Advance Network Reservation and Provisioning for Science

Mehmet Balman[1,3], **Evangelos Chaniotakis**[2], **Arie Shoshani**[1], **Alex Sim**[1]

[1]Computational Research Division, Lawrence Berkeley National Laboratory, CA 94720, USA
[2]Energy Sciences Network, Lawrence Berkeley National Laboratory, CA 94720, USA
[3]Department of Computer Science, Louisiana State University, LA 70803, USA

July 2009

We are witnessing a new era that offers new opportunities to conduct scientific research with the help of recent advancements in computational and storage technologies. Computational intensive science spans multiple scientific domains, such as particle physics, climate modeling, and bio-informatics simulations. These large-scale applications necessitate collaborators to access very large data sets resulting from simulations performed in geographically distributed institutions. Furthermore, often scientific experimental facilities generate massive data sets that need to be transferred to validate the simulation data in remote collaborating sites. A major component needed to support these needs is the communication infrastructure which enables high performance visualization, large volume data analysis, and also provides access to computational resources. In order to provide high-speed on-demand data access between collaborating institutions, national governments support next generation research networks such as Internet 2 [1] and ESnet (Energy Sciences Network) [2]. Delivering network-as-a-service that provides predictable performance, efficient resource utilization and better coordination between compute and storage resources is highly desirable. In this paper, we study network provisioning and advanced bandwidth reservation in ESnet for on-demand high performance data transfers. We present a novel approach for path finding in time-dependent transport networks with bandwidth guarantees. We plan to improve the current ESnet advance network reservation system, OSCARS [3], by presenting to the clients, the possible reservation options and alternatives for earliest completion time and shortest transfer duration.

The Energy Sciences Network (ESnet) provides high bandwidth connections between research laboratories and academic institutions for data sharing and video/voice communication. The ESnet On-Demand Secure Circuits and Advance Reservation System (OSCARS) establishes guaranteed bandwidth of secure virtual circuits at a certain time, for a certain bandwidth and length of time. Though OSCARS operates within the ESnet, it also supplies end-to-end provisioning between multiple autonomous network domains. OSCARS gets reservation requests through a standard web service interface, and conducts a Quality-of-service (QoS) path for bandwidth guarantees. Multi-protocol Label Switching (MPLS) and the Resource Reservation Protocol (RSVP) enable to create a virtual circuit using Label Switched Paths (LSP's). It contains three main components: a reservation manager, a bandwidth scheduler, and a path setup subsystem [3,4]. The bandwidth scheduler needs to have information about the current and future states of the network topology in order to accomplish end-to-end bandwidth guaranteed paths.

The OSCARS bandwidth reservation system keeps track of changes in the network status and maintains a topology graph G which can simply be described as follows. Every port in a router has a maximum bandwidth available for reservation, and each network link connecting two ports (providing communication from one router towards another one) has an "engineering metric" related to the link latency. The web service interface enables users to allocate a fixed amount of bandwidth for a time period between two end-points in the network. A reservation request R contains: source and destination end-points, requested bandwidth, and the start/end times, $R=\{n_{source}, n_{destination}, M_{bandwidth}, t_{start}, t_{end}\}$. Since there might be bandwidth guaranteed paths in the system that are already fully or partially committed, the reservation engine needs to ensure availability of the requested bandwidth from source to destination for the requested time interval. In order to eliminate over commitment, committed reservations between t_{start} and t_{end} are examined, and a snapshot graph G' of the network topology is generated by extracting available bandwidth information for each port in the time period (t_{start}, t_{end}). $G'=G(t_{start}, t_{end})$ represents status of the network in advance. The shortest path on $G'=G(t_{start}, t_{end})$ from source to destination is calculated based on the engineering metric on each link, and a bandwidth guaranteed path is set up to commit and eventually complete the reservation request for the given time period.

On the other hand, if the requested reservation cannot be granted, no further suggestion is returned back to the user by OSCARS, except a failure message. In such a

situation, users have to go through a trial-and-error sequence, and may need to try several advance reservation requests until they get an available reservation. Further, there is no possibility from the user's view-point to be aware of the other possible options that might fit better into his/her requirements. In other words, users cannot make an optimal choice. Moreover, the current method of selecting a path may lead to ineffective use of the overall system such that network resources may not be used as optimally as possible. Therefore, our goal is to enhance the OSCARS reservation system by extending the underlying mechanism to provide a new service in which users submit their constraints and the system suggests possible reservation requests satisfying users' requirements.

In our algorithm, instead of giving all reservation details such as amount of bandwidth to allocate between start/end times, users provide maximum bandwidth they can use, total size of the data requested to be transferred, the earliest start time, and the latest completion time. Moreover, users can set criteria such that they would like to reserve a path for earliest completion time or reserve a path for shortest transfer duration. Such a request can be represented as: $R_s^{'}=\{n_{source}, n_{destination}, M_{MAXbandwidth}, D_{dataSize}, t_{EarliestStart}, t_{LatestEnd}\}$. The maximum bandwidth is related to the capability of the client and server hosts between source and destination end-points. Even if the network can provide a higher bandwidth than the maximum requested, the user may not be able use all the available bandwidth due to some other limitations and bottlenecks in the client and server sites. The reservation engine finds out the reservation $R=\{n_{source}, n_{destination}, M_{bandwidth}, t_{start}, t_{end}\}$ for the earliest completion or for the shortest duration where $M_{bandwidth} \leq M_{MAXbandwidth}$ and $t_{EarliestStart} \leq t_{start} < t_{end} \leq t_{LatestEnd}$.

The foremost question is how to find the maximum bandwidth available for allocation from a source node n_{source} to a destination node $n_{destination}$. The Max-bandwidth path algorithm is well known in quality-of-service (QoS) routing problems in which a path is constructed from source to destination given that each link is associated with an available bandwidth value. The bandwidth of a path is the minimum of all links over the path. Max-bandwidth path is a slightly modified version of Kruskal and Dijkstra's algorithms with the same asymmetrical time complexity [5]. However, we deal with a dynamic network such that the bandwidth value for every link is time dependent, $link=e(_{Router}A\text{-}_{port}1, \ _{Router}B\text{-}_{port}2)$ and $link_{bandwidth}(t_{time})$. Graph algorithms for time-dependent dynamic networks has been studied in the literature especially for max-flow and shortest path algorithms [6,7,8]. The most common approach is the discrete-time algorithms in which the time is modeled as a set of discrete

values and a static graph is constructed for every time interval. As an example, [9] uses time-expanded max flow for data transfer scheduling, [6] presents various shortest path algorithms for dynamic networks with time-dependent edge weights. The following example is given to clarify the dynamic max-bandwidth problem. Assume a vehicle wants to travel from city *A* to city *B* where there are multiple cities between *A* and *B* connected with separate highways. Each highway has a specific speed limit but we need to reduce our speed if there is high traffic load on the road, and we know the load on each highway for every time period. The first question is which path the vehicle should follow in order to reach city *B* as early as possible. Alternatively, we can delay our journey and start later if the total travel time would be reduced. Thus, the second question is to find the route along with the starting time for shortest travel duration. Time-dependent graph algorithms mainly focus on those two questions. However, we are dealing with bandwidth reservation where allocation should be set in advance when a request is received. If we apply this condition to the example problem described above, we have to set the speed limit before starting and cannot change that during the journey. Therefore, known algorithms do not fit into our problem domain. This distinguishes our path calculation from other time-dependent graph algorithms in the literature.

The outline of our approach is as follows. We divide the given search interval into several time windows, and keep snapshots of the network topology about the available bandwidth status for every link in each time window. This information is updated on-the-fly every time a reservation request is committed and stored for further processing during the path calculation phase. A time window represents a period of time in which we have a stable discrete status in terms of available bandwidth over the links. For example, if we have three committed reservations with allocated bandwidth for their time periods $r_1=\{b_1, t_1, t_3\}$, $r_2=\{b_2, t_2, t_5\}$, $r_3=\{b_3, t_2, t_4\}$, where the times t_1, t_2, t_3, t_4, t_5 are distinct values, there will be four time windows $tw_1=\{t_1, t_2\}$, $tw_2=\{t_2, t_3\}$, $tw_3=\{t_3, t_4\}$, $tw_4=\{t_4,t_5\}$ and four snapshots for the time windows G_{tw1}, G_{tw2}, G_{tw3}, G_{tw4}. If a link is associated with all three paths in these three reservations r_1, r_2, r_3, then, the available bandwidth over that link is equal to $bandwidth_{max} - (b_1+b_2+b_3)$ for the time period of (t_2, t_3), which is kept in G_{tw2}. The next step is to search through these time windows in a sequential order to check whether we can satisfy the requested allocation for that time window. For the given example above, first tw_1 (t_1, t_2), and tw_2 (t_2,t_3) will be examined; later, if both cannot satisfy the request, time window tw_{12}, a

combination of tw_1 and tw_2 (t_1, t_3), will be examined. This can easily be computed using G_{tw1} and G_{tw2} such that $G_{tw1\text{-}2}=\{b_{bandwidth}(link_i)=min(\ b_{bandwidht}(G_{tw1}(link_i),\ b_{bandwidth}(G_{tw2}(link_i)\ )\}$. For earliest completion time, the search pattern will be as follows: tw_1, tw_2, $tw_{1\text{-}2}$, tw_3, $tw_{2\text{-}3}$, $tw_{1\text{-}3}$, tw_4, $tw_{3\text{-}4}$, $tw_{2\text{-}4}$, *and* $tw_{1\text{-}4}$. The additive property of G_{tw} makes the process easy, since we only need to store one graph snapshot for each starting time window; for example, to obtain $Gt_{w1\text{-}4}$ we only need $G_{tw1\text{-}3}$ and G_{tw4}.

The complexity of the proposed algorithm is discussed next. Max bandwidth path algorithm is bounded by $O(N^2)$, where N is the number of routers. The number of time windows that need to be searched is bounded by number of committed reservations within the given period of $(t_{EarliestStart},\ t_{LatestEnd})$. In the worst-case, we may require to search all time window combinations, which is $T(T+1)/2$, where T is the number of time windows. If there are r committed reservations in that period, there can be maximum $2r+1$ different time windows in worst-case. Overall, worst-case complexity is bounded by $O(r^2N^2)$. However, r is relatively very small compared to the number of nodes N, in the topology. Bandwidth reservation is used for large-scale data transfers and it is very unlikely to have thousands of committed reservations in a given time period. Also, path calculation from two end-points does not span to all nodes in a real network; therefore, we can trim G and perform calculation on a reduced data set, $G(n_{source},\ n_{destionation})$. Moreover, time windows that are too short in duration to transmit the requested amount of data can be eliminated beforehand. Max bandwidth and shortest path algorithms are quite efficient and the search process over time windows is scalable and practical, considering that the number of reservations in practice is limited. We have tested the performance of the algorithm by simulating very large graphs (with *10K* nodes) and we have observed that computation time is in the order of seconds.

Network provisioning is not sufficient by itself for end-to-end high performance data transfer. In order to take advantage of the available bandwidth, client sites should have storage allocation. For that reason, network provisioning services need coordination between storage resource managers, such as SRM [10] that dynamically reserve and manage storage on demand. According to the storage allocation policy and available storage space in client sites, data files might be split into multiple chunks to be transferred in shortest transfer periods. Our future work includes coordination of storage and network resource allocations.

Acknowledgements

We would like to thank David Robertson and Mary Thompson from ESnet for their generous help in OSCARS client interface during the development and testing of the Network Reservation Engine.

This work was funded by the Office of Advanced Scientific Computing Research, Office of Science, U.S. Department of Energy, under contract no. DE-AC02-05CH11231.

REFERENCES

[1] Internet2: http://www.internet2.edu/

[2] Energy Sciences Network: http://www.es.net/

[3] ESnet On-demand Secure Circuits and Advance Reservation System (OSCARS): http://www.es.net/oscars/

[4] Chin P. Guok, David Robertson, Mary Thompson, Jason Lee, Brian Tierney, William Johnston, "Intra and Interdomain Circuit Provisioning Using the OSCARS Reservation System", Third International Conference on Broadband Communications, Networks, and Systems, IEEE/ICST, 2006

[5] Malpani, N. and Chen, J. A note on practical construction of maximum bandwidth paths. Inf. Process. Lett. 83, 3, 2002

[6] Ariel Orda and Raphael Rom, Shortest-Path And Minimum-Delay Algorithms In Networks With Time-Dependent Edge-Length, Journal of the ACM, vol 37, pg 607-625, 1990

[7] Ding, B., Yu, J. X., and Qin, L. “Finding time-dependent shortest paths over large graphs”. In Proceedings of the 11th international Conference on Extending Database Technology: Advances in Database Technology , vol. 261. ACM, 2008

[8] Ismail Chabini, Discrete Dynamic Shortest Path Problems In Transportation Applications: Complexity And Algorithms With Optimal Run Time, Transportation Research Records, vol 1645, pg 170--175, 1998

[9] William C. Cheng, Cheng-fu Chou, Leana Golubchik, Samir Khuller, Yung-Chun Wan, "Large-scale Data Collection: a Coordinated Approach", in Proceedings of IEEE INFOCOM, pg 218-228, 2003

[10] ArieShoshani, Alexander Sim, JunminGu “Storage Resource Managers: Essential Components for the Grid”,. Grid Resource Management: State of the Art and Future Trends, Edited by JarekNabrzyski, Jennifer M. Schopf, Jan Weglarz, Kluwer Academic Publishers, 2003