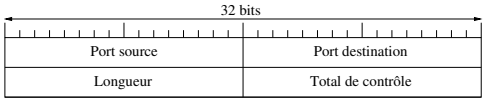
Travaux dirigés

UDP — L'en-tête 1/10

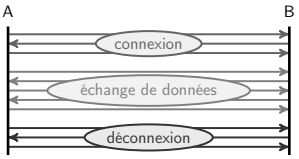
En-tête fixe de 8 octets :



- **Port source** (16 bits) = identifie le processus émetteur
- **Port destination** (16 bits) = identifie le processus destinataire
- **Longueur** (16 bits) = longueur totale du paquet (en-tête + données)
- **Total de contrôle** (16 bits) = code d'erreur calculé sur l'en-tête UDP + une partie de l'en-tête IP

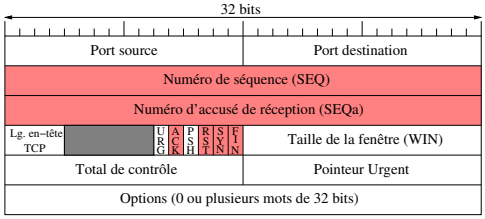
TCP — Déroulement d'une session 2/10

- TCP est un protocole en mode connecté :
  - phase de connexion avant tout échange de données
  - phase de déconnexion une fois l'échange terminé
- La phase de connexion sert :
  - à s'assurer que l'autre processus est prêt à communiquer ;
  - et à échanger des informations nécessaires à la suite de l'échange (dans le cas de TCP : des numéros de séquence).
- La phase de déconnexion sert à libérer des ressources (p.ex., de la mémoire allouée pour le contrôle de l'échange).



TCP — L'en-tête 3/10

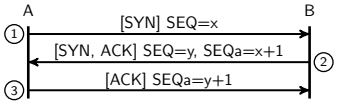
En-tête de 20 octets ou plus :



- **SEQ** (32 bits) = numéro du prochain octet de données envoyé
- **SEQa** (32 bits) = numéro du prochain octet de données attendu
- **ACK** (1 bit) = acquittement
- **RST** (1 bit, ReSeT) = refus de connexion ou déconnexion brutale
- **SYN** (1 bit, SYNchronisation) = demande de connexion
- **FIN** (1 bit, FINalisation) = demande de déconnexion

TCP — Connexion acceptée 4/10

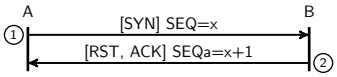
- connexion en 3 temps (three-way handshake)



1. A envoie un paquet
   - de demande de synchronisation (bit SYN=1) ;
   - et contenant un numéro de séquence initial choisi aléatoirement (SEQ=x).
2. B répond par un paquet
   - qui acquitte la demande de A (bit ACK=1 et SEQa=x+1) ;
   - et contient également une demande de synchronisation avec un numéro de séquence initial choisi aléatoirement (SEQ=y).
3. A répond par un paquet qui acquitte la demande de B.

TCP — Connexion refusée 5/10

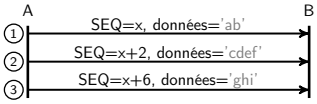
- aucun processus en écoute sur le port de réception du paquet SYN



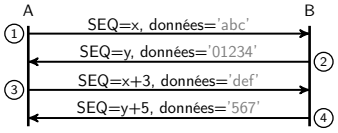
1. A envoie un paquet de demande de synchronisation.
2. B acquitte mais refuse la demande (bit RST=1).

TCP — Échange de données 6/10

- Les octets de données à envoyer sont numérotés.
- SEQ est le numéro de séquence du premier octet de données du paquet.
- Après l'envoi d'un paquet de données, le numéro de séquence est incrémenté du nombre d'octets de données envoyés.

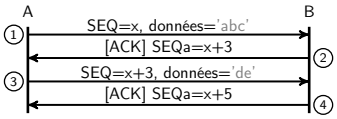


- communication bi-directionnelle ⇒ A et B ont des numéros de séquence indépendants.



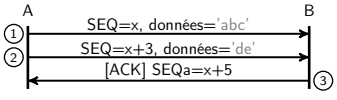
TCP — Acquittements 7/10

- Tout octet de données envoyé doit être acquitté.
- Cela se fait grâce au numéro de séquence acquitté (SEQa).



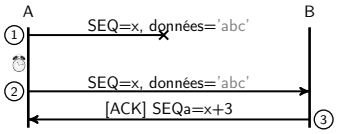
1. A envoie les octets x à $x+2$.
2. B acquitte les octets reçus et indique que le prochain attendu est le $x+3$.
3. A envoie les octets $x+3$ à $x+4$.
4. B acquitte les octets reçus et indique que le prochain attendu est le $x+5$.

- On peut aussi acquitter plusieurs paquets avec un seul acquittement :



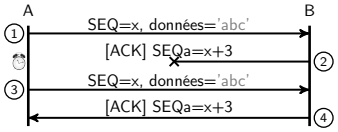
TCP — Temporisations 8/10

- utilisation de temporisation pour détecter les pertes de paquets
- Principe : passé un certain délai, si des données n'ont pas été acquittées, on les considère comme perdues et on retransmet.



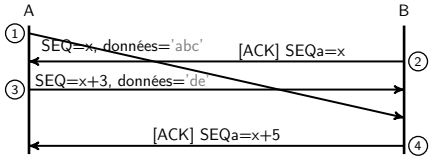
1. A envoie 3 octets de données.
2. Pas d'acquittement dans les temps ⇒ retransmission des 3 octets.

- Autre scénario avec une perte de l'acquittement :



TCP — Déséquencement 9/10

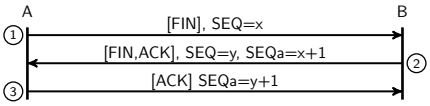
- Possibilité de mémoriser des octets de données non attendus.



1. A envoie 3 octets de données 'abc'.
2. B acquitte mais indique qu'il attend toujours l'octet x.
   Il mémorise les 3 octets.
3. A envoie 2 octets de données 'de'.
4. B acquitte les 5 octets reçus.

TCP — Déconnexion 10/10

- Tout octet de données doit avoir été acquitté avant la déconnexion.
- déconnexion en 3 temps



Exercice 1 — Scénarios d'échange TCP

On considère un client et un serveur TCP. Pour chacun des cinq scénarios ci-dessous, tracez sur un chronogramme les échanges entre les deux processus. Vous indiquerez sur chaque paquet les bits de contrôle de l'en-tête TCP (SYN, ACK, FIN, RST, ...) activés, et les valeurs des compteurs SEQ et SEQa inclus dans cet en-tête. On supposera que le numéro de séquence initial choisi aléatoirement par le client est 100 et que celui choisi par le serveur est 1000.

Q 1.1 La demande de connexion TCP est refusée par le serveur.

Q 1.2 La demande de connexion TCP est acceptée. Le client envoie une requête de 6 octets (`ca va?`). Le serveur répond par une réponse de 4 octets (`yep!`).

Q 1.3 La demande de connexion TCP est acceptée. Le client envoie une requête de 6 octets (`ca va?`). Le serveur répond par deux paquets de données : un premier paquet de 4 octets (`yep!`) et un deuxième paquet de 11 octets (`ca va bien!`).

Q 1.4 Même scénario en supposant que le premier paquet de données envoyé par le serveur (`yep!`) est perdu. On supposera que le client mémorise les paquets arrivés dans le désordre.

Q 1.5 Même scénario en supposant que les deux paquets de données envoyés par le serveur arrivent dans le désordre. On considèrera deux cas différents selon que le client mémorise ou pas les paquets arrivés dans le désordre.

Exercice 2 — Temps de transmission avec UDP et TCP

On considère dans cet exercice deux processus C et S s'exécutant sur deux hôtes d'un réseau Ethernet à 100 Mbit/s. C envoie un message Req de 100 octets à S. S lui répond par un message Rep de 2 000 octets. On rappelle que le MTU sur un réseau Ethernet est de 1 500 octets.

Dans tout l'exercice, on considèrera qu'aucune option IP n'est utilisée et qu'aucun paquet n'est perdu pendant l'échange. Seuls les temps de transmission seront pris en compte : les autres temps, en particulier les temps de propagation, seront considérés comme négligeables.

Dans les questions Q 2.1 à Q 2.4 on s'intéresse d'abord au cas d'UDP.

Q 2.1 Donnez la structure des trames envoyées par C et S qui contiendront des octets de données.

Q 2.2 Déduisez en le nombre maximal d'octets de données pouvant être contenus dans une trame ainsi que la taille (en octets) des trames transmises pour transporter les données.

Q 2.3 Tracez sur un chronogramme les échanges de trames entre C et S.

Q 2.4 Quel est le temps total de l'échange ?

On s'intéresse maintenant (questions Q 2.5 à Q 2.8) au cas de TCP.

Q 2.5 Donnez la structure des trames envoyées par C et S qui contiendront des octets de données.

Q 2.6 Déduisez en le nombre maximal d'octets de données pouvant être contenus dans une trame ainsi que la taille (en octets) des trames transmises pour transporter les données.

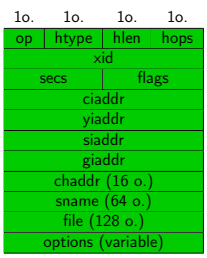
Q 2.7 Tracez sur un chronogramme les échanges de trames entre C et S.

Q 2.8 Quel est le temps total de l'échange ?

Principaux types de messages DHCP 1/12

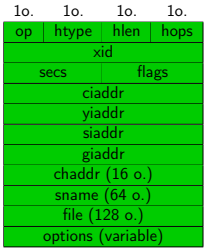
Type	Sens	Contexte(s)
DISCOVER	client→diffusion	demande de bail
OFFER	serveur→client	offre de bail
REQUEST	client→diffusion	acceptation d'une offre
	client→serveur	demande de renouvellement de bail
ACK	serveur→client	REQUEST acceptée
NAK	serveur→client	REQUEST refusée
RELEASE	client→serveur	résiliation du bail (⇔ libération de l'IP)

Structure des messages 2/12



- **op** = type d'opération (0 = req., 1 = rép.)
- **htype** = type d'interface (ex : 1 = ethernet)
- **hlen** = longeur des adresses physiques
- **hops** = nombre de relais visités pour traiter le message. initialisé à 0 par le client
- **xid** = identifiant de transaction. choisi aléatoirement par le client. permet au client de reconnaitre les réponses à ses requêtes
- **secs** = secondes écoulées depuis le début du processus de configuration
- **flags** = 1 si le client ne peut pas accepter de paquets unicast (⇒ le serveur sait qu'il doit envoyer les réponses en diffusion)

Structure des messages 3/12



- **ciaddr** = client IP address (dans un REQUEST de renouvellement ou un RELEASE)
- **yiaddr** = your IP address (dans un OFFER ou un ACK)
- **siaddr** = server IP address (serveur continuant le processus de conf.)
- **giaddr** = gateway IP address (IP du relai ayant retransmis le message)
- **chaddr** = adresse physique du client
- **sname** = nom du serveur
- **file** = chemin du fichier de démarrage (vide si pas de fichier)

(Par défaut, toutes les IP valent 0.0.0.0.)

Structure des messages — Codage des options 4/12

Chaque option est codée sur trois champs :

- code de l'option (1 octet)
- longueur de l'option (1 octet)
- valeur de l'option (selon la longueur)

Par exemple, `03 04 01 02 03 fe` signifie :

- `03` = code de l'option *Router*
- `04` = la valeur de l'option est codée sur 4 octets
- `01 02 03 fe` = valeur de l'option, soit 1.2.3.254

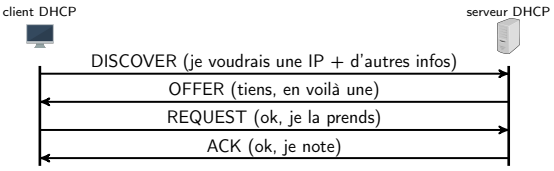
  Rappels :
  - hexadécimal = base 16
  - alphabet hexadécimal = $0, \ldots 9, a, b, c, d, e, f$ ($a = 10, b = 11, \ldots$)
  - donc $fe_{16} = \underbrace{15}_{f} \cdot \underbrace{16}_{\text{base}}^{1} + \underbrace{14}_{e} \cdot \underbrace{16}_{\text{base}}^{0} = 254$

Structure des messages — Options courantes 5/12

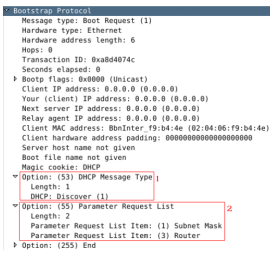
- 53 — **Message type** (obligatoire) — type du message
  1 = DISCOVER 5 = ACK
  2 = OFFER 6 = NAK
  3 = REQUEST 7 = RELEASE
- 255 — **End** (obligatoire, sur 1 octet) — marque la fin des options
- 1 — **Subnet Mask** — le serveur fournit un masque de réseau au client
- 3 — **Router** — le serveur fournit un routeur par défaut au client
- 6 — **Domain Server** — le serveur fournit un serveur DNS au client
- 50 — **Requested IP address** — IP demandée par le client
- 51 — **IP Address Lease time** — durée du bail en sec.
- 54 — **Server identifier** — IP du serveur
- 55 — **Parameter Request List** — codes des options demandées
  Exemple : 37 02 01 03 signifie que le client a demandé deux options :
  - un masque de réseau (option 1 = *Subnet mask*) ;
  - et un routeur (option 3 = *Router*).

Obtention du bail 6/12

- Le client obtient une IP et d'autres informations de configuration (masque, routeur, ...) auprès d'un serveur.
- Le serveur mémorise l'IP attribuée pour ne pas l'attribuer à un autre client.
- processus DORA (Discover, Offer, Request, Ack) en 4 temps
- pour obtenir un bail sous unix : `dhclient interface`



Obtention du bail — Message DISCOVER 7/12

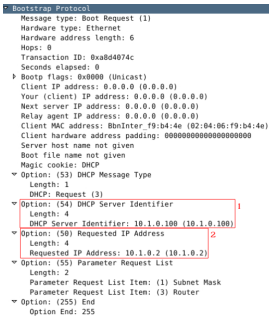


1 Type du message : 1 = Discover
2 Options demandées au serveur :
1 = un masque de réseau
3 = un routeur

Obtention du bail — Message OFFER 8/12

1 IP proposée au client
2 IP du serveur qui répond
(option obligatoire dans un OFFER)
3 durée du bail
(option obligatoire dans un OFFER)
4 masque de réseau
(Le client a demandé un routeur mais le serveur n'en a pas à lui fournir.)

Obtention du bail — Message REQUEST 9/12



1 IP du serveur dont on accepte l'offre (option obligatoire dans un REQUEST pour obtention de bail)
(⇒ si un autre serveur a fait une offre, il sait qu'elle n'est pas retenue)

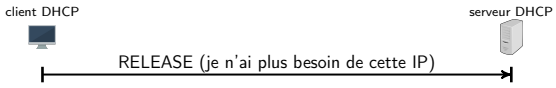
2 IP demandée
(option obligatoire dans un REQUEST pour obtention de bail)

Obtention du bail — Message ACK 10/12

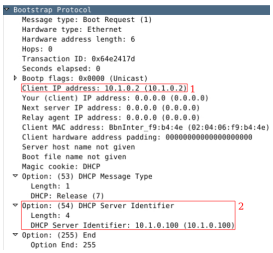
1 mêmes options que dans le OFFER

Résiliation du bail 11/12

- Le client prévient le serveur qu'il ne souhaite plus utiliser l'IP qui lui avait été attribuée.
  ⇒ Le serveur peut attribuer cette IP à un autre client.
- pour résilier un bail sous unix : `dhclient -r interface`



Résiliation du bail — Message RELEASE 12/12

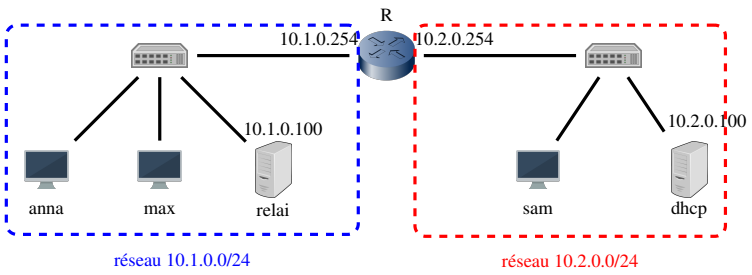


1 IP "libérée"

2 IP du serveur qui avait offert le bail (option obligatoire dans un RELEASE)

Exercice 3 — DHCP

On considère dans cet exercice le réseau ci-dessous :



Les adresses IP qui apparaissent sur la figure sont celles des interfaces configurées statiquement. Toutes les autres interfaces sont configurées automatiquement grâce au serveur DHCP s'exécutant sur la machine dhcp dont le fichier de configuration (incomplet) est donné ci-dessous :

```
option domain-name-servers 9.9.9.9;
default-lease-time __________;  # en secondes
subnet 10.1.0.0 netmask __________ {
  range 10.1.0.1 __________;
  max-lease-time __________;  # en secondes
  option routers __________;
}
subnet 10.2.0.0 netmask __________ {
  range 10.2.0.1 __________;
  option routers __________;
  option domain-name-servers 9.9.9.9;
}
```

Q 3.1 Complétez les trous dans le fichier de configuration, sachant que :

- Il y aura au maximum 20 clients connectés sur le réseau 10.1.0.0/24 et 10 clients connectés sur le réseau 10.2.0.0/24.
- La durée par défaut des baux (si le client ne demande pas de durée particulière) est de 1 heure, de même que la durée maximale des baux sur le réseau 10.1.0.0/24.

Q 3.2 Pourquoi est-il nécessaire d'avoir sur le réseau 10.1.0.0/24 un relai DHCP (s'éxecutant sur la machine relai) pour pouvoir configurer automatiquement les machines anna et max à l'aide du serveur DHCP ?

Q 3.3 Supposons qu'anna émette une requête DHCP Discover pour obtenir un bail IP. Dessinez un chronogramme faisant apparaître les paquets DHCP échangés entre anna, max, le relai et le serveur DHCP suite à l'envoi de cette requête.

Q 3.4 On s'intéresse au contenu du message DHCP Discover reçu par le serveur DHCP. On fait l'hypothèse, qu'en plus d'une IP, anna a demandé les informations suivantes au serveur DHCP : un masque de réseau, un routeur et un serveur de noms ; et qu'elle a demandé un bail de 10 heures. Donnez, en hexadécimal, les valeurs des champs suivants :

- hops
- yiaddr
- giaddr
- la liste des options

Q 3.5 On s'intéresse maintenant au contenu du message DHCP Offer envoyé en réponse au Discover. On suppose qu'aucun bail n'a été attribué par le serveur auparavant. Donnez, en hexadécimal, les valeurs des champs suivants :

- yiaddr
- la liste des options