https://obonaventure.github.io/cnp3blog/ipmininet/
http://csie.nqu.edu.tw/smallko/sdn/ipv6_test.htm
http://windysdn.blogspot.com/2013/10/mininet-script-for-multiple-controllers.html
https://wiki.opendaylight.org/view/OpenDaylight_OpenFlow_Plugin::Mininet_with_multiple_controllers
https://gist.github.com/devvesa/5332005

Schaller Thomas

HTTP2

Une transition vers http2 est en cours, cette nouvelle version du protocole http apporte avec elle plusieurs modification dont une courte description se trouve ci-dessous, dans une deuxième partie nous étudierons plus en profondeur le multiplexage qui a été implémenté dans cette nouvelle version.

Partie 1 : Les innovations d'http2

La nouvelle version d'HTTP est tout d'abord plus fiable. La **transmission des données en binaire** est source de moins d'erreurs que la transmission en mode texte de la version 1.1.

De plus, HTTP/2 permet de réduire le volume de données échangé (et donc d'augmenté le débit utile) avec la **compression des en-têtes HTTP**.

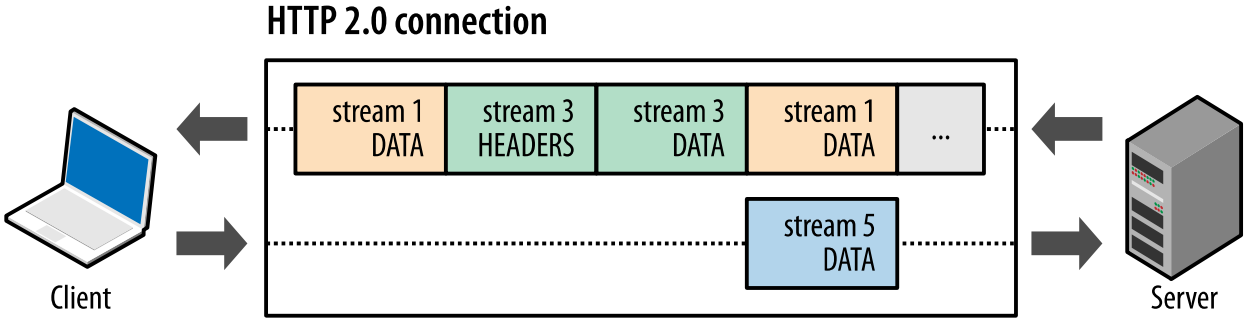
L'ajout d'un mécanisme appelé HTTP/2 Server **Push** permet d'attacher des ressources aux requêtes. En "préchargeant" des fichiers non sollicitées par le navigateur, ce qui permet des sites de plus en plus réactifs.

Pour l'affichage d'une page, HTTP/1.1 demande aux navigateurs d'effectuer pour chaque ressources une requête. Cela multiplie les échanges et ralentit considérablement le temps de chargement d'une page. L'ajout du **multiplexage** à HTTP/2 résout ce problème: il consiste a permettre au navigateur de télécharger avec une seule requête toutes les ressources nécessaires à l'affichage de la page.

De plus sur HTTP/1.1 un travaiil pour le rendre **compatible** avait déjà été éffectué. la mise en place dès-alors d'HTTP/2 est donc possible, La mise en place d'un serveur web implémentant HTTP/2 est dès maintenant imperceptible pour les visiteurs utilisant un navigateur "HTTP/1.1".

Partie 2 : LE MULTIPLEXAGE

Le multiplexage est une méthode d' HTTP / 2 qui permet d'envoyer plusieurs demandes HTTP et de recevoir des réponses de manière asynchrone via une seule connexion TCP.



Le snapshot capture plusieurs flux en vol au sein de la même connexion. Le client transmet au serveur une trame DATA (flux 5), tandis que le serveur transmet au client une séquence de trames entrelacées pour les flux 1 et 3. Il en résulte trois flux parallèles en vol.

La possibilité de décomposer un message HTTP en trames indépendantes, de les entrelacer, puis de les réassembler à l'autre extrémité constitue l'amélioration la plus importante de HTTP / 2. En fait cela offre de grandes améliorations de performances en nous permettant par exemple de:

-faire plusieurs requêtes en parallèles sans en bloquer aucune.

-faire plusieurs réponses en parallèle sans bloquer aucune de ces réponses.

-Utilisez une seule connexion pour envoyer plusieurs demandes et réponses en parallèle.

-Supprimez les solutions de contournement inutiles HTTP / 1.x (voir Optimisation pour HTTP / 1.x, telles que les fichiers concaténés, les images-objets et le partage de domaine).

-Réduire les temps de chargement des pages en éliminant les temps de latence inutiles et en optimisant l'utilisation de la capacité réseau disponible.

La nouvelle couche de cadrage binaire dans HTTP / 2 résout le problème de blocage en tête de ligne détecté dans HTTP / 1.x et élimine le besoin de connexions multiples pour permettre le traitement en parallèle des requêtes et réponses. Cela rend nos applications plus rapides, plus simples et moins coûteuses à déployer.

Coupler a ce système, http/2 implémente une priorisation des flux : Chaque flux peut recevoir un poids entier compris entre 1 et 256, ainsi qu'une dépendance explicite à un autre flux peut être attribuée à chaque flux

La combinaison de ces deux valeurs permet au client de "hiérarchiser une arborescence" qui exprime la manière dont il préférerait recevoir des réponses. À son tour, le serveur utilise ces informations pour donner la priorité au traitement de flux en contrôlant l'allocation de ressources processeur, mémoire et autres, et alloue la bande passante pour garantir une livraison optimale des réponses de haute priorité au client. Mieux encore, le protocole HTTP / 2 permet également au client de mettre à jour ces préférences à tout moment, ce qui permet d'optimiser davantage le navigateur. En d'autres termes, nous pouvons modifier les dépendances et réaffecter les pondérations en réponse à l'interaction de l'utilisateur et à d'autres signaux.

https://www.amenschool.fr/http2-optimiser-son-site-web/

https://developers.google.com/web/fundamentals/performance/http2/