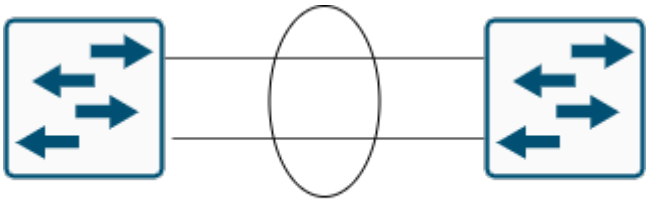
Expérience 4: Agrégat LACP entre 2 switchs

Objectif de l'expérience

Comparer le comportement de STP avec agrégat LACP à celui de deux liens indépendants entre deux switchs, et montrer que STP ne bloque plus un des deux liens, mais voit l’agrégat comme un seul lien logique.

Topologie

Graphique



Description

Liste des équipements:

- Switchs:
  - sw-access-1
  - sw-access-2

Deux liens Ethernet parallèles relis *sw-access-1* et *sw-access-2* (caractéristiques égales), avec un agrégat logique. Sur chaque switch, les deux interfaces sont regroupées dans un même groupe LACP.

Configuration appliquée

Paramètres STP

- Mode STP: `PVST+`
- Priorité STP: `automatique`
- Coût des liens identiques

Configuration LACP

- Sur *sw-access-1*: les interfaces `0/1` et `0/2` sont dans le groupe LACP `Port-Channel1`
- Sur *sw-access-2*: les interfaces `0/1` et `0/2` sont dans le groupe LACP `Port-Channel1`

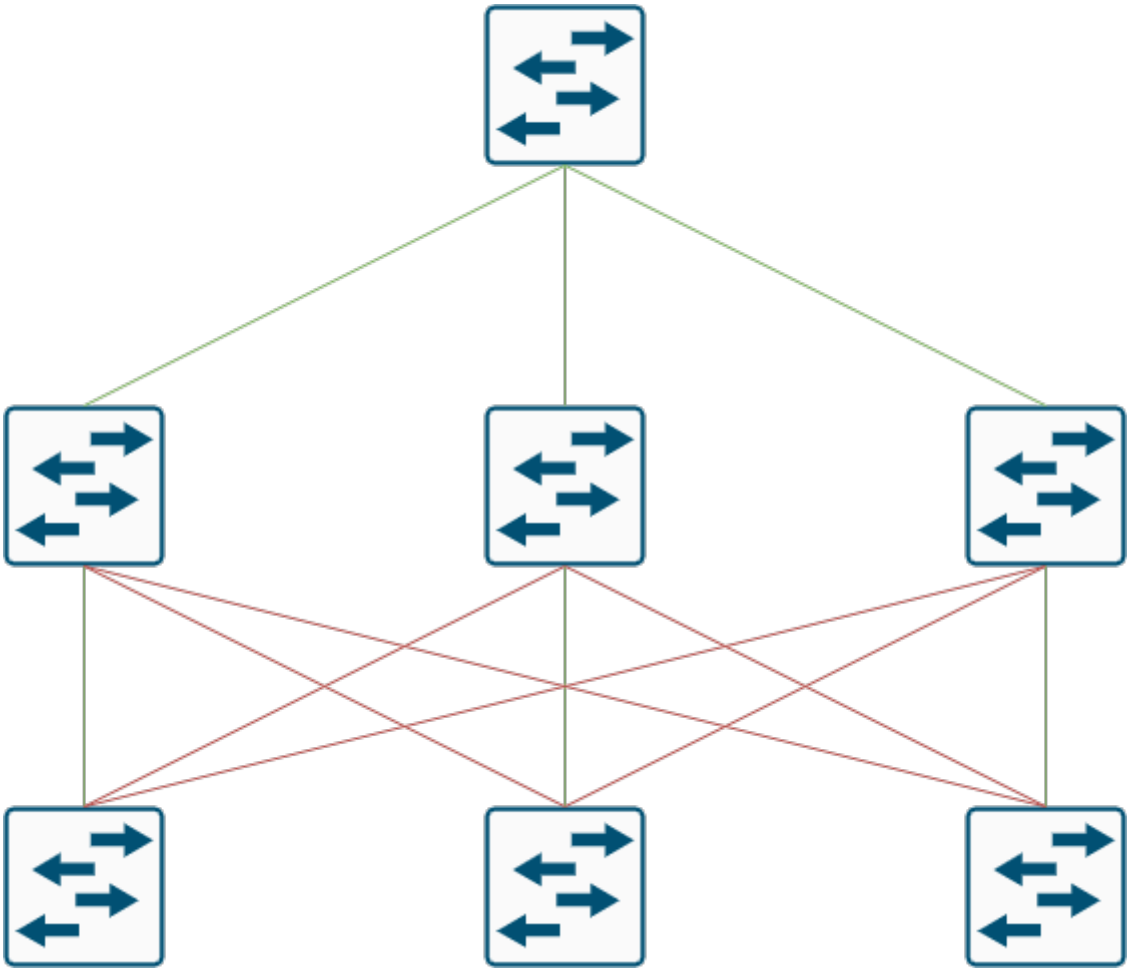
Autres paramètres

- VLAN: uniquement `VLAN 1`

Extrait de configuration

Comportement attendu

Graphique

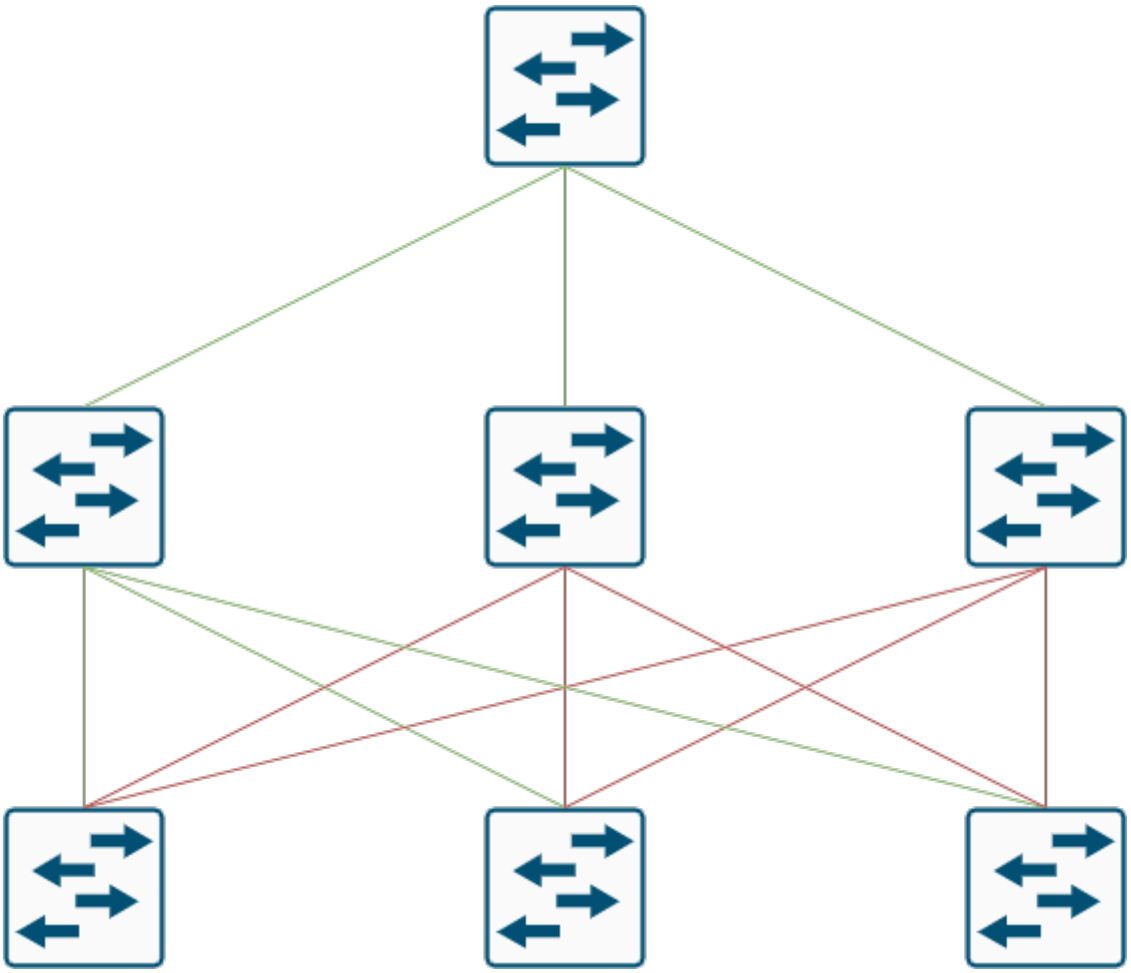


Description

Aucun des liens physiques membres de l’agrégat n’est bloqué par STP.

Résultats observés

Graphique



Description

STP a bien bloqué **deux liens sur chaque switch** d'accès pour casser les boucles. Par contre, les liens `up` ne sont plus reliés qu'à un seul switch de ditribution, les autres ne servants donc plus en temps normal.

Analyse

- Pourquoi 2 liens sur 3 sont bloqués sur chaque accès ?
  - STP impose un seul *Root Port* par switch et par VLAN.
  - Chaque accès a 3 chemins de coût identique vers le *Root Bridge* (via les 3 distrib).
  - STP choisit le chemin « préféré » selon :
    - Coût total vers le root (identique ici).
    - Puis Bridge ID de la distribution (priorité + MAC).
    - Puis Port ID si nécessaire.
    - Le port gagnant devient *Root Port*.
- Les deux autres ports, qui offrent un chemin redondant vers le root, deviennent *Alternate* et sont mis en `Blocking/Discarding` pour casser les boucles.
- Conséquence globale :
  - La topologie logique devient :

    - sw-core <-> sw-distrib-* <-> sw-access-*
  - Mais chaque accès n'utilise qu'une seule distribution en régime normal.
  - Les autres liens ne servent qu'en cas de panne.

Avantages

- Redondance :
  - Si la distribution active d'un accès tombe, STP peut activer un des liens bloqués vers une autre distribution.
  - Le réseau reste joignable, même en cas de perte d'un switch de distribution.
- Hiérarchie claire :
  - Cœur → Distribution → Accès, avec un root bien défini (*sw-core*).
  - Comportement STP cohérent avec la logique de design hiérarchique.
- Prévisibilité :
  - En contrôlant les priorités des distributions, on peut décider quelle distrib sera privilégiée par chaque accès (en jouant sur les coûts ou la topologie)

Inconvénients

- Bande passante gâchée :
  - 2 liens sur 3 par accès sont inutilisés en régime normal.
  - La topologie physique est riche, mais STP n'en exploite qu'une partie. C'est le cas dans notre expérience où en régime normal, deux switchs ne sont pas utilisés.
- Chemins non optimaux :
  - Un accès pourrait être physiquement plus proche d'une autre distribution, mais STP choisit selon les critères de coût/ID, pas forcément selon la logique « géographique ».
- Complexité de compréhension :
  - Sur 7 switchs avec plusieurs liens bloqués, il faut bien analyser les rôles STP pour comprendre le chemin réel des trames.
- Convergence :
  - En cas de panne d'une distribution, STP doit recalculer l'arbre → interruption possible avant activation d'un lien `Alternate`

Conclusion

Cette expérience montre que, dans une topologie cœur–distribution–accès avec accès multi-raccordés à plusieurs distributions, STP ne garde qu'un seul chemin actif par switch d'accès vers le cœur. Les 2 autres liens sont bloqués pour éviter les boucles, même si physiquement la topologie permettrait d'utiliser plus de chemins. On obtient ainsi une redondance fonctionnelle mais une utilisation très partielle de la topologie physique, ce qui met en évidence les limites de STP dans des architectures campus riches en liens

---

Revision #4
Created 2026-03-07 20:01:54 UTC by Arthur Dodin
Updated 2026-03-07 20:12:03 UTC by Arthur Dodin