RESEARCH ARTICLE OPEN ACCESS

A Comparative Study of Techniques Used in Video Streaming of VANETs

Divya Susan Mathew
Department of Computer Science
STIST, Trivandrum
Kerala -India

ABSTRACT
A Vehicular Ad hoc Network or VANET is a technology that uses moving cars as nodes in a network to create a mobile network. VANET turns every participating car into a wireless router or node, allowing cars approximately 100 to 300 metres of each other to connect and, in turn, create a network with a wide range. Vehicular Ad hoc Networks (VANETs) are ill-suited to support streaming media. JXTA is an open network computing platform designed for peer-to-peer (P2P) computing. Its goal is to develop basic building blocks and services to enable innovative applications for peer groups. This paper presents a comparative study of different techniques used for inter vehicular communications, that leverages the characteristics of streaming applications to yield a highly efficient solution. A solution for inter vehicular communications, called Streaming Video is fully distributed and considers volatile topology. In order to maintain high video streaming quality while reducing the wireless service cost, the optimal video streaming process with multiple links is formulated as a Markov Decision Process (MDP). The reward function is designed to consider the quality of service (QoS) requirements for video traffic, such as the startup latency, playback fluency, average playback quality, playback smoothness and wireless service cost. To solve the MDP in real time, we propose an adaptive, best-action search algorithm to obtain a sub-optimal solution. To evaluate the performance of the proposed adaptation algorithm, we implemented a testbed using the Android mobile phone and the Scalable Video Coding (SVC) codec. Experiment results demonstrate the feasibility and effectiveness of the proposed adaptation algorithm for mobile video streaming applications, which outperforms the existing state-of-the-art adaptation algorithms.
***Keywords:* —** Vehicular Adhoc Networks (VANETs), JXTA Protocol, Streaming Video, Markov Decision Process (MDP), Scalable Video Coding (SVC).

I. INTRODUCTION

VIDEO streaming is gaining popularity among mobile users recently. Considering that the mobile devices have limited computational capacity and energy supply, and the wireless channels are highly dynamic, it is very challenging to provide high quality video streaming services for mobile users consistently. It is a promising trend to use multiple wireless network interfaces with different wireless communication techniques for mobile devices. For example, smart phones and tablets are usually equipped with cellular, WiFi and Bluetooth interfaces. Utilizing multiple links simultaneously can improve video streaming in several aspects: the aggregated higher bandwidth can support video of higher bit rate; when one wireless link suffers poor link quality or congestion, the others can compensate for it [1].Streaming Video completely relies on inter vehicular communication: a video stream, generated at a roadside access point, is fed to the nodes and disseminated across the VANET which consists of mobile nodes [2]. The nodes that belong to the distribution structure and forward the streaming video are the relay nodes. The distribution structure is assumed to be a grid, so as to minimize the number of relay nodes required to cover a network area while providing a good range of connectivity. Each relay node not only forwards the streaming traffic but also exploits a GPS and the received power level, to dynamically select its next-hop neighbours, so that the distribution structure approximates a grid as closely as possible. The GPS signal is also used to synchronize all relay nodes to a TDMA transmission, as done in several other works. In order to provide the described services, APs can be utilized to distribute the streaming media to nearby vehicles. Due to the high deployment cost of roadside APs and limited communication range, the entire road cannot be fully covered merely by APs. Therefore, the vehicles have to form a vehicular ad hoc network (VANET) and cooperatively propagate the streaming media when they are out of coverage of Aps.

Streaming Video can achieve a good performance even with spotty, dynamic vehicular connectivity. This work assumes that one or more gateway nodes, either fixed or mobile, provide streaming video to car passengers. Examples of streaming video include news, tourist information, commercial advertisements, football games, or music video clips. Distribution of multimedia data relies on inter vehicular communication and the vehicles may exchange best-effort data traffic in a peer-to-peer fashion: news summaries, public transportation timetables, traffic warnings, and so on.

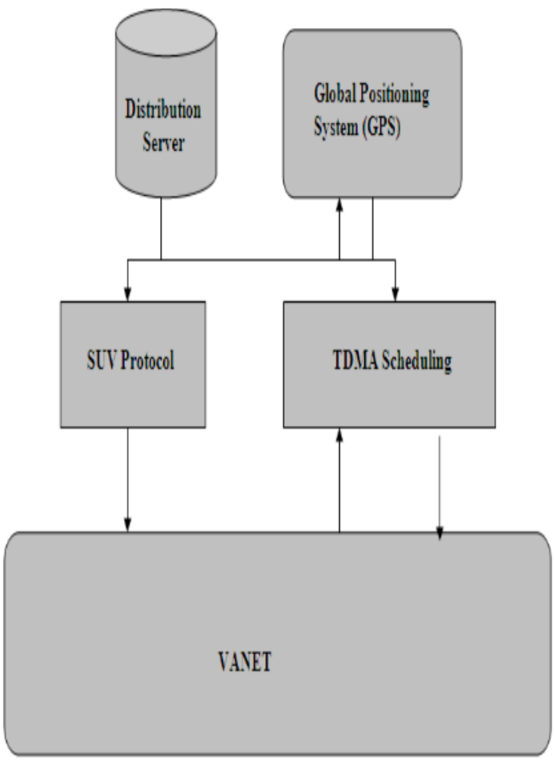

Fig 1: Streaming Urban Video Architecture

II. THE STREAMING VIDEO PROTOCOL

The tenets of the solution proposed are, the selection of relay nodes so as to maximize the coverage area, scheduling of relay nodes in TDMA fashion, scheduled access for streaming video, opportunistic access for streaming video and contention-based access for best-effort traffic.

A. Algorithm

Step 1: The distribution server is executed
Step 2: It checks the IP address and the password
Step 3: It can list the system files
Step 4: Enable the GPS
Step 5: Each of the peer groups are enabled
Step 6: They are given names and addresses
Step 7: Any peer node can send messages to other peer node
Step 8: The files can be shared by the nodes and saved
Step 9: These files can be searched
Step 10: The video file can be downloaded

Step 11: The details of distribution can be displayed

The distribution server provides a fully distributed solution called Streaming Video. IP address is checked to be valid or invalid. A video stream, generated in a point in space, is fed to the nodes and disseminated across the VANET through this server. It is associated with all the users in the VANET network. It also consists of the multimedia content. It is assumed to be a video sequence. The GPS minimizes the number of relay nodes required to cover a network area while providing a good level of connectivity. It is also used to synchronize all relay nodes to a structured TDMA transmission. All network nodes are supposed to be equipped with a positioning system, so that they are aware of their location and accurately synchronized in time. Each vehicle periodically broadcasts an in-band HELLO signaling message, which carries the sender's ID and GPS position. A vehicle can therefore keep an updated list of its 1-hop neighbors, i.e., the nodes from which it receives a HELLO with power level, and the position and the power level received from each of its 1-hop neighbors. The users can join or remove from the VANET group at any time. Basically these VANET groups contain very large number of users. So the group size is also the important factor for effective video streaming without any delay or data loss. The JXTA protocol is used to create a VANET. JXTA is an open source protocol and it is XML based. Basic settings are given. Service settings are given i.e., it can act as rendezvous, proxy or relay.

III. THE JXTA PROTOCOL

JXTA is an open network computing platform designed for peer-to-peer (P2P) computing. Its goal is to develop basic building blocks and services to enable innovative applications for peer groups. It provides a common set of open protocols and an open source reference implementation for developing peer- to-peer applications. The JXTA protocols standardize the manner in which peers communicate. The JXTA network consists of a series of interconnected nodes, or peers. Peers can self-organize into peer groups, which provide a common set of services. JXTA peers advertise their services in XML documents called advertisements. Advertisements enable other peers on the network to learn how to connect to, and interact with, a peer's services. JXTA peers use pipes to send messages to one another. Pipes are an asynchronous and unidirectional message transfer mechanism used for service communication. Messages are simple XML documents whose envelope contains routing, digest, and credential information. Pipes are bound to specific endpoints, such as a TCP port and associated IP address. The steps to be performed are, obtaining the group services, building and publishing a module class advertisement, building pipe advertisement, building and publishing module specification advertisement and waiting for messages. The client peer will need to execute a few steps in order to use the pipe published by the sender peer. The steps are, the receiver peer will attempt to find the advertisement in the local cache, when the attempt fails the remote discovery request is sent by receiver peer using discovery service, module specification advertisement is obtained from remote discovery request, the receiver peer obtains pipe advertisement, pipe data and creates pipe advertisement, the receiver creates output pipe.

B. Algorithm

Step 1: The file manager is created
Step 2: The CMS class is designed
Step 3: The nodes can be added, advertisements can be given and files can be updated
Step 4: The peer nodes can share any file
Step 5: It can be removed
Step 6: The pipe messenger is created for sending and receiving messages
Step 7: Play video is designed for downloading video files and viewing it
Step 8: Search list is designed for getting files and other data
Step 9: The trust manager is created for checking the validity of data
Step 10: Start jxta function is used for creating peer nodes and for other services
Step 11: Messages can be sent and distribution details can be displayed

C. Node Coloring

This paper considers the fact that the network topology composed of relay nodes has a grid-like structure [2]. This first makes use of a regular grid topology, i.e., every node has two neighbors along each spatial dimension and that neighboring nodes are at the same distance R from each other.

This formulates the scheduling problem and provides a solution that is optimal for all cases of practical interest. The obtained scheduling scheme is then applied to a realistic VANET, where, in general, the distance between relay nodes is shorter than R and relay nodes form an irregular grid.

TABLE I
BEST EFFORT TRAFFIC PERFORMANCE

No of Nodes	**5**	**8**	**11**	**14**
Efficiency	.8	.7	.6	.4

Table I reports the access efficiency for what concerns the best effort traffic. It points out that Streaming Video allows best effort to pick up what is left by the streaming media and that every active node acts as a “hot spot.” The access efficiency decreases with the increase of the number of network nodes.

IV. MARKOV DECISION PROCESS

Putting all elements together results in the definition of a Markov decision process, which will be the base model for the large majority of methods.

A Markov decision process is a tuple 〈S, A, T, R〉 in which S is a finite set of states, A a finite set of actions, T a transition function defined as $T:S\times A\times S\to[0,1]$ and R a reward function defined as $R:S\times A\times S\to\mathbb{R}$.

The transition function T and the reward function R together define the model of the MDP. Often MDPs are depicted as a state transition graph for an example) where the nodes correspond to states and (directed) edges denote transitions.

1) ***Optimality Criteria:*** We have to define what the model of optimality is. There are two ways of looking at optimality. First, there is the aspect of what is actually being optimized, i.e. what is the goal of the agent? Second, there is the aspect of how optimal the way in which the goal is being optimized, is. The first aspect is related to gathering reward and is treated in this section. The second aspect is related to the efficiency and optimality of algorithms. The goal of learning in an MDP is to gather rewards. If the agent was only concerned about the immediate reward, a simple optimality criterion would be to optimize E[rt]. The finite horizon model simply takes a finite horizon of length h band states that the agent should optimize its expected reward over this horizon, i.e. the next h steps . One can think of this in two ways. The agent could in the first step take the h-step optimal action, after this the(h−1)-step optimal action , and so on. Another way is that the agent will always take the h-step optimal action , which is called receding-horizon control. The problem, however, with this model, is that the (optimal) choice for the horizon length h is not always known.

V. MOBILE SCALABLE VIDEO STREAMING OVER HTTP

We present an end-to-end HTTP streaming testbed for scalable videos in this section. In our testbed, each mobile device dynamically requests SVC stream fragments to make sure that its buffer always contains next few fragments under dynamic network conditions. Each fragment contains g consecutive GoPs. We pre-process each raw H.264/SVC stream, and generate the description file and manifest file. The description file contains the meta data, including the details of NAL unit s and their offsets in the files. The description file is used to efficiently generate various fragments on-the-fly, based on the current streaming environments. The manifest file is generated for the mobile devices. It includes information such as video length, resolutions, highest temporal ID, total fragment number, and individual fragment sizes. The manifest file allows mobile devices to start video streaming sessions.

D. Streaming Videos

Fig. 2 shows the architecture of our HTTP streaming testbed, which consists of an HTTP-based streaming server, and one or multiple mobile receivers. The user of a mobile device first clicks on the hyperlink to the manifest file of the interested video. The client then downloads the manifest file and parses it for the detailed information of the requested video. Upon parsing the manifest file, the streaming client sequentially re-quests for fragments by specifying the fragment id, resolution, and temporal configuration. The streaming server then creates the next requested fragment on-the-fly, and transmits it to the mobile device. We note that the generated fragments can be stored in a cache for future use. In fact, there is a tradeoff between storage and speed depending on whether we generate fragments in-advance or on-the-fly. The requested data are stored in a circular buffer. Each decoder thread copies data from this circular buffer to decoder buffer before decoding. Once the data in the circular buffer is insufficient to decode, the decoder threads are blocked. The decoder threads continue while the circular buffer is refilled with enough data. The request thread is also blocked when the circular buffer is full. More specifically, the request thre ad is suspended when the circular buffer contains more than b bytes data, and is resumed when it contains less than l bytes data.

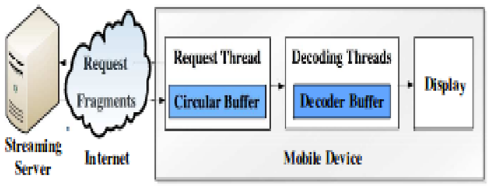

Fig. 2 The HTTP streaming testbed

As shown in fig. 3, the number of packet transmissions increases linearly as the number of nodes in Streaming Urban Video (SUV) protocol.

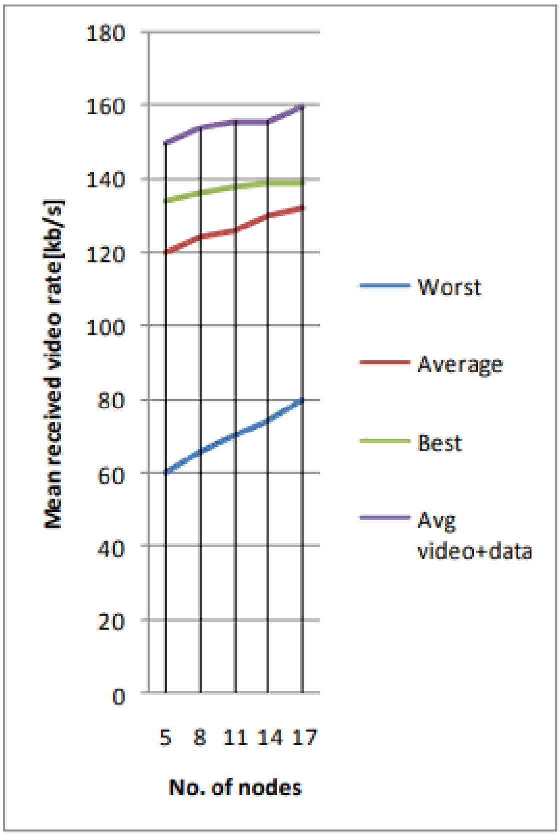

Fig 3: Mean Received Video Rate as a Function of the Number of Nodes in the Network

VI. CONCLUSIONS

Streaming Video can achieve a good performance even with spotty, dynamic vehicular connectivity. It assumes that one or more gateway nodes, either fixed or mobile, provide streaming video to car passengers. This work has addressed the support of video streaming in VANETs by providing a fully-distributed solution, Streaming Video. The video distribution is achieved through a fully distributed, dynamic selection of forwarders. As the distance from the gateway node increases, so does the chance that neighboring nodes are chosen as children by different parents, thus increasing the number of collisions. A relay node in a dense topology can replace a collided video frame with one of its copies, broadcast at a later time in a neighboring sector. Losses are reduced here. Whenever a collision occurs, Streaming Video leverages the properties of video coding to design a collision-free mechanism. It also promotes best-effort traffic exchange in a VANET without any infrastructure support.

The number of packet transmissions increases linearly as the number of nodes in Streaming Urban Video (SUV) protocol. As a future work, for broadcasting in a dense VANET, since there could be too many vehicles urgently demanding different portions of the LMS content, it can be extended to satisfy all their needs in a short time. By considering the time constraints of LMS applications, this can reduce frequent playback skips than in the case of sparse VANETs. Our experimental results also reveal an important trade-off between resolutions and frame rates. Due to resource constraints of mobile devices, a user may only pick high resolution or high frame rate. The user's decision depends on the video genres, device types, and even user preferences. There are active projects, conducting user studies and trying to model the Quality-of-Experience (QoE) of mobile video streaming. These user studies do not leverage scalable videos and can only consider very few resolutions and frame rates of each video. Our end-to-end mobile scalable video streaming testbed allows us to conduct large-scale user studies using H.264/SVC videos on commodity Android devices. This enables us to derive a more flexible QoE model, which is our future work.

ACKNOWLEDGMENT

The author would like to thank the anonymous reviewers for their valuable reviews.

REFERENCES

[1] Min Xing, Siyuan Xiang, Lin Cai, "A Real-Time Adaptive Algorithm for Video Streaming over Multiple Wireless Access Networks", IEEE Journal on Selected Areas in Communications, Vol. 32, Issue: 4, Apr. 2014.

[2] Fabio Soldo, Claudio Casetti, Caria-Fabiana Chiasserini, and Pedro Alonso Chaparro, "Video Streaming Distribution in VANETs," IEEE Transactions on Parallel

and Distributed Systems, Vol. 22, No. 7, pp. 1085-1091, Jul. 2011.

[3] F.J. Ros, P.M. Ruiz, and I. Stojmenovic, “Reliable and Efficient Broadcasting in Vehicular Ad Hoc Networks,” Proc. IEEE Vehicular Technology Conf. (VTC) Spring 2009, pp. 1-5, Apr. 2009.

[4] W. Chen, R.K. Guha, T.J. Kwon, J. Lee, and Y.-Y. H. Wirel, “A Survey and Challenges in Routing and Data Dissemination in Vehicular Ad Hoc Networks,” Wireless Communications and Mobile Computing, Oct. 2009.

[5] M. Li, Z. Yang, and W. Lou, “Codeon: Cooperative popular content distribution for vehicular networks using symbol level network coding,” Technical Report, ECE, WPI, 2010.

[6] F. Soldo, C. Casetti, and C.-F. Chiasserini, “Constant-Step Coloring,” Technical report, Politecnico di Torino, available at http://www.telematica.polito.it/casetti/TechRep_Scheduling.pdf, Mar. 2010.

[7] Y.-C. Chu and N.-F. Huang, “Delivering of Live Video Streaming for Vehicular Communication Using Peer-to-Peer Approach,” Proc. IEEE Mobile Networking for Vehicular Environments (MOVE Workshop), pp. 1-6, 2008.

[8] Fei Xie, Kien A. Hua, Wenjing Wang, and Yao H. Ho, “Performance Study of Live Video Streaming over Highway Vehicular Ad hoc Networks,” IEEE Vehicular Technology Conference(VTC-2007), pp. 2121- 2125, 2007.

[9] V. Naik, A. Arora, P. Sinha, and H. Zhang, “Sprinkler: A Reliable and Energy Efficient Data Dissemination Service for Extreme Scale Wireless Networks of Embedded Devices,” IEEE Trans. Mobile Computing, vol. 6, no. 7, pp. 777-789, July 2007.