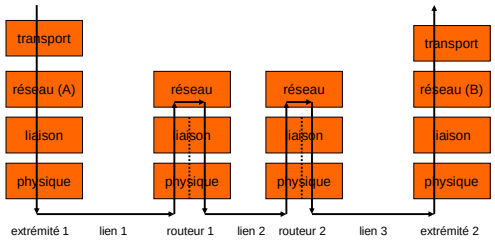
Chapitre 4 La couche réseau



1

1. Problèmes de conception de la couche réseau

- 1.1. Service fourni à la couche transport
  - interface entre le client (utilisateur) et l 'opérateur réseau (fournisseur)
  - indépendant des techniques utilisées dans les sous-réseaux
  - les adresses réseau utilisent un plan uniforme de numérotation

2

1.1. Service fourni à la couche transport

- services en mode connecté (Tel., X.25, RNIS, ATM)
  - service fiable
  - établissement d 'une connexion
  - négociation des paramètres de QoS et de coût
  - transfert de données bidirectionnel et en séquence
  - contrôle de flux
- services en mode non connecté (Internet)
  - qualité « best effort » (non fiable)
  - la fiabilisation est assurée par la couche transport
  - chaque paquet comporte l 'adresse complète de destination

3

1.1. Service fourni à la couche transport

- services en mode connecté
  - complexité au niveau Réseau (proche en proche)
  - transport déchargé des mécanismes de contrôle
  - adapté aux services « synchrones » (audio, vidéo) grâce aux mécanismes de réservation de ressources
- services en mode non connecté
  - complexité au niveau Transport (bout en bout)
  - réseau déchargé des mécanismes de contrôle
  - adapté aux services « asynchrones » (données) ou aux transactions (requêtes / réponses)

4

1.2. Organisation interne de la couche réseau

- protocoles en mode connecté (CV)
  - création d 'un circuit virtuel (CV)
  - la route est fixée à l 'établissement du CV
  - les routeurs ont une table avec une ligne par CV
  - tous les paquets suivront ce chemin jusqu'à la libération du CV
  - plus fiable
  - Problème du reroutage d'un CV
- protocoles en mode non connecté (datagramme)
  - les paquets (datagrammes) sont routés indépendamment les uns des autres
  - les routeurs ont une table avec une ligne pour chaque destination
  - plus robuste (résistance aux pannes)

5

1.2. Organisation interne de la couche réseau

- Philosophie des circuits virtuels
  - chaque routeur tient à jour une table avec une entrée pour chaque CV qui le traverse
  - chaque paquet contient un n° de CV (remplace l 'adresse)
  - quand un paquet arrive dans un routeur,
    - Commutation suivant table :
      entrée, n° CV_entrant => sortie, n° CV_sortant
  - le n° de CV est local à un couple de nœuds
  - exemple de création de CV dans un sous-réseau
  - cas de collision des n° de CV:
    - si deux demandes d 'initialisation de CV se croisent sur deux routeurs adjacents, chaque routeur choisit un n° de CV full duplex. La collision se produit si les routeurs choisissent le même n° de CV

6

2.1. Routage

- Le routage des paquets :
  - Plusieurs phases :
    - 1. obtenir des informations sur le réseau
      - **topologie, ressources**
    - 2. déterminer une route pour atteindre la destination
      - **=> table de routage (RIB : Routing Information Base)**
    - 3. configurer les routeurs (acheminement)
      - **=> table de commutation (FIB : Forwarding Information Base)**
    - 4. puis commuter les paquets
  - 1 à 3 : plan de contrôle
  - 4: plan de données

7

Algorithmes de routage

- Permettent la construction de RIB puis de FIB

- 2.1. Critères de routage
  - Assurer l'acheminement du paquet à destination
    - sans boucle
  - minimiser le délai de transmission
  - maximiser le débit
  - prévenir la congestion
  - assurer l 'équité entre utilisateurs

8

2.1. Critères de routage

- gérer les défaillances du réseau
- s 'adapter aux modifications du réseau
- stabilité
- certains critères peuvent être contradictoires :
  - justice et optimisation
  - minimiser le délai moyen de traversée des paquets et maximiser le flux total du réseau
- solution de compromis :
  - minimiser le nombre de sauts (routeurs traversés)

9

Classement des algorithmes de routage

- non adaptatifs :
  - les routes sont calculées à l 'avance puis téléchargées dans les routeurs à l 'initialisation du réseau
  - routage statique
- adaptatifs :
  - modifient leurs décisions de routage pour refléter les changements de topologie ou de trafic dans le réseau
  - routage dynamique
- Les algorithmes adaptatifs diffèrent selon :
  - l'endroit où ils se procurent leurs infos de routage
  - l'instant où ils changent de route
  - la métrique utilisée

10

Classement des algorithmes de routage

- Centralisé
  - un nœud spécialisé calcule les chemins entre chaque couple de nœuds puis distribue les tables de routage
  - engendre un trafic de signalisation important
  - Permet une vue cohérente et complète (« route server »)
- Distribué
  - chaque nœud construit « indépendamment » sa RIB

11

2.2. Techniques distribuées

- 2.2.1. Routage par inondation (non adaptatif)
  - chaque paquet entrant est émis sur toutes les lignes de sortie sauf sur la ligne d'arrivée
  - génère un très grand nombre de paquets dupliqués
  - techniques d'arrêt de l'inondation (ex. TTL)
  - utilisé dans les protocoles de diffusion ou dans les mises à jour de BD réparties (ex. OSPF)
  - trouve le meilleur chemin

12

2.2. Techniques distribuées

- 2.2.2. Routage adaptatif
  - chaque routeur adapte sa table de routage en fonction d 'information locales, mais aussi globales
    - signalisation de routage avec les autres noeuds
  - Routage à **vecteur de distance**
    - chaque routeur possède une vue partielle du réseau
    - ex RIP (Routing Information Protocol)
  - Routage par **informations d'état des liens**
    - chaque routeur possède une vue globale du réseau
    - ex OSPF (Open Shortest Path First), IS-IS

13

2.2.2.1. Routage à vecteur de distance

- chaque routeur dispose d'une table de routage précisant, pour chaque destination, la meilleure distance connue et par quelle ligne l'atteindre
- vecteur de distance : sous-ensemble de la table de routage : ensemble des couples (destination, coût)
- les routeurs s 'échangent périodiquement leurs vecteurs de distance
- agorithme de Bellmann-Ford ou Ford-Fulkerson
- utilisé par RIP (Internet), IPX (Novell) et réseaux Appletalk
- la métrique généralement utilisée est le nombre de sauts

14

2.2.2.1. Routage à vecteur de distance

- **Algorithme de Bellman-Ford distribué**
  - d(i,j) : distance entre deux entités adjacentes i et j
  - D(i,j) : distance entre deux entités quelconques i et j
  - $D(i,j) = \text{Min}_k [d(i,k) + D(k,j)]$ pour $i \neq j$
  - converge vers une estimation correcte de D(i,j) en un temps fini en l'absence de changements topologiques
  - est exécuté par tous les routeurs
  - se déroule de façon asynchrone
- Chaque routeur envoie périodiquement son vecteur de distance à tous ses voisins :
  - { (k,D(i,k)) pour toute destination connue k }
    - où D(i,k) est l'estimation actuelle de i
  - Chaque routeur i maintient une table (RIB) :
    - (k,r,n) où n=D(i,k) et r est le prochain saut calculé vers k

15

2.2.2.1. Routage à vecteur de distance

- Algorithme de Bellman-Ford (suite) :
  - Lorsque le routeur R reçoit un vecteur en provenance d'un voisin R'' :
  - pour tout couple (A,n'') du vecteur :
    - calculer n' = n '' + d(R, R'')
  - si la table de R contient une ligne (A, R', n) alors :
    - 1. si R' = R'' et n = n' alors pas de changement
    - 2. si R' = R'' et $n \neq n'$ alors mise à jour de la distance (qu'elle soit inférieure ou supérieure)
    - 3. si $R' \neq R''$ et $n \leq n'$ alors pas de changement (stabilité)
    - 4. si $R' \neq R''$ et $n > n'$ alors
      remplacer (A, R', n) par (A, R'', n ')
  - sinon
    - Ajouter une ligne (A,R'',n')

16

2.3.3.1. Routage à vecteur de distance

- Exemple de RIP (RFC 1058)
  - Initialisation des tables de routage
  - Cas de rupture d'une liaison
  - Comptage à l'infini
    - => limiter la métrique maximum
  - Problème des boucles transitoires de routage
  - Optimisations :
    - Horizon partagé (retour empoisonné) :
      => envoyer une distance ∞ au prochain saut vers une destination D
      => évite les boucles à 2 nœuds
  - Problème des boucles (d'au moins 3 nœuds) long à se résorber :
    - mises à jour déclenchées :
      => accélèrent la convergence

17

2.3.3.1. Routage à vecteur de distance

- Exemple de RIP (suite)
  - RIP utilise UDP sur le port n° 520
  - intervalle de transmission des tables : 30 sec
  - durée du timer associé à chaque destination : 3 min
    - après ce délai sans rafraîchissement
    - => destination devient inaccessible (distance = ∞)
  - dans RIP, ∞ = 16

18

2.2.2.2. Routage à état des liens

- Principe
  - lien = connectivité entre
    - routeur et réseau ou entre 2 routeurs
    - = arête d'un graphe
    - sommet du graphe : routeurs et réseaux
  - chaque routeur détermine
    - état des liens adjacents (local)
  - A chaque changement d'état d'un lien local
    - routeur initie **diffusion** nouvel état EL à tous les routeurs

19

2.2.2.2. Routage à état des liens

- Diffusion de proche en proche
  - doit être fiable
    - diffusion par inondation
    - état des liens acquittés, numéro de séquence
- A un petit délai près
  - tous les routeurs ont la même base d'état de tous les liens du réseau (BEL)
- A chaque changement de la base BEL
  - chaque routeur calcule
    - + court chemin vers toutes les destinations

20

2.2.2.2. Routage à état des liens

- calcul de l'arbre des + courts chemins
  - => algorithme de Dijkstra
  - racine de l'arbre = routeur qui calcule
    - prochain saut N vers D = fils dans l'arbre qui contient D
  - => RIB

21

2.2.2.3 Adressage réseau

- Permet d'identifier **globalement** les nœuds du réseau
- doit faciliter le routage
  - => relation entre adresse et position dans le réseau
    - Ex : Numéro de sécu, immatriculation, ou de téléphone mobile
      => identifieur global mais pas d'indication de position
    - Ex : adresse IP ou N° téléphone fixe
      => numéro hiérarchique correspondant à un découpage hiérarchique du réseau
    - Tel : 33 3 90 24 4X YZ : France (33), Est (3), Central (24), poste (4X YZ)
      => routage de proche en proche
    - n° IP :130.79.200.1 netmask 255.255.255.0
      => réseau (130.79), sous-réseau (200), machine (1)
- Table de routage
  - destination = réseau ou sous-réseau ou machine
    - => diminuer la taille des tables

22

3. Algorithmes de contrôle de congestion

- Causes de congestion :
  - files d 'attente d 'un routeur pleines
  - faibles performances des routeurs
  - lignes à faible débit
- la congestion tend à s 'accroître :
  - retransmissions de paquets
  - propagation en amont
- Différence entre « congestion » (CC) et « contrôle de flux » (CF)

23

3.1. Principes généraux du CC

- **sous-réseaux à circuit virtuel**
  - CC mis en place dans la couche réseau
    - réservation de ressources dans les routeurs
    - et/ou rétroaction
- **sous-réseaux datagrammes**
  - CC au niveau réseau (TTL, RSVP) : moins performant que pour les réseaux à CV
  - CC au niveau Transport (TCP)

24

3.2. CC dans les réseaux à CV

Ex : réseaux ATM

- contrat lors de l'établissement du CV
- si tous les CV ont un débit constant
  - il suffit de s'assurer à l'établissement du CV
  - pour chaque liaison : somme débit(CV) < débit liaison
  - => **contrôle d'admission des connexions (CAC)** (peut être refusée)
- Si le chemin le plus court est saturé
  - => routage par un chemin plus long (délestage)
- problème plus complexe si les débits sont variables
  - typique applications informatiques
  - réservation débit maximum => gaspillage
  - (problème de multiplexage statistique)

25

3.2. CC dans les réseaux à CV

- Un CV respecte-t-il son contrat ?
  - s'il dépasse => risque de congestion
- Mise en forme du trafic (traffic shaping)
  - l'émetteur s'assure qu'il ne dépasse pas la réservation
  - ex : algorithme « token bucket » (seau à jeton de taille N)
    - 1 jeton est généré à intervalle régulier T => seau
    - si seau plein : jeton ignoré (réserve d'au plus N jeton)
    - pour envoyer un bloc : consommer un jeton du seau (si pas de jeton attendre)
    - débit « moyen » : 1 paquet par intervalle T
    - rafale de N paquets possible

26

3.2. CC dans les réseaux à CV

- Contrôle de **conformité**
  - réseau vérifie si débit (d'un CV) conforme au contrat (sinon congestion possible)
    - algorithme du « seau percé » (« leaky bucket ») T,N
      - tous les temps T un paquet est enlevé du seau (si non vide)
      - quand un paquet arrive
        - si seau non plein => mis dans le seau
        - si seau plein : paquet non conforme (éliminé ?)
    - problème statistique :
      - taille des différents seaux pour éviter la congestion
      - files d'attente => délais variables (gigue)

27

4. Protocole IP (Internet Protocol)

- Défini dans le RFC 791
  - envoi de datagrammes (paquets IP)
    - de bout en bout à travers des (sous-)réseaux interconnectés (internet)
    - sous réseaux hétérogènes (débit, taille trames, adressage)
  - IP = couche unificatrice
    - IP au dessus de tout protocole (LAN, WAN, satellite, connecté ou non, …)
    - règles d'encapsulation du paquet IP dans une trame liaison
  - peu de fonctions complexes (à part fragmentation)
    - pas de retransmission (non fiable)
    - déséquencement possible
    - pas ou peu de contrôle de flux/congestion
  - modèle de service usuel : « best effort »
  - fonctions évoluées dans la couche transport (TCP)

28

Format du paquet IP

- Entête obligatoire (20 octets) + options facultatives + données (total < 2^{16} octets : 64K)
- **En-tête du paquet IPv4**
  - version : n° de version courante (4)
  - Lg_ent : longueur de l'en-tête (y compris option) en mots de 32 bits (20 octets sans options)
  - type de service : priorité du paquet et type de routage souhaité (par défaut non utilisé)
  - longueur totale : longueur du paquet en nombre d'octets
  - identification, drapeaux, dép_fragment : utilisés par les procédures de fragmentation et réassemblage

29

En-tête IPv4 (suite)

- durée de vie (TTL) : durée de vie maximale d'un paquet en nombre de secondes (décrémenté dans chaque routeur)
- protocole : n° du protocole (de transport) auquel il faut remettre le paquet IP, lorsque la destination est atteinte
  - ex TCP, UDP, ICMP,…
- total de contrôle d'en-tête : contrôle de redondance sur l'en-tête
- adresse IP source (4 octets)
- adresse IP destination (4 octets)
- options IP (éventuelles)
- bourrage : permet que la longueur de l'en-tête soit un multiple de 32 bits

30

IP : Type de service (ToS)

- priorité :
  - valeurs codées de 0 à 7
- bits D, T, R : qualité de service d'acheminement désirée
  - D = 1 : demande un délai d 'acheminement court
  - T = 1 : demande un débit de transmission élevé
  - R = 1 : demande une grande fiabilité
- ces demandes de service d'acheminement ne sont pas perçues comme des exigences
- En standard le champ « type de service » est ignoré dans la plupart des routeurs
- sauf maintenant par « Diffserv » où le champ TOS est redéfini par le champ **DSCP** (« differentiated services codepoint ») qui permet de coder la classe de service

31

IP : Fragmentation et réassemblage

- un paquet IP est encapsulé dans les trames adaptées aux différents sous-réseaux qu'il traverse
- chaque sous-réseau autorise une taille maximum : MTU (Maximum Transfer Unit)
  - MTU Ethernet : 1500 octets, MTU FDDI : 4470 octets
- la fragmentation vise à assurer l'indépendance entre le paquet IP et le MTU des sous-réseaux traversés
  - MTU minimum : 576 octets
- un fragment peut être lui même re-fragmenté
- le réassemblage se fait à l'arrivée
- fragmentation et réassemblage diminuent les performances
  - calculs dans les routeurs

32

IP : Fragmentation et réassemblage

- En-tête et fragmentation
- Drapeaux :
  - bit DF « Don 't fragment » :
    - si DF = 1 alors le paquet ne doit pas être fragmenté
    - si DF = 0 alors la fragmentation est autorisée
  - bit MF « More fragments » :
    - si MF = 1 alors il y a encore des fragments derrière
    - si MF = 0 alors c'est le dernier fragment
  - Pour un fragment le champ offset (dep_fragment) indique sa position dans le paquet d'origine
- Tous les fragments du même paquet ont le même identificateur
- remarque : offset = 0 et MF = 0 => paquet non fragmenté

33

IP (suite et fin)

- Eviter la fragmentation
  - découverte du PMTU : « Path MTU discovery »
    - des paquets de taille décroissante sont émis avec DF= 1
    - un paquet trop gros avec DF = 1 est abandonné par routeur
      - envoi message d'erreur ICMP à la source
    - dès que l'on ne reçoit plus d'erreur ICMP, on a trouvé le PMTU
- Options d'IP :
  - source routing (indique la liste des routeurs intermédiaires)
  - route recording : enregistre les routeurs intermédiaires
  - options souvent ignorées à cause du coût de traitement
  - les routeurs traitent plus rapidement des paquets IP dont l'en-tête est de longueur fixe (sans option)

34

IPv6

- Objectifs
  - Supporter des milliards de hôtes
  - Réduire la taille des tables de routage
  - Simplifier le protocole
  - Meilleure sécurité
  - Traiter la qualité de service (trafic « temps réel »)
  - Améliorer la diffusion (« multicast »)
  - Améliorer la mobilité
  - Coexister avec IPv4
  - Compatibilité avec les protocoles existants
    - TCP, UDP, ICMP, IGMP, OSPF, DNS, …

35

IPv6

- Caractéristiques d'IPv6
  - Adresses plus longues (16 octets)
  - Simplification de l'en-tête
    - => traitement par les routeurs plus rapide
  - Traitement des options amélioré
    - Extensions d'en-tête
  - Authentification et confidentialité
  - Meilleure prise en compte de la QoS
- Conclusion
  - Migration progressive d'IPV4 vers IPv6
    - tunnels

36