Formal Guarantees and evaluation of a Event-Driven Bandwidth Allocation scheme for Camera Networks

Gautham Nayak Seetanadi, Martina Maggio, Karl-Erik Årzén

Department of Automatic Control, Lund University, Sweden

1. INTRODUCTION

Modern computing systems in which a multitude of devices compete for network resources suffer from performance issues derived from inefficient bandwidth allocation policies. This problem is often mitigated in bandwidth-constrained systems by introducing run-time device-level adaptations (e.g. adjustment of operation parameters) to ensure correct information transmission as shown in (Pedreiras and Almeida, 2003; Almeida et al., 2007). Adapting their behavior, the devices are capable of consequently adjusting their bandwidth requirements. Such scenarios present two main issues: (i) the adaptation at the device level can interfere with network allocation policies; (ii) it is quite difficult to obtain formal guarantees on the system's behavior, given that multiple adaptation strategies (network distribution and device-level adaptation) are active at the same time – and hence the presence of multiple independent control loops may lead to interference and result in disruptive effects (Heo and Abdelzaher, 2009).

In this work, we tackle the two aforementioned issues in bandwidth allocation, ensuring the satisfaction of formal properties like convergence to a steady state using model checking. We apply our method to a camera surveillance network, in which self-adaptive cameras compete for network resources to send streams of frames to a central node.

The cameras adapt the quality of the transmitted frames every time a new frame is captured. To ensure the satisfaction of control-theoretical properties, a network manager is triggered periodically to schedule network access, (Seetanadi et al., 2017b). The periodic solution is desirable because it is equipped with a formal guarantee of convergence of the system to a single equilibrium in which all cameras are able to transmit their frames, if this equilibrium exists. In the opposite case, the time-triggered action guarantees that no camera can monopolize the network.

Despite this desirable property, the periodic solution has also severe shortcomings that are mainly related to the choice of the triggering period. The system may be too slow in reacting to camera bandwidth requirements if the period of the manager is too large. On the contrary, the system may exhibit poor performance due to the overhead caused by unnecessary actions, if the network manager is triggered too frequently. These limitations can effectively harm the performance of the system and should be taken into account when designing a network allocation strategy. Based on these considerations, this paper introduces an event-triggering policy for the network manager that

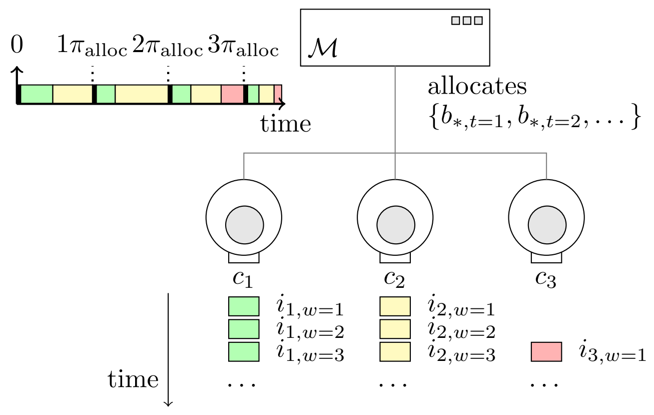

Fig. 1. System architecture.

minimizes the impact of the execution overhead on network performance, while taking into account the dynamic needs of the devices, induced by physical constraints and environmental factors – in our case study, for example, image size fluctuations derived from changes in the scenes captured by the cameras (Seetanadi et al., 2017a).

2. SYSTEM MODEL

The system is composed of a central node receiving video streams from a set of cameras, $\mathcal{C}_t = \{c_1, \dots, c_n\}$, where t represents the current time instant and n is the number of active cameras at time t. The central node also runs a network manager $\mathcal{M}$, in charge of distributing the available network bandwidth $\mathcal{H}$, e.g., $\mathcal{H} = 4Mbps$.

Figure 1 shows the system architecture when there are three cameras that capture frames and the network manager that determines the network access pattern for the cameras. In the network access timeline, black slots are used to show when the network manager uses the network, while the other colors represent the cameras transmitting frames. The third camera, c_3, is turned on when the first two, c_1 and c_2, have already transmitted two frames.

2.1 The camera

This subsection describes the behavior of the cameras where c_p with $p \in \{1, \dots, n\}$ denotes camera p. The camera captures a stream of frames. Each of these frames is compressed by an adequate encoderin our case MJPEG and sent to the central node via the network. The stream of frames is denoted by $I_p = \{i_{p,1}, \dots, i_{p,m}\}$, where p is the camera identifier and m is the cardinality of the set of frames (the longer the system runs, the more frames each camera produces). Each element $i_{p,w}$ in the set, $w \in \{1, \dots, m\}$, has the following characteristics.

The value $q_{p,w}$ represents the *quality* used for frame encoding, as in MJPEG. The quality is an integer number between 1 and 100, initialized using a parameter $q_{p,0}$, and loosely represents the percentage of information preserved during encoding. The value $\hat{s}_{p,w}$ indicates the estimate of the *size* of the encoded frame.

The relationship between the quality used for the encoding $q_{p,w}$, which can be changed by the camera, and the size of the resulting frame $s_{p,w}$ is rather complex . This complexity is explained by the many factors on which the relationship between quality and frame size depends, including but not limited to the scene that the camera is recording (e.g., the amount of artifacts in the scene), the sensor used by the camera manufacturer, and the amount of light that reaches the sensor. In this work we approximate this relationship using the following affine model

$$\hat{s}^*_{p,w} = 0.01 \cdot q_{p,w} \cdot s_{p,\max} + \delta s_{p,w}, \tag{1}$$

where $\delta s_{p,w}$ represents a stochastic disturbance on the frame size. We then use a standard control techniques(a PI Controller) to determine the quality to be applied to the next frame.

2.2 The network manager

To determine how to distribute the network bandwidth we use the approach proposed by (Maggio et al., 2013) for CPU allocation and extend it to handle network bandwidth allocation. The network has a fixed capacity $\mathcal{H}$. The network manager $\mathcal{M}$ is in charge of allocating a specific amount of the available network bandwidth to each of the cameras. For every instant of time t at which the network manager is invoked, $\mathcal{M}$ selects a vector $b_{*,w}$, whose elements sum to one.

$$\forall t, \mathcal{M} \text{ selects } b_{*,t} = [b_{1,t}, \dots, b_{n,t}] \text{ such that } \textstyle\sum_{p=1}^{n} b_{p,t} = 1 \tag{2}$$

The manager allows camera c_p to transmit data for the w-th frame for an amount of time that corresponds to the computed fraction of the Time-Division Multiple Access(TDMA) period $b_{p,t_{\mathcal{M},w}} \cdot \pi_{\text{alloc}}$. The total amount of data that c_p is allowed to transmit for the w-th frame is $B_{p,w}$.

$$B_{p,w} = b_{p,t_{\mathcal{M},w}} \cdot \pi_{\text{alloc}} \cdot \mathcal{H} \tag{3}$$

If the size of the encoded frame is greater than the amount of data that the camera can transmit, $s_{p,w} > B_{p,w}$, the frame is dropped. The network manager is periodically triggered with period $\pi_{\mathcal{M}}$, which must be a multiple of π_{alloc} and a parameter in our implementation. In its first invocation, at time 0, the manager equally divides the available bandwidth among the cameras. The following network manager interventions, happening at times $\{\pi_{\mathcal{M}}, 2\pi_{\mathcal{M}}, 3\pi_{\mathcal{M}}, \dots\}$ assign the bandwidth based on the following relationship,

$$b_{p,t+1} = b_{p,t} + \varepsilon \cdot \{-\lambda_{p,t} \cdot f_{p,t} + b_{p,t} \cdot \sum_{i=1}^{n} [\lambda_{i,t} \cdot f_{i,t}]\} \tag{4}$$

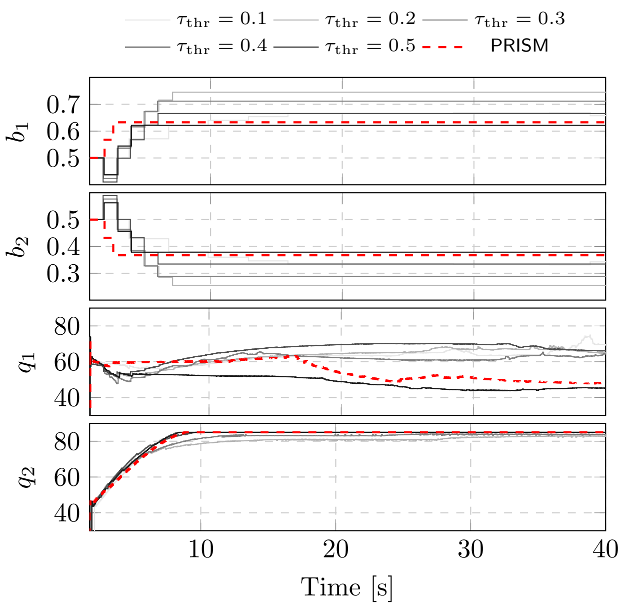

Fig. 2. Event-triggered activation with τ_{thr} and the optimal strategy synthesized by PRISM

3. RESULTS

Figure 2 shows the behavior of our implementation with an experimental result with three physical units: the network manager and two. Each unit runs Fedora 24. The first unit runs the network manager and has a Intel Core i7-4790, 8 core CPU with 32 GB RAM. The other units are off-the-shelf Logitech C270 cameras. [1]

REFERENCES

Almeida, L., Pedreiras, P., Ferreira, J., Calha, M., Fonseca, J.A., Marau, R., Silva, V., and Martins, E. (2007). Online QoS adaptation with the flexible time-triggered (FTT) communication paradigm. In *Handbook of Real-Time and Embedded Systems*.

Heo, J. and Abdelzaher, T. (2009). Adaptguard: Guarding adaptive systems from instability. In *6th ACM International Conference on Autonomic Computing*.

Maggio, M., Bini, E., Chasparis, G., and Årzén, K.E. (2013). A game-theoretic resource manager for rt applications. In *25th Euromicro Conference on Real-Time Systems*.

Pedreiras, P. and Almeida, L. (2003). The flexible time-triggered (FTT) paradigm: an approach to qos management in distributed real-time systems. In *International Parallel and Distributed Processing Symposium*.

Seetanadi, G.N., Camara, J., Almeida, L., Karl-ErikÅrzén, and Maggio, M. (2017a). Event-driven bandwidth allocation with formal guarantees for camera networks. In *RTSS, Real-Time Systems Symposium*.

Seetanadi, G.N., Oliveira, L., Almeida, L., Arzen, K.E., and Maggio, M. (2017b). Game-theoretic network bandwidth distribution for self-adaptive cameras. In *15th International Workshop on Real-Time Networks*.

[1] https://github.com/gauthamnayaks/camnetverification contains the code used for model checking