

# Contenu

[Introduction](#)

[Conditions préalables](#)

[Conditions requises](#)

[Composants utilisés](#)

[Conventions](#)

[Configuration de cache](#)

[Commutateur de la couche 2 contre le commutateur d'écoulement](#)

[Acheminement](#)

[Services](#)

[Layer4 contre la couche 5](#)

[Asymétrie](#)

[Équilibre](#)

[Basculement](#)

[Contournement EQL](#)

[Contournement de cache](#)

[Caractéristique de contournement de paramètres URL](#)

[Pré-chercher la caractéristique](#)

[Informations connexes](#)

## **[Introduction](#)**

Ce document est une instruction pour déterminer la mise en cache et les considérations de compatibilité en mettant en application des services de contenu de Cisco commute (CSS) 11000/11500.

## **[Conditions préalables](#)**

### **[Conditions requises](#)**

Les lecteurs de ce document doivent avoir une bonne connaissance de ce qui suit :

- La passerelle par défaut du cache doit être le CSS.

### **[Composants utilisés](#)**

Ces informations s'appliquent à tout le Cisco CSS (11000 et 11500) exécutant la version de logiciel 3.0x ou ultérieures de WebNS.

Les informations contenues dans ce document ont été créées à partir des périphériques d'un environnement de laboratoire spécifique. Tous les périphériques utilisés dans ce document ont démarré avec une configuration effacée (par défaut). Si votre réseau est opérationnel, assurez-vous que vous comprenez l'effet potentiel de toute commande.

## Conventions

Pour plus d'informations sur les conventions des documents, référez-vous aux [Conventions utilisées pour les conseils techniques de Cisco](#).

## Configuration de cache

### Commutateur de la couche 2 contre le commutateur d'écoulement

Le CSS est un commutateur d'écoulement, par opposition à un commutateur de la couche 2 (L2). Un commutateur d'écoulement trace des écoulements aux ports basés sur les informations IP, par opposition aux informations de couche de MAC comme dans un commutateur L2. La fonctionnalité de commutateur d'écoulement est ce qui permet au CSS pour être intelligente satisfait et regarder le TCP synchronisez/paquet de début (synchronisation) ou le HTTP OBTIENNENT et prennent des décisions satisfaites, par opposition à expédier juste des trames. Dans des environnements de mise en cache, le commutateur d'écoulement exige que vous conduisez tous les paquets provenant le cache destiné aux clients.

### Acheminement

Puisque le CSS doit être la passerelle par défaut pour le cache, et conduit tous les paquets du cache, le CSS doit avoir une table de routage complète.

### Services

Quand vous configurez un service comme cache transparent de type, le CSS réécrit l'adresse MAC de destination avec le MAC du cache et en avant les paquets IP dans le cache afin de préserver l'adresse IP de destination. Si le cache ne peut pas écouter confusément tout le trafic du port 80, alors vous devez configurer le cache pendant que proxy-cache de type ; cependant, les services de proxy-cache ne peuvent pas utiliser la méthode de Basculement de contournement. Puisque vous MAC expédiez les paquets dans le cache, le cache devrait être directement connecté dans la case. Si ceci ne peut pas faire, alors il doit être connecté par un périphérique L2 qui ne peut pas être partagé avec le client ou la connexion Internet.

Quand un service est configuré car proxy-cache ou cache transparent, il y a une liste d'accès invisible automatique (ACL) qui est créée qui permet le source ip du contournement de cache toutes les règles de contenu de sorte qu'il puisse disparaître obteniez les pages du serveur d'origine. Commençant dans la version 4.0 de WebNS, vous ne pouvez sélectionner l'**aucune** commande de cache-**contournement** pour des services de cache transparent et de proxy-cache afin de désactiver le contournement automatique des règles de contenu.

Vous ne pouvez pas avoir deux CSS vivant, chacun avec un cache directement relié, équilibrez la charge les deux caches une fois configuré comme cache transparent de type, à moins que chaque cache ait une connexion à chaque commutateur directement. Un commutateur obtiendra la demande du client et puis choisira peut-être de servir la demande dans le cache connecté à l'autre commutateur, qui alors verrait la demande entrée et l'apparierait contre sa règle et peut-être pour décider de la servir dans le cache connecté au premier commutateur. Puisque ces paquets sont MAC expédié, il n'y a aucune manière de construire un ACL pour faire ceci correctement.

## Layer4 contre la couche 5

Une règle est considérée la couche 5 (L5) quand les informations requises pour apparier la règle ou pour prendre la décision d'Équilibrage de charge ne sont pas dans la synchronisation de TCP, mais est dans le HTTP OBTIENNENT ; donc, quand vous ajoutez un URL, une méthode d'équilibre de domaine, d'informations parasites de domaine, ou d'informations parasites URL, la règle est automatiquement L5 et induit la mystification. La mystification est un avantage même lorsque pas une condition requise parce qu'elle protège le cache contre des attaques par inondation SYN. Une configuration L5 permet pour chacun des deux la liste de qualificatif d'extension (EQL) et le contournement de Basculement, qui fait charrier la case une connexion avec le serveur d'origine, par opposition au cache. Quand une règle de contenu est considérée L5, ceci signifie que le CSS charrie les connexions avec les clients.

## Asymétrie

S'il y a une route de retour vers les clients du serveur d'origine qui ne traverse pas le CSS, alors vous avez l'asymétrie, autrement connue sous le nom de triangulation. Si vous utilisez les règles L5 et il y a un contournement, la case charrie une connexion avec le serveur d'origine, et le chemin de retour va directement au client, où il alors est très rapidement lâché puisque le client n'identifie pas le numéro de séquence. Vous devez réparer votre asymétrie, envoyer tout dans le cache et utiliser le Basculement Linéaire, ou utiliser le Traduction d'adresses de réseau (NAT) scrutant, par NATing vos clients IPS avec des groupes sources.

## Équilibre

Il est recommandé que vous utilisez la méthode d'équilibre d'informations parasites de domaine et d'informations parasites URL. Les informations parasites de domaine font des informations parasites à travers la balise entière d'hôte et puis déterminent quel serveur si les services qui domaine, tenant compte d'un chargement également distribué. La méthode d'équilibre d'informations parasites URL fonctionne pareillement à la méthode d'informations parasites de domaine, mais utilise l'URL comme le point d'émission de données par opposition à la balise d'hôte.

À l'aide d'un proxy-cache de type de service, nous équilibrons basé sur le premier OBTENONS d'une connexion persistante, et tant que la deuxième OBTIENNENT des correspondances la même règle, nous ne rééquilibrons pas, qui pourraient entraîner pour que quelques sites soient reproduits à travers les caches (référez-vous à la section pré-cherchante ci-dessous). Dans WebNS 4.0, vous pouvez configurer le CSS pour casser la connexion persistante et pour la rééquilibrer en n'émettant l'**aucune** commande **persistante** sur la règle de contenu. Toutes les fois que vous n'émettez l'**aucune** commande **persistante**, vous voulez utiliser la **remise de persistance de WebNS 4.0 remap** la commande aussi bien. La commande de **remise de persistance** est utilisée de déterminer comment des connexions persistantes devraient être cassées. Le par défaut est la remise de persistance réorientent, qui fait envoyer le CSS une Réinitialisation TCP et un redirect to du HTTP 302 le client, forçant le client pour établir une nouvelle connexion. L'Internet Explorer (IE) que 5.0 a des problèmes connus avec la réception du multiple réorienté de nouveau au même domaine. La remise de persistance remap des conserve la connexion client et déplace la connexion sur le postérieur d'un serveur à l'autre.

## Basculement

Quand une règle de contenu se compose de services qui sont utiles le type de cache transparent

et une des méthodes d'équilibre de mise en cache, alors la méthode de contournement de Basculement peut être utilisée pour tenir compte des demandes destinées dans un cache spécifique à expédier en fonction au serveur d'origine si le cache descend. Référez-vous à la question d'asymétrie ci-dessus.

## [Contournement EQL](#)

Le CSS peut être configuré avec une liste d'extensions de fichier pour envoyer dans le cache. Cette liste s'appelle un EQL. Si vous ajoutez un EQL à la ligne URL dans une règle de contenu, alors la règle apparie seulement des demandes avec les extensions de fichier qui sont dans la liste. Le CSS peut également être configuré pour sauter un EQL en changeant l'application sur la règle de contenu de sauter. Dans WebNS 4.0, si vous voulez que le CSS serve chaque demande dans une connexion persistante à la destination appropriée basée sur un EQL, vous devez n'émettre l'**aucune** commande **persistante** sur la règle de contenu. Une fois que la règle de contenu a été sautée, si vous voulez le prochain obtenez dans la connexion persistante à envoyer dans le cache s'il est cacheable, alors vous devez émettre la commande globale de **débronnement de persistance de contournement de WebNS 4.0**. Toutes les fois que vous désactivez la persistance, vous voulez émettre les 4.0 que la **remise de persistance remap** la commande globale d'éviter des questions avec IE 5.0.

## [Contournement de cache](#)

Quelques caches ont la capacité de configurer une liste de sites qui devraient être sautés par le cache. Le CSS est incompatible avec la fonctionnalité de contournement de la plupart des caches. Le CSS vous permet aux listes de qualificatif de configurer network (NQLs), qui te permettent pour configurer une liste d'adresse IP ou de réseaux que vous pouvez ajouter à une clause simple dans un ACL.

## [Caractéristique de contournement de paramètres URL](#)

Le CSS peut être configuré pour sauter automatiquement n'importe quel URLs qui ont des paramètres. Ce sont des demandes qu'un cache peut ne jamais satisfaire (par exemple, des valeurs de bourse).

## [Pré-chercher la caractéristique](#)

En fournissant un service de cache aux clients, la variable qui crée le plus grand retard à temps le temps de réponse est quand le cache ne contient pas la page étant demandée, parce qu'elle alors doit aller la récupérer. Le but des caches d'Équilibrage de charge est d'augmenter la probabilité d'une présence dans l'antémémoire. Quelques caches ont une caractéristique appelée pré-chercher, qui permet au cache pour analyser les données renvoyées de l'initiale obtient la demande et va proactivement obtient des objets à cette page avant que le client les demande explicitement. Parfois ceci peut entraîner l'utilisation supplémentaire du réseau fédérateur Internet si le client devaient arrêter la page du chargement, mais c'est probablement une quantité négligeable de trafic.

Si vous utilisez la mise en cache transparente, alors la caractéristique de contournement EQL sur le CSS casse l'algorithme pré-cherchant parce que, par défaut, l'EQL n'inclut pas la page d'accueil, qui est simplement un OBTENIR pour « / . » Sans page d'accueil principale pour un site, le cache n'a pas la page principale à fonctionner hors fonction de pour pré-chercher. Deuxièmement, si nous n'envoyons pas des demandes dynamiques dans le cache, il y aura des

demandes en double allant au serveur d'origine, à un du CSS et à un du cache. Le client doit décider ce qu'est leur but, et puis déterminer quelle fonctionnalité de chaque produit elles veulent combiner

Aux sites où de plusieurs caches sont déployés, l'équilibreur de charge devrait essayer de faire une tentative de meilleur effort d'assurer une présence dans l'antémémoire. Le CSS a plusieurs algorithmes d'Équilibrage de charge qui ont été conçus pour optimiser le nombre de présences dans l'antémémoire. Puisque le commutateur est un commutateur L5, il peut regarder les informations dans le HTTP OBTIENNENT que d'autres équilibreurs de charge ne peuvent pas. Par exemple, nous pouvons déterminer quel cache envoyer une demande a basé sur le nom de domaine ou l'URL étant demandé. Si vous deviez utiliser l'algorithme de hachage de domaine, toutes les demandes du contenu qui a eu les balises identiques d'hôte seraient envoyées dans le même cache, donc augmentant la probabilité que le contenu résiderait sur ce cache. En utilisant pré-chercher avec des caches transparents, le cache va disparaître obtiennent la demande avant le commutateur lui expédiant la demande, qui peut entraîner pour que le cache faux disparaisse obtient la page. Dans ce cas, vous pouvez vouloir désactiver pré-chercher.

En discutant des caches de proxy, l'en-tête IP ne fournit aucune information véritablement valide pour l'Équilibrage de charge en quelque sorte pour prévoir optimiser le nombre de présences dans l'antémémoire ; l'adresse IP de destination ne change pas. Les seules informations à continuer seraient l'adresse IP source, qui ne signifient vraiment rien quand il s'agit de surfer l'Internet. Le commutateur peut encore visualiser la balise d'hôte et les informations URL pour prendre ces décisions. Puisque les caches de proxy mettent à jour typiquement une connexion persistante avec le client, le commutateur fait seulement la décision fondée d'Équilibrage de charge sur le premier HTTP OBTENIR, et tout le GETS ultérieur vont dans le même cache. Puisque nous ne cassons pas la connexion persistante, ceci fonctionne bien avec la configuration pré-cherchante pour tout ultérieur OBTIENT sur la même connexion. Commençant dans la version 4.0 de WebNS, vous pouvez configurer le CSS pour casser ces connexions si vous désirez ainsi en n'émettant l'**aucune** commande **persistante**.

## [Informations connexes](#)

- [Support matériel de Commutateurs de services satisfaits de gamme 11000 CSS](#)
- [Support de produit de Commutateurs de services satisfaits de gamme 11500 CSS](#)
- [Support de logiciel de services réseau de Web de Cisco](#)
- [Logiciel du téléchargement CSS 11000](#)
- [Logiciel du téléchargement CSS 11500](#)
- [Support technique - Cisco Systems](#)