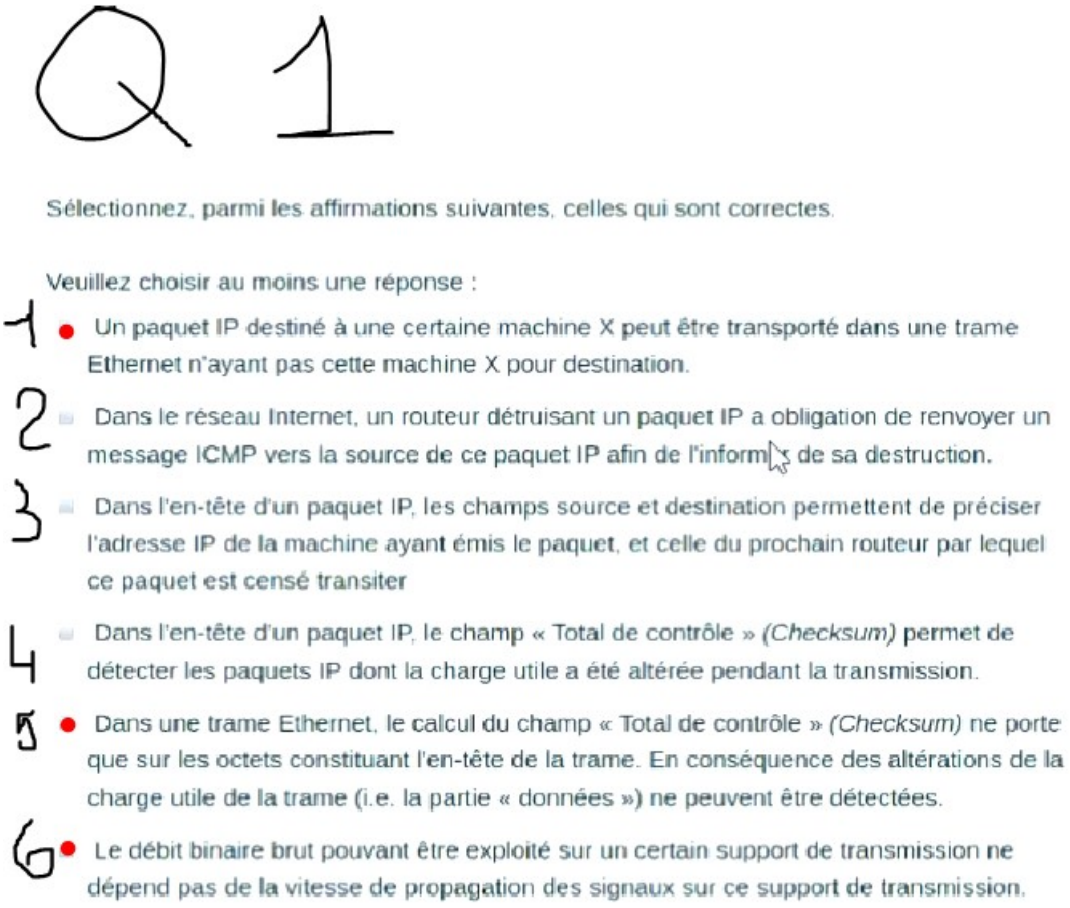


Q 1

1.

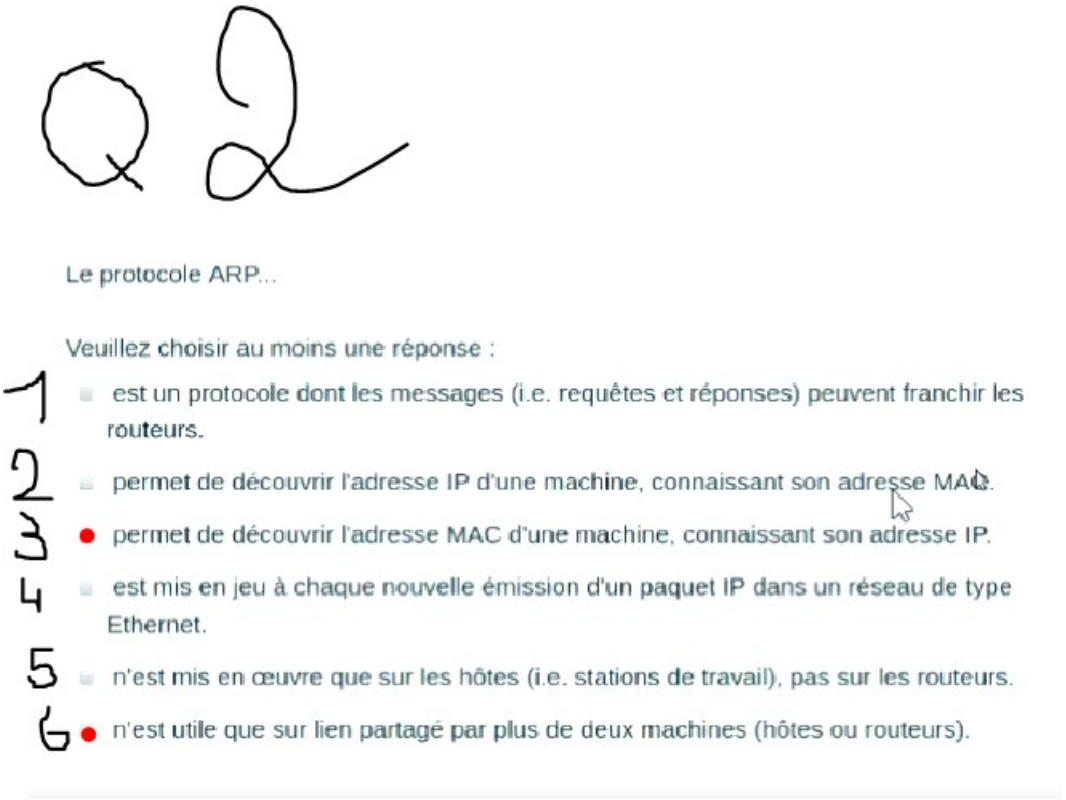
On peut avoir à acheminer un paquet IP dans une trame qui aura pour destination un routeur et pas la destination du paquet IP parce que le routeur est un relais sur le trajet (si on hésite sur ce genre de truc c'est vraiment très très inquiétant)

4.

Le checksum ne se fait que sur l'en tête et non pas la charge utile

6.

Exemple des transmissions spatiales, aucune corrélation entre la notion de débit (volume de données par unité de temps) il n'y a pas de question de vitesse de propagation de signaux là-dedans. La vitesse de propagation des signaux n'a aucun rapport avec le débit. Ex du pigeon.



Q2

2.

Vrai pour RARP

3.

Connaissant l'adresse IP d'une machine d'envoyer une requête en disant « J'ai besoin de découvrir son adresse mac parce que j'ai besoin de lui envoyer une trame »

4.

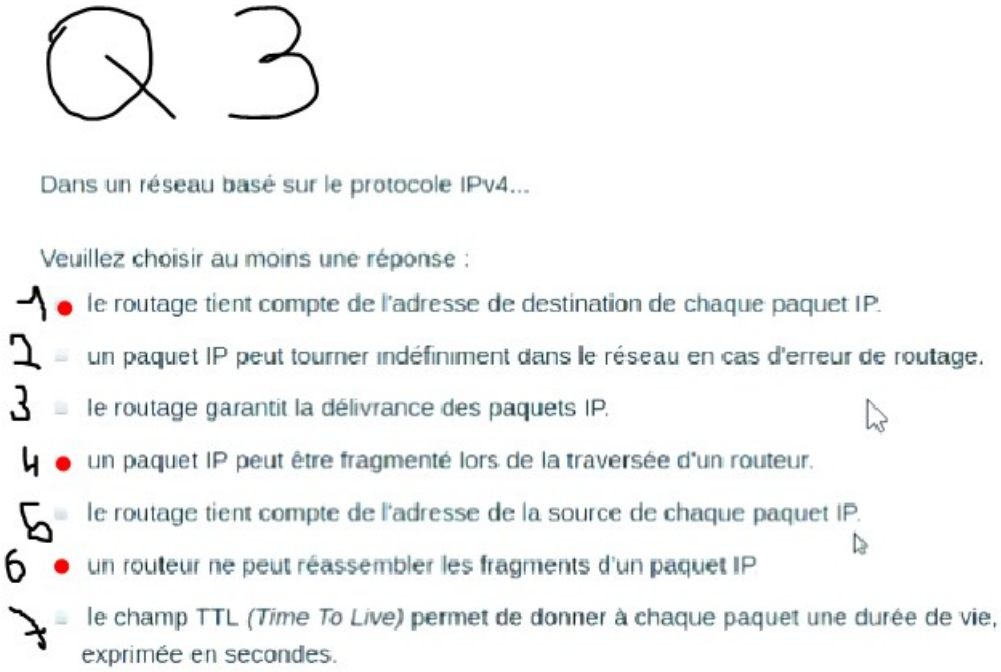
Dans tous les OS récents on met un système de cache, la table ARP, qui permet d'avoir à éviter de faire un cycle requête/réponse ARP à chaque fois qu'on envoie un paquet à une destination, sans quoi on passerait notre vie à faire de l'ARP sur le réseau.

5.

Un routeur recevant un paquet IP, constatant qu'il doit l'envoyer, par exemple à une station hôte à laquelle il est relié, va avoir besoin de découvrir l'adresse MAC de cette station hôte et va faire une requête ARP et attend une réponse pour se faire. Donc un routeur peut aussi avoir à faire des cycles requêtes/réponses ARP

6.

Sur un lien sur lequel il n'y a que 2 machines (un lien point à point), il n'y a pas d'ambiguïté sur le destinataire de la trame que l'on émet -> c'est l'autre machine, pas besoin donc de ARP



Q3

2.

Non, c’est à ça que sert le champ TTL qui fait en sorte qu’on accorde une durée de vie, un nombre de saut plus exactement, à chaque paquet et une fois qu’il a franchi les sauts prévus -> poubelle

4.

On a vu plein d’exemples en TD. Il y en a qui ont réussi l’exploit de me dire que c’était faux quand même. Ce qui est quand même assez surprenant.

5.

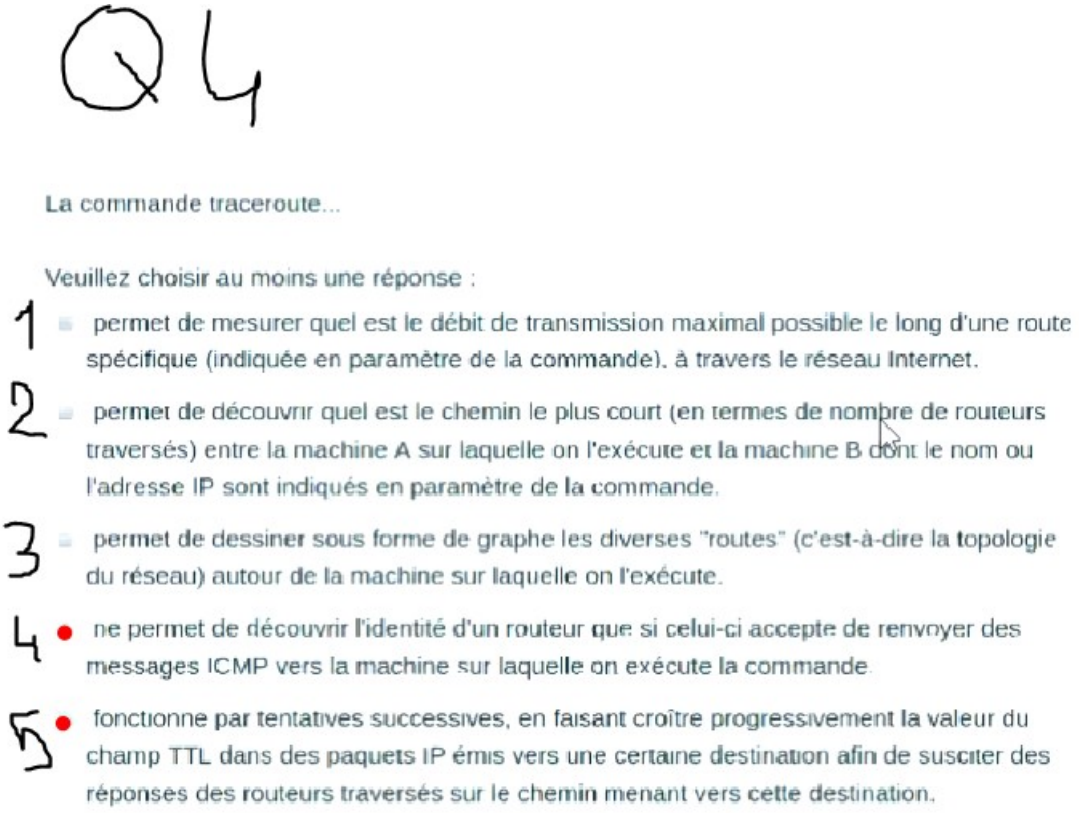
Certaines technologies de réseaux routés mais pas IP, permettent ce qu’on appelle de faire du source roating où effectivement la source est prise en compte dans la stratégie de routage mais ce n’est pas le cas dans internet.

6.

Ce n’est pas son boulot

7.

Durée de vie en nombre de sauts, nombres de liens que l’on va pouvoir franchir de routeur en routeur.



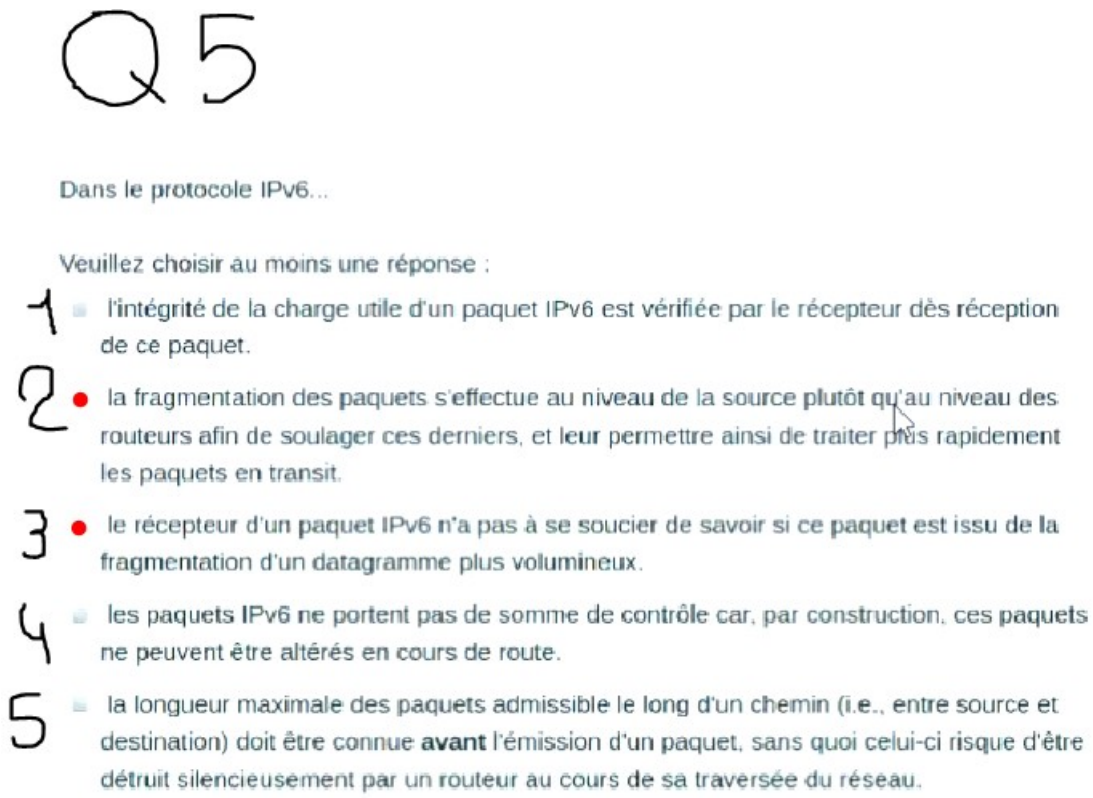
Q4

1.

Aucune indication de débit.

4.

On a vu des exemples en cours dans lesquels on fait un traceroute et de temps en temps on a des petites étoiles qui apparaissent parce qu'on n'a pas de retour d'un certain routeur, il est trop occupé pour renvoyer des messages ICMP donc on ne connait pas l'adresse IP du routeur n° tant qui se situe sur le chemin qu'on est en train d'examiner



Q5

1.

Est-ce qu'il y a une somme de contrôle dans les en-têtes des paquets IPV6 ? Bah non, du coup comment est-ce qu'on va vérifier l'intégrité d'un paquet ? Y'a rien à vérifier, on ne peut pas, pas au niveau de la couche IP. Donc c'est le contenu éventuel du paquet qui pourrait être vérifié au niveau de la couche transport, éventuellement, mais en tout cas au niveau de la couche IPV6, rien n'est prévu pour vérifier l'intégrité des paquets, même pas l'intégrité de l'en-tête.

2.

En effet c'est la responsabilité de la source de fabriquer des paquets qui ont une taille raisonnable pour emprunter un chemin dans le réseau.

3.

Donc là encore il faut se poser la question : Quelles informations sont disponibles dans l'en tête d'un paquet IPV6 concernant la fragmentation ? -> Il n'y en a pas, il n'y a aucune information concernant la fragmentation grosso-modo parce qu'il n'y a pas de fragmentation en cours de route donc il n'y a pas de réassemblage à faire au niveau de la couche IPV6

4.

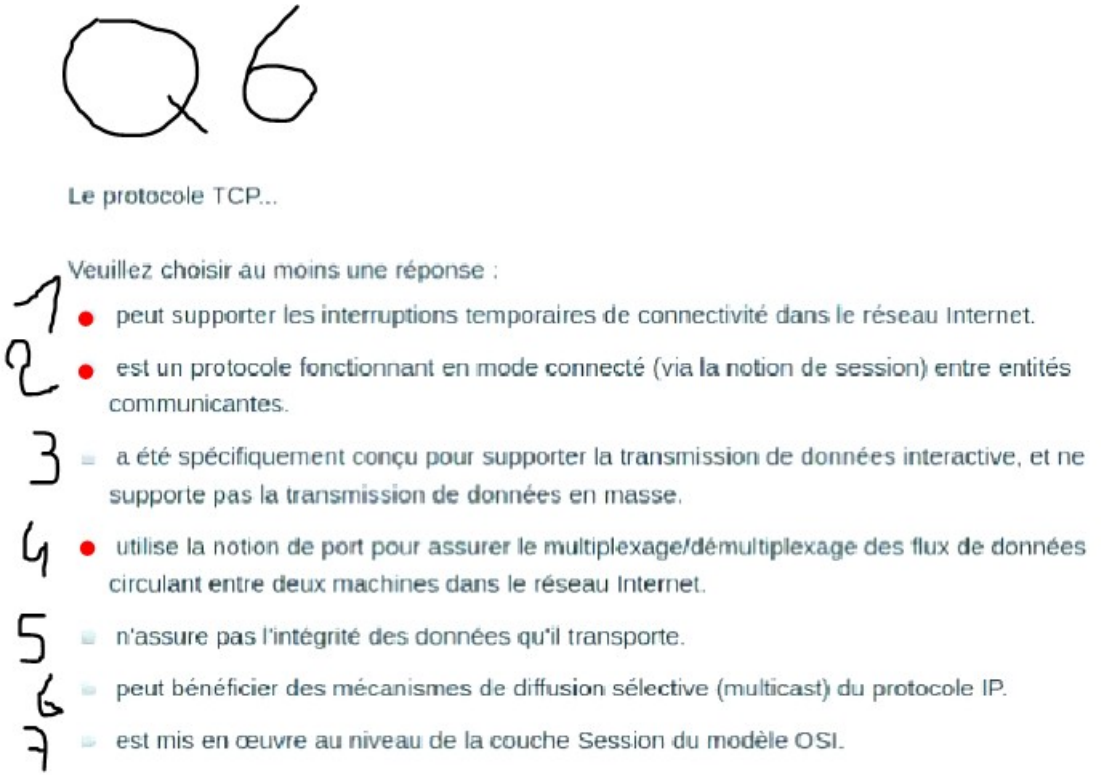
Rappelez-vous ce que je vous ai dit en tout début d'année, un support de transmission fiable ça n'existe pas, donc évidemment les paquets IPV6 peuvent être altérés en cours de route mais si on estime que si on les fait circuler sur des liens qui sont particulièrement peu fiables, on essaie de gérer les problèmes d'intégrité au niveau de la couche liaison de données, en dessous, donc en gros on perdra les paquets IPV6 s'ils sont altérés en cours de route, ils n'arriveront pas à destination, ce n'est

pas au protocole IPV6 de se préoccuper des problèmes d'intégrité. Un paquet n'arrivera pas ou il n'arrivera pas intact et dans ce cas là de toute façon au niveau de la couche IPV6 on ne fait rien de particulier. On fait remonter le paquet au-dessus à charge pour la couche transport, voire même à une couche supérieure de s'occuper de l'intégrité.

« Par construction ces paquets ne peuvent être altérés en cours de route » -> si ils peuvent donc pas à cocher

5.

Je vous avais donné un exemple où on envoie un paquet trop gros dans le réseau, le routeur le met à la poubelle parce qu'il ne peut pas l'envoyer plus loin et il renvoie un message ICMPV6 vers la source pour dire « faites moi des paquets plus petits s'il vous plait ». On peut tout à fait envoyer un paquet trop gros dans le réseau si ce n'est que l'on va prendre en retour dans les gencives un message ICMPV6 en disant « Adaptez-vous au réseau, ce n'est pas le réseau qui s'adapte à vous ». Donc la longueur maximale des paquets admissible le long d'un chemin n'a pas à être connue avant d'envoyer les paquets, par contre en envoyant des paquets on court le risque de se prendre dans les gencives des messages ICMPV6 qui nous disent « Faites mieux la prochaine fois, adaptez vos paquets à la taille que le réseau peut supporter »



Q6

1.

Ça j'en ai parlé en cours, je vous avais dit que TCP était tout à fait capable de supporter des interruptions temporaires ce qui poserait problème c'est le cas où il y a une tentative de transport de données pendant une interruption, là grosso-modo la session TCP va s'interrompre mais si une interruption de connectivité temporaire survient alors que l'on a une session TCP ouverte mais qu'on ne s'en sert pas pour transporter des données cela ne pose aucun problème, c'est même fait pour, cela fait partie du cahier des charges de TCP, c'est écrit au tout début du RFC correspondant.

3.

C'est faux on a même vu dans le cours qu'il y a des mécanismes spécifiques qui sont intégrés à TCP, d'une part pour supporter du mieux possible le trafic interactif et pour supporter du mieux possible le trafic de données en masse. TCP n'a pas été conçu spécifiquement pour l'un ou l'autre, il a vocation à supporter les 2 sortes de trafic du mieux possible.

5.

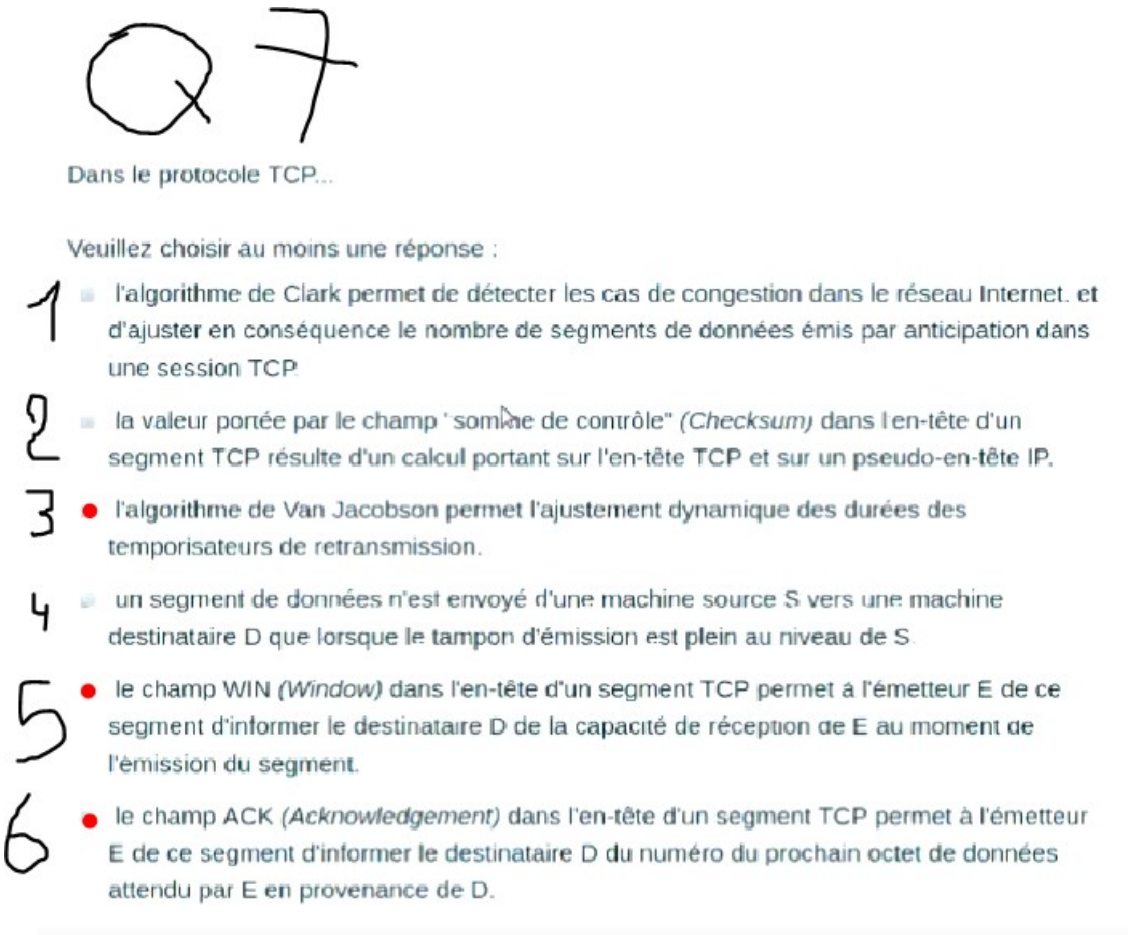
TCP fait tout ce qu'il peut pour assurer l'intégrité des données qu'il transporte et le fait même tellement bien comme je vous l'ai dit plusieurs fois qu'on a tendance à l'utiliser un peu trop facilement aujourd'hui, même pour des applications pour lesquelles TCP n'est pas forcément le meilleur choix.

6.

La diffusion sélective consiste à envoyer le même contenu à plusieurs destinations simultanément avec TCP on est dans une relation de type session donc on a 2 extrémités, et 2 seulement. Du point de vue de chacune des extrémités il n'y a qu'un seul système pair à l'autre bout de la session.

7.

Couche transport, il y en a qui ont coché oui à ça quand même hein.



Q7

2.

Vrai ? Humhum, et la charge utile alors ? Calcul portant sur l'en-tête TCP sur un pseudo en-tête IP et la charge utile; la totalité du segment TCP, pas juste les en-têtes. C'est bien pour ça que lorsqu'on reçoit un segment TCP on est en mesure de se dire que a priori tout ce qu'on a reçu a une bonne bouille.

4.

S'il fallait attendre que le tampon d'émission soit plein il n'y aurait pas de trafic interactif possible et même le trafic en masse serait sérieusement ralenti.

Le procole UDP...

Veuillez choisir au moins une réponse :

1. garantit l'ordonnancement des données grâce à la numérotation des datagrammes.
2. ● est un protocole de type *Best Effort* : il ne garantit pas la remise à destination des données qu'il permet de transporter.
3. ne permet pas au récepteur d'un datagramme de vérifier l'intégrité de ce datagramme.
4. nécessite l'ouverture d'une session entre deux machines A et B avant que A puisse envoyer des datagrammes à B.
5. ● peut exploiter les mécanismes de diffusion globale *(broadcast)* ou sélective *(multicast)* du protocole IP afin de faire parvenir un même datagramme à plusieurs destinataires.

Q8

1.

Il n'y a pas de numérotation de quoi que ce soit dans UDP

2.

C'est ce qui fait toute la légèreté de UDP par rapport à TCP qui est une usine à gaz parce que l'on fait beaucoup d'effort pour s'assurer que les données arrivent à destination, dans le bon ordre et patati et patata. Dans UDP il n'y a rien de tout ça.

3.

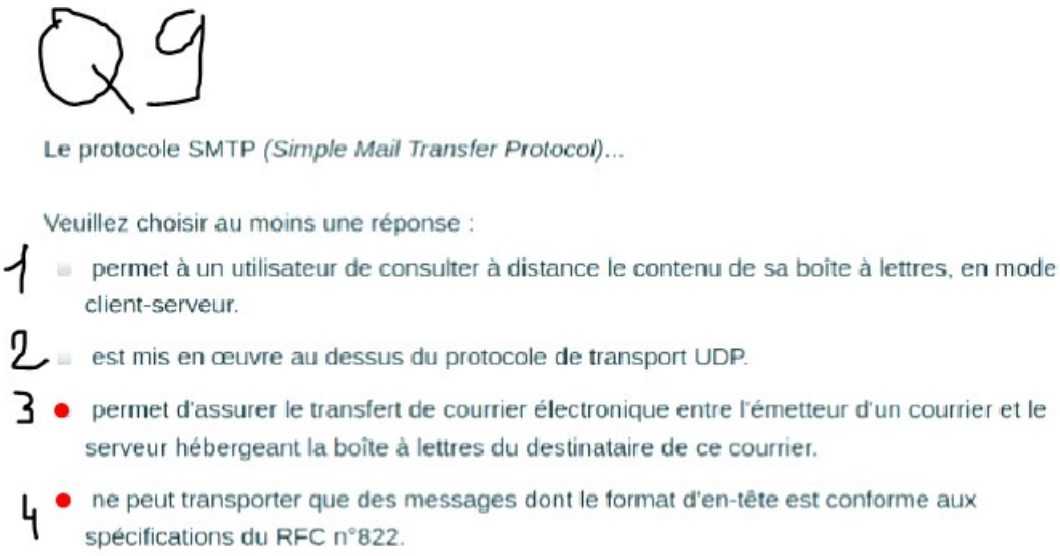
Je vous rappelle qu'il y a une somme de contrôle dans UDP qui porte sur le pseudo en-tête IP + en-tête UDP + ?? . On peut désactiver le champ checksum, mais ce n'est pas recommandé.

4.

Faux sinon on serait en mode session comme dans TCP, justement ce qui caractérise le fonctionnement par datagramme UDP c'est qu'il n'y a pas de session donc pas de démarche avant de commencer à envoyer les données, on les embarque dans un datagramme et les envoie dans le réseau

5.

C'est d'ailleurs un avantage d'UDP par rapport à TCP, de ce point de vue, en particulier pour certains champs applicatifs comme le multimédia dans certains cas ou bien l'IOT par exemple, la collecte de données dans l'IOT



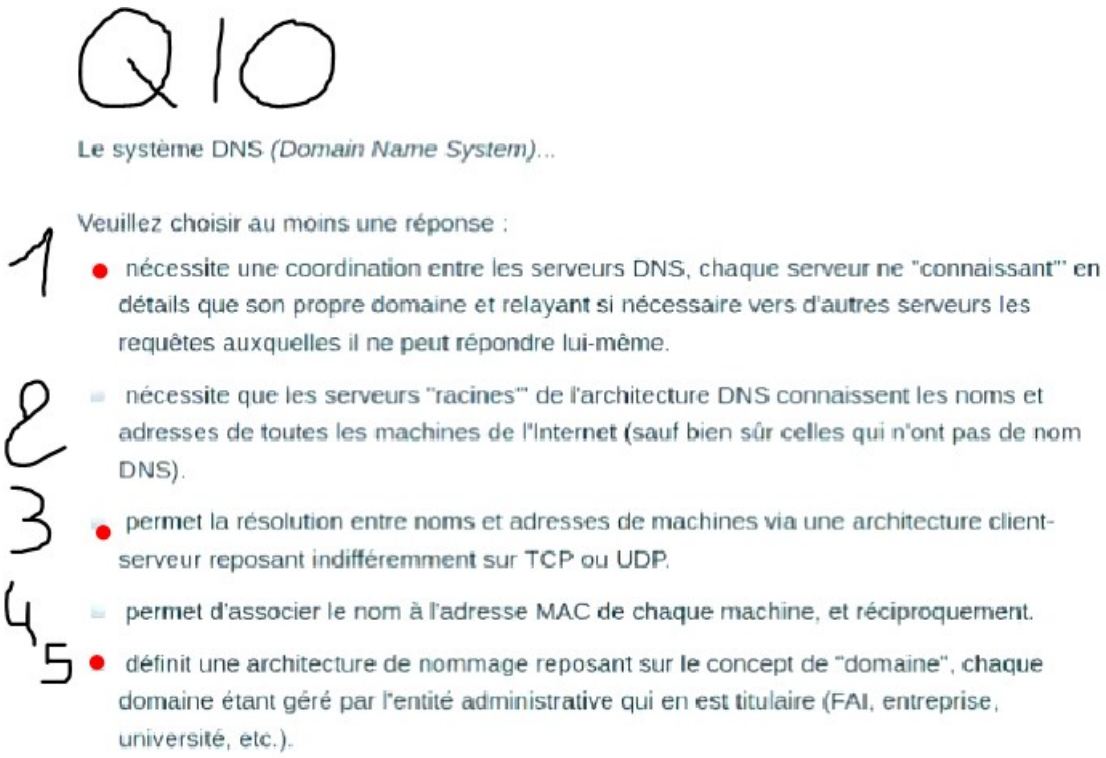
Q9

1.

J’ai bien insisté sur le fait que SMTP permette d’acheminer du courrier vers une boîte aux lettres mais ne permet pas de consulter la boîte aux lettres, pour ça on a besoin d’autres protocoles dédiés tels que POP3 et IMAP4

2.

Il est mis en œuvre au-dessus de TCP port 25 du côté du serveur



Q10

2.

On a vu que le système est arborescent et que chaque serveur ne connait que les identités des machines du domain qu'il dessert s'il y a de telles machines, mais pour le domaine racine il n'y en a pas. Il connait les serveurs qui sont situés juste en dessous de lui dans l'arborescence donc ses sous-domaines.

3.

On a vu des exemples où une requête DNS peut être transportée par un datagramme UDP et il y aura une réponse sous la forme d'un datagramme également ou bien on ouvre une session TCP vers un serveur on lui pose une question et dans cette session on récupère une réponse.

4.

L'adresse MAC on s'en fiche royalement au niveau du DNS, on ne se préoccupe de l'adresse MAC qu'au niveau d'un réseau local, d'un lien que l'on partage avec d'autres machines, pour ça on a un protocole comme APR qui va nous permettre de découvrir l'adresse MAC de la machine qui nous est voisine.