

Exercices Dirigés

Unité d'enseignement RSX101

Réseaux et protocoles pour l'Internet -LAN-

2021-2022

Ce support a été élaboré par l'équipe enseignante "Réseaux et protocoles", auteur M. Gressier Soudan.

ED•Architectures de Réseaux Locaux

Le protocole STP que nous avons vu dans UTC505 correspond à la norme IEEE 802.1D-1998 à son origine ou presque. Cette proposition a été élaborée et portée par Radia Perlman. Les grandes étapes de la construction d'un arbre couvrant sont :

1. On élit un commutateur racine ('root switch'). On choisit comme commutateur racine, le commutateur opérationnel (initialisé et actif) dont l'identifiant (Bridge ID) unique est le plus petit. Dans UTC505, le Bridge ID est la concaténation de la priorité du commutateur et de l'adresse d'un de ses ports, exemple : 8000.00:A0:D6:09:18:12.
2. Pour chaque commutateur on élit un port racine ('root port' ou encore RP). Le port racine est celui dont le coût du chemin vers la racine est le plus petit.

Path cost for different port speed and STP variation		
Data rate	STP cost	RSTP cost
(Link bandwidth)	(802.1D-1998)	(802.1W-2004 default value ¹)
4 Mbit/s	250	5,000,000
10 Mbit/s	100	2,000,000
16 Mbit/s	62	1,250,000
100 Mbit/s	19	200,000
1 Gbit/s	4	20,000
2 Gbit/s	3	10,000
10 Gbit/s	2	2,000
100 Gbit/s	N/A ²	200
1 Tbit/s	N/A	20

Source : https://en.wikipedia.org/wiki/Spanning_Tree_Protocol#Path_cost consulté le 17/12/2019 2h18

3. Pour chaque tronçon de réseau local ou voie de communication on élit un port désigné. Le port désigné d'un tronçon est le port unique d'un commutateur qui connecte ce tronçon à la racine du spanning tree avec un coût minimum et qui n'est pas un port racine. En effet, pour un tronçon donné, il peut exister plusieurs ports connectant ce tronçon à plusieurs commutateurs et donc il y a un choix possible.

Les ports non utilisés ne servent pas pour transférer des données, on dit qu'ils sont bloqués ("blocked port", BP). Par contre, ils participent au trafic de maintenance de l'arbre couvrant en échangeant des BPDU (Bridge Protocol Data Unit).

¹ 802.1W-2004 a été intégrée à 802.1D-2004

²<https://community.extremenetworks.com/faqs-90303/spanning-tree-algorithm-port-path-costs-5815427> (29/12/2019), donne la valeur 1 comme coût pour un lien à 100Gb/s. Il donne d'ailleurs aussi 1 pour un lien à 1Tb/s. Ce lien donne d'ailleurs une autre norme, plus récente, pour le calcul du coût du chemin à la racine : 802.1t mais qui est incorporée dans 802.1D-2004.

Ci-après, le diagramme d'états d'un port pour le STP. La source est TCP/IP Illustrated Volume 1, 2^{ème} édition de K. R. Fall, et, R. Stevens p305... un excellent livre pour qui veut faire du réseau sérieusement.

Dans la norme 802.1D-2004³, certains états de la version 1998 ont été regroupés en "*Discarding*". Quand le port apprend des trames qui lui arrive mais ne les fait pas suivre, il est dans l'état "*Learning*". Quand il apprend et qu'il fait suivre, il est dans l'état "*Forwarding*", c'est ainsi que la norme le décrit p35.

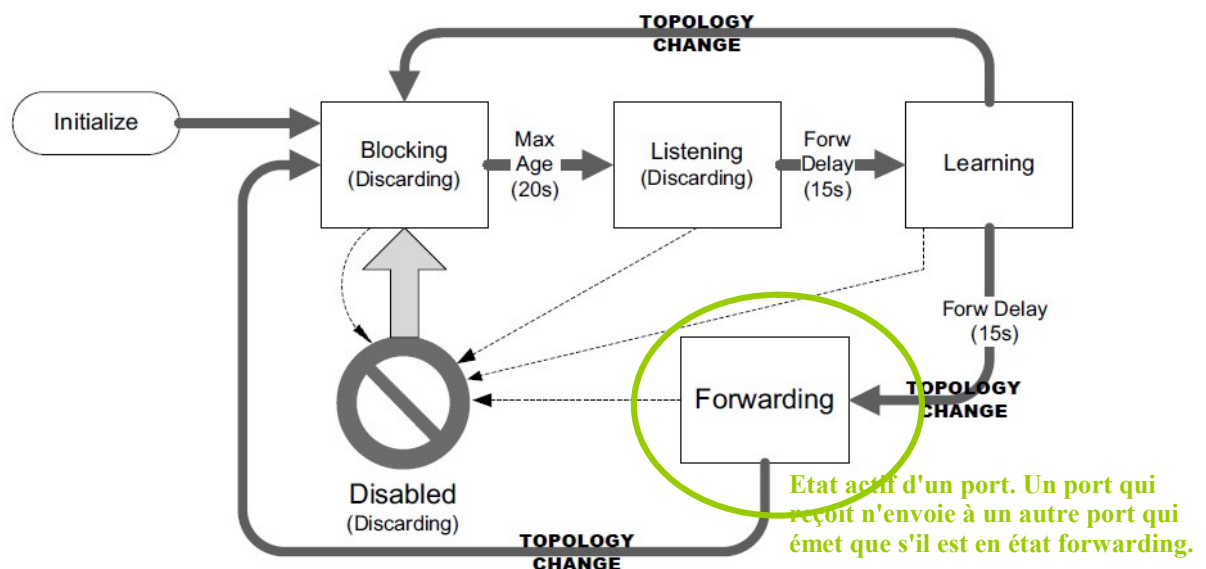


Figure 3-14 Ports transition among four major states in normal STP operation. In the blocking state, frames are not forwarded, but a topology change or timeout may cause a transition to the listening state. The forwarding state is the normal state for active switch ports carrying data traffic. The state names in parentheses indicate the port states according to the RSTP.

³ La norme complète peut être lue sur le lien suivant : <https://www.slideshare.net/basschuck2411/8021-d-2004> (06/12/2020). 802.1D-2004 correspond à RSTP, Rapid Spanning Tree Protocol.

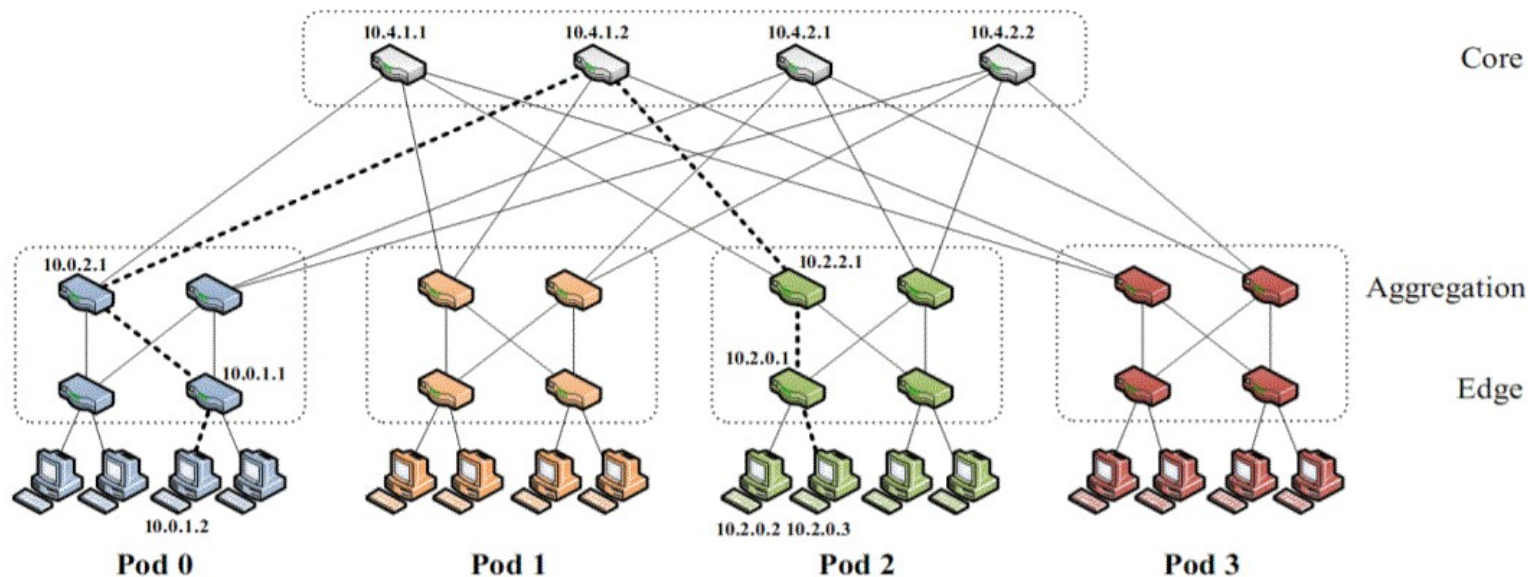
Exercice 1 : Construction d'un arbre couvrant sur un réseau local Ethernet de Data Center.

C'est une révision mais du Spanning Tree avec un niveau de complexité supplémentaire par rapport à ce qui a été fait en UTC505.

Les Data Center, Centres de données en français, ont une architecture particulière car ils servent des données à plusieurs abonnés (tenants en anglais, ou parties prenantes en français). Les communications sont essentiellement Nord-Sud (extérieur-serveurs) plutôt que Est-Ouest (entre serveurs) quoiqu'il semblerait que c'est en train de changer avec le développement des micro-services.

Le schéma ci-après correspond à une architecture de type k-Fat Tree à 3 niveaux : commutateurs de cœur (core), commutateurs d'agrégation (aggregation), et commutateurs de bordure (edge). k représente le nombre de groupes de commutateurs (pods) en bas de l'arbre, et le nombre de commutateurs par pod. Ne pas oublier que switch est la traduction anglaise de commutateur. Cette topologie représente un réseau de Centre de données (Data Center en anglais). Les serveurs sont reliés aux commutateurs de bordure. Pour cette architecture, k vaut 4.

Pour bien comprendre, il faut toujours se souvenir que k est le nombre de ports d'un commutateurs/switch. Par conséquent, $k/2$ ports sont dirigés vers le haut et $k/2$ ports sont dirigés vers le bas. Pour un commutateur edge, il connecte $k/2$ serveurs et $k/2$ commutateurs aggregation. Pour un commutateur aggregation, il connecte $k/2$ commutateurs edge et $k/2$ commutateurs core. Les commutateurs core, par contre, connectent k commutateurs aggregation.



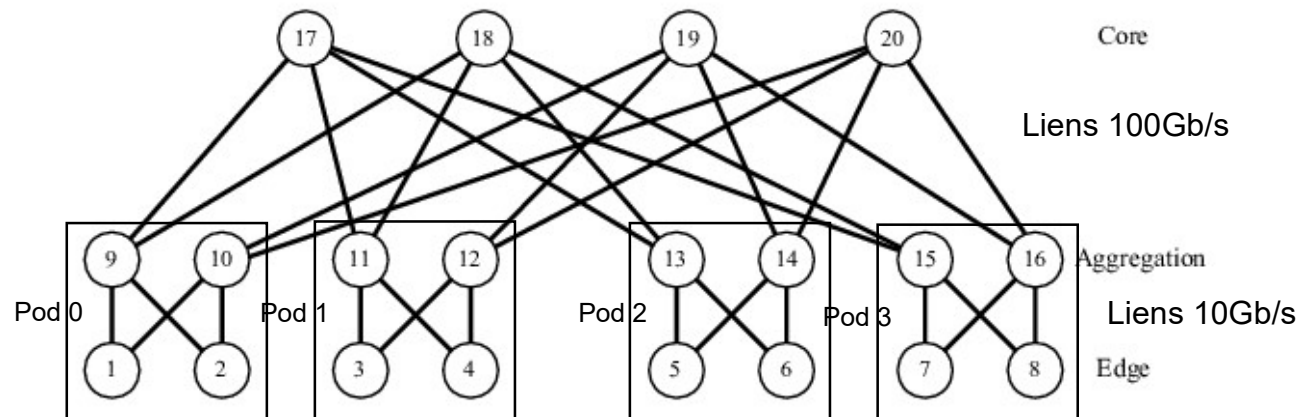
source : <https://www.cs.cornell.edu/courses/cs5413/2014fa/lectures/08-fattree.pdf>, consultée le 31/12/2019

Question 1

On suppose que le commutateur 20 est la racine de ce réseau de commutateurs quand on considère l'algorithme de l'arbre couvrant (Spanning Tree). On suppose que 20 est l'identificateur le plus prioritaire, alors que 1 est le moins prioritaire.

- On suppose que les liens entre les commutateurs d'agrégation (Aggregation) et les commutateurs de coeur (Core) ont un débit de 100Gb/s.
- On suppose que les liens entre les commutateurs de bordure (Edge) et les commutateurs d'agrégation (Aggregation) ont un débit de 10Gb/s.

Pour l'algorithme du Spanning Tree, on suppose que le coût, associé à un lien 100Gb/s, est de 1 tandis que le coût associé à un lien 10Gb/s est de 10.



On a repris une forme épurée du FAT-TREE complet donné en illustration au début de l'exercice. La source de ce dessin est https://www.researchgate.net/publication/285517629_Multipath_Routing_from_a_Traffic_Engineering_Perspective_How_Beneficial_Is_It/figures?lo=1. On l'a complété pour faire ressortir le concept de pod et les données de l'exercice.

Par convention, les pods sont numérotés de gauche à droite, de 0 à k-1. Les switches sont numérotés de gauche à droite aussi et de bas en haut.

On donne la définition d'un FAT-TREE d'après <https://www.cs.cornell.edu/courses/cs5413/2014fa/lectures/08-fattree.pdf> dont l'auteur est Hakim Weatherspoon Assistant Professor, Dept of Computer Science, Cornell University :

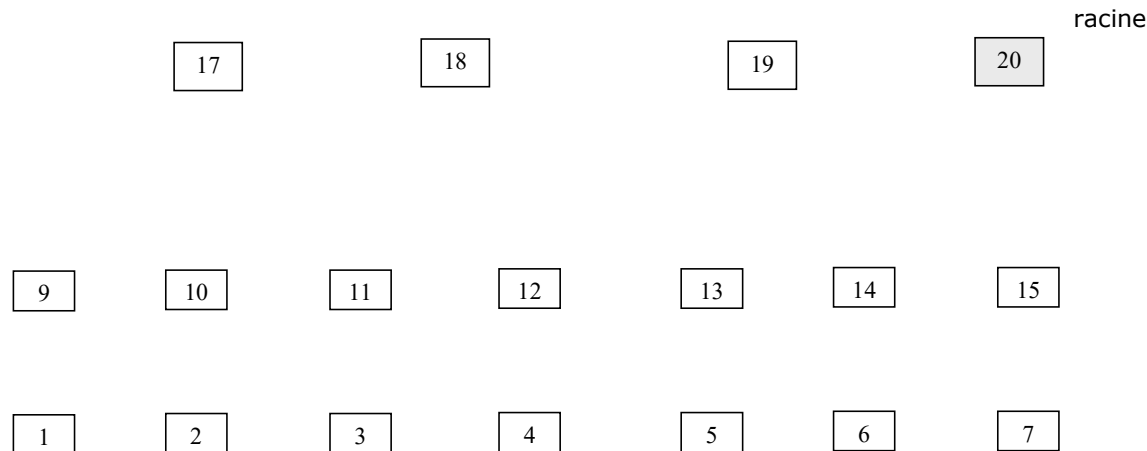
"K-ary fat tree: three-layer topology (edge, aggregation and core)"

- each pod consists of $(k/2)^2$ servers⁴ & 2 layers of $k/2$ k -port switches⁵
- each edge switch connects to $k/2$ servers & $k/2$ aggr. switches
- each aggr. switch connects to $k/2$ edge & $k/2$ core switches
- $(k/2)^2$ core switches: each connects to k pods"

On constate que nos dessins sont conformes à cette définition.

- Les ports du commutateur 20 sont-ils des ports racines ou des ports désignés (ou encore appelés ports relais) ?
- Marquer sur le dessin ci-dessous les ports racines des commutateurs 10, 12, 14, 16.
- Quel est le port racine des commutateurs 1, 2, 3, 4, 5, 6, 7, 8 ? Indiquer sur le dessin ci-dessous.
- Quel est le port racine des commutateurs 9, 11, 13, 15 ? Indiquer sur le dessin ci-dessous.
- Quel est le port racine du commutateur 19 ? Indiquer sur le dessin ci-dessous.
- Quel est le port racine des commutateurs 17 et 18 ? Indiquer sur le dessin ci-dessous.
- Pour chaque lien, marquer les ports désignés et barrer celui qui est désactivé d'une croix.

Dessiner l'arbre couvrant résultant en éliminant les liens désactivés du dessin.



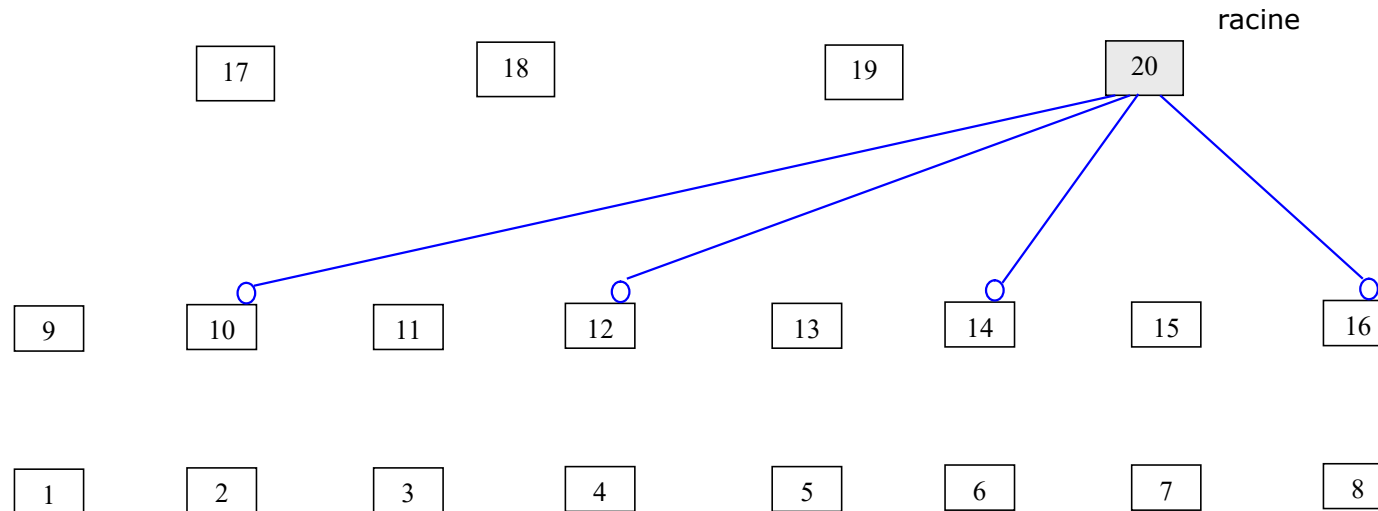
⁴ Dans l'exercice, on ne s'intéresse pas au nombre de serveurs (blades) attachés à un commutateur edge.

⁵ Edge et Agrégation

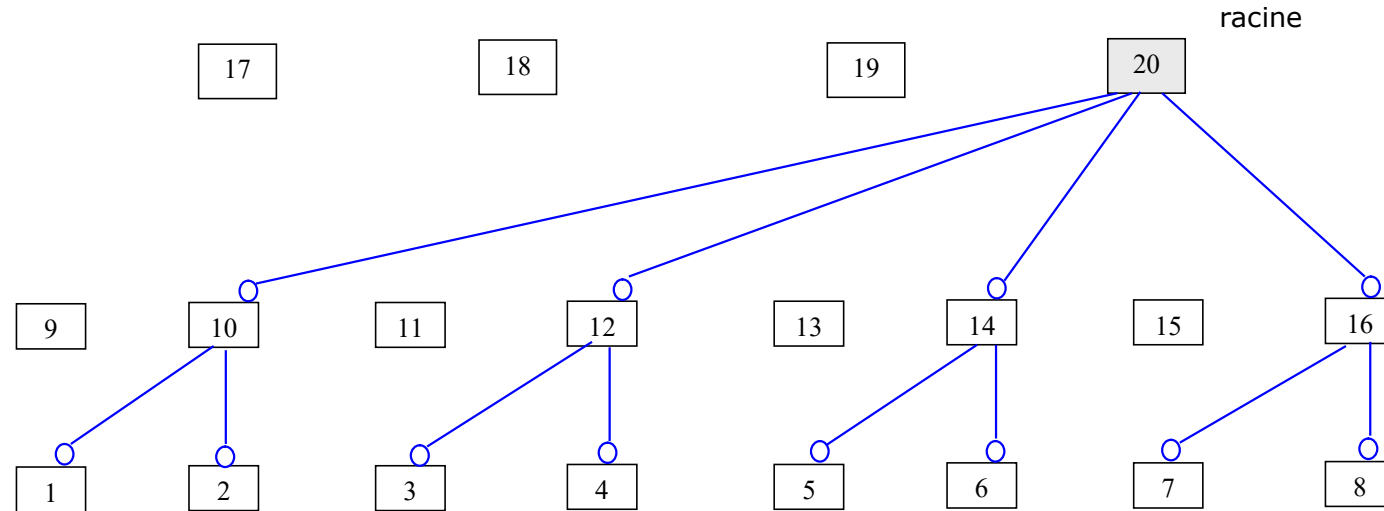


Correction :

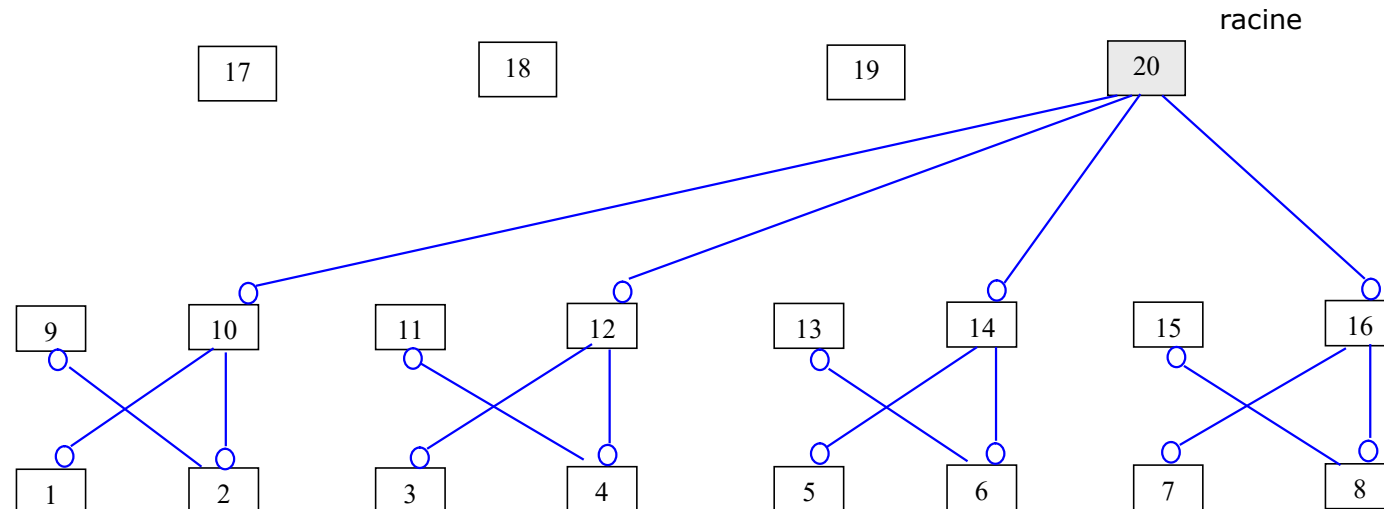
- Les ports du commutateur 20 sont-ils des ports racines ou des ports désignés (ou encore appelés ports relais) ? Les ports du commutateur 20 sont tous des ports désignés car c'est la racine.
- Marquer sur le dessin ci-dessous les ports racines des commutateurs 10, 12, 14, 16. Les ports racine sont tous les ports dont le lien connecte au commutateur 20. Entre les deux ports montants de ces commutateurs, c'est le port de droite. On les représente sur le dessin ci-après.



- Quel est le port racine des commutateurs 1, 2, 3, 4, 5, 6, 7, 8 ? On cherche à aller au plus court chemin. Ces commutateurs passeront donc par les commutateurs d'agrégation de numéro pair.

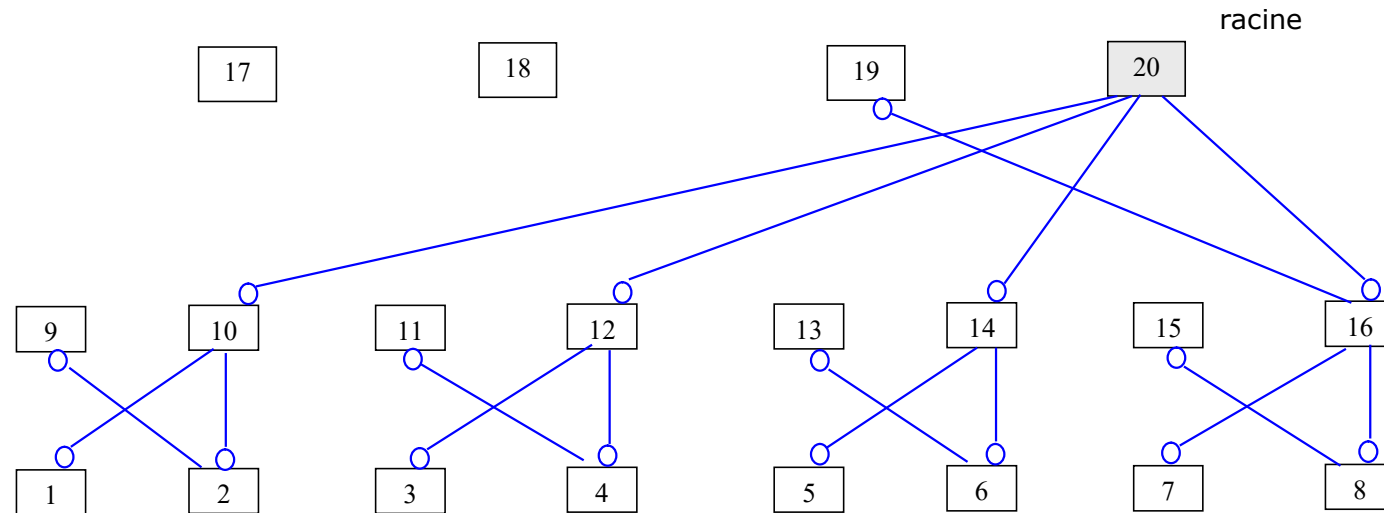


- Quel est le port racine des commutateurs 9, 11, 13, 15 ? Comme 2 est prioritaire sur 1 car de plus forte priorité, 9 passera par 2 puisque les chemins passant par 1 et 2 sont de même coût. Même raisonnement pour 11, 13 et 15.

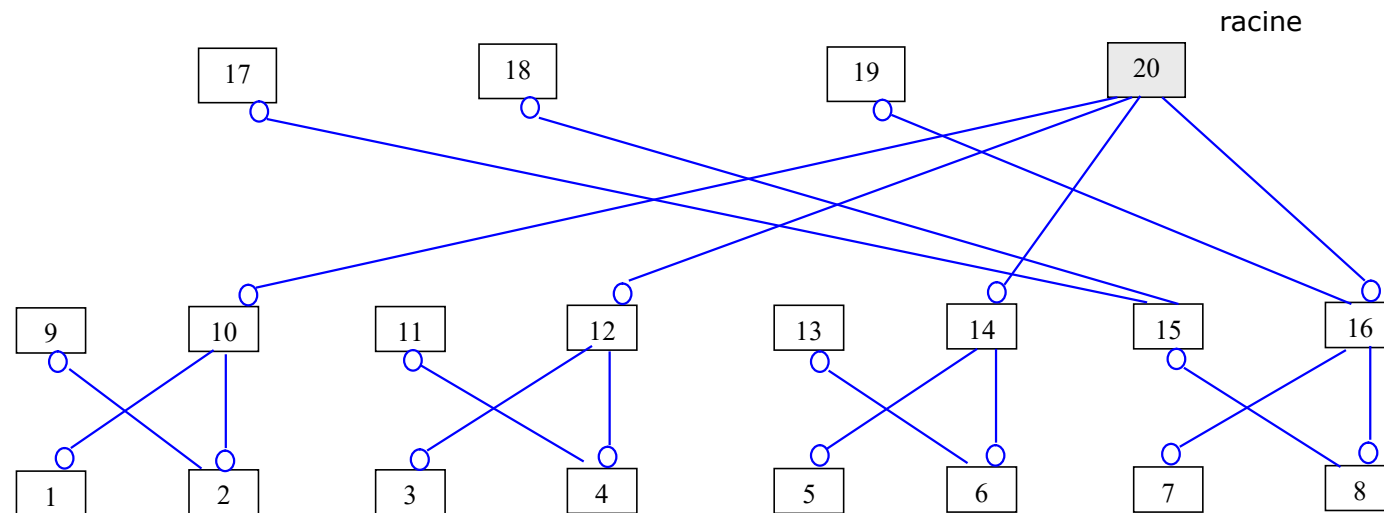


- Quel est le port racine du commutateur 19 ? Pour relier 19 à la racine, on peut passer par 16, 14, 12 ou 10. Chacun des chemins a le même coût. Le commutateur d'agrégation 16 a la plus forte priorité, on va donc relier 19 par le lien qui le mène à

16.



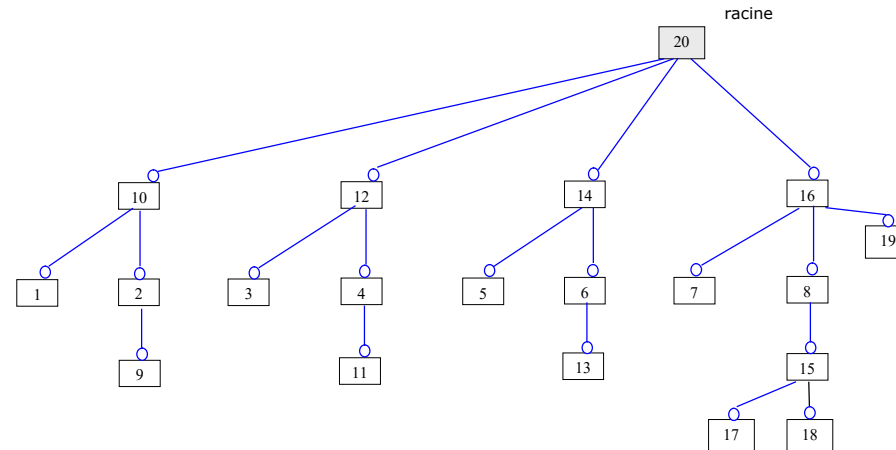
- Quel est le port racine des commutateurs 17 et 18 ? Indiquer sur le dessin ci-dessous. 17 et 18 passent par 9, 11, 13 et 15 avec un coût identique. 15 est plus prioritaire, donc ils vont passer par le commutateur d'agrégation 15.



- Pour chaque lien, marquer les ports désignés et barrer celui qui est désactivé d'une croix. Les liens désactivés sont ceux qui ne

figurent pas sur le dessin final ci-dessus. Ils sont désactivés pour le trafic des trames de données par pour le protocole spanning tree.

On peut essayer de présenter l'arbre couvrant autrement, plus sous la forme d'un arbre...



Question 2

- Qu'en concluez vous sur la pertinence de l'approche Spanning Tree Protocol pour ce type de réseau de Centre de Données ? Justifiez votre point de vue brièvement ?

Correction :

La configuration obtenue à partir du spanning tree protocol est plutôt décevante. On perd beaucoup de liens pour le transfert de données. En particulier, les commutateurs d'agrégation 17 et 18, avec des liens 100Gb/s, repassent par le commutateur 15 de type Edge qui est sur un lien à 10Gb/s pour aller vers la racine.

- Proposer en quelques lignes une ou plusieurs autres approches de routage pour ce type de réseau. Ne pas hésiter à comparer avec le principe des algorithmes de routage vu pour la couche 3 d'Internet.

Correction :

Vous verrez TRILL dans la suite. Il est basé sur un routage à état des liens (comme OSPF) qui permet d'utiliser tous les liens pour router les trames Ethernet. Il serait bien plus efficace pour supporter le trafic d'un data center que le Spanning Tree Protocol.

Bien sûr, on peut argumenter qu'en utilisant des VLANs, on pourrait avoir différents arbres couvrants co-existants sur la même architecture physique. Il faut alors penser à l'administration de l'architecture, ça peut devenir un vrai casse-tête, et être au bout du compte pas au maximum d'efficacité.

Question 3

Les commutateurs sont regroupés en "pods". Dans le schéma ci-dessus les pods contiennent 4 commutateurs : 2 commutateurs de bordure, et 2 commutateurs d'agrégation. Pour un calcul plus général :

3.1. Combien y a t il de commutateurs de coeur en fonction de k ? Expliquez brièvement votre raisonnement.

Correction :

La définition donnée en début d'exercice, le donne directement, il y en $(k/2)^2$. Mais on peut le retrouver puisque $k/2$ commutateurs Edge se connecte à $k/2$ commutateurs Aggregation. Le résultat est donc $k/2 * k/2$.

Dans l'architecture k vaut 4. On a bien $2*2 = 4$ commutateurs par pod.

3.2. Combien y a t il de commutateurs au total en fonction de k ? Expliquez brièvement votre raisonnement.

Correction :

D'après la définition : *"each pod consists of $(k/2)^2$ servers & 2 layers of $k/2$ k-port switches"*, on a donc $2*k/2$ commutateurs par pod, soit k. Comme on a k pods, il y a k^2 commutateurs au niveau pods.

On a $(k/2)^2$ commutateurs de cœur. Donc on a $k^2 + (k/2)^2$ commutateurs dans le k-FAT-TREE.

Soit N le nombre total de commutateurs dans le k-FAT-TREE, $N = 5*k^2/4$.

3.3. Combien y a t il de liens entre les commutateurs en fonction de k ? Expliquez brièvement votre raisonnement. Quelle valeur trouvez vous pour k = 4 ? Est-ce que cela correspond au schéma de l'exercice ?

Correction :

Chaque commutateur edge est connecté à $k/2$ commutateurs aggregation. Comme dans un pod, il y a $k/2$ commutateurs edge, on $(k/2)^2$ liens dans un pod. Comme il y a k pods, on a $k^3/4$ liens qui partent des commutateurs edge vers les commutateurs aggregation.

Chaque commutateur aggregation est connecté à $k/2$ commutateurs core. Il y a $k/2$ commutateurs aggregation dans un pod, et, il y a k pods. On a donc $k*k/2*k/2$ liens entre les commutateurs aggregation et core, soit $k^3/4$.

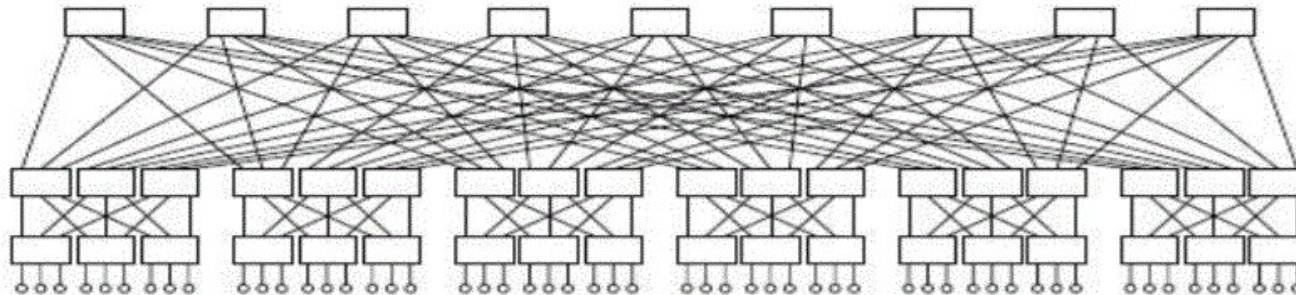
Le nombre total de liens, L, est donc $2* k^3/4$, soit $L = k^3/2$.

Remarque : Il y a $k/2$ commutateurs edge par pod. Chaque commutateur a k ports mais $k/2$ sont utilisés pour connecter des serveurs. Par pod, on peut connecter $k^2/4$ serveurs, et sur un k -FAT-TREE on peut connecter $k \cdot k^2/4$ serveurs, soit $k^3/4$ serveurs "physiques".

L'ensemble des calculs et observations, issus de la question 3, est résumé dans le tableau ci-dessous avec différentes valeurs pour k .

Nombre de pods	k	4	6	24	32
Nombre de Commutateurs de cœur (core)	$(k/2)^2$	4	9	144	256
Nombre de Commutateurs d'agrégation (aggregation)	$k^2/2$	8	18	288	512
Nombre de Commutateurs de bordure (edge)	$k^2/2$	8	18	288	512
Nombre de Commutateurs N	$5 \cdot k^2/4$	20	45	720	1280
Nombre de Liens L entre commutateurs	$k^3/2$	32	108	6912	16384
Nombre max de serveurs	$k^3/4$	16	54	3456	8192

Un exemple d'architecture FAT-TREE 6-pod à 3 niveaux :

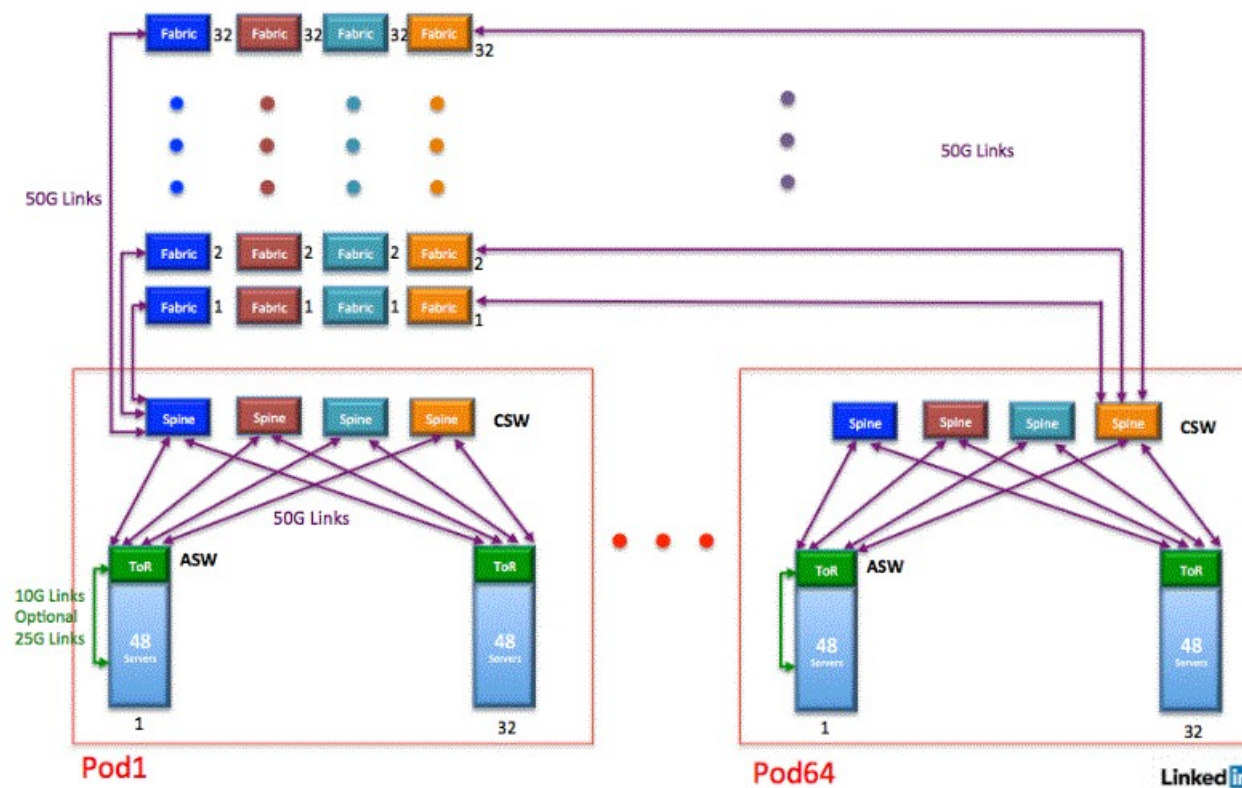


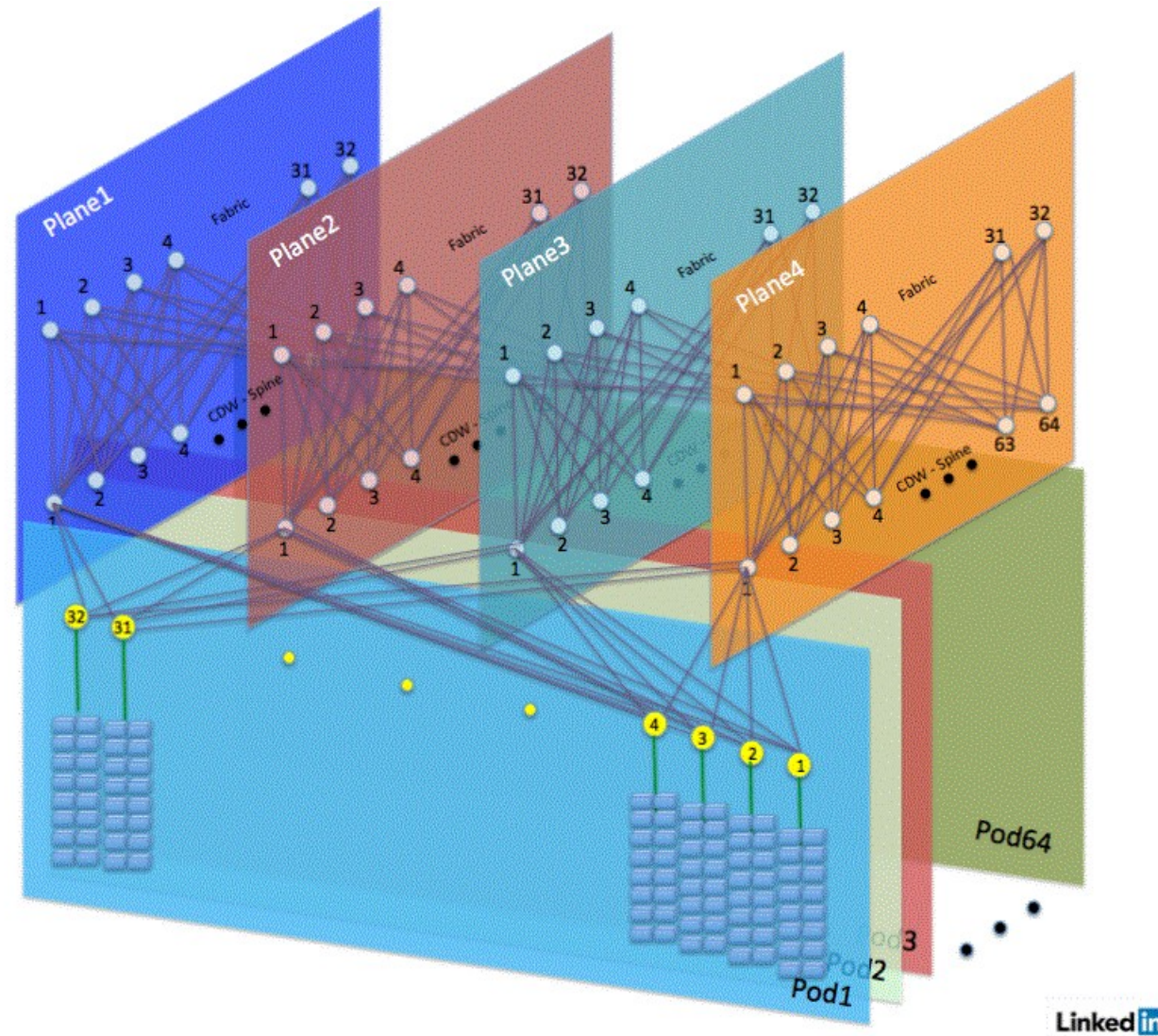
source : <https://www.cs.cornell.edu/courses/cs5413/2014fa/lectures/08-fattree.pdf>

Attention, il existe des FAT-TREE à 2 niveaux, 4 niveaux...

Pour ceux qui sont intéressés, regarder l'architecture du réseau DCN (Data Center Network) de LinkedIn. La source est : <https://engineering.linkedin.com/blog/2016/03/the-linkedin-data-center-100q-transformation>, consultée le 1/01/2020.

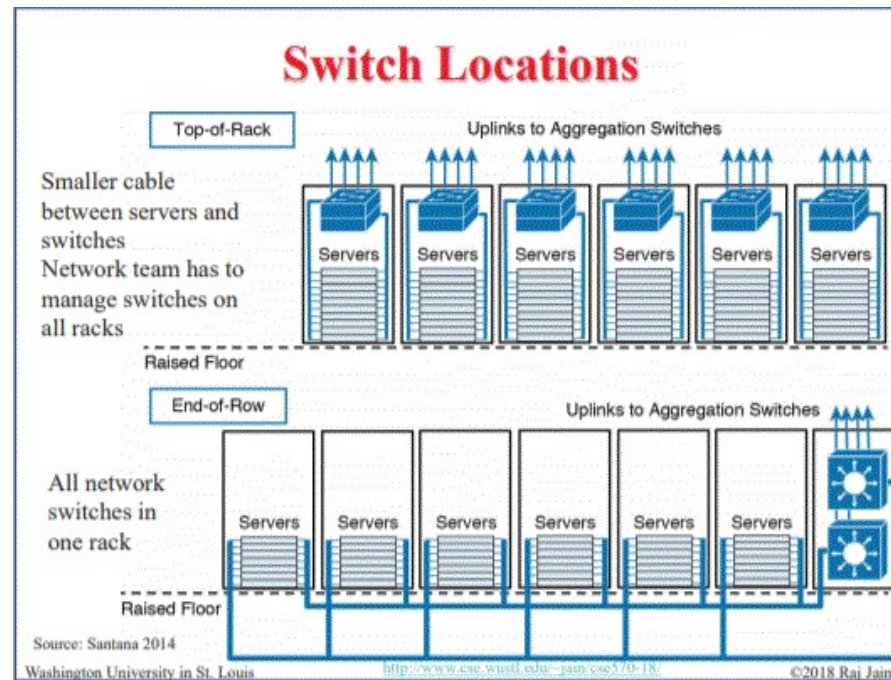
The diagrams below show multiple representation of the fabric architecture for LOR1.




[LinkedIn](#)

Attention, ce n'est pas tout à fait un FAT-TREE. Lire la tentative de décortique de l'architecture par Diptanshu Singh dans "Demystifying DCN Topologies: Clos/Fat Trees – Part2" au lien <https://packetpushers.net/demystifying-dcn-topologies-clos-fat-trees-part2/>.

Pour un panorama complet du problème, il faut aussi avoir en tête la répartition des switches et des serveurs dans les racks :



Extrait de https://www.cse.wustl.edu/~jain/cse570-18/ftp/m_03dct4.pdf, le 01/01/2020, support du cours du professeur Raj Jain, Washington University in Saint Louis.

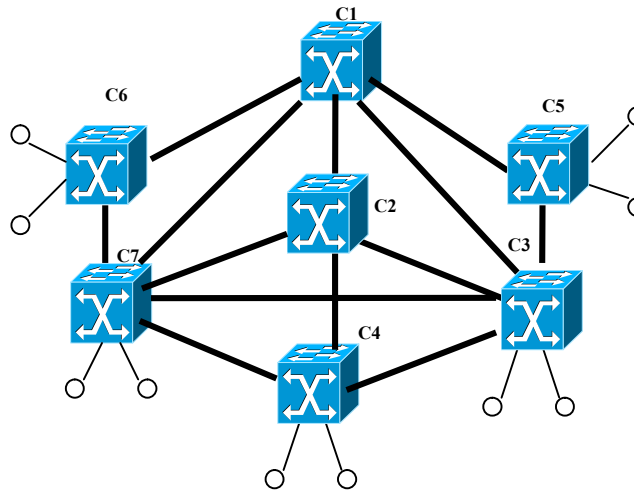
Exercice 2 : Réseaux Locaux et Data Centers – TRILL vs STP (à faire en autonomie)

Ce qui caractérise les Data Centers, c'est le très grand nombre d'équipements à relier entre eux. On va donc s'intéresser à de nouveaux protocoles qui adressent le domaine des réseaux locaux en Data Center. On a bien proposé une amélioration du protocole Spanning Tree (STP). CISCO est à l'initiative de la norme IEEE 802.1w, encore appelée Rapid STP (RSTP) qui s'intègre maintenant à IEEE802.1D-2004. Mais cette proposition n'apporte pas d'évolution radicale et innovante. Par contre, TRILL (TRansparent Interconnection of Lot of Links) aurait pu se substituer au protocole STP, car il introduit une approche innovante. Visiblement, ça n'a pas été le cas.

En Novembre 2021, TRILL a, à peu près, 10 ans. En anglais... Pour une vue très globale et très introductive, vous pouvez la vidéo <https://www.youtube.com/watch?v=l7mvUrc90jQ> (consultée le 30/11/2021) qui est courte 8mn15s. Pour un peu plus de détails, vous pouvez consulter <https://www.youtube.com/watch?v=MRf3V9N8yCg> (30/11/2021) qui dure 1h18mn.

TRILL combine Routage, suivant une stratégie de type protocole à états des liens, et Commutation. OSPF a été vu en cours, il devrait vous inspirer pour répondre aux questions de cet exercice. Les équipements qui exécutent TRILL sont appelés RBridges (Routing Bridge). Si on veut être plus précis, c'est plutôt un autre protocole à état des liens, IS-IS (Intermediate System to Intermediate System) vu dans les exercices complémentaires de la partie routage, qui inspire TRILL.

Soit le réseau local suivant, il est constitué de commutateurs TRILL.



Dans le protocole TRILL on applique un protocole à état des liens pour élaborer le routage entre tous les commutateurs. A l'issue du calcul chaque RBridge aura fabriqué son propre arbre de routage à coût minimal.

Sachant que tous **les liens ont le même débit de 1Gb/s, ils ont tous le même coût**. On prendra la valeur 1 pour le coût d'un lien afin de simplifier les calculs. Ils sont bi-directionnels car on fait du full-duplex, le coût est donc identique dans les deux sens.

On représente la base de données de l'état des liens (LS DB) sous la forme d'une matrice. Comme les liens sont bi-directionnels et les coûts identiques dans les deux sens, la matrice ci-dessous est symétrique. Le coût est infini quand il n'y a pas de lien entre deux nœuds.

Question 1



Compléter la base d'état des liens ci-dessous.

LS DB	C1	C2	C3	C4	C5	C6	C7
C1	0	1	1	∞	1	1	1
C2	1	0	1	1	∞	∞	1
C3	1	1	0	1	1	∞	1
C4	∞	1	1	0	∞	∞	1
C5	1	∞	1	∞	0	?	?
C6	1	∞	∞	∞	?	0	1
C7	1	1	1	1	?	1	0

Correction :

Sur la diagonale, "on reste sur place". Donc le coût vaut 0.

LS DB	C1	C2	C3	C4	C5	C6	C7
C1	0	1	1	∞	1	1	1
C2	1	0	1	1	∞	∞	1
C3	1	1	0	1	1	∞	1
C4	∞	1	1	0	∞	∞	1
C5	1	(	1	(	0	(	(
C6	1	(	(	(	(	0	1
C7	1	1	1	1	(	1	0

Question 2

Pourquoi toutes les bases d'état des liens sont identiques sur chaque commutateur RBridge ?

Correction :

Le principe même des protocoles à état de lien est d'avoir la même base de données de l'état de tous les liens sur chaque site. Sinon, il n'est pas possible de calculer le chemin le plus court pour un équipement de routage vers tous les autres nœuds du réseau avec l'algorithme de Dijkstra.

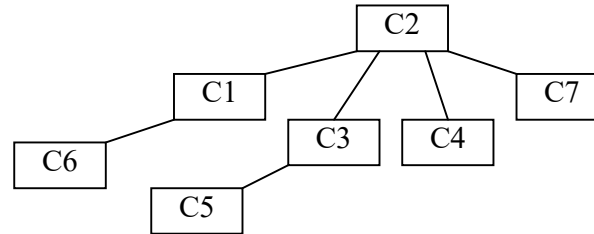
Alors qu'en UTC505 on avait présenté ces informations sous la forme d'une table, ici, nous les présentons sous la forme d'une matrice. C'est équivalent.

Question 3

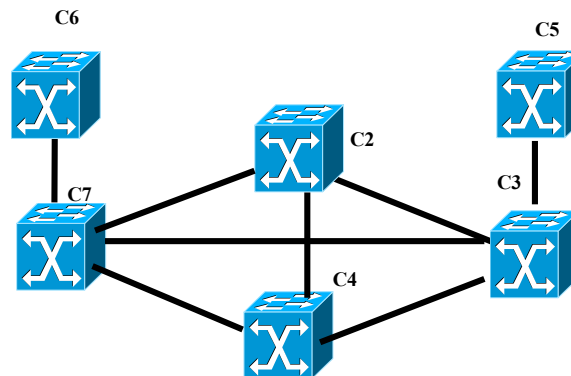
A partir de la base de données d'état des liens et avec l'algorithme de Dijkstra, calculer l'arbre de routage de C2.

Correction :

C2 y est donc le routeur de départ. On obtient l'arbre de routage suivant :



Maintenant C1 tombe en panne. On a alors le réseau local suivant :

**Question 4**

Mettre à jour la matrice LS DB que vont avoir tous les nœuds à la fin de l'exécution du protocole de mise à jour de l'état des liens sur l'ensemble du réseau.

Correction :

LS DB	C1	C2	C3	C4	C5	C6	C7
C1	0	∞	∞	∞	∞	∞	∞
C2	∞	0	1	1	∞	∞	1
C3	∞	1	0	1	1	∞	1
C4	∞	1	1	0	∞	∞	1
C5	∞	∞	1	∞	0	(	(
C6	(	(	(	(	(	0	1
C7	(	1	1	1	(	1	0

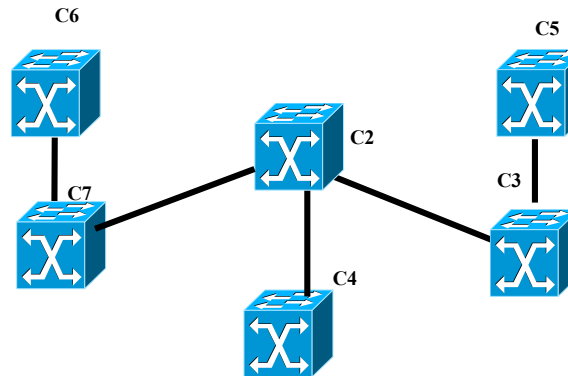
Les liens qui ont été coupés, donnent une valeur infinie dans la matrice LS DB.

Question 5

Donner l'arbre couvrant résultant de la panne de C1 issu du protocole Spanning Tree.

Correction :

L'arbre couvrant issu de STP est le suivant :

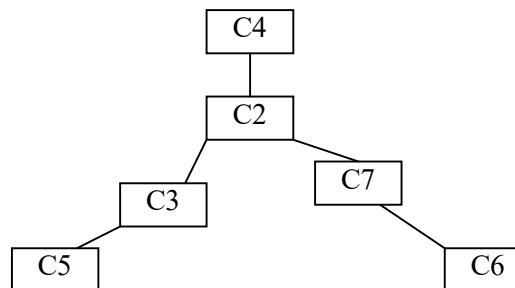


Question 6

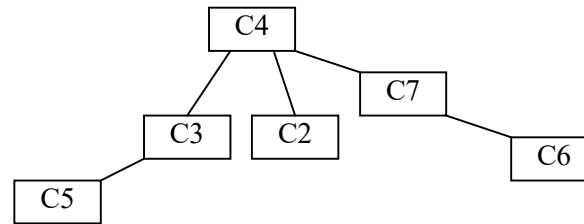
Donner l'arbre de routage du nœud C4 obtenu après l'exécution du protocole STP. Donner l'arbre de routage du nœud C4 obtenu par TRILL. Les résultats sont-ils identiques ? Expliciter la comparaison.

Correction :

L'arbre de routage du nœud C4 qui découle de la topologie obtenue à partir de l'algorithme de l'arbre couvrant est le suivant :



L'arbre de routage qui découle du calcul de l'algorithme de Dijkstra au nœud C4 est le suivant :



On voit tout de suite que pour atteindre les nœuds C5 et C6, il n'y a que 2 liens avec TRILL au lieu des 3 à traverser avec l'arbre couvrant du STP. On observe en plus que C2 n'est plus un passage obligé pour joindre les autres commutateurs.

On peut faire plusieurs conjectures. La première est liée aux algorithmes qui sont de différentes natures. Le STP fait un calcul distribué sur l'ensemble du réseau local commuté (par exemple élection d'un commutateur racine qui n'existe pas dans un protocole à état des liens) après avoir construit un arbre couvrant avec une racine, alors qu'un protocole à état des liens fait N (N étant le nombre de commutateurs) calculs centralisés concurrents avec l'algorithme de Dijkstra.

Question 7

Dans l'entête des messages protocolaires TRILL, il y a un champ de type TTL (Time To Live), à quoi pourrait-il servir ?

Correction :

Comme on a un routage de type Internet, il est possible, même si c'est a priori plus difficile avec un protocole à état des liens, de créer des boucles dans le routage. Le TTL est la meilleure façon d'éliminer les datagrammes capturés dans ces boucles.

Question 8

Du point de vue d'un Data Center, quel est, selon vous, l'avantage du routage global obtenu à l'aide de TRILL par rapport à celui obtenu avec le Spanning Tree ? Donner les arguments les plus importants selon vous.

Correction :

- Le STP élit une racine, qui est un nœud de routage devenant par la force des choses très emprunté. C'est, par construction, un SPOF (single point of failure). Il est vulnérable aux pannes. Compte tenu de sa forte utilisation, c'est susceptible de se produire plus fréquemment. Il n'y a pas cet inconvénient avec un routage à état des liens qui répartit le routage.
- De plus, un routage établi suivant l'algorithme de Dijkstra, garantit que le routage obtenu est optimal en empruntant tous les liens disponibles. Par construction, le routage utilise tous les liens et équilibre la charge sur l'ensemble du réseau et des équipements.
- En cas de panne, le protocole d'élection de nœud racine, puis la construction d'un arbre couvrant, prennent du temps. Probablement plus qu'un calcul de chemin optimal par Dijkstra.

Un routage TRILL est donc a priori plus efficace qu'un routage STP mais pour l'évaluer vraiment, il faut y regarder de plus près. Personnellement, j'ai des doutes.

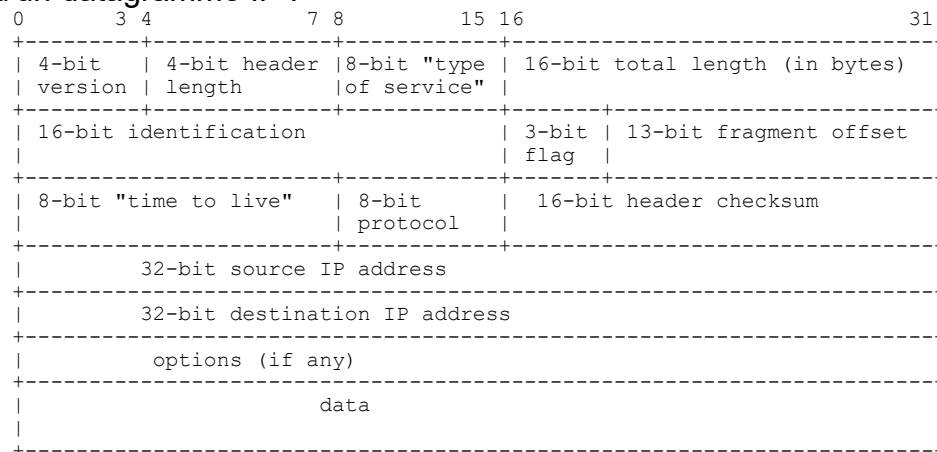
Exercice 3 : L'encapsulation contenant un VLAN.

Dans cet exercice on subit plusieurs encapsulation successives... pas aisé mais c'est intéressant.

Structure d'une trame Ethernet :

AdresseMAC destination	Adresse MAC source	Type	Charge utile - Données	FCS- contrôle d'erreur
6 octets	6 octets	2 octets	46 à 1500 octets	4 octets

Entête d'un datagramme IP :



Entête IPv6 :

		Fixed header format																															
Offsets	Octet	0								1								2								3							
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	Version				Traffic Class								Flow Label																			
4	32	Payload Length																Next Header								Hop Limit							
8	64	Source Address																															
12	96																																
16	128																																
20	160																																
24	192	Destination Address																															
28	224																																
32	256																																
36	288																																

Source : <https://en.wikipedia.org/wiki/IPv6> packet consulté le 27/03/2017 22h45

Entête d'un segment TCP :

		TCP Header																															
Offsets	Octet	0								1								2								3							
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	Source port																Destination port															
4	32	Sequence number																															
8	64	Acknowledgment number (if ACK set)																															
12	96	Data offset				Reserved 0 0 0			N S	C W	E C	U R	A C	P S	R S	S Y	F I	Window Size															
									R	E	G	K	H	T	N	N																	
16	128	Checksum																Urgent pointer (if URG set)															
20	160	Options (if <i>data offset</i> > 5. Padded at the end with "0" bytes if necessary.)																															
...	...	...																															

Vous récupérez la trace ci-dessous avec un analyseur de protocole Wireshark.

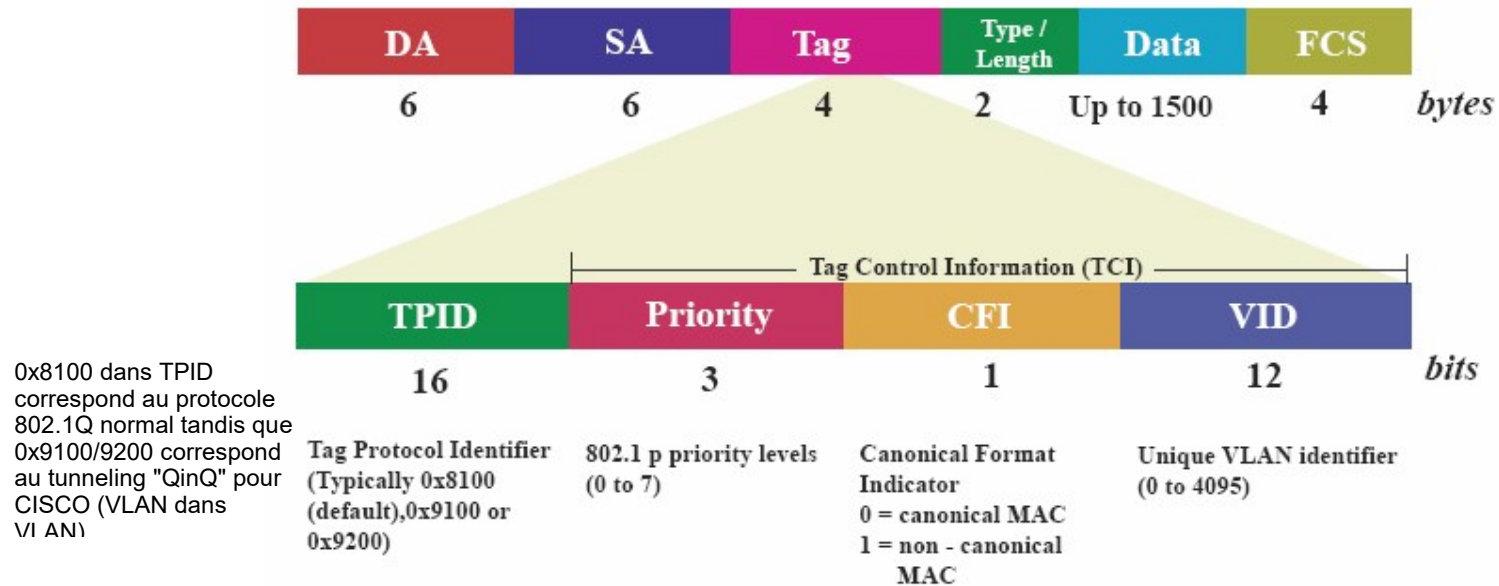
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	2001:67c:2158:a019::ace	2001:0:5ef5:79fd:380c:1d57:a601:24...	TCP	118	53104→13788 [SYN] Seq=0 Win=8192 Len=0 MSS=1412 WS=4 SACK_PERM=1
2	0.071720	2001:0:5ef5:79fd:380c:...	2001:67c:2158:a019::ace	TCP	114	13788→53104 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1220 WS=256 SACK_PERM=1
3	0.072227	2001:67c:2158:a019::ace	2001:0:5ef5:79fd:380c:1d57:a601:24...	TCP	106	53104→13788 [ACK] Seq=1 Ack=1 Win=65880 Len=0
4	0.074223	2001:67c:2158:a019::ace	2001:0:5ef5:79fd:380c:1d57:a601:24...	TCP	202	53104→13788 [PSH, ACK] Seq=1 Ack=1 Win=65880 Len=96

▶	Frame 1: 118 bytes on wire (944 bits), 118 bytes captured (944 bits) on interface 0
▲	Ethernet II, Src: Netgear_35:9b:b2 (20:4e:7f:35:9b:b2), Dst: CiscoInc_08:2d:30 (b4:14:89:08:2d:30)
▶	Destination: CiscoInc_08:2d:30 (b4:14:89:08:2d:30)
▶	Source: Netgear_35:9b:b2 (20:4e:7f:35:9b:b2)
	Type: 802.1Q Virtual LAN (0x8100)
▲	802.1Q Virtual LAN, PRI: 0, CFI: 0, ID: 4
	000. = Priority: Best Effort (default) (0)
	...0 = CFI: Canonical (0)
	 0000 0000 0100 = ID: 4
	Type: PPPoE Session (0x8864)
▲	PPP-over-Ethernet Session
	0001 = Version: 1
	 0001 = Type: 1
	Code: Session Data (0x00)
	Session ID: 0x8122
	Payload Length: 94
▲	Point-to-Point Protocol
	Protocol: Internet Protocol version 4 (0x0021)
▶	Internet Protocol Version 4, Src: 213.141.154.170, Dst: 213.79.83.1
▶	Internet Protocol Version 6, Src: 2001:67c:2158:a019::ace, Dst: 2001:0:5ef5:79fd:380c:1d57:a601:24fa
▶	Transmission Control Protocol, Src Port: 53104, Dst Port: 13788, Seq: 0, Len: 0

0000	b4 14 89 08 2d 30 20 4e	7f 35 9b b2 81 00 00 04	-0 N .5.....
0010	88 64 11 00 81 22 00 5e	00 21 45 00 00 5c 00 00	.d..."^ .!E..\.
0020	40 00 40 29 a1 f0 d5 8d	9a aa d5 4f 53 01 60 00	@.@).... ..OS.^.
0030	00 00 00 20 06 3f 20 01	06 7c 21 58 a0 19 00 00	? . .!X....
0040	00 00 00 00 0a ce 20 01	00 00 5e f5 79 fd 38 0c	^..y.8.
0050	1d 57 a6 01 24 fa cf 70	35 dc 44 dd a0 3d 00 00	.W..\$.p 5.D..=..
0060	00 00 80 02 20 00 58 cf	00 00 02 04 05 84 01 03	X.
0070	03 02 01 01 04 02		

Question 1. Traversée des sous-couches constituant la couche liaison.

1.1 On vous donne la définition d'une entête VLAN 802.1Q ci dessous. Dans cette entête le champ Tag est positionné entre l'adresse MAC source et le champ type de la trame Ethernet telle qu'elle est vu habituellement dans la couche MAC (Medium Access Control).



Source <http://sclabs.blogspot.fr/2014/10/ccnp-switch-vlans-and-trunks.html>, consultée le 27/03/2017 à 21h45.

Retrouvez les champs TPID et VID dans la trace en hexadécimal Ethernet ci-dessous. Entourez ou surlignez les. Donner leur valeur pour chacun des champs.

```
b4|14|89|08|2d|30|20|4e|7f|35|9b|b2|81|00|00|04|88|64|11|00|81|22|00|5e|00|21|45|00|00|5c|00|
00|40|00|40|29|a1|f0|d5|8d|9a|aa|d5|4f|53|01|60|00|00|00|00|20|06|3f|20|01|06|7c|21|58|a0|19|
00|00|00|00|00|00|0a|ce|20|01|00|00|5e|f5|79|fd|38|0c|1d|57|a6|01|24|fa|cf|70|35|dc|44|dd|a0|
3d|00|00|00|00|80|02|20|00|58|cf|00|00|02|04|05|84|01|03|03|02|01|01|04|02|
```

Correction :

```

b4|14|89|08|2d|30|20|4e|7f|35|9b|b2|81|00|00|04|88|64|11|00|81|22|00|5e|00|21|45|00|00|5c|00|
#MAC dest      puis src      TPID puis VLANID Type

```

On peut étendre le champ TCI en binaire ce qui donne :

```

10000001 00000000 0000 0000 00000100
TPID(16b), Priorité(3b), CFI(1b), VID(12b)

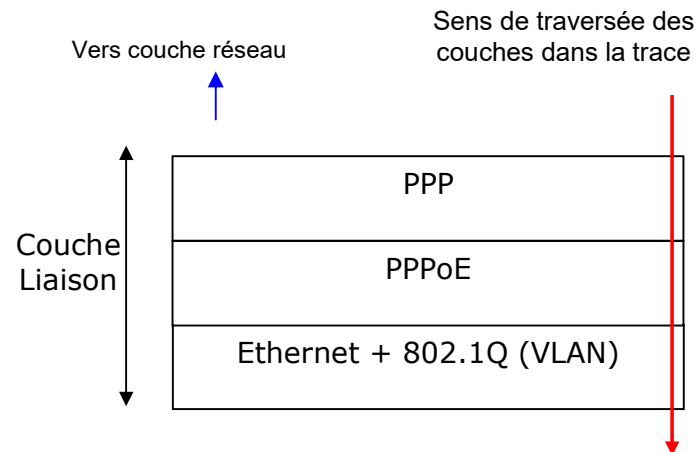
```

```

00|40|00|40|29|a1|f0|d5|8d|9a|aa|d5|4f|53|01|60|00|00|00|00|20|06|3f|20|01|06|7c|21|58|a0|19|
00|00|00|00|00|00|0a|ce|20|01|00|00|5e|f5|79|fd|38|0c|1d|57|a6|01|24|fa|cf|70|35|dc|44|dd|a0|
3d|00|00|00|00|80|02|20|00|58|cf|00|00|02|04|05|84|01|03|03|02|01|01|04|02|

```

La trame Ethernet contient une trame combinaison de Point-to-Point Protocol (PPP)/PPP over Ethernet (PPPoE) d'après le type Ethernet 0x8864 contenu dans la trace Wireshark récupérée pour cet exercice. L'organisation des sous-couches liaison avec PPPoE est la suivante :



Avec :

PPPoE header:<http://www.networksorcery.com/enp/protocol/pppoe.htm>, consultée le 27/03 /2017 à 22h30.

00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Version				Type				Code								Session ID															
Length																Data ...															

1.2. Quelle est la longueur de la trame PPP que nous indique l'entête PPOE dans la trace en hexadécimal ci-dessus, donner la valeur puis entourez ou surlignez la valeur en hexadécimal dans la trace ci-dessus.

Correction :

b4|14|89|08|2d|30|20|4e|7f|35|9b|b2|81|00|00|04|88|64|11|00|81|22|00|5e|00|21|45|00|00|5c|00|

Partie d'entête sur laquelle nous avons travaillé (en fond noir), suivie de l'entête PPPoE (en fond jaune).

La longueur est encadrée. Elle vaut 005e, soit 5*16 + 14 (valeur décimale du chiffre hexadécimal e), soit 80 + 14, qui vaut c'est-à-dire 94 octets. La longueur des données pour PPPoE qui correspondent à la trame PPP est de 94 octets.

Après l'entête PPPoE et dans la partie charge utile (Data) PPPoE, on trouve l'entête PPP et les données PPP.

00|40|00|40|29|a1|f0|d5|8d|9a|aa|d5|4f|53|01|60|00|00|00|00|20|06|3f|20|01|06|7c|21|58|a0|19|

00|00|00|00|00|00|0a|ce|20|01|00|00|5e|f5|79|fd|38|0c|1d|57|a6|01|24|fa|cf|70|35|dc|44|dd|a0|

3d|00|00|00|00|80|02|20|00|58|cf|00|00|02|04|05|84|01|03|03|02|01|01|04|02|

La trame PPP est très simple, l'entête est sur un ou deux octets puis les données suivent. Ici, l'entête PPP, encadrée sur fond blanc, indique le protocole transporté. La valeur 0x0021 correspond à IPv4. Pour information, la valeur 0x0057 correspondrait à IPv6.

Par rapport aux objectifs pédagogiques liés à l'exploration du domaine des réseaux locaux, l'exercice pourrait s'arrêter là. La suite permet d'explorer une série d'encapsulations élaborée. Le reste de l'exercice est à votre convenance.

Question 2. Traversée des sous-couches constituant la couche réseau.

2.1. Dans la trace en hexadécimal ci-dessus entourer les entêtes du datagramme IPv4 et du datagramme IPv6 dès que vous aurez assez d'informations.

Correction :

b4|14|89|08|2d|30|20|4e|7f|35|9b|b2|81|00|00|04|88|64|11|00|81|22|00|5e|00|21|45|00|00|5c|00|

entête PPP, suivie de l'entête du datagramme IP v4 en surligné jaune

00|40|00|40|29|a1|f0|d5|8d|9a|aa|d5|4f|53|01|60|00|00|00|00|20|06|3f|20|01|06|7c|21|58|a0|19|

en surligné bleu turquoise on trouve l'entête IPv6

00|00|00|00|00|00|0a|ce|20|01|00|00|5e|f5|79|fd|38|0c|1d|57|a6|01|24|fa|cf|70|35|dc|44|dd|a0|

3d|00|00|00|00|80|02|20|00|58|cf|00|00|02|04|05|84|01|03|03|02|01|01|04|02|

2.2. Quelle est la longueur de l'entête du datagramme IPv4 en décimal ? Y a-t-il des options ? Pourquoi ?

Correction :

La longueur indiquée dans l'entête IPv4 est 5 qu'il faut multiplier par 4 pour avoir le nombre d'octets. Ce qui fait 20 octets. Comme la longueur est de 20 octets, il n'y a pas d'options dans l'entête du datagramme IPv4.

L'entête d'un datagramme IPv6 est la suivante :

Fixed header format																																		
Offsets	Octet	0								1								2								3								
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	
0	0	Version				Traffic Class								Flow Label																				
4	32	Payload Length																Next Header								Hop Limit								
8	64	Source Address																																
12	96																																	
16	128																																	
20	160																																	
24	192																																	
28	224	Destination Address																																
32	256																																	
36	288																																	

2.3. Quelle est la taille de l'entête IPv6 dans la trace ? Est-elle identique pour tous les datagrammes IPv6, c'est-à-dire cette taille est-elle universelle et invariable pour tout datagramme IPv6 ?

Correction :

L'entête IPv6 est de 40 octets (8 + 16 + 16 octets).

Il n'est pas prévu d'options dans l'entête IPv6 mais on utilise le champ next header si on veut donner des informations de gestion complémentaires ou si on veut indiquer le protocole transporté dans le datagramme. En conséquence, les entête IPv6 sont de taille fixe, c'est ce qui fait leur principal intérêt pour les traiter plus efficacement dans l'électronique embarquée des routeurs.

2.4. Quelle est la valeur du champ DSCP dans les deux entêtes IPv4 et IPv6 ? A quelle(s) classe(s) de qualité de service vue en cours cela correspond ? Est-ce cohérent ?

Correction :

45|00|00|5c|00|00|40|00|40|29|a1|f0|d5|8d|9a|aa|d5|4f|53|01 entête IPv4, champ ToS encadré qui contient le champ DSCP (1 octet)

60|00|00|00|00|20|06|3f|20|01|06|7c|21|58|a0|19|00|00|00|00|00|00|0a|ce|20|01|00|00|5e|f5|79|fd|38|0c|1d|57|a6|01|24|fa entête IPv6, champ Traffic class encadré qui contient le champ DSCP (1 octet pas tout à fait à la même place).

Pas nécessaire d'aller plus en détail, les champs pour chacune des entêtes sont à zéro. On tombe alors sur la classe best Effort. C'est cohérent, on a la même qualité de service dans les deux entêtes sachant que le datagramme IPv4 contient le datagramme IPv6.

2.5. La trace Wireshark donne l'adresse source IPv6 suivante : 2001:67c:2158:a019::ace retrouvez l'adresse IPv6 correspondante en notation étendue dans la trace en hexadécimal ci-dessus.

Correction :

L'entête IPv6 est copiée ci-dessous. On a encadré l'adresse IPv6 source.

60|00|00|00|00|20|06|3f|20|01|06|7c|21|58|a0|19|00|00|00|00|00|00|0a|ce|20|01|00|00|5e|f5|79|fd|38|0c|1d|57|a6|01|24|fa

Voilà l'adresse étendue repérée dans la trace 20|01|06|7c|21|58|a0|19|00|00|00|00|00|00|0a|ce On peut la ré-écrire en notation IPv6 : 2001:067c:2158:a019:0000:0000:0000:0ace On peut enlever des 0 et en comprimant la suite de 0 qu'on remplace par ::, on obtient alors 2001:67c:2158:a019::ace On la compare maintenant avec 2001:67c:2158:a019::ace. C'est identique donc correct.

2.6. La trace Wireshark donne l'adresse destination IPv6 suivante : 2001::5ef5:79fd:380c:1d57:a601:24fa retrouvez l'adresse IPv6 correspondante en notation étendue dans la trace en hexadécimal ci-dessus.

Correction :

L'entête IPv6 est copiée ci-dessous. On a encadré l'adresse IPv6 destination.

```
60|00|00|00|00|20|06|3f|20|01|06|7c|21|58|a0|19|00|00|00|00|00|00|0a|ce|
20|01|00|00|5e|f5|79|fd|38|0c|1d|57|a6|01|24|fa
```

On procède de la même façon :

Adresse IPv6 compacte : 2001::5ef5:79fd:380c:1d57:a601:24fa

Adresse IPv6 étendue : 2001:0000:5ef5:79fd:380c:1d57:a601:24fa

Cela correspond.

2.7. Question pour le fun. Les adresses 2001::/32 sont réservées à Teredo (« Tunneling IPv6 over UDP through NAT » (RFC 4380). Teredo est un protocole de type tunnel. Un tunnel sert à établir un chemin spécifique entre deux interfaces internet et à acheminer de l'IPv6 dans de l'IPv4.

Une des particularités de ces adresses c'est qu'elles contiennent des informations sur le serveur, le client et le port utilisés par les extrémités du tunnel.

Retrouvez les valeurs 94.245.121.253 (adresse IPv4), 58024 (port), 89.254.219.5 (adresse IPv4) dans l'adresse IPv6 du destinataire ci-dessous.

2001::5ef5:79fd:380c:1d57:a601:24fa

Petite indication, il faut travailler sur les représentations en hexadécimal des informations à trouver⁶.

Correction :

On doit retrouver les valeurs 94.245.121.253 (adresse IPv4), 58024 (port), 89.254.219.5 dans l'adresse. Pas la peine de passer en notation étendue sur la zone compactée entre 2001 et 5ef5 puisqu'elle contient des 0.

Pour trouver les différentes adresses données en notation décimale pointée, il faut déjà convertir en hexadécimal et après aviser.

L'adresse IPv4 du serveur Teredo 94.245.121.253 donne 5E.F5.79.FD soit 5EF5:79FD. Sa position est encadrée dans l'adresse IPv6 : 2001::5ef5:79fd:380c:1d57:a601:24fa

• Le numéro de port de l'application cliente 58024 donne E2A8, ça ne suffit pas. On applique not(E2A8) et on trouve 1D57. Sa position est encadrée dans l'adresse IPv6 : 2001::5ef5:79fd:380c:1d57:a601:24fa

⁶ Pour information dans l'adresse IPv6 on trouve 0x380C qui s'écrit en binaire 0011 1000 0000 1100, il se décompose en C0RAAAAUGAAAAAAAAA. Si C vaut 1 c'est que l'équipement est derrière une passerelle NAT. Dans cet exercice, on en déduit que ce n'est pas le cas. Les 12 bits A sont tirés aléatoirement.

- L'adresse IPv4 du client 89.254.219.5 donne 59.FE.DB.05 soit 59FEDB05, ça ne suffit pas. On applique `not(59FEDB05)` et on trouve A60124FA. Sa position est encadrée dans l'adresse IPv6 : 2001::5ef5:79fd:380c:1d57:a601:24fa
- La première des valeurs est en clair, on dit que les deux autres sont "obfusquées".

Question 3. Traversée de la couche transport.

Décrivez les encapsulations successives telles que vous les comprenez dans la trace Wireshark de la trame 1. Combien y en a-t-il au total ? Faites un dessin si cela peut vous aider à répondre.

Correction :

C'est plus facile de travailler à partir du segment TCP et de son contenu.

```
b4|14|89|08|2d|30|20|4e|7f|35|9b|b2|81|00|00|04|88|64|11|00|81|22|00|
5e|00|21|45|00|00|5c|00|00|40|00|40|29|a1|f0|d5|8d|9a|aa|d5|4f|53|01|
60|00|00|00|00|20|06|3f|20|01|06|7c|21|58|a0|19|00|00|00|00|00|0a|
ce|20|01|00|00|5e|f5|79|fd|38|0c|1d|57|a6|01|24|fa|cf|70|35|dc|44|dd|
a0|3d|00|00|00|00|80|02|20|00|58|cf|00|00|02|04|05|84|01|03|03|02|01|
01|04|02|
```

Surligné en rose, l'entête TCP normale. Le reste c'est soit des options soit des données.

C'est un segment d'ouverture de connexion, il y a fort peu de chance qu'il contienne des données. Par contre, on remarque dans la trace Wireshark qu'il y a des options qui sont indiquées : MSS (Maximum Segment Size), WS (Window Scale), SACK (Selective Ack). A priori l'entête TCP fait plus de 20 octets...

Pour être précis, il faut regarder si le segment TCP contient des données à l'aide du champ Data Offset qui est encadré dans la trame. Il vaut 8. Il indique le nombre de mots de 4 octets dans l'entête TCP. Après commencent les données. C'est toujours un multiple de 4 octets si les options ne sont pas à la bonne longueur, on ajoute du bourrage. Ici si on donc $4 * 8$, soit 32 octets, on tombe exactement à la fin de la trace. Il n'y a donc pas de données dans ce segment. Par conséquent, le segment n'encapsule pas des données.

On récapitule les encapsulations :

- Le segment TCP est encapsulé dans un datagramme IPv6. (1 encapsulation)
- Le datagramme IPv6 est encapsulé dans un datagramme IPv4. (1 encapsulation)
- Le datagramme IPv4 est encapsulé dans une trame PPP. (1 encapsulation)
- La trame PPP est encapsulée dans une trame PPPoE. (1 encapsulation)
- La trame PPPoE est encapsulée dans une trame Ethernet 802.1Q. (1 encapsulation)
- On ne compte pas d'encapsulation pour la couche physique.

Au total on a 5 encapsulations.



Exercice 4 : Une histoire de VLANs

On prend deux trames d'une trace Wireshark qui correspondent à un ping et à sa réponse. On vous demande de donner le numéro des VLAN qui sont traversés par les datagrammes ICMP capturés.

ICMP ECHO_REQUEST :

No.	Time	Source	Destination	Protocol	Lengt	Info
1	0.000000	10.118.10.1	10.118.10.2	ICMP	122	Echo (ping) request ic
> Ethernet II, Src: Cisco_df:ae:18 (00:13:c3:df:ae:18), Dst: Cisco_1b:a4:d8						
0000	00 1b d4 1b a4 d8 00 13	c3 df ae 18 81 00 00 76	v			
0010	81 00 00 0a 08 00 45 00	00 64 00 0f 00 00 ff 01	E- d.....			
0020	92 9b 0a 76 0a 01 0a 76	0a 02 08 00 ce b7 00 03	...v...v.....			
0030	00 00 00 00 00 00 00 1f	af 70 ab cd ab cd ab cd	p.....			
0040	ab cd ab cd ab cd ab cd	ab cd ab cd ab cd ab cd				
0050	ab cd ab cd ab cd ab cd	ab cd ab cd ab cd ab cd				
0060	ab cd ab cd ab cd ab cd	ab cd ab cd ab cd ab cd				
0070	ab cd ab cd ab cd ab cd	ab cd				

ICMP ECHO_REPLY :

No.	Time	Source	Destination	Protocol	Lengt	Info
1	0.000000	10.118.10.1	10.118.10.2	ICMP	122	Echo (ping) request ic
2	0.000858	10.118.10.2	10.118.10.1	ICMP	122	Echo (ping) reply ic
v Ethernet II, Src: Cisco_1b:a4:d8 (00:1b:d4:1b:a4:d8), Dst: Cisco_df:ae:18						
0000	00 13 c3 df ae 18 00 1b	d4 1b a4 d8 81 00 00 76	v			
0010	81 00 00 0a 08 00 45 00	00 64 00 0f 00 00 ff 01	E- d.....			
0020	92 9b 0a 76 0a 02 0a 76	0a 01 00 00 d6 b7 00 03	...v...v.....			
0030	00 00 00 00 00 00 00 1f	af 70 ab cd ab cd ab cd	p.....			
0040	ab cd ab cd ab cd ab cd	ab cd ab cd ab cd ab cd				
0050	ab cd ab cd ab cd ab cd	ab cd ab cd ab cd ab cd				
0060	ab cd ab cd ab cd ab cd	ab cd ab cd ab cd ab cd				
0070	ab cd ab cd ab cd ab cd	ab cd				

Correction :

On a vu dans l'exercice précédent que, si on avait à traiter une trame Ethernet qui était marquée par un commutateur en 802.1Q, le TPID - Tag Protocol Identifier, d'une longueur de 16bits prenait la valeur 0x8100. C'est ce qu'on retrouve ici pour les deux trames ICMP, on a encadré cette valeur en bleu.

No.	Time	Source	Destination	Protocol	Lengt	Info
1	0.000000	10.118.10.1	10.118.10.2	ICMP	122	Echo (ping) request ic
> Ethernet II, Src: Cisco_df:ae:18 (00:13:c3:df:ae:18), Dst: Cisco_1b:a4:d8						
0000	00 1b d4 1b a4 d8 00 13	c3 df ae 18 81 00 00 76	v			
0010	81 00 00 0a 08 00 45 00	00 64 00 0f 00 00 ff 01	E- d.....			
0020	92 9b 0a 76 0a 01 0a 76	0a 02 08 00 ce b7 00 03	...v...v.....			
0030	00 00 00 00 00 00 00 1f	af 70 ab cd ab cd ab cd	p.....			
0040	ab cd ab cd ab cd ab cd	ab cd ab cd ab cd ab cd				
0050	ab cd ab cd ab cd ab cd	ab cd ab cd ab cd ab cd				
0060	ab cd ab cd ab cd ab cd	ab cd ab cd ab cd ab cd				
0070	ab cd ab cd ab cd ab cd	ab cd				

IP Champ protocol :
"1" = ICMP

1	0.000000	10.118.10.1	10.118.10.2	ICMP	122	Echo (ping) request	id
2	0.000858	10.118.10.2	10.118.10.1	ICMP	122	Echo (ping) reply	id

▼ Ethernet II, Src: Cisco_1b:a4:d8 (00:1b:d4:1b:a4:d8), Dst: Cisco_df:ae:18

0000	00 13 c3 df ae 18 00 1b d4 1b a4 d8	81 00 00 76	v
0010	81 00 00 0a 08 00 45 00 00 64 00 0f 00 00 ff 01	E- -d.....	
0020	92 9b 0a 76 0a 02 0a 76 0a 01 00 00 d6 b7 00 03	...v...v.....	
0030	00 00 00 00 00 00 00 1f af 70 ab cd ab cd ab cd	p.....	
0040	ab cd ab cd ab cd ab cd ab cd ab cd ab cd ab cd		
0050	ab cd ab cd ab cd ab cd ab cd ab cd ab cd ab cd		
0060	ab cd ab cd ab cd ab cd ab cd ab cd ab cd ab cd		
0070	ab cd ab cd ab cd ab cd ab cd ab cd		

On prend les deux octets suivants dont les 12 derniers bits donnent le numéro de VLAN soit 0×076 qui vaut $7 \times 16 + 6$ soit 118 en décimal.

C'est cohérent entre les deux trames ICMP.

Toutefois, pour les deux trames contenant l'échange ICMP, juste après on trouve de nouveau une étiquette 802.1Q avec la présence à nouveau de 0×8100 encadré en rouge. Le VLAN correspondant est $0 \times 00a$ qui vaut 10 en décimal. C'est cohérent pour les deux trames ICMP.

On en conclut que le VLAN 10 est transporté dans le VLAN 118.

Après les 2 étiquettes 802.1Q on trouve bien un champ type qui vaut 0×0800 qui indique un datagramme IP. Et dans le datagramme IP, on trouve le champ `protocol` qui vaut 1 qui est la valeur associée à ICMP.

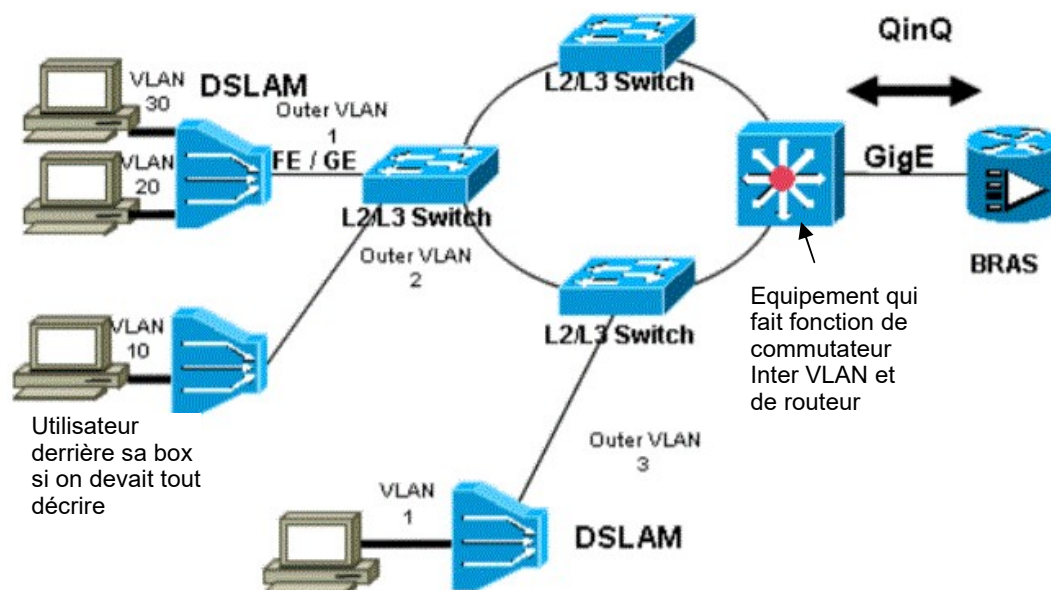
Remarque : Dans les deux trames le TTL vaut 255 ($0 \times ff$), cela veut dire qu'elles n'ont traversé aucun routeur. La valeur du TTL c'est aussi d'une certaine façon la signature de l'OS qui fait tourner le protocole. <https://subinsb.com/default-device-ttl-values/> consultée le 08/12/2020 donne une liste de ces valeurs. Pour ICMP avec un TTL de 255, on peut suspecter que l'OS n'est pas un MACOS, ni un Windows. Il en reste beaucoup, mais c'est déjà indicateur car il y a de nombreuses versions de Windows par exemple.

En perspective : On est en droit de se poser la question de savoir si on est dans une situation d'empilement de VLANs ? En fait non, si c'était le cas, d'après 802.1ad, le premier tag ID à partir de l'entête de trame serait $0 \times 88A8$.

Par contre, CISCO dans ses produits semble utiliser la valeur 0×9100 , la différence est commentée dans <https://www.embeddedsystemtesting.com/2012/09/vlan-standards-qinq.html> consulté le 08/12/2020. Il est à noter que dans un tel tag, le champ CFI est remplacé par DEI, Drop Eligibility Indicator qui sert à l'élimination de trames en cas de congestion du commutateur. Le tag 0×8100 est un tag intérieur, un C-tag, C pour Custommer, associé à un C-VID. $0 \times 88A8$ est un S-tag (Service Tag), il peut y en avoir une série.

802.1ad sert surtout à la commutation de trames en environnement opérateur ou fournisseur d'accès.

Figure 2 *Broadband Ethernet-based DSLAM Model of Q-in-Q VLANs*



VLAN aggregation on a DSLAM will result in a lot of aggregate VLANs that at some point need to be terminated on the broadband remote access servers (BRAS). Although the model could connect the DSLAMs directly to the BRAS, a more common model uses the existing Ethernet-switched network where each DSLAM VLAN ID is tagged with a second tag (Q-in-Q) as it connects into the Ethernet-switched network.

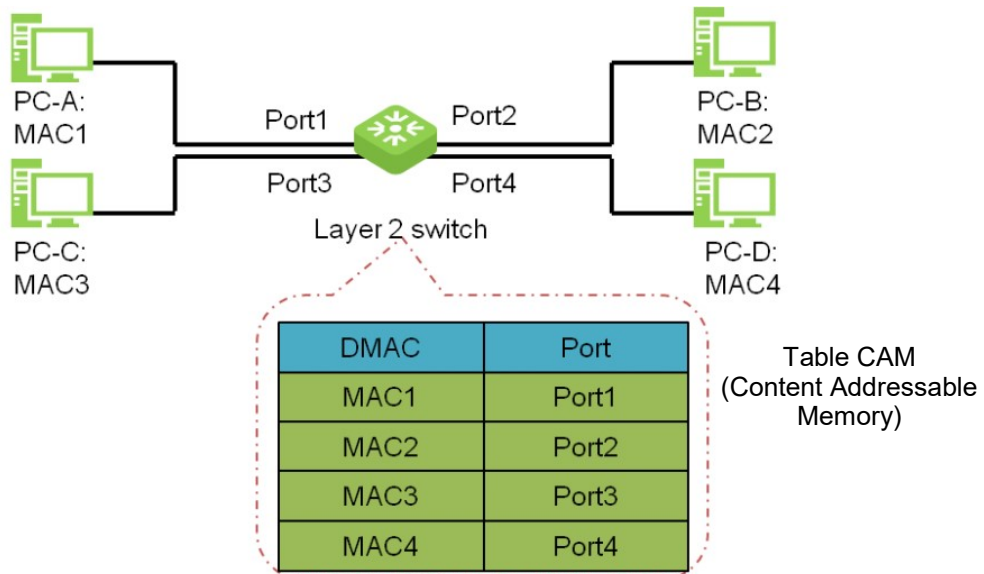
source : https://www.cisco.com/en/US/docs/ios/lanswitch/configuration/guide/lsw_ieee_802.1q.pdf
(08/12/2020)

Cette image donne une idée de l'architecture dont est issue la capture de trame de l'exercice 3 de ce polycopié, et dont l'autre partie est dans la série d'ED sur les tunnels en IPv6. Ça pourrait être une capture dans le réseau d'un opérateur d'un échange IPv6 Teredo initié derrière une box Internet d'abonné relié à son opérateur par PPPoE.

Exercice 5 : Tables de commutation de commutateurs par auto-apprentissage (à faire soi-même, facile)

L'exercice est inspiré de la page suivante : source : <https://forum.huawei.com/enterprise/en/approaching-ne-11-a-packet-s-adventures-on-huawei-routers-layer-2-ethernet/thread/222033-863> consulté le 20/12/2019

Soit le réseau local suivant :

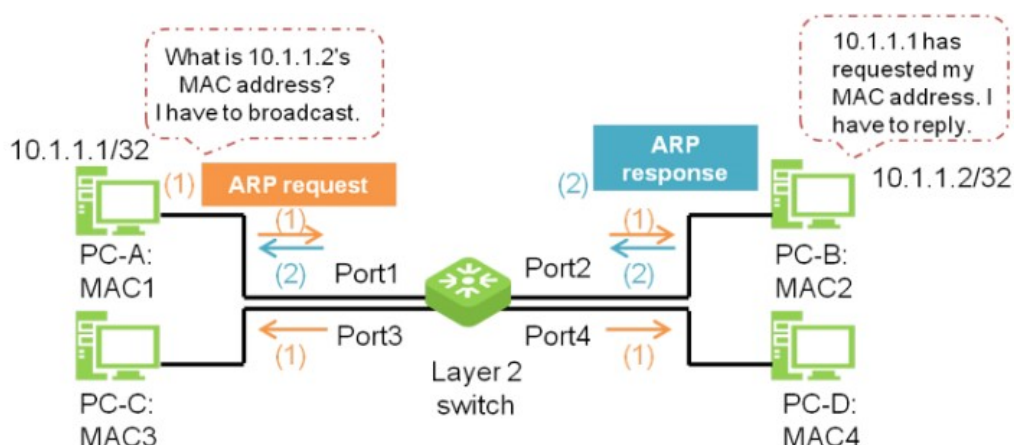


Question 1 : Commutation unicast MAC

Si une trame part de PC-C vers PC-B, comment le commutateur fait-il pour la faire parvenir à destination.

Correction :

Le commutateur reçoit la trame sur le port 3, il examine l'adresse de destination dans la trame reçue et trouve MAC2. Il recherche MAC 2 dans sa table et voit qu'il faut émettre la trame sur le port 2.



Question 2 : Commutation multicast MAC

Le commutateur reçoit une trame contenant une requête ARP de PC-A que se passe-t-il ?

Correction :

Le commutateur reçoit la requête ARP sur le port 1, il la recopie sur tous les autres ports : port 2, port 3, et port 4. Quand la réponse ARP revient elle a pour destination MAC1. C'est une trame unicast MAC, grace à la table de commutation dont il dispose, il envoie la réponse à PC-A puisqu'elle contient l'adresse MAC1 en adresse de destination.

Question 3

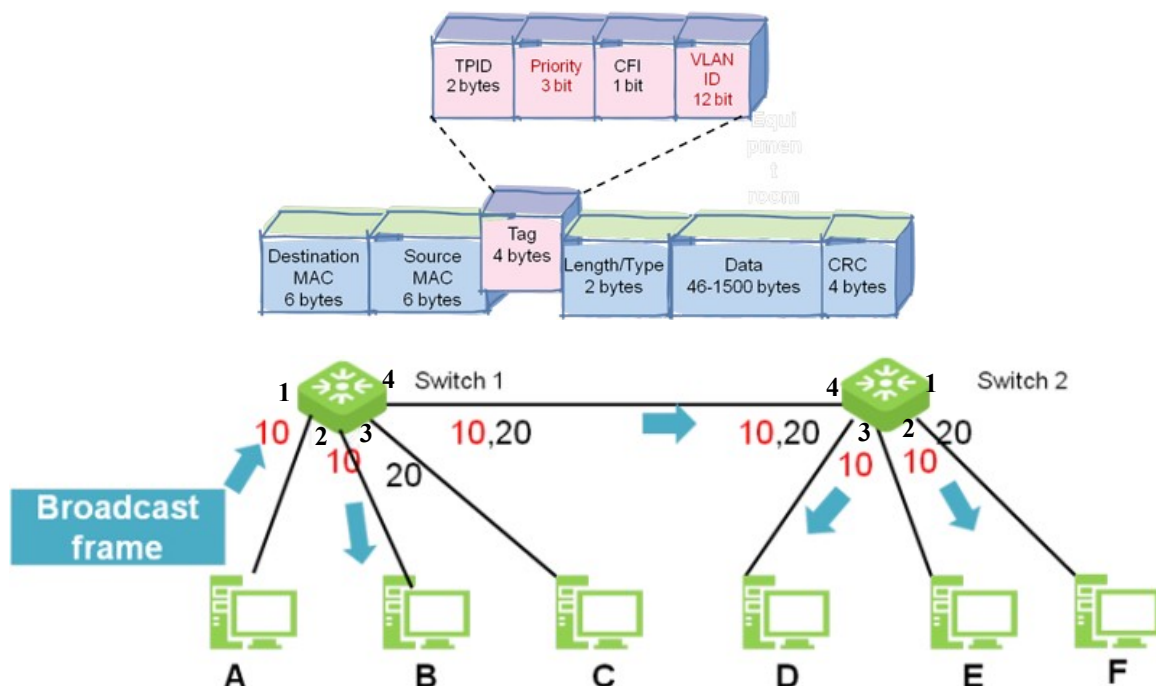
Décrivez comment la table de commutation du commutateur est construite.

Correction :

Quand une trame passe sur un port, le commutateur associe l'adresse source de la trame avec son port de réception. Par exemple, c'est quand PC-A a envoyé une trame qu'il a associé MAC1 avec son port 1.

Comme les machines sont susceptibles de bouger sur le réseau, pour chaque entrée de la table il est associé une temporisation. Si l'adresse MAC associée à un port n'a pas été lue sur celui-ci au bout de ce délai, elle est invalidée dans la table.

On donne l'extension IEEE802.1Q pour que les trames puissent circuler dans des VLANs :

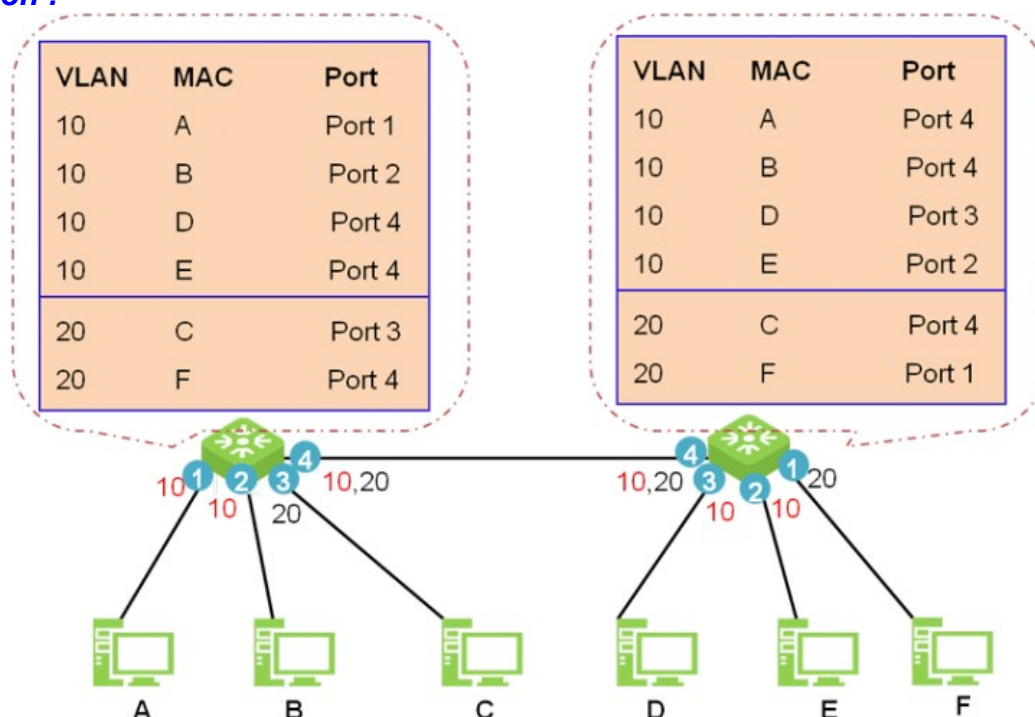


On divise un réseau local avec des réseaux logiques, VLAN pour Virtual Local Area Networks. Un port peut être assigné à un ou plusieurs VLAN. On va donc ajouter une colonne VLANid dans la table de commutation du commutateur.

Les stations A, B, D, E appartiennent au VLAN 10. Les stations C et F appartiennent au VLAN 20.

Question 4

Donner les tables de commutation des commutateurs 1 et 2.

Correction :

Plus de détails sur les commutateurs et les VLANs dans "Packet Guide to Routing and Switching by Bruce Hartpence. Chapter 4. VLANs and Trunking" <https://www.oreilly.com/library/view/packet-guide-to/9781449311315/ch04.html>, consulté le 29/12/2019.

Question 5

Par quelle fonction ou quel équipement, les VLAN communiquent entre eux, c'est-à-dire échangent des flux qui sont normalement isolés les uns des autres et donc associés à des adresses de réseaux locaux ? Un répéteur ? Un commutateur ? Un routeur ? Une passerelle ? Justifier votre choix.

Correction :

Ca ne peut être un répéteur, il ne fait que répéter un signal d'un port sur un autre; c'est un équipement de niveau physique.

Une passerelle se préoccupe d'aspects plus en relation avec l'appliquatif, c'est donc trop haut niveau pour accomplir ce genre de relaiage.

Un commutateur, il a la répartition des destinataires par VLAN, il pourrait donc effectuer le relaiage d'un VLAN à un autre.

Un routeur ou un service de routage peut relayer d'un VLAN sur un autre surtout si les VLAN instancient des réseaux IP différents.

Exercice 6 : Bridge Protocol Data Unit dans un contexte VLAN

Il faut noter pour la mise en œuvre de l'arbre couvrant que les trames Ethernet n'ont plus tout à fait le même format, elles sont exprimées au format IEEE 802.3. Le champ Type est remplacé par un champ longueur (dénnoté L/T plus loin).

Comme la norme qui définit STP est dans la catégorie IEEE 802.1, une couche LLC pour Logical Link Control, IEEE 802.2 apparaît dans le découpage en sous-couches de la liaison pour les réseaux locaux.

Network			
Data Link	IEEE 802.2 Logical Link Control Layer (LLC)		
	IEEE 802.3 CSMA/CD Medium Access Control Layer		
Physical	802.3 - 10Base5	802.3a - 10Base2	802.3i - 10BaseT

Source : <https://www.telecomworld101.com/8022.html>, consultée le 26/12/2019

Dans la trame, il apparaît un entête pour cette sous-couche de la couche 2 spécifique aux protocoles de réseaux locaux. La figure ci-dessous compare une trame Ethernet (document DIX807) encore nommée Ethernet II et une trame IEEE802.3/ISO8802.3.

Preamble	Dest Addr	Source Addr	Type	Info	FCS
8 bytes	6 bytes	6 bytes	2 bytes	46<=N<=1500 bytes	4 bytes

Ethernet

IEEE 802.2 header									
Preamble	SFD	Dest Addr	Source Addr	Length	DSAP	SSAP	Ctrl	Info	FCS
7 bytes	1 byte	6 bytes	6 bytes	2 bytes	1 byte	1 byte	1 byte	variable	4 bytes

IEEE 802.3

Source : <http://www.danzig.jct.ac.il/tcp-ip-lab/ibm-tutorial/3376c28.html>, consultée le 26/12/2019

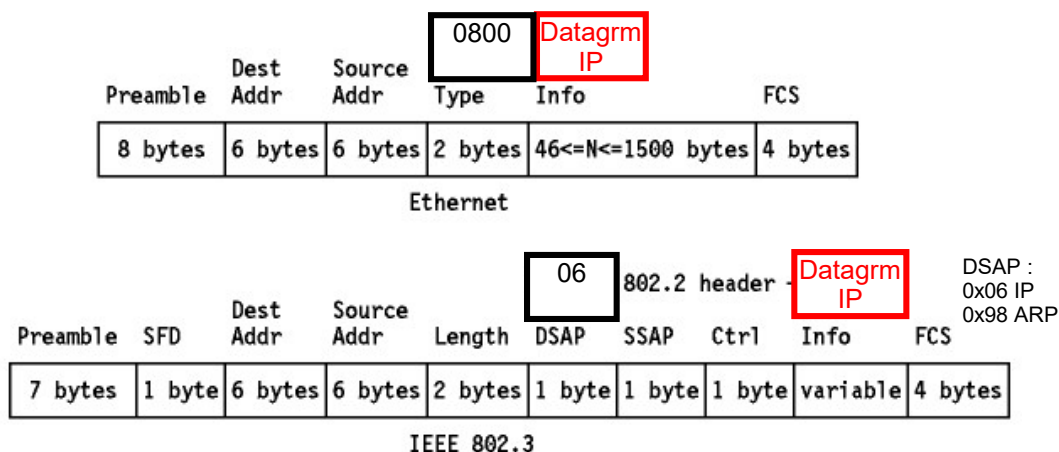
Pour mieux comprendre la différence on peut essayer de prendre des points d'observations caractéristiques :

- Le champ type pour une trame Ethernet, quand il contient un datagramme IP, contient 0x0800. C'est supérieur à 1500 octets, ou à 0x05DC en hexadécimal. Un champ longueur, pour la trame IEEE802.3 va prendre une valeur entre 0x002E (46 octets) et 0x5DC (1500 octets). C'est comme ça qu'on peut faire la différence entre les trames Ethernet DIX80, et les trames IEEE802.3, norme qui est aussi enregistrée sous le numéro ISO8802.3.

Accessoirement, si on souhaite identifier son propre protocole de liaison (propriétaire) dans le champ type, on peut utiliser un nombre entre 0 et 2D (2E-1). De mémoire, c'est ce que faisait Chorus Systems pour son protocole d'IPC (InterProcess Communication) dans le micro-noyau ChorusOS.

- On peut aussi chercher à voir où se placerait un datagramme IPv4 dans une trame Ethernet DIX et dans une trame IEEE802.3. Le schéma ci-dessous éclaire ce point de vue :

⁷ "Historiquement Ethernet est un standard de fait décrit depuis 1980 par les spécifications Ethernet / DIX. Par ailleurs, l'IEEE a publié son propre standard [IEEE 802.3](#) en 1983, s'inspirant de ce standard de fait. Il existe donc en fait un standard Ethernet II / DIX d'une part (de 1982), et une norme [IEEE 802.3](#) d'autre part (de 1983). Les deux standards sont interopérables. Par la suite les mises à jour normatives ont été formalisées par l'IEEE, et [802.3](#) a du reste pris officiellement en compte les aspects de DIX en 1998 (révision 802.3-1998)" source wikipedia, <https://fr.wikipedia.org/wiki/Ethernet>, lien consulté le 26/12/2019.



Attention, le DSAP 0x06 est donné dans deux sources consultées le 12/01/2020:

- <http://www-inf.int-evry.fr/~hennequi/OBS/NetInfos/ieee-lsap-list>,
- https://fr.wikipedia.org/wiki/Contr%C3%B4le_de_la_liaison_logique.

Je n'ai pas trouvé d'autres sources plus normatives. Mais la pratique normale pour encapsuler de l'IP dans LLC semble être de passer par une couche intermédiaire au dessus de LLC qui s'appelle SNAP (Sub-Network Access Protocol). SNAP⁸⁹ utilise le DSAP 0xAA, et possède sa propre entête de 5 octets :

```

+-----+-----+-----+-----+-----+
| 0      0      0      | EtherType | 802.2 SNAP, EtherType =0800 (IP), 0806 (ARP)
+-----+-----+-----+-----+-----+

```

On voit que SNAP introduit un champ `EtherType` de plus.

L'abstraction Service Access Point est typique du modèle ISO. Le concept de SAP (Service Access Point) par couche est propre au modèle ISO. Les SAP sont typés par couches, il y a des LSAP, Link-SAP, il y a des NSAP, Network-SAP, TSAP, Transport-SAP... Chaque couche expose des SAP pour accéder à ses services, par exemple la couche Transport offre des TSAP. Un SAP est donc un point d'accès à une couche protocolaire. Les TSAP sont associés à un NSAP (Network) qui lui-même est associé à un LSAP (Link).

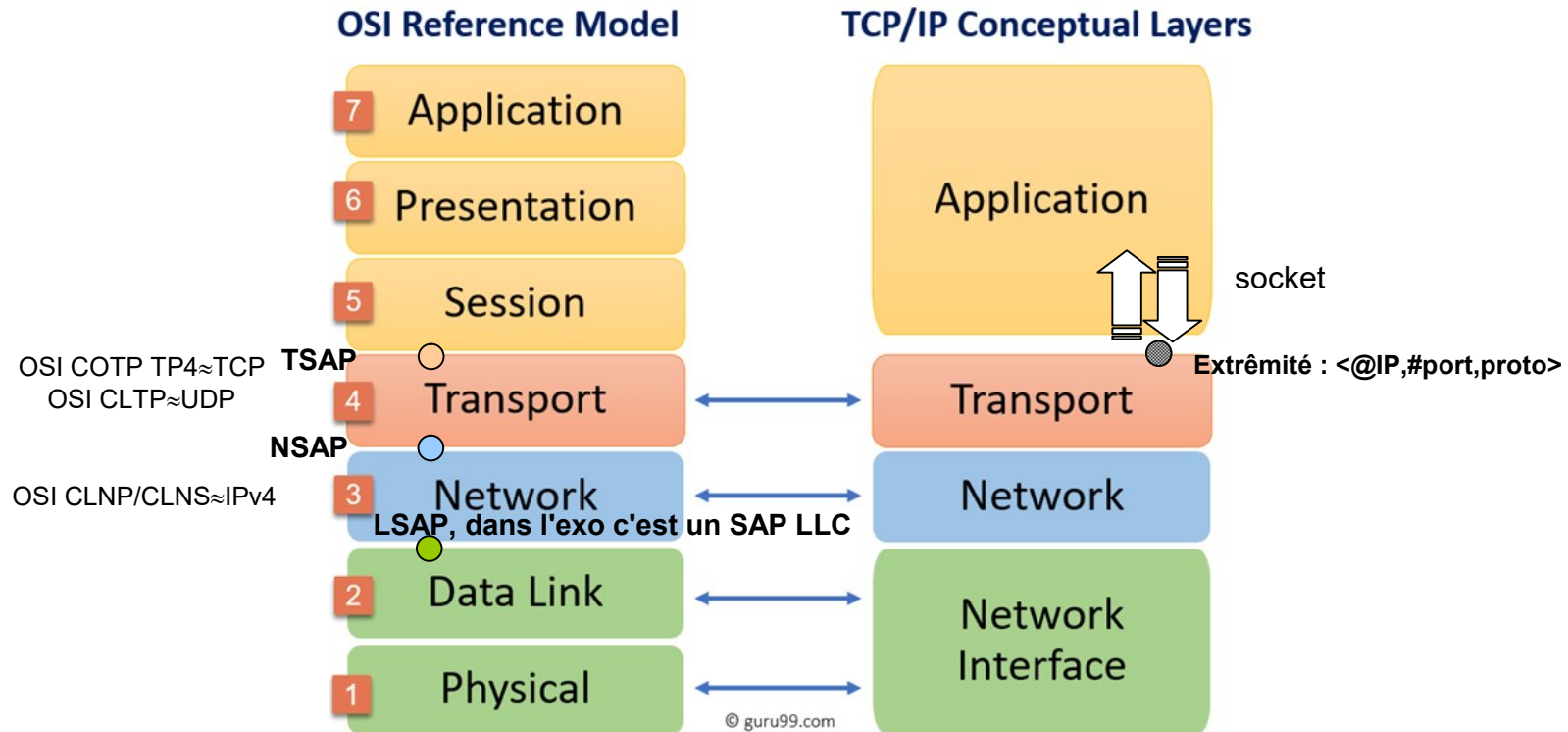
Dans la pile Internet, le seul équivalent que vous connaissez à travers le cours est l'extrémité de connexion, endpoint, au dessus de transport. Attention, l'extrémité de connexion n'est pas la socket, la socket est juste un moyen offert au programmeur pour accéder à la couche TCP ou UDP, donc à l'extrémité de connexion. Rappelez-vous la socket est juste une boîte aux lettres "plantée" sur l'extrémité de connexion.

Dans l'univers Unix, plutôt Unix System V, ce sont les objets "head of streams" qui se rapprochent le plus des TSAP de l'ISO. D'ailleurs, les appels systèmes de la bibliothèque de programmation des streams (flux réseaux) de System V est assez proche de la sémantique ISO. Pas étonnant, ATT qui fut le promoteur de Unix System V est un téléphoniste !

⁸ Attention, quand on utilise LLC et SNAP, on réduit la charge utile maximale, ici, on enlève 8 octets, ce qui fait un MTU de 1492 octets.

⁹ SNAP est référencé dans la RFC1042. Très honnêtement, je n'ai jamais eu à prendre en compte SNAP dans ma propre carrière.

Le schéma ci-après essaie de donner un aperçu des SAP du modèle ISO par rapport aux "endpoints" de la pile Internet. On en a profité pour faire figurer sur la pile ISO les protocoles équivalents à ceux de l'Internet.

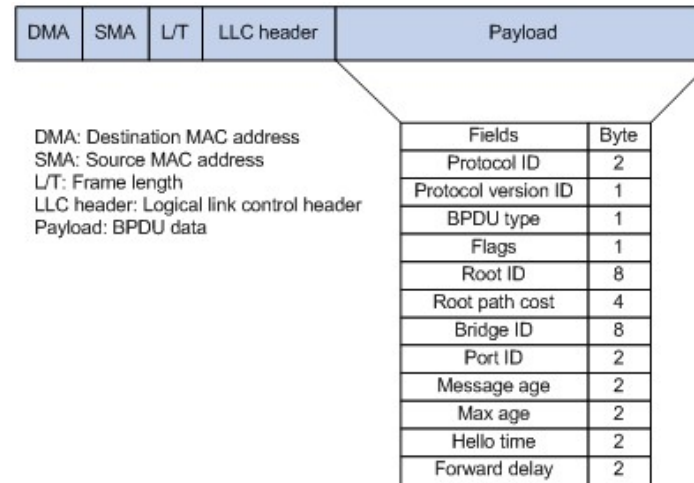


Dessin issu de <https://www.guru99.com/difference-tcp-ip-vs-osi-model.html>, accédé le 12/01/2020, et complété pour le polycopié d'exercices

Les standards correspondants dans la partie modèle ISO sont :

- OSI CLNP/CLNS : ConnectionLess Network Protocol/Services, ISO8473/ ITU-T X233
- OSI CLTP : ConnectionLess Transport Protocol, ISO8602/ITU-T X234
- OSI COTP TP4 : Connection Oriented Transport Protocol Transport Class 4, ISO8073/ITU-T X.224, utilisé avec CNLP/CNLS

Le protocole STP utilise deux types de BPDU. Des BPDU de configuration, et des BPDU de notification de changement de topologie (TCN, Topology Change Notification). C'est le premier type qui nous intéresse.



"The payload of a configuration BPDU includes the following fields:

- **Protocol ID**—Fixed at **0x0000**, which represents IEEE 802.1d.
- **Protocol version ID**—Spanning tree protocol version ID. The protocol version ID for STP is **0x00**.
- **BPDU type**—Type of the BPDU. The value is **0x00** for a configuration BPDU.
- **Flags**—An 8-bit field indicates the purpose of the BPDU. The lowest bit is the Topology Change (TC) flag. The highest bit is the Topology Change Acknowledge (TCA) flag. All other bits are reserved.
- **Root ID**—Root bridge ID formed by the priority and MAC address of the root bridge.
- **Root path cost**—Cost of the path to the root bridge.
- **Bridge ID**—Designated bridge ID formed by the priority and MAC address of the designated bridge.
- **Port ID**—Designated port ID formed by the priority and global port number of the designated port.
- **Message age**—Age of the configuration BPDU while it propagates in the network¹⁰.
- **Max age**—Maximum age of the configuration BPDU stored on the switch.
- **Hello time**—Configuration BPDU transmission interval.
- **Forward delay**—Delay for STP bridges to transit port state.

Devices use the root bridge ID, root path cost, designated bridge ID, designated port ID, message age, max age, hello time, and forward delay for spanning tree calculation."

Source : [http://www.h3c.com/en/Support/Resource_Center/Technical_Documents/Home/Routers/00-Public/Configure/Configuration_Guides/H3C_810_2600_3600_CG\(V7\)-R0707-6W301/04/201904/1166190_294551_0.htm](http://www.h3c.com/en/Support/Resource_Center/Technical_Documents/Home/Routers/00-Public/Configure/Configuration_Guides/H3C_810_2600_3600_CG(V7)-R0707-6W301/04/201904/1166190_294551_0.htm) consultée le 18/12/2019 à 23h45

¹⁰ Compté en secondes



Question 1

On effectue la capture d'une BPDU donnée page suivante, on vous demande d'analyser la trame et d'en extraire des éléments de mise en œuvre du protocole IEEE 802.1D.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	Cisco_87:85:04	Spanning-tree-(for-...	STP	60	Conf. Root = 32768/100/00:1c:0e:87:78:00 Cost = 4 P...
▼ IEEE 802.3 Ethernet						
▼ Destination: Spanning-tree-(for-bridges)_00 (01:80:c2:00:00:00)						
Address: Spanning-tree-(for-bridges)_00 (01:80:c2:00:00:00)						
.... ..0. = LG bit: Globally unique address (factory default)						
.... ..1. = IG bit: Group address (multicast/broadcast)						
▼ Source: Cisco_87:85:04 (00:1c:0e:87:85:04)						
Address: Cisco_87:85:04 (00:1c:0e:87:85:04)						
.... ..0. = LG bit: Globally unique address (factory default)						
.... ..0. = IG bit: Individual address (unicast)						
Length: 38						
Padding: 0000000000000000						
▼ Logical-Link Control						
▼ DSAP: Spanning Tree BPDU (0x42)						
0100 001. = SAP: Spanning Tree BPDU						
.... ..0 = IG Bit: Individual						
> SSAP: Spanning Tree BPDU (0x42)						
▼ Control field: U, func=UI (0x03)						
000. 00.. = Command: Unnumbered Information (0x00)						
.... ..11 = Frame type: Unnumbered frame (0x3)						
▼ Spanning Tree Protocol						
Protocol Identifier: Spanning Tree Protocol (0x0000)						
Protocol Version Identifier: Spanning Tree (0)						
BPDU Type: Configuration (0x00)						
> BPDU flags: 0x00						
▼ Root Identifier: 32768 / 100 / 00:1c:0e:87:78:00						
Root Bridge Priority: 32768						
Root Bridge System ID Extension: 100						
Root Bridge System ID: Cisco_87:78:00 (00:1c:0e:87:78:00)						
Root Path Cost: 4						
▼ Bridge Identifier: 32768 / 100 / 00:1c:0e:87:85:00						
Bridge Priority: 32768						
Bridge System ID Extension: 100						
Bridge System ID: Cisco_87:85:00 (00:1c:0e:87:85:00)						
Port identifier: 0x8004						
Message Age: 1						
Max Age: 20						
Hello Time: 2						

Attention l'affichage de Wireshark adopte la notation IETF, le bit G est à la droite du premier octet de tête de l'adresse MAC

Forward Delay: 15

0000	01 80 c2 00 00 00 00 1c 0e 87 85 04 00 26 42 42	&BB
0010	03 00 00 00 00 00 80 64 00 1c 0e 87 78 00 00 00	d...x...
0020	00 04 80 64 00 1c 0e 87 85 00 80 04 01 00 14 00	...d.....

1.1. Est-ce que l'adresse de destination de ce BPDU est une adresse de diffusion ? Quelle est cette adresse, est-ce un broadcast ou un multicast ? Pourquoi ?

Correction :

Le bit IG de l'adresse de destination est à 1, donc c'est une adresse de diffusion MAC (Broadcast/Multicast). Le bit LG, Universally Administered or Locally Administered bit¹¹, est à 0, c'est donc une adresse répertoriée, ou encore officielle. Un broadcast MAC est avec tous les bits à 1, donc c'est une adresse MAC multicast.

Cette adresse est 01:80:c2:00:00:00. On se doute que cette adresse a un rôle particulier dans l'exécution du STP. C'est en effet, ce qu'indique wikipedia : https://en.wikipedia.org/wiki/Multicast_address, consulté le 27/12/2019 ainsi que le livre de K. R. Fall et R. Stevens TCP/IP Illustrated V1, second Edition, p105 :

01-80-C2-00-00-00	Spanning Tree Protocol (for bridges) IEEE 802.1D
-------------------	--

Et c'est pour cela qu'elle est "Universally Administered".

A propos de la représentation différente des bits dans un octet entre l'IEEE et l'IETF, la RFC 1042 écrit :

"Appendix on Numbers

The **IEEE** likes to specify numbers in bit transmission order, or **bit-wise little-endian order**. The **Internet protocols** are documented in **byte-wise big-endian order**. This may cause some confusion about the proper values to use for numbers. Here are the conversions for some numbers of interest.

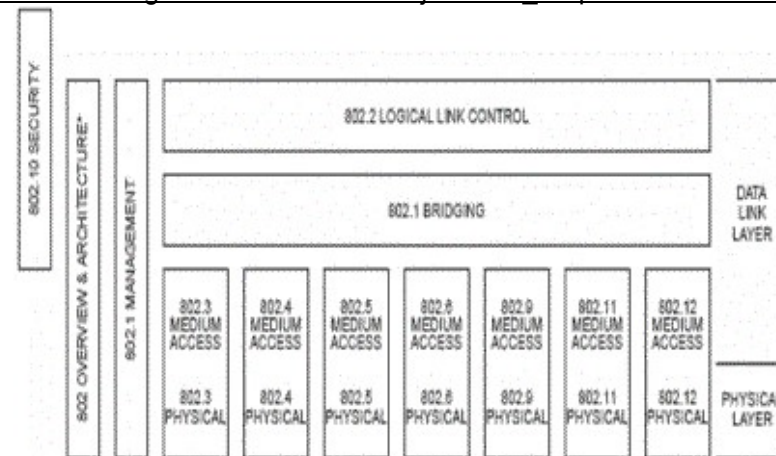
Number	IEEE HEX	IEEE Binary	Internet Binary	Internet Decimal
UI Op Code	C0	11000000	00000011	3"

Compléments sur l'architecture IEEE802 (LAN) : L'organisation des couches protocolaires pour les réseaux locaux définie par l'IEEE est assez riche, et il est parfois difficile de donner une information exacte sachant que les normes IEEE sont payantes. Et que je ne les ai pas... j'améliorerais dès que possible.

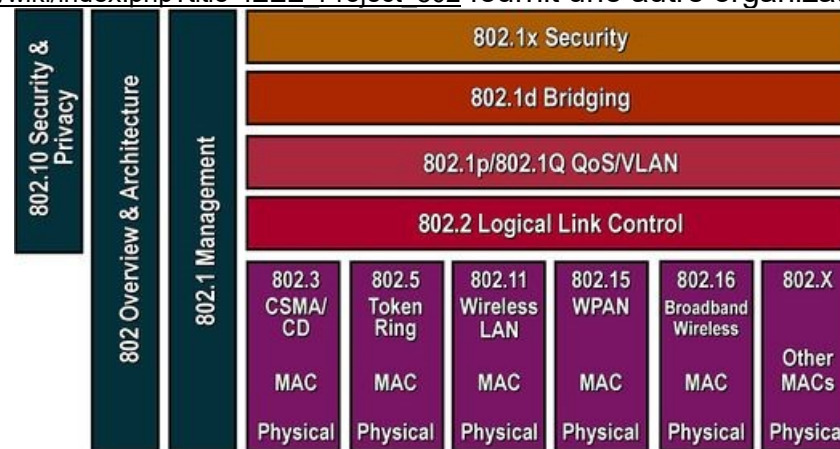
¹¹ The Universally Administered or Locally Administered address bit is used to tell if the MAC address is the burned-in-address (BIA) or a MAC address that has been changed locally. Maybe this bit served a purpose in the past but with modern applications on Windows or *nix systems you can change the MAC address to almost anything and this bit does not have to be set to tell the system that you are no longer using the manufacturer's BIA. If the bit is set to 0 then the MAC address is recognized as a BIA MAC address. When the bit is set to 1 then the MAC address is recognized as being changed from the BIA to a unique MAC address that is locally setup. <https://packetsdropped.wordpress.com/2011/01/13/mac-address-universally-or-locally-administered-bit-and-individualgroup-bit/> consulté le 27/12/2019.



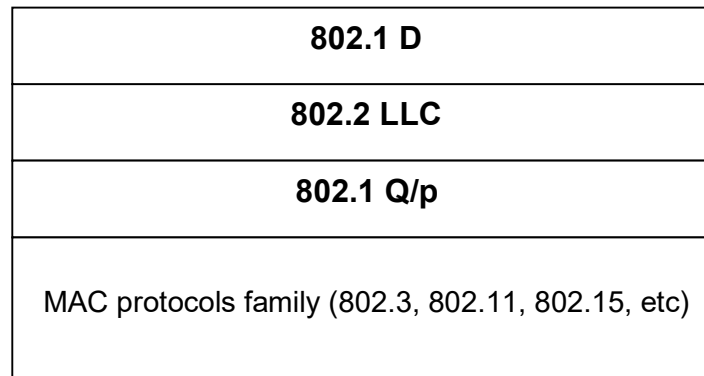
Un dessin indicatif est donné par http://www.oas.org/en/citel/infocitel/2006/junio/wifi_i.asp consulté le 12/01/2020.



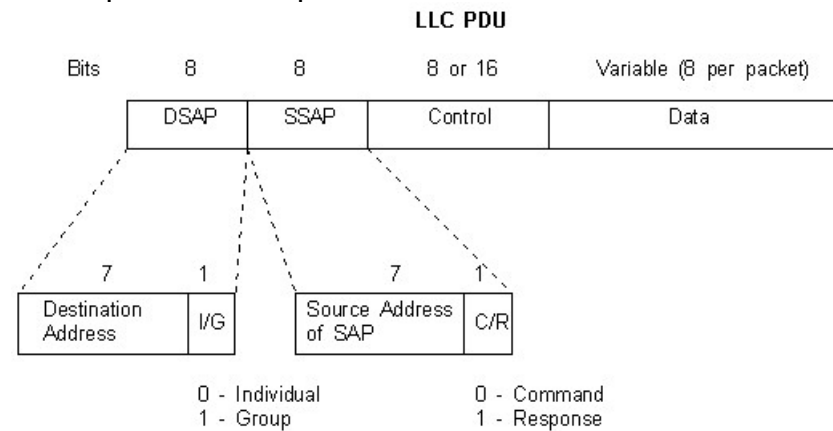
Une autre source http://173.236.11.102/wiki/index.php?title=IEEE_Project_802 fournit une autre organization des couches :



Pour la partie MAC, tout le monde est d'accord. Pour la demi-couche supérieure, on a des points de vue contradictoires. On peut tirer quelques conclusions d'après ce qu'on a observé dans les exercices : 802.1p (priorité&QoS)/802.1Q (VLAN) sont probablement en dessous de LLC puisque le VLAN encapsule des unités de données de protocole LLC. Par contre, 802.1D et toutes les normes associées à la commutation/pontage/bridging sont au-dessus de LLC 802.2 puisqu'on a vu que les BPDU étaient dans la charge utile de PDU LLC. On laissera de côté la partie sécurité. Pour résumer, je vous proposerais bien l'empilement suivant, à confirmer :



Ci-dessous une description de la partie LLC puis du champ Control.



Source : <http://www.rhyshaden.com/hdlc.htm>, consultée le 19/01/2020

TYPE	N° des bits de contrôle.															
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
I	0	N°S							P/F	N°R						
S	1	0	S S		X X X X				P/F	N°R						
U	1	1	M M		P/F	M M M										

Les formats des PDU

Source : http://infoindustrielle.free.fr/Les_reseaux/Reseaux_hm/17N-LLC.htm consultée le 19/01/2020

Pour ceux qui connaissent le protocole HDLC (High-level Data Link Control), ils reconnaîtront quelques similitudes. En effet, LLC et HDLC sont construits dans un esprit proche, par essence ce sont des protocoles point à point. Pour une explication détaillée de chacun des champs, et des différentes commandes du protocole HDLC, consulter https://fr.wikipedia.org/wiki/High-Level_Data_Link_Control, accédé le 27/12/2019 à 18h20.

1.2. Comment s'interprète le champ U d'après les informations données dans la partie commentée de la trace Wireshark ?

Correction :

L'extraction de la trace WireShark pour le champ contrôle donne :

```

▼ Logical-Link Control
  ▼ DSAP: Spanning Tree BPDU (0x42)
    0100 001. = SAP: Spanning Tree BPDU
    .... 0 = IG Bit: Individual
  ▼ SSAP: Spanning Tree BPDU (0x42)
    0100 001. = SAP: Spanning Tree BPDU
    .... 0 = CR Bit: Command
  ▼ Control field: U, func=UI (0x03)

```

Le champ DSAP, Destination Service Access Point, indique l'adresse destination (une autre couche LLC distante) de la trame LLC (encapsulée dans une trame IEEE802.3, ne l'oublions pas).

C'est un DSAP de type unicast. Le SSAP, Source Service Access Point, est identique. Wikipedia, https://en.wikipedia.org/wiki/IEEE_802.2 consulté le 27/12/2019 à 19h20, indique pour cette valeur qui correspond à l'agent qui gère le protocole Spanning Tree :

66 (dec)	42(hex)	IEEE 802.1 Bridge Spanning Tree Protocol ^[3]
----------	---------	---

On est bien dans un échange protocolaire lié au maintien de l'arbre couvrant. La suite de la trame nous en dira plus.

Le champ Control est le 3^{ème} octet de l'entête LLC. Il ne dit pas grand chose à part que ce qu'il y a dans la charge utile de la trame LLC est une information qui n'a pas de numéro de séquence (Unnumbered Information), on peut dire qu'elle est autonome par rapport à d'autres trames LLC pouvant être reçue après ou avant. En particulier cette trame LLC n'est pas soumise à une séquence. C'est le programme qui gère la partie applicative qui va interpréter la nature de la charge utile. Ici, c'est le programme qui gère le protocole STP.

Question 2

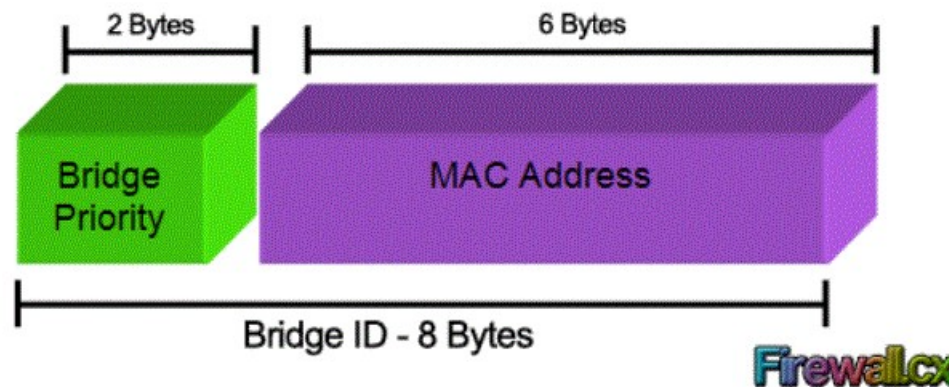
Le BID (Bridge ID) est d'un format différent de celui vu lors de UTC505. A votre avis pourquoi ? Quelle information supplémentaire contiendrait-il ? Pourquoi la priorité est supérieure à la priorité maximale donnée dans l'exercice précédent ?

Pour vous aider, on vous donne un extrait de texte de la littérature Internet (<http://www.firewall.cx/networking-topics/protocols/spanning-tree-protocol/1054-spanning-tree-protocol-root-bridge-election.html>, 29/12/2019), donc à examiner avec précaution.

UNDERSTANDING BRIDGE ID, BRIDGE PRIORITY & SYSTEM ID EXTENSION

In our [earlier article](#) we discussed about the [Spanning Tree Protocol, Rapid STP port costs and port states](#). Before STP decides which path is the best to the **Root Bridge**, it needs to first decide which switch has to be elected as the **Root Bridge**, which is where the **Bridge ID** comes into play. Readers interested can also read our [STP Principles, Redundant Network Links & Broadcast Storms](#) article.

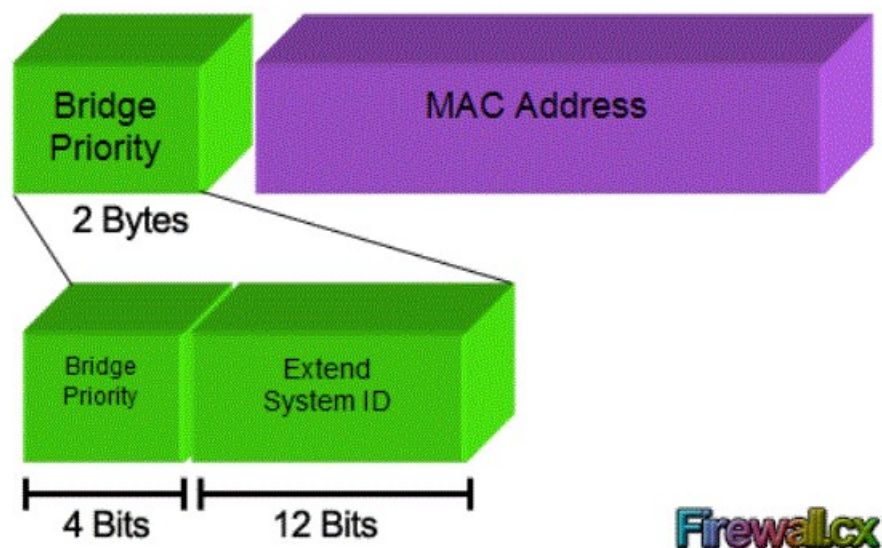
Every switch has an identity when they are part of a network. This identity is called the **Bridge ID** or **BID**. It is an **8 byte field** which is divided into two parts. The first part is a **2-byte Bridge Priority** field (which can be configured) while the second part is the **6-byte MAC address** of the switch. While the **Bridge Priority** is configurable, the **MAC address** is unique amongst all switches and the sum of these two ensures a unique **Bridge ID**.



The above **Bridge ID** assumes there is one Spanning Tree instance for the entire network. This is also called **Common Spanning-Tree (CST)**.

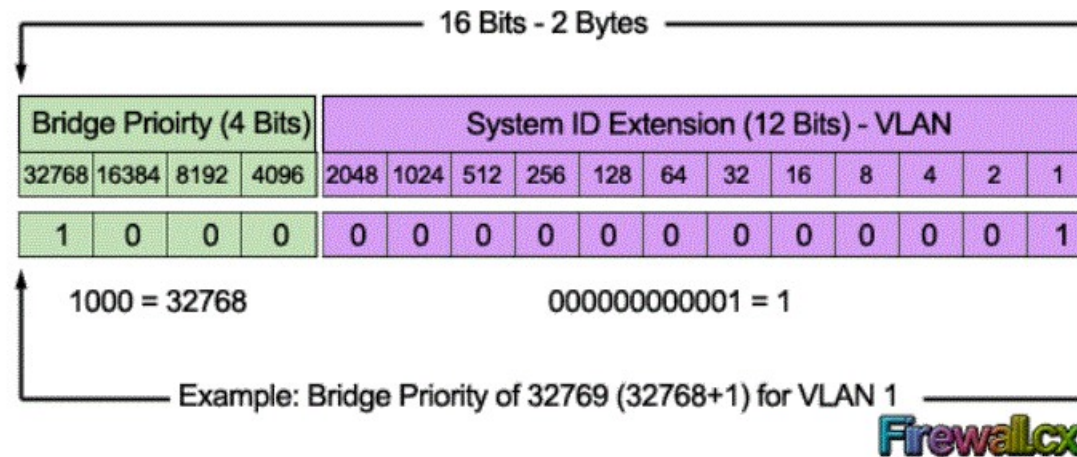
As networks began to grow and become more complex, VLANs were introduced, allowing the creation of multiple logical and physical networks. It was then necessary to run multiple instances of STP in order to accommodate each network - VLAN. These multiple instances are called **Multiple Spanning Tree (MST)**, **Per-VLAN Spanning Tree (PVST)** and **Per-VLAN Spanning Tree Plus (PVST+)**.

In order to accommodate the additional VLAN information, the **Extended System ID** field was introduced, borrowing **12 bits** from the original **Bridge Priority**:



Firewallcx

The **Bridge Priority** value and the **Extended System ID** extension together make up a 16 bit (2-byte) value. The **Bridge Priority** making up the left most bits, is a value of 0 to 61440. The **Extended System ID** is a value of 1 to 4095 corresponding to the respective VLAN participating in STP. The **Bridge Priority** increments in blocks of 4096 to allow the **System ID Extension** to squeeze in between each increment. This is clearly shown in the below analysis:



We should note that the **Bridge Priority Field** can only be set in increments of 4096. This means that possible values are: 4096, 8192, 12288, 16384, 20480, 24576, 28672, 32768 etc. By default, Cisco's **Per-VLAN Spanning-Tree Plus** (PVST+) adds this **System ID Extension** (sys-id-ext) to the **Bridge Priority**.

i The two values (**Bridge Priority + System ID Extension**) together make up the **Bridge ID** used to elect the **Root Bridge**.

Correction :

On a du introduire le numéro de VLAN. C'est un nouveau champ par rapport à ce qui avait été donné dans l'exercice sur l'arbre couvrant dans UTC505 et qui correspondait à la version 1998 de 802.1D. Du coup, la priorité ne peut plus être construite de la même façon. C'est ce que révèle le texte issu de l'Internet ci-dessus.

Ci-après le champ Root Identifier extrait de la trace Wireshark. Sa représentation est différente de la description ci-dessus, c'est pour cela que la valeur hexadécimale est donnée aussi.

```
> Root Identifier: 32768 / 100 / 00:1c:0e:
Root Path Cost: 4
> Bridge Identifier: 32768 / 100 / 00:1c:0
Port identifier: 0x8004
Message Age: 1
Max Age: 20
Hello Time: 2
Forward Delay: 15
```

Root ID, 8 octets—Root bridge ID formed by the priority and MAC address of the root bridge.

Root ID : 32768/100/00:1c:0e:87:78:00 et sa version hexadécimale 80 64 00 1c 0e 87 78 00

Ce qui donne pour les 2 premiers octets traduits en binaire : 1000 000001100100, on retrouve bien la priorité d'après le texte ci-dessus : 1000 = 32768, et 000001100100 nous donnerait le numéro de VLAN soit une fois utilisée la table de calcul suivante

System ID Extension (12 bits) - VLAN											
2048	1024	512	256	128	64	32	16	8	4	2	1

On obtient : 64+32+4 soit 100 en décimal. La suite c'est l'adresse MAC du commutateur.

Question 3

On vous demande d'interpréter la partie spécifique au protocole STP de la BPDU capturée. Qu'est-ce que cette BPDU raconte-t-elle ?

Correction :

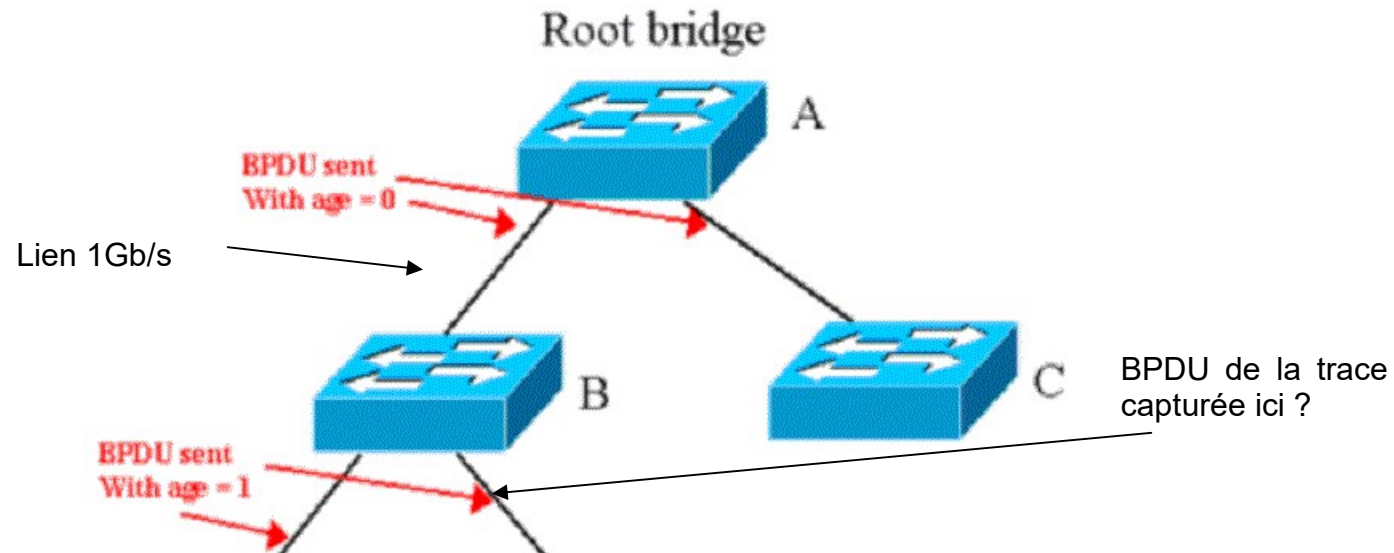
Si on reprend les champs un par un tels qu'ils ont été présentés ci-dessus on a :

- **Protocol ID, 2 octets**—0x0000, c'est le protocole STP, IEEE 802.1d.
- **Protocol version ID, 1 octet**—Spanning tree protocol version ID qui est 0x00, comme attendu.
- **BPDU type, 1 octet**—0x00 pour une BPDU de configuration BPDU.
- **Flags, 1 octet**— 0x00 qui indique l'objet de la BPDU. Le dernier bit correspond à Topology Change (TC) il vaut 0, donc pas de changement. Le premier bit correspond à Topology Change Acknowledge (TCA) il vaut 0. Les autres bits sont réservés.



- **Root ID, 8 octets**—Root bridge ID, soit 32768/100/00:1c:0e:87:78:00 et sa version hexadécimale 80 64 00 1c 0e 87 78 00 cf question précédente
- **Root path cost, 4 octets**—4. Si on reprend la table donnée en rappel au début de l'exercice, en faisant l'hypothèse qu'il n'y a eu qu'un seul lien de parcouru, le lien devrait avoir un débit de 1Gb/s. Mais dans la table donnée question 4, c'est un coût qui est correspond à la plage spécifiée pour 1Tb/s
- **Bridge ID, 8 octets**— On reprend la logique appliquée pour le Root ID, on s'aperçoit que ce commutateur, qui est l'émetteur de la trame, a la même priorité que la racine, et qu'il appartient au même VLAN, ce qui est attendu quand même.
- **Port ID, 2 octets**—port ID : priorité (1^{er} octet) et numéro de port dans le commutateur (2^{ème} octet) : 0x8004 , c'est le 4^{ème} port du switch il a la priorité 128. Le tout se note 128.4. C'est le port par lequel le BPDU a été émis. C'est un port désigné. C'est cohérent puisque c'est le commutateur émetteur de la trame qui détient ce port (adresse Ethernet dans le Bridge ID) 00:1c:0e:87:85:00.
- **Message age, 2 octets**—ici 1. Age de la BPDU de configuration quand elle transite dans le réseau. Chaque commutateur fait +1 sur Message age quand il est traversé. Il est initialisé à 0 sur le commutateur racine. Avec la valeur de Max age, il fait Max age – Message age, et cela donne la durée pendant laquelle il conserve les informations qui y sont contenues. Max age – Message age correspond à une durée de péremption des informations de la BPDU de configuration.
- **Max age, 2 octets**—ici 20 secondes, contribue au calcul de temps de péremption d'une BPDU de configuration, cf champ ci-dessus
- **Hello time, 2 octets**—Paramètre de configuration, périodicité des envois de BPDU, ici 2s.
- **Forward delay, 2 octets**—Pour un port, temps de transition d'un état de calcul d'une topologie SPN à un état où les trames sont forwardées par le port et une topologie d'arbre couvrant a été déterminée.

D'après les informations ci-dessus, on peut imaginer que le commutateur qui a envoyé cette BPDU est dans une position équivalente de celle de B dans le dessin ci-dessous :



source : <https://www.cisco.com/c/en/us/support/docs/lan-switching/spanning-tree-protocol/19120-122.html#tune>, consultée le 30/12/2019

Question 4

Si le coût du chemin à la racine pour un commutateur, soit le champ `Rootpathcost`, est d'une longueur de 4 octets, quelle est la valeur maximale possible pour ce champ ?

Correction :

Le coût maximum possible est $2^{32}-1$ puis qu'on a 4 octets soit 32 bits disponibles. De plus, on compte en nombres entiers non signés. On a donc : $4\ 294\ 967\ 296-1=4\ 294\ 967\ 295$ comme plus grande valeur. Toutefois, comme on additionne des coûts de liens, l'usage propose de limiter le coût d'un chemin entre 0 et 200 000 000 et d'établir les valeurs des coûts de lien en correspondance.

Est-ce que ça suffit ? Le diamètre maximum d'un arbre couvrant est de 7 commutateurs entre les deux machines les plus éloignées du LAN. Si on regarde le tableau en rappel au début de ce corrigé, on a probablement assez en considérant la première colonne. Moins immédiat avec la deuxième colonne, mais on peut supposer que les liens de plus bas débit déployés sont au minimum à 100 Mb/s aujourd'hui.

En fait, quand on considère un champ `Rootpathcost` de 32 bits, il faut revenir à la norme 802.1D de 2004 qui donne le tableau suivant (encore différent de ceux fournis précédemment, c'est vrai on s'y perd à la fin):

In 2004, the revised **802.1D** had its 16-bit path cost increased to a 32-bit value, providing more granularity:

Link Speed	Recommended value	Recommended range	Range
<=100 Kb/s	200 000 000 ^a	20 000 000–200 000 000	1–200 000 000
1 Mb/s	20 000 000 ^a	2 000 000–200 000 000	1–200 000 000
10 Mb/s	2 000 000 ^a	200 000–20 000 000	1–200 000 000
100 Mb/s	200 000 ^a	20 000–2 000 000	1–200 000 000
1 Gb/s	20 000	2 000–200 000	1–200 000 000
10 Gb/s	2 000	200–20 000	1–200 000 000
100 Gb/s	200	20–2 000	1–200 000 000
1 Tb/s	20	2–200	1–200 000 000
10 Tb/s	2	1–20	1–200 000 000

Zone de valeurs en usage aujourd'hui (début 2020)

^aBridges conformant to IEEE Std 802.1D, 1998 Edition, i.e., that support only 16-bit values for Path Cost, should use 65 535 as the Path Cost for these link speeds when used in conjunction with Bridges that support 32-bit Path Cost values.

source : <http://www.firewall.cx/networking-topics/protocols/spanning-tree-protocol/1045-spanning-tree-protocol-port-costs-states.html>, consultée le 30/12/2019.

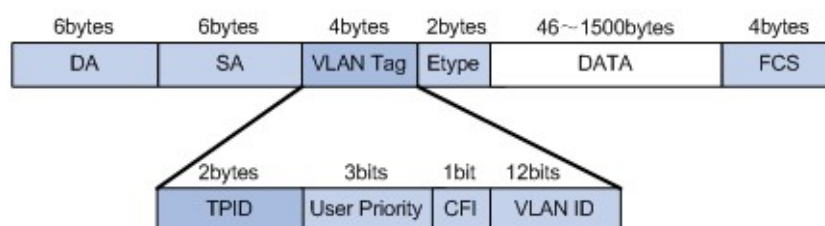
On peut, parfois mélanger STP, 802.1D, et RSTP, pour Rapid STP, ou 802.1D-2004 (via 802.1w dont CISCO est à l'origine¹²). On a peu évoqué la norme 802.1s, "The *Multiple Spanning Tree Protocol* (MSTP), originally defined in IEEE 802.1s and later merged into IEEE 802.1Q-2005, defines an extension to RSTP to further develop the usefulness of virtual LANs (VLANs)."

Enfin, la norme la plus actuelle sur le routage/commutation dans les réseaux de commutateurs est **IEEE 802.1aq, Shortest Path Bridging (SPB)** qui adresse en particulier des réseaux locaux haut débit dans les data centers dont TRILL vu plus loin est un concurrent.

Exercice 7 : VLAN vs VXLAN dans un contexte multi data centers reliés par Internet

On s'intéresse à la façon de faire des réseaux isolés les uns des autres sur un réseau local de type Ethernet. La norme qui gère cet aspect correspond à IEEE802.1Q. On l'a vu elle met en oeuvre le concept de VLAN. Le format d'une trame IEEE802.1Q est le suivant :

¹² <https://www.cisco.com/c/en/us/support/docs/lan-switching/spanning-tree-protocol/24062-146.html> (06/12/2020) peut être consulté pour une explication de cette norme par CISCO. La page semble assez claire.



Par rapport à la trame Ethernet normale, on a inséré 4 octets entre l'adresse source et le champ type. L'identificateur de VLAN qui permet de distinguer les VLAN entre eux, est sur 12 bits.

Question 1

Combien d'identificateurs de VLAN sont disponibles sachant que 0x000, 0xFFFF, et 0x001 sont réservés.

Correction :

On dispose de 12 bits pour l'identificateur de VLAN. Donc 2^{12} possibilités, soit 4096, mais 3 identificateurs sont réservés, il reste 4093 identificateurs disponibles.

Question 2

On se pose la question d'utiliser cette technologie pour isoler des réseaux locaux des clients les uns des autres dans un data center de grande envergure (plusieurs milliers d'entreprises, "tenants", y seraient hébergées). Donner brièvement quels seraient les atouts et les limites de cette solution selon vous ?

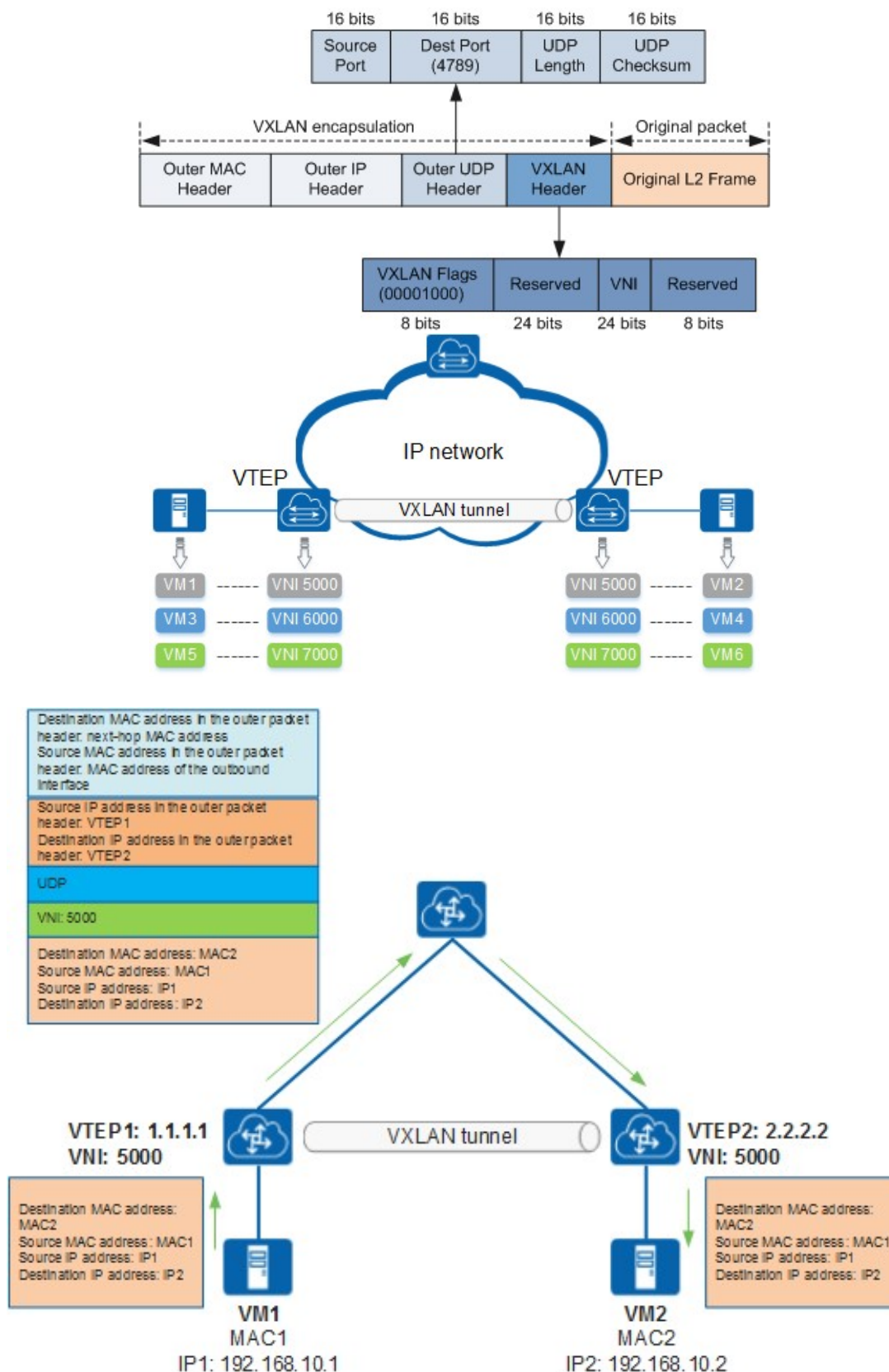
Quel(s) problème(s) voyez-vous à utiliser cette approche pour matérialiser des VLAN IEEE802.1Q entre data center reliés par des liens longue distance à travers Internet ?

Correction :

- Les atouts : une technologie éprouvée et maîtrisée
- Les limites : 4K VLAN c'est peu pour un data center qui peut héberger plusieurs centaines d'entreprises qui elles mêmes peuvent avoir besoin de plusieurs dizaines de VLAN. On a donc essentiellement un problème de passage à l'échelle avec cette technologie.
- Le problème essentiel c'est que le VLAN ne s'étend pas naturellement à travers l'Internet. Il faut passer par un routeur nécessairement, et si on veut rendre l'interconnexion invisible, il faut installer des équipements qui vont offrir un mécanisme de tunnel entre les routeurs pour que les commutateurs de VLAN ne s'aperçoivent de rien.

Une solution alternative existe. Elle se fonde sur VXLAN (Virtual eXtensible Local Area Network). On donne le format d'un message VXLAN. C'est un message échangé entre deux data centers (centres de données) à travers l'Internet suivant une approche dite "réseau overlay" qui se traduirait par en français par "réseau en superposition" tel que montré ci-dessous.

Les schémas ci-après sont issus de <https://support.huawei.com/enterprise/en/doc/EDOC1100023542?section=j016&topicName=vxlan> (consulté le 14/12/2020).



"VXLAN Forwarding process:

1. VM1 sends a packet destined for VM2.
2. After receiving the packet, VTEP1 performs VXLAN encapsulation. The IP address of VTEP2 is the destination IP address in the outer IP header added to the packet. VTEP1 transmits the encapsulated packet to VTEP2 through the IP network based on the outer MAC address and IP address of the packet.
3. VTEP2 decapsulates the received packet, obtains the original packet sent by VM1, and forwards the packet to VM2."

Pour ceux qui seraient curieux sur le fonctionnement de VXLAN dans un contexte de virtualisation pour le Cloud regarder : Introduction to Cloud Overlay Networks - VXLAN, David Mahler https://www.youtube.com/watch?v=Jqm_4TMmQz8 (consultée le 14/12/2020).

Question 3

Combien d'encapsulations successives comptez vous ? Indiquez les consécutivement, du contenu le plus intérieur vers l'encapsulation la plus extérieure.

Est-ce que le modèle ISO, ou le modèle Internet, et leurs principes de couches s'appuyant l'une sur l'autre pour résoudre une fonction de communication bien délimitée, vus en cours sont strictement respectés ? Pourquoi ?

Correction :

D'après ce qu'on peut voir sur le schéma :

- On a une trame Ethernet dans un message VXLAN, 1ère encapsulation
- On a un message VXLAN dans un datagramme UDP, 2ème encapsulation
- On a un datagramme UDP dans un datagramme IP, 3ème encapsulation
- On a un datagramme IP dans une trame Ethernet, 4ème encapsulation

Soit 4 encapsulations successives, sachant qu'on ne compte généralement pas la transformation de la trame finale en suite de bits qui seront transmis sur le medium de communication.

Cette énumération ne compte pas les encapsulations successives des données dans la trame Ethernet la plus interne.

Comme il n'est pas donné beaucoup d'information, on pourrait aussi penser à l'intérieur du message VXLAN qu'il y a une encapsulation Ethernet dans une trame IEEE802.1Q/p.

Le modèle ISO n'est pas strictement respecté puisque on a affaire à un tunnel et qu'il encapsule une trame applicative dans un datagramme UDP pour traverser l'Internet. On a donc de la couche 2 dans de la couche 4.

Maintenant, si on morcelle notre vision en un monde Internet public, et un monde entreprise. Chaque monde pris isolément respecte le modèle ISO en couche. Pour l'Internet à partir de l'entrée du tunnel, la trame ne représente que des données. Pour les datacenters, ils ne voient pas le tunnel excepté l'équipement dédié.

Question 4

L'identificateur VXLAN est sur 3 octets. Combien de réseaux logiques sont possibles maintenant ?

Correction :

Avec 3 octets, on 2^{24} (16 777 216) possibilités pour attribuer des identificateurs de réseaux logiques. Il faut retrancher les identificateurs réservés qui ne sont pas donnés dans la question.

Question 5

Ci-après une trace Wireshark d'un échange VXLAN :

No.	Time	Source	Destination	Protocol	Length	Info
22	35.532341	10.210.33.11	10.210.33.12	ICMP	148	Echo (ping) request id=0x3f0d, seq=1/256, ttl=64 (no response found!)
23	36.531395	10.210.33.11	10.210.33.12	ICMP	148	Echo (ping) request id=0x3f0d, seq=2/512, ttl=64 (no response found!)
24	37.530377	10.210.33.11	10.210.33.12	ICMP	148	Echo (ping) request id=0x3f0d, seq=3/768, ttl=64 (no response found!)
Frame 24: 148 bytes on wire (1184 bits), 148 bytes captured (1184 bits) on interface 0						
Ethernet II, Src: Vmware_64:b2:26 (00:50:56:64:b2:26), Dst: Vmware_6e:4c:54 (00:50:56:6e:4c:54)						
Internet Protocol Version 4, Src: 10.3.0.70, Dst: 10.3.0.71						
User Datagram Protocol, Src Port: 50343, Dst Port: 4789						
Virtual eXtensible Local Area Network						
Flags: 0x0800, VXLAN Network ID (VNI)						
0... .. = GBP Extension: Not defined						
.... ..0... .. = Don't Learn: False						
.... 1... .. = VXLAN Network ID (VNI): True						
.... ..0... .. = Policy Applied: False						
.000 .000 0.00 .000 = Reserved(R): 0x0000						
Group Policy ID: 0						
VXLAN Network Identifier (VNI): 5029						
Reserved: 0						
Ethernet II, Src: Vmware_82:45:cd (00:50:56:82:45:cd), Dst: Vmware_82:1d:46 (00:50:56:82:1d:46)						
Internet Protocol Version 4, Src: 10.210.33.11, Dst: 10.210.33.12						
Internet Control Message Protocol						

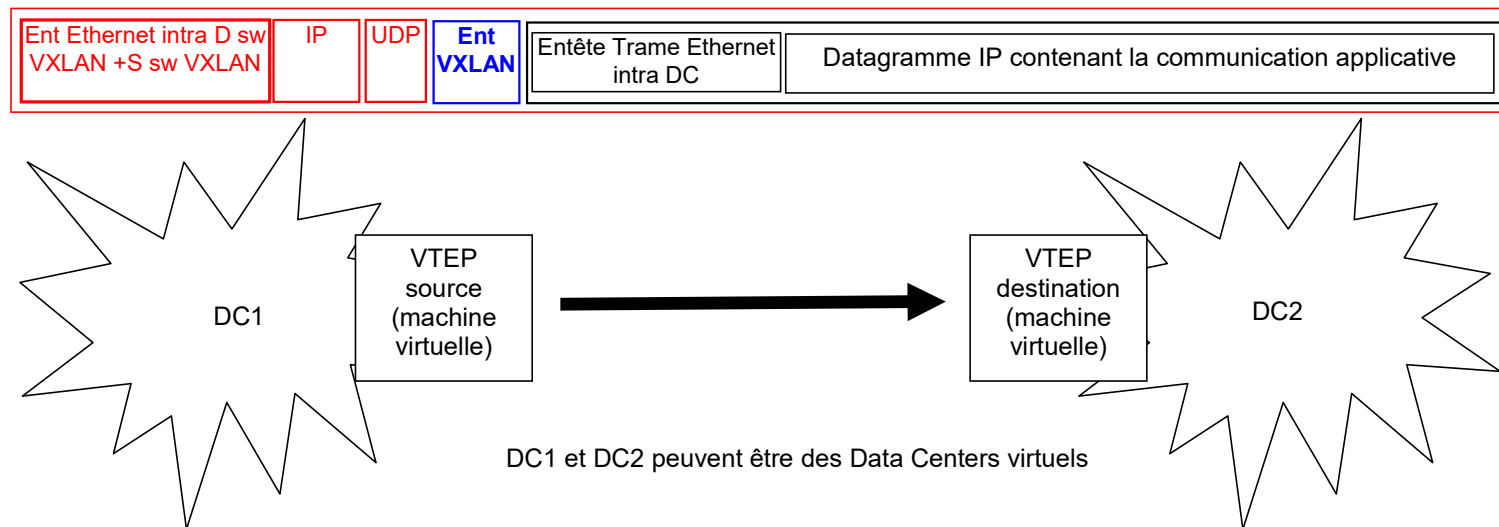
0000	00 50 56 6e 4c 54 00 50	56 64 b2 26 08 00 45 48	.PVnLT.P Vd.&...EH
0010	00 86 00 00 40 00 40 11	25 8d 0a 03 00 46 0a 03	...@.@. %...F..
0020	00 47 c4 a7 12 b5 00 72	00 00 08 00 00 00 00 13	.G.....r
0030	a5 00 00 50 56 82 1d 46	00 50 56 82 45 cd 08 00	...PV..F .PV.E...
0040	45 48 00 54 00 00 40 00	40 01 e2 a6 0a d2 21 0b	EH.T...@. @.....!.
0050	0a d2 21 0c 08 00 1a 44	3f 0d 00 03 e2 47 5a 58	..!....D ?....GZX
0060	00 00 00 00 a0 38 03 00	00 00 00 00 10 11 12 13	8..
0070	14 15 16 17 18 19 1a 1b	1c 1d 1e 1f 20 21 22 23	 !"#
0080	24 25 26 27 28 29 2a 2b	2c 2d 2e 2f 30 31 32 33	\$%&'()*+ ,-. /0123
0090	34 35 36 37		4567

On s'intéresse à la trame 24.

- Quelle est l'adresse Ethernet source de l'émetteur¹³ de la trame, contenant une PDU (Protocol Data Unit) avec une charge VXLAN ?
- Quelle est l'adresse Ethernet du destinataire¹⁴ VTEP ?
- Quelle est l'adresse IP du VTEP source ?
- Quelle est l'adresse IP du VTEP destination de la trame 24 ?
- Quel est le port UDP associé au VTEP destination ?
- Quel est l'identificateur VXLAN (VNI) contenu dans la trame 24 ?
- Quelle est l'adresse IP de la source et de la destination de la trame interne (inner frame) ?

Correction :

Avec la trame dans la trace ci-dessus, il faut bien avoir en tête les encapsulations successives et l'interconnexion entre les DC à chaque extrémité du tunnel :



VTEP : VXLAN Tunnel Entry Point

¹³ C'est une carte NIC, d'une entrée de tunnel VXLAN. (VTEP source)

¹⁴ C'est une carte NIC d'une sortie de tunnel VXLAN. (VTEP destination)

```

Frame 24: 148 bytes on wire (1184 bits), 148 bytes captured (1184 bits)
Ethernet II, Src: Vmware_64:b2:26 (00:50:56:64:b2:26), Dst: Vmware_6e:4c:54 (00:50:56:6e:4c:54)
Internet Protocol Version 4, Src: 10.3.0.70, Dst: 10.3.0.71
User Datagram Protocol, Src Port: 50343, Dst Port: 4789
Virtual eXtensible Local Area Network
Ethernet II, Src: Vmware_82:45:cd (00:50:56:82:45:cd), Dst: Vmware_82:1d:46 (00:50:56:82:1d:46)
Internet Protocol Version 4, Src: 10.210.33.11, Dst: 10.210.33.12
Internet Control Message Protocol

```

000	00 50 56 6e 4c 54	00 50 56 64 b2 26	08 00 45 48	.PVnLT.P Vd.&..EH
010	00 86 00 00 40 00	40 11 25 8d	0a 03 00 46 0a 03	@.@. %...F..
020	00 47 c4 a7 12 b5	00 72 00 00 08 00	00 00 00 00 00 13	.G.....r
030	a5 00			...PV..F .PV.E...
040	45 48 00 54 00 00	40 00 40 01 e2 a6	0a d2 21 0b	EH.T..@. @.....!
050	0a d2 21 0c	08 00 1a 44 3f 0d	00 03 e2 47 5a 58	..!....D ?....GZX
060	00 00 00 00 a0 38	03 00 00 00 00 00	10 11 12 13	8..
070	14 15 16 17 18 19	1a 1b 1c 1d 1e 1f	20 21 22 23	 !"#
080	24 25 26 27 28 29	2a 2b 2c 2d 2e 2f	30 31 32 33	\$%&'()*+ ,-./0123
090	34 35 36 37			4567

- Quelle est l'adresse Ethernet source de l'émetteur de la trame, contenant une PDU (Protocol Data Unit) avec une charge VXLAN ? Attention, c'est le deuxième champ de la trame Ethernet la plus externe (outer frame): 00:50:56:64:b2:26. On remarque au passage que le constructeur étant VMware c'est une carte virtuelle, au sens machine virtuelle d'exécution.
- Quelle est l'adresse Ethernet du destinataire VTEP ? Attention, c'est le premier champ de la trame Ethernet la plus externe (outer frame) : 00:50:56:6e:4c:54. Même remarque. D'ailleurs toutes les adresses Ethernet de la trace correspondent à des cartes virtuelles.
- Quelle est l'adresse IP du VTEP source (outer IP) ? Il faut trouver la fin de l'entête IP du datagramme, et on revient en arrière de 4 octets pour trouver la fin de l'adresse IP: 0a030046 en hexadécimal soit 10.3.0.70.
- Quelle est l'adresse IP du VTEP destination de la trame 24 (outer IP) ? Il faut trouver la fin de l'entête IP du datagramme pour trouver la fin de l'adresse IP: 0a030047 en hexadécimal soit 10.3.0.71.

- Quel est le port UDP associé au VTEP destination ? On cherche dans l'entête IP le port destination. On l'obtient après les 2 premiers octets : `12b5`, pour convertir : $1 \cdot 16^3 + 2 \cdot 16^2 + b \cdot 16^1 + 5 \cdot 16^0$ soit : $4096 + 2 \cdot 256 + 11 \cdot 16 + 5 = 4789$. C'est bien le port associé à un serveur VXLAN (<https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml?search=4789>, consulté le 15/12/2020) depuis le 19/4/2013.
- Quel est l'identificateur VXLAN (VNI) contenu dans la trame 24 ? L'entête VXLAN vient juste après l'entête UDP. L'entête VXLAN fait 8 octets, et dans ces 8 octets, on a 3 octets d'identification qui correspondent au 3ème champ dans l'entête VXLAN, juste avant l'octet réservé. On y trouve `0013a5` soit 5029.
- Quelle est l'adresse IP de la source et de la destination de la trame interne (inner frame) ? `0ad2210b`, 10.210.33.11 et de de l'host destination ? `0ad2210c`, 10.210.33.12

Pour trouver l'entête IP (inner IP), on a sauté l'entête de la trame Ethernet intérieure à la charge utile UDP. Cette partie est noircie dans le schéma ci-dessus. Il faut alors compter les 20 octets de l'entête IP et depuis la fin remonter vers les adresses IP destination puis source.

Remarque : On peut faire plus malin en lisant directement les informations dans la trace Wireshark expliquée dans la fenêtre 2. Ce qui peut être une bonne stratégie pour un examen.

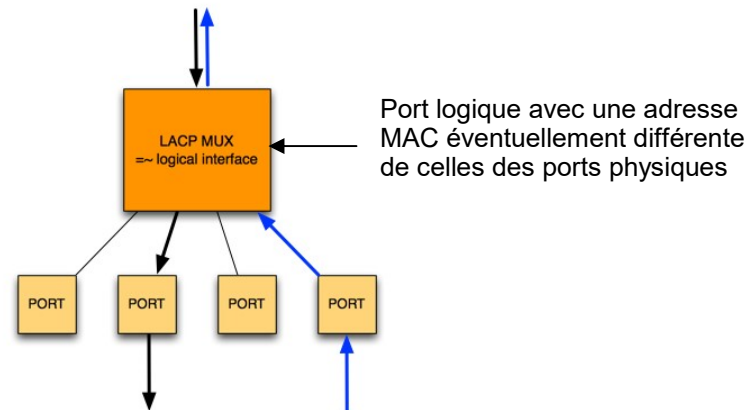
Wireshark arrive à reconnaître les champs de la trame avec ses parties VXLAN et les différentes encapsulations car il doit avoir dans son catalogue d'entêtes le format d'une trame VXLAN. Quand celle-ci arrive, il est alors capable de décomposer les champs en fonction des identificateurs qu'il trouve au fur et à mesure qu'il désencapsule les différents messages.

Exercice 8 : Link Agregation Control Protocol (LACP) ou IEEE802.1AX-2014.

L'objectif de ce protocole est de gérer l'agrégation de plusieurs liens de mêmes caractéristiques (type et débit) entre commutateurs ou entre commutateur et carte NIC qui supporte ce protocole. Cette agrégation peut servir à l'augmentation des débits entre 2 points LAN, pour de l'équilibrage de charge (load balancing), ou de la tolérance aux fautes en particulier. L'agrégation de ces liens physiques forme un lien logique.

L'approche de l'exercice, c'est d'effectuer une observation une situation, à travers des échanges capturés en Wireshark, et d'en déduire des comportements protocolaires. Souvent c'est une bonne façon d'aborder des technologies qu'on ne connaît pas.

A la fin de l'agrégation, on peut voir l'architecture comme dans le dessin ci-après :



source : <http://movingpackets.net/2017/10/17/decoding-lACP-port-state/> (16/12/2020)

Pour que deux entités reliées par un même lien se coordonnent via LACP, on utilise l'adresse MAC multicast 01-80-C2-00-00-02 et le type Ethernet 0x8809 associée à une famille de protocoles très particulière qu'on dénomme "Slow Protocols" dans les normes IEEE liées aux réseaux locaux, dont LACP fait partie. Ce Slow Protocol est identifié par le sous-type 0x01 juste après le champ type Ethernet dans la trame.

Le but de l'exercice est de tenter de découvrir la façon dont procède le protocole à partir de l'observation des échanges. La trace ci-après est issue du lien <https://packetlife.net/captures/protocol/lACP/> (consulté le 14/12/2020).

	Time	Source	Destin	Protocol	Lengt	Info
1	0.000000	Cisco_12:0f...	Slo...	LACP	124	v1 ACTOR 00:13:c4:12:0f:00 P: 22 K: 13 E****G*A PARTNER 00:0e:83:16:f5:00 P: 25 K: 13 **DC*GS*
2	0.917445	Cisco_12:0f...	Slo...	LACP	124	v1 ACTOR 00:13:c4:12:0f:00 P: 22 K: 13 E****G*A PARTNER 00:0e:83:16:f5:00 P: 25 K: 13 **DC*GS*
3	1.880655	Cisco_12:0f...	Slo...	LACP	124	v1 ACTOR 00:13:c4:12:0f:00 P: 22 K: 13 E****G*A PARTNER 00:0e:83:16:f5:00 P: 25 K: 13 **DC*GS*
4	8.408838	Cisco_12:0f...	Slo...	LACP	124	v1 ACTOR 00:13:c4:12:0f:00 P: 22 K: 13 *F***SG*A PARTNER 00:00:00:00:00:00 P: 0 K: 0 *****
5	8.423106	Cisco_12:0f...	Slo...	LACP	124	v1 ACTOR 00:13:c4:12:0f:00 P: 22 K: 13 *FDCSG*A PARTNER 00:00:00:00:00:00 P: 0 K: 0 *****
6	28.949125	Cisco_12:0f...	Slo...	LACP	124	v1 ACTOR 00:13:c4:12:0f:00 P: 22 K: 13 *FDCSG*A PARTNER 00:00:00:00:00:00 P: 0 K: 0 *****
7	55.465881	Cisco_12:0f...	Slo...	LACP	124	v1 ACTOR 00:13:c4:12:0f:00 P: 22 K: 13 *FDCSG*A PARTNER 00:00:00:00:00:00 P: 0 K: 0 *****
8	81.122096	Cisco_12:0f...	Slo...	LACP	124	v1 ACTOR 00:13:c4:12:0f:00 P: 22 K: 13 *FDCSG*A PARTNER 00:00:00:00:00:00 P: 0 K: 0 *****
9	84.962105	Cisco_16:f5...	Slo...	LACP	124	v1 ACTOR 00:0e:83:16:f5:00 P: 25 K: 13 *****SG** PARTNER 00:13:c4:12:0f:00 P: 22 K: 13 *FDC*G*A
10	84.977641	Cisco_12:0f...	Slo...	LACP	124	v1 ACTOR 00:13:c4:12:0f:00 P: 22 K: 13 *F***SG*A PARTNER 00:00:00:00:00:00 P: 0 K: 0 *****
11	84.977659	Cisco_12:0f...	Slo...	LACP	124	v1 ACTOR 00:13:c4:12:0f:00 P: 22 K: 13 *F***G*A PARTNER 00:00:00:00:00:00 P: 0 K: 0 *****
12	84.981772	Cisco_16:f5...	Slo...	LACP	124	v1 ACTOR 00:0e:83:16:f5:00 P: 25 K: 13 *****G** PARTNER 00:13:c4:12:0f:00 P: 22 K: 13 *FDC*G*A
13	84.981788	Cisco_16:f5...	Slo...	LACP	124	v1 ACTOR 00:0e:83:16:f5:00 P: 25 K: 13 *****G** PARTNER 00:13:c4:12:0f:00 P: 22 K: 13 *FDC*G*A
14	85.850119	Cisco_16:f5...	Slo...	LACP	124	v1 ACTOR 00:0e:83:16:f5:00 P: 25 K: 13 *****G** PARTNER 00:13:c4:12:0f:00 P: 22 K: 13 *F***G*A
15	86.768259	Cisco_16:f5...	Slo...	LACP	124	v1 ACTOR 00:0e:83:16:f5:00 P: 25 K: 13 *****SG** PARTNER 00:13:c4:12:0f:00 P: 22 K: 13 *F***G*A
16	90.470637	Cisco_12:0f...	Slo...	LACP	124	v1 ACTOR 00:13:c4:12:0f:00 P: 22 K: 13 *****SG*A PARTNER 00:0e:83:16:f5:00 P: 25 K: 13 *****SG**
17	90.495693	Cisco_12:0f...	Slo...	LACP	124	v1 ACTOR 00:13:c4:12:0f:00 P: 22 K: 13 **DCSG*A PARTNER 00:0e:83:16:f5:00 P: 25 K: 13 *****SG**
18	90.497886	Cisco_16:f5...	Slo...	LACP	124	v1 ACTOR 00:0e:83:16:f5:00 P: 25 K: 13 **DCSG** PARTNER 00:13:c4:12:0f:00 P: 22 K: 13 *****SG*A
19	107.370848	Cisco_16:f5...	Slo...	LACP	124	v1 ACTOR 00:0e:83:16:f5:00 P: 25 K: 13 **DCSG** PARTNER 00:13:c4:12:0f:00 P: 22 K: 13 **DCSG*A
20	112.338735	Cisco_12:0f...	Slo...	LACP	124	v1 ACTOR 00:13:c4:12:0f:00 P: 22 K: 13 **DCSG*A PARTNER 00:0e:83:16:f5:00 P: 25 K: 13 **DCSG**

L'entête Ethernet et la partie réservée au Slow Protocol donnent bien ce qui est attendu :

```

Frame 1: 124 bytes on wire (992 bits), 124 bytes captured (992 bits)
Ethernet II, Src: Cisco_12:0f:0d (00:13:c4:12:0f:0d), Dst: Slow-Protocols (01:80:c2:00:00:02)
> Destination: Slow-Protocols (01:80:c2:00:00:02)
> Source: Cisco_12:0f:0d (00:13:c4:12:0f:0d)
Type: Slow Protocols (0x8809)
Slow Protocols
Slow Protocols subtype: LACP (0x01)
Link Aggregation Control Protocol

```

Dans la trace, on distingue un "ACTOR" et un "PARTNER", à quoi peuvent correspondrent ces concepts ?

Correction :

Dans l'agrégation de liens, "trunking", entre commutateurs ou entre commutateur et interface NIC, on peut supposer qu'il y a un maître (ACTOR) qui prend l'initiative de la construction de l'agrégation, et un assujetti (PARTNER). On peut faire l'hypothèse que c'est le plus



prioritaire qui prend le rôle d'ACTOR. Si la priorité est identique, on peut imaginer que c'est comme avec le spanning tree protocol, c'est l'adresse MAC la plus petit qui l'emporte. L'objet, c'est de constituer un groupe "LAG" (Link Aggregation Group) de ports. On dit aussi, suivant la littérature des équipementiers, pour un groupe : "virtual link" ou "bundle". Mais est-ce bien cela ?

Le lien source : <http://movingpackets.net/2017/10/17/decoding-lacp-port-state/> donne aussi la signification des bits associés à l'état d'un port d'agrégation : "The meaning of each bit is as follows":

Bit	Name	Meaning
0	LACP_Activity	Device intends to transmit periodically in order to find potential members for the aggregate ¹⁵ . This is toggled by mode active in the channel-group configuration on the member interfaces. 1 = Active, 0 = Passive.
1	LACP_Timeout ¹⁶	Length of the LACP timeout. 1 = Short Timeout, 0 = Long Timeout
2	Aggregation	Will allow the link to be aggregated. 1 = Yes, 0 = No (individual link)
3	Synchronization	Indicates that the mux on the transmitting machine is in sync with what's being advertised in the LACP frames. 1 = In sync, 0 = Not in sync ¹⁷
4	Collecting	Mux is accepting traffic received on this port 1 = Yes, 0 = No
5	Distributing	Mux is sending traffic using this port 1 = Yes, 0 = No
6	Defaulted	Whether the receiving mux is using default (administratively defined) parameters, if the information was received in an LACP PDU. 1 = default settings, 0 = via LACP PDU
7	Expired	Port in an expired state ¹⁸ 1 = Yes, 0 = No

Pour comprendre un peu plus, il faut creuser dans la norme, malheureusement, elle n'est pas gratuite, mais il y a des évolutions de cette norme qui sont des documents de travail, donc pas encore normatifs. Ce sont des informations partiellement instables mais qui aident. Le lien <https://1.ieee802.org/wp-content/uploads/2019/03/802-1AX-Rev-d1-0.pdf> (consulté le 17/12/2020) est instructif, d'ailleurs, il est

¹⁵ Irrespective to PARTNER state. Active mode = port sends LACP PDUs.

¹⁶ The long timeout mode-90 seconds (default value). With the long timeout, an LACP PDU is sent every 30 seconds. If no response comes from its partner after 3 LACPDUs are sent, a timeout event occurs, and the LACP state machine transition to the appropriate state based on its current state.
The short timeout mode-3 seconds. In the short timeout configuration, an LACP PDU is sent every second. If no response comes from its partner after 3 LACPDUs are sent, a timeout event occurs, and the LACP state machine transitions to the appropriate state based on its current state.

¹⁷ 0 means this link is not in the right LAG.

¹⁸ 1 mainly means the ACTOR LACP receiving machine is not able to receive.



postérieur à la date de la sortie de la norme IEEE802.1AX-2014. Il nous fournit deux schémas très éclairants. En fouillant dedans, on obtient d'autres informations sur le sens des bits décrits ci-dessus.

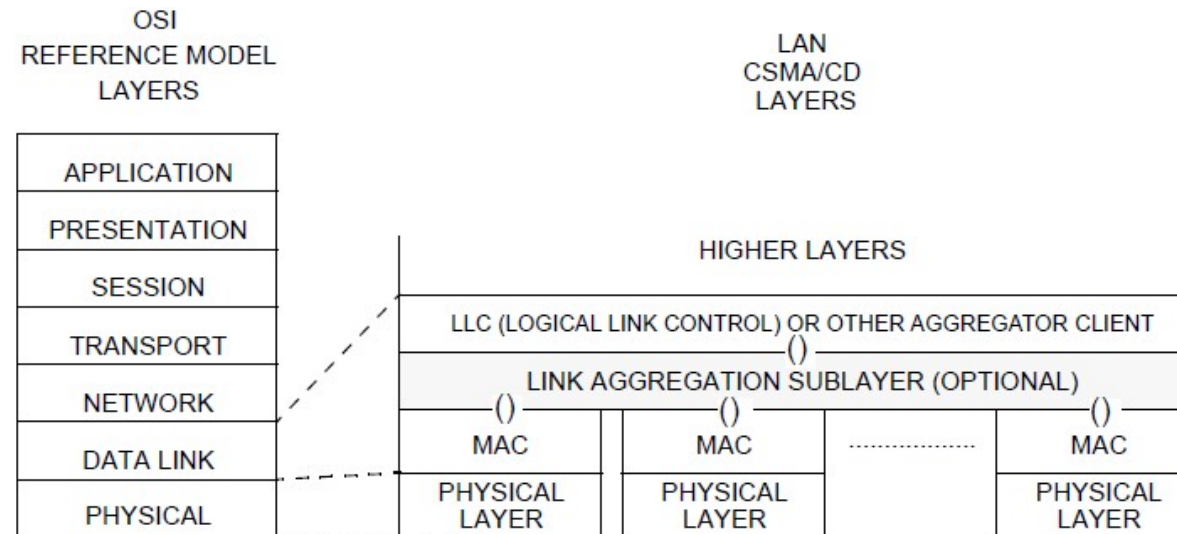


Figure 6-1—Architectural positioning of Link Aggregation sublayer

Figure 6-2 depicts the major blocks that form the Link Aggregation sublayer and their interrelationships.

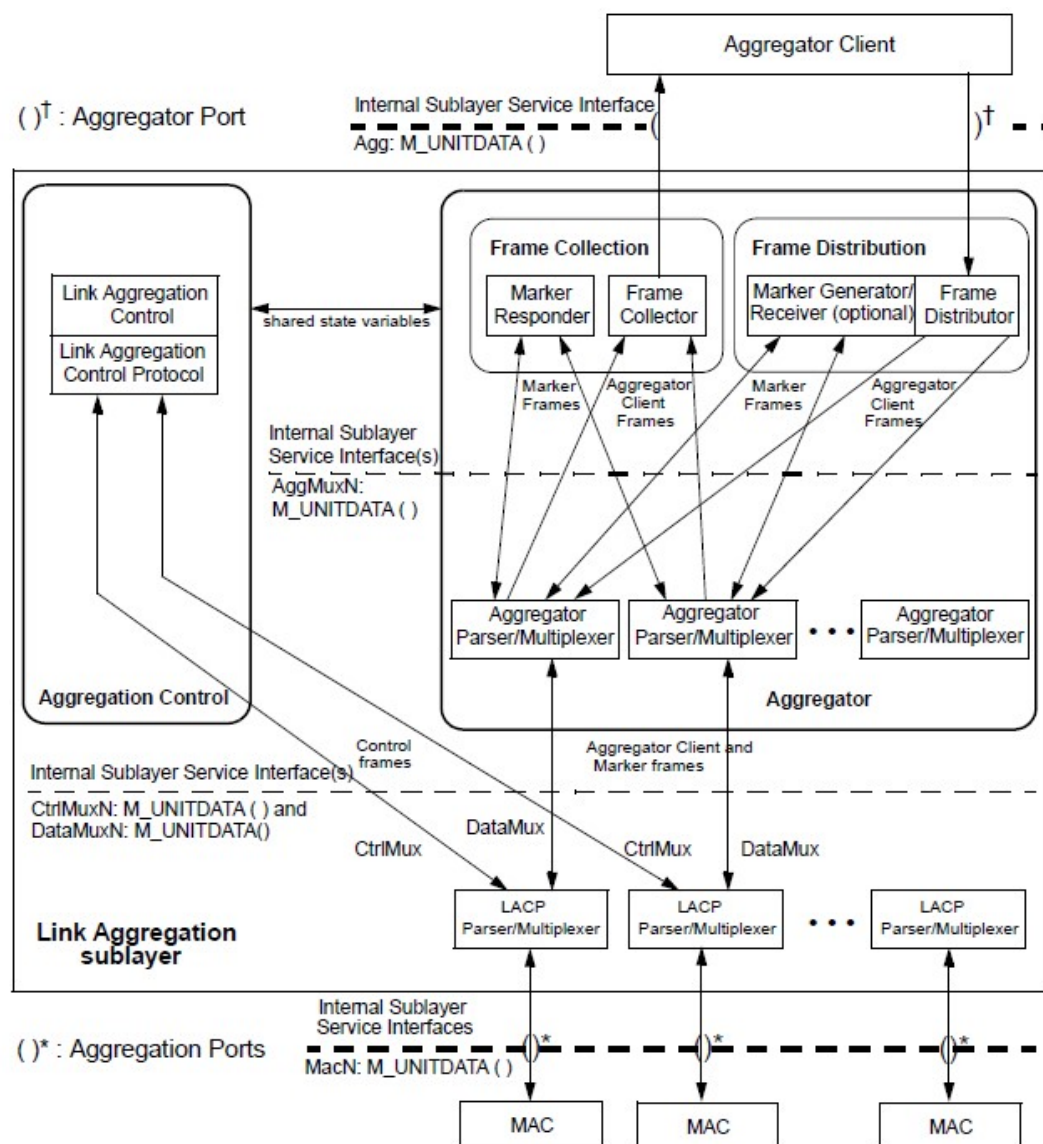
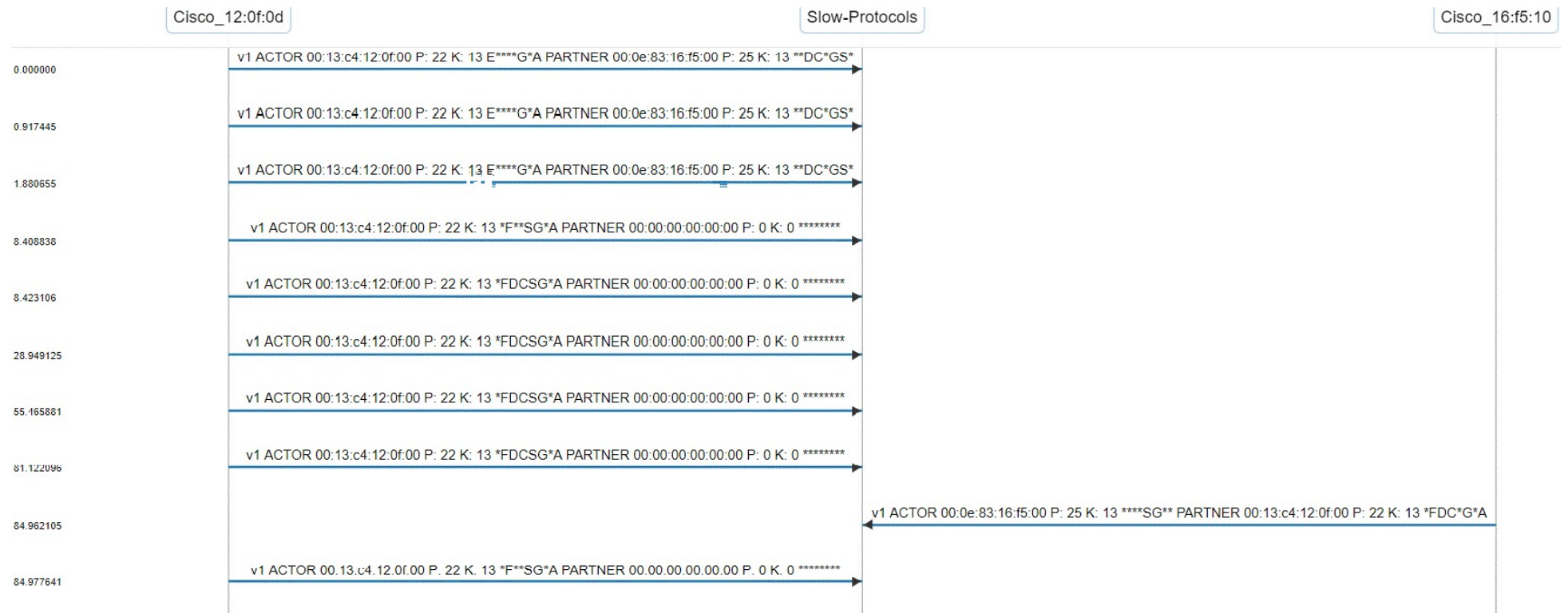


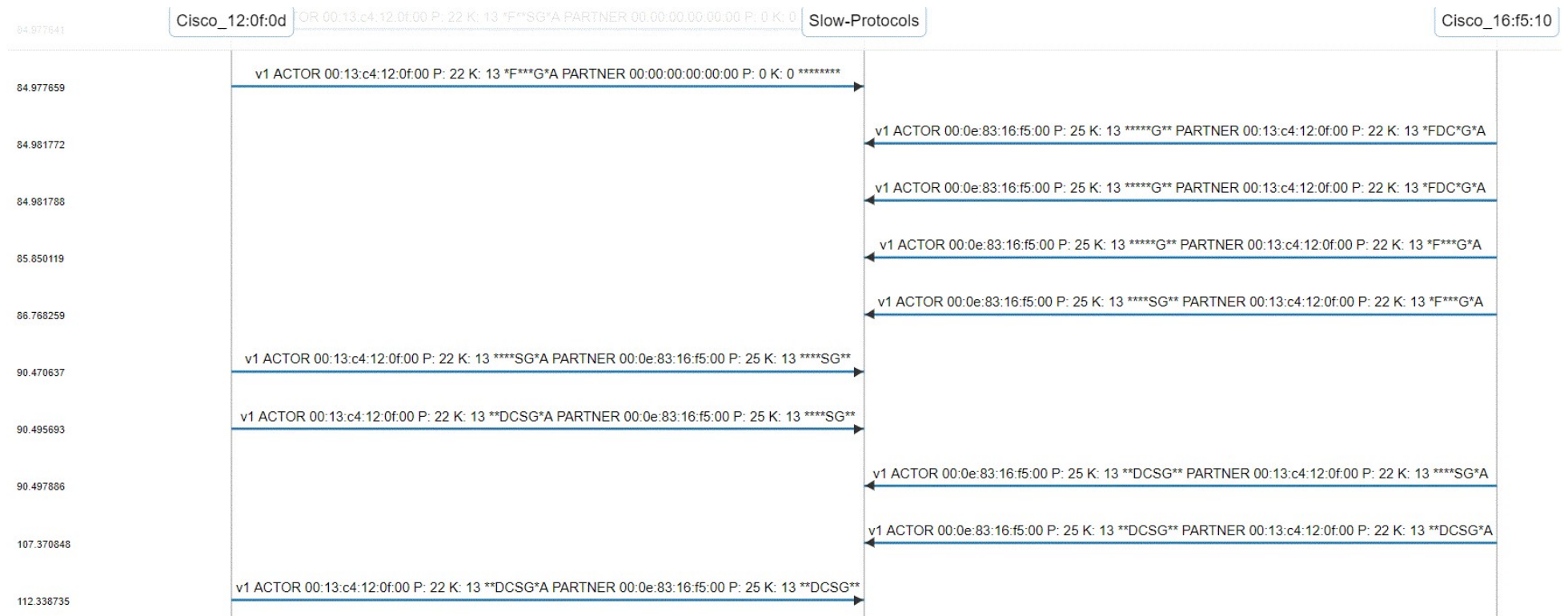
Figure 6-2—Link Aggregation sublayer block diagram



NOTE—IEEE Std 802.1AX-2008 and earlier editions of IEEE Std 802.3, Clause 43, used the term “MAC Client” instead of the term “Aggregator Client,” and placed Link Aggregation between the IEEE 802.3 MAC sublayer and the IEEE 802.3 MAC Control sublayer. This standard repositions Link Aggregation as a sublayer above the MAC sublayer. This rearrangement of the abstract layering diagram is intended to align the standard with common industry practice; it does not cause an implementation that is conformant to earlier editions of the standard to become non-interoperable with this edition.

Les ports MAC sont bidirectionnels, et ont tous le même débit. Ils sont tous liés aux mêmes VLAN. Et l'élément "Agregator" correspond au port logique qui instancie l'agrégation des ports physiques. "Frame Collector" est une fonction de réception des flux de trames de données, respectivement "Frame Distributor" est une fonction d'émission des flux de frames de données. On vous donne, ci-après, le graphique des flots de PDU LACP entre deux commutateurs adjacents qui sont en phase de négociation d'agrégation. On vous demande de déduire de ces échanges autant d'éléments de protocole que possible.





source : <https://www.cloudshark.org/analysis/9ecc8b3e9a86/ladder?order=&filter=eth.type%3D%3D0x8809&endpoints=&label=info> (16/12/2020)

Frame 1: 124 bytes on wire (992 bits), 124 bytes captured (992 bits) on interface unknown, id 0

Slow Protocols

Link Aggregation Control Protocol

```
LACP Version: 0x01
```

TLV Type: Actor Information (0x01)

Actor System Priority: 32768

Actor System ID: Cisco 12:0f:00 (00:13:c4:12:0f:00)

Actor Key: 13

Actor Port Priority: 32768

```
Actor Port: 22
Actor State: 0x85 LACP Activity Aggregation Expired
```

```
[Actor State Flags: [****G*A]
```

Reserved: 000000

TLV Type: Partner Information (0x02)

TLV Length: 0x14
Partno: System Priority: 32768

Partner System: Cisco 16:f5:00 (00:0e:83:16:f5:00)

Partner Key: 13

Partner Port Priority: 32768

```

Partner Port: 25
Partner State: 0x26 LACP Timeout Aggregation Collecting Distributing

```

```
[Partner State Flags: **PC*GS*]
```

```
Reserved: 000000
```

TLV Type: Collector Information (0x03)

TLV Length: 0x10
Collector Max Delay: 33760

Collector Max Delay: 32768
Reserved: 00000000000000000000000000000000

```
TLV Type: Terminator (0x00)
```

```
TLV Length: 0x00
```

[illegible]

Pour la trame 1 on donne le détail des champs ACTOR et PARTNER :

CC BY NC SA 68/72



Actor System Priority: 32768 Actor System ID: Cisco_12:0f:00 (00:13:c4:12:0f:00) Actor Key: 13 ← Key identique pour pouvoir agréger → Actor Port Priority: 32768 Actor Port: 22 Actor State: 0x85, LACP Activity, Aggregation, Expired1 = LACP Activity: Active0 = LACP Timeout: Long Timeout1 = Aggregation: Aggregatable0 = Synchronization: Out of Sync0 = Collecting: Disabled0 = Distributing: Disabled0 = Defaulted: No1 = Expired: Yes [Actor State Flags: E***G*A]	Partner System Priority: 32768 Partner System: Cisco_16:f5:00 (00:0e:83:16:f5:00) Partner Key: 13 Key identique = meme relation inter-switch Partner Port Priority: 32768 Partner Port: 25 Partner State: 0x36, LACP Timeout, Aggregation, Collecting, Distributing0 = LACP Activity: Passive1 = LACP Timeout: Short Timeout1 = Aggregation: Aggregatable0 = Synchronization: Out of Sync1 = Collecting: Enabled1 = Distributing: Enabled0 = Defaulted: No0 = Expired: No [Partner State Flags: **DC*GS*]
--	---

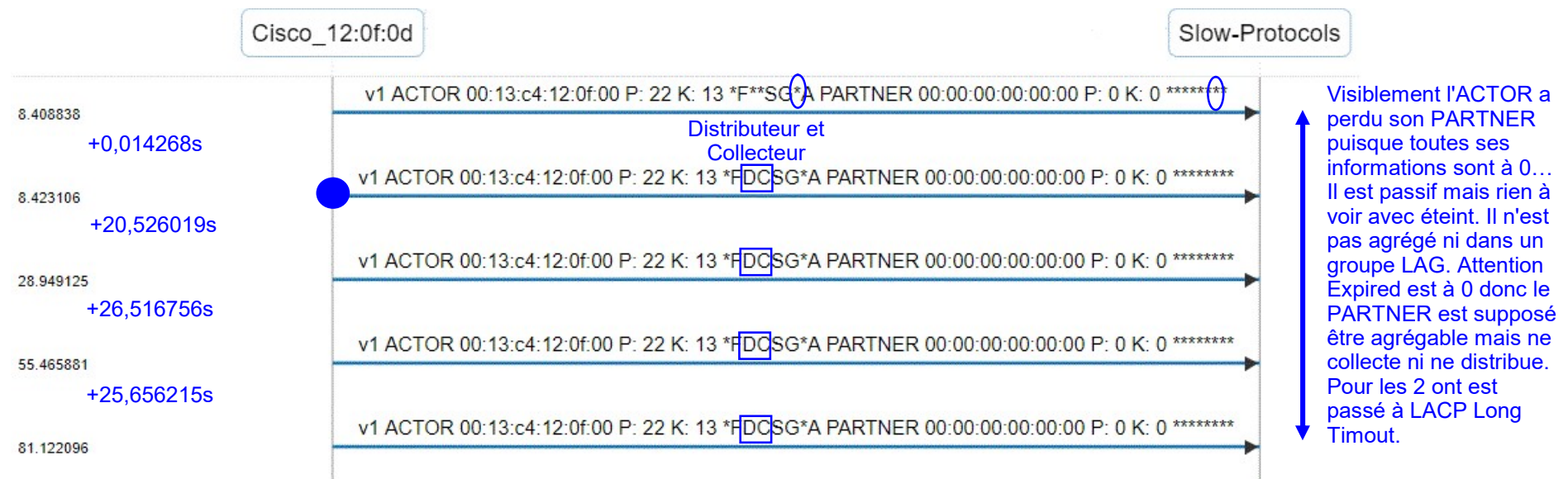
Correction :

De façon préliminaire, cette correction est une proposition car c'est une interprétation de ce qu'il se passe à travers la capture par Wireshark. Il n'y a pas de certitude qu'elle soit précise et totalement juste. Toute contribution est la bienvenue.

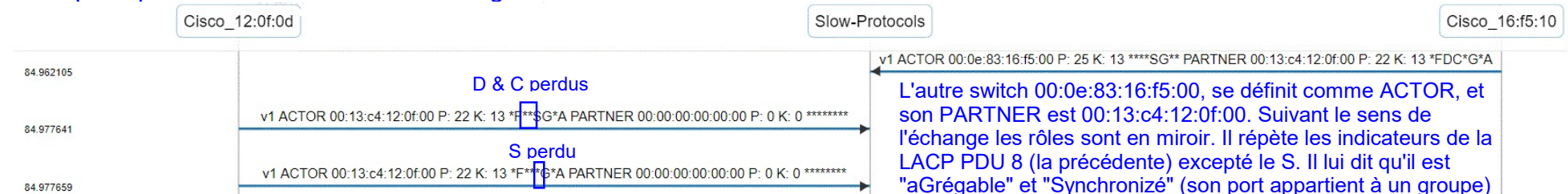


Pour l'ACTOR id(32768,00:13:c4:12:0f:00), via le port 22, d'adresse MAC 00:13:c4:12:0f:0d, on a "Expiré-aGrégable-Actif". L'acteur chercherait à réactiver le trunk avec son homologue PARTNER id(32768,00:0e:83:16:f5:00).

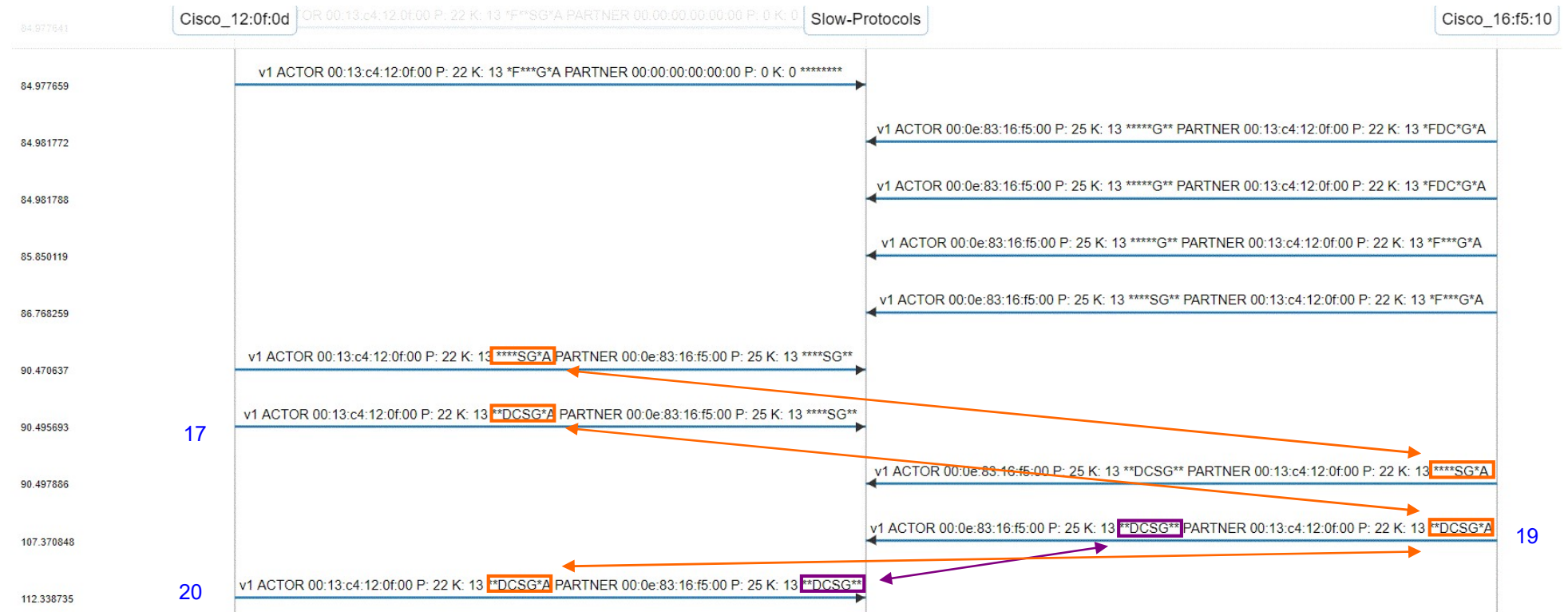
Le PARTNER serait, selon l'ACTOR, dans l'état "Distributeur-Collecteur-aGrégable-ShortTimeout". Compte tenu du bit "ShortTimeout", la requête va être répétée 3 fois, une fois par seconde. Mais pas de réponse.



Dans cet échange, l'adresse MAC du PARTNER, son port est à 0, et surtout sa clef (Key) est à 0. Donc il n'y a plus d'agrégation en cours. Comme on est en Long Timeout, les envois devraient être toutes les 30 secondes à partir du point bleu. On remarque que le switch s'est configuré avec ses paramètres par défaut (bit F defaulted). On peut penser qu'il se réinitialise ? Ci-dessous, on voit qu'il ne reste plus que les bits "A Actif" et "G aGrégable".



On en vient à faire l'observation suivante : l'ACTOR est l'émetteur d'une LACP PDU, et le PARTNER le récepteur. Au moins pour l'implantation de 802.1AX-2014 par CISCO. C'est une conjecture.



La suite des échanges laisse penser à un échange d'états des switches par des LACP PDU. Il semble qu'aux trames 19 et 20 chaque switch voit le bon état de son homologue car les bits d'état sont renvoyés de façon similaire excepté le rôle ACTOR/PARTNER en fonction de celui qui émet la trame. A la fin, le switch de gauche est considéré comme "Actif" et celui de droite passif (A=0 pour lui). Les bits indicateurs sont renvoyés en miroir.

L'échange des LACP PDU 17-19-20 fait penser à une espèce de three way hand shake.

