IOSR Journal of Engineering
Apr. 2012, Vol. 2(4) pp: 636-639

Throughtput Competitive Advance Reservation Using Polynomial Time

S. Soniya Priya and M. Jayasudha
Dept. of Information Techonology, Sri Venkateshwara College of Engineering, Chennai, India.

Abstract: **In response to the high throughput needs of grid and cloud computing applications, several production networks have recently started to support advance reservation of dedicated circuits. An important open problem within this context is to devise advance reservation algorithms that can provide provable throughput performance guarantees independently of the specific network topology and arrival pattern of reservation requests. In this paper, we first show that the throughput performance of greedy approaches, which return the earliest possible completion time for each incoming request, can be arbitrarily worse than optimal. Next, we introduce two new online, polynomial-time algorithms for advance reservation, called BATCH ALL and BATCH LIM. Both algorithms are shown to be throughput-optimal through the derivation of delay bounds for bandwidth augmented networks. The BATCH LIM algorithm has the advantage of returning the completion time of a connection immediately as a request is placed, but at the expense of looser delay performance. The propose system called BATCH ALL+ is a simple approach that limits path dispersion, i.e., the newly arrived batch can be combined with the currently running batch if the destination is same, so time consumption is less when compare to BATCH ALL and BATCH LIM.**

Keywords: High Speed Networks, Routing, Scheduling.

I. INTRODUCTION

Several protocol and algorithms have been proposed in the literature to support advance reservation. However, none of them provides throughput guarantees. Instead, most are based on greedy approaches, whereas each request is allocated a path guaranteeing the earliest completion time at the time request is placed. The most important property of advance reservation is to offer hosts and users the ability to reserve in advance dedicated paths to connect their resources. Advance reservation services have been successfully tested in number experiments projects.

In this paper, first the fundamentals of greedy algorithm is uncovered specially if there exit network topologies and request patterns for which the maximum throughput of these algorithm can be smaller than the optimal throughput as network size increases.

Next a new online algorithm polynomial time called Batch All, Batch Lim, Batch All+ is used, that provably achieves maximum throughput for any network topology. Instead of immediately reserving a path for each incoming request as in greedy algorithm, polynomial time algorithm accumulates several arrivals in a batch and assigns a more efficient set of flow paths to all the batches.

The Batch All algorithm does not return the connection completion time to the user at the time a request is placed, it is returned only when the connection is actually starts. The algorithm called Batch Lim provides the completion time immediately as a request is placed but this algorithm operates by limiting the length of each batch. Third the Batch All+ algorithm is used, the main idea of this algorithm whenever a job request arrives, it first checks if whether it can allocate it to current batch without affecting the completion time of the batch, if yes it allocate bandwidth job.

An algorithm is said to be solvable in polynomial time if the number of steps required to complete the algorithm for a given input is O (n^k) for some nonnegative integer k, where n is the complexity of the input. Polynomial-time algorithms are said to be "fast."Most familiar mathematical operations such as addition, subtraction, multiplication, and division, as well as computing square roots, powers, and logarithms, can be performed in polynomial time.

Too large path dispersion may be undesirable, as it may entail fragmenting a file into a large number of segments and reassembling them at the destination. To address this issue, a simple approach based on the max-flow min-cut theorem, that limits the number of parallel paths while bounding the maximum reduction factor in transmission throughput and prove that this bound is tight. Then propose two algorithms, Batch All Disp and Batch Lim Disp, based upon Batch All and Batch Lim, respectively. These algorithms perform similarly to the original algorithm in terms of the batching process. However, after filling each batch, the algorithm will limit the dispersion of each flow. Although these algorithms are not throughput optimal anymore, they are still throughput competitive.

Furthermore, proposed and evaluated a modified version of the algorithm called Batch All+, whose performance is even closer to the bound with respect to path dispersion. An excellent performance can be achieved with as few as five or so parallel path per connection.

II.RELATED WORK

This paper [2] provided an initial look at how support for advance reservations affects the complexity of the path selection process in networks. Advance reservations are likely to become increasingly important as networks and distributed applications become functionally richer, and there have been a number of previous works and investigations that explored various related aspects. The impact of advance reservations on path selection is a topic that has been left largely untouched. It investigates several service models for advance reservations, which range from the traditional basic model of reserving a given amount of bandwidth for some time in the future, to more

sophisticated models aimed at increasing the flexibility of services available through advance reservations. The focus is primarily on the issue of computational complexity when supporting advance reservations and in that context. This derives a number of algorithms and/or intractability results for the various models.

This paper [7] describes the Schedule resources on Grids is a well-known problem. The extension of Grids to Lambda Grids requires scheduling of lambdas is end-to-end high-speed circuits propose a heuristic for the scheduling of lambdas specifically for file transfers, given that many science applications require high-throughput transfers of large files. And call this heuristic Varying-Bandwidth List Scheduling (VBLS) because the scheduler returns a time-range-capacity (TRC) allocation vector with varying bandwidth levels assigned for different time ranges within the duration of a transfer. The advantage of VBLS over a fixed-bandwidth allocation scheme is that it allows the scheduler to backfill any holes left in resource allocations. Such a scheme is enabled by end host applications specifying the file size in their transfer requests. To characterize VBLS analytical models and ran simulations is used. The results show that VBLS performance is close to packet-switching performance which means that file transfers can take advantage of bandwidth that becomes available subsequent to the start of transfers a critical drawback of typical fixed-bandwidth allocation schemes in circuit-switched networks.

This paper [8] provides many application domains that exists a need to aggregate information from information repositories distributed around the world. In an effort to better link these resources in a unified manner, middleware researchers put forth the notion of a grid. With the context applied to bioinformatics they consider problem of aggregating files from distributed databases to a (grid) computing node over a lambda grid. The challenge is to identify concurrent routes (circuit-switched paths) in the lambda-grid network, along which files should be transmitted, and to schedule the transfers of these files over their respective circuits. To address this challenge, they propose a hybrid approach that combines off-line and on-line scheduling. The approach first constructs an off-line schedule based on past profiling of transfer rates. Then as files are being transferred the schedule is modified on-line, depending on the amount of time that it actually took to transfer the files. The objective is to minimize the total time required for data aggregation. To demonstrate the effectiveness of this approach they present experimental results using grid nodes running over an emulated lambda-grid topology.

Paper [3] focuses on the design and analysis of scheduling approaches for Optical Flow Switching (OFS) serving high performance applications with very stringent time deadline Constraints. In particular it attempt to meet setup times only slightly longer than one roundtrip time with networks at moderate to high loading. It proposes three possible scheduling mechanisms for Optical Flow Switching (OFS) connection setup in a Wave Division Multiplexing (WDM) network: (i) a simple algorithm, which awards pre-emptive priority to applications requiring time deadline performance (ii) a multi-path probing mechanism using only coarse average loading information but without pre-emption and (iii) a multi-path probing mechanism using periodically updated network state information and without pre-emption. The updating scheme calls for a slow control plane, which refreshes and broadcast network states only periodically on the order of seconds or longer. This results show that for a low blocking probability, the update interval must be a Fraction of the mean service time of transactions. Conclusion of this algorithm is a combination of both slow centralized and fast distributed processes delivers an efficient and scalable control design for a high-speed transport network of the future.

Many papers have discussed the issue of path dispersion and attempted to achieve good throughput with limited dispersion. A survey of some results in this field is given in [28]. In [29] and [30] heuristic methods of controlling multipath routing and some quantitative measures are presented. As far as the proposed work the first formal treatment allowing the approximation of a flow using a limited number of path at any desired level of accuracy.

III. PROPOSED ARCHITECTURE DESIGN

The disadvantage in existing system Batch Lim can return the completion time of connection immediately as soon as the request is placed in the path but at expense it will delay the performances. This can be overcome by using the polynomial time algorithm in Batch All+. Splitting of packet is carried as per the batch limit, but rather waiting for second batch, if some packets in second batch to be send to the same destination. If the advanced reservation and completion time of packet s known once the packet is placed in the path. Throughput Time s considered to be faster than Batch All and Batch Lim.

A. Batch All Algorithm

In Batch All Process, the server which acts as the intermediate of Source and the destination will receive the packets from the source, and split the batches as per its batch limit. After allocating the Batch Split-ups the Process of dispatch of packets starts, the estimated time of packets delivery is announced to the Source only after that particular batch is started. Only after finishing the first batch, the second batch of packet delivery is started.

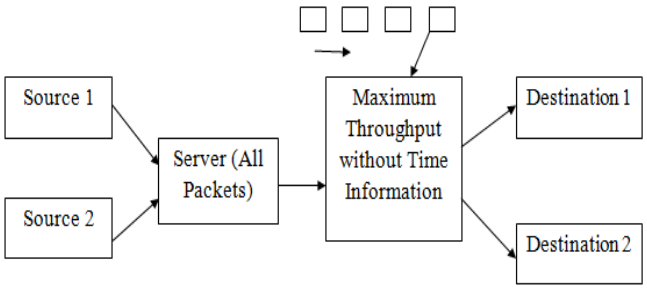

Figure 1: Batch All Block Diagram

Step1: Set tc to 0.
Step2: When a request l={s, d, f} arrives at clk=t, give an immediate connection starting time and a connection ending time of tc = t+maxflow (G, l).
Step3: Set L<-null.
Step4: While clk < tc,

- If a request l = {s,d,f}arrives when clk = t:
- Set L<- LUl(i.e.,add the job to the waiting batch);
- Mark tc as its connection starting time.

Step5: At time clk = tc,

- If L is empty (i.e., there is no pending request), go back to step2.
- Else assign a connection ending time tc = tc +maxflow (G, L) to all requests in the batch L and go bak to step3.

B. Batch Lim Algorithm

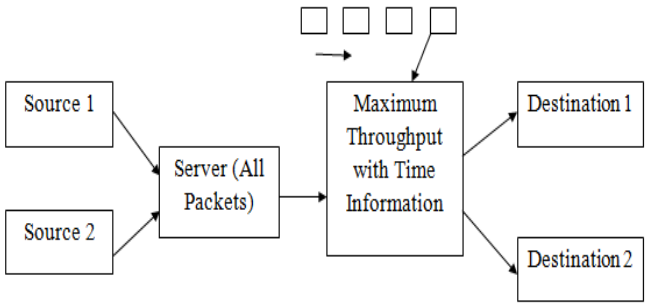

Figure 2: Batch Lim Block Diagram

Batch Lim Algorithm gives larger delay ratio guarantee, but which always returns the completion time as soon as the request is arrived. It also provides throughput optimal. It is as same as the Batch All Process, except the time of packet delivery is announced initially itself.

Step1: if no job is currently running, then:

- Set M = maxflow(G,l);
- Set t2 = t +M.
- Assign job l to interval [t1,t2] and exit.

Step2: Else check if job l can be assigned to an existing interval.

- Set i = 2.
- While I <= n-1:
- If l can be fitted into interval [ti,ti+1] (i.e., multicomm (G, Li U l, ti+1-ti) = true) then assign job l to [ti, ti+1] and exit;
- Else set i = i+1.

Step3: Else create new interval

- Set M = maxflow(G,l)
- If tn -t <M, then tn+1 = tn+M.
- Else tn+1 = 2tn-t.
- Assign job l to interval [tn,tn+1] and exit.

C. Batch All+ Algorithm

In the Batch All+ Process, the server which acts as the intermediate of Source and the destination will receive the packets from the source, and split the batches as per its batch limit. But rather mere making wait of the second batch, if some packets are in next batch to be send to the same destination, those packets are added to the first batch then the delivery of those packets are achieved. This process will utilize the same bandwidth, by which some more packets can also be sent to the destination. Here Time delivery is reported during the initial stage of packets receipt of the server.

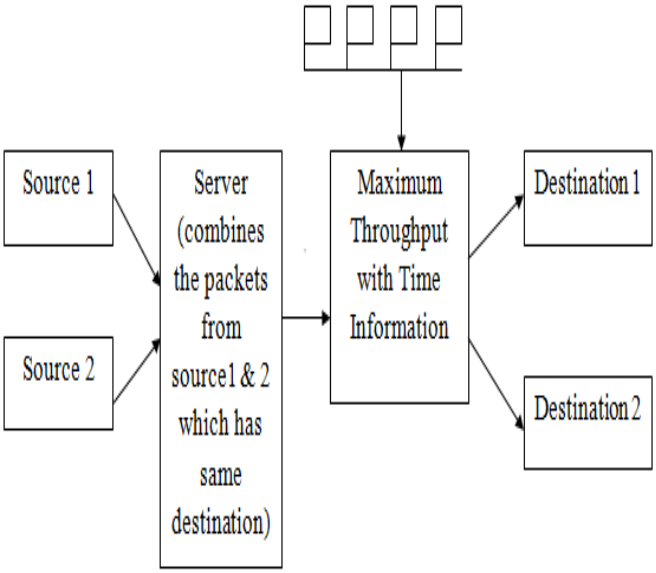

Figure 3: Batch All+ Block Diagram

Step1: While clk < tc,

- If request l = {s, d, f} arrives when clk=t:
- If maxflow(G',l) <=tc – t' then,
- Add the job to the running batch such that it completes simultaneously with the rest of running jobs;
- Deduct all bandwidth assigned to l from G'.
- Otherwise,
- Set L <- LUl (i.e., add it to the waiting batch);
- Mark tc as its connection starting time

IV. CONCLUSION

The problem of devising throughput-optimal and throughput-competitive algorithms for networking architectures supporting advance reservation is considered as a problem of particular relevance to modern grid and cloud computing applications. After showing the limitations of greedy approaches, two new online algorithms for advance reservation, called Batch All and Batch Lim, which are guaranteed to achieve optimal throughput performance were proposed. Specifically, it is proved that both algorithms bound the ratio of the maximum delay of any request in bandwidth augmented networks to the maximum delay of any request. While it has the distinct advantage of returning the completion time of a connection immediately as a request is placed. Path dispersion is essential to achieve full network utilization. However, splitting a transmission into too many different paths may render a flow-based approach inapplicable in many real-world environments. Thus, presented a rigorous theoretical approach to address the path dispersion problem and presented a method for approximating the maximum multi commodity flow using a limited number of paths.

REFERENCES

[1] R. Cohen, N. Fazlollahi, and D. Starobinski, 2009,_Path Switching and Grading Algorithm For Advance Channel Reservation Architecture_,IEEE/ACM Trans.Netw.,Vol.17, no.5,pp.1684-1695.

[2] A.Banerjee, W. Feng, B.Mukherjee, and D.Ghosal, 2005,_Routing and Scheduling Larger File Transfer Over Lambda Grids_, in proc. 3rd PFLD Net, Lyon,France.

[3] V.Chan, G.Weinchenberg,& M. Medard, 2006, _Optical Flow Switching_, BROADNETS, San Jose, CA,pp, 1-8.

[4] L.Lewis- Eytan, J.Naor, and A.orda, 2004 _Admission Control in Networks with Advance Reservation_, Vol.40, pp.293-304.

[5] R.Guerin and A. Orda, 2000, _Networks With Advance Reservation: The Routing Perspective_, in proc IEEE Vol.1,

[6] M.Andrews, A.Feranandez, A.Goel and L.Zhang, 2005, _Source Routing and Scheduling in Packets Networks_, J.ACM, Vol.52, No.4, pp.582-601.

[7] L.Burchard, 2005, _Network with advance Reservation: Application Architecture and Performance, J.Netw. Syst.Manage., Vol.13, pp.429-449.

[8] L.Lweis-Eytan, J.Naor, and A.Orda, 2004,_Admission Control In Networks With Advance Reservation, Vol.40, pp.293-304.

[9] S. Figueira, N. Kaushik, S. Naiksatam, S. Chiappari, and N. Bhatnagar, "Advance reservation of lightpaths in optical-network based grids," in *Proc. ICST/IEEE Gridnets*, San Jose, CA, Oct. 2004.

[10] A. Goel, M. Henzinger, S. Plotkin, and E. Tardos, "Scheduling data transfers in a network and the set scheduling problem," in *Proc. 31st Annu. ACM Symp. Theory Comput.*, 1999, pp. 189–199.

[11] J. Zheng and H. T. Mouftah, "Routing and wavelength assignment for advance reservation in wavelength-routed WDM optical networks," in *Proc. IEEE ICC*, 2002, vol. 5, pp. 2722–2726.

[12] A. Goel, M. R. Henzinger, and S. A. Plotkin, "An online throughputcompetitive algorithm for multicast routing and admission control," *J.Algor.*, vol. 55, no. 1, pp. 1–20, 2005.

[13] B. Awerbuch, D. Holmer, H. Rubens, and R. D. Kleinberg, "Provably competitive adaptive routing," in *Proc. IEEE INFOCOM*, 2005, pp.631–641.

[14] W. Whitt, "The impact of a heavy-tailed service-time distribution upon the M/GI/s waiting-time distribution," *Queue. Syst.*, vol. 36, no. 1/3, pp.71–87, 2000