherait les systèmes RIP-1 de recevoir des mises à jour RIP-2 (ce qui peut être une fonctionnalité souhaitée dans certains cas). Ce interrupteur devrait être configurable au niveau interface.

Le interrupteur dispose de quatre réglages possibles : RIP-1, dans lequel seuls les messages RIP-1 sont envoyées ; RIP-1 compatibility [11], dans lequel les messages RIP-2 sont diffusés ; RIP-2, dans lequel les messages RIP-2 sont transmis en mode multidestinataire ; et « none », qui désactive l'envoi de messages RIP. Il est recommandé que le réglage par défaut soit RIP-1 ou RIP-2, mais pas compatibilité RIP-1. Cela est dû aux problèmes potentiels qui peuvent se produire dans certaines topologies. RIP-1 compatibility ne devrait être utilisé que lorsque les conséquences de son utilisation sont bien comprises par l'administrateur réseau.

Pour être complets, les routeurs devraient également implémenter un interrupteur de contrôle de réception qui déterminerait s'il faut accepter RIP-1 uniquement, RIP-2 uniquement, les deux, ou aucun. Il devrait être configurable au niveau interface. Il est recommandé que le défaut soit compatible avec le défaut choisi pour l'émission de mises à jour.

5.2 Authentification

L'algorithme suivant devrait être utilisé pour authentifier un message RIP. Si le routeur n'est pas configuré pour authentifier les messages RIP-2, alors les messages RIP-1 et les messages RIP-2 non authentifiés seront acceptés ; les messages RIP-2 non authentifiés seront éliminés. Si le routeur est configuré pour authentifier les messages RIP-2, alors les messages RIP-1 et les messages RIP-2 qui passent le test d'authentification seront acceptés ; des messages RIP-2 non authentifiés ou dont l'authentification a échoué seront éliminés. Pour une sécurité maximale, les messages RIP-1 devraient être ignorés quand l'authentification est utilisée (voyez la section 4.1) ; sinon, les informations de routage provenant des messages seront propagés par les routeurs RIP-1 d'une façon non authentifiée.

Puisqu'une entrée d'authentification est marquée par une identificateur de famille d'adresses de 0xFFFF, un système RIP-1 ignorerait cette entrée car elle appartiendrait alors à une famille d'adresses différente d'IP. Il devrait ainsi être noté que l'utilisation de l'authentification n'empêchera pas les systèmes RIP-1 de voir des messages RIP-2. Si c'est ce que l'on souhaite, cela peut être fait en utilisant la transmission multidestinataire, comme décrit dans les sections 4.5 et 5.1.

5.3 Plus grand infini

Tant que nous parlons de la compatibilité, il y a une chose réclamée par certains : l'augmentation de l'infini. La raison principale pour laquelle cela ne peut être fait est que cela violerait la compatibilité descendante. Un infini supérieur troublerait évidemment des versions plus anciennes de RIP. Au mieux, elles ignoreraient la route comme elles ignoreraient une métrique de 16. Il y a également eu une proposition pour que le champ ' métrique ' n'emploie qu'un octet et qu'on utilise les trois octets supérieurs, mais cela briserait la compatibilité avec les implémentations qui traitent la métrique comme une entité sur 4 octets.

5.4 Liens sans adresse

Comme dans RIP-1, les liens sans adresse ne sont pas supportés par RIP-2.

[11] compatibilité avec RIP-1

6 Interactions entre les versions 1 et 2

Étant donné que les paquets de la version 1 ne contiennent pas d'informations sur les sous-réseaux, la sémantique employée par les routeurs sur des réseaux qui utilisent à la fois la version 1 et la version 2 devrait être limitée à celle de la version 1. Sinon, il serait possible de créer des routes « trou noir » (c.-à-d. des routes vers des réseaux qui n'existent pas) ou de créer des informations de routage excessives dans un environnement utilisant la version 1.

Certaines implémentations essaient de résumer automatiquement des groupes de routes adjacentes en des entrées uniques, le but étant de diminuer le nombre total d'entrées. C'est appelé l'auto-résumé.

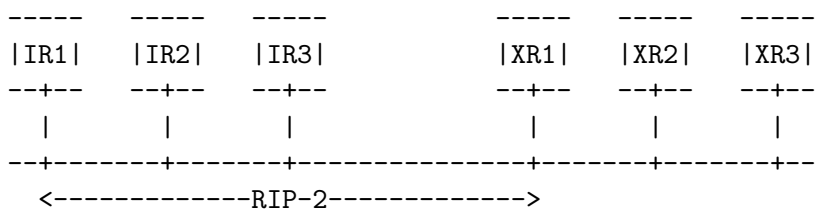
Spécifiquement, lors de l'utilisation conjointe des versions 1 et 2 dans un réseau, un unique masque de sous-réseau devrait être utilisé dans le réseau entier. De plus, les mécanismes d'auto-résumé devraient être désactivés pour de tels réseaux, et les implémentations doivent fournir des moyens pour cela.

7 Considérations de sécurité

Le protocole RIP de base n'est pas un protocole sécurisé. Pour rendre RIP-2 conforme aux protocoles de routage plus modernes, un mécanisme d'authentification extensible a été incorporé dans les améliorations apportées au protocole. Ce mécanisme est décrit dans les sections 4.1 et 5.2. La sécurité est encore davantage améliorée par le mécanisme décrit dans [3].

Annexes

Voici un exemple simple d'utilisation du champ ' saut suivant ' dans une entrée RIP.



Supposons que IR1, IR2 et IR3 sont tous des routeurs « internes » qui dépendent d'une administration (p.ex. un campus) qui a choisi d'utiliser RIP-2 pour IGP. XR1, XR2 et XR3, d'autre part, dépendent d'une administration séparée (p.ex. un réseau régional, duquel fait partie le campus) et utilisent un autre protocole de routage (p.ex. OSPF). XR1, XR2 et XR3 échangent des informations de routage entre eux de sorte qu'ils savent que les meilleures routes vers les réseaux N1 et N2 passent par XR1, vers N3, N4 et N5 par XR2, et vers N6 et N7 via XR3. En fixant le champ de saut suivant correctement (vers XR2 pour N3/N4/N5, vers XR3 pour N6/N7), seul XR1 doit échanger des routes RIP-2 avec IR1/IR2/IR3 pour que le routage se produise sans saut additionnel via XR1. Sans l'utilisation du saut suivant (par exemple si RIP-1 était utilisé), XR2 et XR3 devraient également participer au protocole RIP-2 pour éliminer les sauts excédentaires.

Bibliographie

[1] C. Hedrick, *Routing Information Protocol*, STD 34, RFC 1058, Rutgers University, juin 1988.

[2] G. Malkin, F. Baker, *RIP Version 2 MIB Extension*, RFC 1389, janvier 1993.

[3] F. Baker, R. Atkinson, *RIP-II MD5 Authentication*, RFC 2082, janvier 1997.

[4] R.E Bellman, *Dynamic Programming*, Princeton University Press, Princeton, N.J., 1957.

[5] D.P. Bertsekas, R.G. Gallaher, *Data Networks*, Prentice-Hall, Englewood Cliffs, N.J., 1987.

[6] R. Braden, J. Postel, *Requirements for Internet Gateways*, STD 4, RFC 1009, juin 1987.

[7] D.R. Boggs, J.F. Shoch, E.A. Taft, R.M. Metcalfe, *Pup : An Internetwork Architecture*, IEEE Transactions on Communications, avril 1980.

[8] L.R. Ford Jr., D.R. Fulkerson, *Flows in Networks*, Princeton, N.J., 1962.

[9] Xerox Corp., *Internet Transport Protocols*, Xerox System Integration Standard XSIS 028112, décembre 1981.

[10] S. Floyd, V. Jacobson, *The synchronization of Periodic Routing Messages*, ACM Sigcom '93 symposium, septembre 1993.

[11] F. Baker, *Requirements for IP Version 4 Routers.*, RFC 1812, juin 1995.

Adresse de l'auteur

Gary Scott Malkin
Bay Networks
8 Federal Street
Billerica, MA 01821

Tél. : (978) 916-4237
E-mail : gmalkin@baynetworks.com

Déclaration complète des droits d'auteur