Groupe de travail Réseau
Request for Comments : 1633
Caégorie : Information
Traduction Claude Brière de L'Isle

R. Braden, ISI
D. Clark, MIT
S. Shenker, Xerox PARC
juin 1994

Intégration de services dans l'architecture de l'Internet : généralités

Statut de ce mémoire
Le présent mémoire apporte des informations pour la communauté de l'Internet. Il ne spécifie aucune sorte de norme de l'Internet. La distribution du présent mémoire n'est soumise à aucune restriction.

Résumé
Le présent mémoire expose une proposition d'extension à l'architecture et aux protocoles de l'Internet pour fournir des services intégrés, c'est-à-dire de prendre en charge le temps réel ainsi que le service actuel en différé de IP. Cette extension est nécessaire pour satisfaire le besoin croissant de service en temps réel pour diverses applications nouvelles, parmi lesquelles la téléconférence, les séminaires à distance, la téléscience, et la simulation répartie.

Le présent mémoire représente la production directe des travaux récents de Dave Clark, Scott Shenker, Lixia Zhang, Deborah Estrin, Sugih Jamin, John Wroclawski, Shai Herzog, et Bob Braden, et s'appuie indirectement sur les travaux de nombreux autres.

Table des matières

1. Introduction

Les diffusions groupées des réunions de l'IETF sur l'Internet ont constitué une expérience à grande échelle de l'envoi de la voix numérisée et de vidéo à travers une infrastructure de commutation de paquets. Ces expériences très visibles se sont appuyées sur trois technologies qui les ont rendues possibles. (1) De nombreuses stations de travail modernes sont maintenant équipées de matériels multimédia incorporés, y compris des codecs audio et des collecteurs de trames vidéo, et l'équipement vidéo indispensable est maintenant d(un faible coût. (2) La diffusion groupée IP qui n'est pas encore disponible partout dans les routeurs commerciaux, est fournie par le MBONE, un cœur de réseau de diffusion groupée temporaire. (3) Des applications audio et vidéo numériques hautement sophistiquées ont été développées.

Ces expériences ont aussi montré qu'un important élément technique manque encore : les applications en temps réel ne fonctionnent souvent pas bien à travers l'Internet à cause des délais variables de mise en file d'attente et des pertes dues à l'encombrement. L'Internet, tel que conçu à l'origine, n'offre qu'une qualité de service (QS) très simple, la livraison au mieux des données en point à point. Avant que des applications en temps réel, telles que la vidéo à distance, les conférences multimédia, la visualisation et la réalité virtuelle puissent être largement utilisées, l'infrastructure de l'Internet doit être modifiée pour prendre en charge la QS en temps réel, qui fournit un certain contrôle sur les délais des paquets de bout en

bout. Cette extension doit être conçue à partir du début pour la diffusion groupée, car la simple généralisation à partir du cas de l'envoi individuel (en point à point) ne fonctionne pas.

La QS en temps réel n'est pas le seul problème pour la prochaine génération de gestion du trafic dans l'Internet. Les opérateurs de réseau demandent la capacité de contrôler le partage de la bande passante sur des liaisons particulières entre différentes classes de trafic. Ils veulent être capables de diviser le trafic en plusieurs classes administratives et allouer à chacune un pourcentage minimum de la bande passante de la liaison dans des conditions de surcharge, tout en permettant que la bande passante non utilisée soit disponible aux autres moments. Ces classes peuvent représenter, par exemple, différents groupes d'utilisateurs ou différentes familles de protocoles. Une telle facilité de gestion est couramment appelée du partage de liaison contrôlé. Nous utilisons le terme de services intégrés (IS) pour un modèle de service Internet qui comporte le service au mieux *(best-effort)*, le service en temps réel, et le partage de liaison contrôlé.

Les exigences et les mécanismes des services intégrés ont fait l'objet de nombreuses discussions et recherches dans les dernières années (la littérature est trop importante pour pouvoir faire ici même la liste d'un échantillon représentatif ; voir les références dans [CSZ92], [Floyd92], [Jacobson91], [JSCZ93], [Partridge92], [SCZ93], [RSVP93a] pour une liste partielle). Ce travail a conduit à l'approche unifiée de la prise en charge des services intégrés qui est décrite dans le présent mémoire. Nous pensons qu'il est maintenant temps de commencer les travaux d'ingénierie qui doivent précéder le déploiement des services intégrés dans l'Internet.

La Section 2 du présent mémoire introduit les éléments d'une extension IS de l'Internet. La Section 3 expose les modèles de service en temps réel [SCZ93a], [SCZ93b]. La Section 4 expose le contrôle du trafic, les algorithmes de transmission à utiliser dans les routeurs [CSZ92]. La Section 5 expose le concept de RSVP, un protocole d'établissement de ressources compatible avec les hypothèses de notre modèle IS [RSVP93a], [RSVP93b].

2. Éléments de l'architecture

Le modèle de service fondamental de l'Internet, tel qu'incorporé dans le service de livraison au mieux de IP, est resté inchangé depuis le début du projet de recherche Internet il y a 20 ans [CerfKahn74]. Nous proposons maintenant d'altérer ce modèle pour mettre en application des services intégrés. D'un point de vue académique, changer le modèle de service de l'Internet est une entreprise majeure ; cependant, son impact est atténué par le fait que nous souhaitons seulement étendre l'architecture d'origine. Les nouveaux composants et mécanismes à ajouter vont compléter mais non remplacer le service IP de base.

Abstraitement, l'extension architecturale proposée se compose de deux éléments :
(1) un modèle de service étendu, que nous appelons le modèle IS *(Integrated Service)*, et
(2) un cadre de mise en œuvre de référence qui nous donne un ensemble de vocabulaire et une organisation générique de programme pour réaliser le modèle IS.
Il est important de séparer le modèle de service, qui définit le comportement visible de l'exterieur, de la discussion de la mise en œuvre, qui peut (et devrait) changer durant la vie du modèle de service. Cependant, les deux sont liés ; pour que le modèle de service soit crédible, il est utile de fournir un exemple de la façon dont il pourrait être réalisé.

2.1 Modèle des services intégrés

Le modèle IS que nous proposons inclut deux sortes de services orientés vers le trafic en temps réel : le service garanti et le service prévisible. Il intègre ces services avec le partage de liaison contrôlé, et il est conçu comme fonctionnant bien avec la diffusion groupée aussi bien que l'envoi individuel. La présentation du modèle IS sera faite à la Section 3, et nous allons d'abord exposer les hypothèses clés du modèle.

La première hypothèse est que les ressources (par exemple, la bande passante) doivent être explicitement gérées afin de satisfaire aux exigences d'application. Cela implique que la "réservation de ressource" et le "contrôle d'admission" sont les pierres angulaires du service. Une autre approche, que nous ne suivons pas, serait de tenter de prendre en charge le trafic en temps réel sans aucun changement explicite du modèle de service de l'Internet.

Le service en temps réel exige par nature des garanties de service, et nous estimons que des garanties ne peuvent être menées à bien sans réservations. Le terme de "garantie" est à interpréter ici au sens large ; elles peuvent être absolues ou statistiques, strictes ou approximatives. Cependant, l'usager doit être capable d'obtenir un service dont la qualité soit suffisamment prévisible pour que l'application puisse fonctionner d'une façon acceptable sur une durée déterminée par l'usager. Là encore, "suffisamment" et "acceptable" sont des termes vagues. En général, des garanties plus strictes ont un coût plus élevé en ressources rendues indisponibles pour un partage avec d'autres.

Les arguments suivant ont été soulevés contre les garanties de ressource dans l'Internet.

- "La bande passante deviendra infinie."
  Les incroyables capacités de transport d'une fibre optique en conduisent certains à conclure qu'à l'avenir la bande passante sera si abondante, omniprésente, et bon marché qu'il n'y aura pas de délais de communication autres que la vitesse de la lumière, et donc qu'il ne sera abolument pas nécessaire de réserver des ressources. Nous pensons cependant que cela sera impossible à court terme et peu vraisemblable à moyen terme. Alors que la bande passante brute peut sembler très bon marché, la bande passante fournie au titre d'un service réseau ne deviendra vraisemblablement pas si bon marché qu'on puisse faire de son gaspillage le principe conceptuel le plus justifié économiquement. Même si la bande passante à bas prix devient finalement couramment disponible, nous ne pensons pas qu'elle sera disponible "partout" dans l'Internet. Tant que nous ne saurons pas traiter l'encombrement des liaisons, les services en temps réeel seront tout simplement exclus. Cette restriction est inacceptable.

- "Une simple priorité est suffisante."
  Il est vrai que de simplement donner une priorité plus élevée au trafic en temps réel conduirait à un service en temps réel adéquat à certains moments et dans certaines conditions. Mais la priorité est un mécanisme de mise en œuvre, pas un modèle de service. Si nous définissons le service au moyen d'un mécanisme spécifique, nous pouvons ne pas avoir les caractéristiques exactes recherchées. Dans le cas d'une simple priorité, la question qui se pose est qu'aussitôt que trop de flux en temps réel sont en compétition pour la priorité la plus élevée, qui sera le flux le plus dégradé ?. Restreindre notre service à ce seul mode de défaillance n'est pas acceptable. Dans certains cas, les usagers vont demander que certains flux réussissent alors que certaines nouvelles demandent reçoivent un signal d'occupation.

- "Les applications peuvent s'adapter."
  Le développement d'applications adaptatives en temps réel, comme le programme audio VAT de Jacobson, n'élimine pas le besoin de limiter les délais de livraison du paquet. Les exigences de l'homme en matière d'interaction et d'intelligibilité limitent la gamme possible d'adaptation aux délais du réseau. Nous avons vu en expérimentations réelles que, alors que VAT peut s'adapter à des délais de réseau de plusieurs secondes, les usagers trouvent que l'interaction est impossible dans ces cas là.

Nous en concluons qu'il y a une exigence incontournable pour que les routeurs soient capables de réserver des ressources, afin de fournir une qualité de service particulière pour des flux de paquets spécifiques. Ceci exige à son tour un état spécifique du flux dans les routeurs, ce qui représente un changement important et fondamental du modèle de l'Internet. L'architecture de l'Internet a été fondée sur le concept que tout état en rapport avec le flux devait être dans les systèmes d'extrémité [Clark88]. La conception de la suite de protocoles TCP/IP sur ce concept a conduit à une robustesse qui est une des clés de son succès. Nous discutons dans la section 5 de la façon dont l'ajout de l'état du flux aux routeurs pour la réservation de ressource peut être rendu "conditionnel", pour préserver la robustesse de la suite des protocoles de l'Internet.

Il y a dans la réalité un effet collatéral de la réservation de ressource chez les routeurs. Comme elle implique que certaine utilisateurs obtiennent un service privilégié, la réservation de ressource va devoir mettre en application des contrôles de politique et administratifs. Cela va à son tour conduire à deux sortes d'exigences d'authentification : l'authentification des usagers qui font des demandes de réservation, et l'authentification des paquets qui utilisent les ressources réservées. Cependant, ces questions ne se posent pas seulement pour "IS" ; d'autres aspects de l'évolution de l'Internet, y compris la commercialisation et la sécurité commerciale sonduisent aux mêmes exigences. Nous ne dicuterons pas plus avant de la politique ou de la sécurité dans le présent mémoire, mais cela demande qu'on y prête attention.

Nous faisons une autre hypothèse fondamentale, qu'il est désirable d'utiliser l'Internet comme infrastructure commune pour prendre en charge la communication aussi bien en temps réel qu'en différé. On pourrait autrement construire une infrastructure parallèle entièrement nouvelle pour les services en temps réel, laissant l'Internet inchangé. Nous rejetons cette approche, car elle perdrait les avantages significatifs du partage statistique entre le trafic en temps réel et le trafic en différé, et elle serait beaucoup plus complexe à construire et administrer qu'une infrastructure commune.

En plus de cette hypothèse d'infrastructure commune, nous adoptons un modèle de pile de protocole unifiée, qui emploie un seul protocole de couche internet pour le service aussi bien en temps réel qu'en différé. Donc, nous proposons d'utiliser le protocole existant de couche internet (par exemple, IP ou CLNP) pour les données en temps réel. Une autre approche serait d'ajouter un nouveau protocole de temps réel dans la couche internet [ST2-90]. Notre approche de pile unifiée fait l'économie du mécanisme, et elle nous permet de nous plier aisément au partage de liaison contrôlé. Cela traite aussi le problème de la couverture partielle, c'est-à-dire de permettre l'interopération entre systèmes Internet à capacité IS et systèmes qui n'ont pas reçu l'extension, sans avoir recours à un tunnelage complexe.

Nous adoptons l'idée qu'il devrait y avoir un seul modèle de service pour l'Internet. Si il y avait différents modèles de service dans les différentes parties de l'Internet, il serait très difficile de voir comment on pourrait faire des garanties de qualité de service de bout en bout. Cependant, un seul modèle de service n'implique pas nécessairement une seule mise en œuvre de programmation des paquets ou de contrôle d'admission. Bien que des mécanismes spécifiques de programmation des paquets et de contrôle d'admission qui satisfassent notre modèle de service aient été développés, il est assez possible

que d'autres mécanismes puissent aussi satisfaire le modèle de service. La cadre de mise en œuvre de référence, introduit ci-dessous, est destiné à permettre la discussion des questions de mise en œuvre sans rendre obligatoire une conception unique.

Sur la base de ces considérations, nous pensons qu'une extension IS qui inclurait des états de flux supplémentaires dans les routeurs et un mécanisme explicite d'établissement est nécessaire pour fournir le service voulu. Une solution partielle qui ne tiendrait pas compte de ce point ne serait pas un investissement avisé. Nous pensons que les extensions proposées préservent l'essentiel de la robustesse et de l'efficacité de l'architecture de l'Interent, et qu'elles permettent une gestion efficace des ressources du réseau ; ceci sera un objectif important même si la bande passante devient très bon marché.

2.2 Cadre de référence de mise en œuvre

Nous proposons un cadre de mise en œuvre de référence pour réaliser le modèle IS. Ce cadre comporte quatre composants : le programmeur de paquets, le sous-programme de contrôle d'admission, le classeur, et le protocole d'établissement de réservations. Ils sont exposés brièvement ci-dessous et plus en détails dans les sections 4 et 5.

Dans l'exposé qui suit on définit le "flux" comme une abstraction qui est un flux distinct de datagrammes qui résultent de l'activité d'un seul usager et demandent la même QS. Par exemple, un flux peut consister en une connexion de transport ou un flux de vidéo entre une certaine paire d'hôtes. C'est la plus fine granularité de flux de paquet que peut distinguer l'IS. On définit un flux comme unidirectionnel, c'est-à-dire, comme ayant une seule source mais N destinations. Donc, une téléconférence à N directions va généralement exiger N flux, prenant leur originé à chaque site.

Dans l'Internet d'aujourd'hui, la transmission IP est complètement égalitaire ; tous les paquets reçoivent la même qualité de service, et les paquets sont normalement transmis en utilisant une discipline de mise en file d'attente FIFO *(premier entré – premer sorti)*. Pour les services intégrés, un routeur doit mettre en œuvre une QS appropriée pour chaque flux, en accord avec le modèle de service. La fonction de routeur qui crée différentes qualités de service est appelée "contrôle de trafic". Le contrôle de trafic est à son tour mis en œuvre par trois composants : le programmeur de paquets, le classeur, et le contrôle d'admission.

- Programmeur de paquet
  Le programmeur de paquets gère la transmission des différents flux de paquets en utilisant un ensemble de files d'attente et peut-être d'autres mécanismes comme les temporisateurs. Le programmeur de paquets doit être mis en œuvre au point où les paquets sont mis en file d'attente ; c'est le niveau pilote de sortie d'un système d'exploitation normal, et cela correspond au protocole de couche de liaison. Les détails de l'algorithme de programmeur peuvent être spécifiques du support de sortie particulier. Par exemple, le pilote de sortie aura besoin d'invoquer les contrôles de couche liaison appropriés lorsque il assure l'interface avec une technologie de réseau qui a un mécanisme interne d'allocation de bande passante.

  Un programmeur de paquet expérimental a été construit qui met en œuvre le modèle IS décrit à la Section 3 et dans [SCZ93] ; c'est ce qu'on appelle le programmeur CSZ et on en parle un peu plus à la Section 4. On note que le schéma CSZ n'est pas obligatoire pour réaliser notre modèle de service ; bien sûr, pour les parties du réseau qui sont connues pour n'être pas encombrées, FIFO va fournir un service satisfaisant.

  Il y a un autre composant qui pourrait être considéré comme faisant partie du programmeur de paquet ou comme un élément séparé : l'estimateur [Jacobson91]. Cet algorithme est utilisé pour mesurer les propriétes du flux de trafic sortant, pour tenir des statistiques qui contrôlent la programmation de paquets et le contrôle d'admission. Le présent mémoire considère l'estimateur comme faisant partie du programmeur de paquets.

- Classeur
  Pour les besoins du contrôle de trafic (et de comptabilité) chaque paquet entrant doit être transposé dans une classe ; tous les paquets de la même classe obtiennent le même traitement de la part du programmeur de paquets. Cette transposition est effectuée par le classeur. Le choix d'une classe peut être fondé sur le contenu du ou des en-têtes existants du paquet et/ou d'un numéro de classement supplémentaire ajouté à chaque paquet.

  Un classe peut correspondre à une catégorie large de flux, par exemple tous les flux de vidéo ou tous les flux attributables à une certaine organisation. D'un autre côté, une classe peut ne détenir qu'un seul flux. Une classe est une abstraction qui peut être locale pour un certain routeur ; le même paquet peut être classé différemment par d'autres routeurs le long du chemin. Par exemple, les routeurs de cœur de réseau peuvent choisir de transposer de nombreux flux en quelques classes agrégées, alors que des routeurs plus proches de la périphérie, où il y a beaucoup moins d'agrégation, pourront utiliser une classe différente pour chaque flux.

- Contrôle d'admission
  Le contrôle d'admission met en œuvre l'algorithme de décision qu'utilise un routeur ou un hôte pour déterminer si on peut accorder à un nouveau flux la QS demandée sans pénaliser des garanties antérieures. Le contrôle d'admission est invoqué à chaque nœud pour prendre une décision locale d'acceptation/rejet, au moment où un hôte demande un service en temps réel le long d'un chemin sur l'Internet. L'algorithme de contrôle d'admission doit être cohérent avec le modèle de service, et il fait logiquement partie du contrôle du trafic. Bien qu'il y ait encore des questions ouvertes à la recherche dans le contrôle d'admission, on peut s'appuyer sur [JCSZ92].

  Le contrôle d'admission est parfois confondu avec la régulation ou la mise en application qui sont des fonctions paquet par paquet à la "bordure" du réseau pour s'assurer qu'un hôte ne viole pas les caractéristiques de trafic promises. Nous considérons la régulation comme une des fonctions du programmeur de paquets.

  En plus de s'assurer de la satisfaction des garanties de QS, le contrôle d'admission va s'occuper de la mise en application des politiques administratives sur les réservations de ressources. Certaines politiques vont exiger l'authentification de ceux qui demandent des réservations. Finalement, le contrôle d'admission va jouer un rôle important dans l'établissement des rapports administratifs et de comptabilité.

Le quatrième et dernier composant de notre cadre de mise en œuvre est un protocole d'établissement de réservations, qui est nécessaire pour créer et entretenir un état spécifique du flux chez les hôtes de point d'extrémité et les routeurs le long du chemin d'un flux. La Section 5 expose un protocole d'établissement de réservations appelé RSVP (ReSerVation Protocol) [RSVP93a, RSVP93b]. Il n'est peut-être pas possible d'insister pour qu'il n'y ait qu'un seul protocole de réservation dans l'Internet, mais il nous paraît que le choix entre plusieurs protocoles de réservation serait une cause de confusion. Nous pensons que plusieurs protocoles ne devraient exister que si ils prennent en charge des modes de réservation différents.

Les exigences d'établissement pour la portion à partage de liaison du modèle de service sont beaucoup moins claires que celles sur les réservations de ressources. Bien qu'on s'attende à ce qu'une grande partie en soit faite par les interfaces de gestion de réseau, et n'aient donc pas besoin de faire partie de l'architecture globale, on peut aussi avoir besoin de RSVP pour jouer un rôle dans la fourniture de l'état requis.

Afin de déclarer ses exigences de ressources, une application doit spécifier la QS désirée en utilisant une liste de paramètres qui est appelée une "flowspec" [Partridge92]. La flowspec *(ou spécification de flux)* est portée par le protocole d'établissement de réservation, passée au contrôle d'admission pour vérifier son acceptabilité, et finalement utilisée pour établir les paramètres du mécanisme de programmation de paquet.

La Figure 1 montre comment ces composants peuvent s'assembler dans un routeur IP qui a été mis à niveau pour fournir des services intégrés. Le routeur a deux grandes divisions fonctionnelles : le chemin de transmission en dessous de la double ligne horizontale, et le code d'arrière plan au-dessus de la ligne.

Le chemin de transmission du routeur est exécuté pour chaque paquet et doit donc être très optimisé. Bien sûr, dans la plupart des routeurs commerciaux, sa mise en œuvre implique une assistance matérielle. Le chemin de transmission se divise en trois sections : pilote d'entrée, transmetteur internet, et pilote de sortie. Le transmetteur internet interprète l'en-tête de protocole d'inter réseautage approprié de la suite de protocoles, par exemple, l'en-tête IP pour TCP/IP, ou l'en-tête CLNP pour OSI. Pour chaque paquet, un transmetteur internet exécute un classeur dépendant de la suite puis passe le paquet et sa classe au pilote de sortie approprié. Un classeur doit être à la fois général et efficace. Pour l'efficacité, un mécanisme courant devrait être utilisé pour le classement des resources comme pour la recherche de chemin.

Le pilote de sortie met en œuvre le programmeur de paquet. (Les adeptes de la mise en couche observeront que le pilote de sortie a maintenant deux sections distinctes : le programmeur de paquet qui est largement indépendant des mécaniques détailles de l'interface, et le pilote d'entrée/sortie réel qui n'est concerné que par l'endurance du matériel. L'estimateur se trouve quelque part entre les deux. Nous prenons simplement note de ce fait, sans suggérer qu'on en fasse un principe.)

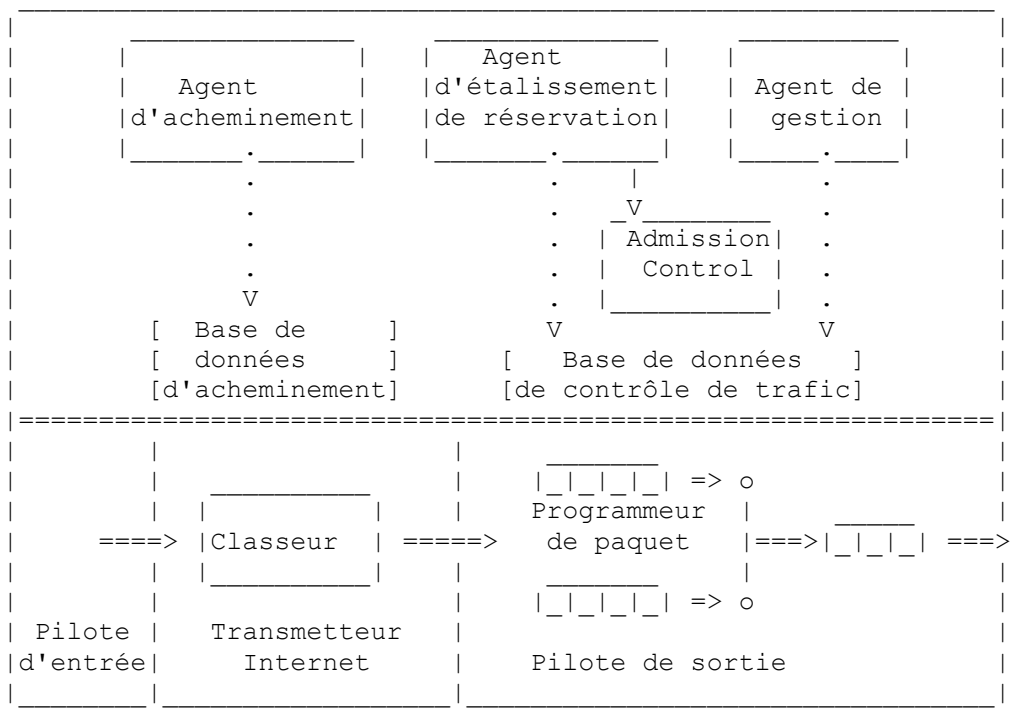

Figure 1 : Modèle de référence de mise en œuvre de routeur

Le code d'arrière plan est simplement chargé dans la mémoire du routeur et exécuté par un CPU généraliste. Ces sous programmes d'arrière plan créent des structures de données qui contrôlent le chemin de transmission. L'agent d'acheminement met en œuvre un protocole d'acheminement et construit une base de données d'acheminement. L'agent d'établissement de réservation met en œuvre le protocole utilisé pour établir les réservations de ressource ; voir la Section 5. Si le contrôle d'admission donne son accord sur une nouvelle demande, les changements appropriés sont faits à la base de données de classeurs et de programmeurs de paquet pour mettre en œuvre la QS désirée. Finalement, chaque routeur prend en charge un agent de gestion de réseau. Cet agent doit être capable de modifier les bases de données de classeur et de programmeur de paquet pour établir le partage de liaison contrôlé et pour régler les politiques de contrôle d'admission.

Le cadre de mise en œuvre pour un hôte est généralement similaire à celui pour un routeur, avec les applications en plus. Au lieu d'être transmises, les données des hôtes sont générées et se terminent dans une application. Une application qui a besoin de la QS de temps réel pour un flux doit d'une certaine façon invoquer un agent d'établissement de réservation local. La meilleure façon d'assurer l'interface avec les applications reste encore à déterminer. Par exemple, il pourrait y avoir une API explicite pour l'établissement de ressources du réseau, ou l'établissement pourrait être invoqué implicitement au titre de la fonction de programmation du système d'exploitation. Le sous-programme de sortie IP d'un hôte peut n'avoir pas besoin de classeur, car l'allocation de la classe à un paquet peut être spécifiée dans la structure locale de contrôle d'entrée sortie correspondant au flux.
Dans les routeurs, le service intégré va exiger des changements à la fois au chemin de transmission et aux fonctions de base. Le chemin de transmission, qui peut dépendre de l'accélération des performances des matériels, sera le plus difficile et le plus coûteux à changer. Il sera vital de choisir un ensemble de mécanismes de contrôle de trafic qui soit général et adaptable à une grande variété d'exigences de politique et de circonstances futures, et qui puisse être mis en œuvre efficacement.

3. Modèle de services intégrés

Un modèle de service est incorporé dans l'interface de service réseau invoquée par les applications pour définir l'ensemble des services qu'elles peuvent demander. Quoique aussi bien la technologie du réseau sous-jacent que la suite d'applications qui s'y exécutent doivent évoluer, la nécessité de la compatibilité exige que cette interface de service reste relativement stable (ou plus exactement, extensible ; on s'attend bien à de nouveaux services à l'avenir mais on sait bien aussi qu'il sera difficile de changer les services existants). À cause de son impact durable, le modèle de service ne devrait pas être conçu par référence à un artifice spécifique du réseau mais plutôt être fondé sur des exigences fondamentales de service.

Nous allons maintenant décrire brièvement une proposition d'un ensemble de cœur de services pour l'Internet ; ce modèle de cœur de service proposé est décrit plus complètement dans [SCZ93a], [SCZ93b]. Ce modèle de cœur de service vise les

services qui se rapportent très directement à l'instant de livraison des paquets. Nous laissons les services restants (tels que l'acheminement, la sécurité, ou la synchronisation des flux) pour d'autres textes sur la normalisation. Un modèle de service consiste en un ensemble d'engagements de service ; en réponse à une demande de service, le réseau s'engage à fournir un service. Ces engagements de service peuvent être rangés dans des catégories selon l'entité à laquelle ils sont faits : ils peuvent être pris pour des flux individuels ou pour des entités collectives (des classes de flux). Les engagements de service pris envers des flux individuels sont destinés à fournir des performances d'application raisonnables, et sont donc conduits par les exigences ergonomiques de l'applications ; ces engagements de service se rapportent à la qualité de service délivrée à un flux individuel. Les engagements de service pris envers des entités collectives sont conduits par des exigences de partage de ressources, ou des exigences économiques ; ces engagements de service se rapportent aux ressources agrégées rendues disponibles aux diverses entités.

Dans cette section, on commence par explorer les exigences de service des flux individuels et on propose un ensemble de services correspondant. On exposera enseuite les exigences de service et les services pour le partage de ressources. Enfin, on conclura par des remarques sur l'abandon de paquet.

3.1 Exigences de qualité de service

Le cœur du modèle de service est concerné presque exclusivement par l'instant de livraison des paquets. Donc, le délai par paquet est la quantité centrale dont le réseau prend des engagements de qualité de service. On fait une hypothèse encore plus restrictive qui est que la seule quantité dont nous faisons un engagement quantitatif de service est la limite sur les délais maximum et minimum.

Le degré auquel les performances d'application dépendent d'un faible délai de service varie largement, et on peut faire plusieurs distinctions qualitatives entre les applications sur la base du degré de leur dépendance. Une classe d'applications a besoin des données de chaque paquet à un certain moment et, si les données ne sont pas arrivée à ce moment là, elles n'ont plus aucune utilité ; on les appelle des applications en temps réel. Une autre classe d'applications va toujours attendre que les données arrivent ; on les appelle des applications "élastiques". Nous allons maintenant examiner séparément les exigences de délai de ces deux classes.

3.1.1 Applications en temps réel

Une importante classe de telles applications en temps réel, qui sont les seules applications en temps réel que nous considérons explicitement dans les arguments qui suivent, sont les applications de "playback". Dans une application de playback, la source prend un signal, le met en paquets, puis transmet les paquets sur le réseau. Le réseau introduit inévitablement des variations dans le délai de livraison des paquets. Le receveur défait les paquets de données puis tente de bonne foi de reproduire le signal. Cela est fait en mettant les données entrantes dans une mémoire tampon puis en reproduisant le signal à un délai de décalage fixé à partir du moment de départ original ; le terme de "point de playback" se réfère au moment de décalage dans le temps de ce délai fixé par rapport au point de départ original. Toutes données qui arrivent avant leur point de playback associé peuvent être utilisées pour reconstruire le signal ; les données qui arrivent après le point de playback sont inutiles dans la reconstruction du signal en temps réel.

Afin de choisir une valeur raisonnable pour le délai de décalage, une application a besoin d'une earactérisation "à priori" du délai maximum que vont rencontrer ses paquets. Cette caractérisation "à priori" pourrait aussi bien être fournie par le réseau dans un engagement de service quantitatif à une limite de délai, que par l'observation des délais rencontrés par les paquets précédemment arrivés ; l'application a besoin de savoir à quels délais s'attendre, mais cette attente n'a pas besoin d'être constante pour toute la durée du flux.

Les performances d'une application de playback sont mesurées sur deux dimensions : la latence et la fidélité. Certaines applications de playback, en particulier celles qui impliquent une interaction entre les deux extrémités d'une connexion comme un appel téléphonique, sont assez sensibles à la latence ; d'autres applications de playback, telles que la transmission d'un film ou d'une conférence, ne le sont pas. De même, des applications présentent une large gamme de sensibilité à la perte de fidélité. Nous allons considérer deux classes selon une dichotomie assez artificielle : les aplications intolérantes, qui exigent une répétition absolument fidèle, et les applications tolérantes, qui peuvent tolérer une certaine perte de fidélité. On s'attend à ce que le vaste domaine des applications audio et vidéo soit tolérant, mais on soupçonne aussi qu'il y aura d'autres applications, comme l'émulation de circuit, qui sont intolérantes.

Le délai peut affecter les performances des applications de playback de deux façons. D'abord, la valeur du délai de décalage, qui est déterminé par les prédictions sur le délai futur de paquet, détermine la latence de l'application. Ensuite, les délais des paquets individuels peuvent diminuer la fidélité de la réexécution en dépassant le délai de décalage ; l'application peut alors soit changer le délai de décalage afin de repasser les paquets en retard (ce qui introduit une distortion) soit simplement éliminer les paquets en retard (ce qui crée un signal incomplet). Les deux façons différentes de traiter les paquets en retard offrent un choix entre un signal incomplet et un signal distordu, et le choix optimal va dépendre des détails de l'application, mais le point important est que les paquet en retard dégradent nécessairement la fidélité.

Les applications intolérantes divent utiliser un délai de décalage fixé, car toute variation du délai de décalage va introduire une distortion dans la reproduction. Pour une distribution donnée des délais de paquets, ce délai fixé de décalage doit être supérieur au délai absolu maximum, pour éviter la possibilité de paquets en retard. Une telle application peut seulement régler de façon appropriée son délai de décalage si on lui donne une limite supérieur parfaitement fiable du délai maximum pour chaque paquet. On appelle un service caractérisé par une limite supérieure de délai parfaitement fiable un "service garanti", et on propose cela comme le modèle de service approprié pour les applications de playback intolérantes.

À l'opposé, les applications tolérantes n'ont pas besoin de régler leur délai de décalage à une valeur supérieure au délai absolu maximum, car elles peuvent tolérer quelques paquets en retard. De plus, au lieu d'utiliser une seule valeur fixée pour le délai de décalage, elles peuvent tenter de réduire leur latence en variant leurs délais de décalage en réponse aux délais de paquet réels rencontrés dans le récent passé. On appelle les applications qui font varier leur délai de décalage de cette manière des application de playback "adaptatives".

Pour les applications tolérantes, nous proposons un modèle de service appelé "service prévisible" qui fournit une limite de délai assez fiable, mais pas parfaitement fiable. Cette limite, à l'opposé de celle du service garanti, n'est pas fondée sur les hypothèses de plus mauvais cas du comportement des autres flux. Cette limite peut plutôt être calculée avec des prévisions relativement prudentes sur le comportement des autres flux. Si le réseau se trouve avoir tort et que la limite est violée, les performances de l'application vont peut-être en souffrir, mais les utilisateurs veulent tolérer de telles interruptions de service en échange du coût présumé inférieur du service. De plus, parce que de nombreuses applications tolérantes sont adaptatives, on enrichit le service prévisible afin qu'il donne aussi un service "minimax", qui essaye de minimiser le délai maximum a postériori. Ce service n'essaye pas de minimiser le délai de chaque paquet, mais plutôt de réduire la queue de la distribution de délais.

Il est clair que si elle a le choix, toutes choses égales par ailleurs, une application ne fera pas pire avec des limites absolument fiables qu'avec des limites assez fiables. Pourquoi alors offrir le service prévisible ? La considération clé est ici l'efficacité ; quand on relâche les exigences de service de parfaitement fiable à assez fiable, cela augmente le niveau d'utilisation du réseau qui peut être soutenu, et donc le prix du servive prévisible va vraisemblablement être inférieur à celui du service garanti. La classe de service prévisible est motivée par la conjecture que la pénalité en performances sera faible pour les applications tolérantes mais le gain d'efficacité globale sera assez grand.

Afin de fournir une limite au délai, la nature du trafic provenant de la source doit être caractérisée, et il doit y avoir un algorithme de contrôle d'admission qui assure qu'un flux demandé peut réellement être traité. Un point fondamental de notre architecture globale est que la caractérisation du trafic et le contrôle d'admission sont nécessaires pour ces services en temps réel à délai limité. Jusque à présent, nous avons supposé que le processus de génération des données d'une application était une propriété intrinsèque non affectée par le réseau. Cependant, il y a aura vraisemblablement de nombreuses applications audio et vidéo qui pourront ajuster leur schéma de codage et donc altérer les processus résultants de génération des données selon le service réseau disponible. Cette altération du schéma de codage va présenter un compromis entre la fidélité (du schéma de codage lui-même, pas du processus de répétition) et les exigences de bande passante du flux. De telles applications de playback à "taux adaptatif" ont l'avantage de pouvoir s'ajuster aux conditions actuelles du réseau non seulement en modifiant le réglage de leur point de playback mais aussi en ajustant le schéma de trafic lui-même. Pour les applications à taux adaptatif, les caractérisations de trafic utilisées dans l'engagement de service ne sont pas immuables. Nous pouvons donc enrichir le modèle de service en permettant que le réseau notifie (implicitement par l'abandon de paquet ou explicitement par des paquets de contrôle) aux applications à taux adaptatif de changer leur caractérisation de trafic.

3.1.2 Applications élastiques

Alors que les applications en temps réel n'attendent pas que les données en retard arrivent, les applications élastiques vont toujours attendre l'arrivée des données. Ce n'est pas que ces applications soient insensibles au délai ; au contraire, une augmentation significative du délai d'un paquet va souvent nuire aux performances de l'application. Le point clé est plutôt que l'application utilise normalement immédiatement les données qui arrivent au lieu de les mettre en mémoire tampon pour une utilisation ultérieure, et va toujours choisir d'attendre les données à venir plutôt que de faire le traitement sans elles. Comme les données qui arrivent peuvent être utilisées immédiatement, ces applications n'exigent aucune caractérisation à priori du service afin que l'application fonctionne. D'une façon générale, il est vraisemblable que pour une certaine distribution des délais de paquet, la perception des performances des applications élastiques va plus dépendre du délai moyen que de la queue de distribution du délai. On peut penser à plusieurs catégories de ces applications élastiques : salves interactives (Telnet, X, NFS), transferts interactifs en vrac (FTP), et transfert en vrac asynchrone (messagerie électronique, FAX). Les exigences de délai de ces applications élastiques varient des applications de salves assez exigeantes pour des applications de salves interactives à des applications assez laxistes pour les transferts asynchrones en vrac, les transferts interactifs en vrac étant intermédiaires entre ces deux extrèmes.

Un modèle de service approprié pour les applications élastiques est de fournir un service "aussitôt que possible", ou ASAP *(As Soon As Possible)*. (Pour la compatibilité avec l'utilisation historique, nous utiliserons le terme de "service au mieux" *(best-effort)* pour nous référer au service ASAP.) Nous proposons de plus d'offrir plusieurs classes de service au mieux pour refléter les sensibilités relatives au délai de différentes applications élastiques. Ce modèle de service permet aux applications de salves interactives d'avoir des délais inférieurs à ceux des applications interactives en vrac, qui à leur tour auront des délais inférieurs à ceux des applications asynchrones en vrac. À la différence des modèles de service en temps réel, les applications qui utilisent ce service ne sont pas soumise au contrôle d'admission.

La taxonomie des applications en exécution tolérante, exécution intolérante, et élastique n'est ni exacte ni complète, mais n'est utilisée que pour guider le développement du modèle du cœur de service. Le modèle résultant de cœur de service devrait être jugé non sur la validité de la taxonomie sous-jacente mais plutôt sur sa capacité à satisfaire de façon adéquate les besoins du spectre entier des applications. En particulier, toutes les applications en temps réel ne sont pas des applications de playback ; par exemple, on peut imaginer une application de visualisation qui affiche simplement l'image codée dans chaque paquet chaque fois qu'il arrive. Cependant, des applications qui ne sont pas de playback peuvent quand même utiliser le modèle de service en temps réel garanti ou prévisible, bien que ces services ne soient pas spécifiquement conçus pour leurs besoins. De même, les applications de playback ne peuvent pas être nettement classées en tolérantes ou intolérantes, mais se distribuent plutôt de façon continue ; offrir les deux services, garanti et prévisible, permet aux applications de faire leur propre compromis entre fidélité, latence, et coût. En dépit de ces évidentes déficiences de la taxonomie, nous pensons qu'elle décrit assez bien les exigences de service des applications actuelles et futures de sorte que notre modèle de cœur de service peut satisfaire de façon adéquate les besoins de toutes les applications.

3.2 Exigences et modèles de service de partage de ressource

Le paragraphe précédent traitait des engagements de qualité de service ; ces engagements imposent la façon dont le réseau doit allouer ses ressources entre les flux individuels. Cette allocation de ressources est normalement négociée flux par flux au fur et à mesure que ceux-ci demandent l'admission au réseau, et ne vise aucune des questions de politique qui surviennent lorsque on regarde des collections de flux. Pour traiter ces questions de politique collective, on va maintenant discuter des engagements de service de partage de ressources. On se rappelle que pour les engagements individuels de qualité de service, on se concentre sur le délai comme seule quantité intéressante. Ici, le postulat est que la quantité la plus intéressante pour le partage de ressources est la bande passanté agrégée sur les liaisons individuelles. Donc, ce composant du modèle de service, appelé le "partage de liaison", traite la question de comment partager la bande passante agrégée d'une liaison parmi les diverses entités collectives en fonction d'un certain ensemble de parts spécifiées. Il y a plusieurs exemples qui sont couramment utilisés pour expliquer les exigences du partage de liaison entre des entités collectives.

Partage de liaison entre plusieurs entités. -- Une liaison peut être acquise et utilisée conjointement par plusieurs organisations, agences gouvernementales ou autres. Elles peuvent souhaiter s'assurer qu'en cas de surcharge, la liaison est partagée de façon contrôlée, peut-être en proportion de l'investissement en capital de chaque entité. En même temps, elles peuvent souhaiter que lorsque la liaison est peu utilisée, n'importe laquelle des entités puisse utiliser toute la bande passante inutilisée.

Partage de liaison multi protocoles. -- Dans un Internet multi protocoles, il peut être souhaité empêcher une famille de protocoles (DECnet, IP, IPX, OSI, SNA, etc.) de surcharger la liaison et d'exclure ainsi les autres familles. Ceci est important parce que les différentes familles peuvent avoir des méthodes différentes pour détecter l'encombrement et y répondre, et que certaines méthodes peuvent être plus "agressives" que d'autres. Cela pourrait conduire à une situation dans laquelle un protocole se dégage plus rapidement qu'un autre en cas d'encombrement, et finit par n'avoir plus de bande passante. Cela peut exiger un contrôle explicite au routeur pour le corriger. Là encore, on peut s'attendre à ce que ce contrôle ne s'applique qu'en cas de surcharge, tout en permettant qu'une liaison inactive soit utilisée dans n'importe quelles proportions.

Partage multi services – Au sein d'une famille de protocoles comme IP, un administrateur peut souhaiter limiter la fraction de bande passante allouée aux diverses classes de service. Par exemple, un administrateur peut souhaiter limiter la quantité de trafic en temps réel à une certaine fraction de la liaison, pour éviter de préempter le trafic élastique comme FTP.

En termes généraux, le modèle de service de partage de liaison est fait pour partager la bande passante agrégée selon des parts spécifiées. On peut étendre ce modèle de service de partage de liaison à une version hiérarchisée. Par exemple, une liaison pourrait être divisée entre un certain nombre d'organisations, dont chacune diviserait l'allocation résultante entre un certain nombre de protocoles, dont chacun serait divisé entre un certain nombre de services. Ici, le partage est défini par une arborescence avec des parts allouées à chaque nœud d'extrémité.

Un modèle fluide idéalisé de partage instantané de liaison avec un partage proportionnel des excédents est le modèle de partage à processeur fluide (introduit dans [DKS89] et exploré plus en détails dans [Parekh92] et généralisé au cas hiérarchisé) où à chaque instant, la bande passante disponible est partagée entre les entités actives (c'est-à-dire, celles qui

ont des paquets dans la file d'attente) en proportion des parts allouées de la ressource. Ce modèle fluide présente le comportement de politique désiré mais est, bien sûr, une idéalisation irréaliste. Nous proposons alors que le modèle de service réel s'approche d'aussi près que possible du partage de bande passante produit par ce modèle fluide idéal. Il n'est pas nécessaire d'exiger que l'ordre spécifique de départ des paquets corresponde à celui du modèle fluide car on suppose que toutes les exigences détaillées de délai de paquet des flux individuels sont traitées par les engagements de qualité de service et que de plus, la satisfaction du service de liaison partagée fourni ne va probablement pas dépendre très sensiblement des petites différence de programmation impliquées par le modèle fluide de partage de liaison.

Nous avons précédement observé que le contrôle d'admission était nécessaire pour assurer que les engagements de service en temps réel pouvaient être satisfaits. De même, le contrôle d'admission sera aussi nécessaire poir assurer que les engagements de partage de liaison peuvent être satisfaits. Pour chaque entité, le contrôle d'admission doit empêcher le trafic garanti cumulé et le trafic prévisible de dépasser la part de liaison allouée.

3.3 Abandon de paquet

Jusque à présent, nous avons implicitement supposé que tous les paquets au sein d'un flux étaient également importants. Cependant, dans de nombreux flux audio et vidéo, certains paquets sont plus précieux que d'autres. Nous proposons donc d'enrichir le modèle de service d'un service de paquets "préemptables", par lequel certains des paquets au sein d'un flux pourraient être marqués comme préemptables. Lorsque le réseau court le danger de ne pas satisfaire certains de ses engagements quantitatifs de service, il pourrait exercer l'option de préemption de certains paquets et éliminer le paquet (et pas simplement de le retarder, car cela introduirait des problèmes de déclassement). En éliminant ces paquets préemptables, un routeur peut réduire les délais des paquets non préemptés.

De plus, on peut définir une classe de paquets qui ne soit pas soumise au contrôle d'admission. Dans le scénario décrit ci-dessus où les paquets préemptables ne sont abandonnés que lorsque des engagements quantitatifs de service sont en danger d'être violés, l'hypothèse est que les paquets préemptables vont presque toujours être livrés et donc qu'ils doivent être inclus dans la description de trafic utilisée dans le contrôle d'admission. Cependant, on peut étendre la préemptabilité au cas extrême des paquets "consommables" (le terme consommable est utilisé pour noter un degré extrême de préemptabilité) où l'hypothèse est qu'un grand nombre de ces paquets consommables pourrait n'être pas livré. On peut alors exclure les paquets consommables de la description de trafic utilisée dans le contrôle d'admission ; c'est-à-dire que les paquets ne sont pas considérés comme faisant partie du flux dans la perspective du contrôle d'admission, car il n'y a pas d'engagement à les livrer.

3.4 Retours sur l'utilisation

Une autre question importante dans le service est le modèle pour les retours sur l'utilisation, appelés aussi la "comptabilité", pour empêcher l'abus des ressources du réseau. Le service de liaison partagée décrit plus haut peut être utilisé pour fournir des limites imposées administrativement à l'utilisation. Cependant, un modèle plus libéral de l'accès au résseau sera nécessaire pour réduire la pression sur les utilisateurs pour la réservation des ressources du réseau. Ceci est un sujet de fortes controverses, et nous ne sommes pas prêts à en dire beaucoup plus sur lui pour l'instant.

3.5 Modèle de réservation

Le "modèle de réservation" décrit comment une application négocie un niveau de QS. Le modèle le plus simple est que l'application demande une certaine QS et que le réseau l'accorde ou la refuse. Souvent, la situation sera plus complexe. De nombreuses applications seront capables d'obtenir un service acceptable à partir d'une gamme de niveaux de QS, ou plus généralement, à partir de n'importe où au sein d'une certaine région de l'espace multi dimensionnel d'une spécification de flux.

Par exemple, plutôt que de simplement refuser la demande, le réseau peut accorder un niveau de ressource inférieur et informer l'application de la QS qu'il accorde en fait. Un exemple plus complexe est celui du modèle de réservation à "deux passes", Dans ce schéma, une flowspec "offerte" est diffusée le long de l'arborescence de distribution en diffusion groupée de chaque envoyeur Si à tous les receveurs Rj. Chaque routeur le long du chemin enregistre ces valeurs et peut-être les ajuste pour refléter les capacités disponibles. Les receveurs obtiennent ces offres, génèrent les flowspec de "demande" correspondantes, et les propagent en arrière le long des mêmes chemins vers les envoyeurs. À chaque nœud, une réconciliation locale doit être réalisée entre la flowspec offerte et celle demandée pour créer une réservation, et une flowspec de demande modifiée de façon appropriée est ainsi passée. Ce schéma à deux passes permet que des propriétes étendues comme le délai permis soient distribuées à travers les bonds le long du chemin [Tenet90], [ST2-90]. Il reste encore du travail à faire pour définir les détails, avec un niveau correspondant de complexité, qui sont nécessaires pour le modèle de réservation.

4. Mécanismes de contrôle de trafic

Nous allons d'abord revoir très brièvement les mécanismes possibles de contrôle du trafic. Puis au paragraphe 4.2 nous appliquons un sous ensemble de ces mécanismes pour prendre en charge les divers services que nous avons proposés.

4.1 Fonctions de base

Dans le chemin de transmission du paquet, il y a en fait un ensemble très limité d'actions que peut avoir un routeur. Pour un paquet particulier, un routeur doit choisir un chemin ; le routeur peut de plus le transmettre ou l'abandonner, et le routeur peut le réordonner par rapport aux autres paquets qui attendent de partir. Le routeur peut aussi conserver le paquet, même si la liaison est inactive. Ce sont les pierres angulaires à partir desquelles nous devons façonner le comportement désiré.

4.1.1 Programmation des paquets

La fonction de base de la programmation de paquets est de réarranger la file d'attente de sortie. De nombreux articles ont été écrits sur les façons possibles de gérer la file d'attente de sortie, et le comportement résultant. Peut-être que l'approche la plus simple est un schéma de priorité dans lequel les paquets sont rangés par ordre de priorité et les paquets de la plus forte priorité quittent toujours la file en premier. Ceci a pour effet de donner à certains paquets une préférence absolue sur les autres ; si il y a assez de paquets de la plus haute priorié, la classe de priorité inférieure peut être complètement privée d'envoi.

Un schéma de programmation de remplacement est le "round-robin" ou une de ses variantes, qui donne aux différentes classes de paquets l'accès à une part de la liaison. Une variante appelée mise en file d'attente à pondération équitable (WFQ, *Weighted Fair Queueing*) a montré qu'elle alloue la totalité de la bande passante d'une liaison selon les parts spécifiées.

Il y a des schémas plus complexes pour la gestion de file d'attente, dont la plupart impliquent d'observer les objectifs de service des paquets individuels, comme la limite de livraison, et d'oerdonner les paquets sur la base de ces critères.

4.1.2 Abandon de paquet

L'abandon contrôlé des paquets est aussi important que leur programmation.

Évidemment, un routeur doit abandonner des paquets quand ses mémoires tampons sont pleines. Ce fait ne détermine cependant pas quel paquet devrait être abandonné. Abandonner le paquet qui arrive, bien que simple, peut causer un comportement non souhaité.

Dans le contexte de l'Internet d'aujourd'hui, et le fonctionnement de TCP avec le service IP au mieux, l'abandon d'un paquet est lu par TCP comme un signal d'encombrement et l'amène à réduire sa charge sur le réseau. Donc, prélever un paquet à abandonner est la même chose que mettre une source au ralenti. Sans rentrer dans un algorithme particulier, cette simple relation suggère que des contrôles spécifiques de l'abandon devraient être mis en œuvre dans les routeurs pour améliorer le contrôle de l'encombrement.

Dans le contexte de services en temps réel, l'abandon se rapporte plus directement à la réalisation de la qualité de service désirée. Si une file d'attente se construit, l'abandon d'un paquet réduit le délai de tous les paquets derrière lui dans la file d'attente. La perte d'un peut contribuer au succès de beaucoup. Le problème pour la mise en œuvre est de déterminer quand les objectifs du service (la limite de délai) sont en danger d'être violés. On ne peut pas regarder la longueur d'une file d'attente comme une indication de la durée du stationnement des paquets dans la queue. Si il y a un schéma de priorité, les paquets de priorité inférieure peuvent être préemptés indéfiniment, de sorte que même une courte file d'attente peut contenir de très vieux paquets. Bien qu'on puisse en fait utiliser les horodatages pour mesurer les temps de rétention, la complexité peut en être inacceptable.

Certains schémas simples d'abandon, comme celui de combiner toutes les mémoires tampons en un seul réservoir global, et d'abandonner le paquet arrivant si le réservoir est plein, peut contrer les objectifs de service d'un schéma de programmation WFQ. Donc, abandon et programmation doivent être coordonnés.

4.1.3 Classement des paquets

L'exposé ci-dessus sur programmation et abandon suppose que le paquet a été classé dans un flux ou séquence de paquets qui devraient être traités d'une façon spécifique. Un préliminaire de cette sorte de traitement est la classification elle-même. Aujourd'hui, un routeur regarde l'adresse de destination et choisit un chemin. L'adresse de destination n'est pas suffisante pour choisir la classe de service que doit recevoir un paquet ; il y a besoin de plus d'informations.

Une approche serait d'abandonner le modèle du datagramme IP pour un modèle de circuit virtuel, dans lequel un circuit est établi avec des attributs de service spécifiques, et le paquet porte un identifiant de circuit. C'est l'approche de l'ATM ainsi que de protocoles comme ST-II [ST2-90]. Un autre modèle, moins hostile à IP, est de permettre au classeur de regarder plus de champs dans le paquet, comme les champs d'adresse de source, de numéro de protocole et d'accès. Donc, les flux de vidéo pourraient être reconnus par un champ particulier d'accès bien connu dans l'en-tête UDP, ou un flux particulier pourrait être reconnu en regardant les deux numéros d'accès de source et de destination. Il serait possible de regarder encore plus profondément dans les paquets, par exemple en vérifiant un champ dans la couche application pour choisir un sous ensemble d'un flux vidéo à codage hiérarchique.

La question de la mise en œuvre du classeur est une surcharge de complexité et de traitement. L'expérience actuelle suggère que la mise en œuvre soigneuse d'algorithmes efficaces peut conduire à une bonne classification des paquets IP. Ce résultat est très important, car il nous permet d'ajouter la prise en charge de la QS aux applications existantes, telles que Telnet, qui se fondent sur les en-têtes IP existants.

Une approche pour réduire la surcharge de la classification serait de fournir un champ "identifiant de flux" dans l'en-tête de paquet à la couche Internet. Cet identifiant de flux serait un outil qui pourrait être mis en antémémoire et utilisé pour court-circuiter la classification du paquet. Il y a un certain nombre de variantes à ce concept, et l'ingénierie est nécessaire pour choisir le meilleur concept.

4.1.4 Contrôle d'admission

Comme nous l'avons dit dans l'introduction, le service en temps réel dépend de l'établissement d'états dans le routeur et de prendre des engagements sur certaines classes de paquets. Pour s'assurer que ces engagements peuvent être tenus, il est nécessaire que les ressources soient demandées explicitement afin que la demande puisse être refusée si les ressources ne sont pas disponibles. La décision sur la disponibilité de la ressource est appelée contrôle d'admission.

Le contrôle d'admission exige que le routeur comprenne les demandes qui sont actuellement faites sur son compte. L'approche traditionnellement proposée est de rappeler les paramètres de service des demandes passées et de faire un calcul sur la base du plus mauvais cas de limites de chaque service. Une proposition récente, qui va vraisemblablement fournir une meilleure utilisation des liaisons, est de programmer le routeur de façon qu'il mesure l'utilisation réelle par les flux de paquets existants, et d'utiliser ces mesures d'informations comme base pour l'admission des nouveaux flux [JCSZ92]. Cette approche est sujette à un plus fort risque de surcharge, mais peut se révéler beaucoup plus efficace pour l'utilisation de la bande passante.

Noter qu'alors que le besoin de contrôle d'admission fait partie du modèle de service global, les détails de l'algorithme qui fontionne dans chaque routeur sont une affaire locale. Donc, les fabricants ont toute liberté pour développer et commercialiser en toute concurrence de meilleurs algorithmes de contrôle d'admission, qui conduisent à une meilleure charge de liaison avec moins de surcharge de service.

4.2 Application du mécanisme

Les divers outils décrits ci-dessus peuvent être combinés pour prendre en charge les services qui ont été exposés à la section 3.

- Limites de délai garanties
  Un résultat théorique exposé dans [Parekh92] montre que si le routeur met en œuvre une discipline de programmation WQF, et si la nature de la source de trafic peut être caractérisée (par exemple, si elle tient entre certaines limites telles qu'un baquet de jetons) il y aura alors une limite supérieure absolue au délai de réseau du trafic en question. Ce résultat simple et très puissant ne s'applique pas seulement à un commutateur, mais au réseau général des routeurs. Le résultat est constructif ; Parekh affiche le comportement d'une source qui conduit à la limite, et montre que ce comportement est le pire possible. Cela signifie que la limite qu'il calcule est la meilleure qui puisse être dans cette hypothèse.

- Partage de liaison
  Le même schéma WQF peut fournir le partage de liaison contrôlé. Les objectifs de service ne sont pas d'encadrer le délai, mais de limiter les parts de surcharge sur une liaison, tout en permettant tout mélange de trafic si il y a de la capacité inutilisée. Cette utilisation de WFQ est disponible dans les routeurs du commerce, et est utilisée pour séparer le trafic en classes sur la base de choses comme le type de protocole ou d'application. Par exemple, on peut allouer des parts distinctes à TCP, IPX et SNA, et on peut assurer que le trafic de contrôle du réseau obtient une part garantie de la liaison.

o Service prévisible en temps réel
Ce service est en fait plus subtil que le service garanti. Son objectif est de donner une limite de délai qui est, d'un côté, aussi basse que possible, et d'un autre côté, assez stable pour que le receveur puisse l'estimer. Le mécanisme WFQ conduit à une limite garantie, mais pas nécessairement une faible limite. En fait, le mélange du trafic en une seule file d'attente, plutôt que de le séparer comme avec WFQ, conduit à de plus faibles limites, pour autant que le mélange de trafic soit généralement similaire (par exemple, méler du trafic provenant de plusieurs codeurs vidéo a un sens ; mélanger de la vidéo et FTP n'en a pas).

Cela suggère qu'on a besoin d'un mécanisme à double détente, dans lequel la première étape séparé le trafic qui a des objectifs de service différents, et la seconde étape programme le trafic au sein de chaque classe de première étape afin de satisfaire ses objectifs de service.

4.3 Exemple : le schéma CSZ

Comme preuve du concept, un paquetage de code a été mis en œuvre qui réalise les services discutés ci-dessus. Il utilise en fait un certain nombre des outils de base, combinés d'une façon spécifique des besoins du service. On décrit son fonctionnement en termes généraux, pour suggérer comment peuvent être réalisés les services. On souligne qu'il y a d'autres moyens de construire un routeur qui satisfasse les mêmes besoins de service, et il y a en fait d'autres mises en œuvre qui sont utilisées aujourd'hui.

Au niveau supérieur, le code CSZ utilise WFQ comme mécanisme d'isolement pour séparer les flux garantis les uns des autres, ainsi que du reste du trafic. Le service garanti abtient la plus haute priorité quand et seulement quand il a besoin de l'accès pour tenir ses délais. WFQ apporte une garantie distincte pour chacun des flux garantis.

Le service prévisible et le service au mieux sont séparés par les priorités. Au sein de la classe du service prévisible, une autre priorité est utilisée pour fournir aux sous classes des limites de délai différentes. À l'intérieur de chaque sous classe prévisible, la simple mise en file d'attente FIFO est utilisée pour mélanger le trafic, qui semble produire un bon comportement de délai global. Cela fonctionne parce que l'algorithme de l'étape supérieure a séparé le trafic au mieux tel que celui de FTP.

Au sein de la classe au mieux, WFQ est utilisé pour fournir le partage de liaison. Comme il peut y avoir une exigence de parts incorporées, ce code WFQ peut être utilisé de façon récurrente. Il y a donc deux utilisations différentes de WFQ dans ce code, une pour séparer les classes garanties, et une pour séparer les partages de liaison. Elles sont similaires, mais diffèrent dans le détail.

Au sein de chaque liaison partagée de la classe au mieux, la priorité est utilisée pour permettre que plus de trafic sensible au temps précède un autre trafic élastique, par exemple, pour permettre à un trafic interactif de précéder des tranferts asynchrones en vrac.

Le code CSZ utilise donc à la fois WFQ et la priorité de façon alternée pour construire un mécanisme de prise en charge d'une gamme assez sophistiquée d'offres de service. Cet exposé est très bref, et n'aborde pas un certain nombre de questions significatives, telles que la façon dont le code CSZ s'adapte au trafic en temps réel dans les objectifs de partage de liaison. Mais les blocs de base de la construction sont très simples, et très puissants. En particulier, alors que la priorité a été proposée comme la clé des services en temps réel, WFQ peut être le plus général et le plus puissant de ces deux schémas. Il prend en charge, plutôt que la priorité, le service garanti et le partage de liaison.

5. Protocole d'établissement de réservation

Un certain nombre d'exigences doivent être satisfaites par un concept de protocole d'établissement de réservations. Il devrait être fondamentallement conçu pour un environnement de diffusion groupée, et il doit s'accommoder de besoins de service hétérogènes. Il doit donner un contrôle souple sur la façon dont les réservations peuvent être partagées le long des branches des arborescences de livraison de diffusion groupée. Il devrait être conçu autour des actions élémentaires d'ajout d'un envoyeur et/ou receveur à un ensemble existant, ou de leur suppression. Il doit être robuste et bien s'adapter à de grands groupes de diffusion groupée. Enfin, il doit prévoir la réservation de ressources à l'avance, et la préemption que cela implique. Le protcole d'établissement de réservation RSVP a été conçu pour satisfaire à ces exigences [RSVP93a], [RSVP93b]. Cette section fait un survol de la conception de RSVP.

5.1 Vue générale de RSVP

La Figure 2 montre la livraison de données multi-sources, multi-destinations pour une application répartie partagée

particulière. Les flèches indiquent les flux de données provenant des envoyeurs S1 et S2 aux receveurs R1, R2, et R3, et le nuage représente le maillage de distribution créé par le protocole d'acheminement de diffusion groupée. La distribution de diffusion groupée duplique chaque paquet de données d'un envoyeur Si, pour sa livraison à chaque receveur Rj. On traite la livraison en envoi individuel de S1 à R1 comme un cas particulier, et on appelle ce maillage de distribution de diffusion groupée une session. Une session se définit par l'adresse de destination IP commune (diffusion groupée) du ou des receveurs.

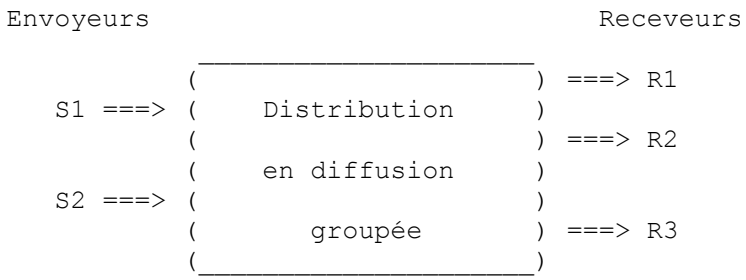

Figure 2 : Session de distribution en diffusion groupée

5.1.1 Spécification de flux et spécification de filtre

En général, une demande de réservation RSVP spécifie la quantité de ressources à réserver pour tous les paquets, ou un sous ensemble d'entre eux, d'une session particulière. La quantité de ressources est spécifiée par une spécification de flux (flowspec), alors que le sous ensemble de paquets qui va recevoir ces ressources est spécifié par une spécification de filtre. En supposant que le contrôle d'admission réussisse, la flowspec va être utilisée pour paramétrer une classe de ressources dans le progammeur de paquets, et la spécification de filtre sera instanciée dans le classeur de paquets pour transposer les paquets appropriés dans cette classe. Le sous ensemble de l'état de classeur qui sélectionne une classe particulière est appelé un "filtre" (de paquet) dans la documentation RSVP.

Les mécanismes du protocole RSVP fournissent une facilité très générale de création et d'entretien d'un état de réservation réparti à travers le maillage des chemins de livraison de diffusion groupée. Ces mécanismes traitent les flowspecs et les spécifications de filtre comme des données binaires presque opaques, les passant à la machine de contrôle du trafic locale pour leur interprétation. Bien sûr, le modèle de service présenté à une application doit spécifier comment coder les flowspecs et les spécifications de filtre.

5.1.2 Styles de réservation

RSVP offrre plusieurs "styles" de réservation différents, qui déterminent la manière dont les exigences de ressources de receveurs multiples sont agrégées dans les routeurs. Ces styles permettent que les ressources réservées satifassent plus efficacement les exigences de l'application. Actuellement il y a trois styles de réservation, "générique" *(wildcard)*, "filtre fixe", et "filtre dynamique". Une réservation générique utilise une spécification de filtre qui n'est pas spécifique de la source, de sorte que tous les paquets destinés à la destination associée (session) peuvent utiliser un réservoir commun de ressources réservées. Cela permet de faire une seule allocation de ressource sur tous les chemins de distribution pour le groupe. Le style de réservation générique est utile pour la prise en charge des conférences audio, où au plus un petit nombre de sources sont simultanément actives et peuvent partager l'allocation de ressources.

Les deux autres styles utilisent des spécifications de filtre qui sélectionnent des sources particulières. Un receveur peut désirer recevoir d'un ensemble de sources fixé, ou alors peut désirer que le réseau commute entre différentes sources, en changeant de façon dymamique sa ou ses spécifications de sources. Une réservation de style filtre fixe ne peut pas être changée pendant sa durée de vie sans réinvoquer le contrôle d'admission. Les réservations de filtre dynamiques permettent à un receveur de modifier son choix de sources à tout moment sans contrôle d'admission supplémentaire ; cependant, cela exige que des ressources suffisantes soient allouées pour faire face aux plus mauvais cas quand tous les receveurs vers l'aval prennent des entrées de différentes sources.

5.1.3 Initiation du receveur

Une question de conception importante est celle de savoir si l'envoyeur ou les receveurs devraient être chargés d'initier les réservations. Un envoyeur connait les qualités du flux de trafic qu'il peut envoyer, alors qu'un receveur sait ce qu'il veut (ou peut) recevoir. Peut-être que le choix le plus évident est de laisser l'envoyeur initier la réservation. Cependant, cela s'adapte assez mal pour les grandes arborescences de livraison de diffusion groupée dynamiques et pour les receveurs hétérogènes.

Ces deux problèmes d'adaptation sont résolus en rendant le receveur responsable de l'initiation d'une réservation.

L'initiation par le receveur traite facilement la question des receveurs hétérogènes, chaque receveur demandant simplement une réservation appropriée pour lui-même, et toutes les différences parmi les réservations provenant des différents receveurs sont résolues ("fusionnées") au sein du réseau par RSVP. L'initiation par le receceur est aussi cohérente avec la diffusion groupée IP, dans laquelle un groupe de diffusion groupée est créé implicitement par les receveurs qui s'y joignent.

Bien que la réservation à l'initiative du receveur soit le choix naturel pour les sessions de diffusion groupée, la justification de l'initiation par le receveur peut apparaître plus faible pour les sessions en envoi individuel où l'envoyeur peut être l'initateur logique de la session. Cependant, on prévoit que toute application en temps réel aura son protocole de signalisation et de contrôle de niveau supérieur, et que ce protocole pourra être utilisé pour signaler au receveur d'initier une réservation (et peut-être indiquer la flowspec à utiliser). Pour la simplicité et l'économie, un protocole d'établissement devrait ne prendre en charge qu'une seule direction d'initiation, et une initiation par le receveur nous apparaît comme étant le clair vainqueur.

RSVP utilise l'initiation des réservations par le receveur [RSVP93b]. Un receveur est supposé apprendre les flowspecs offertes par l'envoyeur par un mécanisme de niveau supérieur ("hors bande") et générer ensuite la propre flowspec qu'il désire et la propager vers les envoyeurs, en faisant les réservations dans chaque routeur le long du chemin.

5.1.4 État conditionnel

Il y a deux styles différents possibles pour les protocoles d'établissement de réservations, l'approche "d'état fixe" (HS, *Hard State*) (aussi appelée "orienté connexion") et celle de l'état conditionnel (SS, *soft state*) (aussi appelée "sans connexion"). Dans les deux approches, la distribution en diffusion groupée est effectuée en utilisant un état spécifique du flux dans chaque routeur le long du chemin. Dans l'approche HS, cet état est créé et supprimé d'une façon complètement déterministe par la coopération entre les routeurs. Une fois qu'un hôte demande une session, le "réseau" prend la responsabilité de créer, et ensuite de supprimer, l'état nécessaire. ST-II est un exemple de l'approche HS [ST2-90]. Dans la mesure où la gestion de l'état de session HS est complètement déterministe, le protocole d'établissement HS doit être fiable, avec accusé de réception et retransmissions. Afin de réaliser le nettoyage déterministe des états après une défaillance, il doit y avoir un mécanisme pour détecter les défaillances, c'est-à-dire, un protocole de "montage/démontage". Le routeur en amont (vers la source) d'une défaillance prend la responsabilité de reconstruire l'état nécessaire sur le ou les routeurs le long d'un chemin de remplacement.

RSVP adopte l'approche SS, qui regarde l'état de réservation tel que le donnent les informations en antémémoire qui sont installées et périodiquement rafraîchies par les hôtes d'extrémité. Un état inutilisé est périmé par les routeurs. Si le chemin change, les messages de rafraîchissement installent automatiquement l'état nécessaire le long du nouveau chemin. L'approche SS a été choisie pour obtenir la simplicité et la robustesse qui ont été démontrées par les protocoles sans connexion tels que IP [Clark88].

5.2 Acheminement et réservations

Il y a une interaction fondamentale entre établissement de réservations de ressources et acheminement, car la réservation exige l'installation de l'état du flux le long du chemin des paquets de données. Si et quand un chemin change, il doit y avoir un mécanisme pour établir une réservation le long du nouveau chemin.

Certains ont suggéré que l'établissement des réservations exige nécessairement l'établissement d'un chemin, c'est-à-dire impose un circuit virtuel de couche internet. Notre but est cependant simplement d'étendre l'architecture de l'Internet, et non de la remplacer. La couche internet sans connexion fondamentale [Clark88] a été un grand succès et nous souhaitons la garder comme fondement de l'architecture. Nous proposons plutôt de modifier un peu du mécanisme de pure transmission de datagramme de l'Internet présent pour s'accomoder de "IS".

Il y a quatre questions d'acheminement qui se présentent à un protocole d'établissement de réservations tel que RSVP.

1. Trouver un chemin qui prenne en charge la réservation de ressources.
   C'est simplement l'acheminement par "type de service", une facilité qui est déjà disponible dans certains protocoles d'acheminement modernes.

2. Trouver un chemin qui ait suffisamment de capacités non réservées pour un nouveau flux.
   Les premières expériences sur l'ARPANET ont montré qu'il est difficile de faire de l'acheminement dynamique en fonction de la charge paquet par paquet sans problèmes d'instabilité. Cependant, l'instabilité ne devrait pas être un problème si un acheminement selon la charge est effectué seulement au moment de l'établissement de la réservation.

   Deux approches différentes peuvent être suivies pour trouver un chemin avec des capacités suffisantes. On pourrait modifier le ou les protocoles d'acheminement et les interfacer avec le mécanisme de contrôle du trafic, de sorte que le

calcul du chemin puisse considérer la charge moyenne récente. Autrement, le protocole d'acheminement pourrait être conçu (ou redessiné) pour fournir plusieurs chemins de rechange, et l'établissement des réservations pourrait être tenté le long de chacun à son tour.

3. S'adapter à une défaillance du chemin.
   Lorsque un nœud ou une liaison connaît une défaillance, l'acheminement adaptatif trouve un chemin de remplacement. Les messages périodiques de rafraîchissement de RSVP vont automatiquement demander une réservation le long du nouveau chemin. Bien sûr, cette réservation peut échouer à cause de l'insuffisance de capacités disponibles sur ce nouveau chemin. Ceci est un problème d'approvisionnement et d'ingénierie du réseau qui ne peut pas être résolu par les protocoles d'acheminement ou d'établissement.

   Il y a un problème d'à propos pour l'établissement de l'état de réservation sur le nouveau chemin. Le mécanisme de robustesse de bout en bout des rafraîchissements est limité en fréquence par la redondance qui peut causer un trou dans le service en temps réel lorsque il y a une rupture de l'ancien chemin et qu'un nouveau est choisi. Il devrait être possible de revoir RSVP pour compléter le mécanisme global de rafraîchissement avec un mécanisme de réparation local, en utilisant des indications sur les changements de chemin à partir des mécanismes d'acheminement.

4. S'adapter à un changement de chemin (sans défaillance).
   Des changements de chemin peuvent survenir même sans défaillance du chemin affecté. Bien que RSVP puisse utiliser les mêmes techniques de réparation que décrites en (3), ce cas soulève un problème de robustesse des garanties de QS. Si il devrait arriver que le contrôle d'admission soit défaillant sur le nouveau chemin, l'usager verrait une dégradation de service non nécessaire et capricieuse, car le chemin d'orgine fonctionnerait toujours.

   Pour éviter ce problème, un mécanisme appelé "épinglage de chemin" *(route pinning)* a été suggéré. Cela modifierait la mise en œuvre du protocole d'acheminement et l'interface avec le classeur, de sorte que les chemins associés aux réservations de ressources seraient "épinglées". Le protocole d'acheminement ne changerait pas un chemin épinglé si il était encore viable.

Il serait éventuellement possible de mettre ensemble les problèmes d'acheminement et d'établissement de réservations, mais nous n'en comprenons pas encore suffisament pour le faire. De plus, le protocole de réservation doit coexister avec un certain nombre de différents protocoles d'acheminement utilisés dans l'Internet. Donc, RSVP est actuellement conçu pour fonctionner avec tout protocole d'acheminement de la génération actuelle sans modification. Ceci est un compromis à court terme, qui peut résulter en une défaillance occasionnelle à créer la meilleure session en temps réel, sinon tontes, ou une dégradation occasionnelle du service due à un changement de chemin. On s'attend à ce que les futures générations de protocoles d'acheminement suppriment ce compromis, en incluant des outils et mécanismes qui, en conjonction avec RSVP, résoudront les problèmes énumérés de (1) à (4). Ils prendront en charge l'épinglage de chemin, les notifications de RSVP pour déclancher les réparations locales, et la sélection des chemins avec prise en charge de "IS" et la capacité adéquate.

Le dernier problème en relation avec l'acheminement est fourni par les hôtes mobiles. Notre conjecture est que la mobilité n'est pas essentiellement différente des autres changements de chemin, de sorte que le mécanisme suggéré en (3) et (4) suffira. D'autres études et expérimentations sont nécessaires pour prouver ou infirmer cette conjecture.

6. Remerciements

De nombreux chercheurs de l'Internet ont contribué au travail décrit dans ce mémoire. Nous voulons remercier particulièrement Steve Casner, Steve Deering, Deborah Estrin, Sally Floyd, Shai Herzog, Van Jacobson, Sugih Jamin, Craig Partridge, John Wroclawski, et Lixia Zhang. Cette approche des services intégrés dans l'Internet a été discutée initialement et organisée dans le groupe de recherche de bout en bout de l'équipe de recherche de l'Internet, et nous sommes reconnaisants à tous les membres de ce groupe pour les discussions intéressates (et parfois animées) qui y ont eu lieu.

Références

[CerfKahn74] Cerf, V., and R. Kahn, "A Protocol for Packet Network Intercommunication", IEEE Trans on Comm., Vol. Com-22, n° 5, mai 1974.

[Clark88] Clark, D., "The Design Philosophy of the DARPA Internet Protocols", ACM SIGCOMM '88, août 1988.

[CSZ92] Clark, D., Shenker, S., and L. Zhang, "Supporting Real-Time Applications in an Integrated Services Packet Network: Architecture and Mechanisms", Proc. SIGCOMM '92, Baltimore, MD, août 1992.

[DKS89] Demers, A., Keshav, S., and S. Shenker, "Analysis and Simulation of a Fair Queueing Algorithm", Journal of Internetworking: Research and Experience, 1, pp. 3-26, 1990. Aussi dans Proc. ACM SIGCOMM '89, pp 3-12.

[SCZ93a] Shenker, S., Clark, D., and L. Zhang, "A Scheduling Service Model and a Scheduling Architecture for an Integrated Services Packet Network", soumis à ACM/IEEE Trans. on Networking.

[SCZ93b] Shenker, S., Clark, D., and L. Zhang, "A Service Model for the Integrated Services Internet", Travail en cours, octobre 1993.

[Floyd92] Floyd, S., "Issues in Flexible Resource Management for Datagram Networks", Minutes du 3e Atelier sur les réseaux à très grande vitesse, mars 1992.

[Jacobson91] Jacobson, V., "Communication privée", 1991.

[JCSZ92] Jamin, S., Shenker, S., Zhang, L., and D. Clark, "An Admission Control Algorithm for Predictive Real-Time Service", résumé développé dans Proc. Third International Workshop on Network and Operating System Support for Digital Audio and Video, San Diego, CA, novembre 1992, pp. 73-91.

[Parekh92] Parekh, A., "A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks", Technical Report LIDS-TR-2089, Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, 1992.

[Partridge92] C. Partridge, "Proposition de spécification de flux", RFC1363, septembre 1992. *(Info.)*

[RSVP93a] Zhang, L., Deering, S., Estrin, D., Shenker, S., and D. Zappala, "RSVP: A New Resource ReSerVation Protocol", Accepté pour publication dans IEEE Network, 1993.

[RSVP93b] R. Braden, éd., L. Zhang, S. Berson, S. Herzog, S. Jamin, "Protocole de réservation de ressource (RSVP) -- version 1, spécification fonctionnelle", RFC2205, septembre 1997. (*MàJ par* RFC2750, RFC3936, RFC4495) *(P.S.)*

[ST2-90] C. Topolcic, éditeur, "Protocole expérimental de flux Internet, version 2 (ST-II)", RFC1190, octobre 1990. *(obsolète, voir la* RFC1819*)*

[Tenet90] Ferrari, D., and D. Verma, "A Scheme for Real-Time Channel Establishment in Wide-Area Networks", IEEE JSAC, Vol. 8, n° 3, pp 368-379, avril 1990.

Considérations pour la sécurité

Comme noté au paragraphe 2.1, la capacité à réserver des ressources va créer une exigence d'authentification, autant des utilisateurs qui demandent des garanties de ressource que des paquets qui prétendent avoir le droit d'utiliser ces garanties. Ces questions d'authentification ne sont pas autrement traitées dans le présent mémoire, mais feront l'objet d'études complémentaires.

Adresse des auteurs

Bob Braden
USC Information Sciences Institute
4676 Admiralty Way
Marina del Rey, CA 90292
téléphone : (310) 822-1511
mél : Braden@ISI.EDU

David Clark
MIT Laboratory for Computer Science
545 Technology Square
Cambridge, MA 02139-1986
télphone : (617) 253-6003
mél : ddc@LCS.MIT.EDU

Scott Shenker
Xerox Palo Alto Research Center
3333 Coyote Hill Road
Palo Alto, CA 94304
téléphone : (415) 812-4840
mél : Shenker@PARC.XEROX.COM