Dépannage des performances TCP sur Nexus 9000 (NX-OS)

Table des matières

- Introduction
- Conditions préalables
  - Exigences
  - Composants utilisés
- Informations générales
  - Qu'est-ce que TCP
  - Trois avantages clés
  - Présentation de l'encapsulation TCP/IP
  - En-tête Ethernet (IEEE 802.3)
  - En-tête IP (IPv4)
  - Structure d'en-tête TCP
  - Options TCP (Commun 10)
  - Séquence TCP et comportement des accusés de réception (y compris SYN/FIN)
    - Exemple 1 : SYN avec données (TCP Fast Open)
    - Exemple 2 : FIN avec données (fin de connexion)
  - MSS et sa relation avec MTU
    - Fonctionnement de la négociation MSS dans la connexion TCP en trois étapes
    - Règle de clé : MSS est directionnel
    - La source peut-elle envoyer plus de données utiles TCP que la destination MSS ?
    - Informations pratiques pour le dépannage
  - Taille de fenêtre (contrôle de flux)
- Dépannage du plan de données TCP sur Cisco Nexus 9000 (NX-OS)
  - Validation initiale (accessibilité)
  - Identification du chemin de trafic (interfaces)
    - Configuration ELAM (évolutivité du cloud Nexus 9300)
    - Référence
  - Validation Au Niveau De L'Interface
  - Routage et stabilité ARP
  - Vérification que le trafic n'est pas dirigé vers le processeur
  - Détermination de la latence de transfert de paquets
    - SPAN vers CPU (capture de paquets pour le plan de données)
    - Validation De Limitation De Débit Du Plan De Contrôle
    - Validation basée sur ICMP avant TCP
    - Détermination de la latence de transfert du commutateur Nexus par capture de paquets
    - Références
- Analyse du trafic TCP à partir de la capture de paquets de l'hôte source
  - Analyse de la connexion TCP en trois étapes
    - Identification Du Trafic
    - Analyse du temps de parcours aller-retour initial (iRTT)

- Identification du port TCP
- Analyse de la taille de fenêtre TCP
- Analyse du débit, du temps de transfert et des conditions requises
- Longueur d'en-tête IP et TCP
- Analyse des options TCP et durée de vie

Analyse TCP RTT : RTT ACK et RTT initial

Analyse des retransmissions TCP et des retransmissions erronées

- Retransmissions TCP dans le temps
- Retransmissions TCP erronées

Analyse efficace du débit

Analyse des données en vol (fenêtre TCP)

Analyse de la charge utile TCP par rapport à MSS dans le temps

Analyse de la cause première (RCA) : Dégradation des performances TCP

Conclusion

Solution

- Réflexion technique

Introduction

Ce document décrit les principes fondamentaux de TCP, l'analyse approfondie des paquets Wireshark et le dépannage pratique pour optimiser les performances de bout en bout.

Conditions préalables

Exigences

Cisco vous recommande de prendre connaissance des rubriques suivantes :

- IP/TCP

Composants utilisés

Les informations contenues dans ce document sont basées sur les versions de matériel et de logiciel suivantes :

- Cisco Nexus 9000 Cloud Scale avec Cisco NX-OS 10.6(X).

Remarque : Toute question relative à la configuration et à l'interopérabilité de logiciels ou de matériels tiers n'est pas prise en charge par Cisco. L'utilisation d'outils tiers est une démonstration de votre configuration et de votre fonctionnement avec les équipements Cisco.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. Si votre réseau est en ligne, assurez-vous de bien comprendre l’incidence possible des commandes.

Informations générales

Qu'est-ce que TCP

Le protocole TCP (Transmission Control Protocol) est un protocole fondamental de la couche transport qui fonctionne au niveau de la couche 4 du modèle OSI et fournit une livraison fiable, ordonnée et contrôlée des erreurs d’un flux d’octets entre des applications communiquant sur un réseau IP.

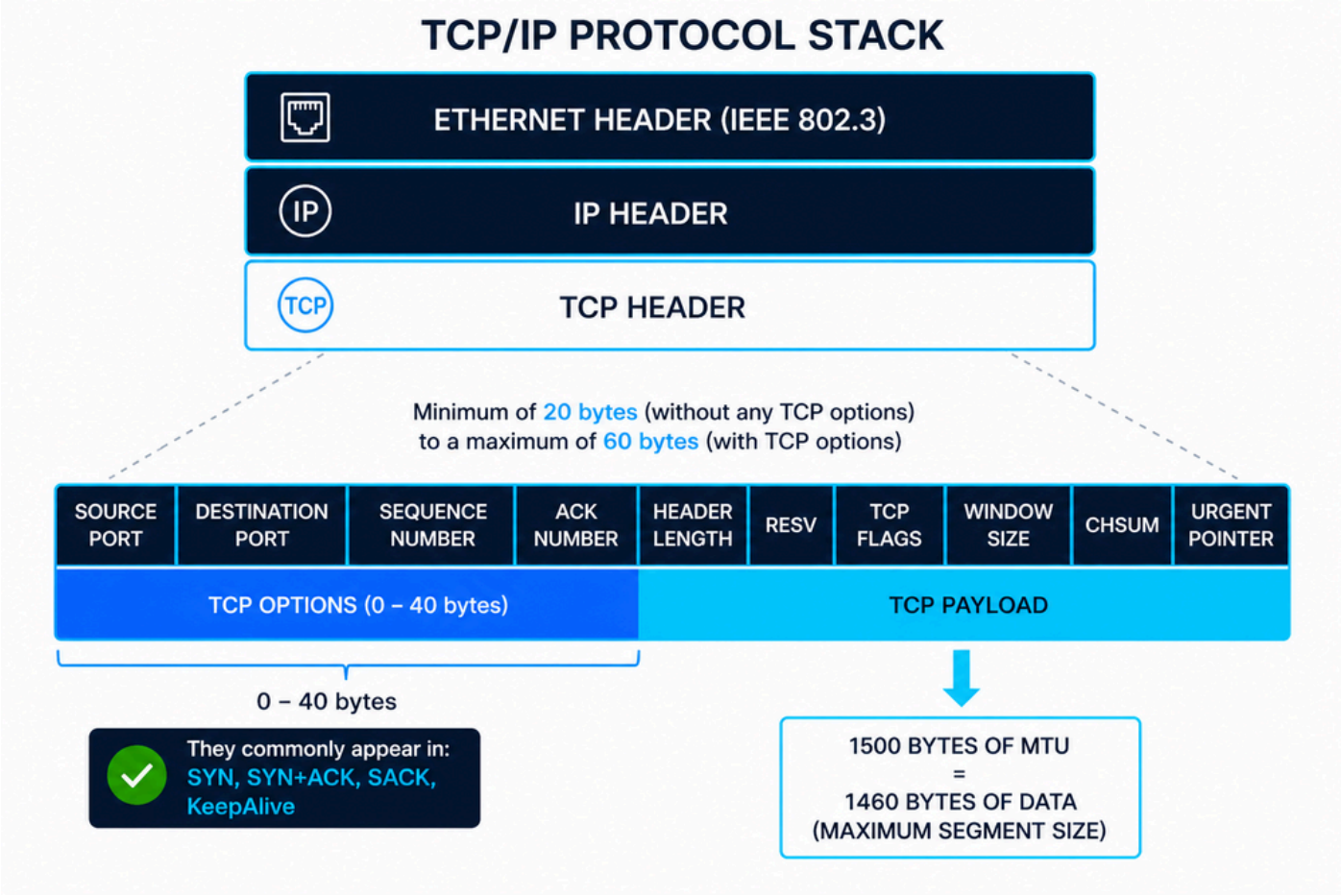
Trois avantages clés

1. Fiabilité : le protocole TCP est orienté connexion et garantit la livraison en exigeant des accusés de réception de la part du destinataire. Si un paquet est perdu ou corrompu pendant la transmission, le protocole TCP retransmet automatiquement les données pour s’assurer qu’elles atteignent leur destination.
2. Livraison commandée : Comme les paquets réseau peuvent arriver dans le désordre, le protocole TCP attribue des numéros de séquence à chaque segment. Cela permet au système récepteur de réassembler les données dans l'ordre exact dans lequel elles ont été envoyées.
3. Contrôle de flux et de congestion : Le protocole TCP gère de manière dynamique le taux de transmission des données afin de faire correspondre la capacité de traitement des récepteurs et les conditions actuelles du réseau, évitant ainsi la perte de données due à des dépassements de mémoire tampon ou à un encombrement du réseau.

Présentation de l'encapsulation TCP/IP

Le schéma représente la pile TCP/IP dans laquelle un segment TCP (couche 4) est encapsulé dans un paquet IP (couche 3), puis à l’intérieur d’une trame Ethernet (couche 2) définie par la

norme IEEE 802.3. Cette approche en couches assure une communication modulaire, où chaque couche ajoute ses propres informations de contrôle (en-têtes) pour garantir la livraison, le routage et l'intégrité des données.



En-tête Ethernet (IEEE 802.3)

L'en-tête Ethernet est généralement de 14 octets, composé des éléments suivants :

- Adresse MAC de destination (6 octets)
- Adresse MAC source (6 octets)
- EtherType/Length (2 octets)

En outre, les trames Ethernet incluent une queue de bande FCS (Frame Check Sequence) de 4 octets pour la détection des erreurs au niveau de la couche 2. IEEE 802.3 définit le tramage, les tailles de trame minimales/maximales et les contraintes de livraison physique qui ont un impact direct sur les protocoles de couche supérieure tels que TCP.

En-tête IP (IPv4)

L'en-tête IPv4 a une taille minimale de 20 octets, extensible jusqu'à 60 octets avec options. Les champs clés sont les suivants :

- Adresses IP de source et de destination
- Durée de vie (TTL)
- Protocole (identifie TCP comme charge utile)

La couche IP est responsable de l’adressage logique et du routage sur les réseaux, mais elle ne garantit pas la fiabilité.

Structure d'en-tête TCP

L'en-tête TCP comporte entre 20 et 60 octets, selon les options disponibles. Les champs clés sont les suivants :

- Ports source/de destination
- Numéro d'ordre
- Numéro de reçu
- Indicateurs (SYN, ACK, FIN, RST, etc.)
- Taille de fenêtre
- Somme De Contrôle

Le protocole TCP ajoute une livraison fiable, un séquençage approprié et un contrôle de flux à la communication IP.

Options TCP (Commun 10)

Les options TCP étendent le protocole de base. Les plus courants sont les suivants :

1. Maximum Segment Size (MSS) : définit la charge utile TCP la plus importante qu'un hôte peut accepter.
2. Échelle de fenêtre : étend la fenêtre de réception au-delà de 65 535 octets.
3. Accusé de réception sélectif autorisé (SACK autorisé) : active la fonctionnalité d'accusé de réception sélectif.
4. Accusé de réception sélectif (SACK) - Spécifie les blocs de données reçus pour éviter les retransmissions complètes.
5. Horodatages - Utilisés pour le calcul de la RTT et la protection contre les numéros de séquence encapsulés (PAWS).
6. No-Operation (NOP) : remplissage pour l'alignement des options.
7. End of Option List (EOL) : marque la fin des options TCP.

8. TCP Fast Open (TFO) : permet l'échange de données lors de la connexion initiale.
9. TCP multichemin (MPTCP) : active plusieurs chemins réseau pour une seule session TCP.
10. User Timeout Option (UTO) : contrôle la durée pendant laquelle les données transmises peuvent rester sans accusé de réception.

Séquence TCP et comportement des accusés de réception (y compris SYN/FIN)

Les indicateurs SYN et FIN utilisent chacun un numéro de séquence, même en l'absence de données utiles. Le protocole TCP utilise un modèle de séquençage orienté octet, dans lequel chaque octet transmis, ainsi que des indicateurs de contrôle spécifiques, font avancer l’espace de séquence. Ce comportement est essentiel pour une analyse TCP précise dans les captures de paquets et pour diagnostiquer les incohérences de séquençage ou d’accusé de réception.

ACK = SEQ + longueur de la charge utile + (SYN ? 1: 0) + (FIN ? 1: 0)

Where:

- SEQ = Numéro de séquence initial
- Longueur de la charge utile = taille des données en octets
- SYN ? 1: 0 = Ajoute 1 si l'indicateur SYN est défini, sinon 0
- FIN ? 1: 0 = Ajoute 1 si l'indicateur FIN est défini, sinon 0
- ACK = Prochain octet attendu

Exemple 1 : SYN avec données (TCP Fast Open)

- SEQ = 1000
- SYN = 1
- Longueur de la charge utile = 200 octets

Calcul ACK :

- ACK = 1 000 + 200 + 1 + 0 = 1 201

Cela reflète un scénario dans lequel des données sont envoyées pendant la connexion TCP. La charge utile et l'indicateur SYN consomment tous deux de l'espace de séquence.

Exemple 2 : FIN avec données (fin de connexion)

- SEQ = 3 000
- FIN = 1
- Longueur de la charge utile = 150 octets

Calcul ACK :

- ACK = 3 000 + 150 + 0 + 1 = 3 151

Cela montre que le protocole TCP peut inclure des données lors de l'interruption de la connexion, et que la charge utile et l'indicateur FIN incrémentent le numéro de séquence.

MSS et sa relation avec MTU

La taille de segment maximale (MSS) définit la charge utile maximale que TCP peut envoyer dans un segment.

- MTU Ethernet type = 1 500 octets
- MSS = MTU − En-tête IP − En-tête TCP
- MSS standard = 1 460 octets (1 500 − 20 − 20)

Si des options TCP sont présentes, MSS est réduit en conséquence. MSS est négocié pendant la connexion TCP en trois étapes et empêche la fragmentation au niveau de la couche IP.

Fonctionnement de la négociation MSS dans la connexion TCP en trois étapes

La taille de segment maximale (MSS) est échangée lors de la connexion TCP en trois étapes à l'aide de l'option MSS dans les paquets SYN :

- Hôte A → Hôte B (SYN) : annonce sa MSS (par exemple, 1460)
- Hôte B → Hôte A (SYN-ACK) : annonce son MSS (par exemple, 1380)

Chaque partie se contente de dire :

Il s'agit de la charge utile TCP la plus importante acceptée.

Règle de clé : MSS est directionnel

Les MSS ne sont pas négociées en tant que valeur unique convenue.

Au lieu de cela :

- Chaque hôte utilise le MSS annoncé par l’autre côté.
- Cela crée deux limites indépendantes, une par direction.

Par conséquent :

- A envoie des données en utilisant le MSS de B.
- B envoie des données à l'aide du MSS de A.

La source peut-elle envoyer plus de données utiles TCP que la destination MSS ?

Dans une pile TCP fonctionnant correctement : No.

- L'expéditeur doit respecter la MSS annoncée par le destinataire.
- L'envoi de segments plus importants risquerait :
  - Fragmentation IP (si MTU est dépassé)
  - Suppression de paquets (si la fragmentation est bloquée ou non prise en charge)
- Cela mène à :
  - Retransmissions
  - Dégradation des performances
  - Problèmes tels que les trous noirs PMTUD (Path MTU Discovery)

Informations pratiques pour le dépannage

- Vérifiez toujours les valeurs MSS dans la connexion TCP en trois étapes (paquets SYN/SYN-ACK).
- Recherchez les incohérences causées par :
  - Tunnels (VXLAN, GRE, IPsec)
  - Pare-feu modification MSS (verrouillage MSS)
- Sur des plates-formes telles que Cisco NX-OS, l'ajustement MSS est souvent utilisé pour empêcher la fragmentation entre les chemins encapsulés

Taille de fenêtre (contrôle de flux)

La taille de fenêtre définit la quantité de données que le récepteur peut accepter sans accusé de réception.

Ce que c'est :

- Mécanisme de contrôle de flux pour empêcher le dépassement de tampon.

Objectif:

- Garantit que l'expéditeur ne submerge pas le destinataire.

Où l'obtenir :

- Visible dans les captures de paquets (par exemple, Wireshark).
- Provient de la configuration de la pile TCP du système d'exploitation et de la taille du tampon.

Variabilité fournisseur/système d'exploitation :

- Différentes mises en oeuvre (Linux, Windows, Cisco NX-OS) utilisent l'évolutivité dynamique et le réglage de la mémoire tampon, ce qui entraîne des tailles de fenêtre variables.

Condition de fenêtre zéro :

- Lorsque Taille de fenêtre = 0, la mémoire tampon du récepteur est pleine.
- L'expéditeur interrompt la transmission et envoie des sondes périodiques.

Mécanismes de fenêtres variables

- Contrôle De Flux Basé Sur Le Débit
    - Il attribue à l'expéditeur un débit de données fixe et veille à ce que les données ne dépassent jamais cette allocation.
    - Idéal pour les applications de streaming.
    - Diffusion et multidiffusion
- Contrôle De Flux Basé Sur Fenêtre
    - La taille de la fenêtre varie avec le temps.
    - Le récepteur assure le contrôle de flux en signalant la fenêtre autorisée aux mises à jour de la fenêtre de l'expéditeur.

Dépannage Utilisation :

- Fenêtres petites ou zéro → Goulot d'étranglement côté récepteur (processeur, mémoire, application).
- Grandes fenêtres mais débit faible → Problèmes réseau (latence, congestion).
- L’analyse du comportement des fenêtres est essentielle pour diagnostiquer les problèmes

de performances dans les sessions TCP.

Dépannage du plan de données TCP sur Cisco Nexus 9000 (NX-OS)

Cette section décrit une méthodologie pratique pour diagnostiquer si un commutateur Cisco Nexus exécutant NX-OS affecte le transfert du trafic TCP ou introduit des problèmes de performances. L'approche est présentée à travers un scénario hypothétique.

Lorsque la latence TCP ou la dégradation des performances sont observées, il est courant de soupçonner que le réseau en est la cause. Toutefois, cette hypothèse doit être validée par une analyse axée sur les données. La méthode de dépannage TCP faisant autorité est la capture de paquets, idéalement exécutée :

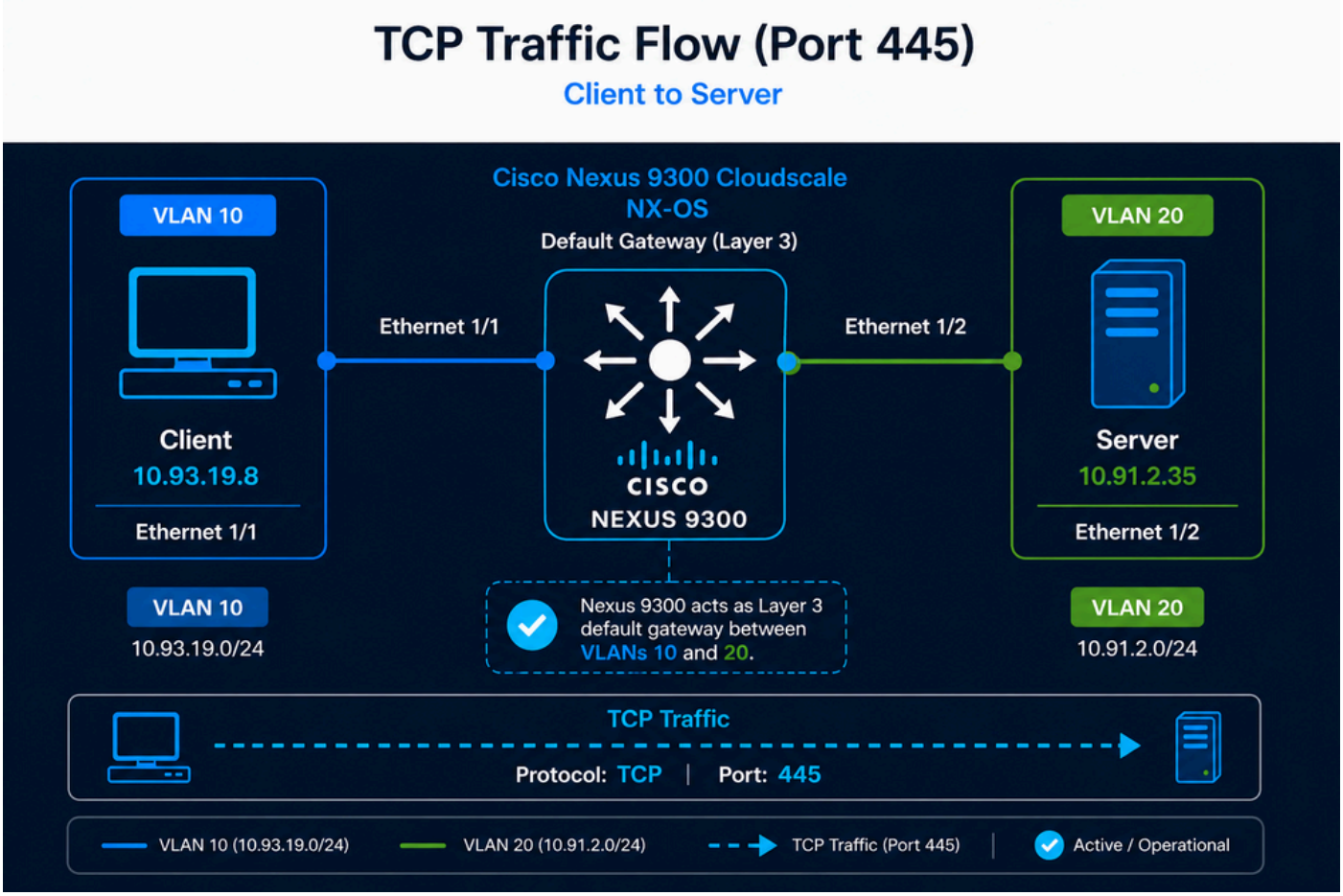
- Simultanément à la source et à destination
- Avant le lancement du trafic

Cela garantit la visibilité de la connexion TCP en trois étapes, au cours de laquelle des paramètres critiques tels que MSS, Window Scale et SACK sont négociés et ne sont pas répétés plus tard dans la session. Si des captures simultanées ne sont pas possibles, l'analyse peut se poursuivre avec une capture unique, mais les conclusions sont limitées.

Définition de scénario

Un utilisateur a constaté que le processus de sauvegarde d'un jeu de données d'application d'environ 7,5 To, qui était auparavant effectué en environ 9 heures, prend désormais près de 21 heures. Bien que les sessions TCP entre le client et le serveur soient toujours établies avec succès, l’augmentation significative de la durée de sauvegarde suggère une dégradation potentielle du débit ou des performances TCP globales. Étant donné que le commutateur Nexus est le seul périphérique réseau sur le chemin et qu'il fournit également une fonctionnalité de passerelle de couche 3, l'administrateur réseau soupçonne que le commutateur Nexus est la cause du problème.

- Client : 10.93.19.8 (VLAN 10)
- Serveur : 10.91.2.35 (VLAN 20)
- Nexus 9300 agissant comme passerelle par défaut
- Port TCP 445



Validation initiale (accessibilité)

- Ces commandes sont utilisées pour valider la MTU de chemin (PMTU) entre une source et une destination en envoyant des paquets ICMP avec le bit Ne pas fragmenter (DF) défini. Cela permet de déterminer la taille de paquet maximale qui peut traverser le réseau sans fragmentation. Ce processus doit être effectué à la fois sur la source et la destination.
- Vérifiez toujours la MTU de l'interface physique sur la source et la destination.
- Dans ce scénario, l'accès est disponible uniquement pour l'hôte source, où un MTU de 1500 a été identifié.

```
Linux: ping -c 10 -I 10.93.19.8 -s 1472 -M do 10.91.2.35
```

- -c 10 → Envoie 10 requêtes d'écho ICMP
- -I 192.168.10.10 → Utilise cette interface/IP source spécifique
- -s 1472 → Définit la taille de la charge utile ICMP à 1472 octets
- -M do → Définit le bit DF (Ne pas fragmenter)
- 192.168.20.20 → Adresse IP de destination

```
Windows: ping -n 10 -l 1472 -f 10.91.2.35
```

- -n 10 → Envoie 10 requêtes d’écho ICMP
- -l 1472 → Définit la taille de la charge utile ICMP à 1472 octets
- -f → Active l'indicateur Ne pas fragmenter (DF)
- 192.168.20.20 → Adresse IP de destination

Pourquoi 1 472 octets ?

- Charge utile ICMP = 1 472 octets
- En-tête IP = 20 octets
- En-tête ICMP = 8 octets
- Taille totale du paquet : 1 472 + 20 + 8 = 1 500 octets (MTU standard)
- Ceci teste si le chemin prend en charge un MTU de 1500 octets sans fragmentation. Si vous tentez d'envoyer 1 500 octets de données utiles ICMP, la requête ping peut échouer, car la taille totale du paquet dépasserait le MTU standard après l'ajout des en-têtes IP et ICMP.

Ce qui peut être conclu

- Si la requête ping aboutit (aucune perte de paquet), le chemin prend en charge au moins une MTU de 1 500 octets et aucune fragmentation n’est requise.
    - Nettoyer les résultats ICMP → passer à l’analyse TCP
    - Succès de la requête ping intermittente → perte de paquets possible, congestion transitoire, limitation du débit ou problème de transmission ; procéder à l’analyse des pertes de paquets, car le protocole TCP nécessite un chemin sans perte pour fonctionner efficacement.
- Si la requête ping échoue avec l'erreur « Fragmentation requise » ou expire, il y a une liaison dans le chemin avec une MTU inférieure à 1500 octets, le paquet ne peut pas être transféré en raison du bit DF, et cela indique un problème de MTU de chemin.

Comment l'utiliser pour le dépannage

- Réduisez progressivement la taille de la charge utile (par exemple, 1 472 → 1 400 → 1 300) pour identifier la taille la plus grande qui fonctionne.
- Une fois identifié, calculez la MTU à l'aide de la formule MTU = charge utile + 28 octets (en-têtes IP + ICMP).

Pertinence pratique pour le protocole TCP

- Si la MTU est plus petite que prévu, les segments TCP peuvent être fragmentés ou

abandonnés.
- Cela entraîne des retransmissions, une latence accrue et un débit réduit, ce qui a un impact direct sur les performances des applications.

Identification du chemin de trafic (interfaces)

Pour dépanner efficacement les performances TCP sur un commutateur Cisco Nexus 9000, il est essentiel de déterminer quelles interfaces reçoivent et transfèrent le trafic entre la source et la destination.

Dans les topologies simples, cela peut être déduit directement des connexions physiques. Par exemple, si le client est connecté à Ethernet1/1 et le serveur à Ethernet1/2, le chemin du trafic est direct. Cependant, dans les environnements réels avec plusieurs interfaces actives, canaux de port ou configurations vPC, cette identification n'est pas toujours facile.

Dans ce cas, l'approche recommandée consiste à utiliser le module ELAM (Embedded Logic Analyzer Module), qui offre une visibilité au niveau ASIC (matériel de plan de données).

ELAM vous permet de capturer un paquet pendant qu'il est traité par le pipeline de transfert et révèle des informations critiques telles que :

- Interface d'entrée
- Interface de sortie
- Décision de transfert (résultat de la recherche L2/L3)

Cette méthode est beaucoup plus précise que l'utilisation d'outils de plan de contrôle, car elle reflète le chemin de transfert matériel réel.

Il est important de noter qu'ELAM ne capture qu'un seul paquet à la fois, de sorte que les critères de filtrage doivent être définis avec précision pour correspondre au trafic souhaité (par exemple, IP source, IP de destination, port TCP). Si les filtres sont trop larges, il y a un risque de capture du trafic non lié tel que ICMP ou UDP au lieu du flux TCP prévu.

En outre, ce processus doit être répété pour les deux sens de trafic :

- Source → Destination
- Destination → Source

Dans les environnements utilisant vPC ou ECMP, la charge du trafic peut être équilibrée sur plusieurs chemins. Par conséquent, le trafic de transfert et de retour peut traverser différents

commutateurs ou interfaces. Dans ces scénarios, ELAM doit être exécuté sur chaque commutateur Nexus concerné pour garantir une visibilité complète.

En identifiant avec précision les interfaces d'entrée et de sortie, la portée du dépannage est considérablement réduite, ce qui permet une validation ciblée des compteurs d'interface, des politiques QoS, des paramètres MTU et des points de congestion potentiels le long du chemin de transfert exact.

Configuration ELAM (évolutivité du cloud Nexus 9300)

Cet exemple filtre le trafic avec l'IP source 10.93.19.8, l'IP de destination 10.91.2.35 et le port de destination TCP 445.

Configuration d'ELAM

<#root>

```
switch#

debug platform internal tah elam


switch(TAH-elam)#

trigger init


Slot 1: param values: start asic 0, start slice 0, lu-a2d 1, in-select 6, out-select 0
switch(TAH-elam-insel6)#

set outer ipv4 src_ip 10.93.19.8


switch(TAH-elam-insel6)#

set outer ipv4 dst_ip 10.91.2.35


switch(TAH-elam-insel6)#

set outer l4 l4-type 0


switch(TAH-elam-insel6)#

set outer l4 dst-port 445
```

```
switch(TAH-elam-insel6)#

start
```

Après avoir généré le trafic, récupérez le résultat :

<#root>

```
switch(TAH-elam-insel6)#

report
```

Capture inverse du trafic (obligatoire pour une visibilité totale)

Pour valider le chemin de retour, répétez la configuration en échangeant les adresses IP source et de destination :

<#root>

```
switch#

debug platform internal tah elam


switch(TAH-elam)# trigger init
Slot 1: param values: start asic 0, start slice 0, lu-a2d 1, in-select 6, out-select 0
switch(TAH-elam-insel6)#

set outer ipv4 dst_ip 10.93.19.8


switch(TAH-elam-insel6)#

set outer ipv4 src_ip 10.91.2.35


switch(TAH-elam-insel6)#

set outer l4 l4-type 0


switch(TAH-elam-insel6)#
```

```
set outer l4 dst-port 445
```

```
switch(TAH-elam-insel6)#
```

```
start
```

notes opérationnelles

- ELAM ne capture qu'un seul paquet, afin de s'assurer que le trafic circule activement lors du démarrage de la capture.
- Les filtres doivent être précis pour éviter de capturer le trafic non lié.
- Dans les environnements vPC, exécutez ELAM sur les deux commutateurs, car le trafic peut être haché différemment dans chaque direction.
- Les résultats affichent l'interface d'entrée, l'interface de sortie et la décision de transfert dans le matériel, offrant ainsi une visibilité faisant autorité sur le plan de données.

Référence

Guide ELAM ASIC évolutif pour le cloud Cisco Nexus 9000

Validation Au Niveau De L'Interface

La validation au niveau de l'interface garantit que le commutateur Nexus n'introduit aucune contrainte ou anomalie affectant le trafic TCP. L'objectif est de vérifier que la configuration, l'état opérationnel et les compteurs matériels sont cohérents avec le comportement attendu pour un transfert de plan de données hautes performances.

Validation de configuration

- Vérifiez qu'aucune liste de contrôle d'accès restrictive n'est appliquée aux interfaces :

<#root>

```
switch#
```

```
show running-config interface ethernet1/1-2 | include access-group
```

- Vérifier qu'aucune stratégie QoS non intentionnelle n'affecte le trafic (QoS globale et au niveau de l'interface, y compris la mise en file d'attente, la réglementation et la mise en forme) :

<#root>

switch#

show running-config interface ethernet1/1-2 | include service-policy

switch#

show policy-map interface ethernet1/1-2

<#root>

switch#

show policy-map

<#root>

switch#

show class-map

<#root>

switch#

show class-map type network-qos

<#root>

switch#

show policy-map type network-qos

<#root>

switch#

show policy-map system type network-qos

<#root>

switch#

show queuing interface ethernet1/1-2

<#root>

switch#

show policy-map type queuing

- Confirmez la configuration de la couche 2 ou 3 (port de commutation ou interface routée), y compris l'appartenance au VLAN, l'état STP et l'adressage IP :

<#root>

switch#

show running-config interface ethernet1/1-2

<#root>

switch#

show interface ethernet1/1-2 switchport

<#root>

switch#

show spanning-tree interface ethernet1/1-2

<#root>

switch#

show ip interface ethernet1/1-2

Validation de l'état opérationnel

- Vérifiez la cohérence MTU et assurez-vous qu'elle correspond à la configuration attendue (par exemple, 1 500 ou 9 000 octets) :

<#root>

switch#

show interface ethernet1/1-2 | include MTU

- Confirmez les paramètres de vitesse et de duplex des interfaces :

<#root>

switch#

show interface ethernet1/1-2 | include speed|duplex

- Validation de la stabilité de l’interface (pas de battement ni de transitions fréquentes) :

<#root>

switch#

```
show interface ethernet1/1-2 | include rate|flap
```

Validation du compteur d'erreurs

- Effacer les compteurs avant le test :

<#root>

switch#

```
clear counters interface all
```

- Contrôler les compteurs d'erreurs (valeurs non nulles uniquement) :

<#root>

switch#

```
show interface counters errors non-zero | include Port|Eth1/1|Eth1/2
```

Validation post-test

- Réexécutez le test de trafic TCP et observez à nouveau les compteurs :

<#root>

switch#

```
show interface counters errors non-zero | include Port|Eth1/1|Eth1/2
```

- Les compteurs ne doivent pas augmenter ; toute augmentation indique des problèmes potentiels liés à la couche 1 ou au matériel, tels que des erreurs de liaison physique, des erreurs CRC/FCS ou des dépassements/abandons de mémoire tampon.

Routage et stabilité ARP

Il est essentiel de garantir la stabilité du routage et du protocole ARP pour confirmer que le

commutateur Nexus dispose d'une accessibilité de couche 3 cohérente et n'introduit pas de problèmes de résolution intermittents susceptibles d'affecter les performances TCP. L'instabilité des entrées de routage ou de la résolution ARP peut entraîner une perte de paquets, une latence accrue ou un blocage du trafic.

Critères de validation

- Les entrées de routage pour la source et la destination doivent être présentes, stables et ne pas changer fréquemment.
- Les entrées ARP doivent être résolues et ne doivent pas être continuellement actualisées ou manquantes.

<#root>

switch#

show ip route 10.93.19.8

<#root>

switch#

show ip route 10.91.2.35

<#root>

switch#

show ip arp detail | include 10.93.19.8

<#root>

switch#

show ip arp detail | include 10.91.2.35

Vérification que le trafic n'est pas dirigé vers le processeur

Dans les commutateurs Cisco Nexus 9000, le transfert est effectué dans le matériel (ASIC) et le processeur n'est pas impliqué dans les opérations normales du plan de données. Par conséquent, l'observation du trafic TCP d'hôte à hôte dans le plan de contrôle est anormale et indique que des paquets sont en cours de pontage en raison d'exceptions ou de mauvaises configurations. Une fois que le trafic doit être traité par le processeur, il est soumis à la réglementation du plan de contrôle, et on s'attend à ce que des abandons puissent être observés si le trafic dépasse le débit du plan de contrôle autorisé.

Méthode de validation

- Capturez le trafic atteignant le plan de contrôle à l'aide d'Ethanalyzer :

<#root>

```
switch#

ethanalyzer local interface inband display-filter "ip.addr==10.93.19.8 and ip.addr==10.91.2.35" limit-ca
```

Comportement attendu

- Aucun trafic de plan de données TCP d'hôte à hôte ne peut être observé dans le processeur.

Comportement Inattendu

- Si les paquets correspondant au flux sont visibles, le trafic est acheminé, ce qui peut être dû à :
  - Traitement de paquets exceptionnel (expiration TTL, journalisation ACL, redirections)
  - Configuration incorrecte ou fonctionnalités non prises en charge
  - Programmation matérielle incorrecte

Détermination de la latence de transfert de paquets

La latence de transfert de paquets dans les commutateurs Nexus 9000 dépend de la taille des paquets, du mode de transfert et des fonctionnalités activées. Les spécifications Cisco font généralement référence à la latence pour les paquets de 64 octets sous transfert cut-through.

```
+----------------------+----------------------+-------------------------+------------------------------
| Switch Model         | ASIC / Architecture  | Ports (example config)  | Typical Forwarding Latency
|                      |                      |                         | (64B packet)
```

Nexus 93180YC-EX	Cloud Scale (EX)	48x25G + 6x100G	~1.0 - 1.2 microseconds
Nexus 93180YC-FX	Cloud Scale (FX)	48x25G + 6x100G	~0.9 - 1.0 microseconds
Nexus 93180YC-FX2	Cloud Scale (FX2)	48x25G + 6x100G	~0.8 - 0.9 microseconds
Nexus 9364C	Cloud Scale	64x100G	~1.0 microsecond
Nexus 9336C-FX2	Cloud Scale (FX2)	36x100G	~0.8 microseconds
Nexus 93240YC-FX2	Cloud Scale (FX2)	48x25G + 12x100G	~0.8 - 0.9 microseconds
Nexus 92300YC	Broadcom Trident II	48x10/25G + 6x40/100G	~2 - 3 microseconds
Nexus 92160YC-X	Broadcom Tomahawk	48x25G + 6x100G	~2 microseconds

- Transfert Cut-through (par défaut dans Nexus 9000) :
  - Commence le transfert avant la réception du paquet complet.
  - Réduit la latence (de moins d'une microseconde à environ 1 µ).
- Stockage et retransmission :
  - Le paquet entier doit être reçu avant le transfert.
  - Ajoute une latence proportionnelle à la taille du paquet.

Des fonctionnalités supplémentaires peuvent introduire une latence incrémentielle :

- Encapsulation/décapsulation VXLAN
- Recherches ACL (traitement TCAM)
- Classification QoS et mise en file
- Télémétrie (NetFlow, ERSPAN, sFlow)
- Tampon pendant la congestion

Cependant:

- Ces opérations sont effectuées dans des pipelines matériels.

Le seul scénario réaliste où la latence augmente de manière significative est la congestion :

- Les paquets sont mis en mémoire tampon dans les files d'attente de sortie.
- Le délai dépend de :
  - Profondeur de file
  - Utilisation des interfaces
  - Stratégies QoS

Même dans ces cas :

- La latence est généralement comprise entre quelques microsecondes et quelques centaines de microsecondes.
- Un retard prolongé de quelques millisecondes impliquerait :

- Congestion sévère
- Sursouscription
- QoS ou mise en mémoire tampon incorrectement configurées

SPAN vers CPU (capture de paquets pour le plan de données)

Cela permet la mise en miroir du trafic du plan de données dans le plan de contrôle pour la capture de paquets et l'exportation vers un fichier .pcapng, permettant une analyse détaillée dans Wireshark.

Configuration

```
monitor session 1
 source interface ethernet1/1 both
 source interface ethernet1/2 both
 destination interface sup-eth0
 no shut
```

Exécution de capture

<#root>

```
switch#
```

ethanalyzer local interface inband mirror capture-filter "tcp port 445" limit-capture 0 write bootflash:

Considérations techniques

- Le trafic mis en miroir sur le processeur est soumis à la réglementation du plan de contrôle (CoPP).
- Si le trafic dépasse CoPP :
  - Les paquets ne peuvent être abandonnés que dans le plan de contrôle.
  - Cela crée des faux positifs pendant l'analyse.
- La fonctionnalité SPAN vers CPU est recommandée pour les scénarios de trafic faible à modéré.
- Pour les environnements à haut débit :
  - Utiliser la fonctionnalité SPAN locale (analyseur externe)
  - Utiliser ERSPAN pour la capture à distance

Méthode	Avantage	Limite
PORTÉE	Précision, pas d'encapsulation	Requiert une connexion physique.
REPORTER	Capacité de capture à distance	Sensible à la congestion du réseau.

Validation De Limitation De Débit Du Plan De Contrôle

Pour garantir la fiabilité des captures SPAN vers CPU, il est nécessaire de vérifier que le plan de contrôle ne supprime pas les paquets en miroir en raison de la limitation du débit.

Commande de validation

```
switch(config)# show hardware rate-limiter | i Allowed|span

Allowed, Dropped & Total: aggregated bytes since last clear counters

R-L Class     Config Allowed Dropped Total

span          50           0       0     0 <<<

span-egress   disabled     0       0     0
```

Méthodologie de validation

- Exécutez la commande à des intervalles d'environ 3 secondes.
- Observez les compteurs de pertes liés à la fonctionnalité SPAN.

Interprétation

- L'absence d'incrément dans les compteurs de dépôt pour la ligne SPAN indique une capture fiable.
- L'augmentation des compteurs de perte indique une perte de paquets dans le plan de contrôle, ce qui rend la capture peu fiable.

Si des pertes sont observées, la méthode de capture doit être remplacée par SPAN ou ERSPAN.

Validation basée sur ICMP avant TCP

Les tests ICMP fournissent une validation de base de l'intégrité du plan de données avant d'effectuer une analyse TCP complexe. Comme le protocole ICMP est plus simple et sans état, il permet de détecter rapidement les pertes de paquets, les duplications ou les incohérences de chemin.

Comportement attendu dans la capture SPAN

- Chaque paquet ICMP peut apparaître deux fois :
  - Une fois en entrée
  - Une fois en sortie
- Pour une requête ping standard :
  - Demande d'écho → 2 paquets
  - Réponse d'écho → 2 paquets

Cela confirme le transfert correct et l'absence de perte de paquets dans le plan de données.

Comportement Anormal

- Les doublons manquants ou le nombre asymétrique de paquets indiquent des pertes de paquets potentielles ou des limitations de capture.
- Les délais d'attente intermittents évoquent des problèmes de couche 1, de congestion ou en amont.

Si le trafic ICMP est transmis de manière cohérente sans perte, il est très probable que le trafic TCP soit également transmis correctement au niveau de la couche 2/3.

Détermination de la latence de transfert du commutateur Nexus par capture de paquets

Lorsque le trafic est capturé à l'aide de la fonctionnalité SPAN vers CPU (ou SPAN/ERSPAN), chaque paquet peut être observé deux fois : une fois en entrée et une fois en sortie. Cette duplication peut être utilisée pour estimer la latence de transfert introduite par le commutateur Nexus en calculant la différence de temps entre les deux instances du même paquet.

En pratique, cette latence peut être mesurée à l'aide du trafic ICMP précédemment capturé en comparant le délai entre les paquets de requête d'écho dupliqués et les paquets de réponse d'écho. Cela fournit une base simple et efficace pour les performances de transfert de commutateur. Si une analyse plus approfondie est nécessaire, la même méthodologie peut être appliquée au trafic TCP en capturant le flux et en mesurant la différence de temps entre les paquets TCP dupliqués.

Méthode

- Identifiez un paquet et son doublon (même numéro de séquence).
- Mesurez le délai entre les copies d'entrée et de sortie.
- Ce delta représente une estimation de limite supérieure de la latence de transfert du commutateur, car il peut inclure la mise en miroir et la surcharge d'horodatage.

Configuration Wireshark

- Activer l'affichage des écarts horaires :

```
View > Time Display Format > Seconds Since Previous Displayed Packet
```

- Ajouter une colonne personnalisée pour l'écart temporel :

```
Right-click on "Time Delta from Previous Displayed Packet" → Apply as Column
```

- Filtrer le trafic approprié (exemple) :

```
ip.addr==10.93.19.8 and ip.addr==10.91.2.35 and tcp
```

- Trier les paquets par numéro de séquence ou flux TCP :

```
Right-click packet → Follow → TCP Stream
```

Interprétation

- Le délai entre les paquets dupliqués peut être de l'ordre de la microseconde.
  - Si tel est le cas, le commutateur Nexus n'introduit pas de latence dans le transfert de paquets.
- Les faibles deltas constants confirment les performances de transfert basées sur le matériel.
- Des deltas plus élevés ou incohérents peuvent indiquer :
  - Congestion ou mise en mémoire tampon

Références

- Fiches techniques de la gamme Cisco Nexus 9000
- Guides de conception des commutateurs Cisco Nexus 9000
- Livre blanc sur la gestion intelligente des tampons sur les commutateurs Cisco Nexus 9000

Analyse du trafic TCP à partir de la capture de paquets de l’hôte source

Cette section fournit une méthodologie détaillée pour analyser une capture de paquets TCP dans Wireshark, y compris la configuration du profil, dans le cas hypothétique décrit précédemment. Les images présentées ont été prises directement à partir de Wireshark. Pour rappel, le scénario est le suivant :

Un utilisateur a constaté que le processus de sauvegarde d'un jeu de données d'application d'environ 6,5 To, qui était auparavant effectué en environ 9 heures, prend désormais près de 21 heures. Le seul périphérique réseau accessible est un commutateur Cisco Nexus 9300 connecté au serveur source (10.93.19.8). La MTU configurée sur l'interface du commutateur est de 9 000 octets (trames jumbo), alors que la MTU sur le serveur est inconnue. Une capture de paquets à partir du serveur source est disponible et toutes les étapes de validation Nexus précédentes ont déjà été effectuées sans qu'aucune anomalie ne soit détectée.

Principales observations et contraintes

- Le commutateur Nexus a été exclu :
  - Aucun abandon de paquet
  - Aucune erreur d’interface
  - Aucun impact QoS ou ACL
  - Transfert matériel confirmé
- Configuration d'interface :
  - Port d'accès
  - MTU: 9000 octets
- Données disponibles :
  - Capture de paquets à la source
  - Connaissances de bout en bout en matière de MTU
    - La requête ping s’est terminée sans fragmentation à l’aide d’un paquet de 1 500 octets contenant 1 472 octets de données.
- Données manquantes :
  - Visibilité de la destination

- Aucune capture de paquet n’est disponible sur le serveur de destination.

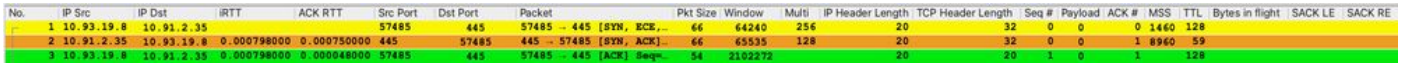
Dans Wireshark, vous pouvez créer des profils personnalisés en fonction du type d'analyse que vous souhaitez effectuer.

Description de colonne

- tcp.analysis.initial_rtt (iRTT) : Estime le temps de parcours aller-retour initial en fonction de la connexion TCP en trois étapes.
- tcp.analysis.ack_rtt (ACK RTT) : Mesure le temps entre un segment TCP et son accusé de réception correspondant.
- tcp.window_size (Fenêtre) : Indique la taille de fenêtre TCP annoncée du récepteur avant l'application de la mise à l'échelle.
- tcp.options.wscale.multiplicateur (Multi) : Représente le facteur d'échelle de fenêtre utilisé pour calculer la fenêtre de réception effective.
- tcp.seq (Seq#) : Affiche le numéro de séquence du premier octet dans le segment TCP.
- tcp.len (charge utile) : Affiche la taille de la charge utile TCP en octets pour ce segment.
- tcp.ack (ACK#) : Indique l'octet suivant attendu de l'expéditeur (accusé de réception cumulatif).
- tcp.options.mss_val (MSS) : Affiche la taille maximale de segment annoncée pendant la connexion TCP.
- ip.ttl (TTL) : Affiche la valeur de durée de vie, utile pour identifier le nombre de sauts et le comportement de routage.
- tcp.analysis.bytes_in_flight (Octets en vol) : Représente la quantité de données sans accusé de réception actuellement en transit.

Analyse de la connexion TCP en trois étapes

La capture de la connexion TCP en trois étapes est obligatoire, car elle contient des paramètres critiques tels que MSS, Window Scale et SACK qui définissent le comportement de la session. Sans ces informations, toute analyse TCP est incomplète et peut conduire à des conclusions incorrectes sur les performances ou la cause première.



No.	IP Src	IP Dst	iRTT	ACK RTT	Src Port	Dst Port	Packet	Pkt Size	Window	Multi	IP Header Length	TCP Header Length	Seq #	Payload	ACK #	MSS	TTL	Bytes in flight	SACK LE	SACK RE
1	10.93.19.8	10.91.2.35			57485	445	57485 → 445 [SYN, ECE,…	66	64240	256	20	32	0	0	0	1460	128			
2	10.91.2.35	10.93.19.8	0.000798000	0.000750000	445	57485	445 → 57485 [SYN, ACK]…	66	65535	128	20	32	0	0	1	8960	59			
3	10.93.19.8	10.91.2.35	0.000798000	0.000048000	57485	445	57485 → 445 [ACK] Seq=…	54	2102272		20	20	1	0	1		128			

Identification Du Trafic

À partir de la capture de paquets :

- Adresse IP source : 10.93.19.8

- Adresse IP de destination : 10.91.2.35

Analyse du temps de parcours aller-retour initial (iRTT)

Le RTT initial (iRTT) est calculé comme suit :

- iRTT = 798 microsecondes

Cette valeur provient de :

- Paquet 2 (SYN-ACK) ACK RTT : 750 µs → Temps de réponse de la destination au SYN.
- Paquet 3 (ACK) ACK RTT : 48 µs → Temps nécessaire à la source pour accuser réception du message SYN-ACK.

La majeure partie de la latence (~94 %) se trouve dans le chemin de transfert (client → serveur → client), tandis que le temps de réponse de la source est minimal, ce qui indique l'absence de délai de l'UC ou de l'application sur le client.

Identification du port TCP

- Port TCP de destination : 445

Le port 445 correspond à Microsoft Server Message Block (SMB), couramment utilisé pour le partage de fichiers, les lecteurs réseau et les services d'authentification Windows. Ce protocole est sensible à la fois à la latence et au débit, ce qui le rend très dépendant de l’efficacité du protocole TCP et de la stabilité du réseau.

Analyse de la taille de fenêtre TCP

- Fenêtre source (mise à l'échelle) : 64,240 octets
- Fenêtre de destination : 65,535 octets

La fenêtre TCP représente la quantité de données pouvant être envoyées avant d'attendre l'accusé de réception. Dans ce cas, la source est légèrement plus restrictive que la destination. Ces valeurs sont relativement faibles dans les environnements modernes et peuvent limiter le débit, en particulier lorsque la RTT augmente.

Le débit théorique maximal peut être estimé à l'aide des éléments suivants :

Débit = Taille de fenêtre TCP / RTT

Remplacement des valeurs observées :

- Taille de fenêtre TCP = 64 240 octets
- RTT = 798 microsecondes = 0,000798 seconde

Débit ≈ 64 240 / 0,000798 ≈ 80,5 Mbit/s (~644 Mbit/s)

Cela représente le débit maximum en supposant :

- Aucune perte de paquets
- Aucune retransmission
- Conditions réseau idéales

Analyse du débit, du temps de transfert et des conditions requises

Avec un débit actuel de 644 Mbits/s, le transfert d'un fichier de 6,5 To prend environ 23,5 heures, ce qui correspond à la dégradation observée. Pour obtenir une fenêtre de transfert de 9 heures, le débit doit augmenter à environ 1,68 Gbit/s, nécessitant soit une fenêtre TCP plus importante (~2,7 fois plus importante), soit une valeur RTT considérablement plus faible (~291 µ).

Dans les conditions actuelles (fenêtre de 64 Ko et RTT d'environ 798 µ), il n'est pas possible d'atteindre l'objectif de 9 heures, car le débit TCP est limité par le produit de délai de bande passante. Sans augmentation de la taille de la fenêtre ou réduction de la latence, le protocole ne peut pas utiliser une bande passante disponible plus élevée, ce qui rend la cible inaccessible.

Scénario	Débit	Temps de transfert estimé (6,5 To)	Fenêtre TCP requise	RTT requis
État actuel	644 Mbit/s (~80,5 Mbit/s)	~23,5 heures	64 Ko	µs
Objectif (9 heures)	~1 683 Mbit/s (~210 Mbit/s)	9 heures	~172 Ko	~291 µ

Cela a fonctionné précédemment, indiquant qu'une modification s'est produite sur le réseau, l'application, la source ou la destination. Il est important de noter que, sur la seule base de cette analyse initiale, une conclusion importante peut déjà être établie : dans les conditions actuelles de taille de fenêtre TCP et de RTT, l’objectif de 9 heures n’est pas possible.

Les tableaux comparent les variations du débit en fonction de l’augmentation ou de la diminution de la taille des fenêtres RTT et TCP.

Impact de RTT sur le débit (taille de fenêtre fixe = 64 240 octets)

RTT	Débit (Mo/s)	Débit (Mbits/s)
200 µ (0,0002 s)	~321 Mo/s	~2 568 Mbit/s
798 µ (0,000798 s)	~80,5 Mo/s	~644 Mbit/s
2 ms (0,002 s)	~32,1 Mo/s	~257 Mbit/s
10 ms (0,01 s)	~6,4 Mo/s	~51 Mbit/s

Impact sur la taille de fenêtre TCP (RTT fixe = 798 µs)

Taille de fenêtre TCP	Débit (Mo/s)	Débit (Mbits/s)
16 Ko (16 384 Mo)	~20,5 Mo/s	~164 Mbit/s
64 Ko (64 240 Mo)	~80,5 Mo/s	~644 Mbit/s
256 Ko (262 144 Mo)	~328 Mo/s	~2 624 Mbit/s
1 Mo (1 048 576 Mo)	~1 314 Mo/s	~10,5 Gbit/s

Interprétation technique

- Le débit est inversement proportionnel à la latence plus élevée du RTT → et réduit les performances.
- Le débit est directement proportionnel à la taille des fenêtres TCP → les fenêtres plus grandes augmentent la capacité.
- Les fenêtres de petite taille limitent considérablement le débit, même dans les environnements à faible latence.
- Les réseaux haut débit (10G+) nécessitent une mise à l'échelle des fenêtres pour utiliser pleinement la bande passante.

Cela démontre que la taille de fenêtre RTT et TCP sont des facteurs critiques dans les performances TCP et doivent être analysés ensemble lors du dépannage des problèmes de débit.

Longueur d’en-tête IP et TCP

- Longueur d'en-tête IP : 20 octets
- Longueur d'en-tête TCP : 32 octets

Un en-tête IP de 20 octets indique qu'aucune option IP n'est présente. L'en-tête TCP de 32 octets confirme que les options TCP sont utilisées, ajoutant 12 octets au-delà de l'en-tête de base. Ces options incluent généralement MSS, Échelle de fenêtre et SACK autorisé.

Analyse des options TCP et durée de vie

L'accusé de réception sélectif (SACK) est activé sur les deux terminaux. Ce n'est pas visible sur l'image. La fonction SACK permet au destinataire d'accuser réception de blocs de données non contigus, informant ainsi l'expéditeur avec exactitude des segments reçus.

Par exemple, si les segments 1000-2000 et 3000-4000 sont reçus mais que 2000-3000 est manquant, le destinataire peut l'indiquer explicitement. Sans SACK, l'expéditeur retransmettrait toutes les données après l'intervalle ; avec SACK, seule la partie manquante est retransmise. Ceci améliore considérablement les performances dans les environnements avec perte de paquets.

Analyse du paquet 1 (SYN)

- Numéro de séquence : 0 (Wireshark normalisé)
- Charge utile : 0 octets
- N° ACK : 0
- MSS : 1460 octets
- TTL : 128

Wireshark normalise le numéro de séquence à zéro pour la lisibilité, bien qu'il s'agisse en pratique d'une valeur aléatoire importante. L'absence de données utiles est attendue lors de l'établissement de la connexion. La valeur MSS de 1 460 octets indique une MTU de 1 500 octets (en-tête IP de 20 octets + en-tête TCP de 20 octets). Un TTL de 128 peut être un hôte Windows, et le fait de voir cette valeur dans la capture indique que la capture a probablement été effectuée à la source ou très près de celle-ci via la couche 2.

Analyse du paquet 2 (SYN-ACK)

- N° ACK : 1

La valeur ACK est 1 car l'indicateur SYN utilise un numéro de séquence, même si aucune charge

utile n'est présente. Par conséquent, ACK = SEQ + 1.

- TTL : 59

La durée de vie observée de 59 suggère une durée de vie initiale de 64, ce qui signifie que le paquet a traversé environ 5 sauts de routage (64 − 59 = 5). Chaque saut routé décrémente la durée de vie de un.

Risque de fragmentation et impact sur le réseau

La présence d'environ cinq sauts de routage présente des risques potentiels en termes de performances, notamment en ce qui concerne les incohérences et la fragmentation des MTU.

Si une liaison intermédiaire a une MTU inférieure à la taille de paquet d'origine, une fragmentation peut se produire. Cela entraîne plusieurs conséquences :

- Latence accrue due à la fragmentation et à la surcharge de réassemblage.
- Probabilité plus élevée de perte de paquets, car la perte d'un seul fragment nécessite la retransmission de l'intégralité du paquet.
- Débit réduit, car le protocole TCP interprète la perte comme un encombrement et réduit son débit d'envoi.
- Utilisation accrue du CPU sur les périphériques réseau gérant la fragmentation.
- Risque d'échec de la détection PMTUD (Path MTU Discovery) si le protocole ICMP est bloqué, entraînant des abandons de paquets silencieux.

Compte tenu de ces facteurs, il est essentiel d'assurer une MTU cohérente sur le chemin ou de mettre en oeuvre un verrouillage MSS si nécessaire.

Analyse TCP RTT : RTT ACK et RTT initial

Lorsque ACK RTT est supérieur à iRTT, cela indique que la latence a augmenté par rapport à la ligne de base établie lors de la connexion TCP.

Cela signifie que le réseau ou les points d'extrémité introduisent un délai supplémentaire pendant la session, généralement en raison des facteurs suivants :

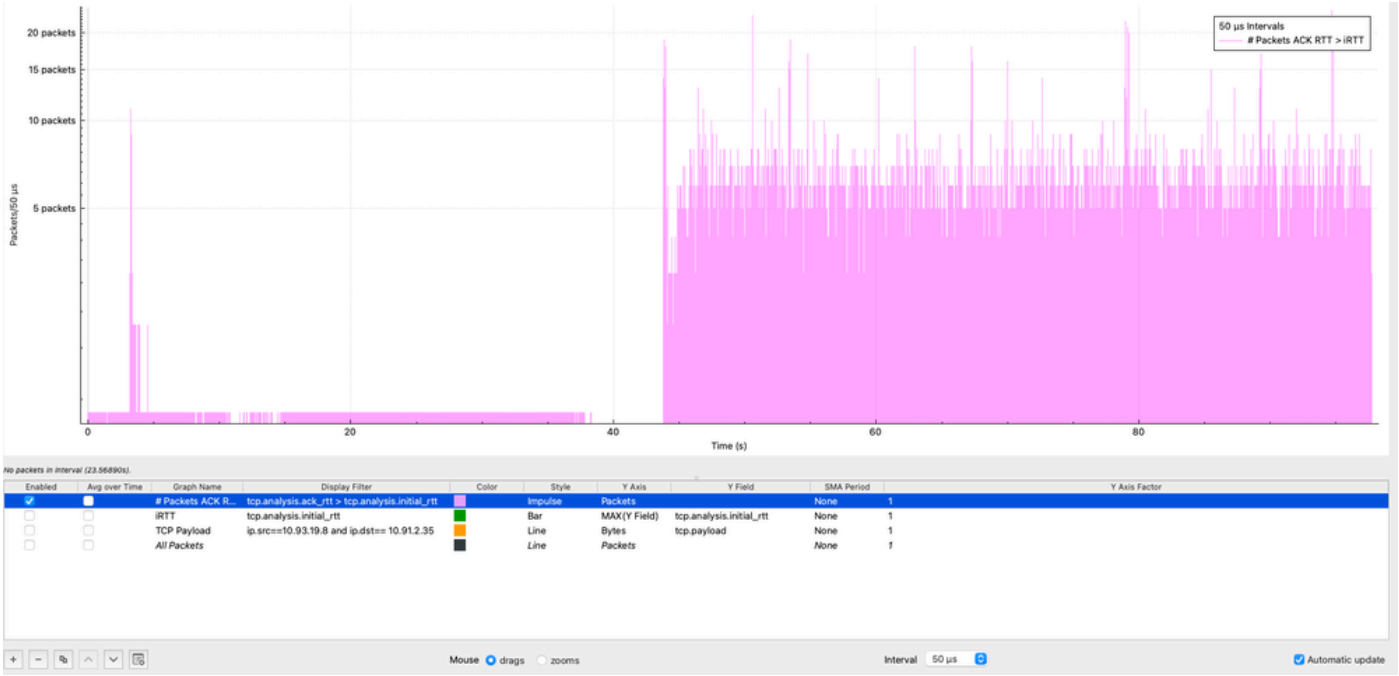
- Encombrement du réseau ou file d'attente
- Délais de traitement du récepteur ou des applications
- Périphériques intermédiaires (pare-feu, équilibreurs de charge)
- Retransmissions

Si cette condition persiste tout au long de la session TCP, elle entraîne :

- Débit TCP réduit
- Utilisation inefficace des fenêtres
- Performances applicatives dégradées

Dans Wireshark, il est possible de visualiser la fréquence à laquelle la condition ACK RTT > iRTT se produit en utilisant la fonctionnalité I/O Graphs sous : Statistiques → Graphiques d'E/S, application du filtre d'affichage (tcp.analysis.ack_rtt > tcp.analysis.initial_rtt), sélection du style Impulse, définition de l'axe Y sur Packets et utilisation d'un intervalle de 50 microsecondes.

Dans le graphique, les impulsions violettes représentent le nombre de paquets qui remplissent cette condition dans chaque intervalle de 50 microsecondes. Comme observé, cette condition persiste tout au long de la capture de paquets, indiquant que la latence pendant la session est constamment supérieure à la ligne de base initiale. Ce comportement suggère fortement une dégradation soutenue des performances plutôt qu'une condition transitoire, ce qui renforce la nécessité d'étudier les sources potentielles telles que la congestion, la mise en mémoire tampon ou les retards de traitement des points d'extrémité sur le chemin de bout en bout.



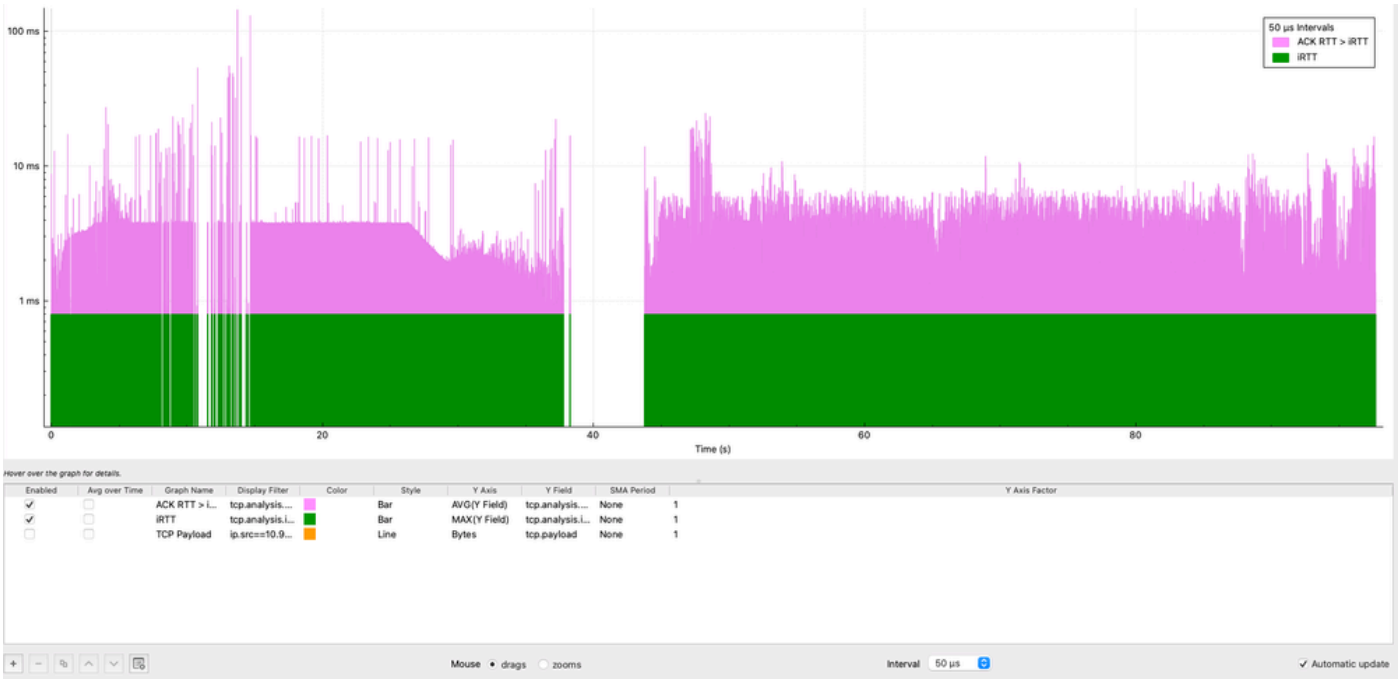
Il est également important de déterminer pendant combien de temps le délai de péremption initial est dépassé, et pas seulement à quelle fréquence. Bien que Wireshark ne permette pas directement la soustraction entre les champs, une comparaison visuelle peut être effectuée à l'aide de graphiques d'E/S :

- Accédez à Statistiques → Graphiques E/S
- Graphique 1 :

    - Filtre d'affichage tcp.analysis.ack_rtt > tcp.analysis.initial_rtt
    - Style: Barre
    - Axe Y : MOYENNE
    - Champ Y : tcp.analysis.ack_rtt
    - Intervalle : 50 microsecondes
- Graphique 2 :
    - Filtre d'affichage tcp.analysis.initial_rtt
    - Style: Barre
    - Axe Y : MAXIMUM
    - Champ Y : tcp.analysis.initial_rtt
- Cliquez ensuite avec le bouton droit sur le graphique et activez l'échelle du journal.

Dans cette visualisation, le graphique violet représente la condition ACK RTT > iRTT, qui est présente de façon constante tout au long de la session TCP. Les données montrent une inflation de latence soutenue, avec de multiples pics atteignant 11 millisecondes et un pic maximal de plus de 100 millisecondes, représentant 11x à 100x le iRTT de base.

Ce comportement confirme que l'augmentation de la latence n'est pas transitoire mais persistante, indiquant un problème systémique affectant la session au fil du temps. Cette déviation soutenue suggère fortement des facteurs tels que l'encombrement du réseau, la mise en mémoire tampon (bufferbloat) ou les retards de traitement des points d'extrémité.



Analyse des retransmissions TCP et des retransmissions erronées

Cette section évalue la fiabilité du protocole TCP en analysant les retransmissions dans le temps,

ce qui permet de vérifier si la perte de paquets contribue à la dégradation des performances.

Retransmissions TCP dans le temps

Le graphique montre la répartition des retransmissions TCP dans le temps. Au total, 42 retransmissions ont été observées, représentant seulement 0,00125 % du trafic total.

Ce niveau de retransmissions est négligeable et indique clairement que la perte de paquets n'est pas un facteur contributif dans ce scénario.

Configuration Wireshark (retransmissions TCP)

```
Statistics → I/O Graphs
```

- Filtre d'affichage

```
tcp.analysis.retransmission and !tcp.analysis.spurious_retransmission
```

- Style: Impulsion ou barre
- Axe Y : Paquets
- Intervalle : 1 s

Retransmissions TCP erronées

Le graphique montre le nombre de retransmissions TCP suspectes dans des intervalles d'une seconde générés par la source 10.93.19.8.

Dans Wireshark, une retransmission TCP erronée indique qu'un hôte a retransmis un segment qui n'a pas été réellement perdu. Le paquet d'origine a atteint le récepteur, mais l'expéditeur a supposé à tort une perte due à une estimation de synchronisation inexacte. Ce comportement n'indique pas une perte réelle de paquets, mais plutôt une logique de retransmission inefficace au niveau de l'expéditeur.

Dans cette capture :

- La source 10.93.19.8 retransmet les paquets après seulement ~8 microsecondes.

- Alors que les temporisateurs de retransmission typiques sont de l'ordre de ~200 millisecondes.

Cela confirme que le comportement de retransmission est entièrement contrôlé par la pile TCP source, et non par le réseau.

Le nombre total de retransmissions erronées observées est de 1 112, ce qui représente 0,032 % du trafic total capturé.

Configuration Wireshark (retransmissions TCP erronées)

```
Statistics → I/O Graphs
```

- Filtre d'affichage

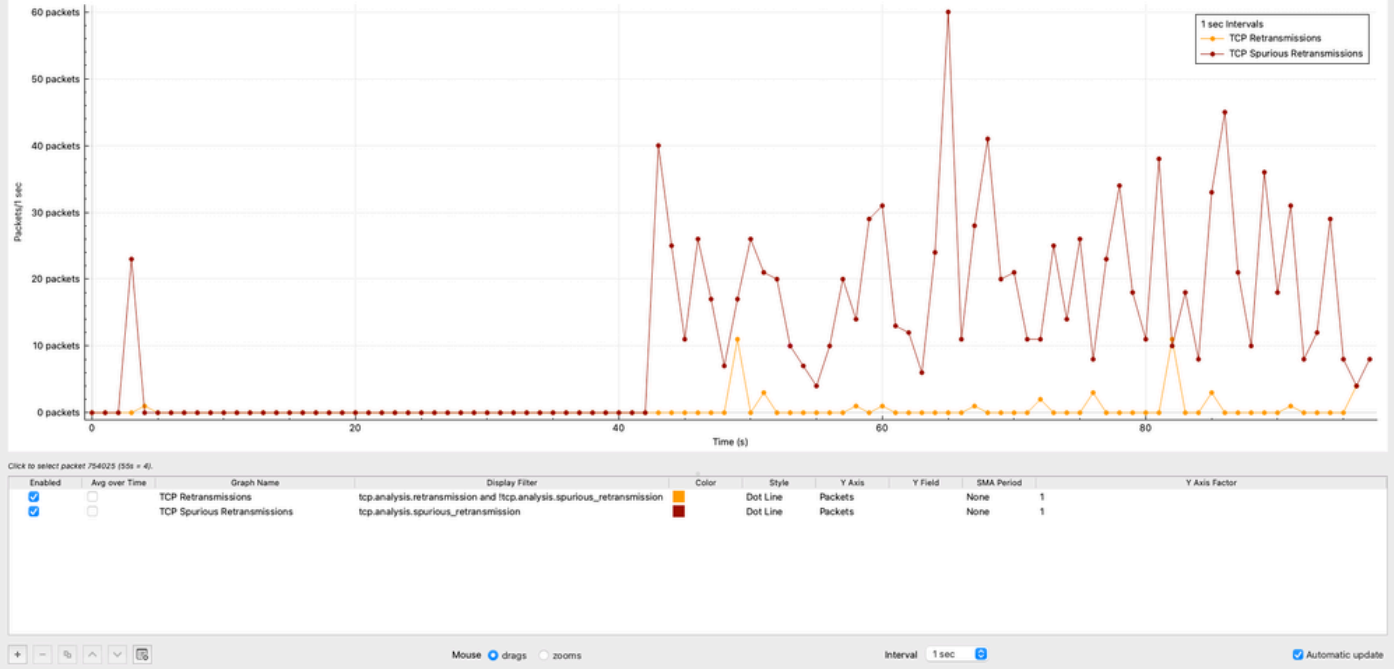
```
tcp.analysis.spurious_retransmission and ip.src==10.93.19.8
```

- Style: Impulsion ou barre
- Axe Y : Paquets
- Intervalle : 1 s

Interprétation technique

- Le pourcentage extrêmement faible de retransmissions réelles confirme que la perte de paquets n’est pas présente dans le réseau.
- La présence de retransmissions erronées indique des décisions de retransmission prématurées par l’hôte source.
- Ce comportement peut avoir un léger impact sur l'efficacité, mais il n'est pas la principale cause de dégradation importante du débit.

Cette analyse confirme que le problème n'est pas lié à la fiabilité du réseau, mais plutôt au comportement TCP, à la latence ou aux performances des terminaux.

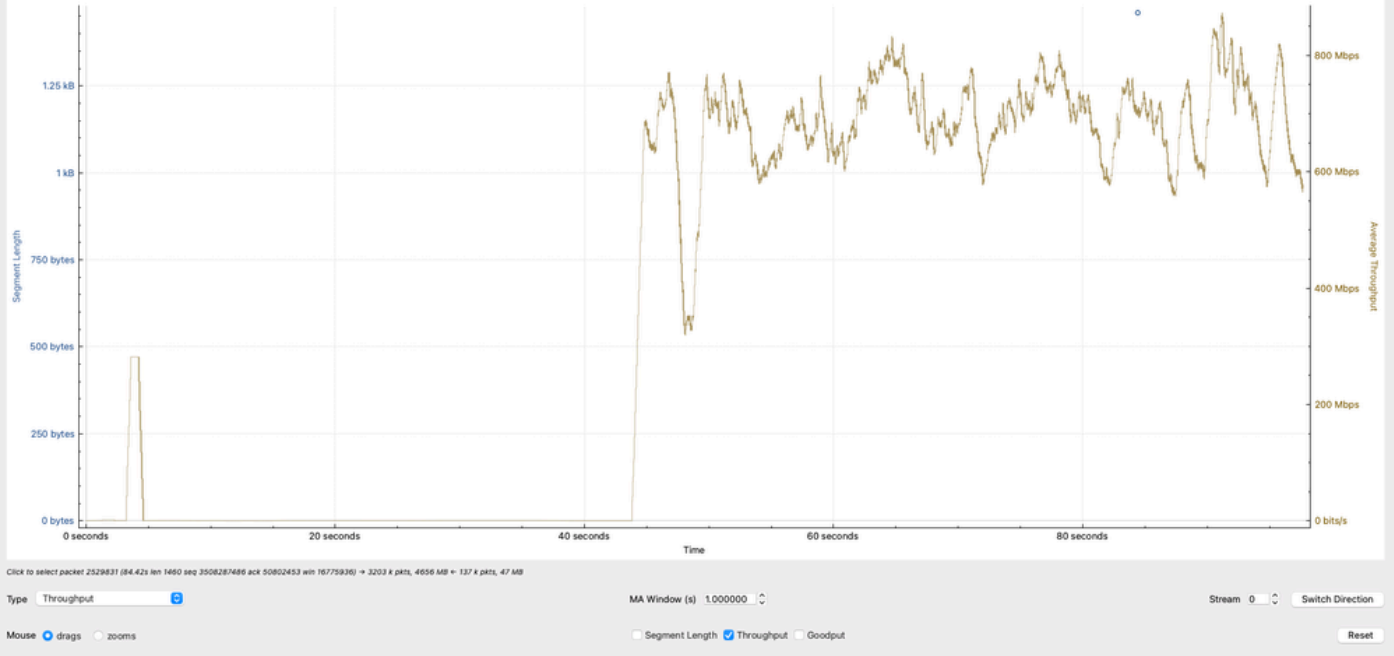


Analyse efficace du débit

Le graphique montre le débit effectif, calculé en fonction de la charge utile TCP (données réelles transférées) en mégabits par seconde. Le débit observé oscille principalement entre 600 Mbits/s et 800 Mbits/s, ce qui indique que, pendant que le réseau transfère activement des données, il n’atteint pas un potentiel de bande passante plus élevé.

Configuration Wireshark (débit effectif)

```
Statistics → TCP Streams Graphs → Throughout
```



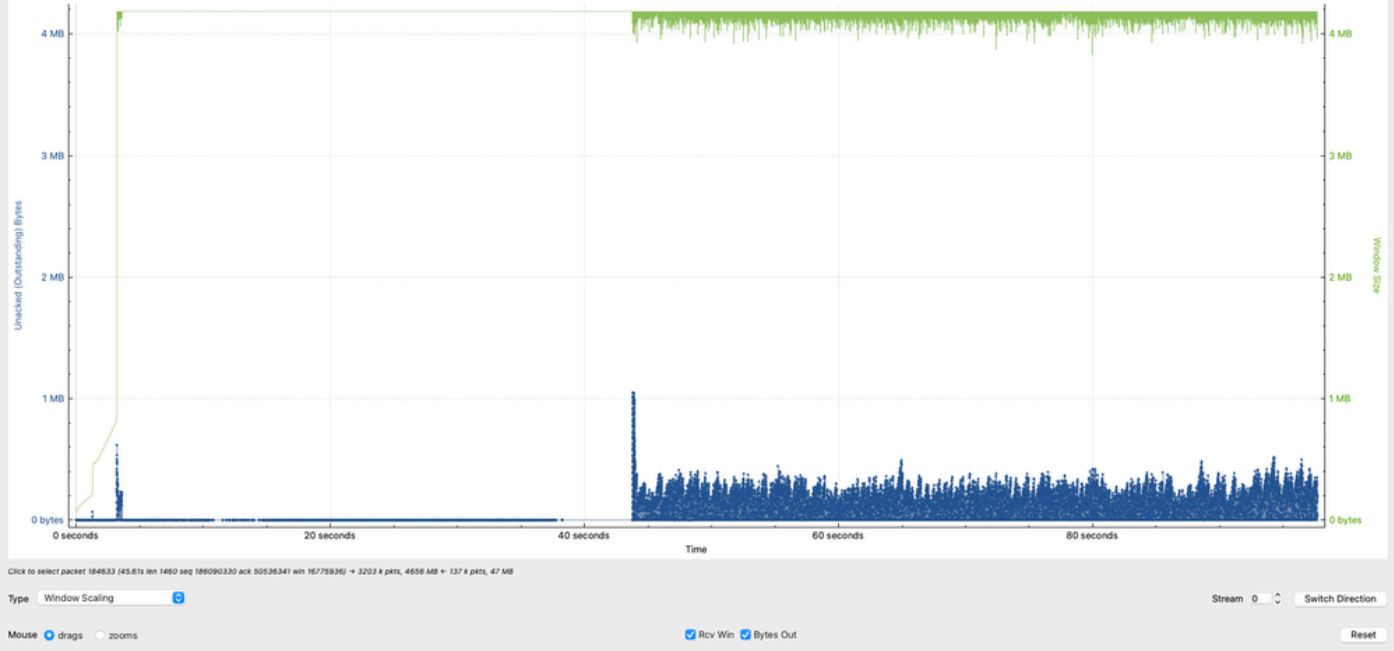
Interprétation technique

- La plage de débit de 600 à 800 Mbits/s correspond aux calculs précédents basés sur la taille de fenêtre TCP et le RTT.
- La variabilité du débit reflète :
  - Fluctuations de RTT
  - Ajustements du contrôle d’encombrement TCP
  - Synchronisation ou mise en mémoire tampon des applications
- Comme le débit ne se rapproche pas du débit de ligne (par exemple, 10 G), la limitation n'est pas la bande passante physique, mais plutôt les contraintes d'efficacité TCP.
- Cette analyse confirme que le débit observé est conforme aux limitations TCP (taille de fenêtre et latence), ce qui renforce le fait que le goulot d’étranglement n’est pas dû à la perte de paquets ou à la capacité de l’interface, mais au comportement de la couche transport et aux conditions des points d’extrémité.

Analyse des données en vol (fenêtre TCP)

Le graphique met en évidence un comportement critique dans la session TCP en comparant la capacité du récepteur aux données réelles en transit (octets en vol).

- La ligne verte représente la quantité de données TCP que 10.91.2.35 (récepteur) peut accepter (fenêtre de réception effective).
- La ligne bleue représente la quantité de données TCP actuellement en cours à partir de 10.93.19.8 (expéditeur).



Les données observées en vol atteignent des pics d'environ 1 Mo, avec des pics supplémentaires d'environ 8 Ko et 5 Ko, mais elles sont principalement concentrées entre 1 Ko et 250 Ko.

Cela indique que, bien que le récepteur soit capable de traiter des volumes de données plus importants, l'expéditeur n'utilise pas systématiquement la fenêtre disponible.

Configuration Wireshark (données en vol ou en fenêtre)

```
Statistics → TCP Streams Graphs → Throughout
```

Interprétation technique

- Le récepteur (10.91.2.35) annonce une fenêtre sensiblement plus grande, indiquant qu'il est capable de recevoir davantage de données.
- L'expéditeur (10.93.19.8) sous-utilise la fenêtre disponible, comme le montrent les valeurs inférieures et incohérentes Données en vol.
    - L'expéditeur peut idéalement conserver les valeurs Données en vol plus près de la fenêtre des destinataires annoncés (~1 Mo) pour maximiser le débit.
    - L’incapacité à maintenir des niveaux élevés de données en vol limite directement le débit et constitue un indicateur important de l’inefficacité du protocole TCP à la source, et non un problème de capacité du réseau.

Analyse de la charge utile TCP par rapport à MSS dans le temps

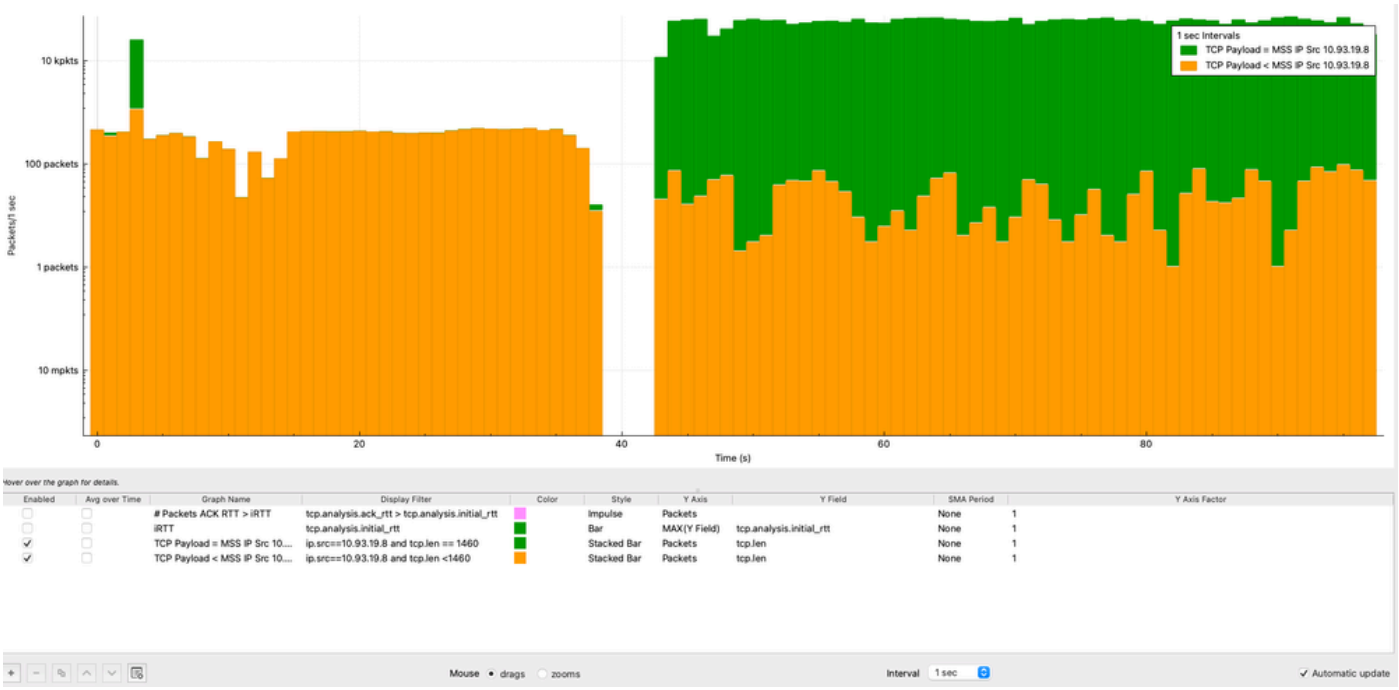
L'analyse de la taille de la charge utile TCP par rapport à MSS permet de déterminer si l'expéditeur utilise efficacement chaque segment TCP. Cette analyse est effectuée du point de vue de l’adresse IP source (10.93.19.8).

Dans Wireshark, les graphiques sont configurés comme suit :

- Graphique 1 (paquets de taille MSS) :
  - Filtre d'affichage ip.src==10.93.19.8 et tcp.len == 1460
  - Style: Barres empilées
  - Axe Y : Paquets
  - Intervalle : 1 seconde
- Graphique 2 (tous les paquets ≤ MSS) :
  - Filtre d'affichage ip.src==10.93.19.8 et tcp.len <= 1460
  - Style: Barres empilées
  - Axe Y : Paquets
  - Intervalle : 1 seconde
- Appliquer une échelle logarithmique pour une meilleure visualisation

De l'analyse :

- La majorité des paquets (>10 000 paquets par seconde) atteignent systématiquement la valeur MSS de 1 460 octets.
- Une plus petite partie des paquets transporte moins de données utiles en raison du comportement TCP normal (ACK, segmentation ou données de fin de flux).



Analyse de la cause première (RCA) : Dégradation des performances TCP

Cette analyse montre que l’identification de la cause première des problèmes de performances TCP nécessite une approche globale de bout en bout, plutôt que de supposer que le réseau est la principale source de dégradation.

Le commutateur Cisco Nexus 9300 a fait l'objet d'une validation approfondie, notamment en ce qui concerne les compteurs d'interface, les politiques QoS, la stabilité du routage et du protocole ARP, la vérification des points de CPU, la capture de paquets basée sur la fonctionnalité SPAN et la validation du transfert au niveau ASIC à l'aide d'ELAM. Tous les résultats ont confirmé de façon constante que le commutateur fonctionnait selon les paramètres prévus :

- Aucun abandon de paquet
- Aucune latence anormale (plage de microsecondes)
- Pas d'impact QoS ou plan de contrôle
- Transfert matériel correct

En outre, l'analyse TCP a révélé :

- Retransmissions négligeables (0,00125 %)
- Aucune preuve de perte de paquets
- Utilisation MSS cohérente à la source
- Débit aligné sur la fenêtre TCP et les contraintes RTT
- Sous-utilisation de la fenêtre TCP disponible (analyse des données en cours de vol)
- Le réseau n'est pas le goulot d'étranglement
- Le serveur source limite les performances

Conclusion

La dégradation des performances est due au fait que le serveur source fonctionne avec la MTU 1500 dans un environnement compatible jumbo, ce qui empêche une utilisation efficace de la capacité réseau disponible.

Solution

Augmentez la MTU sur le serveur source de 1500 à 9000 octets pour l'aligner sur l'infrastructure de destination et de réseau. Les avantages :

- Activer les segments TCP plus volumineux

- Réduire la surcharge des paquets
- Améliorer le débit global

Réflexion technique

L’un des principaux enseignements de cette analyse est l’importance d’éviter des conclusions prématurées lors du dépannage des performances du réseau. Bien qu’il soit courant d’attribuer initialement des problèmes au réseau, ce cas démontre clairement que le réseau fonctionnait correctement sur l’ensemble du chemin du plan de données. Ce n’est qu’en effectuant une analyse TCP approfondie à la fois du point de vue de la source et de la destination (y compris les paramètres de connexion, le comportement RTT, l’utilisation de la fenêtre, les retransmissions et l’efficacité de la charge utile) qu’il a été possible d’identifier avec précision le véritable goulot d’étranglement.

Prendre le temps d’analyser en détail le comportement TCP permet d’éviter les erreurs de diagnostic, de réduire les modifications inutiles du réseau et de s’assurer que les efforts de correction sont dirigés vers la cause réelle.

À propos de cette traduction
Cisco a traduit ce document en traduction automatisée vérifiée par une personne dans le cadre d'un service mondial permettant à nos utilisateurs d'obtenir le contenu d'assistance dans leur propre langue.

Il convient cependant de noter que même la meilleure traduction automatisée ne sera pas aussi précise que celle fournie par un traducteur professionnel.