

Léa THIBOUT
Anne-Gaëlle SCHALL
3ème année ingénieur informatique à l'EISTI

8 février 2019

PFE - Liste de suivi d'animés

PFE
Liste de suivi d'animés

Professeurs :
Florent DEVIN
Yannick LE NIR

Table des matières

1	Problématique	3
2	Résumés des références	3
3	Plan de travail	6
4	Objectif de réalisation	6
5	Technologies envisagées	7
6	Pitch	7

1 Problématique

Comment mettre une base de données à jour automatiquement ainsi que détecter le visionnage d'un flux vidéo sur mobile ?

Sujet : Une application permettant un suivi de séries animées et se mettant à jour automatiquement (BDD et lors du visionnage d'un épisode sur mobile via le navigateur).

Notre but est de créer une application mobile Android qui permet d'avoir un suivi en ce qui concerne les séries animées (japonaises) que l'utilisateur regarde. L'application posséderait donc une grande base de données de séries animées qui sera mise à jour régulièrement automatiquement en fonction des sorties récentes.

Comme la mise à jour du visionnage des épisodes doit être automatique, l'application devra détecter lorsque l'utilisateur regarde un flux vidéo. Celle-ci récupèrera donc les données de ce flux pour mettre sa liste d'animés à jour dans l'application.

Nous nous restreindrons à la détection de vidéos sous le navigateur Firefox pour mobile.

2 Résumés des références

Afin de répondre à notre problématique nous avons effectué des recherches sur la détection automatique de flux vidéo sur mobile. Voici les résumés des recherches les plus appropriées à notre problématique :

2008 - Automatic detection and segmentation of music videos in an audio video stream :

Un "analyseur de vidéoclips" détecte et segmente automatiquement les vidéos dans un flux multimédia audio-vidéo combiné. La détection et la segmentation automatiques sont obtenues en intégrant la détection des limites de prise de vue, la détection de texte vidéo et l'analyse audio pour détecter automatiquement les limites temporelles de chaque vidéoclips dans le flux multimédia. Les informations d'identification de chanson comme un nom de chanson, un nom d'artiste, un nom d'album, etc., est automatiquement extrait du flux multimédia à l'aide d'un "optical character recognition" (OCR) de vidéo. Ces informations sont ensuite utilisées pour cataloguer, indexer et sélectionner des vidéos musicales particulières, et pour maintenir des statistiques telles que les heures de lecture de vidéos musicales particulières et le nombre de fois que chaque vidéo musicale a été lue.

2011 - Network Characteristics of Video Streaming Traffic :

Ce papier présente les caractéristiques du réseau de deux services de vidéo streaming populaire : Netflix et Youtube.

Les auteurs présentent 3 stratégies de streaming, qui dépendent de l'application (navigateurs web, applications mobiles) et du conteneur (Flash, HTML5, Silverlight) utilisé pour streamer la vidéo. Ils ont aussi réalisé un modèle mathématiques pour évaluer l'impact des stratégies de streaming sur les propriétés stochastiques (aléatoires) de l'ensemble du trafic vidéo.

Le protocole TCP est utilisé pour le streaming vidéo mais parfois les caractéristiques du trafic ne sont pas un flux standard TCP (notamment quand c'est l'application qui contrôle le taux de flux). Les serveurs de streaming envoient des données auxiliaires, en dehors du contenu de la vidéo, pour des détails de la vidéo associée ou de la pub. . .

Pour Youtube :

- conteneur par défaut pour PC : Flash
- conteneur par défaut pour les application native mobile : HTML5
- conteneur par défaut pour les vidéos HD : Flash

Pour Netflix :

- unique conteneur : Microsoft Silverlight

Au cours d'une session de streaming typique, le contenu vidéo est transféré en deux phases :

- une phase de "mise en mémoire tampon" (les données transférées sont limités par la bande passante)
- suivie d'une phase en "régime permanent" (le taux de téléchargement est modérément plus grand que le taux d'encodage de la vidéo)

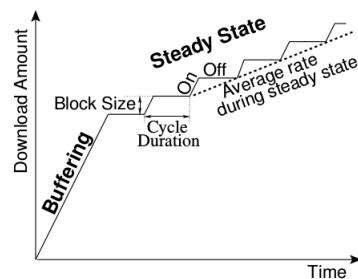


Figure 1: Phases of video download. *Video streaming begins with a buffering phase followed by a steady state phase. Cycles of ON-OFF periods in the steady state phase are used to limit the download rate.*

Types de stratégies :

- Pas de cycles On-Off (buffering phase only)
- Petits cycles On-Off (<2.5MB)
- Grands cycles On-Off (>2.5MB)

Infos intéressantes : HTML5 supporte les vidéos sans avoir besoin d'un quelconque plugin. Pour le moment il y a une adoption massive pour l'HTML5 plutôt que Flash (voir page 1 petit 3). Notamment avec l'augmentation de l'utilisation des mobiles (qui utilisent l'HTML5).

2012 - A QoE based performance study of mobile peer-to-peer live video streaming :

Les réseaux mobiles "Ad Hoc" (MANET) poste à poste (P2P) sont largement considérés comme une plate-forme pratique pour les applications mobiles de streaming vidéo en direct (par exemple, IPTV mobile). Cependant, les performances d'une telle solution de streaming sont encore largement inconnues. C'est pourquoi, dans ce document, les auteurs visent à quantifier les émissions en utilisant une approche basée sur la qualité d'expérience (QoE). Leurs résultats de simulation indiquent que la performance du streaming vidéo est très sensible à la taille du bloc vidéo. Plus précisément, si la taille des morceaux est petite, la performance, en termes de QoE et de QoS, est garantie, mais au détriment d'une augmentation des frais généraux. D'autre part, si la taille des morceaux est augmentée, les performances peuvent se dégrader assez rapidement. Il est donc nécessaire d'affiner soigneusement la taille des morceaux pour obtenir des performances satisfaisantes en matière de qualité d'expérience.

2012 - Quality Adaptation in P2P Video Streaming Based on Objective QoE Metrics :

Aujourd'hui, la transmission de données vidéo constitue une part importante du trafic sur Internet. Internet étant un environnement très dynamique, l'adaptation de la qualité est essentielle pour faire correspondre les ressources de l'appareil utilisateur à la qualité de la vidéo en continu. Ceci peut être réalisé en appliquant des mécanismes qui suivent la norme SVC (Scalable Video Coding), qui permet l'évolutivité de la qualité vidéo dans de multiples dimensions. Dans le streaming basé sur le SVC, les décisions d'adaptation ont longtemps été guidées par des indicateurs de qualité de service (QoS), tels que le débit. Cependant, ces mesures ne correspondent pas à la façon dont les utilisateurs humains perçoivent la qualité vidéo. Par conséquent, dans cet article, l'approche classique de streaming vidéo basée sur le SVC est élargie pour prendre en compte la Qualité d'Expérience (QoE) pour les décisions d'adaptation. La qualité vidéo est évaluée à l'aide des techniques objectives existantes avec une forte corrélation avec la perception humaine.

2014 - Détection du trafic de flux vidéo utilisant un système de FireSIGHT

Afin de détecter le trafic visuel de votre réseau, vous pouvez utiliser la fonctionnalité de contrôle d'accès et la caractéristique de Filtrage URL d'un système de FireSIGHT. Ce document décrit comment configurer un système de FireSIGHT à cet effet.

Les informations contenues dans ce document ont été créées à partir des périphériques d'un environnement de laboratoire spécifique.

2016 - Measuring Video QoE from Encrypted Traffic :

But : maintenir une qualité d'expérience satisfaisante pour les services de streaming vidéo. Le suivi et le maintien d'une qualité d'expérience satisfaisante pour les services de streaming vidéo deviennent plus difficiles que jamais pour les opérateurs de réseaux mobiles. Le téléchargement et le visionnage de contenu vidéo sur des appareils mobiles est actuellement une tendance croissante chez les utilisateurs, ce qui entraîne une demande pour une bande passante plus large et un meilleur dimensionnement de l'infrastructure réseau. Dans le même temps, la demande populaire de protection de la vie privée a conduit de nombreux services de streaming en ligne à adopter le cryptage de bout en bout, ne laissant aux fournisseurs qu'une poignée d'indicateurs pour identifier les problèmes de QoE. Afin de relever ces défis, les auteurs proposent une nouvelle méthodologie pour détecter les problèmes de qualité d'expérience de la diffusion vidéo en continu à partir du trafic crypté. Ils développent des modèles prédictifs pour détecter les différents niveaux de dégradation de la qualité d'expérience causés par trois facteurs d'influence clés, à savoir le décrochage, la qualité vidéo moyenne et les variations de qualité. Les modèles sont ensuite évalués sur le réseau de production d'un opérateur mobile à grande échelle, où nous pouvons voir qu'en dépit du cryptage, leur méthodologie est en mesure de détecter avec précision les problèmes de QoE avec une précision de 72%-92%, tandis que les performances sont encore meilleures lorsque le trafic en texte clair est traité.

2018 - Online Detection of Action Start in Untrimmed Streaming Videos :

ODAS (Online Detection of Action Start) a pour but de détecter le début d'une instance d'action, avec une grande précision de catégorisation et une faible latence de détection.

ODAS est important notamment pour la génération d'alertes précoces pour permettre une intervention rapide en matière de sécurité ou d'intervention d'urgence.

Ils proposent 3 nouvelles approches pour répondre spécifiquement aux défis de la formation des modèles ODAS.

2018 - Silhouette – Identifying YouTube Video Flows from Encrypted Traffic

Le trafic de streaming vidéo domine souvent les réseaux sans fil mobiles, obligeant les fournisseurs de services Internet (Internet Service Providers, IPS's) à déployer du "shaping" vidéo pour identifier puis gérer le trafic pendant les périodes de saturation. Mais l'utilisation du cryptage de bout en bout (TSL / SSL, par exemple) rend difficile l'identification des flux vidéo, même avec une inspection approfondie des paquets. En guise d'alternative, cet article présente Silhouette - une méthode de classification vidéo en temps réel. Silhouette utilise uniquement des statistiques de flux (c'est-à-dire de «shape») pour l'identification vidéo, ce qui la rend efficace pour identifier un flux vidéo même lorsque celui-ci est cryptée.

3 Plan de travail

Début : 31 janvier 2019

Fin : 5 avril 2019.

Pour la répartition de notre charge de travail nous prévoyons :

- 1 mois pour les fonctionnalités critiques (voir section 4. Objectif de réalisation).
- 1 mois pour la correction d'erreurs et amélioration de l'application (voir section 4.).

Dernière semaine : préparation de la présentation.

4 Objectif de réalisation

Nous avons pour but de réaliser une application mobile qui aura les fonctionnalités (critiques) suivantes :

- Un utilisateur peut s'inscrire ou se connecter.
- Un utilisateur peut avoir accès à une liste d'animés.
- Un utilisateur peut chercher un animé par titre.
- Un utilisateur peut marquer un/des épisodes d'un animé comme étant "vu(s)".
- L'application reposera sur une base de donnée d'animé (de leurs saisons et épisodes) qui se mettra régulièrement et automatiquement à jour.
- L'application détectera un flux vidéo sur un navigateur web (sûrement restreint à Firefox). Puis mettra à jour automatiquement le visionnage de l'animé correspondant dans la base de données de l'utilisateur.

Pistes d'améliorations sans ordre de priorité (fonctionnalités non critiques) :

- Un animé possédera une fiche d'informations (synopsis, date de sortie, personnages principaux ...).
- Un animé passe en mode "vert" ou "tout visionné" si son utilisateur a regardé tous les épisodes associés à celui ci.
- Un utilisateur peut noter la qualité d'un animé. La moyenne de la note de l'animé sera visible par tous les utilisateurs.
- Un utilisateur peut regarder une vidéo Youtube et voir sa liste sur l'application se mettre à jour.
- Un utilisateur pourra utiliser un autre navigateur que Firefox.
- Un utilisateur peut avoir accès à des statistiques divers sur ses visionnages (nombres d'heures moyen passé à regarder des animés ...).
- Un utilisateur peut chercher un animé par catégorie.

Sur ce site nous pouvons avoir un compte et les listes d'animés que nous suivons.

5 Technologies envisagées

Notre projet se divisera en 4 parties :

- une extension Web : pour Firefox (puisque Firefox mobile permet le développement d'extensions contrairement à Chrome) et en javascript (pour l'analyse du code HTML des sites web visités)
- une application mobile : en React Native (pour la simplicité et la modularité avec les composants)
- une API pour faire communiquer l'extension Web et l'application : avec une architecture REST (pour HTTP et apprendre à se servir ce type d'architecture de plus en plus prisé)
- une BDD pour les comptes utilisateur et leur catalogue associé : en MySQL (plus simple à utiliser avec nos connaissances et la BDD ne sera pas la tâche la plus conséquente du projet)

6 Pitch

L'application mobile "Mimashita", vous permet de lister tous vos animés et ainsi de garder un suivi de vos derniers épisodes visionnés.

L'application propose un large choix d'animés se mettant automatiquement à jour.

De plus, grâce à une extension Firefox, vous pourrez voir votre liste d'animés mise à jour automatiquement dans votre application dès que vous regarderez une vidéo sur votre navigateur.

Références

- [1] Détection du trafic de flux vidéo utilisant un système de firesight, 2014.
- [2] Giorgos Dimopoulos, Ilias Leontiadis, Pere Barlet-Ros, and Konstantina Papagiannaki. Measuring video qoe from encrypted traffic. 2016.
- [3] Kwok-Chun Fung and Yu-Kwong Kwok. A qoe based performance study of mobile peer-to-peer live video streaming. 2012.
- [4] Feng Li, Jae Won Chung, and Mark Claypool. Silhouette - identifying youtube video flows from encrypted traffic. 2018.
- [5] Lie Lu, Yan-Feng Sun, Mingjing Li, Xian-Sheng Hua, and Hong-Jiang Zhang. Automatic detection and segmentation of music videos in an audio/video stream. feb 2008.
- [6] Ashwin Rao, Yeon-sup Lim, Chadi Barakat, Arnaud Legout, Don Towsley, and Walid Dabbous. Network characteristics of video streaming traffic. 2011.
- [7] Julius Rückert, Osama Abboud, Thomas Zinner, Ralf Steinmetz, and David Hausheer. Quality adaptation in p2p video streaming based on objective qoe metrics. 2012.
- [8] Zheng Shou, Junting Pan, Jonathan Chan, Kazuyuki Miyazawa, Hassan Mansour, Anthony Vetro, Xavi Gir-i Nieto, and Shih-Fu Chang. Online detection of action start in untrimmed, streaming videos. jul 2018.