

Chapitre 2

Notion de voie de communication

2.1. Introduction

L'objet de ce chapitre est de présenter les concepts et les services fondamentaux des réseaux. Pour cela, nous allons utiliser une approche intuitive et pragmatique grâce à un service de communication utilisé par tous, le service de la Poste. Nous allons y trouver à peu près tout ce qui fait un réseau de transmission de données. Nous regarderons d'autres services tel le téléphone pour débusquer les différences avec le service postal. Nous recommandons lors d'une difficulté de compréhension de faire l'analogie avec le service postal. A la suite de cet exemple, les principaux concepts des réseaux sont définis.

2.1.1. Service à datagramme

Historiquement, le premier service de transport d'information est celui offert par la Poste. La Poste a des clients : toute personne qui sait écrire. Chaque client connaît le mode d'utilisation du service postal. Le service peut se résumer ainsi :

- le client poste une lettre dans une interface publique appelée « boîte à lettres »,
- le client fournit avec chaque lettre le nom et l'adresse du destinataire,
- chaque individu possède une adresse qui lui est propre. Chaque client met à la disposition de la Poste une boîte aux lettres pour recevoir son courrier¹.
- la lettre est éventuellement remise ultérieurement au destinataire dans une interface privée appelée aussi « boîte à lettres »,
- le contenu de l'information reste inconnu du prestataire du service postal (i.e. la Poste),

1. Certains services postaux (la Poste du Cameroun par exemple) louent à leurs clients des boîtes postales pour recevoir leur courrier. La boîte postale est située au bureau de poste qui en reste le propriétaire. L'adresse est alors le numéro de boîte postale du bureau de poste de la ville.

— les moyens par lesquels la Poste réalise le transport de la lettre sont en général inconnus des utilisateurs. Ces moyens sont variés (bicyclette, avion, train, camion...).

La Poste offre donc à chacun de ses clients une voie de communication avec tout autre client ayant une adresse et une boîte à lettres. Quelles sont les propriétés de ce service pour le client ?

- il doit construire une lettre pour chaque information à échanger ;
- la quantité d'information qui peut être mise dans une lettre est limitée (par le poids dans le cas de la Poste). Chaque lettre transporte soit l'intégralité du document, soit un fragment d'un document plus large ;
- il doit connaître l'adresse de son correspondant et la mettre sur chaque lettre. Il peut aussi inscrire son adresse sur l'enveloppe, mais cela n'est pas obligatoire et ne sera utilisé par la Poste qu'en cas de non-distribution (erreur) ;
- il peut envoyer une lettre quand bon lui semble sans aucun avertissement à la Poste. Il n'a pas besoin de s'abonner pour utiliser le service de la Poste. Il lui suffit de trouver une boîte à lettres publique ;
- il n'est pas nécessaire que le destinataire soit présent au moment de l'envoi, pendant le transfert de la lettre ni même à l'arrivée de la lettre. Il n'est pas nécessaire de demander l'accord du destinataire pour lui envoyer une lettre ;
- bien que la Poste française ne couvre pas toute la planète, le client peut envoyer des lettres dans le monde entier (Japon, Chine, Pays arabes...).

Nous qualifierons le service postal de service à datagramme ou lettregramme. Les caractéristiques ci-dessus sont typiques de cette famille de services.

Notez que, bien que la Poste fasse de son mieux :

- une lettre peut ne pas arriver (perte par la Poste, destruction pendant son transport à destination, adresse inexistante ou erronée, défaut de timbrage). Dans la plupart des cas, l'émetteur ne sera pas informé de la perte ;
- une lettre peut subir un délai d'acheminement variable (1, 2, 3 jours ou plus) ;
- deux lettres postées consécutivement peuvent arriver dans un ordre différent du fait de ce délai d'acheminement. On dira que le service postal n'assure pas le séquençement du courrier.

Le service postal ne garantit pas que toutes les lettres arriveront. Il ne garantit pas non plus le délai d'acheminement de chaque lettre. Une lettre peut être arbitrairement retardée. Il n'existe pas de service de transport d'information totalement fiable, et tout service tolère une proportion non nulle (peut-être infime) de pertes.

2.1.2. Protocole du cuisinier

Nous allons prendre pour illustrer les conséquences des propriétés de ce service un exemple simple. Imaginons un gâte-sauce (marmiton ou apprenti cuisinier) dans la bonne ville de Brest et un chef cuisinier à Rennes. Le chef cuisinier donne ses ordres à son gâte-sauce par une succession de lettres. Chaque lettre contient une partie de la recette de cuisine à exécuter. Notre chef cuisinier veut faire exécuter à son gâte-sauce

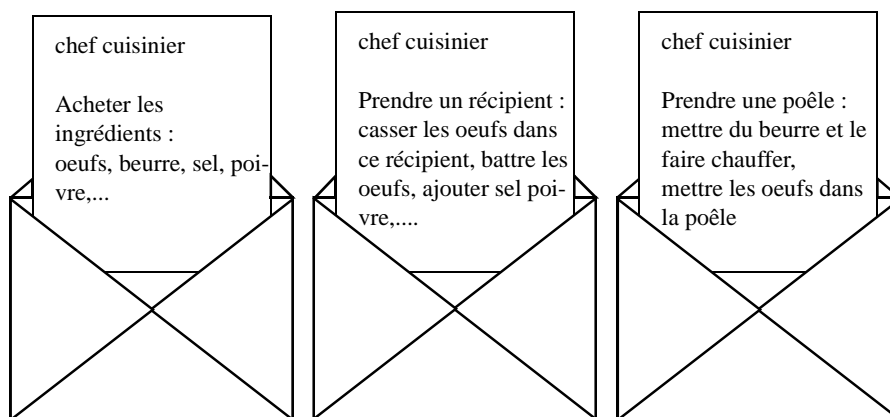


Figure 2.1. Les trois lettres envoyées par le chef cuisinier du restaurant "La Mère Poulard"

une omelette. Le gâte-sauce exécute sans état d'âme les ordres qui sont écrits dans chaque lettre au fur et à mesure que l'on lui remet les lettres. Les trois lettres écrites par le chef cuisinier sont décrites sur la figure 2.1. Bien sûr chaque lettre est mise dans une enveloppe dûment timbrée avec l'adresse du restaurant et le nom du gâte-sauce. La Poste ne s'intéresse qu'aux informations qui sont sur l'enveloppe et en aucun cas au contenu de la lettre. Les informations qui sont sur l'enveloppe n'ont aucun intérêt pour notre application de cuisine. Elles font partie des informations de service nécessaires à la Poste pour effectuer sa fonction d'acheminement du courrier.

Le chef cuisinier poste la première lettre lundi, la seconde mardi et la troisième mercredi. Si les trois lettres arrivent respectivement mardi, mercredi et jeudi, tout va bien et notre recette de cuisine sera exécutée correctement. Mais le service postal ne garantit pas cette propriété. Les lettres peuvent arriver de manière différente (cf. figure 2.2.) : imaginez ce qui va arriver si les lettres arrivent ainsi, mardi la première, jeudi la troisième et vendredi la seconde. Notre gâte-sauce exécute scrupuleusement les ordres reçus. La recette de cette omelette en est sensiblement modifiée. Une erreur ne conduit pas toujours à créer la tarte Tatin !

Le problème vient du fait que :

- le service postal traite chaque courrier indépendamment des autres et ne garantit pas le délai d'acheminement. Ainsi, la deuxième lettre a subi un délai d'acheminement de trois jours, ce qui est rare mais pas impossible. Le résultat de ce délai exceptionnel est que la séquence (suite) de lettres reçues est différente de la séquence émise. L'ensemble des lettres est bien reçu mais dans un ordre différent. Nous parlerons de déséquence ;

- le gâte-sauce exécute immédiatement les actions à la réception de la lettre. Plusieurs actions sont exécutées en parallèle. La Poste achemine une lettre. Le gâte-sauce exécute les ordres de la dernière lettre reçue. Le parallélisme, l'exécution

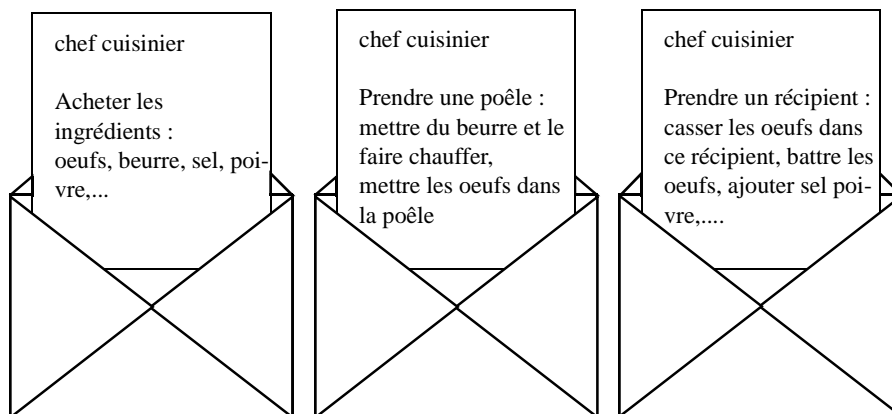


Figure 2.2. Lettres reçues

simultanée et concurrente des actions du chef cuisinier, du gâte-sauce et de la Poste, est une caractéristique de tout fonctionnement en réseau.

Il faut donc imaginer une solution pour éviter cette erreur de séquençage dans l'exécution de la recette. La solution va être construite sur un ensemble de règles, séquences d'actions et/ou conventions établies préalablement à tout échange entre chef cuisinier et gâte-sauce. On appelle protocole¹ cet ensemble de règles.

2.1.3. Spécification informelle du protocole

Notre protocole sera le suivant : chaque lettre aura en en-tête l'annotation suivante, i/j , qui signifie que la lettre reçue est la $i^{\text{ème}}$ parmi j lettres (cf. figure 2.3.). Le chef cuisinier ajoute cette information pour numéroté ses lettres. Pour un protocole, on doit spécifier les actions à faire dans une situation particulière. Voici ci-dessous une spécification informelle d'un protocole.

2.1.3.1. Protocole de base

Le gâte-sauce interprète cette information de la manière suivante : si toutes les lettres de numéro inférieur à i sont reçues et exécutées, alors le contenu de cette lettre peut être exécuté. Si une lettre de numéro inférieur à i n'est pas encore arrivée, alors le gâte-sauce attend cette lettre et conserve la lettre reçue dans un classeur dans l'ordre des numéros. Quand une lettre de numéro i arrive alors que des lettres de numéro $i+1$, $i+2...$ sont déjà arrivées, le gâte-sauce exécute les ordres de la lettre i puis $i+1$, $i+2$,

1. Définition du Petit Robert : « Recueil de formules en usage pour les actes publics, la correspondance officielle. Recueil de règles à observer en matière d'étiquette, de préséance, dans les cérémonies et les relations officielles ».

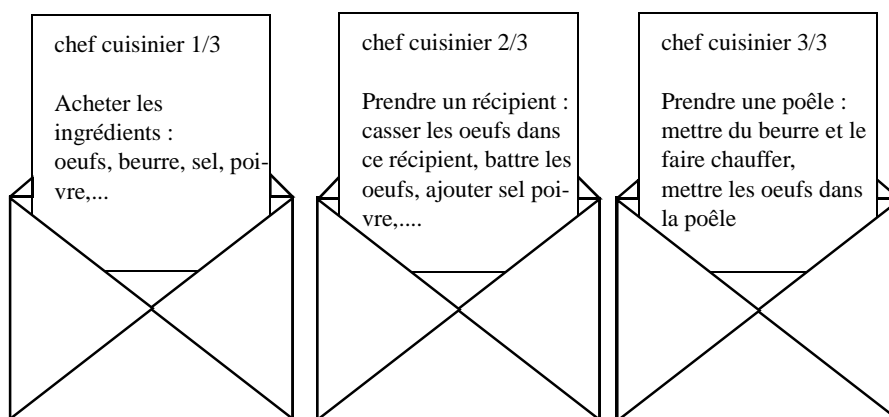


Figure 2.3. Les mêmes lettres avec l'information de numérotation du protocole

etc. La figure 2.3. montre où est placée cette information de protocole. On notera que cette information est indépendante de la recette à exécuter. Il s'agit d'une information exclusivement dédiée à la correction des déséquences potentiels. Nous appellerons enveloppe du protocole « gâte-sauce - chef cuisinier » cette information. Elle est en effet destinée à une fonction nouvelle, la fonction de correction des déséquences. Cette information supplémentaire surcharge la lettre.

On pourrait souhaiter que ni le gâte-sauce ni le chef cuisinier ne connaissent ce protocole. Ce protocole n'a rien à voir avec leur métier et pourrait être utile dans de nombreuses autres situations. On dira que ce mécanisme est générique. Pour cela il faut faire intervenir deux secrétariats (un de chaque côté) qui exécutent ce protocole. Le secrétariat du chef cuisinier prend les lettres et ajoute les informations de numérotation en fonction de l'ordre de soumission. Le secrétariat du gâte-sauce ouvre les lettres et exécute le protocole décrit précédemment. Comme il conserve les lettres hors séquence jusqu'à réception de la lettre manquante, il ne délivrera que les lettres en séquence. Le gâte-sauce n'aura plus à se préoccuper d'attendre une lettre manquante, pas plus que le chef cuisinier n'a à se préoccuper de numéroter ses lettres.

Pour nos deux cuisiniers, le service rendu par le secrétariat est toujours un service de communication, une voie de communication. Mais ce service a une propriété nouvelle que n'a pas le service de la Poste utilisé. Il garantit que les lettres arrivent dans l'ordre de soumission. Le but d'un protocole de communication est généralement d'améliorer le service de communication pour le rendre plus apte à rendre les services désirés par les utilisateurs.

2.1.3.2. Evolutions possibles et problèmes associés à ce protocole de base

Dans cet exemple nous pouvons enrichir le service du secrétariat du chef cuisinier de la manière suivante. Le chef cuisinier dicte toute la recette sans se préoccuper de la taille de la lettre. Le secrétariat découpe la recette en autant de lettres (fragments de

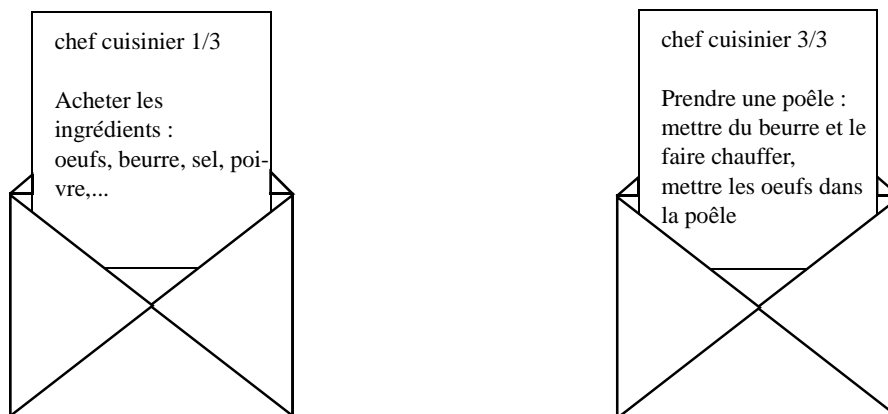


Figure 2.4. Une lettre est perdue que va-t-il se passer ?

recettes) que nécessaire pour pouvoir être transmises par le service postal et assure la numérotation des lettres. Nous avons ainsi simplifié la vie du chef cuisinier, qui n'a plus à connaître les contraintes du service postal. Cela ne veut pas dire qu'il n'a pas à connaître les contraintes du service de communication qu'il utilise. Ainsi, ce service peut entraîner un délai plus long entre la réception de deux lettres. Le chef cuisinier aura peut-être intérêt à ajouter des instructions ou à former préalablement son gâte-sauce pour mettre au frais ses produits. Mais cela relève exclusivement d'un protocole « cuisine ».

Notez aussi que ce protocole permet au secrétariat d'utiliser le service de la Poste de manière plus efficace. En effet, il n'est plus utile d'espacer l'émission des lettres pour espérer une arrivée dans le bon ordre. Il est possible de jeter les trois lettres le même jour dans la boîte tout en ayant la garantie que le secrétariat à l'arrivée les délivrera dans le bon ordre au gâte-sauce.

Le secrétariat est une fonction intermédiaire construite explicitement pour créer un nouveau service de communication plus adapté aux besoins des utilisateurs. En réseau on appelle couche de communication cette notion de service intermédiaire. La couche (fonction) secrétariat utilise le service de la Poste pour construire son propre service de communication. Le secrétariat apporte une valeur ajoutée au service de la poste. Néanmoins, le service rendu reste ici un service de communication. Cela n'est pas exclusivement réservé à la transmission d'informations : si notre chef cuisinier et le gâte-sauce ne parlent pas la même langue, un secrétariat bilingue pourrait assurer un service de traduction. Il pourrait aussi assurer un service d'archives, d'estampillage des courriers...

Ce protocole est-il suffisant pour parer aux inconvénients du service postal ? Que va-t-il se passer si une lettre se perd et n'arrive jamais (figure 2.4.) ?

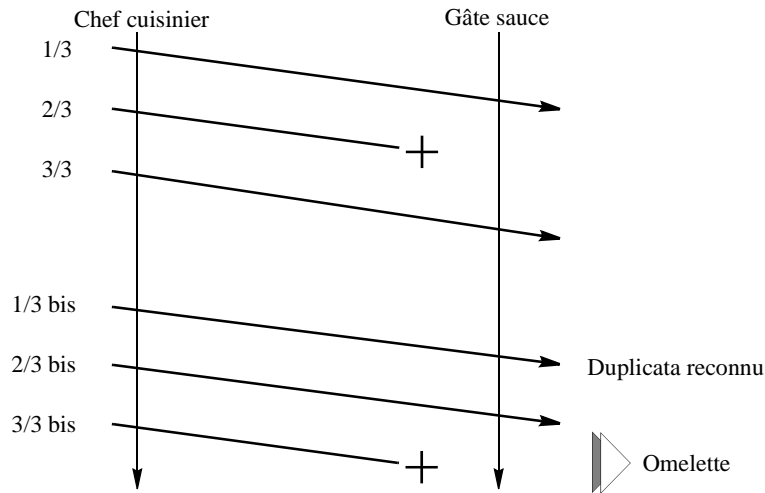


Figure 2.5. Chronogramme ou diagramme temporel d'un protocole récupérant les pertes de messages par répétition systématique

2.1.4. Correction d'une perte

Revenons au service de base de la Poste et à notre premier protocole à numérotation. Supposons que la lettre 2 n'arrive jamais. Notre gâte-sauce achète les oeufs quand on lui remet la lettre 1. Puis il attend à l'infini la lettre 2. Bien évidemment, la lettre 3 sera conservée mais ne pourra, conformément au protocole décrit, être utilisée. Les produits achetés ne pourront être utilisés impunément après une trop longue attente. Il faut donc spécifier un protocole pour traiter ces pertes. Nous allons imposer à ce protocole de travailler uniquement à l'aide du service postal. Plusieurs possibilités sont envisageables :

- le chef cuisinier envoie systématiquement plusieurs copies de toutes ses lettres. Cela coûte cher, mais réduit la probabilité que toutes les lettres se perdent. Il faudra toutefois ajouter une information qui identifie de manière unique la recette afin que le gâte-sauce n'exécute pas plusieurs fois la recette (la figure 2.5. montre la séquence temporelle, appelée diagramme temporel ou chronogramme des échanges, de ce protocole). Cette information lui permettra de jeter les copies de lettres déjà reçues ;

- le gâte-sauce attend pendant quelques jours, puis, s'il ne reçoit rien, envoie une lettre au chef cuisinier pour lui demander une copie de la lettre manquante (cf. figure 2.6.). Il faut que le chef cuisinier garde une copie de ses courriers pour pouvoir en renvoyer éventuellement une ;

- le gâte-sauce envoie une lettre de confirmation de bonne réception (qui peut être aussi appelée accusé de réception) pour toutes les lettres qu'il reçoit (cf. figure 2.8.). Le chef cuisinier renvoie une copie des lettres pour lesquelles il n'a pas reçu de confirmation de bonne réception dans un certain délai.

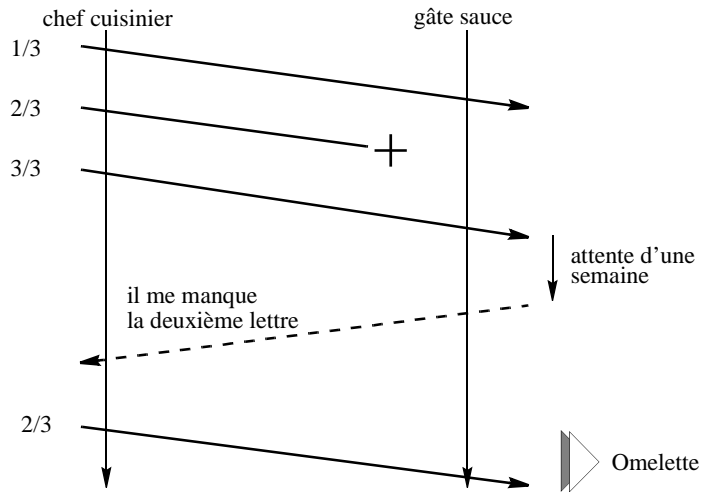


Figure 2.6. Chronogramme d'un protocole récupérant les pertes de messages par répétition sélective

— d'autres solutions sont envisageables. Par exemple, le chef cuisinier envoie périodiquement un message contenant uniquement la liste des lettres qu'il a

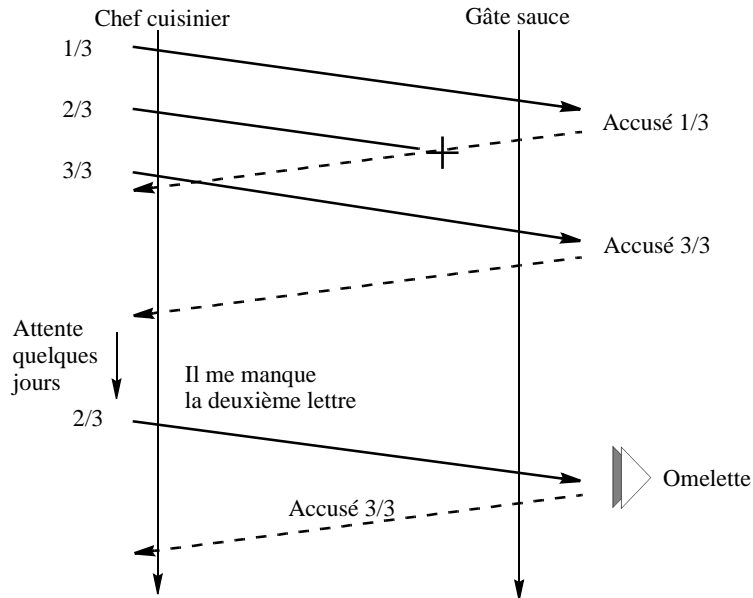


Figure 2.7. Chronogramme ou diagramme temporel d'un protocole récupérant les pertes de messages par envoi systématique d'accusé de réception

précédemment envoyées. Ce message ne contient aucune recette. Le gâte-sauce répond en envoyant la liste des lettres qu'il a reçues et donc les recettes qu'il a pu exécuter (cf. figure 2.8.). Sur la base de cette réponse, le chef cuisinier peut renvoyer les lettres non encore reçues ;

— etc.

Ces quelques exemples ont pour but de montrer qu'il existe une très grande variété de protocoles possible.

On appelle acquittement négatif ou positif les lettres envoyées en retour. Elles vont permettre de faire des retransmissions quand cela paraît nécessaire. On notera qu'en cas de perte seule la première solution permet de garantir que la recette sera exécutée sans trop de retard. Les deux autres solutions retardent l'exécution de la recette du temps nécessaire à décider d'une retransmission. Nous ne décrirons pas plus en détail ces protocoles dans ce chapitre. Nous y reviendrons dans les chapitres ultérieurs.

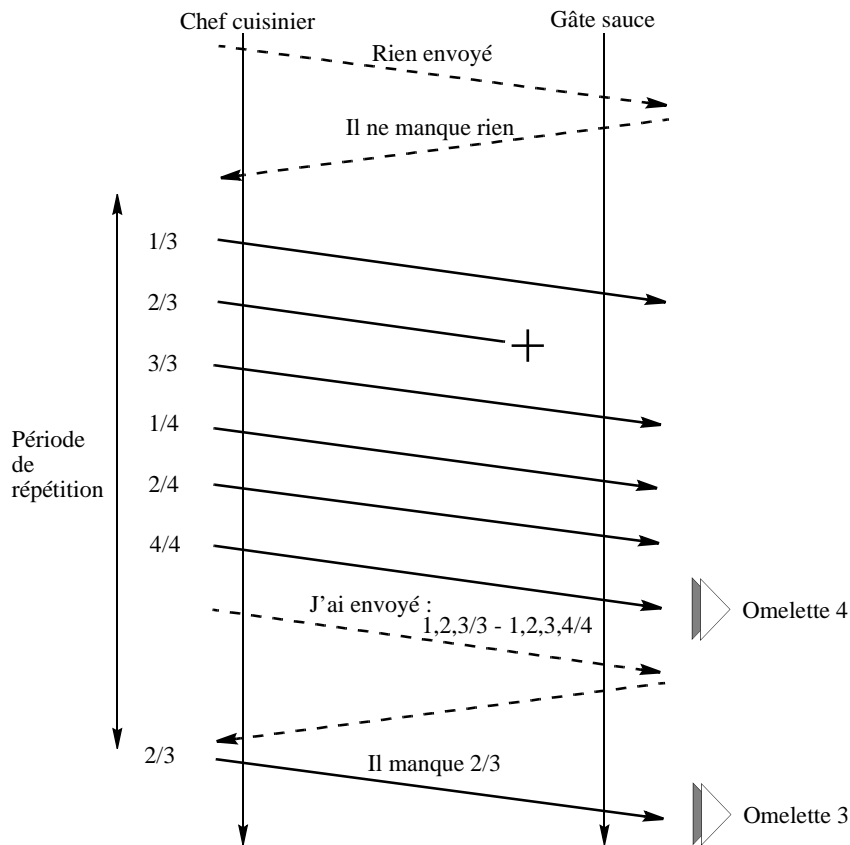


Figure 2.8. Chronogramme d'un protocole récupérant les pertes de message par envoi périodique de messages donnant la liste des échanges

chef cuisinier 1/3 Acheter les ingrédients : oeufs, beurre, sel, poivre,...	chef cuisinier 3/3 Prendre une poêle : mettre du beurre et le faire chauffer, mettre les oeufs dans la poêle ...	chef cuisinier 2/3 Prendre un récipient : casser les oeufs dans ce récipient, battre les oeufs, ajouter sel poi- vre,....	chef cuisinier 2/3 Prendre un récipient : casser les oeufs dans ce récipient, battre les oeufs, ajouter sel poi- vre,....
--	--	--	--

Figure 2.9. Une lettre est répétée. Que va-t-il se passer ?

Ces protocoles sont loin de résoudre tous les problèmes. En effet, il faut éviter que la création de copies (les messages retransmis, cf. figure 2.9.) ne bloque le gâte-sauce ou le service. Imaginez le cas où le courrier n'est pas perdu mais retardé excessivement longtemps (figure 2.9.). Ce courrier va arriver ainsi que sa copie. Le gâte-sauce va croire qu'il faut exécuter deux fois la recette. Il faut donc ajouter un mécanisme pour reconnaître les lettres relatives à une même recette, comme le montre la figure 2.10.

Etablir des informations qui vont permettre de corriger les erreurs doit être fait préalablement aux échanges de données. Il faut établir un contexte de communication. Le contexte est un ensemble d'informations qui sont gérées par les deux partenaires – dans notre exemple, le gâte-sauce et le chef cuisinier – pendant leur communication.

Dans le protocole avec secrétariat, le contexte est géré par les secrétaires. On dit qu'un service qui doit établir un tel contexte est un service sur connexion, car les lettres et les partenaires (gâte-sauce et chef cuisinier) sont associés entre eux grâce aux informations de contexte et aux informations de gestion du protocole mises dans les lettres (ex. *ij*). L'association est faite préalablement à l'échange de données.

2.1.4.1. Notion de flot ou flux

Cette application exhibe un flot de documents discrets, c'est-à-dire où chaque document est logiquement indépendant des autres. On parlera de flot discret ou de messages. On a vu dans cette même application que les messages discrets ne sont pas toujours indépendants les uns des autres et que des séquences doivent être reconstruites. Il s'agit alors d'un flux continu sur une petite période. La Poste ignore cette notion de flux continu.

Certaines applications, télévision, radio, utilisent exclusivement des flux continus. Les images d'un film doivent arriver séquentiellement au destinataire afin de pouvoir reproduire celui-ci correctement. On parlera dans ce cas de flux continu. Ce flux a en outre des propriétés temporelles : les images se succèdent à un rythme fixe (constant,

24 images par seconde). On parlera dans ce cas de flux temps réel, car les images doivent être délivrées au même rythme qu'elles ont été produites, comme le montre la figure 2.11.

2.1.5. Service sur connexion

Un service sur connexion se distingue du service à datagramme par le fait qu'il faut préalablement à toute communication établir un contexte de connexion. Ce contexte contient les informations nécessaires à la gestion de l'échange. Le téléphone est un bon exemple de service sur connexion. Sa vocation est comme tout réseau d'offrir à ses utilisateurs une voie de communication pour transmettre leurs informations. La manière d'utiliser le téléphone est très différente de la manière d'utiliser le service postal. Avec le téléphone, tout échange d'information nécessite trois phases :

— établissement de la communication téléphonique. L'utilisateur doit donner au service du téléphone le numéro (l'adresse) du correspondant désiré. Le correspondant doit être présent et accepter la communication (en l'occurrence décrocher). Cette phase ne permet pas de transmettre d'informations, du moins officiellement¹. Des

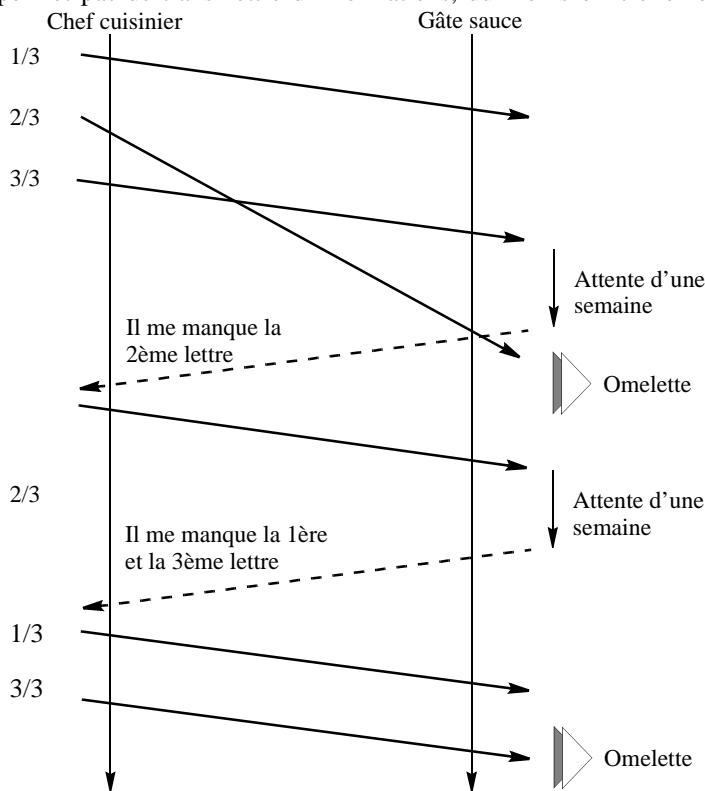


Figure 2.10. Chronogramme ou diagramme temporel du protocole avec retard d'un message

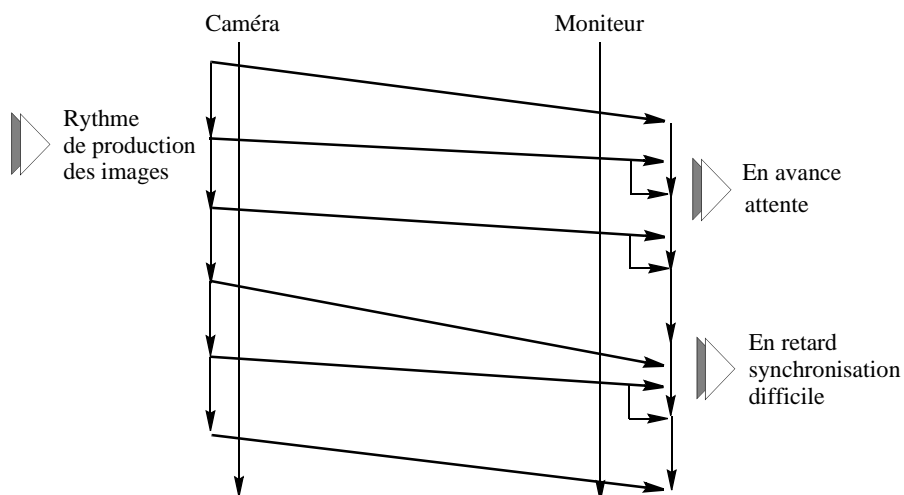


Figure 2.11. Chronogramme ou diagramme temporel d'un flot d'images

ressources sont affectées pour communiquer. Un contexte de communication a été créé à votre profit ;

— une fois la communication établie, vous pouvez transmettre vos informations et en recevoir. Vous n'avez plus besoin de rappeler au service l'adresse de votre correspondant. Cette information est conservée dans le contexte de la connexion. Il est nécessaire que le destinataire soit présent au moment de l'envoi, pendant le transfert et à l'arrivée de l'information ;

— enfin, lorsque vous avez terminé votre conversation, vous libérez la connexion. Les ressources qui vous ont été affectées (louées temporairement par France Télécom) sont libérées afin de pouvoir être utilisées par d'autres utilisateurs.

Les caractéristiques ci-dessus sont typiques de la famille des services sur connexion.

D'autre part, il est toujours nécessaire de demander l'accord du destinataire pour lui envoyer des informations dans un service sur connexion. Il y a souvent besoin de s'abonner pour utiliser un service sur connexion.

1. Les petits futés qui sonnent trois fois chez leurs amis pour indiquer qu'ils sont arrivés utilisent un canal de communication caché, c'est-à-dire une utilisation détournée du téléphone qui leur permet de construire une voie de communication.

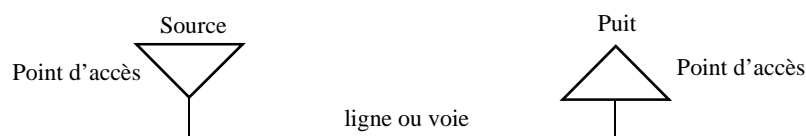


Figure 2.12. Composants d'une voie de communication

2.2. Voie de communication

Essayons maintenant de formaliser les composants qui permettent de construire un réseau de communication. A l'inverse de l'approche précédente, cette présentation est abstraite et indépendante de tout moyen technique (support) permettant de réaliser le service.

2.2.1. Unité d'information

L'unité d'information transmise ou traitée porte différents noms : message, paquet, cellule, trame, PDU... Tous définissent un bloc fini de bits. Nous les utiliserons indifféremment dans ce chapitre. Ces blocs sont transmis de l'entrée d'une voie à la sortie de celle-ci. Ces termes manquent en général de précision lorsqu'ils sont utilisés hors du contexte particulier d'une norme ou d'un produit. Aussi, nous utiliserons de préférence l'acronyme PDU pour désigner un message. Ce terme sera défini avec précision dans le prochain chapitre.

2.2.2. Notion de voie

Le but d'un service de communication est de transmettre des informations issues d'une source de données (mémoire, caméra, micro, capteur...) vers un ou plusieurs destinataires ou puits de données (mémoire, écran, haut-parleur, activateur...) qui la consommera. La figure 2.12. montre les composants de ce service de communication. Le lecteur prendra bien soin de noter qu'un service est réalisé par un ensemble de composants. Aucun service réseau n'est réalisé par un seul composant. Par exemple le réseau téléphonique, Transpac, ATM, Internet sont des services. Un routeur, un combiné téléphonique, un modem sont des composants matériels ; un protocole (le programme qui le réalise), un « mailer »... sont des composants logiciels : aucun d'eux ne suffit à lui seul à réaliser le service complet.

La voie de communication (aussi appelé ligne, canal, liaison, tuyau) est le moyen par lequel l'information est transmise de la source vers le puits. Tous moyens aptes à faire transiter une information d'une origine vers une destination peuvent être considérés comme une voie de communication. Ainsi, le service postal offre une multitude de voies de communication à partir de n'importe quelle boîte aux lettres. Le service téléphonique offre une voie de communication exclusive entre une source et un puits lorsque la communication est établie. Un câble métallique est une voie de communication qui propage une onde électrique (information analogique). L'air est

une autre voie de communication qui propage l'onde de pression générée par vos cordes vocales. Mais il ne faut pas assimiler voie de communication avec support physique. Le support physique (câble électrique, véhicule, air, lumière) est nécessaire pour le transit de l'information, mais n'est pas suffisant pour construire une voie apte à transporter des données.

Pour passer de la source (le support originel) à la voie de communication, une interface, ou point d'accès, est nécessaire. Réciproquement, à l'arrivée une seconde interface est nécessaire qui permet de passer de la voie vers le puits. Le téléphone prend comme source votre voix : l'interface est le micro qui transforme une onde de pression en onde électrique. Réciproquement, l'écouteur effectue la fonction inverse. Il n'y a aucune raison pour que les interfaces d'émission et de réception soient identiques. Nous y reviendrons dans les chapitres suivants. Dans un système informatique, on appelle modem, pour modulateur/démodulateur, l'équipement électronique qui fait l'interface entre les informations numériques qui sont dans une machine et le type de signal qui peut être transmis (propagé) sur la voie.

Ligne ou voie	Point d'accès ^a
Câble métallique	Modem
Câble optique	Modem optique
Connexion téléphonique	Téléphone
Transpac	Minitel, modem
Courrier postal	Boîte à lettres
Courrier électronique	Boîte à lettres + utilitaire
SNCF	Guichet

a Ici les points d'accès cités sont tous des composants

Figure 2.13. Exemple de voies et de points d'accès

Présenter la source et le puits de données uniquement comme des informations n'est pas suffisant. En effet, une donnée est essentiellement une entité passive. Il faut donc associer à ces données un composant actif : homme ou programme. En informatique, on appelle processus une entité active qui agit sur un flot de données.

Nous utiliserons le terme « entité active » pour désigner les processus qui reçoivent et émettent des données. Une entité active, processus, est associée à la source et un processus est associé à chaque puits. Ces processus interagissent avec les interfaces de la voie. Vue d'un processus, l'interface est constituée d'un ensemble de

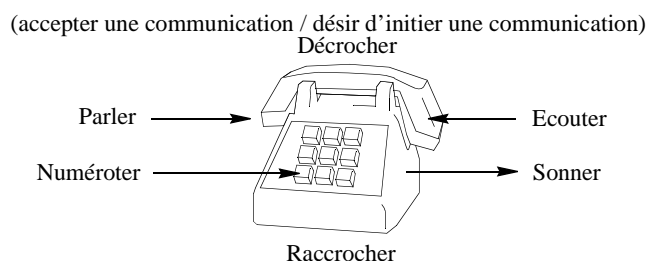


Figure 2.14. Interaction service téléphonique-processus utilisateur

fonctions de base. Par exemple, votre interface téléphonique dispose des primitives : numéroté, sonner, parler, écouter, que vous utilisez selon des règles prédéfinies (figure 2.14.). Ces fonctions sont élémentaires. On ne peut accéder à ce qu'elles recèlent, on ne peut que les utiliser en leur fournissant éventuellement des paramètres : la fonction numérotée prend le numéro comme paramètre. On peut qualifier ces fonctions de « fonctions primitives » et par la suite nous les appellerons « primitives » ou primitives de service. On notera que ces primitives peuvent être laissées à l'initiative du processus (numéroté, parler, écouter) et que d'autres sont à l'initiative de l'interface (sonner).

Les schémas développés montrent qu'au moins trois activités ont lieu simultanément :

- le processus source,
- l'activité de la voie,
- chaque processus de réception.

Toute activité d'échange d'information implique une activité simultanée, parallèle d'au moins trois processus. Dans un amphithéâtre où un cours magistral a lieu, le parallélisme est nettement plus élevé ; il y a un seul orateur, le processus physique de propagation du signal vocal dans l'air ; un processus d'écoute par élève (espérons le du moins).

Ces activités sont asynchrones, c'est-à-dire qu'il n'y a pas de moyen de forcer ces processus à fonctionner au même instant sur la même grandeur (physique ou donnée) ; nous y reviendrons au paragraphe 2.3.1.4.

Les points d'accès au service ne sont pas des entités actives mais des points de référence adressables (designables). Ils constituent l'interface entre l'utilisateur et le service. Le service y dépose les informations destinées à un abonné. Réciproquement, un abonné dépose les informations qu'il souhaite voir transmises par un service donné sur l'un des points d'accès de ce service, généralement accompagné de l'adresse d'un ou plusieurs autres points d'accès du service pour lui indiquer ou déposer cette information.

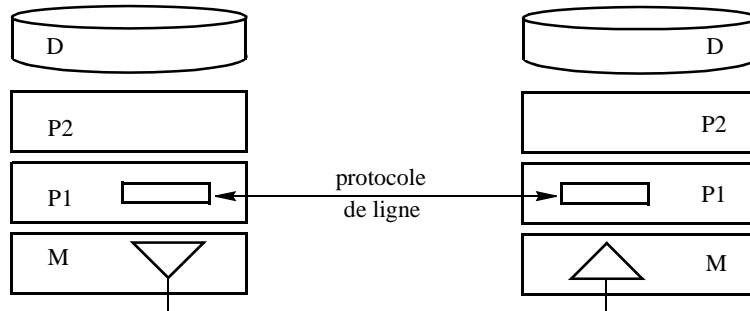


Figure 2.15. *Activité d'échange d'informations*

Donc, l'objet essentiel d'un réseau de communication est de permettre le transfert de données d'au moins un point source vers au moins un point destination (cf. figure 2.15.). Tout service de communication a pour but de construire une « voie » M ou canal de communication. On utilise aussi les termes connexion ou liaison pour le transfert des données. Cette voie pourra être utilisée pour construire une nouvelle voie M', à partir de M et de mécanismes de traitement, dont les propriétés sont mieux adaptées aux besoins d'une famille d'utilisateurs (cf. figures 2.16. et 2.17.). Nous parlerons de voie logique pour M', car elle est constituée de plus d'entités qu'une voie physique.

Vu du programme qui traite les données, le schéma équivalent à la figure 2.16. est représenté figure 2.17. L'association d'un ensemble de processus identiques permet de construire un service de communication. M' est un nouveau service de communication construit sur les entités M et la voie inférieure sur nos schémas.

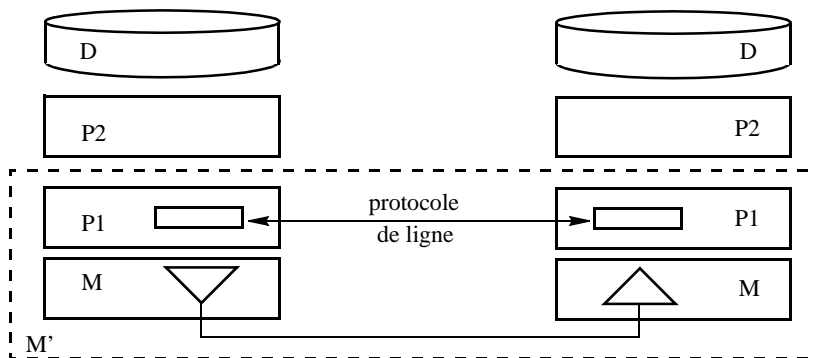


Figure 2.16. *L'association d'une voie M et d'un protocole P produit une voie de communication M' dont les propriétés sont modifiées par rapport à M. M' forme une voie logique*

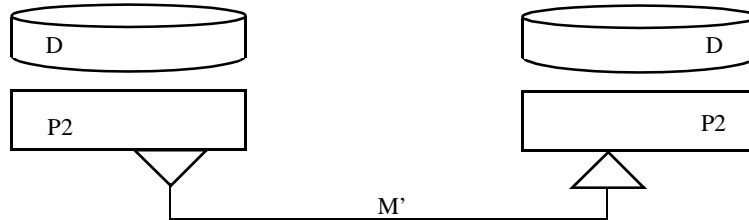


Figure 2.17. Figure équivalente à la figure 2.16, la voie M sous-jacente est masquée. M' est une voie logique

Les machines à communiquer que sont M ou M' sont potentiellement parallèles. Les deux composants peuvent exécuter simultanément des programmes et des tâches de communication.

Le niveau de parallélisme d'une voie dépend de la manière dont elle est réalisée. Il n'y a donc pas de règle générale. Ainsi, sur la figure 2.15, une implantation pourra par exemple affecter un processeur à chaque entité $P1$, $P2$ et M . Le niveau de parallélisme sera alors 7, les six entités sont : $P1$, $P2$, M sources, M , $P2$, $P1$ puits en jeu et le processus de propagation sur la voie physique.

La figure 2.18. montre le parallélisme effectif à un instant particulier. L'entité $P2$ source traite le message m_8 , pendant que simultanément l'entité $P1$ source traite le message précédent m_7 , les entités M émission et réception sont respectivement en train d'émettre et recevoir le message m_6 , l'entité $P1$ puits traite le message m_5 et l'entité $P2$ puits traite le message m_4 . Chaque entité a besoin d'un temps de traitement pour chaque message. Chaque entité ne peut traiter qu'un seul message à la fois.

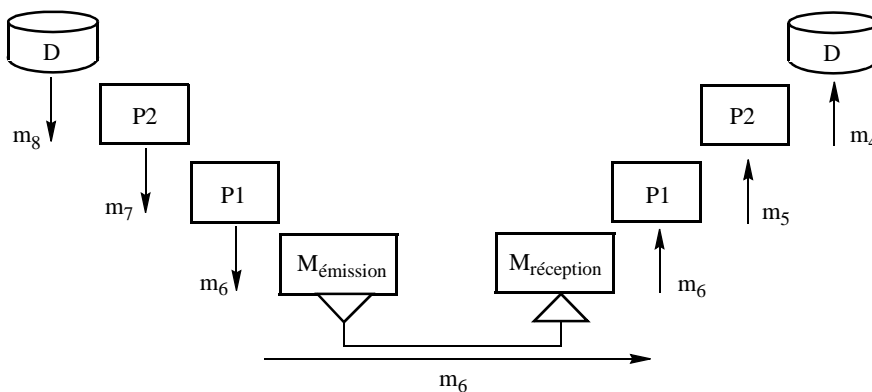


Figure 2.18. Activités simultanées possibles sur la voie décrite selon l'empilement d'entités protocolaires des figures 2.15. et 2.16

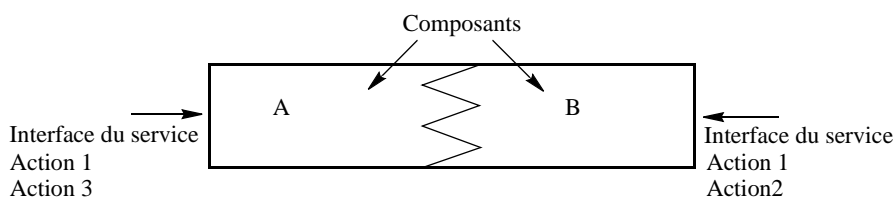


Figure 2.19. *Vision externe d'une interface*

Une autre implantation pourra se rapprocher de la description de la figure 2.16. et attribuer un seul processeur aux entités M et P1 et un autre aux entités P2 de chaque site. Le site ici est matérialisé sur la figure 2.16 par la position gauche ou droite des entités. En pratique, un site correspond à l'emplacement physique de l'équipement qui abrite ces entités. Avec ces hypothèses, le parallélisme sur la figure 2.16. sera 5 soit les deux entités P2 et 3 entités pour la voie M' constituée des processus qui implantent les processus P1, M en chaque site et le processus de propagation sur la voie physique.

L'étude du parallélisme sur la figure 2.17. est plus délicate. En effet, sur ce schéma seule la voie logique M' apparaît. Une analyse similaire à celle faite pour la figure 2.15. nous conduirait à dire que le parallélisme est 3, deux processus P2 et le processus d'acheminement sur la voie logique M'. Or l'on vient de voir que cette voie avait elle-même un parallélisme de niveau 3. Lorsque l'on étudie une architecture de réseau, en général construite à partir de voies logiques, il n'est pas possible de connaître le niveau exact de parallélisme de la voie logique.

2.2.3. Interface entre composants et protocole

Pour que la coopération soit possible, les composants doivent disposer de moyens d'interaction. Une action exécutée par un composant doit pouvoir produire un résultat sur d'autres composants. On convient d'appeler « interface » les moyens d'interaction entre un ensemble de composants. Ces moyens consistent en un choix de primitives et un formatage des données échangées en fonction de la nature de ces données. Vue d'un composant, l'interface est constituée de l'ensemble des actions qu'il peut entreprendre sur les autres composants et des résultats d'actions qu'il peut subir de leur part (cf. figure 2.19.).

La coopération de plusieurs composants dans la réalisation d'une fonction est réglée par un ensemble de conventions et de règles qu'on appelle protocole (cf. figure 2.20.). Un protocole définit donc les relations entre les comportements des différents composants de la fonction, au travers de leur interface. Un protocole est une classe particulière d'algorithme, en ce sens que l'algorithme est distribué, c'est à dire exécuté par plusieurs processeurs qui communiquent par des messages. Chaque processeur exécute une partie du traitement de l'algorithme.

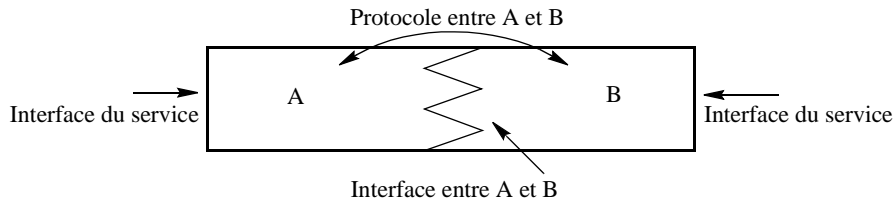


Figure 2.20. Protocole, service et interfaces

Un protocole entre un ensemble de composants est généralement décomposé en un ensemble de protocoles entre paires de composants, si bien qu'on parle le plus souvent du protocole entre deux composants.

2.2.4. Organisation des services, composants et protocoles

L'organisation d'un service sur le réseau peut être caractérisée, d'une part, par la distribution des composants qu'il met en jeu et, d'autre part, par la répartition du contrôle du service entre les différents composants. Bien qu'on sache dans certains cas réaliser des assemblages complexes de composants, les types d'organisation les plus couramment rencontrés (il faut tenter de les retrouver car ce sont les plus simples) sont l'assemblage en couches, ou « en pelure d'oignon » (cf. figure 2.21.), l'assemblage en série (linéaire), ou « en cascade », et le « contrôle de bout-en-bout ».

Dans l'assemblage « en pelure d'oignon », encore appelé « hiérarchie de protocoles » (figure 2.21.), un composant C_i est mis en sandwich entre deux autres composants C_{i+1} et C_{i-1} .

Nous appellerons la boîte aux lettres qui sert à l'échange entre deux couches un point d'accès au service ou SAP, *Service Access Point*. Le composant C_i dialogue avec le composant C_{i+1} (et réciproquement) en utilisant une interface I_i un point d'accès appelé SAP_{i+1} et un protocole local $P_{i,i+1}$, qui lui est propre. I_i , SAP_{i+1} et $P_{i,i+1}$ masquent les activités des composants de couches inférieures : par exemple, le composant C_{i-1} assure l'accès du réseau, le composant C_i assure la transmission d'une trame contenant

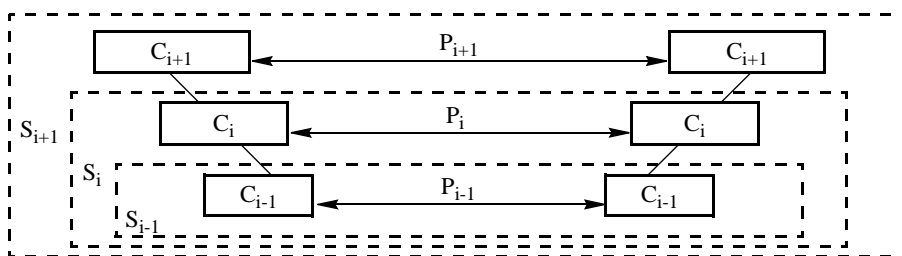


Figure 2.21. Assemblage en pelure d'oignon ou hiérarchie de protocoles

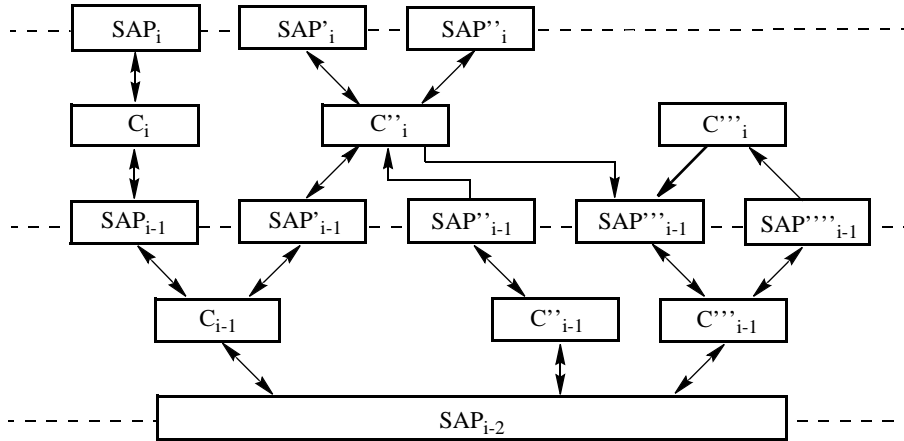


Figure 2.22. Relations entre les SAP et les composants

un paquet, enfin le composant C_{i+1} est chargé du découpage d'un message en plusieurs paquets.

L'ensemble des composants C_i coopèrent pour offrir un service S_i . Les entités C_i sont dites entités homologues. Les entités C_i communiquent et coopèrent entre elles selon un protocole P_i au travers de l'ensemble des services fournis par les services S_{i-1} de la couche $i-1$. Les entités C_i accèdent aux services S_{i-1} fournis à partir de points d'accès appelés SAP_{i-1} . Chaque point d'accès est identifié par une adresse. Chaque SAP_{i-1} délivre les messages à un seul composant C_i et prend en dépôt les messages d'un seul composant C_{i-1} (cf. figure 2.22.). Par contre plusieurs composants C_i (sur la figure 2.22. C''_i , C'''_i par exemple) peuvent soumettre sur un même SAP_i (SAP''_i sur la figure 2.22.). De même un composant C_i peut recevoir des messages de plusieurs SAP_i : il assure en cela une fonction de multiplexage, et recevoir de plusieurs SAP_{i-1} : il assure en cela une fonction de démultiplexage. Un composant C_{i-1} peut déposer des messages reçus dans plusieurs SAP_{i-1} . Un composant C_{i-1} peut prendre des messages dans plusieurs SAP_{i-1} .

Les composants d'un service de niveau i noté, S_i , coopèrent entre eux grâce à un protocole P_{i+1} . Lorsque plusieurs services S_{i-1} , S_i , S_{i+1} sont associés, les composants propres à chaque service coopèrent entre eux selon un protocole qui leur est propre : P_{i-1} , P_i et P_{i+1} .

2.2.4.1. Assemblage en série

Dans un assemblage en cascade ou en série (figure 2.23.), les composants C_i sont associés deux à deux par une voie de communication et utilisent un protocole sur chaque voie. Il peut y avoir un protocole distinct par couple de composants. Le protocole est défini pour chaque couple de composants adjacents. Il est fréquent que le protocole soit identique tout au long de la chaîne (exemple : mise bout à bout de

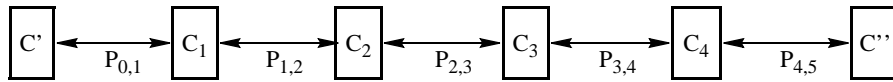


Figure 2.23. Assemblage en cascade ou en série

chaînon de données par des commutateurs de paquets), mais cela n'est ni nécessaire ni le cas général.

2.2.4.2. Contrôle de bout-en-bout

L'appellation « contrôle de bout-en-bout » (figure 2.24.) est généralement réservée à la mise « en sandwich » d'un assemblage en cascade (a) ou de tout autre assemblage complexe (b) entre deux composants d'extrémités coopérant directement entre eux deux pour contrôler globalement la cascade. Cette superposition d'un contrôle de bout en bout sur un assemblage en cascade (ou sur un assemblage plus complexe) est très largement utilisée, car elle permet de simplifier énormément le contrôle en cascade en lui évitant par exemple d'avoir à prendre directement en compte la défaillance d'un des composants. Dans la voie C' - C'', on appelle systèmes intermédiaires les composants C₁, C₂, C₃ et C₄ qui permettent sa construction.

2.2.4.3. Voie logique et voie physique

Nous appellerons par la suite une voie physique celle constituée d'un seul support physique. Par contre, nous appellerons voie logique celle construite par l'assemblage de voies physiques ou logiques en cascade ou en série. L'entité qui utilise un service

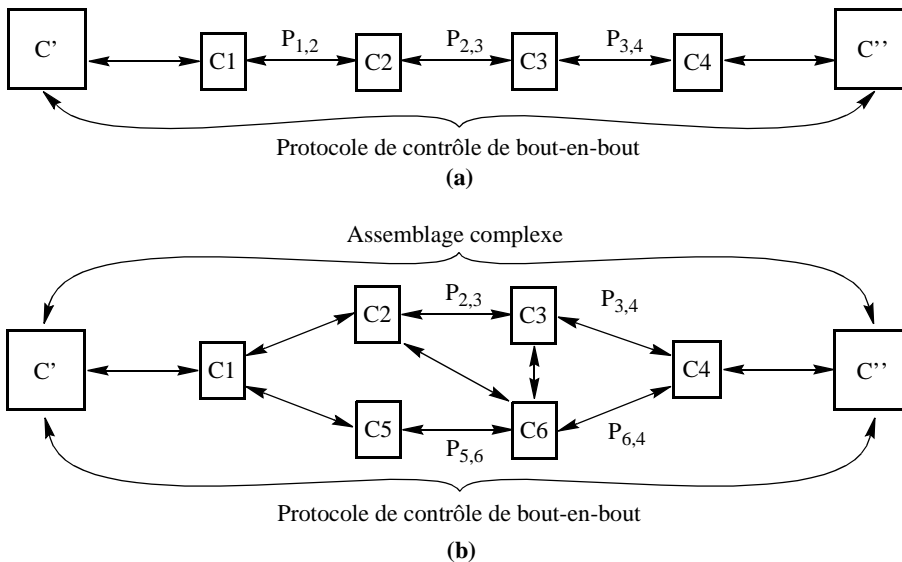


Figure 2.24. Contrôle de bout-en-bout

de communication ne sait pas nécessairement comment sont construites les voies qu'elle utilise.

2.3. Fonction transport d'information

2.3.1. Caractéristiques d'une voie

2.3.1.1. Débit ou bande passante

On appelle débit ou bande passante la quantité d'information qu'une voie peut écouler par seconde. Comme nous nous intéressons essentiellement au transfert d'information numérique (l'information originelle, image, son, température... éventuellement numérisée), la bande passante ou débit, D , sera exprimé en bit par seconde noté bp/s. Une voie ayant un débit D de 19,6 Kb/s écoule 19 600 bits par seconde ($K = 10^3$ bp/s, $M = 10^6$ bp/s, $G = 10^9$ bp/s).

Le débit est une caractéristique de la voie utilisée. Il n'est pas nécessaire que le débit offert à la source soit identique au débit offert au puits. La voie peut être symétrique ; le débit est le même à toutes les extrémités ou asymétrique ; le débit n'est pas identique à chaque interface (ex : le Minitel). Notons D_e le débit d'émission et D_r le débit de réception.

On appelle « débit utile » le débit demandé (fourni) par l'application.

L'information utile soumise doit pouvoir contenir n'importe quelle configuration binaire. Si pour une raison quelconque un service de communication utilise certaines séquences binaires particulières pour son protocole, cela devra rester transparent à l'utilisateur. On appelle transparence cette propriété. Elle sera assurée par des mécanismes protocolaires particuliers (cf. dans le paragraphe 5.2.3. le mécanisme de bit stuffing de HDLC).

2.3.1.2. Débit nominal

Le débit nominal est généralement le débit fourni par le support physique. Sur une voie physique, le débit d'émission et le débit de réception sont identiques. Ce n'est pas forcément le cas sur une voie logique.

Le débit étant exprimé en b/s, on appelle durée d'émission/réception le temps nécessaire pour que l'interface émette/reçoive un certain nombre de bits. Soit T_m la taille en bits d'un message. La durée d'émission, d_e , ou de réception, d_r , est bien évidemment égale à :

$$d_e = \frac{T_m}{D_e} \quad d_r = \frac{T_m}{D_r}$$

Le Minitel est un bon exemple de voie dissymétrique. Il reçoit au débit $D_r = 1\,200$ b/s et émet au débit $D_e = 75$ b/s.

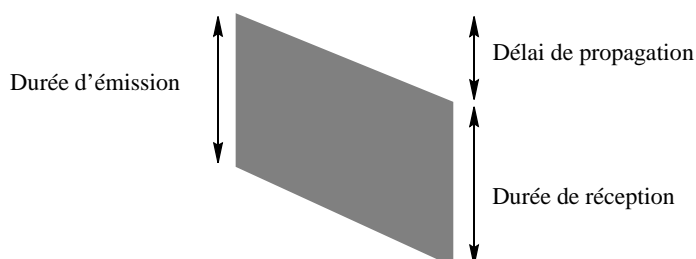


Figure 2.25. Relations entre délai de propagation et durées d'émission/réception au niveau physique

2.3.1.3. Taux d'occupation ou taux d'utilisation

C'est la proportion de la bande passante, débit disponible, consommée sur une voie pendant une période de temps. C'est-à-dire le volume de données effectivement transmises par rapport à la capacité de la voie.

Le taux d'utilisation, noté τ , aussi appelé taux d'occupation, est le rapport entre T_e , la durée d'usage effectif de la voie de communication, et la durée totale de la mesure. Soit pendant une durée T : $\tau = (\text{Données transmises}) / (D_e * T)$. Un taux d'occupation est toujours inférieur ou égal à 1. Il peut aussi être mesuré de manière équivalente par la durée d'usage effectif, temps pendant lequel des informations sont effectivement transmises, sur la période de mesure : $\tau = T_e / T$.

En téléphonie, on mesure cette valeur en Erlang : un Erlang est le taux d'utilisation d'une voie téléphonique. Pour votre ligne privée, c'est le rapport entre le temps d'utilisation de votre ligne téléphonique (temps passé au téléphone) sur le temps total de la mesure.

2.3.1.4. Délai d'acheminement

Il s'agit du temps que met un message pour aller de la source (ou point de départ) au puits (ou point d'arrivée ou de destination). On parle aussi parfois du délai de transit de bout en bout. Il est composé selon les cas de la somme de plusieurs grandeurs : délai de propagation, délai de transit, durée de commutation, durée d'émission et réception... Le parallélisme inhérent au fonctionnement d'une voie peut masquer certaines durées.

2.3.1.4.1. Délai de propagation

C'est le temps nécessaire à la propagation d'un signal physique de la source vers le puits. C'est donc une grandeur physique incompressible. La propagation d'un signal n'est jamais instantanée, la figure 2.26. rappelle quelques vitesses de propagation¹.

Donc, un signal subit pour aller d'un point à un autre un délai de propagation égal à d_p fonction de L , la distance à parcourir et de la vitesse V de propagation dans le support.

$$d_p = L/V$$

Support	Grandeur modulée	Vitesse de propagation
Vide	Lumière	300 000 Km/s
Air	Lumière	300 000 Km/s
Air	Pression (sons)	330 m/s
Métal	Courant électrique	220 000 Km/s
Fibre optique	Lumière	220 000 Km/s

Figure 2.26. Vitesses de propagation sur différents supports

Ainsi, un signal met environ 0,12 s pour aller de la terre à un satellite géostationnaire situé à 36 000 km (cf. figure 2.27.).

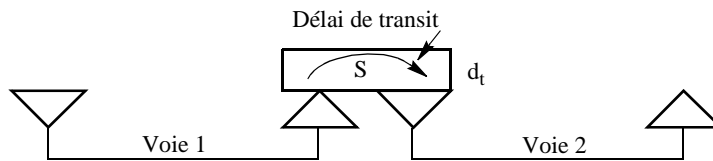


Figure 2.27. Place du délai de transit dans un assemblage en cascade (*S* est par exemple un satellite)

2.3.1.4.2. Délai de traitement ou de transit

L'exécution des fonctions réalisées par un composant prend un temps non nul, appelé durée de traitement ou délai de transit (cf. figure 2.28.).

Une voie peut être composée d'une succession de voies ou d'un empilage de voies. Lorsqu'on passe d'une voie à une autre, il y apparaît un délai de traitement pour traverser le(s) composant(s) appelé délai de transit. Ce délai est fonction des caractéristiques des composants qui assurent l'assemblage des deux voies. Les figures 2.27. et 2.28. montrent la place du délai de transit dans les assemblages en cascade et hiérarchique.

Un assemblage peut être constitué de plusieurs niveaux comme sur la figure 2.29. : par exemple, une cascade selon le protocole de bout en bout P1 et des hiérarchies de

1. Il convient de ne pas confondre le délai de propagation avec la durée d'émission, qui sont deux notions totalement différentes. Le délai de propagation est une grandeur physique qui dépend exclusivement de la longueur du support et de ses caractéristiques, en aucun cas du débit.

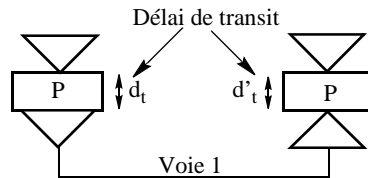


Figure 2.28. Délai de transit dans une architecture hiérarchique

protocoles P1 et P2. Nous reverrons plus loin des architectures complexes de ce type. Sur la figure 2.29, apparaissent six délais de transit distincts.

Le délai de traitement dans une entité, d_t , a une influence sur le débit de la voie. En effet, si un message de T bits est soumis à une entité, celle-ci ne pouvant traiter qu'un message à la fois fournit un débit nominal de T/d_t . Le débit peut donc être limité par la durée de traitement des entités.

Si l'on reprend l'exemple de la figure 2.27, le débit que l'on peut atteindre si la source soumet continuellement des messages de taille T bits sera donc le plus petit débit offert par la voie physique (débit nominal) et l'entité S (débit du au traitement), soit : $\text{Min}(D, T/d_t)$.

Si l'on reprend l'exemple de la figure 2.28, le débit que l'on peut atteindre si la source soumet continuellement des messages de taille T bits sera donc le plus petit débit offert par la voie physique (débit nominal), l'entité P source et l'entité P puits, soit : $\text{Min}(D, T/d_t, T/d'_t)$.

2.3.1.4.3. Calcul du délai d'acheminement

Le calcul du délai d'acheminement (ou du délai de transit de bout en bout) se déduit trivialement à partir de la connaissance de l'assemblage réalisé pour construire la voie. Il s'agit de la somme :

- délai de transit à chaque assemblage,

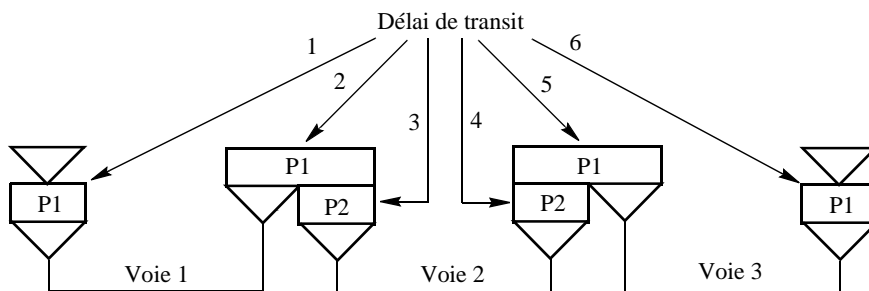
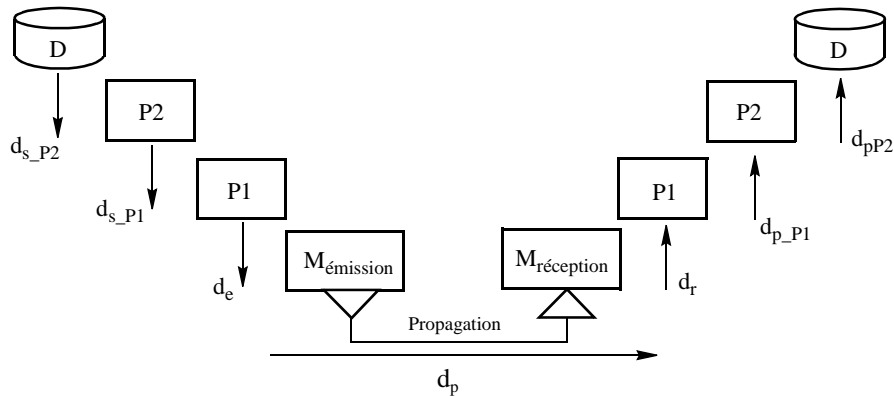


Figure 2.29. Place des délais de transit qui participent le délai d'acheminement dans un assemblage en cascade et hiérarchique



Les notations d_i indiquent les durées de traitement propres à chaque entité ainsi que le délai de propagation d_p .

Figure 2.30. Succession des temps de traitement nécessaires à l'acheminement d'un message selon l'empilement d'entités protocolaires des figures 2.15. et 2.16

- délais de propagation sur chaque voie physique ou délai d'acheminement sur une voie logique,
- $\max[d_e, d_r]$: on ne comptera qu'une seule durée puisque ces opérations sont effectuées en parallèle.

La figure 2.30. montre la succession des traitements que doit subir un message pour son acheminement sur la voie décrite sur les figures 2.15. et 2.16. La figure ne fait pas complètement apparaître le parallélisme qui existe entre les activités d'émission, de propagation et de réception. Le délai d'acheminement sur cette voie est donc égal à :

$$d_a = d_{s_P2} + d_{s_P1} + d_{p_P2} + d_{p_P1} + \max(d_e + d_r) + d_p$$

La figure 2.27. schématise une architecture comprenant un satellite. Le délai d'acheminement est composé de :

- délai de propagation sur les voies 1 et 2, soit environ $2 \times 0,12$ s,
- délai de transit dans le satellite, quelques micro secondes (peut être négligé),
- délai de transit ou durée de traitement à la source et au puits,
- durées d'émission du message sur les voies 1 et 2 puisque la durée d'émission et de réception est identique.

Pour la figure 2.28. le délai d'acheminement est égal à la somme des délais suivants :

— deux fois le délai de transit (temps de traitement) par les entités P, notées d_t et d'_t sur la figure 2.28.

— $\max(d_e, d_r)$, le maximum entre les durées d'émission ou de réception,

— le délai d'acheminement sur la voie 1 noté d_{v1} , v_i étant une voie logique offerte par un service dont on ne connaît pas le détail de l'architecture.

$$d_a = 2xd_p + \max(d_e, d_r) + d_{v1}$$

Le délai d'acheminement pour la figure 2.29., est égal à :

$$d_a = 4xd_{p1} + 2xd_{p2} + d_{v1} + d_{v2} + d_{v3} + \max(d_{ev1}, d_{rv1}) + \max(d_{ev2}, d_{rv2}) + \max(d_{ev3}, d_{rv3})$$

où

— d_{p1} est le temps de traitement dans chaque entité P1 (supposé identique en chaque site, ce qui est une hypothèse simplificatrice mais peu réaliste) et d_{p2} est le temps de traitement dans chaque entité P2,

— d_{v1} , d_{v2} et d_{v3} sont les délais de propagation ou d'acheminement selon le type de voie physique ou logique sur les voies V1, V2 et V3,

— d_{ev_i} , la durée d'émission sur la voie i , d_{rv_i} la durée de réception sur la voie i .

La compréhension des composants du délai d'acheminement est essentiel. Pour une application interactive, on souhaite un délai d'acheminement court afin de réduire le délai qui s'écoule avant que le destinataire reçoive et interprète l'information.

Ces exemples montrent que le délai d'acheminement est toujours une fonction du débit et de la taille du message à cause de $\max(d_e, d_r)$, délai d'émission ou de réception, des temps de traitement dans chaque entité protocolaire (P_1, P_2, \dots) et enfin des délais d'acheminement ou de propagation sur les voies utilisées.

Le délai d'acheminement est cumulatif. Le message doit passer successivement par les différents traitements. Le parallélisme ne permet pas de le réduire.

2.3.1.5. Asynchronisme

Ces délais d'acheminement expliquent l'asynchronisme que nous avons déjà évoqué entre le processus source, le processus puits et la voie. Il n'y a aucun moyen de forcer la synchronisation temporelle entre deux entités distantes. En effet, pour donner à plusieurs entités l'heure (« au troisième top, il est midi »), il faut transmettre l'information « heure ». Or, cette information subit un délai d'acheminement. L'information arrive après avoir été émise. Le retard associé est égal au délai d'acheminement. Le problème serait simple si le délai d'acheminement était constant dans le temps et constant pour tous les puits depuis l'origine (source, point de départ) du message. Mais cette propriété n'est en général pas vraie.

Cette propriété d'asynchronisme est fondamentale dans le traitement des problèmes réseau. En réseau, il n'est pas possible de provoquer l'arrêt simultané des processus. Il n'y a plus de base de temps unique (universelle). Les évidences ci-dessus

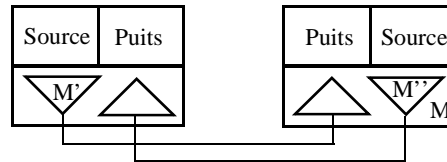


Figure 2.31. Voie bidirectionnelle

énoncées ont des conséquences majeures qui distinguent radicalement la conception d'une application centralisée de celle d'une application parallèle (en réseau).

2.3.1.6. Voie bidirectionnelle

Une voie peut être utilisée pour transmettre des données dans les deux sens, comme c'est le cas sur une voie téléphonique. On dit que cette voie est bidirectionnelle, ou *full duplex*. Une voie peut être unidirectionnelle, ou *half duplex*. C'est le cas d'un talkie-walkie qui vous offre la possibilité d'émettre ou de recevoir, mais ne vous permet pas les deux simultanément.

Le fait qu'une voie soit full duplex implique qu'il existe de chaque côté de la voie un processus source et un processus puits. Ces processus doivent en général être actifs simultanément, à moins de mettre en œuvre un protocole qui organise le sens de communication sur la voie. Une voie bidirectionnelle est donc composée de deux voies unidirectionnelles, une dans chaque sens (figure 2.31.).

2.3.1.7. Voie multipoint ou point à point

Une voie est dite point à point lorsqu'elle ne relie que deux correspondants, un correspondant étant soit une source, soit un puits, soit un couple source-puits dans le cas le plus fréquent d'une voie bidirectionnelle. La plupart des services réseau sont conçus pour offrir un service point à point (téléphone, Poste...).

Une voie est dite multipoint lorsqu'elle relie plus de deux correspondants. De nombreuses voies physiques possèdent cette propriété. C'est le cas de l'air que nous modulons pour parler. A l'intérieur d'un local, tout le monde peut parler (émettre) et entendre (recevoir). Il s'agit d'une voie multipoint full duplex. De nombreuses applications utilisent une voie multipoint unidirectionnelle. La radio et la télévision sont des réseaux construits sur ce principe. Il y a un seul émetteur et autant de récepteurs que l'on souhaite. Les réseaux utilisant un satellite sont typiquement une voie multipoint puisque tout poste radio ou télévision (station) peut recevoir. Si tous les abonnés de la voie peuvent émettre (ont un processus source), il y a lieu de créer un protocole de partage de la voie.

2.3.1.8. Fiabilité d'une voie

Il n'existe pas de voie de communication parfaite. Toute voie subit des erreurs de transmission en quantité plus ou moins importante. On appelle taux d'erreur la probabilité de perdre un bit transmis sur une voie. Ainsi, un taux d'erreurs de 10^{-6} signifie une probabilité de perdre un bit sur un million de bits transmis.

Supposons que l'on veuille transmettre sur une voie, notée v , ayant un taux d'erreur $P(v)$ un fichier de T bits. Le nombre moyen de bits erronés est : $T * P(v)$. Si $T > 1/P(v)$, il est sûr que le fichier sera abîmé (en supposant une loi de distribution uniforme). Si T inférieur mais proche de $1/P(v)$, la probabilité que le fichier soit transmis sans erreur est forte.

Afin de détecter les erreurs de transmission, on ajoute aux données transmises une séquence de bits de vérification (bit de parité, checksum, cf. paragraphes 4.2 et 4.3). Ces informations ne sont destinées ni à localiser ni à corriger les erreurs, mais à détecter avec une probabilité plus ou moins élevée qu'une erreur s'est produite pendant la transmission. Il faut donc mettre en œuvre des mécanismes, appelés protocoles de correction des erreurs. Ceux-ci sont basés sur les principes utilisés au cours d'une conversation. Lorsque vous ne comprenez pas ce qui vous est dit (parce que le son n'est pas parvenu à votre oreille ou qu'il est parvenu brouillé, abîmé, par une autre source de son...), vous demandez à votre interlocuteur de répéter. Les protocoles réseau utilisent intensivement ce principe. Supposons que le récepteur dispose d'un moyen pour demander la répétition d'un message qui lui est parvenu abîmé. Soit T_m la taille du message et $P(v)$ le taux d'erreur de la voie. Soit D le débit de la voie. Pour les besoins de l'explication, nous utiliserons une application numérique $T_m = 10^7$ bits, $P(v) = 10^{-6}$, $D = 10$ kb/s. Nous négligerons le délai de propagation et le délai d'acheminement ainsi que la perte possible des acquittements. La durée d'émission du message est de 1 000 s (soit environ un quart d'heure). La probabilité que le message arrive erroné est certaine ($T * P(v) > 1$). Ainsi, si l'on transmet ce message dans son ensemble, il faudra 1 000 s pour que le récepteur s'aperçoive que le message arrive erroné et à nouveau 1 000 s pour le retransmettre (figure 2.32.). Afin de limiter cet effet, on préfère découper le message en paquets de petite taille. La taille est choisie de telle sorte que la probabilité de destruction d'un paquet reste faible. Dans l'exemple précédent, on prendra par exemple comme taille de paquet 10 000 bits.

Ainsi, la probabilité de perte d'un paquet est de 1 %. Le message va être découpé en 1 000 paquets. On appelle cette opération fragmentation. On remarque que la durée d'émission de chaque paquet est de 1 s. Ainsi, chaque retransmission coûte 1 s au lieu de 1 000. Dans notre exemple, avec $P(v) = 10^{-6}$ pour transmettre 1 000 paquets en moyenne, 10 paquets seront abîmés. Il faudrait donc dix retransmissions. La transmission de l'ensemble du message prendra 1 010 s en présence d'erreurs, au lieu de 2 000 s dans le cas précédent. Cet exemple simple montre clairement l'intérêt de la transmission par paquets afin d'uniformiser le temps de transmission des messages en présence d'erreurs. Bien sûr, pour un grand nombre de messages, la durée totale d'émission (incluant les retransmissions) sera la même. Par contre, certains messages transmis en bloc subiront des délais très élevés, alors que, transmis par paquets, l'augmentation sera faible. L'ensemble du raisonnement précédent suppose que le taux d'erreur est uniformément réparti dans le temps, ce qui n'est pas vrai dans la réalité.

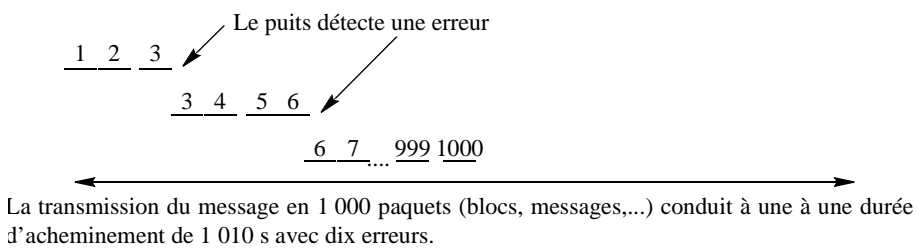
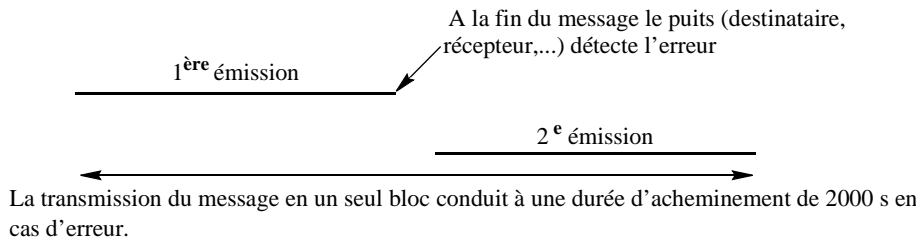


Figure 2.32. Intérêt de la transmission par paquets pour réduire la durée d'acheminement en présence d'erreurs

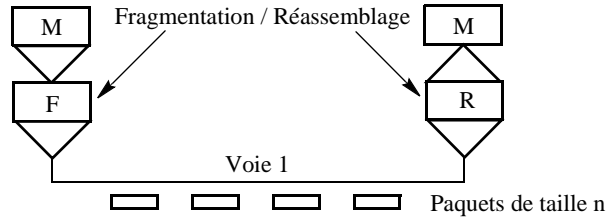


Figure 2.33. Processus de fragmentation / Réassemblage sur une voie unidirectionnelle

2.3.2. Fragmentation/réassemblage

Les messages ou paquets émis sur une voie ont pour les raisons qui viennent d'être développées une taille maximale. A priori un utilisateur n'a aucune raison de se soucier de la taille de la voie utilisée lorsqu'il construit un message à transmettre. La fonction de fragmentation permet de découper un message utilisateur de taille M en autant de paquets de taille n que nécessaire qui eux pourront être transmis sur la voie.

A l'inverse, on appelle réassemblage la fonction qui regroupe les paquets reçus pour reconstruire le message initial de taille M. La figure 2.33. montre l'architecture de ce mécanisme.

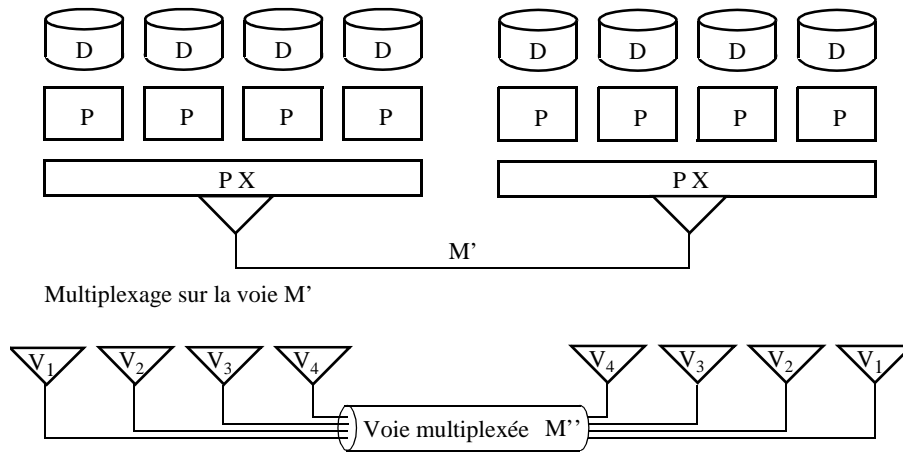


Schéma équivalent.

Figure 2.34. *Multiplexage, partage de la ligne (ou voie) de communication*

2.3.3. Multiplexage

Une voie de communication peut être réservée à un seul utilisateur. Cela se justifie si cet utilisateur a besoin de l'essentiel de la capacité de la voie (du débit, ou bande passante de la voie). Ce n'est pas le cas dans la majorité des applications. En outre, affecter toute la capacité de la voie à un seul utilisateur brime les autres utilisateurs. On appelle multiplexage l'opération qui permet le partage d'une voie de communication entre plusieurs utilisateurs. La figure 2.22. montre deux entités C_i et C'_i qui utilisent le service offert par les entités C_{i-1} . Pour que deux entités d'un même sous-système de niveau i puissent utiliser le service rendu par une entité du même sous-système du niveau inférieur, il faut que cette entité sache séparer les données des entités utilisatrices (de niveau supérieur). L'entité C_{i-1} doit donc savoir multiplexer et démultiplexer les données de ses différents utilisateurs. La figure 2.34. montre plusieurs processus sources et un processus PX chargé du multiplexage. A l'arrivée, un processus inverse, appelé démultiplexeur, assure la restitution des données aux destinataires. Chacun des couples émetteur/destinataire forme une nouvelle voie.

Il existe de nombreux moyens pour réaliser du multiplexage.

Un sac postal est une voie multiplexée. Plusieurs lettres sont acheminées par cette voie. Il s'agit d'un multiplexage par regroupement de plusieurs messages pour former une entité unique, qui seule sera acheminée par les protocoles de communication. Le traitement porte sur l'entité sac postal en bloc, au lieu de devoir manipuler chaque lettre. Cet exemple montre un autre intérêt du multiplexage : le regroupement de plusieurs messages ayant une destination commune sur une même voie permet de réduire le coût de traitement. On appelle parfois conduit un tel regroupement.

Un moyen simple et fréquemment utilisé consiste à attribuer à chaque source une étiquette (numéro de circuit logique de X25, de Frame Relay ou de ATM par exemple) qui est ajoutée à chaque message émis par une source. Chaque source dispose d'une étiquette différente. L'étiquette identifie à l'arrivée, pour la fonction de démultiplexage, le destinataire du message, autrement dit le SAP d'arrivée du flux produit par la source.

L'Ether est partagé en bandes de fréquence afin de pouvoir acheminer simultanément un grand nombre de canaux. Il s'agit d'un multiplexage dans l'espace. Une assemblée partage le temps de parole entre les individus présents en donnant la parole à tour de rôle aux opérateurs. Il s'agit d'un multiplexage temporel, puisque l'on partage le temps.

Le multiplexage est statique si la fraction de voie affectée à chaque voie multiplexée l'est de manière permanente. Que la voie multiplexée soit ou non utilisée, sa fraction de ressource (voie, canal...) est réservée et ne peut être utilisée par aucune autre voie multiplexée. C'est le cas par exemple des voies en radio (canal de fréquence réservé). Le multiplexage est dynamique si la fraction de voie est affectée de manière temporaire. La fraction de voie peut être utilisée par d'autres voies si la voie multiplexée n'est pas utilisée.

Il se peut que le débit, somme des voies sources multiplexées, excède temporairement la capacité de la voie support. On parle de multiplexage statistique lorsque l'on se base sur la somme des débits moyens pour allouer la voie. En moyenne, la somme des débits sources est égale ou inférieure au débit de la voie de sortie. Un réservoir, un tampon mémoire, doit pouvoir absorber les surcharges temporaires.

La route et le chemin de fer sont deux exemples de voies multiplexées. Le chemin de fer utilise un multiplexage temporel strict. Chaque segment de voie ferrée est affecté à un train (convoi ou message) pendant une période de temps, à l'exclusion de tout autre. Au contraire, la route n'exclut aucun véhicule du droit de circuler. Donc, chaque véhicule (message) se déplace quand il le souhaite. Si les véhicules se répartissent judicieusement dans le temps, la route peut écouler plus de trafic que la voie ferrée. Par contre, quand une quantité de véhicules supérieure à la capacité de la route se présente, une congestion (embouteillage) se produit. La voie ferrée, grâce à son multiplexage statique, ne court pas ce risque. Vous trouverez fréquemment le choix dans les solutions réseau entre :

- une réservation a priori (statique) qui garantit aux usagers l'accès à la ressource mais qui gaspille en moyenne la capacité,
- une solution probabiliste, sans réservation, qui en moyenne optimise l'usage de la ressource au risque de congestions temporaires.

2.3.3.1. *Multiplexage temporel*

Le multiplexage temporel est une technique utilisée intensivement dans les applications temps réel et en téléphonie. La voie de communication est partagée en tranches de temps constantes. Par exemple la tranche de temps utilisée pour le réseau téléphonique est de 125 μ s. Ces tranches se répètent à l'identique dans le temps sur la

	Statique	Dynamique	Mixte
Temporel	Emissions radios ou télévisions planifiées à l'avance	Communication orale entre humains	Téléphonie
Spatial	Allocation des fréquences hertziennes aux opérateurs	Radio amateurs	Allocation des canaux hertziens par les contrôleurs aériens pour la communication avec les avions en vol. Canal d'appel prédéfini plus groupe de canaux alloués dynamiquement

Figure 2.35. Exemples des multiplexages possibles sur une voie

voie de communication. Chaque tranche est elle-même découpée en 32 intervalles temporels. Chaque intervalle temporel contient un octet. La figure 2.36. montre le multiplex temporel téléphonique connu sous le terme MIC 32. Une communication téléphonique produit 1 octet (échantillon du signal vocal numérisé) toutes les 125 µs. Pour chaque communication téléphonique établie, un intervalle de temps est alloué parmi les 32 disponibles. Les échantillons sont toujours transmis dans le même intervalle temporel pendant toute la durée de la communication.

2.3.3.2. Concaténation/groupage

Cette fonction permet dans une entité de mettre bout à bout dans un même paquet plusieurs messages soumis à cette entité. La fonction inverse est appelée séparation. Le groupage est similaire (cf. figure 2.38.).

2.3.3.3. Eclatement

Une entité qui dispose de plusieurs voies pour atteindre une autre entité peut disperser les paquets sur les différentes voies dont elle dispose. Cela lui permettra ainsi d'utiliser de manière cumulative la bande passante des voies dont elle dispose. On appelle éclatement ce procédé. Il permet soit d'augmenter le débit entre deux entités, soit d'améliorer la fiabilité. A l'arrivée, une fonction de réassemblage est utilisée. Une

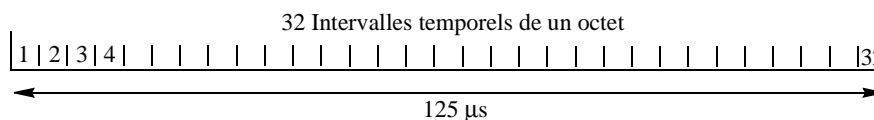


Figure 2.36. Structure d'un multiplex temporel (MIC 32)

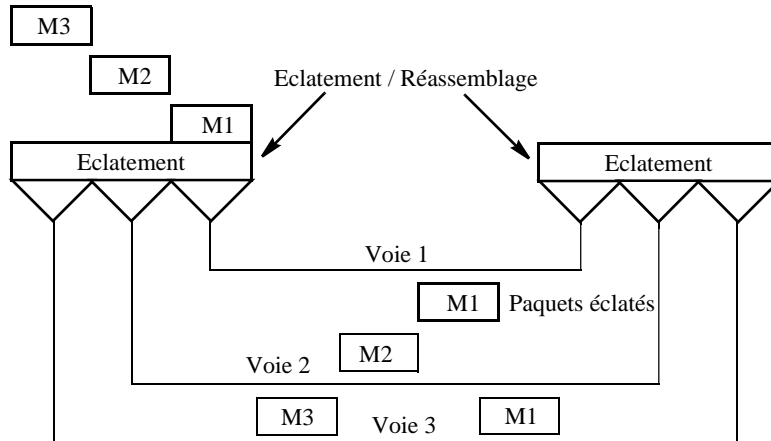


Figure 2.37. Eclatement du trafic sur plusieurs voies

fonction de remise en ordre à l'arrivée peut aussi être nécessaire du fait des pertes qui peuvent se produire sur les voies, ou du fait que les voies n'ont pas le même débit, ou encore que les fragments n'ont pas la même taille. La figure 2.37. montre ce procédé. Cette fonction peut être vue comme l'inverse du multiplexage.

2.3.4. Interconnexion de voies

En général il n'y a pas de voie directe entre deux équipements ou abonnés du réseau. Par assemblage en cascade de voies, on peut prolonger une voie simple et relier une source avec un puits qui n'ont pas de voie commune directe (cf. figure 2.39.). L'équipement C qui assure la connectivité, continuité, des deux voies V_1 et V_2 assure dans ce cas une fonction de répétition ou relaiage. C'est le cas, par exemple, de la diffusion hertzienne. Une antenne a une portée limitée et des stations de relaiage assurent la couverture de nouvelles zones géographiques.

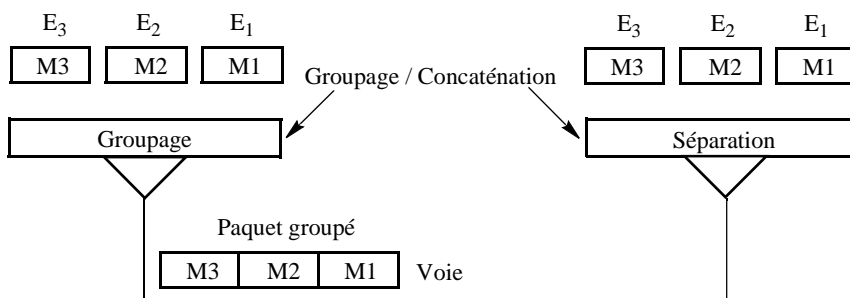


Figure 2.38. Groupage/concaténation du trafic sur une voie

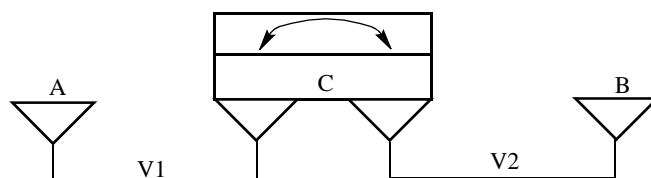


Figure 2.39. Assemblage en cascade réalisant la connectivité entre deux abonnés A et B

2.3.4.1. Fonction de commutation

En général les interconnexions sont des carrefours qui relient plus de deux voies. On appelle nœud de commutation, commutateur ou routeur un composant qui interconnecte plus de deux voies (figure 2.40.). Pour assurer la connectivité, c'est-à-dire la propriété d'atteindre tout abonné du réseau, des réseaux maillés doivent être construits.

Les nœuds de commutation assurent l'aiguillage des messages arrivant sur une voie vers la voie de sortie qui permettra au message reçu d'atteindre sa destination. Les nœuds de commutation assurent le choix des chemins à un carrefour. Pour cela, il y a nécessité d'un adressage : chaque abonné du réseau susceptible de recevoir des messages doit avoir une adresse unique. Pour chaque lettre postée, une adresse doit être fournie. Cette adresse correspond à une boîte aux lettres d'arrivée unique. De même, le numéro de téléphone correspond à un abonné unique. Cela ne veut pas dire dans les deux exemples qu'un individu ne puisse pas avoir plusieurs adresses. Cela implique seulement qu'un message ne peut aller qu'à une seule adresse.

Nous excluons à ce stade de notre exposé la possibilité de diffuser une même information à un groupe de destinataires possédant ou non la même adresse, ce service n'existe ni à la Poste ni au téléphone. Il existe toutefois dans un service de radiodiffusion, mais dans ce cas il n'y a pas de routage, et la voie d'accès est directe. Si la voie n'est pas directe, un schéma de relayage du type de la figure 2.39. assure la rediffusion.

Chaque abonné possède une adresse. On appelle fonction de commutation les règles qui permettent de choisir le chemin pour atteindre une destination, un abonné désigné par son adresse. La fonction de commutation est mise en œuvre dans chaque nœud de commutation. Un centre de tri des PTT est un nœud de commutation, le postier ou la machine de tri exécutent la fonction de commutation. On peut schématiser cette fonction de la manière simpliste suivante :

- lire l'adresse,
 - selon destination, mettre la lettre dans la case correspondant à cette destination.
- Pour cela la fonction de commutation utilise une « base d'information » renseignée par l'algorithme de routage.

Dans un nœud de commutation, la case de dépôt est une voie de sortie du nœud. Il existe une case particulière, le SAP, ou boîte aux lettres locale, pour les lettres qui sont

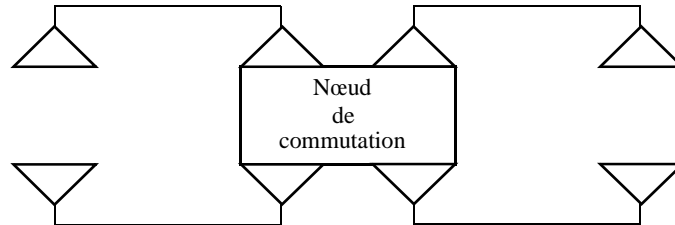


Figure 2.40. *Équipement d'interconnexion de quatre voies*

destinées aux abonnés relevant de ce centre de tri (nœud de commutation). On peut aussi considérer le facteur qui distribue les lettres comme le centre de tri final. En fait il achemine physiquement les lettres dans la boîte aux lettres de chaque destinataire.

Notez que la fonction de commutation n'est pas nécessairement finie car une boîte aux lettres correspondant à une famille ou une société (adresse unique pour la Poste) est le point d'accès SAP de l'interface d'entrée. Il peut y avoir plusieurs individus ou services différents derrière cette adresse. La fonction de commutation est une fonction que nous retrouverons donc plusieurs fois sous des termes différents. Le démultiplexage, que nous avons vu au paragraphe précédent, est somme toute un cas particulier du routage. On préfère distinguer ces deux notions et réserver le terme routage à l'acheminement de messages vers leur destination finale.

2.3.4.2. L'algorithme de routage

L'algorithme de routage est un mécanisme pouvant élaborer le chemin pour aller d'un point à un autre indépendamment de l'existence de messages. Cet algorithme permet de renseigner les tables de routage ou vecteurs de routage qui autorisent l'exécution de la fonction de commutation. L'algorithme de routage n'est pas, en général, exécuté pour chaque message. Il est invoqué périodiquement pour élaborer les routes ou chemins disponibles. Généralement l'algorithme de routage cherche de « bons » chemins, bon au sens de chemin de meilleur coût.

Vous effectuez vous-même un algorithme de routage lorsque vous cherchez sur une carte les différents itinéraires possibles. Le résultat de l'algorithme est un ou plusieurs itinéraires (chemins) possibles. Lorsque vous circulez en voiture, à un carrefour vous effectuez la fonction de commutation si vous décidez et prenez la route suivante en fonction de critères très variés propres à votre humeur : beauté du paysage, prix de la voie, importance de la file d'entrée dans la voie (la plus courte toujours, par exemple)... La connaissance des itinéraires vous autorise ce choix sous réserve que vous ayez noté ces indications au moment de la construction des itinéraires (pendant l'exécution de l'algorithme de routage).

2.3.4.3. Maillage d'un réseau

Pour assurer la connectivité, c'est-à-dire la propriété d'atteindre toute adresse, il faut construire des réseaux maillés. On appelle réseau maillé un assemblage complexe de voies et de nœuds de commutation. Les nœuds de commutation sont des éléments

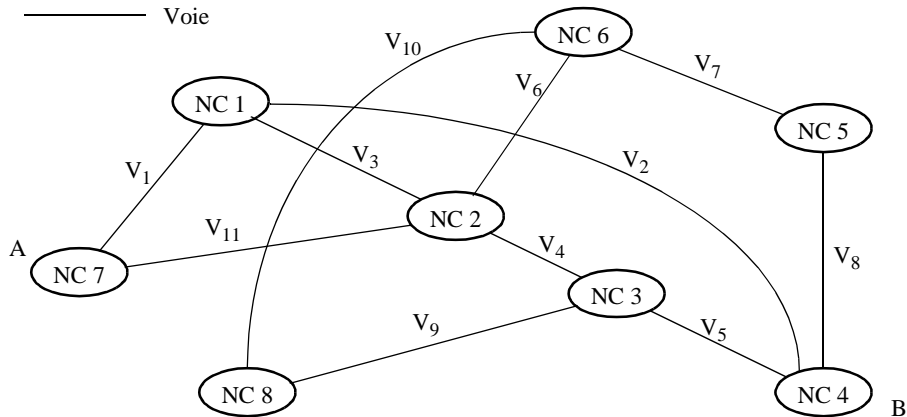


Figure 2.41. Exemple de réseau maillé composé de huit nœuds et de onze voies

essentiels des réseaux de transmission de données. Tout réseau construit à partir d'un assemblage de voies point à point est un réseau maillé. La figure 2.41. montre un exemple de réseau maillé composé de 8 nœuds de commutation et de 11 voies. Les abonnés, noté A, B... sont supposés être raccordés directement aux nœuds de commutation.

Dans un réseau maillé, il existe plusieurs chemins. La construction du maillage permet de tolérer des pannes de voie en créant plusieurs chemins pour aller d'un point à un autre. Sur la figure 2.41. il existe au moins deux chemins pour aller de tout nœud vers tout autre nœud. La perte de la voie 1 n'interdit pas à A de continuer à communiquer avec le reste du réseau. Par contre, si V_1 et V_{11} sont simultanément en panne, les abonnés du nœud 1 sont isolés du réseau.

Sur la figure 2.41. il existe de nombreux chemins entre A et B. En voici quelques-uns :

- NC 7 - V_1 - NC1 - V_2 - NC4
- NC 7 - V_1 - NC1 - V_3 - NC2 - V_4 - NC3- V_5 - NC4
- NC 7 - V_{11} - NC2 - V_4 - NC3- V_5 - NC4

En voici un autre plus « bucolique »

- NC 7 - V_1 - NC1 - V_3 - NC2 - V_6 - NC6 - V_{10} - NC8 - V_9 - NC3- V_5 - NC4

Deux chemins sont différents s'ils utilisent au moins une voie ou un nœud de commutation différent.

2.3.4.4. Commutation de lettres, paquets, messages (mode datagramme ou

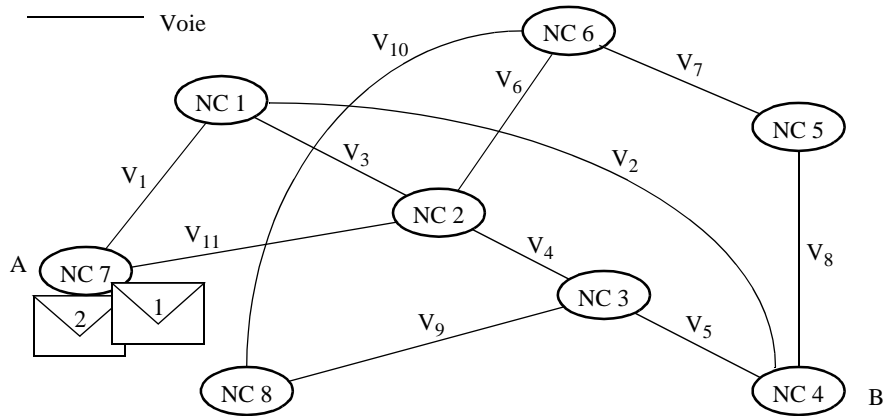


Figure 2.42. Exemple de cheminement de lettres dans un réseau maillé composé de huit nœuds et de onze voies

lettrogramme)

Chaque nœud de commutation (centre de tri) utilise une partie de l'information d'adresse pour acheminer (router) la lettre vers le destinataire final en utilisant les voies de sortie dont il dispose. Il n'y a aucun chemin préétabli entre la source et la destination. Chaque lettre doit contenir les informations d'adresses (adresse destination, ou chemin pour aller vers la destination). Les centres de tri traitent chaque lettre indépendamment les unes des autres. Il n'y a pas de réservation de ressources pour les différents flux de courrier entre sources et destination. Ainsi, les figures 2.42. et 2.43. montrent deux lettres issues de A à destination de B qui suivent des chemins différents.

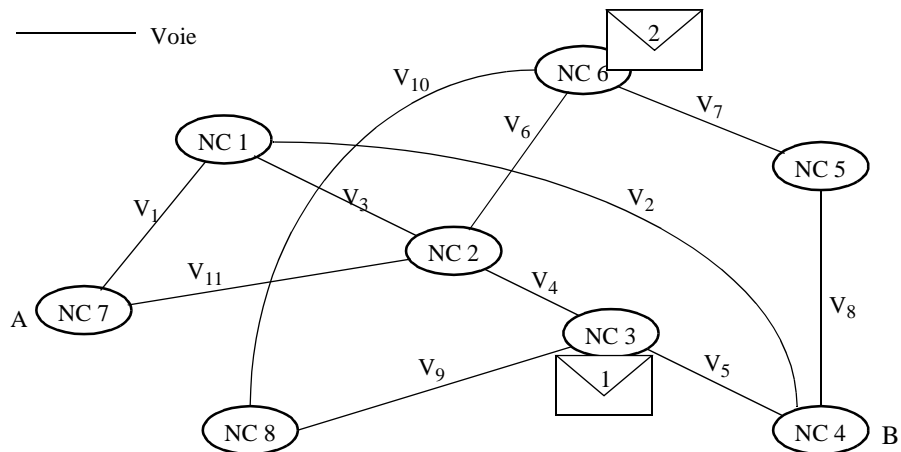


Figure 2.43. Exemple de cheminement de lettres dans le réseau maillé exemple

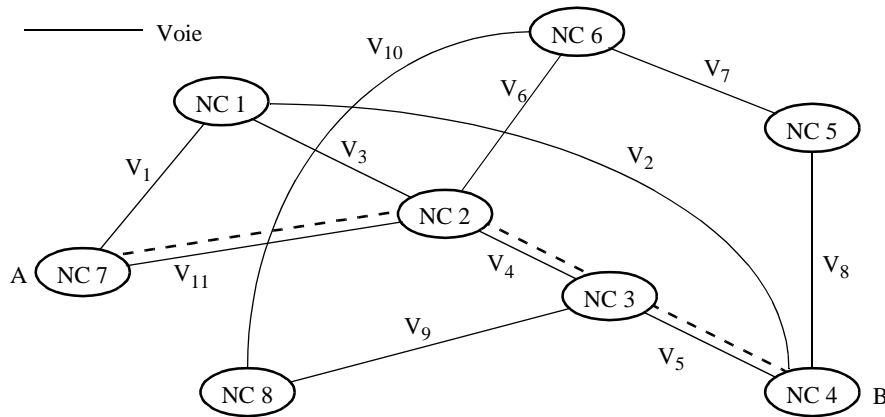


Figure 2.44. Exemple de circuit virtuel établi

Chaque nœud de commutation ayant plusieurs choix possibles décide en fonction de critères¹ qui lui sont propres. Ces critères sont nombreux : voie la moins chère, la plus rapide, la moins chargée, la plus sûre... Pour la Poste, ce peut être : train, avion ou bateau. NC1, NC2 ont deux possibilités de choix selon des critères tels que : voie rapide, voie bon marché, voie sécuritaire...

2.3.4.5. Commutation de circuits

L'ensemble des voies utilisées pour transmettre les données sont réservées à l'ouverture de la communication.

Phases d'une communication :

1. Etablissement du chemin avant toute communication de données. Un chemin ou circuit virtuel est défini et réservé (cf. figure 2.44.). L'algorithme de routage est éventuellement exécuté pour trouver le chemin.

2. Transfert des informations. Toutes les informations passeront par ce chemin ou circuit virtuel. Il n'y a plus lieu d'exécuter l'algorithme de routage. Seule la fonction de commutation est exécutée.

3. Libération des ressources réservées.

Dans chaque nœud de commutation, la mémoire de votre circuit virtuel est conservée et des ressources lui sont affectées. Ces éléments constituent le contexte de la connexion. Une fraction de voie est aussi réservée pour votre circuit virtuel. En pratique, les voies sont partagées (multiplexées) entre les différents circuits virtuels. Qu'il y ait ou non transfert d'information, les ressources (voies et mémoires dans les

1. Ces critères sont généralement renseignés par l'algorithme de routage.

nœuds) sont monopolisées entre la phase 1 et la phase 3. Le réseau téléphonique fonctionne typiquement sur ce principe.

On dira qu'une communication se fait en mode connecté s'il y a obligation d'établir la communication avec le correspondant.

2.3.5. Contrôle de flux

On appelle contrôle de flux le mécanisme qui évite que l'émetteur ne produise ses données plus rapidement que permis. Cela pourrait conduire le récepteur à perdre des données si l'émetteur les produit plus vite qu'il ne peut les consommer. On distingue deux types de contrôle de flux : le contrôle de flux à l'interface entre la source et l'interface, le contrôle de flux de bout en bout entre la source et le puits.

2.3.5.1. A l'interface

L'utilisateur veut soumettre plus de données que la voie de communication¹ ne peut ou n'accepte d'en prendre en charge. Ainsi, si un abonné a souscrit un abonnement à un débit D, le contrôle de flux à l'interface va faire respecter ce débit à l'utilisateur même si la capacité du réseau pourrait soutenir un débit plus important. Il est évident que ce contrôle de flux est nécessaire si la capacité du réseau est limitée au débit D. L'utilisateur ne peut soumettre un débit plus élevé que D, sinon il va perdre des données. Ainsi, votre clavier de Minitel n'accepte que 75 b/s et, si vous tapez plus de 7 caractères par seconde, ceux en supplément seraient perdus.

2.3.5.2. De bout en bout

L'émetteur produit les données plus vite que le destinataire ne peut les traiter, sans pour autant saturer le réseau. Il faut qu'un mécanisme alerte la source afin qu'elle arrête temporairement de produire des données. C'est ce que fait votre ordinateur avec votre clavier quand vous tapez trop vite des caractères. Un caractère spécial « Xoff », envoyé par l'ordinateur au clavier, bloque le clavier afin de cesser la prise en compte de ce qui est tapé. Vous ne voyez plus votre frappe s'afficher, et en principe vous arrêtez de taper. Un second caractère « Xon » débloque le clavier.

2.3.6. Informations de service

Ce sont les informations nécessaires à la gestion de la communication. Les adresses sur l'enveloppe font partie des informations de service. De manière générale, les protocoles ajoutent des informations de service au message pour pouvoir remplir leur mission. Dans l'exemple d'omelette décrit au début de ce chapitre, la

1. On utilise aussi le terme prestataire de services pour désigner le service fourni par réseau. Ce service réseau est toujours une voie de communication. Les propriétés du service diffèrent selon les offres.

numérotation « *i/j* » ajoutée sur chaque lettre est une information de service liée à ce protocole. L'information de service n'a pas d'intérêt pour l'utilisateur du service. Elle est, en général, ajoutée par le côté émetteur du protocole et retirée par le côté récepteur avant de délivrer le message au destinataire.

Mais les informations de service ne se limitent pas aux seules informations ajoutées aux messages. Par exemple, pour effectuer l'établissement de la communication, le prestataire de services doit échanger des informations entre ses composants. Ces informations sont transparentes à l'utilisateur. L'utilisateur du service ne verra jamais ces informations à son interface. Elles ont pourtant un effet bien réel sur les performances du protocole, et donc sur le débit offert par le service à ses utilisateurs.

2.3.7. Service avec séquençement

C'est un service qui garantit que les données échangées entre deux utilisateurs A et B arriveront dans le même ordre que celui auquel elles sont soumises. On a vu qu'un service à datagramme ne garantit pas que les données arrivent dans le même ordre que celui soumis par la source. Certaines applications, le transfert de la voix par exemple, requièrent que les blocs d'information soient transmis et reçus dans le même ordre. On appelle séquençement cette propriété. Un transfert de fichier requiert aussi cette propriété de séquençement, parce qu'il n'est pas question que le contenu d'un fichier programme vous rende les lignes de code dans un ordre arbitraire. Par contre, une application de messagerie n'impose pas en général de contrainte de séquençement.

En général, ce service est fourni par les services en mode connecté.

2.3.8. Rendement d'un protocole

On appelle rendement d'un protocole le rapport entre la quantité d'information soumise à la transmission par l'utilisateur (débit utile) et la quantité effectivement transmise (débit effectif) sur la voie utilisée.

Rendement d'un protocole : débit utile/débit effectif.

2.4. Conclusion

Ce chapitre a présenté un grand nombre de définitions et de concepts utilisés dans les réseaux. Ils vont être développés et illustrés dans les chapitres suivants.

Les notions de délai d'acheminement et de parallélismes sont essentielles. Ils seront en permanence présents dans tous les services et architectures de réseau que vous rencontrerez ultérieurement. Bien que très simples, ces principes motivent l'essentiel des efforts de recherche et de développement qui sont faits actuellement pour les autoroutes de l'information. Cela est dû au fait que les caractéristiques des solutions techniques qui ont permis de construire les services et architectures réseau

des années 1970 et 1980 ont été profondément modifiées à la fin des années 1980. Conséquence : l'effort de conception des protocoles qui portait dans les années 1970 - 1980 sur la réduction du taux d'erreur et l'optimisation de l'utilisation des voies physiques s'est reporté sur le délai de traitement dans les entités. La durée d'émission était le facteur déterminant dans les délais d'acheminement. Aujourd'hui, la durée d'émission est devenue beaucoup plus courte que les durées de traitement et les taux d'erreur sur les voies physiques en fibre optique sont devenus très faibles.