Video upload from public transport vehicles using multihomed systems

Video Upload from Public Transport Vehicles using Multihomed Systems

Ali Safari Khatouni[1], Marco Ajmone Marsan[1,2], Marco Mellia[1]

[1]Politecnico di Torino, Italy - name.lastname@polito.it

[2]Institute Imdea Networks, Spain

***Abstract*— We consider a surveillance system for public transport vehicles, which is based on the collection of on-board videos, and the upload via mobile network to a central security system of video segments corresponding to those cameras and time intervals involved in an accident. We assume that vehicles are connected to several wireless interfaces, provided by different Mobile Network Operators (MNOs), each charging a different cost. Both the cost and the upload rate for each network interface change over time, according to the network load and the position of the vehicle. When a video must be uploaded to the central security, the system has to complete the upload within a deadline, deciding i) which interface(s) to use, ii) when to upload from that interface(s) and iii) at which rate to upload. The goal is to minimize the total cost of the upload, which we assume to be proportional to the data volume being transmitted and to the cost of using a given interface. We formalize the optimization problem and discuss greedy heuristics to solve it. Then, we discuss scientific and technical challenges to solve the system.**

Keywords*—*Smart city, public transport, security, video upload, wireless network, scheduling.

I. Introduction

We study a video surveillance system for public transport vehicles, which is based on the collection of on-board videos and their wireless transmission to a central security system. Our interest is motivated and inspired by the real needs of public transport operators. On public transport vehicles, already now, several video cameras are installed, each producing a video stream with rate from 1 Mb/s to 10 Mb/s. Continuous real-time video streaming from vehicles to the central security system is considered too expensive in data volume and in cost, and largely useless, because nothing relevant happens on the vehicles most of the time. Videos are thus stored on board, and when an alarm is triggered (e.g., when a customer or a driver reports a problem, or after a complaint is filed), the Security Operator (SO) on duty in the central security control station needs to access the portion of the on-board videos which refers to the period of time of the accident. In traditional systems, videos are uploaded to the central security system when the vehicle enters the depot, where cheap and high-speed wireless connectivity is available. This forces the SO to wait a long time before being able to investigate the accident.

Here, we consider a novel solution, which provides the SO with near-real-time access to videos corresponding to those cameras and time intervals involved in the accident. We assume that the vehicle is connected to the network by means of different wireless interfaces, through different Mobile Network Operators (MNOs), each charging a different cost, from cheap WiFi, to 3G/4G interfaces, or satellite links. Both the cost and the upload rate for each network interface change over time, according to the network load and the position of the vehicle. We assume that, thanks to the repetitiveness of the public vehicles routes, the system can create a performance map to collect information about the expected network connectivity performance along the route. (The creation of such map is outside the scope of this paper.)

Once the SO requests a video, the system has to complete the upload from the vehicle storage system within a given deadline. The system has to decide i) which interface(s) to use, ii) when to upload from that interface(s), and iii) at which rate to upload. The goal is to minimize the total cost of the upload, which we assume to be proportional to the data volume being transmitted and to the cost of using a given interface. For instance, assume that a video must be uploaded with a deadline of 5 minutes, and that the cost of using a given operator (slow and expensive) 3G interface is higher than the cost of using a (fast and cheap) WiFi interface of a second operator. However, the bus will enter the coverage area of the latter only in 3 minutes. In this context, is it better to wait entering under WiFi coverage, or to start uploading the video now?

The video upload problem can be seen as an optimization problem for which it is possible to obtain different formulations, depending on the assumptions. We discuss possible formulations and heuristics to solve it. Next, we faced the system design challenges that must be solved when engineering the entire system, like how to estimate and predict capacity, impact of transport protocol, impact of uncertainty, etc.

II. Model Formulation

We model the scheduling problem using a directed graph $G = (N, E)$, where $N = \{i\}$ is the set of nodes and $E = \{(i, j)\}$ is the set of edges. Referring to Fig. 1, the leftmost node represents the video source, i.e., the vehicle. The second group of nodes represents the video files to be uploaded. Each video k (2 videos in the example) is of volume V_k, and can be uploaded through different interfaces, at different time slots, represented by the third group of nodes. Each node in this group represents a given interface and time slot. For ease of visualization, nodes referring to the same interface (2 interfaces in the example) are grouped by a box. The number of available time slots (5 in the example) represents the deadline to meet (recall that we consider slotted time). The rightmost node represents the sink, i.e., the server receiving the videos.

All edges in E have a label containing two values: a cost and a capacity. The label of edge (i, j) is denoted $(c_{i,j}, r_{i,j})$.

This work was funded by the *MONROE* project (grant agreement no. 644399) in the H2020-ICT-11-2014.

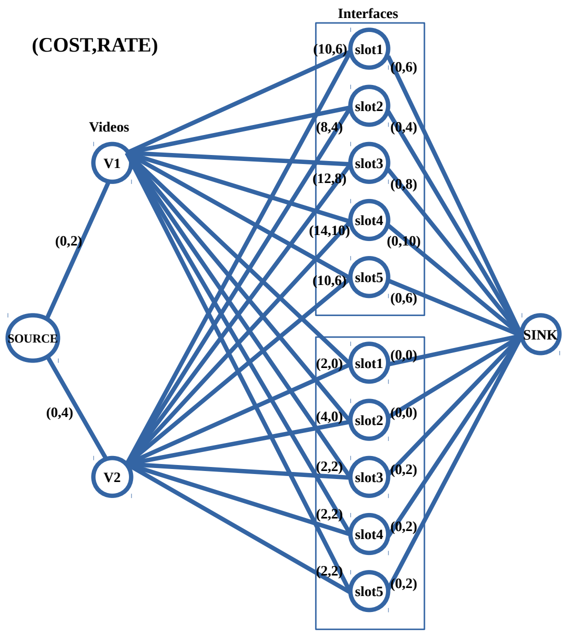

Fig. 1: An example to represent the model.

The source node is connected to each video node. Edges exiting from the source node have zero cost, and capacity equal to the video file total size. Each video node is connected by a directed edge to nodes representing a time slot and interface. These edges are characterized by the cost per bit of using such time slot and interface ($c_{i,j}$), and the maximum flow $f_{i,j}$ that can be supported by such time slot and interface ($r_{i,j}$) in bits/s. This model allows videos to have different deadlines. Indeed, each video is connected only to the slots it can use. Each node representing a time slot and interface is connected to the sink with an edge with zero cost, and capacity equal to the time slot capacity.

A. *Model formulations*

In the first model, we assume that any interface can be shared between any video at any time slot and transmission capacity can be freely shared among videos. The problem can be seen as Minimum Cost Flow Problem (MCFP), in which we look for the maximum flow that the network can carry, with the minimum total cost. With the second model, each interface and time slot can be assigned for transmission of a single video only with no capacity sharing. Variables become binary variables, equal to 1 if the edge is used to transfer data, 0 otherwise. Then, the problem falls into Bin Packing Problem (BPP).

B. *Heuristic Approaches*

We consider three simple and intuitive greedy heuristics: i) Greedy-in-time (GT) - This algorithm uploads all videos through all interfaces as soon as possible. In other words, the video with closest deadline is transmitted as soon as any interface has an available slot to upload (part of) the video. ii) Greedy-in-rate (GR) - This algorithm sorts time slots according to decreasing transmission rate, and schedules transmission through the highest-rate time slots. If rates are equal, earlier time slots are preferred. iii) Greedy-in-cost (GC) - This algorithm sorts time slots according to increasing cost, and schedules transmission through the cheapest time slots. If costs are equal, earlier time slots are preferred.

All heuristics stop when all videos are uploaded. The first greedy algorithm guarantees that the transfer is completed as soon as possible, while the second one minimizes the number of time slots to use. Both disregard the upload cost. Only the third algorithm explicitly considers the cost of using different interfaces at different times.

C. *Experiments*

We test the performance of the greedy algorithms and compare them against the optimal solution. To use a realistic setting, we adopt a trace-driven approach. The results in Fig. 2 refer to the total cost of a simple scenario, in which 2 videos have to be uploaded via 3 network interfaces. As can be expected, the GR algorithm incurs the highest upload cost, followed by GT. This is due to the fact that both those heuristics neglect the cost values. The GC algorithm provides cost values which are almost equal to those of the optimal solution.

The two main conclusions that we can draw from these results are: i) that GR is dominated by GT, since the latter provides lower cost, ii) that GC achieves practically the same cost (not lower, of course) as the optimal solution.

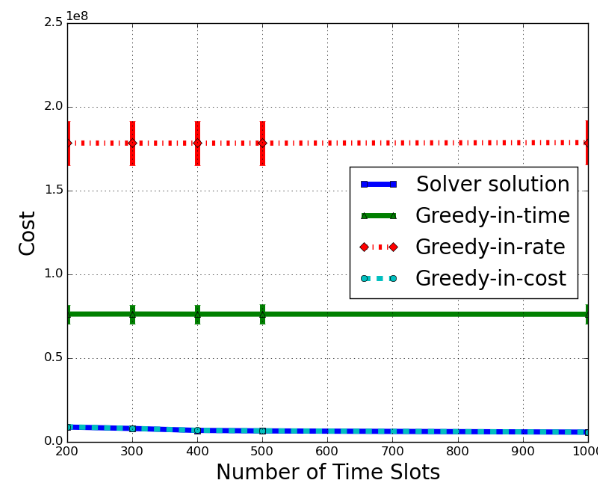

Fig. 2: Cost

III. Conclusions and Outlook

The further steps of the analysis will consider the variable nature of mobile networks, whose performance depends on parameters which cannot be predicted precisely. For instance, social events, congestion, network outages, vehicle changes of path, etc., can affect the actual upload performance. In this case, online algorithms that can dynamically compute the scheduling for the residual upload workload must be devised. Practical issues must also be faced, e.g., the impact of the transport protocol, the choice of video coding approaches, the granularity of time slots, the interference between simultaneous requests, etc. All of these features make this problem quite challenging, and worth further investigation.