Design and Evaluation of a Flexible Advance Bandwidth Reservation Algorithm for Media Production Networks

Maryam Barshan, Hendrik Moens, Bruno Volckaert and Filip De Turck

Department of Information Technology, Ghent University – IMEC
Technologiepark-Zwijnaarde 15, 9052 Gent, Belgium
Email: maryam.barshan@intec.ugent.be

***Abstract*—In media production companies, exchanging large media files is daily business. Due to the predictable nature of network transfers in the media production industry, timeslot-based advance bandwidth reservation results in higher bandwidth utilization and improved network performance. Timeslot-based advance reservation can be based on flexible or fixed timeslot sizes. As the flexible approach is highly beneficial under bursty and limited network traffic conditions, this paper focuses on that approach. We design, implement and evaluate a novel algorithm based on flexible timeslots, taking into account the specific characteristics of media transfers and compare it with a fixed timeslot algorithm to quantitatively study the quality and complexity of both scenarios. We have defined a set of realistic media production use cases that serve as a basis for the evaluations. Results shows that the highest admittance ratio is consistently achieved by using the flexible time interval algorithm, while the execution time of this approach is up to 12 times lower, compared to the approach with fixed timeslot sizes.**

***Index Terms*—Advance bandwidth reservation, flexible timeslot size, media production networks.**

I. Introduction

In the media production industry, multiple actors must process large quantities of data at different geographical locations. Media production networks require an efficient collaboration between these distributed actors and offer predictable network traffic, making it possible to exploit this knowledge of future transmissions and use advance reservation services to improve the number of admitted requests and increase network utilization. In advance reservation approaches, to manage the time domain of reservations, timeslot-based solutions are introduced. Timeslot-based approaches can be deployed based on fixed-size or flexible timeslot sizes.

In our previous work, we have addressed this problem by proposing several timeslot-based advance bandwidth reservation algorithms [1], [2], taking into account the characteristics of requests in media production networks. To offer predictable complexity, easier implementation and regular reconfiguration of network devices, these algorithms were designed making use of fixed time slots. There are, however, some cases where using a fixed timeslot approach can negatively affect algorithm quality and execution speed. According to [3], deploying advance reservation approaches based on fixed and predefined timeslot sizes is inefficient when there are only few submitted requests, as the complexity of fixed approaches highly depends on timeslot granularity rather than the number of requests. As a consequence, we have analysed the benefits and drawbacks of using flexible timeslots from a theoretical point of view in [4], finding that the flexible approach is highly beneficial when dealing with bursty traffic conditions in a low-demand network with long-term downtimes. The flexible approach has the potential to significantly reduce the number of timeslots needed, resulting in execution speed improvements. In addition, using flexible timeslots could make the timings fit better with the timing of incoming requests, offering potential improvements in result quality.

In this work, we therefore design, implement and evaluate an advance bandwidth reservation algorithm based on flexible timeslot sizes and compare the quality and complexity of this approach with our previously designed fixed size advance reservation algorithm. The near-optimal SARA (Static Advance Reservation Algorithm), proposed in [2], has been extended to add the capability of offering flexible timeslot sizes.

The remainder of this paper is structured as follows. In Section II, we discuss related work. Section III, provides brief information about the media production industry and elaborates on the timeslot-based advance reservation approaches. The heuristic-based flexible advance reservation scheduling algorithm is described in Section IV. Section V provides simulation results, comparing the proposed algorithm to the fixed size approach. Finally, Section VI concludes the paper.

II. Related work

The advance reservation scheduling problem has been well studied in literature. While some have focused on rescheduling [5], [6], [7] and multi domain reservation [8], others particularly take into account real-world deployments [9], [10], [11], [12], and WDM optical networks [3]. Nevertheless, only two advance reservation algorithms [13], [14] support elastic reservation, and both consider fixed start time for the requests [3], while we consider flexible or unspecified start times. Advance bandwidth reservation for on-demand and flexible data transfer in scientific applications is investigated in [15]. However, they purely focus on data transfers, not video

978-3-901882-89-0 @2017 IFIP

streaming requests, the routing mechanism is based on single-path in contrary to our multi-path approach and dependency among different transfers is ignored. Flexible advance reservation for cloud resources has also been investigated by [16], [17].

This paper is in line with our previous works on media production network bandwidth reservation approaches. Since the combination of our requirements, like i.e. dependent requests, elastic reservations, different transfer types are not supported by alternative approaches, we first proposed optimal [1] and near-optimal advance bandwidth reservation algorithms [2], paying particular attention to the media production network transfers. These proposed approaches were based on a fixed size timeslot-based approach which is reported to be inefficient when the number of requests is limited [3]. This was the motivation of our recent work [4], in which a theoretical comparison between fixed size and flexible reservations is drawn to analyze which approach would be more appropriate for media production environments. The work presented in this paper differs from our recent work as the main focus of this work is to design, implement and evaluate a novel advance bandwidth reservation algorithm based on flexible time windows. The quality and complexity of the simulated flexible timeslot approach has been compared to the fixed size timeslot-based approach.

III. PROBLEM DESCRIPTION

A. Media production industry

In the media production industry, various actors involved in the media production process, such as recording studios, on-site filming crews, broadcasters, datacenters, etc. are connected to a shared wide-area network, consisting of interconnected routers or switches. The network supports the exchange of raw and encoded videos, both in the form of file-based (FB) and video streaming transfers (VS). We refer to each transfer as a *request*. With each request, we associate one or multiple paths from source to sink with a specific amount of reserved bandwidth. We assume that for FB requests, volume and for VS requests duration is always known. The allocated bandwidth for the video stream n must be equal to their required bandwidth demand, from the start time (t_s^n) until the end time (t_e^n), because their demand is fixed and non-variable. However, for the file-based request n, the volume of data is the determining factor. The file can be transferred whenever possible from the time the file is ready to be transferred (t_s^n) and must be fully transferred by its deadline (t_e^n). The residual demand of file-based videos is modified whenever a part of the video file is transferred.

In the media production industry, several actors are usually working in forms of projects, consisting of several interdependent video transfer requests. If one of those requests is not successfully transferred the whole project can be affected. This forms dependencies among different transfers. We refer to the set of all transfers of a project as a *scenario*. We assume that when multiple requests depend on each other, either all or none of them are admitted. This implies that, in case the deadline of even one request of a scenario cannot be guaranteed, the whole scenario will be rejected by the reservation interface, during the admission control phase.

This advance reservation based media production platform can be used in conjunction with Software Defined Networking (SDN) techniques, such as OpenFlow. We assume that the media production environment is a controlled and dedicated dark fiber network with an SDN-based centralized expert system, which offers a management layer. The management layer provides a reservation interface, that allows the users to submit their requests in order to reserve bandwidth over a specified or non-specified time period in the future.

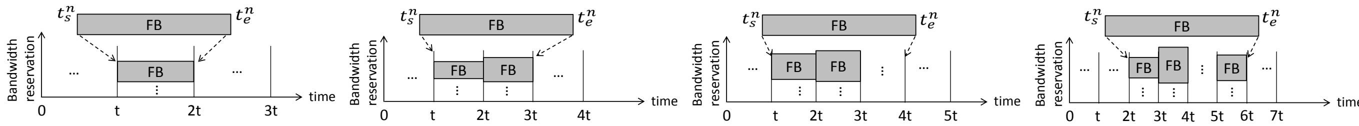
The advance bandwidth reservation algorithms are responsible for admission control and scheduling of submitted requests and reserving the required amount of bandwidth resources for all admitted requests, according to the agreed SLA. The output of the scheduling algorithms takes the form of a set of temporal routing policies (i.e., the paths associated with all requests over time) and bandwidth reservations (i.e., the amount of bandwidth resources to associate with each flow over time). This information can be transferred to the network controllers, that use it to configure the switches in the network. The controllers keep track of the temporal aspects of the policies, adjusting configurations whenever needed.

B. Time domains in advance reservation approaches

Timeslot-based approaches are introduced as an efficient solution for management of the time domain in advance reservation approaches [3], [18]. Based on these solutions, the entire time span is discretized into a set of timeslots. In each timeslot, aggregated information about network capacity consumption and network residual capacity is maintained. Timeslot-based solutions can be static with fixed size timeslots or dynamic with flexible time intervals. Static timeslot-based classification is easy to implement, due to a fixed number of predetermined-length timeslots. The complexity of the approach highly depends on the granularity of timeslot sizes and the amount of network state information is to some extent independent of the number of requests. In a dynamic timeslot solution, duration and number of future timeslots changes as soon as a new request enters into the reservation system. Therefore, the number of reservation requests in the network has a great impact on the complexity of these approaches.

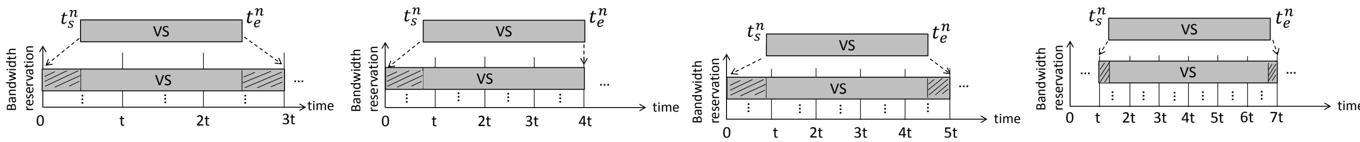
C. Restrictions of the fixed size timeslot-based approach

In literature, the static solution with fixed size timeslot sizes is followed by the majority of timeslot-based approaches. Fixed timeslots can, however, be inefficient for advance reservation systems with a small number of reservation requests, according to [3]. In this section, we distinguish the factors which restrict the capabilities of fixed size timeslot-based advance bandwidth reservation approaches in media production networks. We discuss how variable timeslot sizes can be more suitable for media production or similar industries. These restrictions are as follows:



(a) long timeslot size. (b) medium timeslot size. (c) short timeslot size. (d) tiny timeslot size.

Fig. 1: Impact of timeslot size for file-based requests in fixed size timeslot-based advance bandwidth reservation approach.



(a) long timeslot size. (b) medium timeslot size. (c) short timeslot size. (d) tiny timeslot size.

Fig. 2: Impact of timeslot size for video streaming requests in fixed size timeslot-based advance bandwidth reservation approach.

Request characteristics in media production networks: The first factor is related to the characteristics of requests in media production industries. Due to different characteristics of each type, a different behavior should be developed for different types of requests. To elaborate more on this, Figures 1 and 2 are depicted for FB requests and VS requests respectively with different timeslot sizes. These figures reveal how the reserved bandwidth is influenced by the size of timeslots in the fixed timeslot-based approach, depending on the type of request.

In Figure 1, the file can be transferred whenever possible from the time when the file is ready to be transferred (t_s^n) until its deadline (t_e^n). The start of a reservation for a file has to be restricted to the beginning of the next timeslot and the start of the timeslot in which the request deadline fits. For file-based requests, the volume of file is the determining factor. The allocated bandwidth can vary from one timeslot to another. The residual demand of file-based videos is modified whenever a part of the video file is transferred. There is no restriction for the amount of bandwidth allocation as long as its deadline is met and enough reservations have been made to fully transfer of the file. Moreover, there might even be no reservation for a file-based request in some timeslots during the requested period. There are two reasons for this. First, this may happen due to lower priority of this file compared to other concurrent requests and not having sufficient capacity in the physical network to service all requests, as can be seen in Figure 1d. The second reason is because enough reservations have already been made for the request, and therefore no reservation is needed in future timeslots (Figure 1c). In the fixed timeslot-based approach, these restrictions for the file-based requests imply that the file has a tighter time opportunity for transmission and therefore the probability of timely transfer is decreased.

In contrast to the FB requests, the allocated bandwidth for the video streams is constant and must equal to their required bandwidth demand, from the start time (t_s^n) to the end time (t_e^n), as their demand is constant for the entire reservation period. Figure 2 shows that in the fixed timeslot-based approaches, for the video streams the reservation has to be made from the start of the timeslot in which the start time of request (t_s^n) fits, until the end of the timeslot to which the request's end time (t_e^n) belongs.

It should be noted that the reservations based on the fixed size approach for video streaming requests lead to a waste of resources due to making unused reservations. These unused reservations start from the time when the reservations have been made until t_s^n and also from t_e^n until the end of reservations (shown as hachured areas in Figure 2). The size of timeslots has a direct impact on amount of these unused reservations. These figures also show that how the size of timeslots plays a significant role on the impact of these effects. Generally, the fine-grained timeslot size is expected to decrease the negative impact of fixed-size timeslots and therefore leads to better results, in terms of number of admitted requests. Our evaluation in [2] also verifies this. However, this is not the case for each individual request. To elaborate on this, compare Figure 2b and Figure 2c. Figure 2c is more fine-grained compared to Figure 2b. Nevertheless, the amount of unused reservations is also higher. This is not generally expected and highly depends on the timing requirement of the requests and how the request can fit within timeslots.

Another point is that the fixed size of timeslots is more restrictive when there are dependencies among different transfers, meaning that one request can only start when other requests on which this request depends, have been finished. This implies that even a small part of two interdependent requests can not be accommodated in one timeslot. In high-bandwidth networks with plenty of unused capacity, a chain of interdependent request may remain longer in the schedule compared to the flexible approach, which can be problematic for future requests. This impacts the request admittance ratio.

Contrary to the fixed size approach, the use of flexible time windows can eliminate these restrictions for both video streaming and file-based requests. Regardless of the type of request, the start and the end of time windows can be tuned up to the start and end time of each request. The interdependent requests can also be scheduled as soon as the dependencies have been eliminated without having to wait for the start of the next timeslot.

Delay prior to request processing: Predefined timeslot sizes imply that each new request arrival has to wait until the start of the next timeslot to be processed. This waiting time equals the timing gap between the request start time (t_s^n) and the start of the next timeslot. Although more fine-grained timeslot sizes can shorten the delay, it can be completely eliminated by deploying flexible time intervals.

Optimized timeslot size: In the fixed size advance reservation approaches, timeslot size is of great importance, because it has a high impact on the complexity and quality of the advance bandwidth reservation system. In the fixed size approaches, it is not trivial to find a good value for this. Nevertheless, this is not an issue with flexible timeslots.

Unnecessary periodic computations for long transfers: The periodic nature of the fixed timeslot-based solutions leads to unnecessary periodic computations for long-term video streaming requests and large video files. In the fixed size advance bandwidth reservation approaches, the residual demand of ongoing requests is periodically updated at each timeslot, and new and updated requests are periodically reallocated together. This issue causes unnecessary computational overhead, which becomes worse with very fine-grained timeslots. Again, this is not an issue in the flexible approach.

High computational complexity for long-term schedules: Another problem with the fixed timeslot-based approaches, specially with fine-grained timeslots, is that the computational complexity of these approaches mostly depends on the number of timeslots, making it impractical or at least unrecommended for long-term schedules, e.g. 1 week or longer.

IV. Flexible vs. fixed timeslot size advance bandwidth reservation algorithms

In our previous work [2], we proposed the SARA approach based on the fixed size timeslot-based scheme. In this section, we extend this approach to incorporate the capability of flexible timeslot mechanism.

We briefly explain the SARA approach and how this approach is extended to support variable timeslot sizes. For more detailed explanations about fixed size timeslot-based SARA solutions we refer to [2].

1) SARA (Static Advance Reservation Algorithm): In the SARA approach, first the scenarios are sorted. This sorting is based on the earliest average start time of the scenario's requests. If two scenarios have the same value, the one requiring more resources is chosen. Then each scenario in the sorted list is sequentially processed as follows. The prioritization algorithm assigns priorities to the scenario's requests, taking two parameters into account: the estimated hard deadline and the volume. Since the deadline may not be specified for all requests, the hard deadline (i.e., the latest possible

```
input: scenarios' requests, network infrastructure,
       approach
sortedQueue ← AverageStartSort(all scenarios);
for (scenario ∈ sortedQueue) do
    Set scenario status as Pending;
    currentstate ← Save the current system state;
    Prioritization(scenario's requests);
    sysReqList.Add(scenario's requests);
    if (approach = Fixed) then
        feasible ← FixedTimeSlot(sysReqList,
        timeslotSize);
    else
        feasible ← FlexibleTimeSlot(sysReqList);
    end
    if (feasible) then
        Update the scheduling;
        Set scenario status as Admitted;
    else
        Set current system state to CurrentState;
        Set scenario status as Rejected;
    end
end
```

Algorithm 1: SARA (Static Advance Reservation Algorithm), updated to support the capability of offering flexible timeslot sizes.

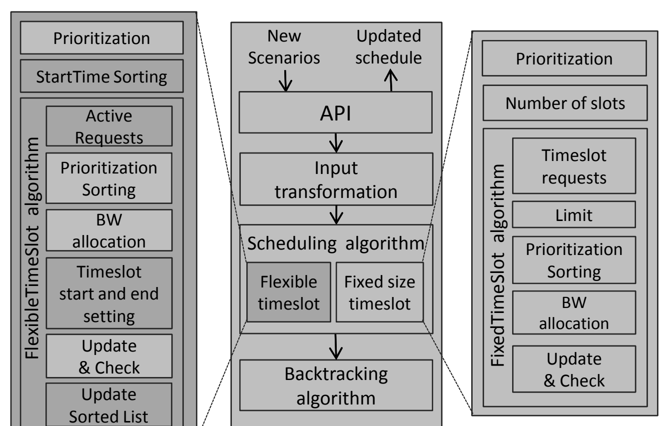


Fig. 3: Components of fixed-size and flexible scheduling algorithms of advance bandwidth reservation system.

deadline) for those with no specific deadline is estimated. This time is calculated by assuming that all requests on which the request depends, use the entire network at once. This gives the latest possible deadline for the requests. In the prioritization algorithm, the sooner deadline has the higher priority and volume comes into consideration only when the hard deadlines are equal, where a higher priority is assigned to larger demands. The scenario's requests are added to the list of system requests (*sysReqList*). Then, according to the desired solution, the flexible or fixed timeslot-based approach is chosen and based on the result of either the *FixedTimeSlot* or the *FlexibleTimeSlot* algorithm, SARA decides to admit or

```
input: sysReqList, timeSlotGraphs, timeslotSize
NumberofIntervals ← scheduleDuration/timeslotSize;
for (t ∈ Time Intervals) do
    currentReq ← TimeSlotRequests(t,sysReqList);
    if (currentReq ≠ ∅) then
        Limit(currentReq);
        sortedList ← PrioritySorting(currentReq);
        reservation ← BWallocation(sortedList);
        if (!UpdateAndCheckFeasibility(reservation))
        then
            return false;
        end
    end
end
return true;
```

Algorithm 2: The FixedTimeSlot algorithm.

reject the scenario. If a feasible schedule has been achieved, the previous reschedule is updated, otherwise the algorithm has to backtrack to the previous feasible state.

2) FixedTimeSlot algorithm: The FixedTimeSlot algorithm, shown in Algorithm 2, iterates over the timeslots and consists of the following components for each time interval.

TimeSlotRequests: determines which unserved requests can be served in the current timeslot. Independent requests are added to the list of current requests if the current interval is greater or equal to the request start time. For requests with start time dependencies, the algorithm checks whether the requests on which this request depends are finished or not. These requests can be added provided that all the requests on which the request depends are fulfilled.

Limit: This is where the size of the timeslot impacts the bandwidth allocation for file-based requests. The limit component determines the maximum amount of bandwidth reservations for each request in each timeslot. The limit for the file-based requests is calculated as follows: the residual volume of this file, which is modified whenever a part of a video file is transferred, is divided by the size of timeslot, in order to avoid the extra reservation for the requests. The limit for the video streams is their required demand, because their demand is fixed and non-variable.

PrioritySorting: sorts the requests based on their priorities, which have already been calculated by prioritization component.

BWallocation: depending on the type of requests, two different bandwidth allocation algorithms are designed for video streams and video files because their requirements are dissimilar. Details of these algorithms can be observed from [2].

UpdateAndCheckFeasibility: updates the requests requirements and checks the feasibility of the results. For any request, if the hard deadline has not been met, rescheduling is infeasible.

3) FlexibleTimeSlot algorithm: The FlexibleTimeSlot algorithm, shown in Algorithm 3, consists of the following components.

StartTimeSorting: All requests submitted to the reservation system are chronologically sorted based on their start time and stored in *sortedSysReqList*. This list does not contain the request with unfulfilled dependencies, because their start time is not specified. These dependent requests have to wait. Then, the FlexibleTimeSlot algorithm jumps to the start time of the earliest request in the sortedSysReqList, sets the end of the current timeslot and the start of the next timeslot to the start time of the earliest request.

ActiveRequests: This component sequentially looks for any other requests in the sortedSysReqList which can be started simultaneously in the current timeslot and keeps these requests in the *currentReq* list. For the requests with start time dependencies, the algorithm checks if the requests' dependencies have been eliminated. This implies that all the other requests on which this request depends, have already been scheduled.

BWallocation: This algorithm is similar to the algorithm in FixedTimeSlot approach but does not take any limitations into account during the bandwidth allocation for the file-based requests.

MinDuration: The duration of the current timeslot is calculated by this algorithm. The size of timeslot is determined as the earliest time either an active request is finished or a new request is started. As soon as the timeslot duration is determined, the end of the current timeslot can be set.

UpdateAndCheckFeasibility: As duration of timeslots is not predefined, this component has been modified (compared to the same component in the FixedTimeSlot algorithm) to take into account the calculated size of the timeslot when updating the requests' demands and checking the feasibility of the schedule.

Update the sortedSysReqList: All the admitted and scheduled requests have to be removed from the sortedSysReqList.

V. Performance evaluation

This section evaluates the proposed flexible timeslot-based scheduling algorithms and compares the quality and execution time of this approach to the fixed-size algorithm. For this analysis, the SARA approach is evaluated in which all the requests are known in advance, before the start of scheduling. In the fixed size timeslot-based solution, timeslot granularities of 5 minutes to 60 minutes are used. The influence of the available bandwidth and network load and their execution times are assessed.

A. Evaluation setup

The media production network topology used for this evaluation contains 8 nodes including service provider, production studio and broadcaster and 5 other random locations and 16 bidirectional links. After discussion with our industrial partners, 3 scenario types are defined: a soccer after-game discussion program, an infotainment show and a news broadcast program, consisting of 5, 18 and 8 interdependent file-based and video streaming requests respectively. Each request

input: sysReqList
sortedSysReqList ← StartTimeSorting(sysReqList);
New Timeslot.setStart(FirstReqStartTime);
while *(sortedSysReqList ≠ ∅)* **do**
  currentReq ← ActiveRequests(sortedSysReqList);
  sortedList ← PrioritySorting(currentReq);
  reservations ← BWallocation(sortedList);
  duration ← MinDuration(reservations);
  timeSlotEnd ← duration + Timeslot.getStart();
  Timeslot.setEnd(timeSlotEnd);
  if *(!UpdateAndCheckFeasibility(reservations, timeSlotEnd))* **then**
    return false;
  else
    New Timeslot.setStart(timeSlotEnd);
    Update the sortedSysReqList;
  end
end
return true;

Algorithm 3: The FlexibleTimeSlot algorithm.

is represented with a source node, a destination node, the start time for video streams or the time when the data is ready to be transferred for file-based requests, the deadline for file-based requests or fixed end-time for video streams, the volume for file-based requests or duration and the bandwidth requirement for video streams. Several instances of each type are generated, based on randomized input parameters. A detailed overview of the randomized variables of requests and network topology can be observed from [2] and [19] respectively. Throughout this section, SARA[XXmin] denotes that timeslot size of *XX* minutes is used in the fixed-size timeslot-based advance reservation algorithm. Each simulation run covers a 24-hour period. All results are averaged over 20 runs with different randomized inputs, error bars denote the standard error.

B. Evaluation results

In Figures 4 and 5, the media network infrastructure has been configured for different available bandwidths, respectively for heavy and light network traffic conditions, to investigate the impact of available network capacity on the performance of our algorithms. The number of admitted requests in the SARA approach for different timeslot granularities, varying from 5-minute to 1-hour sizes, are evaluated. In Figure 4, 20 scenarios (in total 209 interdependent requests) are submitted to the bandwidth reservation system and the network capacities vary from 200 Mbps to 500 Mbps. As can be seen in this figure, the highest percentage of admitted requests is achieved by the flexible advance scheduling algorithm. In the fixed-size approaches the longest timeslot size of 1 hour shows the worst performance in terms of number of admitted requests and this quality is improved when the time interval size is more fine-grained. Comparing to the best results obtained by near-optimal fixed size timeslot-based approach (i.e. 5 minutes), the flexible timeslot size SARA approach provides up to 0.43% higher percentage of admitted requests.

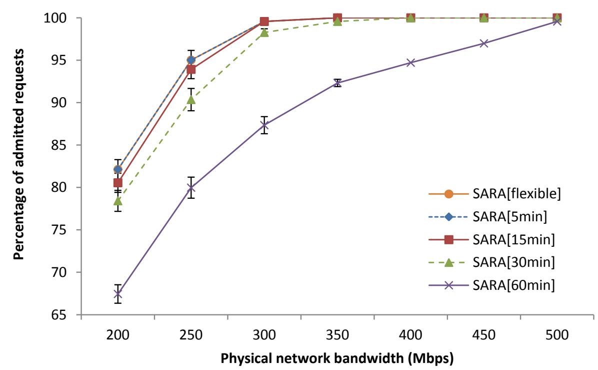


Fig. 4: Comparing the performance of flexible and fixed timeslot-based SARA approach with different timeslot sizes in the 8-node topology (20 iterations). The number of requests is 209.

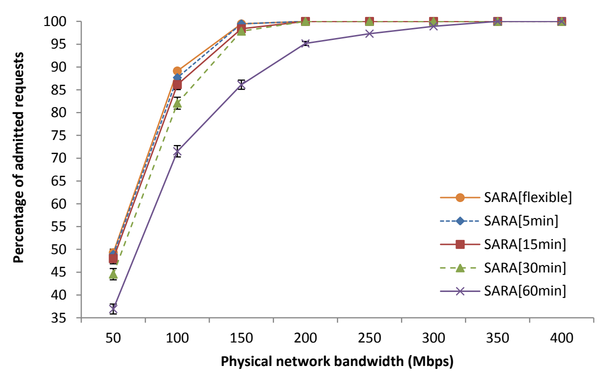


Fig. 5: Comparing the performance of flexible and fixed timeslot-based SARA approach with different timeslot sizes in the 8-node topology (20 iterations). The number of requests is 67.

In Figure 5, the same trend can be seen for 67 interdependent requests (7 scenarios). Bandwidth capacity per link varies from 50Mbps to 400Mbps. In this evaluation the flexible timeslot-based approach is able to achieve up to 1.46% improvement in request admittance ratio, compared to a 5-min timeslot size in the fixed size approach.

As shown in Figures 4 and 5, comparing the fixed size experiments, the fine-grained experiment with the shortest timeslot size results in the highest request admittance ratio. However, the execution time of the algorithm also increases. As can be seen in Figure 6, the fixed-size SARA approach with 1-hour timeslot granularity is between 12.3 up to 16.7 times faster than the solution with 5-minute timeslot sizes. Nevertheless, the flexible timeslot size in addition to providing the highest admittance ratio, shows an acceptable execution time. In this figure, its execution times are similar to that of the 15-minute fixed size approach, while its acceptance rate is

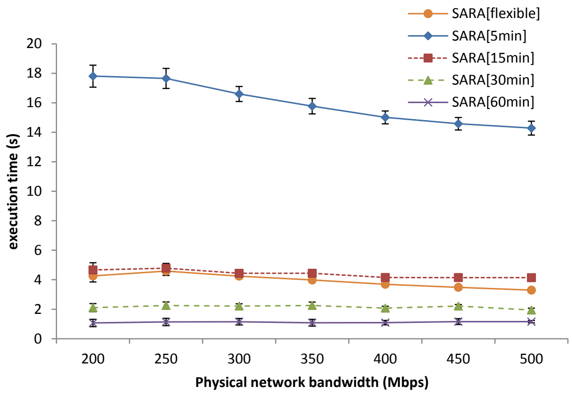


Fig. 6: Comparing the execution time of flexible and fixed timeslot-based SARA approach with different timeslot sizes in the 8-node topology (20 iterations). The number of requests is 209.

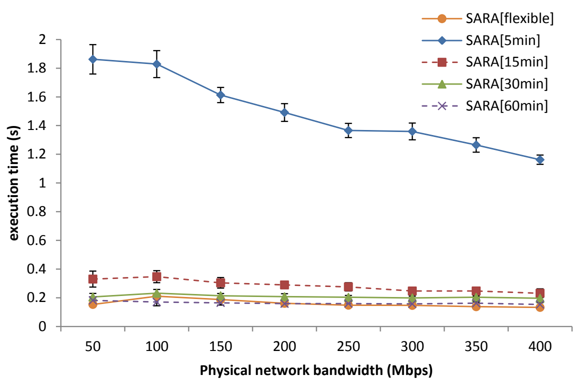


Fig. 7: Comparing the execution time of flexible and fixed timeslot-based SARA approach with different timeslot sizes in the 8-node topology (20 iterations). The number of requests is 67.

up to 1.6% higher.

In Figure 7, the same trend is observed when the number of requests is decreased to 67. This figure reveals that for a smaller number of reservation requests, not only the highest number of admitted requests is achieved by the flexible timeslot-based approach, but it also achieves the lowest execution time in more than half of the experiments.

Figure 8 compares the request admittance ratio for the fixed size and flexible approaches when the network load increases, from 2 to 20 scenarios, showing that in all experiments, flexible timeslots on average show the most desirable performance specially when the number of requests increases, up to 0.81% higher percentage of admitted requests, compared to the fixed 5-min timeslot size.

The execution time of both approaches are also compared in Figure 9 in function of number of scenarios. This figure shows that in the fixed size approach, in addition to the number of scenarios, size of timeslot has a great impact on the execution

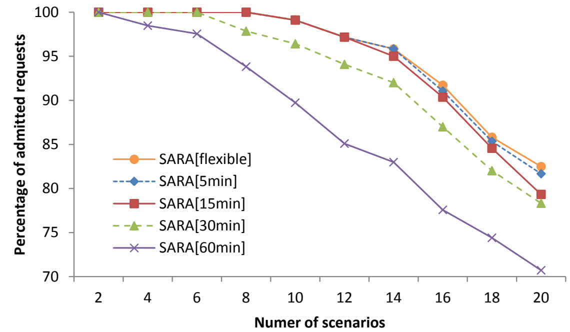


Fig. 8: Comparing the performance of flexible and fixed timeslot-based SARA approach with different number of scenarios in the 8-node topology (20 iterations). Network capacity is 200Mbps.

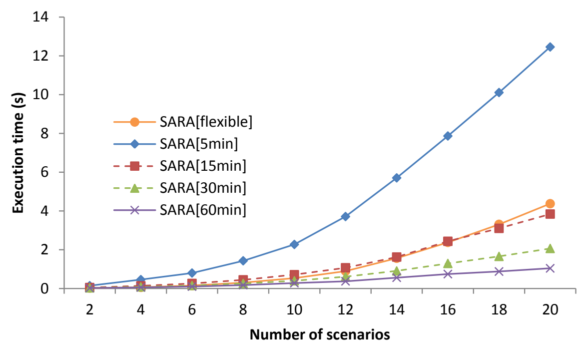


Fig. 9: Comparing the execution time of flexible and fixed timeslot-based SARA approach with different number of scenarios in the 8-node topology (20 iterations). Network capacity is 200Mbps.

time. However, in the flexible approach an increase in the number of scenarios leads to a large number of timeslots, resulting in a steeper increase when the number of scenarios grows. This is further illustrated in Figure 10, which shows the number of timeslots in flexible and fixed size advance reservation mechanisms in a 24-hour timespan. In the fixed size approaches, the number of timeslots is constant when the number of scenarios grows. However, in the flexible approach, as the number of timeslots depends on the number of requests, increasing the number of requests leads to an increase of the number of timeslots. In the flexible approach, timeslots are started with any request start time and end with either the arrival of a new request or the earliest end time of current requests. As such, from a theoretical point of view, worse

TABLE I: Correspondences between the number of scenarios and the number of requests.

# Scenarios	**2**	**4**	**6**	**8**	**10**	**12**	**14**	**16**	**18**	**20**
# Requests	23	36	62	85	98	124	147	160	186	209

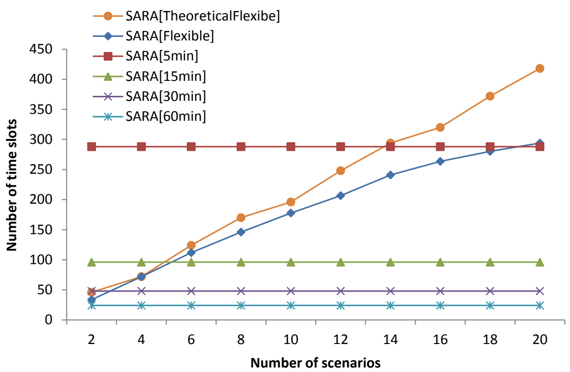


Fig. 10: Comparing the number of timeslots (in 24 hours) in function of number of requests in fixed and flexible size timeslot-based advance bandwidth reservation approaches.

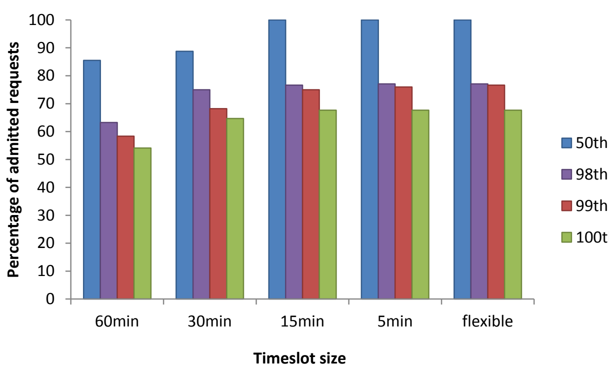


Fig. 11: Comparing the percentiles in the flexible and fixed size advance reservation approaches in the 8-node topology (1000 iterations). Number of scenarios is 14 (147 requests) and network capacity is 200Mbps.

case the number of time slots is twice of the number of reservation requests. However, Figure 10 shows that in practice the number of timeslots is on average 16.76% lower.

Figure 11 compares the 50th, 98th, 99th and 100th percentiles of request admittance ratio in the flexible and fixed size advance bandwidth reservation approaches for 14 scenarios consisting of 147 requests. The 8-node topology is used and the number of iterations is 1000. This figure reveals that the flexible approach in 99% of cases outperforms the fixed size approach for at least 0.62%. For the 50th, 98th and 100th percentiles the flexible approach behaves statistically identical to the fixed 5-minute approach.

VI. Conclusions

This paper aimed at investigating the possibility of using a flexible timeslot-based advance bandwidth reservation approach in media production or similar industries. We showed that flexible timeslots are by nature more compatible with requests in such networks. A novel flexible algorithm is designed, implemented and evaluated to compare the quality and complexity of the proposed algorithm with our previously designed fixed size approach. Our simulation studies prove that flexible time windows not only result in slightly higher request admittance ratio, up to 1.46%, but also execute faster, up to 12 times, compared to the execution time of the best result achieved by the fixed size approach, i.e. with 5-minute time intervals.

In future work, we intend to work on a hybrid approach to provide a trade-off between the number of timeslots and quality of the results to address the situations where a large number of requests would result in an excessive number of timeslots and the flexible approach is defeated due to uncontrolled complexity.

Acknowledgment

The research leading to these results has been performed within the context of ICON MECaNO. It is a project co-funded by iMinds, a digital research institute founded by the Flemish Government. Project partners are SDNsquare, Limecraft, VideoHouse, Alcatel-Lucent, and VRT, with project support from IWT under grant agreement no. 130646.

References

[1] M. Barshan, H. Moens, J. Famaey, and F. De Turck, "Algorithms for advance bandwidth reservation in media production networks," in *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, pp. 183–190, IEEE, 2015.

[2] M. Barshan, H. Moens, J. Famaey, and F. De Turck, "Deadline-aware advance reservation scheduling algorithms for media production networks," *Computer Communications*, 2015.

[3] N. Charbonneau and V. M. Vokkarane, "A survey of advance reservation routing and wavelength assignment in wavelength-routed wdm networks," *IEEE Communications Surveys & Tutorials*, vol. 14, no. 4, pp. 1037–1064, 2012.

[4] M. Barshan, H. Moens, B. Volckaert, and F. De Turck, "A comparative analysis of flexible and fixed size timeslots for advance bandwidth reservations in media production networks," in *NOF 2016-7th International Conference on Network of the Future*, IEEE, 2016. submitted.

[5] K. Rajah, S. Ranka, and Y. Xia, "Advance reservations and scheduling for bulk transfers in research networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 20, pp. 1682–1697, Nov. 2009.

[6] C. Xie, H. Alazemi, and N. Ghani, "Rerouting in advance reservation networks," *Computer Communications*, vol. 35, no. 12, pp. 1411–1421, 2012.

[7] L. Zuo, M. M. Zhu, and C. Q. Wu, "Fast and efficient bandwidth reservation algorithms for dynamic network provisioning," *Journal of Network and Systems Management*, 2013.

[8] H. Alazemi, F. Xu, C. Xie, and N. Ghani, "Advance reservation in distributed multi-domain networks," *IEEE Systems Journal*, 2013.

[9] C. Guok, E. N. Engineer, and D. Robertson, "Esnet on-demand secure circuits and advance reservation system (oscars)," *Internet2 Joint*, 2006.

[10] B. Gibbard, D. Katramatos, and D. Yu, "Terapaths: end-to-end network path qos configuration using cross-domain reservation negotiation," in *Broadband Communications, Networks and Systems, 2006. BROADNETS 2006. 3rd International Conference on*, pp. 1–9, IEEE, 2006.

[11] J. Gu, D. Katramatos, X. Liu, V. Natarajan, A. Shoshani, A. Sim, D. Yu, S. Bradley, and S. McKee, "Stornet: Integrated dynamic storage and network resource provisioning and management for automated data transfers," in *Journal of Physics: Conference Series*, vol. 331, p. 012002, IOP Publishing, 2011.

[12] S. Sharma, D. Katramatos, D. Yu, and L. Shi, "Design and implementation of an intelligent end-to-end network qos system," in *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*, SC '12, (Los Alamitos, CA, USA), pp. 68:1–68:11, IEEE Computer Society Press, 2012.

[13] S. Naiksatam and S. Figueira, "Elastic reservations for efficient bandwidth utilization in lambdagrids," *Future Generation Computer Systems*, vol. 23, no. 1, pp. 1–22, 2007.

[14] L.-O. Burchard, H.-U. Heiss, and C. A. De Rose, "Performance issues of bandwidth reservations for grid computing," in *Computer Architecture and High Performance Computing, 2003. Proceedings. 15th Symposium on*, pp. 82–90, IEEE, 2003.

[15] M. Balman, E. Chaniotakisy, A. Shoshani, and A. Sim, "A flexible reservation algorithm for advance network provisioning," in *2010 ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis*, pp. 1–11, IEEE, 2010.

[16] H. Bai, F. Gu, K. Shaban, J. Crichigno, S. Khan, and N. Ghani, "Flexible advance reservation models for virtual network scheduling," in *Local Computer Networks Conference Workshops (LCN Workshops), 2015 IEEE 40th*, pp. 651–656, IEEE, 2015.

[17] M. A. Netto, K. Bubendorfer, and R. Buyya, "Sla-based advance reservations with flexible and adaptive time qos parameters," in *International Conference on Service-Oriented Computing*, pp. 119–131, Springer, 2007.

[18] C. Barz, U. Bornhauser, P. Martini, and M. Pilz, "Timeslot-based resource management in grid environments," in *IASTED Conference on Parallel and Distributed Computing and Networks, PDCN*, 2008.

[19] M. Barshan, H. Moens, and B. Volckaert, "Dynamic adaptive advance bandwidth reservation in media production networks," in *2016 IEEE NetSoft Conference and Workshops (NetSoft)*, pp. 58–62, IEEE, 2016.