Table of Contents

Contexte 4
Le problème 5
Investigation 6
1. Valider la stabilité réseau de base 6
2. Valider le tunnel WireGuard 6
3. Observer le comportement du transport RDP 7
4. Corréler les coupures de session avec les métriques système 8
Cause racine 8
Hypothèses trompeuses courantes 9
Solutions recommandées 10
1. Augmenter les ressources serveur 10
2. Réduire la charge graphique 10
Un point de départ pratique pour le dimensionnement RDS 10
Mise en perspective 12
Points à retenir 13
Conclusion 13
Articles liés 13
Tags 14

Quand ce n'est pas le réseau : une enquête RDP qui a mené ailleurs

Auteur(s) : Louis Ouellet

Dans le cadre d'un déploiement récent, on m'a demandé d'enquêter sur des sessions Remote Desktop (RDP) instables vers un serveur distant accessible au moyen d'un VPN site-à-site.

Au départ, l'explication semblait simple : le VPN était instable. Les utilisateurs étaient déconnectés, le serveur cible était distant, et tous les symptômes semblaient pointer dans la même direction.

Sur papier, l'environnement était pourtant simple :

- Réseau local : `192.168.115.0/24`
- Réseau distant : `192.168.201.0/24`
- VPN initial : IPsec
- VPN temporaire de remplacement : WireGuard
- Serveur cible : `192.168.201.100`
- Plusieurs utilisateurs se connectant en RDP

Mais comme c'est souvent le cas en infrastructure, la première explication s'est révélée être la plus pratique — pas la plus exacte.

C'est le genre de situation où tout ressemble à un problème réseau — jusqu'à ce qu'on commence à prouver ce qui fonctionne réellement, et ce qui ne fonctionne pas.

Contexte

Le problème n'a pas commencé avec WireGuard.

À l'origine, la connectivité entre les deux environnements reposait sur un tunnel IPsec. Lorsque ce tunnel a cessé de fonctionner, la première affirmation a été que le tunnel apparaissait toujours actif du côté distant, donc que le problème devait forcément se trouver de mon côté.

De mon côté, les faits racontaient une autre histoire. Les captures de paquets sur l'interface WAN montraient clairement que mon pare-feu envoyait du trafic, mais ne recevait aucune réponse. Un traceroute est allé encore plus loin : le point de terminaison IPsec distant n'était même plus joignable.

En parallèle, les utilisateurs signalaient déjà des lenteurs lorsqu'ils se connectaient à `192.168.201.100`. Ces plaintes existaient avant même la panne IPsec, ce qui compliquait la situation. Il y avait maintenant un tunnel brisé, mais aussi un problème de performance plus ancien affectant ce même serveur distant.

Même avant d'aller plus loin dans l'analyse, certains signaux laissaient déjà entrevoir un possible problème de capacité. Lors d'échanges avec l'équipe distante, il a été mentionné que l'utilisation CPU et mémoire tournait généralement autour de 80 % en conditions normales. D'après mon expérience avec les environnements RDS et VDI, cela laissait très peu de marge pour absorber les pics d'activité des utilisateurs.

Cela ne prouvait rien à lui seul, mais rendait l'hypothèse d'une saturation des ressources crédible et digne d'être validée.

Avant que le tunnel IPsec ne tombe complètement, j'avais déjà partagé quelques recommandations afin de réduire la charge du côté distant, par exemple en publiant l'application sous forme de RemoteApp ou en réduisant la charge graphique imposée au serveur. Ces suggestions n'ont pas fait avancer la situation, en grande partie parce que l'expérience semblait différente pour d'autres utilisateurs.

Une fois le tunnel IPsec devenu inutilisable, la prochaine étape proposée a été de passer à WireGuard.

Il a été suggéré de déployer WireGuard directement sur un serveur RDS, mais ce n'était pas une direction que j'étais prêt à prendre. Faire terminer un VPN directement sur un serveur de sessions de production crée des problèmes inutiles de sécurité et d'architecture réseau. Comme la mise à niveau de pfSense pour supporter WireGuard correctement aurait nécessité une fenêtre de maintenance, j'ai plutôt mis en place une machine virtuelle dédiée sous Ubuntu Server 24.04 LTS pour servir de passerelle WireGuard temporaire, puis j'ai ajouté le routage nécessaire sur pfSense.

C'est à ce moment-là que la véritable enquête a commencé.

Le problème

Une fois le chemin WireGuard en place, les utilisateurs ont pu se reconnecter, mais le problème principal demeurait.

Ils constataient :

- Des déconnexions RDP aléatoires
- Une instabilité notable lors d'une utilisation active
- Des sessions qui se reconnectaient partiellement, avec les lecteurs et imprimantes redirigés qui revenaient d'eux-mêmes
- Aucune perte de paquets évidente lors des tests de connectivité de base

Cela créait une situation classique et très trompeuse :

Le ping était stable, mais l'application ne l'était pas.

C'est exactement le genre de symptôme qui peut faire dévier une enquête dans la mauvaise direction si l'on s'arrête à la première hypothèse.

Investigation

Plutôt que de considérer le VPN comme coupable par défaut, j'ai abordé le problème couche par couche.

---

1. Valider la stabilité réseau de base

La première étape consistait à vérifier si le chemin lui-même était instable.

J'ai commencé par des tests ICMP et une validation du MTU :

```
ping 192.168.201.100 -f -l 1300
```

Après plusieurs tests itératifs, j'ai constaté qu'un MTU d'environ :

- `1380 octets`

fonctionnait sans fragmentation.

Les tests réseau de base montraient systématiquement :

- Aucune perte de paquets
- Une latence stable

Conclusion : le chemin lui-même était suffisamment stable pour transporter le trafic de manière fiable.

---

2. Valider le tunnel WireGuard

L'étape suivante consistait à vérifier la passerelle VPN elle-même.

J'ai vérifié le tunnel avec :

```
wg show
conntrack -L
iptables -L
```

Ce que j'ai constaté :

- Des handshakes stables et réguliers
- Aucune saturation de conntrack
- Aucun blocage au niveau du pare-feu expliquant le comportement
- Un trafic bidirectionnel circulant correctement à travers la passerelle

Conclusion : le tunnel WireGuard fonctionnait correctement et ne subissait pas de coupures franches.

---

3. Observer le comportement du transport RDP

Puisque le chemin de base et le VPN paraissaient tous deux sains, je suis monté d'une couche et j'ai observé le trafic RDP lui-même à l'aide de `tcpdump`.

Les captures montraient :

- Une forte utilisation d'UDP
- Un trafic en rafales lors des interactions
- Des paquets se situant généralement entre ~1000 et 1200 octets
- Un comportement de repli en TCP moins stable qu'attendu

Exemple :

```
IP 192.168.201.20.51020 > 192.168.201.100.3389: UDP, length 1237
IP 192.168.201.20.51020 > 192.168.201.100.3389: UDP, length 1005
```

À ce stade, il semblait encore possible que le problème soit lié au transport. Le motif du trafic était clairement actif, et RDP faisait un usage intensif de l'UDP, surtout lors des interactions utilisateur.

Cela ressemblait encore à un problème réseau — mais les preuves ne concordaient plus avec une perte de paquets.

---

4. Corréler les coupures de session avec les métriques système

Le point de bascule est arrivé lorsque j'ai cessé de regarder uniquement les paquets et commencé à corréler les déconnexions avec la performance du serveur.

Pour cela, j'ai surveillé le CPU et la mémoire sur `192.168.201.100` :

```
typeperf "\Processor(_Total)\% Processor Time" "\Memory\% Committed Bytes In Use"
```

Le résultat était clair :

- L'utilisation CPU montait régulièrement à `100 %` au moment exact où la session RDP devenait instable ou tombait

C'est à ce moment que l'enquête a changé de direction.

Avec le recul, les données de performance confirmaient les observations initiales : le serveur avait très peu de marge dès que l'activité utilisateur augmentait.

À partir de là, la question n'était plus de savoir si les paquets traversaient le VPN. Ils le traversaient. La vraie question était de savoir si le serveur distant pouvait suivre la charge qui lui était imposée.

Cause racine

Le problème principal n'était pas le VPN. Il s'agissait d'une saturation des ressources côté serveur, en particulier d'un épuisement du CPU sur `192.168.201.100`.

Les sessions RDP, surtout lorsque les utilisateurs travaillent activement, génèrent une quantité surprenante de charge graphique et de traitement de session. Dans ce cas précis, les déconnexions n'avaient rien d'aléatoire. Elles correspondaient aux moments où le serveur arrivait à saturation.

Cela explique pourquoi les symptômes étaient si trompeurs :

- Le tunnel restait actif
- Le ping continuait de fonctionner
- Les paquets continuaient de circuler
- Mais la session utilisateur devenait tout de même instable

Quand le CPU atteint 100 % sur l'hôte RDP :

- Le traitement des sessions prend du retard
- Les mises à jour graphiques s'accumulent
- La réactivité du transport se dégrade
- Le client finit par se comporter comme si la connectivité était défaillante

Ce qui ressemblait à un problème réseau était en réalité un goulot d'étranglement de calcul se manifestant comme un symptôme réseau.

Hypothèses trompeuses courantes

Plusieurs causes possibles ont dû être testées puis écartées tout au long du processus :

- Un tunnel VPN défaillant
- Des problèmes de MTU ou de fragmentation
- Le comportement du transport UDP par rapport à TCP
- Des limitations liées au pare-feu ou à conntrack

Toutes ces hypothèses étaient raisonnables. Elles n'étaient simplement pas la cause principale de l'instabilité.

Un ping stable ne prouve pas qu'une application est en bonne santé, et une application instable ne signifie pas automatiquement que le réseau est en cause.

Solutions recommandées

Une fois la cause racine mieux comprise, les solutions étaient relativement simples.

1. Augmenter les ressources serveur

La correction la plus directe aurait été de donner plus de marge au serveur :

- Ajouter de la capacité CPU
- S'assurer d'une mémoire suffisante
- Surveiller la charge soutenue pendant l'activité utilisateur

2. Réduire la charge graphique

La seconde option consistait à réduire la quantité de travail que le serveur doit effectuer par session.

Publier l'application sous forme de RemoteApp aiderait à :

- Réduire la quantité de rendu de bureau nécessaire
- Diminuer la surcharge graphique des sessions RDP complètes
- Améliorer la scalabilité pour plusieurs utilisateurs

D'autres mesures d'atténuation possibles incluent :

- Réduire la résolution d'affichage
- Limiter l'usage de plusieurs moniteurs
- Désactiver les effets visuels non essentiels

Un point de départ pratique pour le dimensionnement RDS

L'une des leçons de ce cas est que le dimensionnement d'un hôte de session doit être basé sur la charge de travail réelle des utilisateurs, et non simplement sur leur capacité à se connecter

techniquement.

Les recommandations de Microsoft font une distinction importante entre les charges de travail **light**, **medium**, **heavy** et **power**. En pratique, cela signifie que la bonne taille pour un hôte de session dépend moins du seul nombre d'utilisateurs que de ce que ces utilisateurs font réellement tout au long de la journée.

Le tableau ci-dessous propose une façon simplifiée de réfléchir aux profils d'utilisateurs les plus courants dans les environnements réels.

Profil utilisateur	**Comportement typique**	**Applications / activités typiques**	**Charge relative**
Léger	Multitâche limité, tâches surtout répétitives	Saisie de données, applications métier, outils en ligne de commande, pages web statiques	Faible
Moyen	Multitâche modéré, bureautique	Word, Outlook, applications web, lecture de PDF, plusieurs onglets de navigateur	Modérée
Élevé	Multitâche fréquent avec outils de communication et applications dynamiques	Outlook, Teams, Zoom, applications Office, nombreux onglets, accès à des fichiers distants	Élevée
Intensif	Charges visuelles ou de calcul exigeantes	Outils de développement, création de contenu, CAO, montage vidéo, multi-écran haute résolution	Très élevée

Dans de nombreux environnements de petites entreprises, les utilisateurs se situent souvent quelque part entre `moyen` et `élevé`, plutôt qu'au niveau `léger`, surtout lorsque le multitâche, les appels vidéo et les applications métier dans le navigateur font partie du quotidien.

Comme point de départ pratique, le guide de dimensionnement suivant est souvent plus réaliste pour les déploiements RDS multi-session :

Profil utilisateur	**Point de départ suggéré**	**Densité utilisateur approximative**	**Notes**
Léger	8 vCPU / 16 Go RAM	Jusqu'à 6 utilisateurs par vCPU	Convient à des tâches de base avec peu de multitâche
Moyen	8 vCPU / 16 Go RAM	Jusqu'à 4 utilisateurs par vCPU	Mieux adapté à la bureautique et au multitâche modéré
Élevé	Minimum 8 vCPU / 16 Go RAM, souvent davantage en pratique	Jusqu'à 2 utilisateurs par vCPU	Plus approprié pour les utilisateurs de Teams, Zoom, de nombreuses applications et du multitâche actif
Intensif	16+ vCPU / 56 Go+ RAM, parfois avec GPU	Environ 1 utilisateur par vCPU ou moins	Idéal pour les charges graphiques, haute résolution ou fortement orientées calcul

Ces chiffres doivent être considérés uniquement comme un point de départ. La capacité réelle dépend du comportement des utilisateurs, des habitudes de connexion, des performances de stockage, des conditions réseau et de la marge restante pour absorber les pointes d'activité.

Autrement dit, si un hôte de session tourne déjà autour de 80 % d'utilisation dans des conditions normales, il a probablement très peu de tolérance pour la demande en pointe.

Mise en perspective

Du point de vue du dépannage, l'enquête conduisait à une conclusion assez claire : le goulot d'étranglement se trouvait sur le serveur distant.

Les deux réponses pratiques étaient :

- Augmenter les ressources disponibles sur `192.168.201.100`
- Réduire la charge transmise au moyen de sessions RDP complètes

Cependant, ce n'est pas la direction qui a finalement été retenue.

À la place, la réponse choisie a été de déployer des postes clients supplémentaires et d'établir des tunnels distincts pour chaque utilisateur.

Cette distinction est importante, car elle met en lumière un décalage fréquent entre le diagnostic et sa mise en œuvre :

- L'enquête pointait vers un problème de calcul
- La réponse choisie s'est concentrée sur la topologie réseau

Multiplier les tunnels ne résout pas un goulot d'étranglement lié aux ressources du serveur.

Cela ne rend pas l'enquête moins utile. Au contraire, cela rappelle une réalité importante du travail en infrastructure : trouver la cause racine et faire corriger la cause racine ne sont pas toujours la même chose.

Points à retenir

- Ne supposez pas que le réseau est en cause simplement parce que le service est distant
- Validez chaque couche indépendamment avant de passer à la suivante
- Un tunnel VPN en bonne santé ne garantit pas une session RDP stable
- Une instabilité applicative peut être causée par un épuisement des ressources côté serveur, même lorsque la connectivité semble normale
- Une analyse de cause racine n'est réellement utile que si la solution choisie correspond à la couche où le problème se situe réellement

---

Conclusion

Ce cas a été un rappel utile que les problèmes d'infrastructure sont souvent façonnés autant par les hypothèses que par la technologie elle-même.

Ce qui avait commencé comme une enquête sur un « VPN instable » s'est transformé en une analyse plus approfondie du comportement des sessions, des flux réseau, des transports utilisés, puis finalement des performances côté serveur. Au terme du processus, le chemin lui-même avait été validé, la passerelle WireGuard temporaire avait été prouvée fonctionnelle, et le véritable goulot d'étranglement avait été identifié ailleurs.

Toutes les enquêtes ne se terminent pas avec la solution que l'on aurait choisie. Mais il reste toujours utile de faire le travail correctement, de documenter les preuves, et de comprendre où réside réellement le problème.

Articles liés

- Comment Ajouter un Nouveau Serveur à un Domaine Existants
- Comment configurer des sauvegardes sur Windows Server 2022
- Comment connecter une cible iSCSI sur Windows Server 2022
- Comment désactiver l'application Cortana
- Comment désactiver la télémétrie sur Windows 11

Tous les articles liés

Tags

general, blog, rdp, network, wireguard, investigation, security, windows, linux
View the discussion thread.

From:
https://laswitchtech.com/ - **LaswitchTech**

Permanent link:
https://laswitchtech.com/fr/blog/2026/03/22/when-it-s-not-the-network-an-rdp-investigation-that-led-elsewhere

Last update: **2026/03/23 18:26**