



An Optimized Resilient Advance Bandwidth Scheduling for Media Delivery Services

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

► To cite this version:

Maryam Barshan, Hendrik Moens, Bruno Volckaert, Filip De Turck. An Optimized Resilient Advance Bandwidth Scheduling for Media Delivery Services. 11th IFIP International Conference on Autonomous Infrastructure, Management and Security (AIMS), Jul 2017, Zurich, Switzerland. pp.79-93, 10.1007/978-3-319-60774-0_6 . hal-01806052

HAL Id: hal-01806052

<https://inria.hal.science/hal-01806052>

Submitted on 1 Jun 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

An Optimized Resilient Advance Bandwidth Scheduling for Media Delivery Services

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

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

Abstract. In IP-based media delivery services, we often deal with predictable network load and traffic, making it beneficial to use advance reservations even when network failure occurs. In such a network, to offer reliable reservations, fault-tolerance related features should be incorporated in the advance reservation system. In this paper, we propose an optimized protection mechanism in which backup paths are selected in advance to protect the transfers when any failure happens in the network. Using a shared backup path protection, the proposed approach minimizes the backup capacity of the requests while guaranteeing 100% single link failure recovery. We have evaluated the quality and complexity of our proposed solution and the impact of different percentages of backup demands and timeslot sizes have been investigated in depth. The presented approach has been compared to our previously-designed algorithm as a baseline. Our simulation results reveal a noticeable improvement in request acceptance rate, up to 9.2%. Moreover, with fine-grained timeslot sizes and under limited network capacity, the time complexity of the proposed solution is up to 14% lower.

Keywords: Advance bandwidth reservation, resilient reservation, fixed timeslot size, media delivery service.

1 Introduction

Currently, in the media-centric industries, the distribution of media content is generally performed by either people transporting the content on a physical storage media or over dedicated point-to-point high-speed optical links. However, these are highly inefficient and costly methods. In order to support decentralized collaboration, reduce capital expenditures and increase network resource utilization, media related environments tend to switch to the cost-effective IP-based WAN approaches. Deploying a shared IP-based WAN solution enables the existing media content owners and their collaborators to work together in a cost effective way, while new actors can more easily find new collaboration opportunities, thus fostering the whole industry’s further growth.

As media-centric networks usually offer predictable network traffic, this knowledge of future transmissions can be exploited to use advance reservation (AR) services. This makes it easier to offer guarantees in advance, improves the number of admitted requests and increases network utilization. In AR techniques, users submit requests for future data transfers, generally encompassing a start time in the future, a deadline, and total data transfer size or rate. To allocate the necessary resources (more specifically network bandwidth), a scheduling algorithm is needed to ensure that all admitted requests finish before their specified deadline, while admitting as many requests as possible. Clearly, AR has advantages for next generation media related networks: it allows network operators to better plan resource usage, leading to greatly increased resource utilization and guaranteed Quality of Service (QoS).

Reliability of the transport is also of crucial importance in the digital-centric media transfer process, when different media actors are geographically located far apart. Therefore, strategies to deal with network dynamics such as failures should be defined to enable reliable transmission of accepted requests without any loss in QoS upon occurrence of a failure. For example, in media production networks, meeting transfers' deadlines is of crucial importance. Consider a live show or a news program which is broadcasted everyday at a specific time. Clearly, even slight delays in transfer of pre-production contents are intolerable in such a setting.

Media-centric networks impose requirements not supported by existing AR scheduling techniques, such as different types of video or audio transfers, flexible or unspecified start or end times, strict deadlines, interdependent requests, reliability, etc. Addressing these unexplored aspects was the main focus of our previous contributions. First optimal and near optimal AR scheduling algorithms, customized for media production networks, have been proposed [1]. To offer reliable reservations, we have further presented the resilient version of our approach based on a protection mechanism to improve the reliability of the AR systems [2]. The proposed scheme is capable of covering single link failures using pre-reserved disjoint backup paths. Additionally, the resilient solution improves the scheme's availability compared to the non-resilient approach.

Continued research has shown us that the resilient bandwidth allocation algorithm, in [2], can be further improved and this is the main contribution of the work presented in this paper. This algorithm is optimized to reduce network reservation waste by proposing a more efficient solution to finding an optimal allocation of bandwidth for each file transfer request. We have made a tradeoff between the complexity and performance of the resilient bandwidth allocation algorithm for file-based transfers. This results in better network utilization and consequently higher request admittance ratio.

The remainder of this paper is structured as follows. In Section 2, we discuss the related work. Section 3, provides brief information about the media delivery services and elaborates on the resilient AR approach. The proposed solution to improve the performance of resilient AR scheduling approach is described in

Section 4. Designed algorithms are explained in Section 5. Section 6 provides simulation results and finally, Section 7 concludes the paper.

2 Related work

The authors in [3] survey the AR algorithms, mostly focusing on Wavelength-Division Multiplexing (WDM) networks, and provide a taxonomy for classifying these algorithms. Advance reservation requests can be classified in 4 individual categories, which are also valid for different types of requests in media related networks. However, based on this survey, only two works offer variable-bandwidth reservation in their scheduling process [4, 5]. While both approaches consider a fixed start time for the requests, all four classes for requests with specified or unspecified time and duration are supported in our work. Current research on AR scheduling mostly focuses on rescheduling [6–8], multi-domain reservations [9], and real-life deployments [10–13]. Nevertheless, reliability and fault tolerance properties have not been investigated.

Resilient AR systems can be deployed either through restoration or protection failure recovery mechanisms [14]. In protection approaches, backup resources are reserved in advance before any failure happens in the network, while in case of restoration backup resources are found upon failure detection. The former results in more resource consumption but the recovery time is quite fast. In [15], the authors propose a restoration technique to deal with link failures. In their work, the active requests and the scheduled requests for the future which are affected by a failure are restored. In [16, 17], optimal ILP-based solutions were proposed to provide shared and dedicated path protection. Authors in [18, 19] also provide resiliency through shared path protection. Since meeting strict deadlines and QoS requirements is of great importance in our approach, we have made use of protection mechanisms.

The work detailed in this paper presents a significant optimization to our previous works on bandwidth reservation approaches [1, 2]. In [1], we proposed a theoretical Integer Linear Programming (ILP) based model and heuristic algorithms for advance bandwidth reservation with no support for failure recovery. In [2], the media production reservation system is enhanced by following a protection mechanism and provisioning backup reservations for each request. This resilient solution guarantees 100% recovery against any single link failure. While this approach strives to minimize the needed bandwidth for the backup paths and determination of allocations is fast it does not fully utilize the network capacity. This work aims at optimizing the resilient solution proposed in [2] and improving the request admittance ratio by reducing wasted network capacity and thus improving network utilization.

3 Background

We briefly explain about the specific properties of media delivery networks and provide a summary of the resilient advance bandwidth reservation approach.

3.1 Media delivery networks

In the digital-centric media related industries, various actors are connected to a shared wide area network to build a collaboration over IP media contents. This shared network supports the exchange of different media contents, e.g. raw and encoded video and audio files and streaming transfers. We refer to each transfer as a *request*. A request can have a fixed or unspecified start and end times. Media delivery requests, supported in our work, are of 4 different classes: independent streaming requests, independent file transfers, dependent streams and dependent file based transfers. The requests of independent type can be started at the specified start time but dependent requests have to wait until the requests upon which they are dependent have finished. Dependency among different transfers implies that either all or none of the interdependent requests must be admitted. We refer to a set of interdependent requests as a *scenario*.

We assume that *volume* for file-based requests, and *duration* for streaming requests duration must be specified. The allocated bandwidth for the streams must be equal to their required bandwidth demand, from the start time to the end time, because their demand is fixed. However, for file-based requests, the volume of file is the determinative factor. The file can be transferred whenever possible from the time the file is ready to be transferred till its deadline. The residual demand of file-based transfers is modified whenever a part of the file is transferred.

3.2 Resilient AR scheduling

In order to have a quick response to sudden changes such as failures in the network, we use a protection mechanism which finds backup paths for connections in advance, before the occurrence of any failure to ensure there is enough capacity left when failures occur. The objective is to minimize the resource usage by the protection paths while full recovery is guaranteed against any single link failure. In this scheme, first the primary paths for a given request are determined using an advance reservation algorithm which we presented in [1]. Then disjoint backup paths are found corresponding to these primary paths [2]. Note that in the proposed schemes, the user can indicate the amount of required backup for each request. This way, the higher priority requests can be fully protected while the ones with lower priority can remain partially protected or unprotected.

Using shared backup path protection (SBPP) [19] and multi-path approaches, to provide full protection against a single link failure, the backups have to provide the maximum bandwidth allocated on the links of the primary paths. In the resilient approach, when bandwidth is allocated to a request, we look for disjoint paths to be reserved as backup paths for that request. In practice, a request may not ask for full recovery of bandwidth demand upon occurrence of a failure and it is sufficient that a portion of the request demand is transmitted to the destination. Therefore, we compare the maximum primary allocation and the amount of requested backup and select the smaller value as the limit to be fulfilled by the backups. If this backup limit can not be found, it backtracks to

the initial state and retries the bandwidth allocation with half of the primary bandwidth demand. This division by 2 is repeated until both primary and backup demands are satisfied. If the file can not be accommodated by its deadline, the scenario to which this request belongs, is rejected and all of its reservations have to be sent back to the network resource pool. Full details can be found in [2].

3.3 Resilient AR scheduling architecture

In this section, we briefly explain the heuristic-based resilient AR architecture, detailed in [2], for the reliable bandwidth scheduling problem. There are different blocks in this approach, presented in Figure 1. As can be seen, new scenarios enter the scheduler through an API which can be transformed using the input transformation block. Then the resilient scheduling algorithm is invoked which follows a timeslot-based scheme. A timeslot is defined as a period of time in which reservations remain invariant. The new scenario is admitted and the schedule is updated provided that this algorithm succeeds in allocating bandwidth to all the scenario's requests. Otherwise, the previous scheduling remains unchanged.

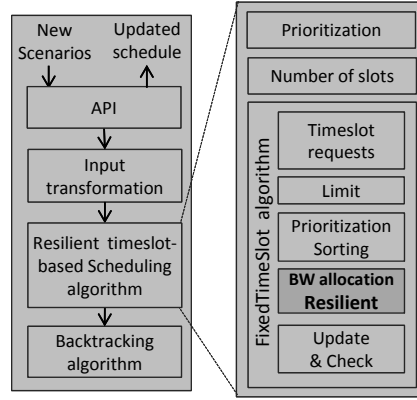


Fig. 1: Components of the resilient advance bandwidth reservation approach.

In the resilient scheduling component, first the requests within the scenario are prioritized and then the *FixedTimeSlot* algorithm is called. In the prioritization algorithm, the requests' priorities are assigned first based on the deadline, and then the request's demand. Requests with sooner deadline and higher demand receive higher priorities.

The *FixedTimeSlot* algorithm iterates over the time slots with 5 sub-modules. 1) *TimeSlotRequests*: determines the eligible requests which can be served at the current time slot. 2) *Limit*: defines a limitation for each request. For the streams this limit is equal to their demand which is fixed. For file-based requests the residual demand divided by timeslot size is considered as the limit. 3) *Sorting*:

sorts the requests based on their priorities, assigned by the prioritization algorithm. 4) BWallocationResilient: responsible for resilient bandwidth allocation to the requests depending on their types. 5) Update and check for feasibility: Once the required demands are allocated to the requests, the schedule is updated if all the deadlines are met. Otherwise this schedule is infeasible. Full details can be found in [2].

4 Optimized resilient AR scheduling

In this section, we elaborate on how the reliable scheduling approach has been improved to achieve a higher request admittance ratio.

In the resilient approach, if the requested backup can not be found, it retries the primary allocation with fraction of the primary bandwidth demand (50% in our case). Although this is a fast approach, we found that halving the request demand does not always lead to an optimal solution because we may miss the opportunity to transfer a higher volume of the file and the network capacity may not be fully utilized if other concurrent requests can not make use of it. As such, we propose to make better use of leftover capacities by deploying the binary search mechanism [20] for finding the maximum value which satisfies both primary allocation and the requested bandwidth demand. This way, per-step complexity of the reservations increases while a higher amount of allocations will potentially be achieved. We elaborate more on this with two examples.

Example 1: consider Figure 2 with a file-based request of 300 Gb and timeslot size of 5 min. The limit component sets 1Gbps (300 Gb/300s) for the limitation of bandwidth reservations for this request. The resilient approach, first checks if it can fulfill both a 1Gbps primary allocation and the requested backup demand. The amount of backup allocation depends on the percentage of requested protection and also the way the primary reservations are allocated. If the request's backup demand can not be fulfilled, the limit of the request is divided by 2. Then the same procedure is repeated for the lower limit of 500 Mbps. If both 500 Mbps primary demand and its corresponding backup are available, the primary and backup allocations are reserved and the request demand is updated by reducing reserved network capacity and the rest of the file has to wait for the next timeslots to be processed. Otherwise, this division by 2 is continued until the request limit is fulfilled or the file is finally rejected. However, this division by 2 is not efficient if the network is able to provide e.g. 400 Mbps. Based on this approach the resilient reservation approach can only provide 250 Mbps. This is shown in Figure 2b following a multi-path allocation approach, i.e. 200+50 Mbps for primary and 200Mbps (to cover single link failure) for shared backup paths. Although the 150 Mbps can be occupied by other requests, this is not optimal and may results in a waste of network resources.

In order to improve the network utilization, we propose to make use of Binary Search to find an optimal value for the amount of reservations. This way if the algorithm recognizes that 500 Mbps is not available but 250 Mbps can be offered, instead of returning 250 Mbps, the algorithm tries to find the maximum available

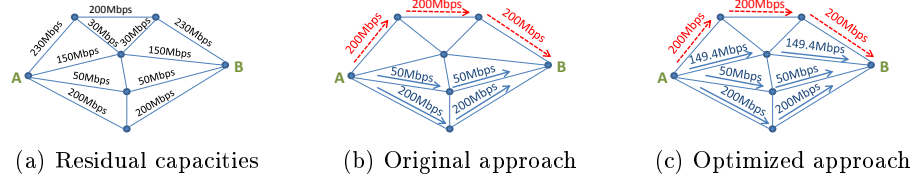


Fig. 2: An example of primary and backup reservations based on the original and the optimized resilient AR approaches. (Black: network capacity, Blue: Primary reservation, Red and dashed: Backup reservation)

value between 250 Mbps and 500 Mbps. The proposed solution first takes the middle value (375 Mbps) and checks the possibility of reservations again. As this is available, the algorithm again checks for the middle value between 375 Mbps and 500 Mbps which is 437.5 Mbps. This trend is continued with 406.25 and then 390.625, etc. until the difference between the upper and lower bounds is smaller than a given margin, which we refer to as ϵ . Assuming 2Mbps as this margin the algorithm stops at 399.4 Mbps. The reservation based on the optimized resilient AR approach is shown in Figure 2c. This margin can be altered to make a tradeoff between achieving a precise optimal value and solution complexity.

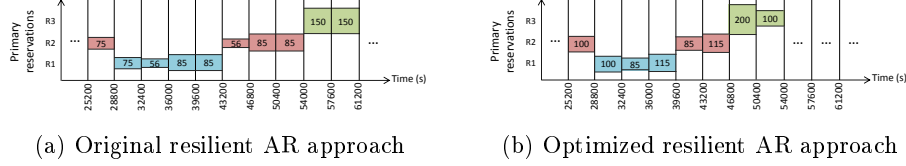


Fig. 3: Comparing the primary allocations of the original and optimized versions of resilient timeslot-based advance bandwidth reservation approach.

Example 2: Figure 3 shows an example of a schedule for 3 file-based requests, R1, R2 and R3. The timeslot size is 1 hour and in both figures only primary reservations are shown. In Figure 3a and Figure 3b, the reservations are made using the original and optimized version of the resilient advance bandwidth reservation approaches respectively. Figure 3b reveals how the optimized resilient approach can improve network utilization and increase the probability of admittance for future requests. As can be seen, by allocating a higher volume of a given file, this file can be potentially transferred earlier compared to the original approach. This way, higher capacity is available for requests in future and the request admittance ratio will be potentially increased.

5 Resilient timeslot-based AR algorithms

In this section, we elaborate on the optimized *BWallocationResilient+* algorithm, shown in Algorithm 1, which first assigns a cost to each network link using a cost allocation module. We have previously designed two algorithms for resilient bandwidth allocation depending on the type of the request, which we refereed to as *BWallocationFBResilient* and *BWallocationVSResilient* for file-based and streaming requests respectively. As we have optimized the resilient approach for file transfers, we do not elaborate on the *BWallocationVSResilient* algorithm. The common part for both algorithms is repeatedly finding the least-cost paths between source and destination of a given request until the limit of that request is fulfilled. However, provided that the limit of the request is not available, a different trend is followed by each approach. For the file-based request, maximum available capacity is reserved as the rest of the file can be processed during the next timeslots. For the streams, if there is not enough capacity to allocate to the request, it can not be served and thus the feasibility is set to false. The next step in the resilient algorithms is to find the backup paths.

```

Data: sortedReqList
costAllocation(Links);
for req ∈ sortedReqList do
  if req is FB then
    | reservation ← BWallocationFBResilient+(req);
  else
    | reservation ← BWallocationVSResilient(req);
  end
end
return reservation;

```

Algorithm 1: BWallocationResilient+

Depending on the backup demands and primary allocations, the amount of backup demand is first calculated. Both algorithms check if the backup can be fulfilled. In order to cover single failures, the backups have to be disjointed from the primary paths. As such, the links used in the primary paths are removed from the network and the bandwidth allocation algorithms are reused on the residual network to find the backup paths for that request. If the backups can be found, the primary and backup paths can be successfully allocated for the request. Otherwise, the primary paths have to be removed. Again if the backups for the streaming request are not fulfilled, the scheduling is not successful. However, for file transfers if the backup can not be provided, the algorithm tries with a lower primary bandwidth demand. This is repeated until both primary and backup demands of the file are satisfied. If this algorithm is being executed in the timeslot prior to the request deadline, and both primary and backup demands can not be fulfilled, the entire scenarios to which the file belongs, is rejected.

5.1 BWallocationFBResilient+ algorithm

The main idea behind proposing the *BWallocationFBResilient+* is to improve the performance of the *BWallocationFBResilient* algorithm. In the *BWallocationF-*

```

Data: an FB request
currentState  $\leftarrow$  Save the current network state
currentLimit  $\leftarrow$  Limit(req);
while currentLimit > 0.1 do
    reservation  $\leftarrow$  BWallocationFB(req, currentLimit, graph);
    maxBW  $\leftarrow$  max Bandwidth(reservation);
    graphReduced  $\leftarrow$  remove the links in reservation from the network graph;
    backupLimit  $\leftarrow$  min(maxBW, requestedBackup(req));
    backupReservation  $\leftarrow$  BWallocationFB(req, backupLimit, graphReduced);
    if !backupReservation then
        set current network state to currentState;
        currentLimit  $\leftarrow$  currentLimit/2;
    else
        return reservation, backupReservation;
    end
end

```

Algorithm 2: BWallocationFBResilient for file-based requests

BResilient algorithm, if the backup for a given request can not be provided, the limit of primary allocations is repeatedly halved and the possibility of the reservation is checked with this lower limit. We argue that this halving cycle can be improved by deploying a binary search algorithm. That is, given a file-based request, we seek for maximum available bandwidth which satisfies both primary and backup demands. Therefore, if the *BWallocationFBResilient+* algorithm finds that value X can satisfy both primary and backup demands, instead of returning this value, which was the case in the *BWallocationFBResilient* algorithm, a higher value based on the binary search approach is investigated and this is repeated until a near-optimal value (within an ϵ margin) is calculated and returned. This process is shown in Algorithm 3.

```

Data: an FB request
currentState  $\leftarrow$  Save the current network state
upperBound  $\leftarrow$  Limit(req);
optimalLimit  $\leftarrow$  Limit(req);
lowerBound  $\leftarrow$  0;
while optimalLimit > 0.1 do
    reservation  $\leftarrow$  BWallocationFB(req, currentLimit, graph);
    maxBW  $\leftarrow$  max Bandwidth(reservation);
    graphReduced  $\leftarrow$  remove the links in reservation from the network graph;
    backupLimit  $\leftarrow$  min(maxBW, requestedBackup(req));
    backupReservation  $\leftarrow$  BWallocationFB(req, backupLimit, graphReduced);
    if !backupReservation then
        set current network state to currentState;
        upperBound  $\leftarrow$  optimalLimit;
        optimalLimit  $\leftarrow$  lowerBound+upperBound/2;
    else
        if upperBound - lowerBound  $\geq \epsilon$  then
            set current network state to currentState;
            lowerBound  $\leftarrow$  optimalLimit;
            optimalLimit  $\leftarrow$  lowerBound+upperBound/2;
        else
            return reservation, backupReservation;
        end
    end
end

```

Algorithm 3: BWallocationFBResilient+ for file-based requests

6 Performance evaluation

This section evaluates the quality and execution time of the proposed solution, compared to our previously designed resilient timeslot-based scheduling algorithms. For this analysis, SARA (Static Advance Reservation Approach) is evaluated in which all requests are known in advance, before the start of scheduling. The influence of the available network capacity, network load, backup demand, timeslot granularity and execution times are assessed.

6.1 Evaluation setup

The network topology used for this evaluation contains 8 nodes and 16 bidirectional links. After discussion with our industrial partners in media production industry, 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. A detailed overview of the randomized variables of requests and network topology can be observed from [1] and [21] respectively. In the fixed size timeslot-based solution, timeslot granularities of 5, 15 and 30 minutes and backup demand of 50% and 100% are used.

Throughout this section, $SARA[XX\%, YYmin]$ denotes that backup demand of $XX\%$ and timeslot size of YY minutes is considered in the fixed-size timeslot-based advance reservation algorithm. *SARA+* refers to the optimized resilient bandwidth reservation approach. In this approach the margin, which we referred to as ϵ , equals 2 Mbps. Each simulation run covers a 24-hour period. All results are averaged over 50 runs with different randomized inputs, error bars denote the standard error. All algorithms in this section are implemented in Java 8.

6.2 Evaluation results

Evaluation of network capacity, backup demands and timeslot sizes:

In Figures 4 and 5, the network infrastructure has been configured for different available bandwidths, respectively for 100% and 50% backup demands to investigate the impact of available network capacity, backup demands and timeslot sizes on the performance of our algorithms in terms of percentage of admitted requests. In both evaluations, 7 scenarios (67 requests) are submitted to the bandwidth reservation system and the network capacities vary from 50 Mbps to 400 Mbps.

What we can observe in these evaluations is as follows: first, the finer the timeslot size, the higher gain achieved by the *SARA+* approach. As can be observed from Figure 4, the *SARA+* approach is able to achieve on average up to 3.6%, 7.3% and 9.2% higher admittance ratio in 30-, 15- and 5-minute timeslot sizes, respectively. Second, with higher backup demand, the performance of *SARA+* is more significant. In Figure 5, with 50% backup demand, *SARA+* is able to outperform the *SARA* approach on average up to 8.5% with 5-minute timeslots. Comparing Figures 4c and 5c, the *SARA+* with 100% back up demand

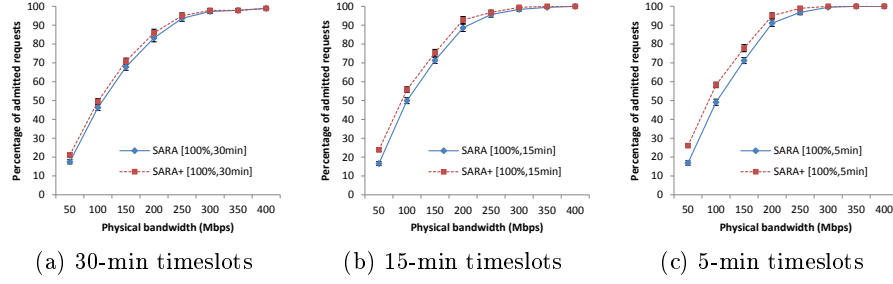


Fig. 4: Impact of timeslot size with 100% backup demand in the timeslot-based advance bandwidth reservation approach.

improves the request admittance ratio on average up to 2.8 times comparing to the 50% backup demand.

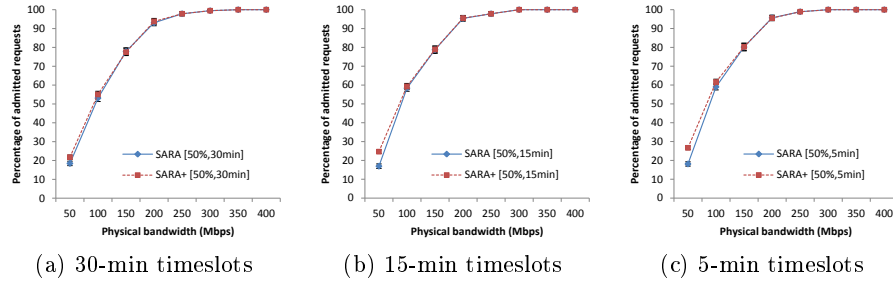


Fig. 5: Impact of timeslot size with 50% backup demand in the timeslot-based advance bandwidth reservation approach.

Evaluation of network load, timeslot sizes and execution times: Figures 6 and 7 compare the influence of the network load and timeslot sizes on the quality and time complexity of our algorithms. Backup demand of 100% and network capacity of 200 Mbps are used.

As can be seen in Figure 6, by increasing the number of scenarios, the percentage of admitted requests decreases and the *SARA+* approach performs better with fine-grained timeslot sizes. We notice that the advance bandwidth reservation system gains more by deploying the *SARA+* approach and with the 5-minute timeslot size, shows up to 7.3% higher request admittance ratio.

The time complexity of the approaches are evaluated in Figure 7 for an increasing range of scenarios. This figure reveals that the granularity of timeslot size impacts the execution times of both approaches differently. While with 30-

minute timeslot size, the execution time of *SARA+* is up to 147 milliseconds higher compared to the *SARA* approach, with 5-minute timeslots, this time is up to 4.5 second lower. These results indicate that the quality and complexity of the advance bandwidth reservation system can be improved by deploying the *SARA+* approach with fine-grained timeslot sizes.

For further investigation of the execution time, we have assessed the impact of network capacity on the execution time, when the timeslot granularity of 5 minutes is used. This has been shown in Figure 8. The number of scenarios is 7 and 14 in Figure 8a and 8b respectively. This evaluation shows that when there is enough bandwidth capacity available, the *SARA* approach is able to perform faster while *SARA+* can better manage the time under stressed network conditions, i.e. limited network capacity.

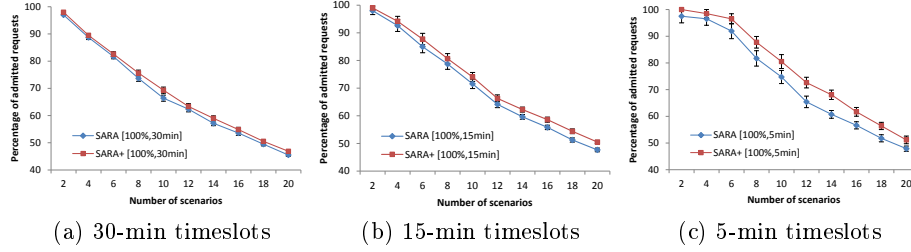


Fig. 6: Impact of network load in the fixed size timeslot-based advance bandwidth reservation approach.

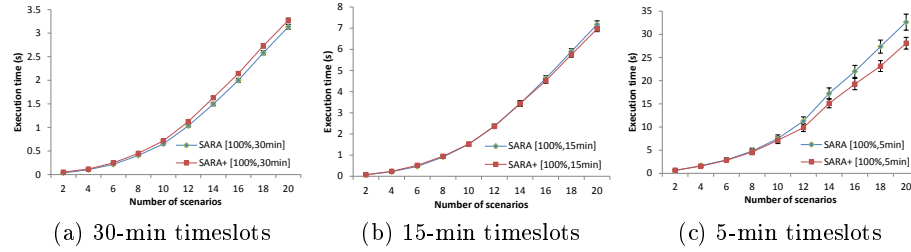


Fig. 7: Comparing the execution times in the fixed size timeslot-based advance bandwidth reservation approach.

7 Conclusions

In this paper, we have optimized the resilient scheduling algorithms, previously presented for advance bandwidth reservation in media-centric networks. In the original version, for a given file transfer, if both primary and backup demands can not be fulfilled, the algorithm is repeatedly executed with 50% of primary

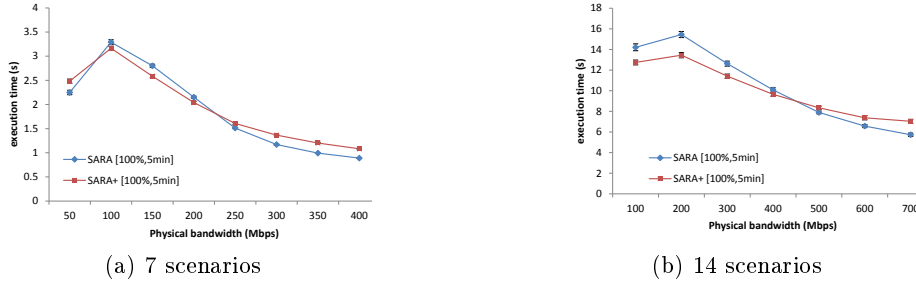


Fig. 8: Comparing the execution times in the function of network capacity in the fixed size timeslot-based advance bandwidth reservation approach.

demand until both demands are fulfilled or the request is rejected. We proposed to make use of binary search instead of halving the primary demand and showed that this optimization improves the performance of the timeslot-based advance reservation system in terms of request admittance ratio. The impact of available capacity, network load, timeslot sizes and backup demands is evaluated. Based on the results, we can conclude that the proposed solution specifically performs well under limited network capacity and with fine-grained timeslot sizes. The proposed approach outperform the original one both in terms of the execution time, with 5-minute timeslot size, and the percentage of admitted requests, up to 9.2%.

Acknowledgment

This work has been performed within the context of ICON MECaNO, 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, "Deadline-aware advance reservation scheduling algorithms for media production networks," *Computer Communications*, 2015.
2. S. Sahhaf, M. Barshan, W. Tavernier, H. Moens, D. Colle, and M. Pickavet, "Resilient algorithms for advance bandwidth reservation in media production networks," in *International Conference on the Design of Reliable Communication Networks (DRCN)*, pp. 130–137, IEEE, 2016.
3. N. Charbonneau and V. M. Vokkarane, "A survey of advance reservation routing and wavelength assignment in wavelength-routed wdm networks," *Communications Surveys & Tutorials, IEEE*, vol. 14, no. 4, pp. 1037–1064, 2012.
4. L.-O. Burchard, H.-U. Heiss, and C. De Rose, "Performance issues of bandwidth reservations for grid computing," in *Symposium on Computer Architecture and High Performance Computing*, pp. 82–90, 2003.

5. S. Naikstam and S. Figueira, "Elastic reservations for efficient bandwidth utilization in lambdagrids," *Future Generation Computer Systems*, vol. 23, no. 1, pp. 1–22, 2007.
6. 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.
7. C. Xie, H. Alazemi, and N. Ghani, "Rerouting in advance reservation networks," *Computer Communications*, vol. 35, no. 12, pp. 1411–1421, 2012.
8. 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.
9. H. Alazemi, F. Xu, C. Xie, and N. Ghani, "Advance reservation in distributed multi-domain networks," *IEEE Systems Journal*, 2013.
10. C. Guok, E. N. Engineer, and D. Robertson, "Esnet on-demand secure circuits and advance reservation system (oscars)," *Internet2 Joint*, 2006.
11. B. Gibbard, D. Katramatos, and D. Yu, "Terapaths: end-to-end network path qos configuration using cross-domain reservation negotiation," in *3rd International Conference on Broadband Communications, Networks and Systems (BROAD-NETS)*, pp. 1–9, IEEE, 2006.
12. 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.
13. 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, (USA)*, pp. 68:1–68:11, IEEE Computer Society Press, 2012.
14. T. Watanabe, T. Omizo, T. Akiyama, and K. Iida, "Resilientflow: Deployments of distributed control channel maintenance modules to recover SDN from unexpected failures," in *11th International Conference on the Design of Reliable Communication Networks (DRCN)*, pp. 211–218, IEEE, 2015.
15. S. Tanwir, L. Battestilli, H. Perros, and G. Karmous-Edwards, "Dynamic scheduling of network resources with advance reservations in optical grids," *International Journal of Network Management*, vol. 18, no. 2, pp. 79–106, 2008.
16. T. Li, B. Wang, C. Xin, and X. Zhang, "On survivable service provisioning in wdm optical networks under a scheduled traffic model," in *Global Telecommunications Conference, 2005. GLOBECOM'05. IEEE*, vol. 4, pp. 5–pp, IEEE, 2005.
17. T. Li and B. Wang, "On optimal survivability design in wdm optical networks under a scheduled traffic model," in *5th International Workshop on Design of Reliable Communication Networks (DRCN)*, pp. 8–pp, IEEE, 2005.
18. C. Cavdar, M. Tornatore, F. Buzluca, and B. Mukherjee, "Dynamic scheduling of survivable connections with delay tolerance in wdm networks," in *INFOCOM Workshops 2009, IEEE*, pp. 1–6, IEEE, 2009.
19. C. Cavdar, M. Tornatore, F. Buzluca, and B. Mukherjee, "Shared-path protection with delay tolerance (sdt) in optical wdm mesh networks," *Journal of Lightwave Technology*, vol. 28, no. 14, pp. 2068–2076, 2010.
20. D. H. Lehmer, "Teaching combinatorial tricks to a computer," in *Proc. Sympos. Appl. Math. Combinatorial Analysis*, vol. 10, pp. 179–193, 1960.
21. M. Barshan, H. Moens, and B. Volckaert, "Dynamic adaptive advance bandwidth reservation in media production networks," in *IEEE NetSoft Conference and Workshops (NetSoft)*, pp. 58–62, IEEE, 2016.