QoE-Driven Video Streaming System over Cloud-Based VANET

Sofiene Jelassi[1], Amna Bouzid[2], and Habib Youssef[3]

[1] ISIM of Monastir, Route Korniche - BP n 24, 5000 Monastir, Tunisia
sofiene.jelassi@infcom.rnu.tn
[2] Isitcom, GP1 - Hammam Sousse, Tunisia
amna.bouzid@isetsousse.tn
[3] CCK, University Campus of Manouba, Tunisia
habib.youssef@fsm.rnu.tn

Abstract. The cloud-based VANETs (Vehicular Ad-Hoc NETwork) will be deployed in the near future that is going to extend VANET applications to include multimedia services. Motivated by that fact, we build a video streaming framework over a cloud-based VANET architecture that is composed of a central cloud, roadside cloudlet and vehicular cloud. The video streaming content can be seen by in-move vehicles' passengers or static vehicles' drivers that maintain a synchronized list of available video content located inside our system facilities. A *video streaming scheduling* is created upon the reception of a video content request at the vehicle, the cloudlet or the center cloud as a function of the vehicular cloud conditions and content localization. The video streaming scheduling defines the involved nodes in a video delivery session and its perceived quality. A set of VMs are installed over the cloudlets that plays the role of roadside video streaming servers. They exchange control information about vehicles' mobility and contents using a background process in order to achieve uninterrupted streaming sessions. The vehicles include a QoE-monitor that measures end-user quality and reports it to the QoE-controller, which resides in VMs or in the vehicles.

1 Introduction

It is required to support video streaming over vehicular networks, VANET (Vehicular Ad-hoc NETwork), in order to extends its range of applications and enhance users' experience. This includes in addition to infotainment services, navigation, safety and maintenance applications [1]. However, providing an acceptable quality of video streaming service is a challenge given the characteristics of VANETs, such as high network topology dynamics inducing short-lived sessions, an error-prone wireless channel and a limited end-to-end bandwidth. Existing work for providing video streaming on vehicular networks use often traditional techniques in order to consider VANET features, such as light signaling protocols, predictive handovers, resources' reservation, network coding, packet header compression, efficient coding scheme and loss concealment algorithms [2] and [3]. Moreover, proposals that use

M. Kassab et al. (Eds.): Nets4Cars/Nets4Trains/Nets4Aircraft 2015, LNCS 9066, pp. 84–93, 2015.
DOI: 10.1007/978-3-319-17765-6_8

relay nodes for enabling video streaming have been thoroughly investigated in the literature [4]. However, they were concerned with emergency video dissemination rather than unicast entertainment-oriented video streaming [1] and [2].

The effective penetration of cloud technology in the supply chain of legacy services was a stimulating factor to think about integrating a cloud system and its underlying principle in a VANET environment [5] and [6]. This explains why a number of cloud-based VANET architectures have been recently proposed and developed in the literature [7], [8] and [9]. In addition to the basic cloud functionalities, namely storage and computing sharing, a cloud can host many business logics composing a given service.

This paper introduces an innovative solution in order to provide a satisfactory video streaming quality over a cloud-based VANET architecture. The improvement of video streaming delivery conditions over VANET is realized using cloudlets installed across roads and highways deployed by roadside infrastructure providers. They are connected to the roadside units using high speed connections. Each roadside cloudlet run at least one Virtual Machine (VM) that includes a video streaming server that represents a partial image of the central video streaming server installed at the central cloud. In order to prevent service interruption, a video streaming scheduling is built by the client, the cloudlet or the central cloud as a function of content availability and vehicles trajectory.

The remaining parts of this paper are organized as follows; in the section 2, we briefly describe cloud-based VANET architectures. Section 3 presents our proposed framework for providing QoE-centric video streaming service over cloud-based VANET. Section 4 outlines our strategy to prevent service interruption. Section 5 describes the workflow of video streaming components at vehicle and cloudlets. The conclusion is presented in Section 6.

2 The Cloud-Based VANET Architecture at a Glance

A comprehensive cloud-based VANET architecture has been proposed in [9]. The whole system is graphically illustrated in Fig. 1. As we can see, it integrates a central cloud (CC), a roadside cloud (RSC) as well as a vehicular cloud (VC). The CC is located at the root or backend of the proposed cloud-based VANET architecture. It manages Virtual Machine (VM) and dispatches tasks and content across the RSC as a function of resources availability, clients' requests and vehicles' mobility. The RSC includes a set of roadside cloudlets (RCL) that can be seen as a small-scale cloud connected directly to a given RSU (RoadSide Unit). This means that access delay to a RSU from a cloudlet is almost equal to zero. A RCL is intended to provide cloud services to vehicles located in its coverage [9]. A cloudlet is able, similar to a CC, to virtualize physical resources, such as CPU, memory and hard disk. Moreover, it acts as a local cloud provider that offers essential cloud services, such as storage and computing. The geographical proximity of RSC and vehicles enables a reduced last-hop access and transfer

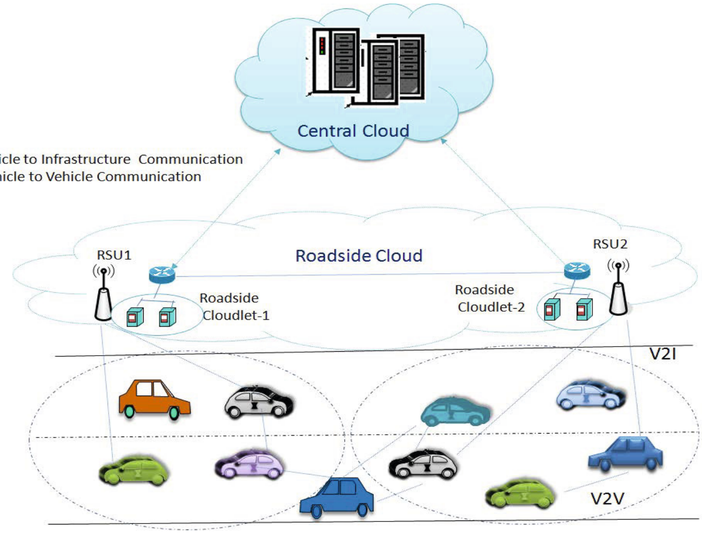

Fig. 1. The global architecture of a typical cloud-based VANET

delays. A vehicle can select a nearby roadside cloudlet in order to use virtualizable resources defined in the term of a VM. In order to assure an uninterrupted cloud service, the customized VM should be synchronously transferred between the roadside cloudlets as a function of vehicle mobility. This is widely known as VM migration that is a fundamental functionality in cloud environment.

A Vehicular Cloud (VC) is located at the bottom of the cloud-based VANET architecture. It is defined as a transient mobile cloud composed by a collection of vehicles that would like to share their resources. The vehicles that belong to a VC should be able to communicate directly to each other. In general, a VC is formed only if there are no enough resources at the cloudlet.

The main goal of our work is to provide video steaming over the cloud-based VANET architecture. This is actually a challenging task given its high sensitivity to delay and bandwidth fluctuations. However, characteristics of a cloud-based VANET architecture facilitate providing video streaming service. The different components of our video streaming system and their interaction are going to be developed in a way that; (1) account for architectural characteristics of cloud-based VANET and (2) maximize the QoE of delivered video streams by optimizing jointly sender and receiver applications as well as midway involved entities.

3 The Engineering of Video Streaming System over a Cloud-Based VANET

In order to enable video streaming over a cloud-based VANET, we propose a complete video streaming system that aims to achieve a satisfactory Quality of

Experience (QoE) using (1) the flexibility and potential of cloud technology and (2) the adaptation capabilities of source and receiver video streaming sides. Basically, our video streaming solution integrates three main components (see Fig. 2):

1. A QoE-driven video streaming player that is installed on each vehicle.
2. A VM that is hosted at the roadside cloudlet and serves one or multiple video streaming clients. It is intended to play the role of a mobile video streaming server that is able to communicate with mobile streaming servers running on adjacent cloudlet.
3. A VM installed on the center cloud that manages and monitors all VMs instantiated at various cloudlets as well as their content. This is more related to remote management and monitoring of VMs in a distributed cloud environment. This issue is currently outside the scope of our work.

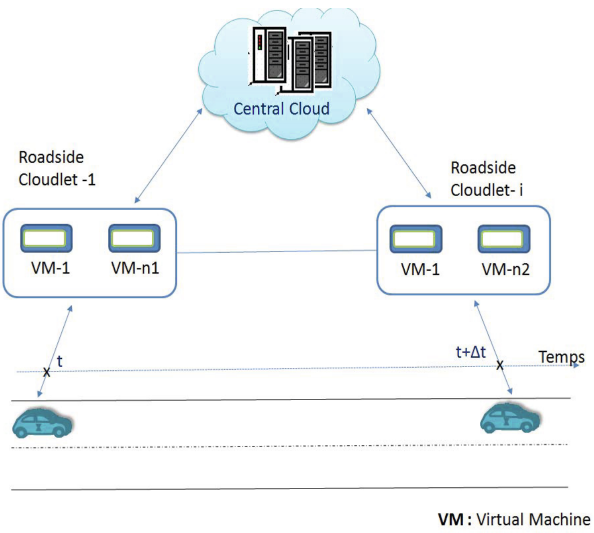

Fig. 2. QoE-driven video streaming architecture of cloud-based VANET

As we can see in Fig. 2, a vehicle is going to be served by a series of VMs running over crossed roadside cloudlets. The layout of all components installed at the cloudlet and the vehicle sides are presented in Fig. 3. As we can see, a cloudlet includes a manager of VM running in Dom0 that manages all VMs and is responsible for configuring running video streaming servers. Moreover, the cloudlet maintains a VDB (Video Data Base) and a PVDB (Popular Video Data Base). The latter is used in order to cache popular video contents. The VM (domU) used as a video streaming server includes the following main components:

1. The video streaming manager - cloudlet side: It receives clients requests and creates video streaming scheduling according to available resources, content distribution and vehicle information grid.
2. The video prefetching manager - cloudlet side: It fetches a requested content in its video source index. If the content is not found then the request in relayed to the central cloud.
3. The QoE-Controller – server side: It defines on-line streaming parameters as a function of available channel capacity and vehicle information grid.
4. The video streaming sender: It encodes, packetizes and sends video streaming chunks as a function of parameters set by the QoE-Controller – sender side.
5. The streaming buffering manager: It allocates and releases the internal VM buffer's space as a function of scheduled transfer tasks.

At the vehicle side, we propose a QoE-driven video streaming clients that is composed of the following key elements:

1. The video streaming manager - vehicle side: It receives clients requests and creates, if possible, a video streaming scheduling over a vehicular cloud. Otherwise, it relays the request to the video streaming manager - cloudlet side.
2. The video prefetching manager - vehicle side: It fetches a requested content in its video source index. If the content is not found then the request is relayed to the cloudlet.
3. The video streaming receiver: It receives video packet stream that are analyzed, filtered, decoded than forwarded to the streaming buffer manager.
4. The streaming buffering manager: It inserts correctly received video frames in the play-out buffer using configuration parameters calculated by the QoE-Controller.
5. The QoE-Controller – receiver side: It adapts the parameters of the play-out process as a function of received data rate and content. It mainly controls the play-out buffer delay and its content.
6. The QoE-Monitor: It probes played streams at different levels in order to measures QoE values of on-going video streaming sessions. This is realized using a set of technical metrics and installed QoE-models. The measured value are reported to QoE-Controller located at the vehicle and cloudlet sides.

Tables 1 and 2 give an exhaustive list of the main components of our proposed video streaming system installed at the cloudlet and vehicle sides. The QoE-Monitor aims to provide feedbacks about encountered QoE by the end-users. The measurement of QoE values is realized at the vehicle for a given configuration using a set of gathered local metrics. It is then sent to the QoE-controllers located at the vehicle and/or cloudlet side at specific time interval. The technical metrics considered by our QoE-monitor are the followings:

– The video prefetching delay: The time spent to prefetch a video from PVDB, VDB or in central cloud.

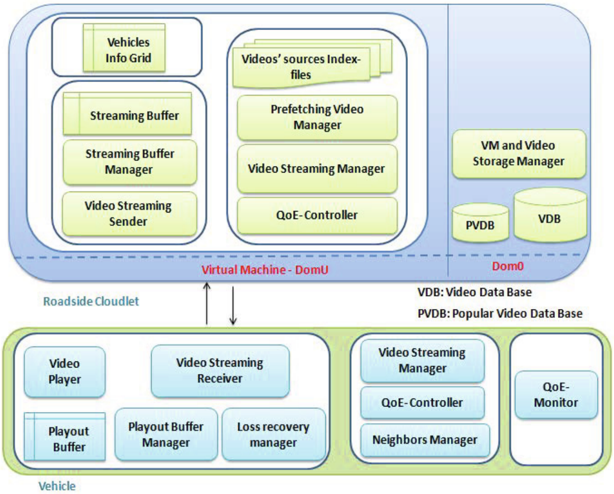

Fig. 3. The vehicle and cloudlet architectures

– The frame resolution and rate: It defines the spatial and temporal resolution of displayed frames that may be adapted during the service.
– The encoding scheme its encoding rates: It defines the used codec and its resulting bitstream rate.
– The packet size and interval: They are used by the packet stream packetizer in order to create data packets. The packet interval refers to the video delay included into a packet.
– The playout delay: The buffering delay at a vehicle before starting playback processes. This is a well-known techniques used to filter-out packet delay variation of received video chunks.
– The ratio of lost packets and the mean loss and inter-loss duration: This metrics capture characteristics of of encountered packet loss process.
– The one-way delay and its variation: The average transit delay from the source to the receiver.
– The number of rebuffering: The number of rebuffering events is the number of times the Video Player stopped to rebuffer frames after it started playing.

A QoE-Controller is deployed at the vehicle and cloudlet sides. They are responsible for tunning and selecting optimal video streaming key parameters in order to enhance QoE. The QoE-Controller residing at the vehicle side controls mainly play-out delay, startup delay, number of re-buffering, the space of the playout buffer and the loss concealment techniques. On the other hand, QoE-Controller residing at the cloudlet controls sending rate, delivery paths, packet time interval, redundancy scheme, handover strategy, etc. The operational mode of QoE-Controllers is going to be detailed in a future work.

Table 1. The video streaming components at the roadside cloudlet

	Module	Role
The cloudlet server (Dom0)	VMs and video storage Manager	It manages VM and related storage spaces
	Video Data Base (VDB)	A storage space of vehicles' videos
	Popular Video Data Base (PVDB)	A storage space of most popular videos with respect to social networks
The Virtual Machine (DomU)	Prefetching Video Manager	It fetches video content in PVDB, VDB or in nearby roadside cloudlet or in central cloud.
	QoE-Controller	It adapts the transmission bitrate as well as the spatial and temporal resolutions of the video content according to bandwidth occupancy and playback buffer state
	Video Streaming Manager	It manages video content requests and creates a video streaming scheduling according to the content chunk distribution and vehicle mobility. It also manages handover across cloudlets
	Streaming Buffer Manager	It manage the internal streaming buffer of a VM used by the server to accelerate transmission of video chunks

Table 2. The video streaming components at the vehicle

Module	Role
QoE-Monitor	It measures and reports video quality value to QoE-Controllers.
QoE-Controller	It defines the configuration parameters used by a video player, such as startup delay and frame rate.
Video Streaming Manager	It handles video content requests and creates a video streaming scheduling. It may decide to cache a received video chunk in its local buffer
Loss recovery manager	It can ask for a missing video chunk from either the serving VM or its neighbors vehicles.
Video Player	It reads the video image from a playback buffer
Playout Buffer manager	It places received video chunk in their suitable location inside the playback buffer

4 The Video Streaming Scheduling for Service Continuity

To provide a stable and satisfactory QoE of video streaming to mobile vehicles, a session management protocol should be used. Its main goal is to prevent video freezing during a handover process between cloudlets. To date, there is no a dedicated management protocol of video streaming session that is specifically designed for a roadside cloud. In order to maintain video streaming continuity, we assume that each cloud member runs a VM hosting a video streaming server. They can exchange control information about vehicles' mobility and video session. A streaming scheduling specifying the entity which is going to provide a given video portion at a specific time.

The video streaming scheduling may be built by the client, the cloudlet or the central cloud. Precisely, if a client is a member of a functional vehicular cloud, then it searches for the requested content in its local video index file that includes reference to available video content. If the content is available over the vehicular cloud, then a streaming scheduling is locally built. Otherwise, a request is sent to the nearest cloudlet that performs a similar processes. If the content is unavailable at the cloudlet, then the streaming scheduling is built by the central cloud. The video streaming scheduling is built as a function of the current vehicle position, its velocity and direction. The information about vehicle mobility may be obtained from the vehicle traffic mining process. A dual-connection is opened during a handover interval in order to perform a seamless transition. This assures a quick transition of the serving cloudlet that helps in maintaining service continuity. Currently, we are evaluating our video streaming service continuity protocol over cloud-based VANET. More details and results will be given in our future publications.

5 The Workflow of Video Streaming Service Components

5.1 The Video Streaming Server Workflow at the Roadside Cloudlet

The video streaming server installed at a roadside cloudlet follows the workflow presented in Fig. 4 using the UML activity diagram. Upon the reception of a video content request, the video manager call the prefetching video manager that checks if an image exists in its local PVDB and if not found in VDB using video index file. If the local image is found then a video streaming scheduling is built by the video streaming manager as a function of vehicle speed and direction, and subsequently sends to its neighbor cloudlets. However, if the video content does not exist at the cloudlet then the received client request is relayed to the center cloud that builds a streaming scheduling and sends it to all involved cloudlets in the conjunction with the suitable video content. In both cases, the video streaming process starts as soon as a streaming scheduling is received by all involved elements. Once a session is activated, the QoE-monitor reports periodically to the QoE-controller estimates regarding QoE values in order to select the configurations that optimize video streaming service.

5.2 The Video Streaming Client Workflow at the Vehicle

The vehicle adapts dynamically the configuration parameters of the playback process in order to optimize the QoE of played video. The workflow of our proposed video streaming player dedicated to VANETs is given in Fig. 5. The playback policy is supervised by the QoE-controller that set configuration parameters of the playback process. Moreover, the QoE-controller decides to send a request for retransmitting a missing video segment if its expected arrival time is less than its calculated play-out time. Furthermore, the QoE controller may force the QoE-monitor to send urgent QoE report to the serving cloudlet if an exceptional event is observed.

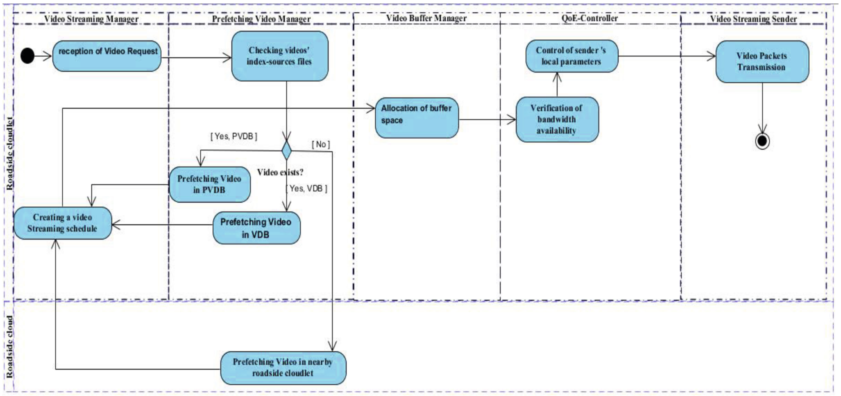

Fig. 4. The activity diagram of workflow realized in the roadside cloudlet

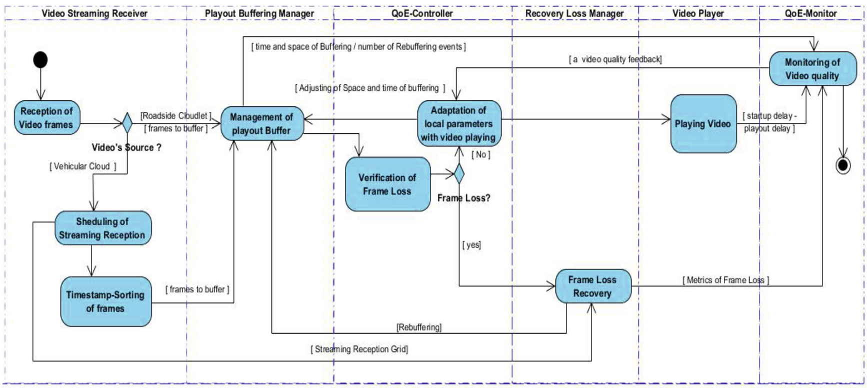

Fig. 5. The activity diagram of QoE-driven workflow realized in the vehicle

6 Conclusion

This work proposes a scheme and a strategy that is going to improve QoE of video streaming over VANET. The main idea consists of using an emerging cloud-based VANET architecture that helps in reducing quality impairment factors observed by ordinary VANETs. Each cloud member includes a VM in order to run video streaming component at server and client sides. Each roadside cloudlet configures installed VM that are responsible for prefetching, caching and streaming videos to vehicles. A video streaming scheduling is built at a session start-up in order to prevent video streaming interruptions. This framework is currently under investigation using simulation and experimentation. The results are going to be published in our future publications.

References

1. Park, J.-S., Lee, U., Gerla, M.: Vehicular communications: Emergency video streams and network Coding. Journal of Internet Services and Applications 1, 57–68 (2010)
2. Naeimipoor, F.: Video streaming and multimedia broadcasting over vehicular Ad Hoc networks. PhD thesis, University of Ottawa (2013)
3. Bradai, A., Ahmed, T.: ReViV: Selective rebroadcast mechanism for video streaming over VANET. In: Proc. of the IEEE 79th Vehicular Technology Conference (IEEE VTC 2014), Seoul, Korea (2014)
4. Rezende, C., Mammeri, A., Boukerche, A., Loureiro, A.: A Receiver-based Video Dissemination Solution for Vehicular Networks with Content Transmissions Decoupled from Relay Node Selection. Ad-Hoc Network 17, 1–17 (2014)
5. Whaiduzzaman, M., Sookhak, M., Gani, A., Buyya, R.: A survey on vehicular cloud computing. Journal of Network and Computer Applications 40, 325–344 (2014)
6. Abuelela, M., Olariu, S.: Taking vanet to the clouds. In: Proceedings of the 8th International Conference on Advances in Mobile Computing and Multimedia
7. Olariu, S., Eltoweissy, M., Younis, M.: Towards Autonomous Vehicular Clouds. ICST Transactions on Mobile Communications and Applications 11, 1–11 (2011)
8. Hussain, R., Son, J., Eun, H., Kim, S., Oh, H.: Rethinking Vehicular Communications: Merging VANET with cloud computing. In: Proc. of IEEE 4th International Conference on Cloud Computing Technology and Science, Taipei, Taiwan (2012)
9. Yu, R., Zhang, Y., Gjessing, S., Xia, W., Yang, K.: Toward Cloud-based Vehicular Networks with Efficient Resource Management. IEEE Network 27, 48–55 (2013)