

Design and Evaluation of a Dual Dynamic Adaptive Reservation approach in Media Production Networks

Maryam Barshan^{a,*}, Hendrik Moens^a, Bruno Volckaert^a, Filip De Turck^a

^a *Department of Information Technology, Ghent University – iMinds
Technologiepark-Zwijnaarde 15, 9052 Gent, Belgium*

Abstract

Media production networks require an efficient collaboration between geographically distributed actors and offer predictable workloads, making it possible to exploit this predictability and use advance bandwidth reservation services to achieve greater bandwidth utilization and service guarantees. To offer reliable reservations, the incorporation of fault-tolerance related features in bandwidth reservation strategies is a necessity, although this imposes a waste of capacity and extra performance overhead. As a first provisional stage to offer a robust reservation system, deploying protection mechanisms ensures that the reservations remain valid when the system is in operation. To maximally utilize the network, and to ensure there is a quick response in a dynamic network environment, constant monitoring and optimization is needed. In this article, we propose an efficient dual optimization approach consisting of two consecutive processes. First, a schedule is produced by a resilient advance reservation algorithm. Then, the generated schedule is continually updated over time using a runtime adaptation approach in order to be capable of dynamically adapting the network to changing conditions and mitigating the side effects of provisioned reliability. This step uses the inter-connecting network links' leftover capacity, resulting in an increased performance both in steady and unsteady network conditions. Our evaluations show that in failure-prone environments, the proposed approach leads to significant increase in the success rate of admitted requests, up to 6.77 times, compared to the resilient advance reservation algorithms.

Keywords: Advance bandwidth reservation, media production network, runtime adaptation, network failure handling.

1. Introduction

Media production processes have become more complex and more data/network-intensive as they are increasingly dealing with high bitrate videos, deadline-constrained network transfers and geographically distributed media production teams. Large quantities of data must be processed by multiple collaborating parties at different geographical locations. Media production environments are highly dynamic due to the arrival and departure of several requests of different sizes and requirements. In order to provide high-performance collaboration between different sites, next generation network reservation systems have to provide predictable performance and efficient bandwidth utilization. To ensure that bandwidth needs, delivery deadlines and requirements of different transfers are met, advance bandwidth reservation is needed. In general, advance reservation benefits the network operators as knowledge of future transmissions can be used to improve the admission control and provisioning to increase network utilization. It is also beneficial for the user as the network can provide better QoS to (future) requests with declared arrival and holding times [1], guaranteeing that the needed

network capacity will be available. Advance reservation approaches can be either static or dynamic. While in a static approach all requests are known before scheduling, requests arrive one by one over time in a dynamic model.

In the media production industry, advance reservation scheduling of network transfers [2] is very important in order to make correct decisions on rejection or acceptance of future requests. In uncertain network conditions, such as sudden changes in network configuration, network fluctuations, failures, etc., additional precautions must be taken to guarantee successful transfers. The reliability of transfers in the media production networks is of prime importance and it can be enhanced using protection mechanisms. However, there are arguments against this redundancy as a large portion of network capacity will be wasted if the capacity assigned for this redundancy cannot be reused. As such, making use of idle network capacity and updating the resilient schedule over time, based on the current state of the network and running and planned transfers is of great advantage.

This work has been performed within the context of ICON MECaNO project [3], which provides solutions for the transmission of large media contents over an IP-based infrastructure, tailored to the quality and timing requirements of current and future media production process requests. In our previous work [4] and [5], we proposed

*Corresponding author

Email address: maryam.barshan@intec.ugent.be (Maryam Barshan)

both static and dynamic advance reservation scheduling approaches for a couple of interdependent requests of two types, video streams (VS) and file-based video transfers (FB). We have further presented the resilient version of these approaches based on a protection mechanism to improve the reliability of the advance reservation system [6]. The proposed scheme is capable of covering single link failures using pre-reserved disjoint backup paths. In this article, we make a tradeoff between reliability and resource usage in 3 ways: 1) the percentage of redundancy is defined for each individual request based on an input parameter provided by the customer, to influence the importance of reliability for each individual connection, 2) shared backup path protection (SBPP) [7, 8] is used, significantly reducing the bandwidth requirements for backup purposes, and 3) redundant reservations and network leftover capacity are reused as long as those are not being used for their primary purpose.

The main contribution of this paper is to make use of backups and idle bandwidth capacities to push more data into the network as long as advance reservations are partially unused as well as rapid reaction to sudden changes in uncertain network conditions using an event-based approach. Based on the resilient advance reservation approach, backups are ready for use, but are only activated when failures occur, leaving capacity unused. In addition, we have found that reservations made for video streams, are not completely utilized throughout the requested time. Video streams can be resumed and played-back multiple times during the reserved period, which causes idle reservations between resumes and playbacks. In our proposed approach, these unused capacities can be exploited to transfer additional data. This means that we use these reserved capacities as double-purpose, prioritizing their original purpose. In doing so, as long as these reserved capacities are idle, additional data can be transferred and as soon as for example a video stream becomes active, an event will be raised to prioritize the advance reservation made for this streaming request over the extra data transfers.

The proposed approach consists of two sequential processes. First, the network and transfers status are being continually monitored and the advance reservations are periodically updated. Second, the backup and unused network capacities, e.g. unused video stream reservations, are re-utilized to transfer more data than the schedule made by the resilient advance reservation algorithms. In unreliable networks, as soon as any failure is detected, an event will trigger our proposed algorithm to adapt the ongoing network transfers according to the current state of the network. This leads to a better utilization of substrate network resources, higher success rate and rapid reaction to sudden changes when the network is in operation.

The rest of this article is organized as follows. Section 2, describes background and related work. The envisioned media production network and the runtime adaptation approach are explained in Section 3. The proposed algorithms are described in Section 4. Section 5 provides

simulation results and Section 6 concludes the article.

2. Related work

2.1. Advance resource reservation

Advance network resource reservation has applications for both wide-area and grid networks and has been studied frequently in recent years [9, 10, 11, 12, 13, 14, 15]. Current research mostly focuses on optical networks in combination with wavelength division multiplexing [1]. Advance reservation requests can be classified in 4 individual categories [1]. This classification is also valid for different types of requests in media production networks and all classes are supported in our work. In optical networks, the static advance reservation problem is first introduced by Kuri et al. [16, 17], who focus on requests with specified start time and duration and proposed heuristics and meta-heuristics to solve the static problem. The authors in [18, 19] were the first to propose dynamic advance reservation in fixed time-slotted networks. By introducing the percentage of known requests in [4], both static and dynamic traffic models are considered in our approach. Xie et al. in [20] proposed ILP-based models and heuristic approaches on re-routing in advance reservation networks in order to maximize admittance of new requests. The authors in [21, 22] focused on advance bandwidth reservation for on-demand data transfer in scientific applications. These approaches, however, purely focus on data transfers, not video streaming requests, dependencies among different transfers are ignored and no fault tolerance techniques are considered for possible failures.

2.2. Resilient reservation

Adding resilience into a reservation system can be achieved through restoration or protection failure recovery mechanisms [23]. In [24] a resilient advance reservation mechanism is proposed in optical grids. Due to the lower cost of restoration mechanisms, they use the latter. Burchard et al. in [25] consider a recovery mechanism for advance reservations in grid environments. The idea is to re-schedule failed but unstarted requests whenever failure occurs, but the main focus is on estimating the downtime. The authors in [26] have also focused on a proactive approach by taking resource statistical failure information into account. Their method relies on failure prediction and avoiding vulnerable resources. The authors in [27] present a fault-tolerant job scheduling approach for grid environments using adaptive task replication, which is a recovery approach. Providing resiliency in optical WDM networks through shared path protection has been proposed in [28, 29, 30, 31]. Since meeting strict deadlines and QoS requirements is of great importance in our approach, using protection mechanisms tends to be more reliable.

2.3. Media production networks

The work presented in this article consists of two complementary approaches for media production networks. The combination of a customized resilient AR approach with a highly dynamic event-driven runtime adaptation approach consists of several functions which are of essential importance in the considered media production networks and have not been previously studied in the context of advance reservations.

This work proposes a dual approach which partially makes use of our previous works [5] and [6]. In [5], we devised an Integer Linear Programming (ILP) based model and proposed heuristic approaches for an exact solution. We showed that the heuristics yielded favorable results in much less time complexity than the linear programs. In [6], we enhanced the media production reservation system and made it more reliable in case of failures by following a protection mechanism and provisioning backup reservations for each request. As redundancy imposes cost and resource waste, the main motivation of our approach is to mitigate the side-effect of using redundant reservations by employing underutilized network capacities for transferring extra data as long as those are not needed for redundancy purposes. This work in the extension of [32] in which design of the proposed approach is explained in depth and [33] where the initial evaluation of our proposed approach without considering the impact of failure rates (stable network conditions) and video stream pauses/restarts was presented. This work differs from our previous work as it provides a highly dynamic, complementary and discrete-event driven approach which improves both reliability and performance of media production reservation systems over the time when the network is in operation. In this article, the impact of different failure rates on the performance of the runtime adaptation approach is extensively evaluated.

3. Runtime adaptation approach in media production networks

3.1. Envisioned media production network

The envisioned media production network for MECaNO is depicted in Figure 1. The different actors and locations involved in the media production process, such as recording studios, on-site filming crews, broadcasters, and storage datacenters, are connected to a shared wide-area network. The management layer provides a reservation interface that allows the users of the network to submit their requests. The management layer contains two complementary processes of the proposed approach: the dynamic version of resilient advance reservation algorithm and the runtime adaptation approach, which we refer to as DARA and RA respectively. The DARA scheduling component is responsible for reserving the required amount of bandwidth including backup capacities for all requests and the RA component dynamically re-optimizes the request

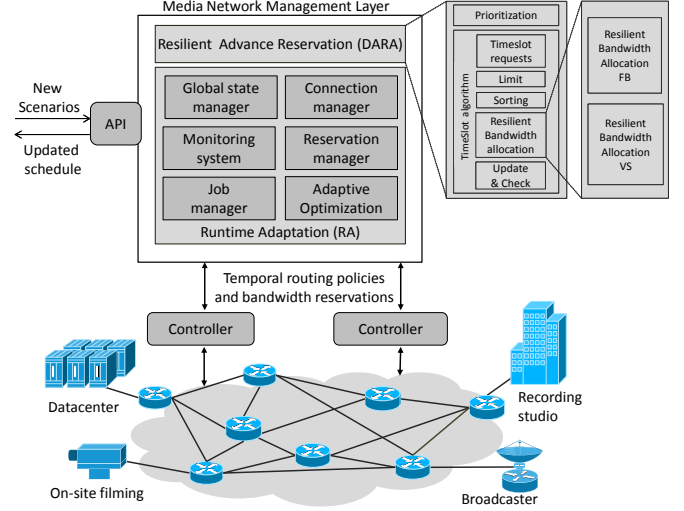


Figure 1: Different components of media production network.

transmissions. Different components of the advance reservation scheduler are extensively explained in [5] and [6].

In the DARA approach, we deal with dynamic time-dependent reservations based on time constraints given by the user, stating the earliest start time and latest completion time. We discretize dynamic network into several snapshots and apply bandwidth allocation algorithms efficiently on every snapshot of the graphs and produce a schedule consisting of distinct reserved capacities in every time interval for each admitted request. A time interval or timeslot is a period of time in which reservations remain invariant. Multiple requests in media production networks may depend on each other, meaning that one request can only start when other requests that are dealt with, have been finished. This interdependency is explicitly incorporated in our approaches. A multi-path routing scheme is followed and the bandwidth scheduling algorithms is based on extending the classical shortest path and maximum flow problems, i.e. modified version of the Dijkstra [34] and Edmonds-Karp [35] algorithms.

An example of advance reservation schedules provided by the DARA algorithm is shown in Figure 2. As can be seen, in every timeslot each request has been served with different allocations, considering the request demands and network capacities. Two individual schedules are generated separately for primary and backup reservations. This scheduling is employed by the reservation system as long as no new scenario is submitted to the system. In this context, scenario refers to a set of several interdependent video transfer requests.

Our advance reservation algorithms support rescheduling in order to incorporate new requests at runtime. As shown in [36, 37, 38], advance reservations decrease network utilization if dynamic reservations are also supported. To improve network utilization, in DARA approach, we assume that whenever a new scenario enters into the reservation system, all new and advance-scheduled requests are being re-scheduled. Since a fixed timeslot-based approach

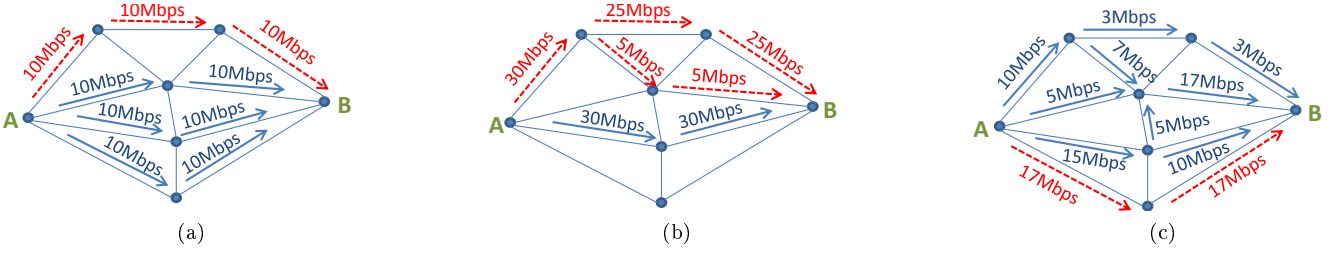


Figure 3: Dependency of backup demand on the reserved primary paths. (Blue: Primary reservation, Red and dashed: Backup reservation)

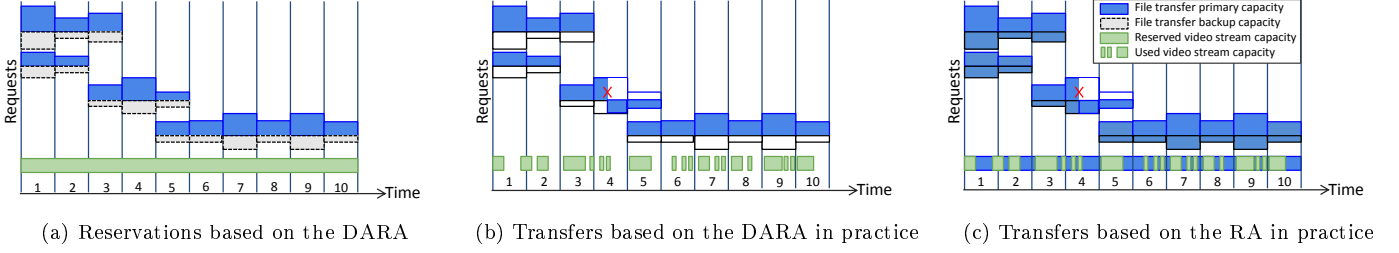


Figure 4: Comparing the DARA algorithm in theory, in practice and the RA approach contribution.

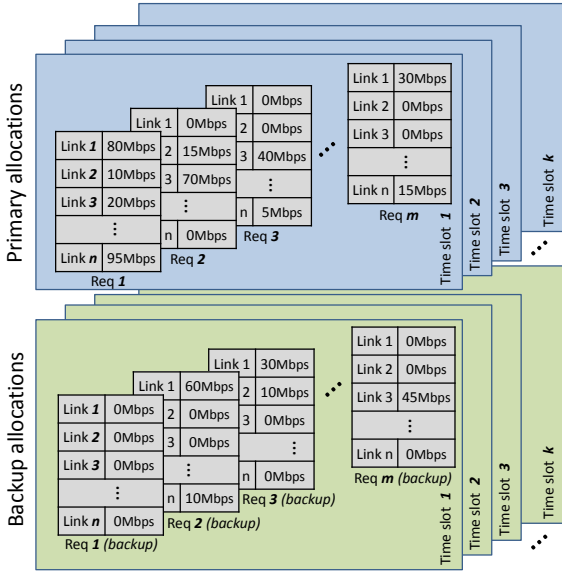


Figure 2: The primary and backup schedules provided by DARA.

has been followed, if the new scenario is admitted, the entire schedule will be updated from the next time interval.

In the DARA algorithm, the backup paths are disjoint from the primary ones. The provisioned protection method guarantees a single link failure recovery. The backups are determined to fulfill the maximum bandwidth allocated on the links of the primary paths. This means that to provide 100% backup, there is no need to allocate the exact amount of bandwidth as in the primary paths [6]. The amount of backup reservation also depends on how the primary demands are allocated. To make it clearer Figure 3 is depicted for a request with

30Mbps primary allocation and 100% backup demand. This figure indicates that how different ways of allocating primary paths can affect the amount of backup demand. In Figure 3a three paths of 10Mbps are allocated to the request. Therefore, it is sufficient for the shared backup to provide 10Mbps. In figure 3b one path is dedicated as primary. In this case the backup has to offer full primary capacity which is 30Mbps. In the third case, backup path offers 17Mbps, which equals to the maximum bandwidth reservation among all primary allocations.

Based on the outcome of the DARA approach, the requests are either rejected or admitted. However, in presence of failures, not all the admitted requests can be completely transferred. Hence, the admitted requests can be categorized as *succeeded*, *degraded* or *failed*. Succeeded requests are those that have been fully transmitted. Deciding on the degraded or failed states depends on the users' preference. In this work, we assume that the users asked for the same value as percentage of backup demand, i.e. if a request has a demand for 60% backup, this request is considered as degraded if at least 60% (but less than 100%) of its volume has been transferred by its deadline, otherwise the request is failed. It should be noted that for 0% and 100% of backup demand, no degradation has been considered. Those requests are either fully-transferred or failed.

3.2. Runtime adaptation (RA) methodology

Figure 4a illustrates an example schedule for four file transfers and one video stream based on the DARA algorithm in a time span of 10 time intervals. For the file transfers, the parts in blue show the primary bandwidth allocations and the parts in gray refer to the backups which are provisioned to be used when a failure occurs in order

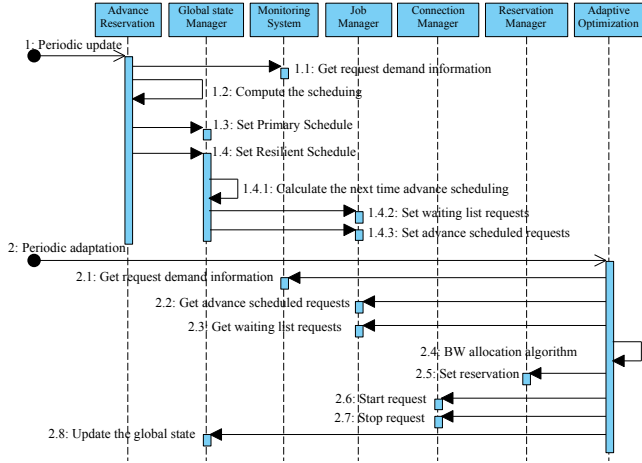


Figure 5: The collaboration between different components in every timeslot in the RA approach.

to be able to transfer the video according to the agreed SLA (Service Level Agreement). Figure 4b shows how the network operates in practice: the backup paths are seldom in use and the reserved bandwidth for the video streams are not continuously utilized, resulting in wasted network capacity. As can be seen in Figure 4c, to have a higher performance and network utilization, we propose a hybrid approach that combines the DARA scheduling approach with an online adaptation system which uses wasted network capacity to increase network utilization.

The RA approach follows two sequential phases in every timeslot: 1) the periodic update and 2) the periodic adaptation. Dynamic network conditions (such as fluctuations, failures, etc.) affect the allocated capacities and network status. As such, the periodic update is repeated before the end of every timeslot to take into account the real transmitted data instead of scheduled ones and update the schedule based on recent information. The periodic update is followed by the periodic adaptation, which is a complementary step to continually adapt network transfers, taking into account the current state of network and transfers and making use of idle network capacity. The periodic adaptation phase is in operation throughout the next time interval.

As illustrated in Figure 5, the RA approach consists of seven components as follows:

- **Advance reservation:** in charge of producing a schedule using the DARA scheduling algorithm. The DARA algorithm is invoked under two circumstances. First, when new scenarios enter the reservation system, leading to an update of the entire schedule for all admitted and unfinished requests. Second, when the schedule needs to be updated during the periodic update. Since periodic adaptation algorithms make use of idle network capacities and real transfers potentially run ahead of schedule, the latter is necessary to take into account the extra transfers and do the re-scheduling for the residual demands. In both cases the

schedule is modified at the start of the next timeslot.

- **Global state manager:** contains all information about scheduling, network and request reservations, connections, demands, deadlines, etc. The time when the current timeslot is started or when it finishes can be retrieved from the global state.
- **Monitoring system:** keeps track of monitored times, residual demand and current allocated bandwidth for all requests. The monitoring system also regularly checks network conditions and raises an event as soon as a failure is detected.
- **Job manager:** contains the list of current advance-scheduled requests and current waiting list requests. Advance-scheduled requests refer to the requests that have already been scheduled by the DARA algorithm to be transferred in the current timeslot. The waiting-list requests are those requests that can potentially be started in this timeslot, but are postponed due to limited network capacity.
- **Connection manager:** decides what to do when a transfer is started or stopped. As long as there are requests with active connections, this component is operational. Whenever a connection for a file transfer is terminated, the links those were in use by this connection become free. In order to improve network utilization, this capacity can be used by other active requests if shared links were in use. To achieve this, after completion of a file transfer, an event will be raised.
- **Reservation manager:** collects all the information about the reservations of each request. Primary allocations, backup reservations, extra allocations made during the periodic adaptation phase and allocated network resources can be retrieved from this component.
- **Adaptive optimization:** in charge of optimization to try and push more data than what has been guaranteed through advance reservation. The Adaptive Optimization (AO) algorithm is the main algorithm in this component which is triggered by several events: start of a timeslot, start and end of file based requests, link failures and repairs. Based on this algorithm, the current schedule is analyzed and adapted to use idle bandwidth capacities.

The rest of this section describes the way the RA phases make use of these components to contribute in performance improvement of media production networks.

3.3. First phase: Periodic update

During the periodic update, first the current status of the network and transfers are monitored and then the DARA algorithm is invoked. This process updates the entire schedule based on the information retrieved from the monitoring system. This new information will be set in the global state manager. Then the next timeslot reservations are derived from the advance reservation schedule

and are set as advance-scheduled requests in the Job manager. The list of advance-scheduled requests contains all requests which have been scheduled to be transferred from now on. Take into account that potentially there are other requests which could be started, but have been postponed due to bandwidth constraints. These requests are kept in a waiting list and used in the periodic update phase.

3.4. Second phase: Periodic adaptation

The Adaptive Optimization (AO) algorithm is the main algorithm in the periodic adaptation phase. The AO algorithm is triggered several times, i.e. whenever a file-based video starts/finishes transferring, and in the case of any link failure or link repair. The first invocation of this algorithm is before the start of the next timeslot. Based on this algorithm, in this step the scheduling over the next timeslot is analyzed and modified to make use of idle bandwidth capacities. To achieve this, the advance-scheduled requests are retrieved from the job manager and then the reservations for backups and video streams are ignored (Because video streams may not always be active and can be resumed/played-back multiple times throughout the reserved period). This gives us a network in which only the primary reservations occupy the network capacities.

3.5. Modeling of the runtime adaptation methodology

In order to model the dynamic aspect of the proposed approach, we have designed a discrete-event based simulator in which the following events are noteworthy:

- **Scheduling update:** When the DARA algorithm is finished, this event is raised.
- **File-based video transfer start time:** When a file transfer starts, the AO algorithm is invoked for all active and new requests.
- **File-based video transfer stop time:** The fully completed request is removed, the other active requests' demands are updated and the AO algorithm is invoked. The previously calculated end times of other active requests are canceled.
- **Video stream start time or play-back:** The video stream transmission is started and allocated bandwidth for the affected file transfers are updated based on the information provided by the VS activation/deactivation handler.
- **Video stream stop time or resume:** The allocated bandwidth for the affected file transfers reset to the previous value provided by the AO algorithm.
- **Global state update:** This event is raised to update reservations and connections, etc.
- **Failure:** As soon as any failure is detected, the link failure/repair handler is invoked, based on which the failed link is removed from the network graph and the AO algorithm is invoked to adapt ongoing transfers.
- **Repair:** When the failure is resolved, the link failure/repair handler is invoked.

4. Runtime Adaptation (RA) algorithms

In this section the algorithms which are used in the periodic update and periodic adaptation phases of the RA approach, shown in Figure 6, are described.

4.1. Periodic update algorithms

The periodic update phase consists of two main steps: The *UpdateRequestsInfo* algorithm, shown in algorithm 1, and the DARA algorithm. We do not elaborate on the DARA algorithm in-detail as it has already been explained in-depth in [6]. In the *UpdateRequestsInfo* algorithm, the demand of submitted requests is updated. To achieve this, first finished and unadmitted requests are removed from the reservation system and then demand of all other submitted requests is updated based on the type of request. For file-based requests however, we deal with volume, so the allocated bandwidth is not fixed and may vary from one timeslot to another. For video steaming requests we deal with fixed bandwidth requirements. Therefore, for file transfers the last monitoring time, last allocated bandwidth and residual transfer volume are updated based on monitored information. If the residual demand of a file-based request is zero, the request has been finished and has to be removed. For video streams, the requests whose deadlines ($t_e^{(rq)}$) are expired are removed. As our approach supports interdependencies among requests, for all requests, list of dependencies are adjusted in case there is any start time dependency to removed requests.

4.2. Periodic adaptation algorithms

The *Adaptive Optimization (AO)* algorithm, which is frequently triggered in the periodic adaptation phase of the RA approach, is shown in Algorithm 2. This algorithm also triggers the *UpdateRequestsInfo* algorithm. Therefore, demands of all requests are already updated whenever the AO algorithm is called.

Based on the AO algorithm, the advance-scheduled requests (ASReq) and the list of waiting requests (WLReq) are retrieved from the job manager. In both lists the

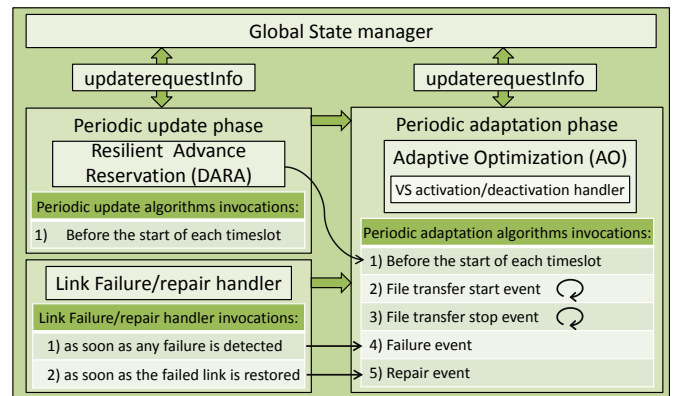


Figure 6: Algorithms used in periodic update and periodic adaptation phases of the RA approach. Narrow arrows show invocations.

```

for ( $rq \in \text{Set of submitted requests}$ ) do
  if ( $\text{request not admitted}$ ) then
     $\text{removedReq.add}(rq)$ ;
  else if ( $\text{Type}(rq) = FB$ ) then
     $\text{LastMonitoringTime}(rq) \leftarrow \text{current time}$ ;
     $\text{LastAllocatedBW}(rq) \leftarrow \text{monitoring.getBW}(rq)$ ;
     $\text{residualVol}(rq) \leftarrow \text{monitoring.getVol}(rq)$ ;
    if ( $\text{residualVol}(rq) = 0$ ) then
       $\text{removedReq.add}(rq)$ ;
    else if ( $\text{Type}(rq) = VS \ \& \ t_e^n(rq) < \text{current time}$ ) then
       $\text{removedReq.add}(rq)$ 
    end
  end
end
for ( $rq \in \text{Set of submitted requests}$ ) do
   $\text{SetDependencylist}(rq).\text{remove}(\text{removedReq})$ ;
end

```

Algorithm 1: The UpdateRequestsInfo algorithm.

```

UpdateRequestInfo();
ASReq  $\leftarrow \text{JobManager.getASReq}(\text{current time})$ ;
WLReq  $\leftarrow \text{JobManager.getWLReq}(\text{current time})$ ;
PriorityBasedSorting (ASReq);
PriorityBasedSorting (WLReq);
graph  $\leftarrow \text{GlobalState.getGraph}(\text{current time})$ ;
graph.removeReservations(VSs, Backups);
ExtraAlloc  $\leftarrow \text{BWallocation}(\text{ASReq}, \text{graph})$ ;
for ( $rq \in \text{ASReq}$ ) do
  TotalAlloc(rq)  $\leftarrow \text{PrimaryAlloc}(rq) + \text{ExtraAlloc}(rq)$ ;
  FinishTime(rq)  $\leftarrow \text{residualVol}(rq) / \text{TotalAlloc}(rq)$ ;
   $\text{rq.SetReservations}(\text{TotalAlloc}(rq))$ ;
   $\text{rq.StartConnection}(\text{current time})$ ;
   $\text{rq.FinishConnection}(\text{FinishTime}(rq))$ ;
end
WLAlloc  $\leftarrow \text{BWallocation}(\text{WLReq}, \text{graph})$ ;
for ( $rq \in \text{WLReq}$ ) do
  if ( $\text{WLAlloc}(rq) \neq 0$ ) then
    FinishTime(rq)  $\leftarrow \text{residualVol}(rq) / \text{WLAlloc}(rq)$ ;
     $\text{rq.SetReservations}(\text{WLAlloc}(rq))$ ;
     $\text{rq.StartConnection}(\text{current time})$ ;
     $\text{rq.FinishConnection}(\text{FinishTime}(rq))$ ;
  end
end

```

Algorithm 2: The Adaptive Optimization (AO) algorithm which is the main algorithm in the periodic adaptation phase of the RA approach.

requests are sorted. The main factor for this sorting is the estimated deadline: the sooner the deadline, the higher the priority. The second factor, volume, comes into consideration only when the deadlines are equal, the higher the demand, the higher the priority. Then, reservations made for backups and video streams (VSs) are ignored. This gives us a network in which only the primary reservations occupy network capacity. Then, for the advance-scheduled requests, new extra allocations (ExtraAlloc) for requests over this residual graph are computed. Extra reservations are aggregated with the primary reservations of requests

```

input: failed/repared link, network graph
FailedLinks.add/remove(link);
if ( $\text{FailedLinks} \neq \text{empty}$ ) then
  graph.removeFailedLinks(FailedLinks);
  GlobalState.setGraph(current time);
end
for ( $rq \in \text{requests affected by failure}$ ) do
  if ( $\text{PrimaryAlloc}(rq).\text{contains}(\text{FailedLinks})$ ) then
     $\text{rq.UpdatePrimaryAllocations}(\text{FailedLinks})$ ;
  else if ( $\text{BackupAlloc}(rq).\text{contains}(\text{FailedLinks})$ ) then
     $\text{rq.UpdateBackupAllocations}(\text{FailedLinks})$ ;
  end
end
AdaptiveOptimization();

```

Algorithm 3: Link failure/repair event handler.

and their assigned bandwidth will be potentially increased. For each request, new allocations are updated in the reservation manager. Based on these new allocations, the start time and finish time of requests are configured in the connection manager. All the reservation and connection information is saved in the global manager. The same steps are repeated for the waiting-list requests. The only difference is that there is no primary reservations for those requests.

A request that finishes will raise an event which first cancels the stop time events of all other active requests. Then, the AO algorithm is triggered to calculate extra allocations and finish times. Since a request just finished, these new finish times will be earlier than the previously canceled ones. Moreover, whenever a new request is ready to be started within a time interval, this may also raise another event to trigger the AO algorithm. This cycle is repeated as long as requests trigger events.

Detecting a failure or repair throws an event whose handler is shown in Algorithm 3. Based on this algorithm, first the failed/restored network elements are removed from/restored to the network. By calculating the effect of failures on each request, the primary and backup reservations of all affected requests are updated. The AO algorithm is invoked to re-optimize ongoing transfers, taking into account network failure status. In general, Algorithm 3 allows the reservation system to adjust in-advance reservations for affected requests and makes new additional reservations over residual network capacity.

Algorithm 4 shows how file based transfers' bandwidth allocations are varied when transmission of video streaming requests (with B^{VS} bandwidth demand over the VS-links) are started. This algorithm is executed whenever the AO algorithm is invoked to determine which requests have to be restrained to serve the video streaming request whenever it becomes active. In order to do this, first the algorithm checks the residual network capacity to update the video stream demand if part of the advance reservations for this request is still untouched. Then, ongoing file transfers are sorted from lowest priority to highest, common links (Clinks) between video stream and extra alloca-

```

input: request VS
VLinks  $\leftarrow$  AdvanceReservedLinks(VS);
for ( $l \in VLinks$ ) do
    Vsdemand( $l$ )  $\leftarrow B^{VS} - \text{residualCapacity}(l)$ ;
    if ( $Vsdemand(l) \leq 0$ ) then VLinks.remove( $l$ );
end
for ( $FB \in \text{File-Based requests}$ ) do
    CLinks  $\leftarrow$  CommonLinks(ExtraAlloc(FB), VLinks);
    for ( $l \in CLinks$ ) do
         $x \leftarrow \text{ExtraAlloc}(FB, l)$ ;
        if ( $x \geq Vsdemand(l)$ ) then  $x \leftarrow Vsdemand(l)$ ;
        ExtraAlloc(FB)  $\leftarrow$  ExtraAlloc(FB) -  $x$ ;
        Vsdemand( $l$ )  $\leftarrow$  Vsdemand( $l$ ) -  $x$ ;
        if ( $Vsdemand(l) = 0$ ) then VLinks.remove( $l$ );
    end
end

```

Algorithm 4: VS activation/deactivation handler.

tion of file transfers are checked and the common allocations are removed from the extra allocations of file-based request. This is repeated until the video stream demand is fulfilled. Note that this is a provisioning algorithm to react immediately as soon as a video stream request starts using its reservations.

4.3. Clarifying examples

Figure 7 shows an example of the difference between the reservations made by the DARA algorithm and the reservation as input of the AO algorithm in periodic adaptation phase of the RA approach. As can be seen in Figure 7a, two file-based transfers and one video stream are active with primary and backup paths in one timeslot. FB1 is a file-based request from node A to node B with 200Mbps multipath allocation, FB2 is similar to FB1 but from node C to node D and 700Mbps reservation, video stream VS1 has a requirement of 50Mbps, from node E to node F. As shown in this figure, for each individual request, primary and backup paths do not have any link in common. All three requests had a requirement of 100% backup and the amount of capacity reserved as backups equals the maximum allocated bandwidth along the primary paths, which is equal to 200Mbps, 400Mbps and 50Mbps for FB1, FB2 and VS1 respectively.

Figure 7b shows the reservations taken into account in the AO algorithm before the start of the next timeslot. The video stream and all the backups are eliminated and only file transfer primary paths are kept. Taking this network into account, extra allocations for each request are calculated. These extra reservations will be aggregated with the primary reservations of the requests and their assigned bandwidth will be increased. For each request the new allocations are updated in the reservation manager. Whenever a request is finished, the bandwidth reserved for that finished request is returned to the network resource pool and the AO algorithm is re-invoked to determine new allocations. Based on the new allocations,

the start time and finish time of the requests are set and kept in the connection manager. This cycle is repeated until all requests are finished or the timeslot ends. In case a video stream is activated, the advance reservation for this request is prioritized and other extra allocations that have made use of this capacity, are interrupted. Applying the AO approach, the reservations will not remain fixed during each time interval. **As for the actual transfers, we make use of the extra allocations in addition to the primary allocations, in stable network conditions the actual transfers are higher than what had been previously envisioned by the DARA approach.**

Figure 8 clarifies the differences between bandwidth allocation algorithms in the DARA approach and in the AO algorithm of the RA approach. The key difference is that the advance reservation algorithm takes the size of the timeslots into account. For example for a 10 GB file, the primary allocated bandwidth is 80 Mbps which is enough for the file to be transferred in a 1000-second timeslot. Note that based on the DARA approach, the allocated bandwidth may vary from timeslot to timeslot but it is fixed within each timeslot. Using the RA approach, the allocations may vary when the network is in operation, even within timeslots. The extra allocation for the request is calculated, depending on the spare network capacity. Considering the extra allocation of e.g. 160Mbps, the sum of all allocations is three times higher than the primary allocation. Therefore, the file transfer is finished in a third of the time of the finish time which was computed in advance when no failures occur.

During the runtime, any early finish will trigger an event which indicates that the present connection can be torn down. The links which were in use by this request are now free, allowing other active requests from the advance-scheduled list, to use more bandwidth if the shared links were in use. Furthermore, there can be other future requests in the advance reservation schedule which can make use of some of these links. Since these links have already been reserved for a finished request, the future requests stored in the waiting list, could be analyzed and potentially scheduled. To achieve this, the finished request is removed from the request list and the AO algorithm is triggered. In doing so, the corresponding file transfers begin earlier than they were scheduled by the advance reservation scheduler, improving link utilization.

As the reserved capacities for video streams and backups have a double purpose, pre-determining how to manage the conflicts before they happen is crucial. To achieve this, two important functions called failure/repair handler and VS activation/deactivation handler are proposed. The VS activation/deactivation handler determines when any video stream is active, **the extra allocation of which file-based requests are affected and how the extra allocation of the affected request is adjusted to reflect this activation.** The failure/repair handler determines how to handle the conflicts when the backup reservations are active for their original purpose. During the runtime, any failure/repair or

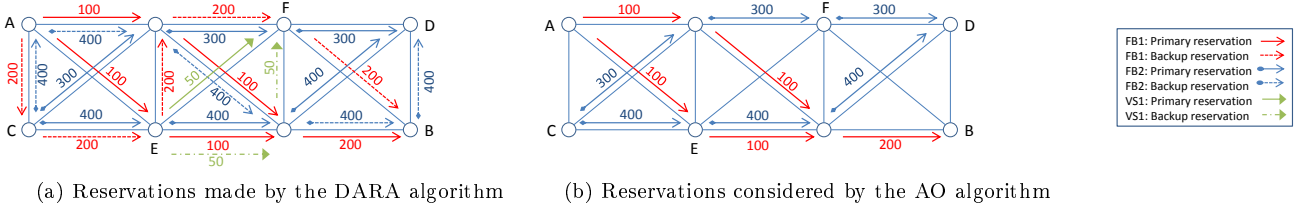


Figure 7: Comparing a snapshot view of bandwidth reservations for 3 requests as output of the DARA algorithm and as input of the AO algorithm (Full lines: primary reservations, dashed lines: backup reservations, Open arrows: file-based transfers).

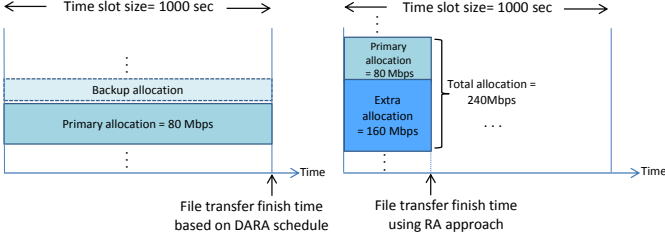


Figure 8: Differences between bandwidth allocation algorithms in the DARA and RA approaches.

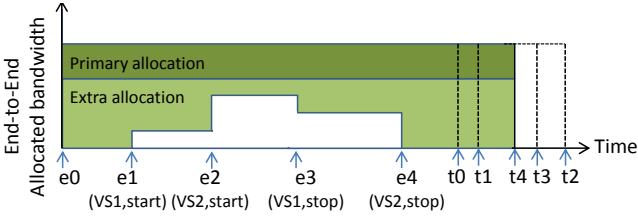


Figure 9: The impact of video streaming requests activation/deactivation on file transfer finish time in the periodic adaptation phase of the RA approach.

any video streaming start/stop or resume/play-back will prompt an event and the information provided by these functions allows the management system to quickly handle the event. To elaborate more on this, Figure 9 shows how video streaming requests' activation/deactivation affect the extra allocation of a file transfer, allocated by the RA approach. As can be observed from this figure, the finish time of a file transfer is adjusted as soon as a video stream request starts/stops. Five events e_0 - e_4 cause 5 different finish time t_0 - t_4 for the file transfer request. e_0 is the first event in the periodic adaptation phase, at the beginning of the timeslot when no VS is active, and the estimated finish time is t_0 . VS1 activation raises an event (e_1), based on which t_1 is calculated as the new finish time and then t_0 is substituted by t_1 . Other VSs activations/deactivations have the same impact on the file transfer finish time. Eventually as e_4 is the final event in this timeslot, the file transfer is finished at t_4 .

Several invocations of the AO algorithm during the periodic adaptation phase are shown in depth in Figure 10. The reservations made by the DARA algorithm for the first timeslot, starting at 0 and ending at 300s, are illustrated in Figure 10(a). We assume that no failures are

occurred during this time interval. Three file-based transfers FB1, FB2 and FB3 are reserved with 100%, 50% and 30% backup (shown in dashed rectangles) respectively. Figures 10(b)-(f) show how the AO algorithm in the periodic adaptation phase is being used to optimize the transmissions. In Figure 10(b), the primary reservations are adapted by the first invocation of the AO algorithm before the timeslot starts. Based on these new allocations, FB1 finishes at 140s, FB2 at 200s and FB3 at 100s. As FB3 has the earliest finish time, the AO is invoked again at 100s. This is shown in Figure 10(c). As the capacity used by FB3 has now been released, FB4 has the opportunity to start earlier. Based on the advance reservations, FB4 should have started in the next time interval at 300s, but thanks to the AO algorithm, it can be started earlier at 100s. The stop time of FB1 and FB2 is also updated from 200s to 150s and from 140s to 130s respectively. As can be seen in Figure 10(d), the next time for re-invocation of the AO algorithm is at 130s when FB2 is terminated.

It should be noted that for some requests there could be enough capacity to be transferred but due to inter-dependencies on other requests, they have to be postponed. The RA approach tries to accommodate these requests as soon as their dependencies are eliminated. To elaborate more on this, we assume that FB5 has a dependency to FB2 and can only be started when FB2 finishes. Based on the advance reservation schedule FB5 cannot start in this timeslot as FB2 is in operation. Now, using the RA approach, it can be started earlier at 130s instead of the next timeslot. As Figure 10(e) shows, only FB4 and FB5 are active from time 140s. This means that all primary reservations have been transferred by then which indicates that the transfer of advance-scheduled requests is 160s ahead of the resilient schedule. The last invocation takes place at 190s when FB4 finishes and from then only FB5 is being transferred. This is shown in Figure 10(f). This example also shows that the RA approach tries to mitigate the side-effects of fixed time intervals. In periodic adaptation step, not only the spare capacities are re-used, but also the allocated bandwidths vary within timeslots, resulting in a more flexible transfers comparing to the DARA schedule.

Figure 11 illustrates the impact of periodic update and periodic adaptation algorithms on the performance of bandwidth reservation system within 3 timeslots. FB1,

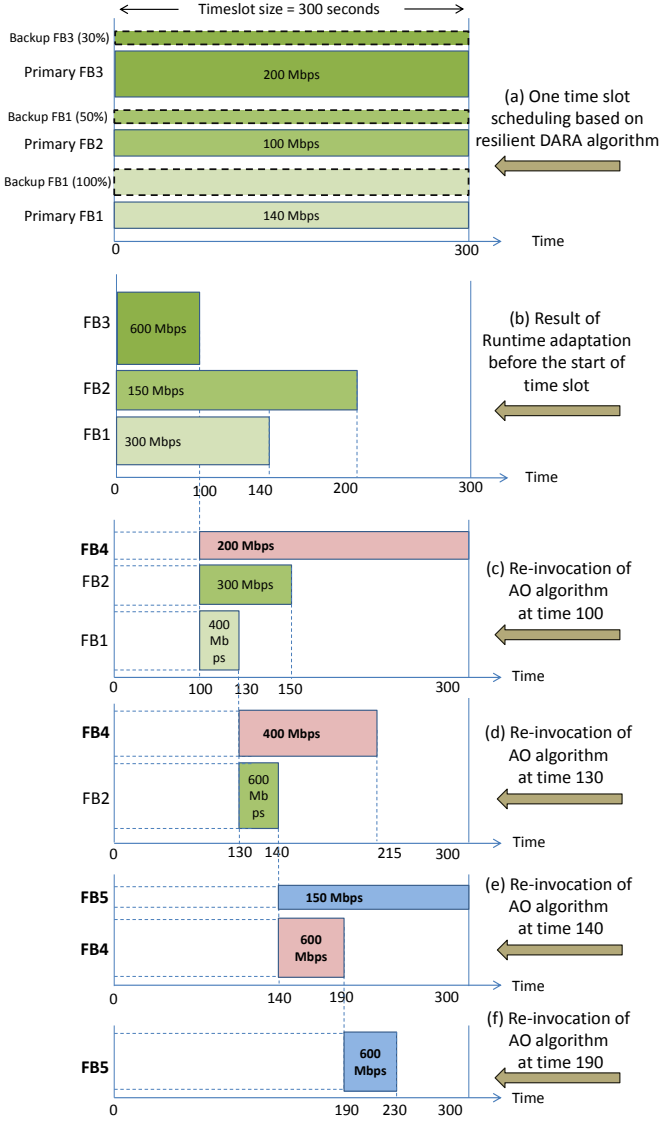


Figure 10: Single timeslot reservations made by the DARA algorithm and multiple re-investigations of the AO algorithm during the periodic adaptation phase of the RA approach.

FB2 and FB3 asked for 100%, 30% and 50% backup respectively. According to the advance reservations, FB1 has bandwidth allocations of 200Mbps in the first and 100Mbps in the second timeslots, FB2 150Mbps in the first and FB3 300Mbps in the third time interval. Applying the RA approach and by several invocations of the AO algorithm, which ignores the backup reservations, FB1 and FB2 have completely transferred in the first timeslot and transfer of FB3 has already been started. Before the start of the second timeslot, the schedule is updated during the periodic update phase. Therefore, the reminder of FB3 is shifted to the second timeslot and the bandwidth reservations for FB3 in the third timeslot are completely released. As can be seen when a new scenario is submitted to the reservation system at the end of the second timeslot, if the RA approach is not used, the management system would not have been able to serve the new scenario, but it

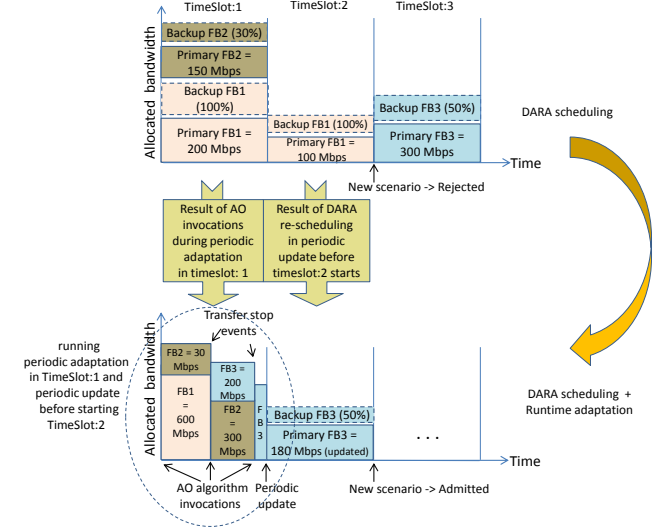


Figure 11: The impact of invocations of algorithms in periodic update and periodic adaptation phases of the RA approach on performance of reservation system.

is admitted thanks to the RA approach.

5. Performance evaluation

In order to model the dynamic aspect of our model, we have designed a discrete-event-based simulator using the MASON multi-agent simulation toolkit [39]. In this section the impact of using a runtime adaptation approach is thoroughly evaluated and compared to the DARA algorithm. The DARA approach can be configured for different percentages of requests known in advance. This refers to the percentage of scenarios that are known at the start of the simulated period. In these evaluations, we assume that none of the scenarios are known in advance, which is the most realistic case. It should be noted that the DARA approach has been previously validated compared to an exact optimal ILP-based solution.

Throughout this section, $DARA[XX\%,YY\%]+RA$ denotes that dynamic version of resilient advance reservation approach with $XX\%$ of backup and failure rate of $YY\%$ is used. The second part (RA), is optional and specifies if the runtime adaptation approach was used or not.

5.1. Evaluation Setup

In this evaluation we have used 8-nodes and 25-nodes media production network topologies, depicted in Figure 12. The 25-node topology is the well-known ATT North America topology [40] consisting of 25 nodes and 56 bidirectional links (112 links in total) which matches to the size of realistic media production networks. The default network capacity is 300Mbps per link. We have previously defined three scenario templates based on the information gathered from several Belgian media production actors, including a broadcaster, service provider and recording facility provider [5]. Each scenario contains a

collection of interdependent file and video streaming transfers with randomized parameters. Template1 is composed of 5 different file transfer requests. Template2 comprises 18 interdependent file transfers. The third template includes 4 file transfer requests and 4 video streams.

For the 8-node topology, the number of scenarios equals 20, of which 7, 7 and 6 are of template1, template2 and template3 respectively (209 requests in total). For the larger 25-node topology, the number of scenarios is 50, of which 17, 17 and 16 are of the first, second and third templates respectively (519 requests in total). A fixed time interval granularity of 1 hour is used. It should be noted that, every 1-hour timeslot in the AR approach is divided into several flexible timeslots by the RA approach. As in the considered scenario templates the requests are only known a few hours in advance, each simulation run covers a 24 hour period. All results are averaged over 50 runs with different generated scenarios, error bars denote the standard error.

In order to generate MTBF (mean time between failures), MTTR (mean time to repair) and video stream activation/deactivation events, we used a normal distribution function with equal values for both mean and standard deviation. This value equals 5 minutes for video stream activation/deactivation. It is not trivial to assign a value for MTTR, as it depends on multiple factors, e.g. type of links, type of failures, underlying technology [41]. The main focus of this section is to evaluate the performance of our approach under catastrophic failures in failure-prone networks. As such, 48 minutes is chosen as mean/standard deviation value for the MTTR to experience higher unavailability. However, in unstressed network conditions this value is reduced to 5 minutes. To give an insight in number of concurrent failed links for each topology, Table 1 shows the minimum, maximum and average number of failed links for different MTBF values under stressed network conditions.

The bandwidth contention per link for the 8-node and 25-node topologies is shown in Figure 13. When calculating bandwidth contention, we assume that all requests are admitted and use a single shortest path from source to sink. Contrary to the video streaming requests which have a fixed bandwidth demand, the bandwidth requirement of file-based transfers has to be estimated. Due to interdependencies among requests of each scenario, some

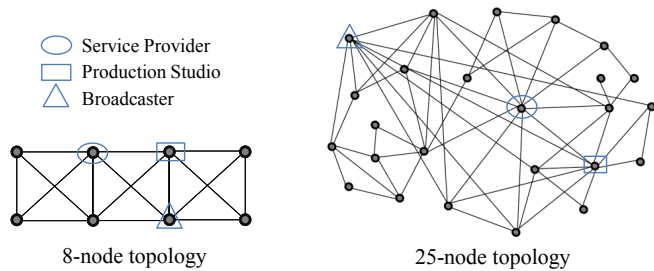
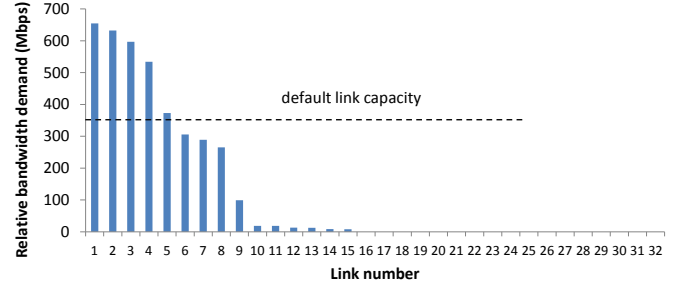
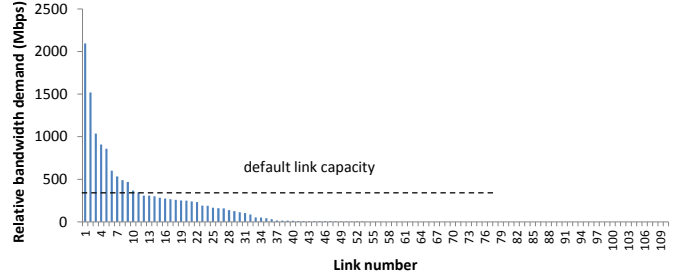


Figure 12: Media production network topologies used for evaluation.



(a) Bandwidth contention for the 8-node topology.



(b) Bandwidth contention for the 25-node topology.

Figure 13: Bandwidth contention per link for 20 scenarios in 8-node topology and 50 scenarios in 25-node topology.

Table 1: Minimum, maximum and average number of concurrent failures for different failure rates in 8-node and 25-node topologies.

Failure rate	8-node topology (32 links)		25-node topology (112 links)	
	AVG	MAX	AVG	MAX
1h	14.45	22	47.82	63
2h	8.76	17	31.97	46
4h	5.3	10	16.64	26
10h	2.17	6	8.47	20
20h	1.2	4	3.9	8

requests may not have specified start or stop times. In order to have an estimation, the volume of all file transfer requests belonging to a scenario are divided by the time from when the earliest request of the scenario is ready to be transferred until its final deadline (the deadline of the latest request). Actors between scenarios move, except for some common locations. As shown in Figure 12, prime locations i.e. service provider, production studio and broadcaster, are located at highly connected nodes. Locations of other actors are randomly chosen. Therefore, Figures 13a and 13b show the connection to prime locations as a set of hotspots with high intensity. In the 8-node and 25-node topologies the highest contention per link is at maximum 655Mbps and 2,095Mbps respectively.

5.2. Impact of different failure rates, fixed backup demand

5.2.1. Impact of available bandwidth

First, we evaluate the impact of changing the network link capacity. Figure 14 and Figure 15 show the impact of available bandwidth and different failure rates on the performance of the RA approach for the 8-node and 25-node topology respectively. In both topologies, bandwidth

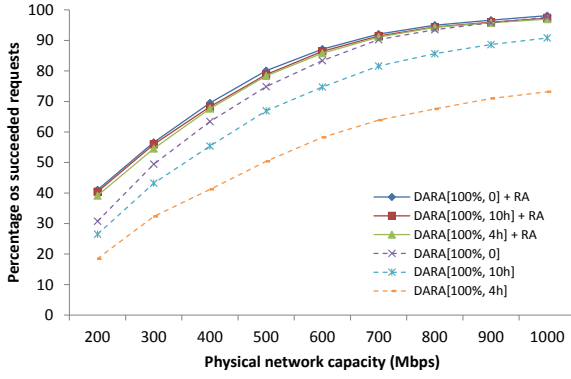


Figure 14: Impact of network capacity and failure rates on the performance of using the RA approach for the 8-node topology.

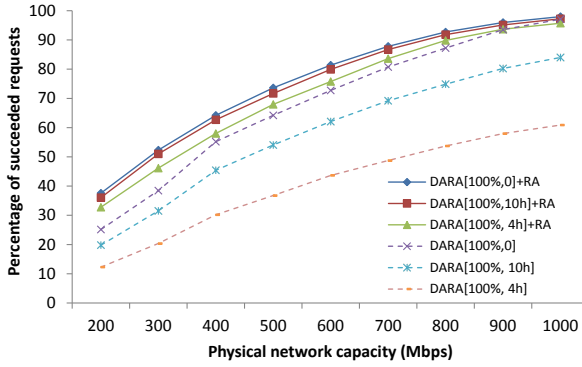


Figure 15: Impact of network capacity and failure rates on the performance of using the RA approach for the 25-node topology.

capacity per link is parametrized from 200Mbps to 1Gbps and all scenarios' requests demand 100% of backup. These figures show that the RA outperforms the DARA up to 27.6% and 36.12% with 4-hour failure rates for 8-node and 25-node networks respectively. The standard error at maximum reaches to 1.16% in smaller and 1.5% in larger topologies (not shown for greater legibility). Our results show that regardless of the failure rate, the RA approach almost always outperforms the DARA approach.

5.2.2. Impact of network load

Figure 16 and Figure 17 show the impact of network load and different failure rates on the performance of the RA approach using the 8-node and 25-node topology respectively with a backup demand of 100%. Since the network capacity remains fixed, adding more requests leads to an increase in rejection rate. The results show this for both smaller and larger topologies, the RA approach improves the percentage of admitted requests up to 23% and 30% on average respectively.

5.3. Impact of different backup demands, fixed failure rate

5.3.1. Impact of available bandwidth

Figures 18a, 18b, 19a and 19b analyze the impact of network capacity and percentage of backup demand on

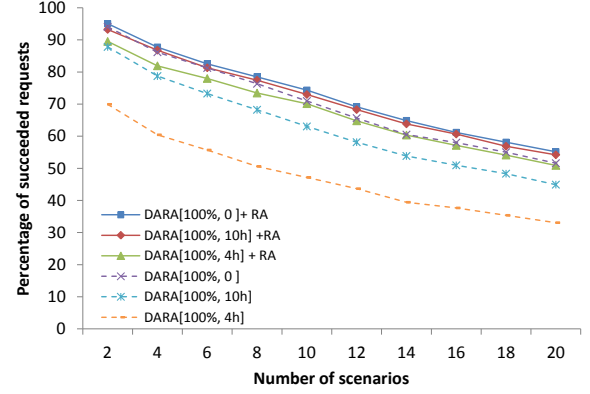


Figure 16: Impact of network load and failure rates on the performance of using the RA approach for the 8-node topology.

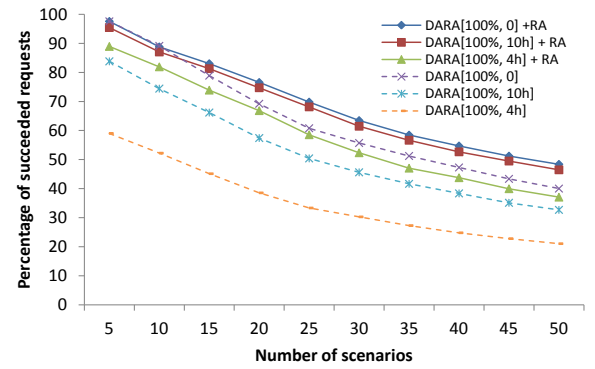
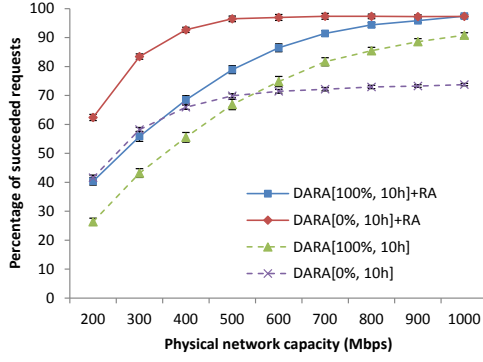


Figure 17: Impact of network load and failure rates on the performance of using the RA approach for the 25-node topology.

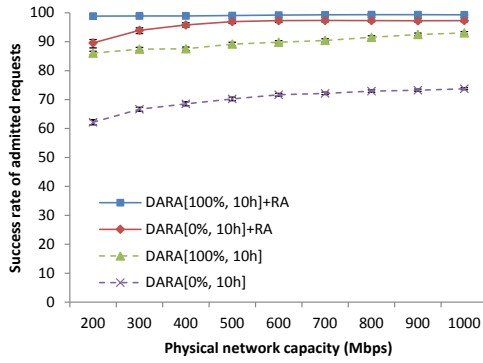
the performance of our approaches. In these evaluations, backup demand of 0% and 100% and failure rate of 10 hours are taken into account. Figure 18a and Figure 19a show the average percentage of succeeded requests (out of all submitted requests) in 8-node and 25-node networks respectively. Figure 18b and Figure 19b compare the same experiments for the success rate of admitted requests. As can be seen in these figures, the RA approach has noticeably improved the request success rate. In Figure 18a, the highest performance in terms of number of succeeded requests is achieved when dropping backup requirements and using the RA approach. Nevertheless, as can be observed from Figure 18b with a backup setting of 100% the highest QoS (success of admitted requests) can be achieved. These evaluations also reveal that when there is sufficient capacity in the network (1Gbps), the RA approach is able to achieve the same quality when dropping backup requirements, for 100% of backup demand in terms of succeeded requests. The same trend can be observed in 19a and 19b for the 25-node topology. For both 0% and 100% of backup demand, the RA approach outperforms the DARA approach.

5.3.2. Impact of network load

Figures 20a and 21a show the percentage of succeeded requests (out of submitted ones) and compare the impact

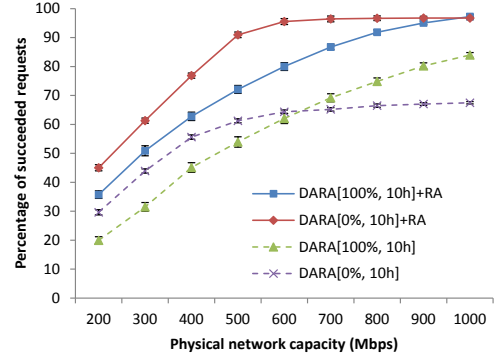


(a) Percentage of succeeded requests.

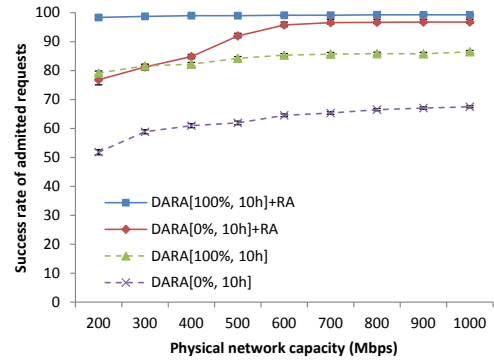


(b) Success rate of admitted requests.

Figure 18: Impact of available bandwidth and backup demands on the performance of using the RA approach for the 8-node topology.



(a) Percentage of succeeded requests.



(b) Success rate of admitted requests.

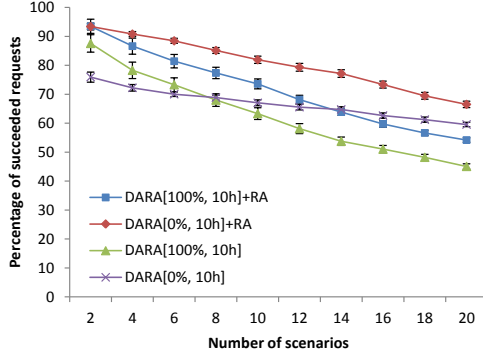
Figure 19: Impact of available bandwidth and backup demands on the performance of using the RA approach for the 25-node topology.

of scenario load and different percentages of backup demands on the performance of proposed approaches. Figures 20b and 21b show the success rate of admitted requests for the same experiments, using a backup setting of 0% and 100% and 10-hour failure rate. For the smaller topology the number of scenarios increases 2 by 2, up to 20 and for the larger topology, this number increases 5 by 5, up to 50. These figures show that the RA approach achieves the best performance in terms of number of succeeded requests and success rate of admitted requests. Figures 20a and 21a show that when there is sufficient network capacity, no backup requirements outperform the 100% backup demand in the DARA approach. However, Figures 20b and 21b reveal that using DARA with no protection leads to the worst performance in terms of success of admitted requests. Interestingly, it can also be observed that using the RA approach without protection provides almost always higher success rate than the DARA approach even with 100% of protection.

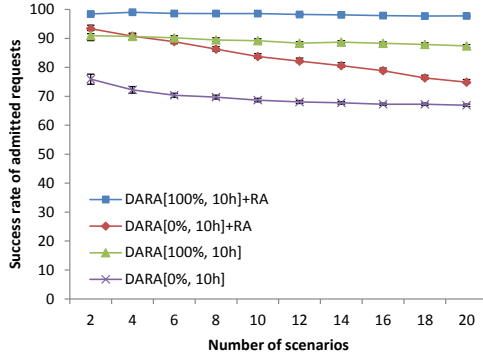
5.3.3. Stressed versus non-stressed network conditions

Figure 22 and Figure 23 elaborate on the final state of requests in stressed and non-stressed network conditions. The stressed network condition is defined as having 300Mbps bandwidth per link, 2-hour failure rate and de-

fault 48-minute repair rate. In non-stressed conditions, the available bandwidth is increased to 800Mbps and the MTTR is reduced to 5 minutes. Figure 22a and Figure 23a show that in general having more protection in the DARA approach leads to more succeeded requests. In both figures, the percentage of succeeded requests with 100% backup demand is almost 2 times higher than when no backup is provisioned. Figure 22b and Figure 23b show the impact of deploying the RA approach with the same experiments (note that in these figures the y-axis starts from 90%). These figures reveal crucial advantages of the RA approach. First, comparing to Figures 22a and 23a, we can see that the RA approach leads to a lower percentage of rejection, up to 1.65% and 4.87% in Figure 22b and Figure 23b respectively. Second, the percentage of failed requests significantly improves both from lower to higher protection and also compared to the DARA approach. To be precise, with 100% of backup demand in the 8-node topology, the percentage of failed requests is reduced from 1.47% to 0.08% when compared to 0% backup demand in the RA approach, and more importantly, when compared to the DARA approach, this percentage is reduced from 15% to only 0.08%. Third, dropping backup requirements and under non-stressed network conditions, runtime adaptation significantly improves the success rate of reserva-



(a) Percentage of succeeded requests.



(b) Success rate of admitted requests.

Figure 20: Impact of network load and backup demands on the performance of using the RA approach for the 8-node topology.

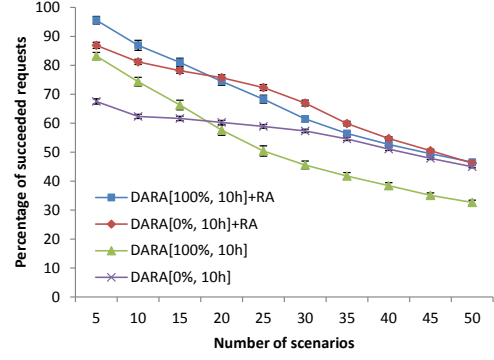
tion system. It can be seen that the number of succeeded requests is highest when dropping backup requirements. However, compared to 100% backup a noticeable number of admitted requests fail.

5.4. Impact of different backup demands, varying failure rates

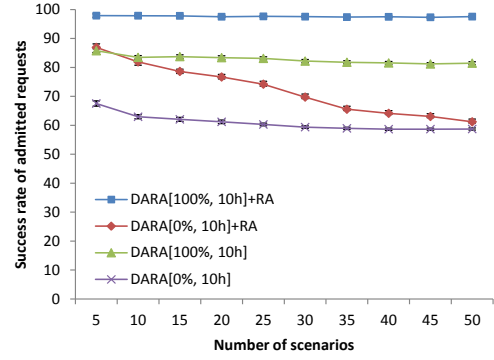
Figure 24a and Figure 24b compare the success rate of admitted requests in the RA approach to the DARA approach for the 25-node topology. In these experiments, failure rates vary from 1h to 20h and backup demands of 0% up to 100% are assessed. In both figures, regardless of the failure rate, the highest success rate comes with 100% backup. It can also be observed that by employing the RA approach, the number of succeeded requests increases significantly up to 6.77 times with 1-hour failure rate and 100% of backup demand. The same trend has been observed for the smaller topology in which the RA approach can provide up to 5.3 times higher success rates.

5.5. Evaluation of execution times

Figure 25 and Figure 26 compare the computational time of the DARA algorithm and the proposed RA approach, using 100% backup capacity. The execution time of the RA approach is the sum of all invocations of the periodic update and the periodic adaptation algorithms and



(a) Percentage of succeeded requests.



(b) Success rate of admitted requests.

Figure 21: Impact of network load and backup demands on the performance of using the RA approach for the 25-node topology.

Table 2: Execution time per algorithm invocation (ms) of main algorithms of the RA approach. Failure rate is 2h.

execution time (ms)	8-node topology			25-node topology		
	AVG	std error	#invocation/24h	AVG	std error	#invocation/24h
UpdateRequestsInfo	0.005	0.001	498.38	0.025	0.006	1578.4
AO	0.28	0.01	475.38	2.61	0.07	1555.4
DARA	51.96	2.05	23	551.4	15.9	23

the execution time of the DARA approach is the sum of all re-scheduling invocations whenever a new scenario enters to the reservation system. Our results indicate that deploying the RA approach increases the execution time by 2.75 and 2.12 times on average in the 8-node and 25-node topologies respectively.

The number of invocations and the average execution time of a single invocation of main algorithms in the RA approach is shown in Table 2. The number of scenario is 20 in 8-node and 50 in 25-node topology and 2h failure rate is used. This evaluation shows that the AO algorithm, with 0.28 ms in the smaller and 2.61 ms in the larger topology, is fast enough to immediately reconfigure the network and react to sudden changes.

6. Conclusions

In order to deliver reliable data transfers, we have previously proposed a resilient advance reservation approach. Offering guaranteed video delivery in media

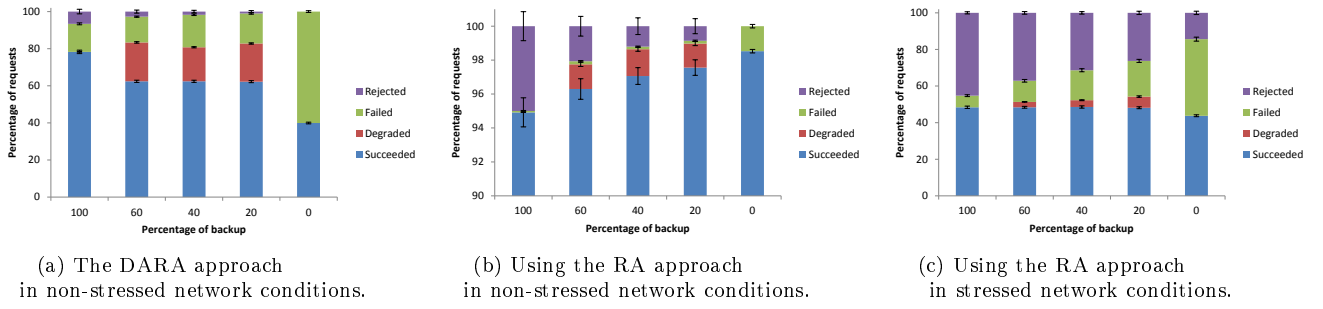


Figure 22: Final state of requests in stressed and non-stressed 8-node topology with a failure rate of 2h.

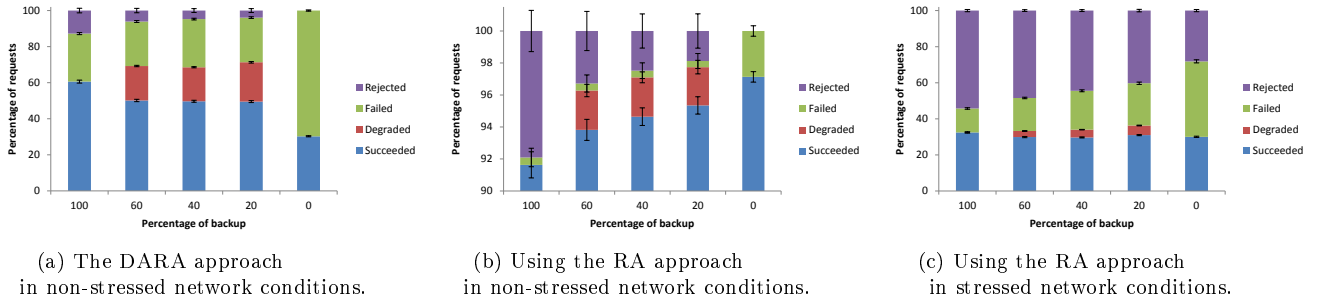


Figure 23: Final state of requests in stressed and non-stressed 25-node topology with a failure rate of 2h.

production networks is of prime importance, however, using redundancy imposes significant performance overheads and extra costs. In this article, we proposed a dual optimization approach for exploiting underutilized network capacities to transfer more data than what has been scheduled as long as no failures are detected. This article deals with the design, development and evaluation of the proposed approach in which a constant monitoring, adaptation and re-optimization is being applied during runtime, taking into account potential failures. The main objective is to mitigate the side-effect of redundant allocations and dynamically reconfigure transmissions in response to sudden changes in network conditions. The experimental results showed that our approach works efficiently both in stable and failure-prone networks. Deploying this approach will noticeably increase the performance of the advance reservation systems by increasing the number of succeeded requests and with computational time of less than 3ms for all evaluated cases, our solution is fast enough to react immediately and re-configure the network in response to sudden changes.

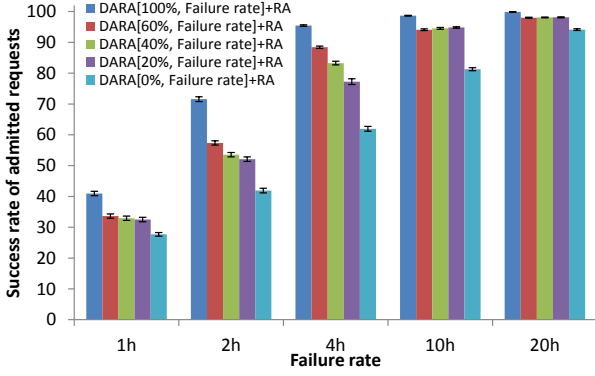
Acknowledgment

The computational resources (Stevin Supercomputer Infrastructure) and services used in this work were provided by the VSC (Flemish Supercomputer Center), funded by Ghent University, the Hercules Foundation and the Flemish Government - department EWl. The research leading to these results has been performed within the context of ICON MECaNO, co-funded by iMinds, a digital

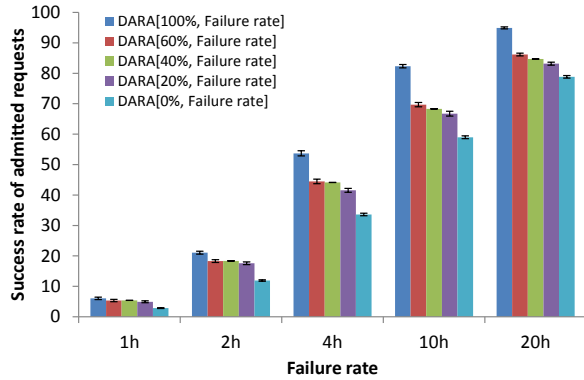
research institute founded by the Flemish Government under grant agreement no. 130646.

References

- [1] Charbonneau N, Vokkarane VM. A survey of advance reservation routing and wavelength assignment in wavelength-routed WDM networks. *IEEE Communications Surveys & Tutorials* 2012;14(4):1037–64.
- [2] Rajah K, Ranka S, Xia Y. Advance reservations and scheduling for bulk transfers in research networks. *IEEE Trans Parallel Distrib Syst* 2009;20(11):1682–97. doi:10.1109/TPDS.2008.250.
- [3] ICON MECaNO project. <http://www.iminds.be/en/projects/mecano/>; Jan 2014 - Dec 2015. Accessed: 2016-04-06.
- [4] Barshan M, Moens H, Famaey J, De Turck F. Algorithms for advance bandwidth reservation in media production networks. In: *IFIP/IEEE International Symposium on Integrated Network Management (IM)*. 2015, p. 183–90. doi:10.1109/INM.2015.7140291.
- [5] Barshan M, Moens H, Famaey J, De Turck F. Deadline-aware advance reservation scheduling algorithms for media production networks. *Computer Communications* 2016;77:26–40. doi:10.1016/j.comcom.2015.10.016.
- [6] Sakhaf S, Barshan M, Tavernier W, Moens H, Colle D, Pickavet M. Resilient algorithms for advance bandwidth reservation in media production networks. In: *International Conference on the Design of Reliable Communication Networks (DRCN)*. IEEE; 2016, p. 130–7.
- [7] Józsa BG, Orincsay D. Shared backup path optimization in telecommunication networks. In: *International Conference on the Design of Reliable Communication Networks (DRCN)*. 2001, p. 251–7.
- [8] Haahr JT, Stidsen T, Zachariasen M. Heuristic methods for shared backup path protection planning. In: *4th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*. IEEE; 2012, p. 712–8.



(a) Success rate of using the RA approach.



(b) Success rate of the DARA approach.

Figure 24: Comparing the success rate of the DARA and the RA approaches with different backup demands for the 25-node topology.

- [9] Depoorter W, Vanmechelen K, Broeckhove J. Advance reservation, co-allocation and pricing of network and computational resources in grids. *Future Generation Computer Systems* 2014;41:1–15.
- [10] Bochenina K, Butakov N, Boukhanovsky A. Static scheduling of multiple workflows with soft deadlines in non-dedicated heterogeneous environments. *Future Generation Computer Systems* 2016;55:51–61.
- [11] Bai H, Shaban K, Khodair M, Gu F, Crichigno J, Khan S, et al. Overlay network scheduling design. *Computer Communications* 2016;82:28–38.
- [12] Guok C, Engineer EN, Robertson D. Esnet on-demand secure circuits and advance reservation system (OSCARS). In: *Internet2 Joint Techs Workshop*, Salt Lake City, Utah. 2005,.
- [13] Gibbard B, Katramatos D, Yu D. Terapaths: end-to-end network path QoS configuration using cross-domain reservation negotiation. In: *3rd International Conference on Broadband Communications, Networks and Systems (BROADNETS)*. IEEE; 2006, p. 1–9.
- [14] Chunlin L, Xiu ZJ, Layuan L. Resource scheduling with conflicting objectives in grid environments: Model and evaluation. *Journal of Network and Computer Applications* 2009;32(3):760–9.
- [15] Gu J, Katramatos D, Liu X, Natarajan V, Shoshani A, Sim A, et al. Stornet: Co-scheduling of end-to-end bandwidth reservation on storage and network systems for high-performance data transfers. In: *IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE; 2011, p. 121–6.
- [16] Kuri J, Puech N, Gagnaire M, Dotaro E. Routing foreseeable lightpath demands using a tabu search meta-heuristic. In: *IEEE Global Telecommunications Conference (GLOBE-*

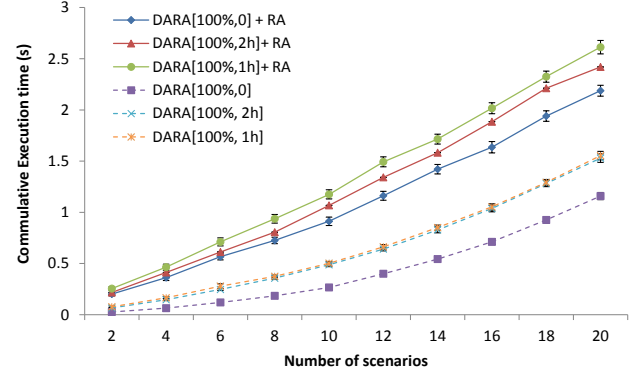


Figure 25: Evaluation of execution times for the 8-node topology.

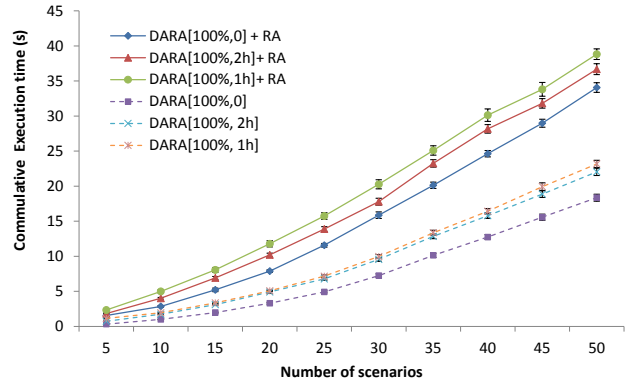


Figure 26: Evaluation of execution times for the 25-node topology.

- COM'02); vol. 3. IEEE; 2002, p. 2803–7.
- [17] Kuri J, Puech N, Gagnaire M, Dotaro E, Douville R. Routing and wavelength assignment of scheduled lightpath demands. *IEEE Journal on Selected Areas in Communications* 2003;21(8):1231–40.
- [18] Zheng J, Mouftah HT. Supporting advance reservations in wavelength-routed WDM networks. In: *Tenth International Conference on Computer Communications and Networks*. IEEE; 2001, p. 594–7.
- [19] Zheng J, Mouftah HT. Routing and wavelength assignment for advance reservation in wavelength-routed WDM optical networks. In: *IEEE International Conference on Communications (ICC)*; vol. 5. IEEE; 2002, p. 2722–6.
- [20] Xie C, Alazemi H, Ghani N. Rerouting in advance reservation networks. *Computer Communications* 2012;35(12):1411–21.
- [21] Balman M, Chaniotakis E, Shoshani A, Sim A. A flexible reservation algorithm for advance network provisioning. In: *2010 International Conference for High Performance Computing, Networking, Storage and Analysis (SC)*. IEEE; 2010, p. 1–11.
- [22] Balman M. Advance resource provisioning in bulk data scheduling. In: *IEEE 27th International Conference on Advanced Information Networking and Applications (AINA)*. IEEE; 2013, p. 984–92.
- [23] Watanabe T, Omizo T, Akiyama T, Iida K. 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)*. IEEE; 2015, p. 211–8.
- [24] Tanwir S, Battestilli L, Perros H, Karmous-Edwards G. Dynamic scheduling of network resources with advance reservations in optical grids. *International Journal of Network Management* 2008;18(2):79–106.
- [25] Burchard LO, Heiss HU, Linnert B, Schneider J, De Rose CA. VRM: a failure-aware grid resource management system. *Inter-*

- national journal of high performance computing and networking 2008;5(4):215–26.
- [26] Latchoumy P, Khader P, Abdul S. Job scheduling with failure prevention strategies in grid computing environment. *International Journal Of Research In Advance Technology In Engineering* 2013;1.
 - [27] Nazir B, Qureshi K, Manuel P. Replication based fault tolerant job scheduling strategy for economy driven grid. *The Journal of Supercomputing* 2012;62(2):855–73.
 - [28] Cholda P, Jaglarz P. Optimization/simulation-based risk mitigation in resilient green communication networks. *Journal of Network and Computer Applications* 2016;59:134–57.
 - [29] Cavdar C, Tornatore M, Buzluca F, Mukherjee B. Dynamic scheduling of survivable connections with delay tolerance in WDM networks. In: *IEEE INFOCOM Workshops*. 2009, p. 1–6. doi:10.1109/INFCOMW.2009.5072134.
 - [30] Wang B, Li T. Survivable scheduled service provisioning in WDM optical networks with iterative routing. *Optical Switching and Networking* 2010;7(1):28–38.
 - [31] Jaekel A, Chen Y, Bari A. Stable logical topologies for survivable traffic grooming of scheduled demands. *Journal of Optical Communications and Networking* 2010;2(10):793–802.
 - [32] Barshan M, Moens H, Volckaert B, De Turck F. Design of a dynamic adaptive reservation system in media production networks. In: *IEEE/IFIP Network Operations and Management Symposium (NOMS)*. IEEE; 2016, p. 1149–52.
 - [33] Barshan M, Moens H, Volckaert B. Dynamic adaptive advance bandwidth reservation in media production networks. In: *IEEE NetSoft Conference and Workshops (NetSoft)*. IEEE; 2016, p. 58–62.
 - [34] Cormen T. *Introduction to Algorithms*. MIT Press; 2009. ISBN 9780072970548.
 - [35] Edmonds J, Karp RM. Theoretical improvements in algorithmic efficiency for network flow problems. *Journal of the ACM (JACM)* 1972;19(2):248–64.
 - [36] Smith W, Foster I, Taylor V. Scheduling with advanced reservations. In: *14th International Parallel and Distributed Processing Symposium (IPDPS)*. IEEE; 2000, p. 127–32.
 - [37] Ernemann C, Hamscher V, Yahyapour R. Economic scheduling in grid computing. In: *Job Scheduling Strategies for Parallel Processing*. Springer; 2002, p. 128–52.
 - [38] Cao J, Zimmermann F. Queue scheduling and advance reservations with COSY. In: *18th International Parallel and Distributed Processing Symposium*. IEEE; 2004, p. 63.
 - [39] Luke S, Cioffi-Revilla C, Panait L, Sullivan K, Balan G, Mason: A multiagent simulation environment. *Simulation* 2005;81(7):517–27.
 - [40] Knight S, Nguyen HX, Falkner N, Bowden R, Roughan M. The internet topology zoo. *IEEE Journal on Selected Areas in Communications* 2011;29(9):1765–75.
 - [41] Alam Khan MM. Multi-path link embedding for survivability in virtual networks. Master's thesis; University of Waterloo; 2015.