A Resource Reservation Scheme for Synchronized Distributed Multimedia Sessions

by W. Zhao, S.K. Tripathi

T.R. 97-14

Sponsored by
the National Science Foundation
Engineering Research Center Program,
the University of Maryland,
Harvard University,
and Industry

A Resource Reservation Scheme for Synchronized Distributed Multimedia Sessions [*] *(Extended Abstract)*

Wei Zhao and Satish K. Tripathi
Department of Computer Science
University of Maryland at College Park
College Park, Maryland 20742
Email: {zw,tripathi}@cs.umd.edu

Abstract

Guarantees of services in a networked environment are provided by the proper allocation and utilization of network and system resources. In this paper, we propose a resource reservation scheme for a class of multimedia applications. We characterize this class of multimedia applications as synchronized distributed multimedia sessions, which we believe represent a large number of multimedia applications. In addition to this particular class of applications, the reservation scheme can also be applied to other applications with synchronized resource requirements. Mechanisms are devised so that the scheme can find a feasible reservation whenever one exists, in contrast to existing protocols where request can fail unpredictably even when there is available resource. Based upon a layer of resource abstraction, the scheme suits well with today's heterogeneous network environment.

1 Introduction

Traditional best-effort services provided by computer systems are inadequate for new generation applications such as multimedia applications. Guarantees on the quality of service are needed. A lot of research work has been done at the system level to achieve better resource utilization while providing service guarantees expressed in terms of throughput, delay, delay jitter, etc, for a given set of resources.

In this paper, we propose a scheme to translate the service guarantees at the system level into the guarantees at the application level for multimedia applications. In particular, we propose a resource reservation scheme for a large class of multimedia applications. We characterize this class of multimedia applications as synchronized distributed multimedia sessions (SDMS's). In addition to this particular class of applications, the reservation scheme can also be applied to other applications with synchronized resource requirements.

An SDMS consists of a number of multimedia streams from distributed sources whose presentation time intervals are interrelated. An SDMS executes successfully only when all the component streams are presented with the specified qualities during their respective time intervals as determined by its synchronization requirements. The synchronization requirements of SDMS implies resource requirements in a synchronized fashion. For example, if stream B is to be presented immediately after stream A, then once stream A finishes presentation, the resource required by stream B must be available for B's presentation. Note that, the resources required by multimedia applications include both network resources for the data transmission and

[*] Prepared through collaborative participation in the Advanced Telecommunication/Information Distribution Research Program (ATIRP) Consortium sponsored by the U.S. Army Research Laboratory under Cooperative Agreement DAAL01-96-2-0002.

end-system resources for the data processing.

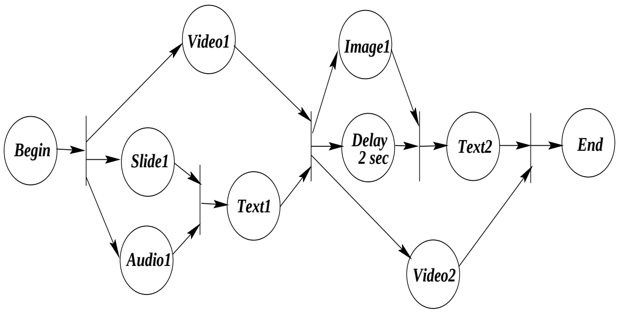

Figure 1: A Petri-Net Synchronization Representation

Currently available resource reservation protocols such as RSVP and STII are inadequate to provide this kind of resource requirement: the resources reserved by these protocols are time-invariant over a session's life time and there is no way to do a synchronized resource reservation where resource requirement changes during a session; and, the reservation requests are of an immediate fashion, each request either succeeds with the resource allocated immediately or fails due to insufficiency of currently available resource. It is up to the application to decide when or whether to resubmit the reservation request if it fails. Furthermore, there is no way of knowing when the request will succeed or whether it will succeed at all.

2 Methodology

We propose a resource reservation scheme that overcomes the above problems.

Essentially, the resource reservation scheme for an SDMS collects the individual resource requirements generated by multimedia objects and inquires resources about the satisfiabilities of individual resource requirements. After collecting temporal intervals in which individual resource requirements can be satisfied, the scheme integrates them to provide the resource guarantee for the whole SDMS by determining the feasible session start time via temporal intersection finding algorithm. Instead of the *yes/no* type of responses on resource satisfiabilities that other resource reservation protocols use, we express resource satisfiabilities in terms of available temporal intervals. The use of temporal intervals enables us to coordinate available resources for

1. Client c sends *Object Reservation Request* for all multimedia objects to their locations. That is, for each object o, c sends object reservation request $(o, I(o), Q(o), c)$ to the location of the object $E(o)$, where I(o) and Q(o) represent the interval and quality requirements of object o respectively.

2. For each source (called Server) s that receives the object reservation request (o, i, q, c),

   2.1. Based upon the object o's properties, s determines the resource requirement $R(o, q, c, i)$ for its presentation.

   2.2. s sends the *Resource Inquiry Request* for the resource requirement computed in 2.1 to the resource managers for each resource involved, including itself.

3. For each resource manager that receives the resource inquiry request, it checks its own resource reservation table and sends back to client c the *Start Times Intervals* when the requested resource is available.

4. When client c gets back all the responses of resource inquiry requests from resource managers, it runs the *Interval Intersection Finding Algorithm* to find the start time when all resource requirements can be satisfied.

   4.1. If no intersection is found, it returns FAIL to the application.

   4.2. If a start time is found, it sends back to the resource managers the *Confirmation Message* with the start time to confirm the reservation. And returns the start time to the application.

5. When resource managers receive the reservation confirmation messages, they modify their resource reservation tables to mark the resources as reserved.

Figure 2: The Resource Reservation Scheme

the whole SDMS on the temporal axis and makes it possible for in-advance reservations.

Compared to RSVP and STII, our scheme reserves resources in a synchronized fashion as required by an SDMS. It is also an in-advance resource reservation scheme. When the required resource is not available at the time of the request, the scheme returns the time that the resource will be available and reserve the resources accordingly. Unlike many multimedia presentation schedule generation techniques, our scheme, like RSVP, leaves the resource satisfiability decisions to the resource side, making the scheme very flexible.

```
function Intersection
        $T \leftarrow B(0,0)$
        for $k \leftarrow 0$ to $N-1$ do
                $C(k) \leftarrow 0$
                $T(k) \leftarrow -1$
        end for
        $i \leftarrow 1$
        while TRUE do
                if $T = T(i)$ return $T$
                $j \leftarrow C(i)$
                while ($j < 0$ or $E(i,j) < T$) do
                        if $j < I(i) - 1$ then $j \leftarrow j + 1$
                        else return FAIL
                $C(i) \leftarrow j$
                $T \leftarrow max(B(i,j), T)$
                $T(i) \leftarrow T$
                $i \leftarrow (i+1)$ mod N
        end while
end
```

Figure 3: The Interval Intersection Finding Algorithm

The temporal interval intersection finding algorithm used in the scheme is given in figure 3. The goal of the algorithm is to find a time instant that falls in at least one interval of every resource. In figure 3, N is the number of resources; T is the current candidate intersection; $I(i)$ is the number of intervals of resource i; $B(i,j)$, $E(i,j)$ are the end points of the jth interval of resource i; $C(i)$ and $T(i)$ keep track of the current interval and candidate intersection on resource i, respectively.

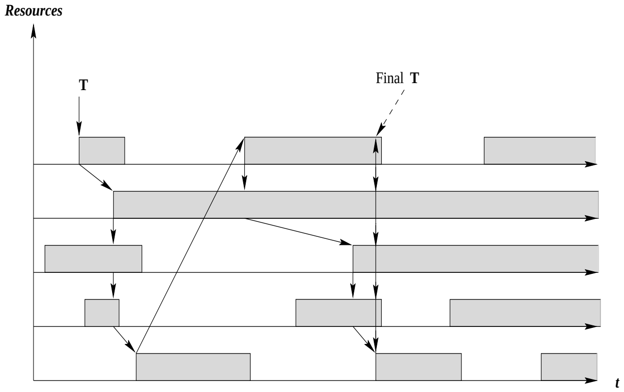

Figure 4: An Example of Time-Axis Walk in the Insection Finding Algorithm

Figure 4 shows a running instance of the algorithm on a sample set of intervals. Basically, the algorithm is a time-axis walk by looping through resources and inspecting the intervals in order. An intersection is found once the candidate intersection T remains the same after a loop through the set of resources.

For improved efficiency, we devised a early start and greedy progress strategy for running the intersection finding algorithm. The idea is that the algorithm does not have to wait till all intervals are collected. Instead, it can start running as soon as the first interval arrives and as long as the inspected interval is available. Immediately after an intersection is found, the algorithm stops and sends messages to resource managers to stop the generation of more intervals. The greedy strategy reduces the reservation latency by exploiting the execution structure of the algorithm.

Due to the delay between resource inquiry and commit messages, resource allocation may have changed between these two time instants. A commit message may fail due to this kind of resource conflict. To handle multiple concurrent resource reservation requests with pos-

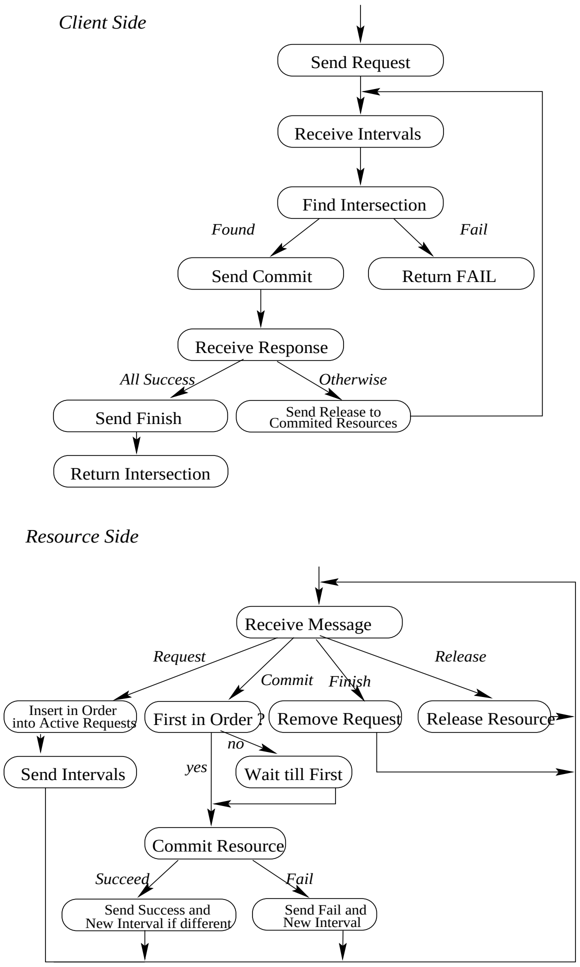

Figure 5: Partially-Ordered Optimistic Concurrency Handling Technique

sible resource conflicts, a few concurrency handling techniques for the reservation scheme are proposed. The optimistic technique checks for conflicts only when the commit messages are received. Start time is recalculated if the commit message fails. However, the scheme can cause infinite delay. For example, there could be some requests whose commit messages always fail because of resource conflicts with other incoming requests and never get completed. A partially ordered optimistic technique is proposed to order the sequence of commiting requests to ensure that the infinite delay situation will not happen. The partial order is a projection of a total order on active requests on some resource. The technique is proved to guarantee progress. A relaxed version of partially ordered technique further enhanced the parallelism while maintaining the guaranteed progress property. The partially ordered technique is shown in figure 5.

For the resource reservation scheme, we did not assume any particular resource management and scheduling policies. We use only the resource satisfiability decisions made by the resources to provide resource guarantees, no matter what techniques resource managers use. As a result of this design principle, the resource reservation scheme can be applied to a wide range of heterogeneous environments. The flexibility also allows the seamless integration of any new resource management and scheduling techniques as they are developed, thus providing good progressive scalability in fastly evolving environments.

3 Conclusion

We have proposed a resource reservation scheme capable of scheduling and reserving the resource requirements generated by a class of multimedia applications we characterize as synchronized distributed multimedia sessions. The scheme is not an attempt to solve individual resource management problems in specific systems, instead, it utilizes and integrates the functionalities of individual resource management enitities to provide a global resource guarantee. Our scheme is advantageous to existing reservation protocols in a few ways. Based on the idea of advance reservation, the scheme can handle the synchronized resource allocations as required by many multimedia applications. Second, the scheme is guaranteed to find a feasible schedule of resources whenever one exists and also eliminates the problem of application-level reservation failure-retry which may adversely affect the network utilization. Furthermore, instead of simply failing the entire reservation as in existing protocols in the event of resource contention, we proposed concurrent reservation techniques that guarantees progress.

This work can be extended in several ways. First, some multimedia applications have soft or elastic temporal relationships, where the synchronization among multimedia objects is not fixed but rather can vary within certain constraints. Whether and/or how the proposed reservation scheme can be extended to incorporate this scenario is worth further study. Second, the scheme uses advance reservation which increases the resource management complexity, especially at reservation stage. Future work needs to be done to study and quantify the increased complexity. Third, the reservation dynamics of the reservation scheme can be studied, from which potential problems can be found and addressed. [†]

4 References

1. L. Zhang, S. Deering, D. Estrin, S. Shenker, D. Zappala, *RSVP: A New Resource Reservation Protocol*, IEEE Network, September 1993.

2. C. Topocic, *Experimental Internet Stream Protocol Version 2 (ST-II)*, IETF RFC 1190, October 1990.

3. L. Delgrossi and L. Berger, *Internet Stream Protocol Version 2 (ST2) Protocol Specification - Version ST2+*, August 1995.

4. L. Delgrossi, R. Herrtwich, C. Vogt, L. Wolf, *Reservation Protocols for Internetworks: A Comparison of STII and RSVP*, Proceedings of 4th International Workshop on Network and Operating System Support for Digital Audio and Video, November 1993.

5. L. Wolf, L. Delgrossi, R. Steinmetz, S. Schaller, H. Wittig, *Issues of Reserving Resources in Advance*, Proceedings of 5th International Workshop on Network and Operating System Support for Digital Audio and Video, April 1995.

6. L. Hardman, D. Bulterman, G. Rossum, *The Amsterdam Hypermedia Model*, Communications of the ACM, Vol 37, No. 2, February 1994.

7. N. Hirzalla, B. Falchuk, A. Karmouch, *A Temporal Model for Interactive Multimedia Scenarios*, IEEE Multimedia, Fall 1995.

8. D. Ferrari A. Gupta, G. Ventre, *Distributed Advance Reservation of Real-time Connections*, Proceedings of 5th International Workshop on Network and Operating System Support for Digital Audio and Video, April 1995.

[†] The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government.