Virtual Overlay Network Scheduling

Feng Gu, Chongyang Xie, Min Peng, Çiçek Çavdar, Samee Khan, and Nasir Ghani

***Abstract*—As applications continue to expand, there is a growing need to provide advance reservation scheduling for higher-end virtual overlay network services. Hence this work presents an optimization formulation for this problem and studies several heuristic solutions using network simulation.**

***Index Terms*—Advance reservation, network scheduling, topology overlay, virtual network services**

I. Introduction

THE need to schedule future user demands, i.e., *advance reservation* (AR), is becoming an integral requirement for a range of applications in grid-computing, e-science, storage backup, community networking, special event broadcasting, etc. Hence researchers have proposed a range of connection scheduling schemes for various AR service models (fixed start/duration, variable start/duration, etc) in bandwidth-provisioning and optical wavelength-routing networks, see [1]-[3]. Others have also looked at broader AR rerouting [4], survivability [5] and distributed implementation [6] strategies.

However, with expanding user application scenarios, there is a growing need to extend AR schemes to support *multipoint* connectivity between several sites. Now many studies in *virtual private network* (VPN) design and overlay network provisioning [7] have looked at interconnection strategies over physical networks using optimization and heuristic methods. Notable examples here include the work in [8],[9] as well as dynamic solutions in [10], [11]. Earlier efforts have also studied *virtual topology* (VT) design for optical networks, see [12]. Nevertheless, the above efforts have only treated immediate requests that arrive in an "on-demand" or a-priori manner.

In light of the above, there is a critical need to develop scheduling solutions for multipoint network overlay services, i.e., termed here as *virtual overlay network scheduling* (VONS). Indeed, there are no known studies in this area. Along these lines, this paper is organized as follows. An idealized *integer linear programming* (ILP) model for the VONS problem is first presented in Section II along with a heuristic solution in Section III using graph-theoretic schemes. Section IV then presents detailed simulation results, and conclusions and future work directions are presented in Section V.

Manuscript received April 16, 2011. The associate editor coordinating the review of this letter and approving it for publication was J. Wang.

F. Gu, C. Xie, and N. Ghani are with the University of New Mexico (e-mail: nghani@ece.unm.edu).

M. Peng is with Wuhan University.

Ç. Çavdar is with KTH Royal Institute of Technology.

S. Khan is with North Dakota State University.

Digital Object Identifier 10.1109/LCOMM.2011.11.110819

II. Optimization Formulation

An ILP model of the VONS problem is presented. The formulation assumes idealized settings where all overlay requests are known a-priori. Consider the requisite notation. The network is modeled as a graph, $G(V, E)$, where V is the set of router/switch nodes and E the set of physical links. Without loss of generality, all links have capacity, C, and connectivity is bidirectional, i.e., two uni-directional links between adjacent nodes. Each physical link $e \in E$ also has a bandwidth-time function, i.e., $c_e(t)$, which tracks the used capacity at future time. In addition, the n-th virtual overlay network request is denoted by the 5-tuple $r^n = (S^n, L^n, t_s^n, t_e^n, b^n)$, where S^n is the set of node sites ($S^n \subseteq V$), L^n is the set of virtual links between nodes in S^n, t_s^n is the start time, t_e^n is the stop time, and b^n is the requested bandwidth, $b^n \leq C$. Normally, request setup entails scheduling the set of connections corresponding to the virtual links in L^n, i.e., connection endpoints designated by virtual link endpoints. As per ILP requirements, time is discretized into fixed timeslots, i.e., t_s^n and t_e^n are integral multiples of a fixed timeslot interval T. In addition, some other variables are also defined:

- R is the set of all VONS requests
- $T_m = \max_{r^n \in R} \{t_e^n\}$ is the maximum stop timeslot across all requests $r^n \in R$
- $v_i^n \in S^n$ is the i-th node selected in S^n
- $L_{i,j}^n$ is the virtual link between v_i^n and v_j^n,$v_i^n \neq v_j^n$
- $p_{i,j}^{n,e,k}$ is a binary flag which denotes virtual link occupation in time slot k, i.e., $p_{i,j}^{n,e,k} = 0$ if $L_{i,j}^n$ does not use link $e \in E$ at time slot k; $p_{i,j}^{n,e,k} = 1$ if $L_{i,j}^n$ uses link $e \in E$ at time slot k
- $v \rightarrow e$ if $v \in V$ is the egress node of link $e \in E$; $e \rightarrow v$ if $v \in V$ is the ingress node of link $e \in E$

Using the above, the objective function is defined as:

$$minimize \sum_{r^n \in R} \sum_{v_i^n \in S^n} \sum_{v_j^n \in S^n} \sum_{e \in E} \sum_{0 \leq k \leq T_m} b^n p_{i,j}^{n,e,k} \quad \text{(Eq.1)}$$

subject to the following constraints:

$$\sum_{v_i^n \rightarrow e} p_{i,j}^{n,e,k} = 1 \qquad r^n \in R, t_s^n \leq k \leq t_e^n, v_i^n \in S^n, v_j^n \in S^n \quad \text{(Eq.2)}$$

$$\sum_{e \rightarrow v_i^n} p_{i,j}^{n,e,k} = 0 \qquad r^n \in R, t_s^n \leq k \leq t_e^n, v_i^n \in S^n, v_j^n \in S^n \quad \text{(Eq.3)}$$

$$\sum_{e \rightarrow v_j^n} p_{i,j}^{n,e,k} = 1 \qquad r^n \in R, t_s^n \leq k \leq t_e^n, v_i^n \in S^n, v_j^n \in S^n \quad \text{(Eq.4)}$$

$$\sum_{v_j^n \to e} p_{i,j}^{n,e,k} = 0 \qquad r^n \in R, t_s^n \leq k \leq t_e^n, v_i^n \in S^n, v_j^n \in S^n \tag{Eq.5}$$

$$\sum_{e \to v} p_{i,j}^{n,e,k} = \sum_{v \to e} p_{i,j}^{n,e,k} \qquad r^n \in R, t_s^n \leq k \leq t_e^n, v \notin \{v_i^n, v_j^n\}, v_i^n \in S^n, v_j^n \in S^n \tag{Eq.6}$$

$$\sum_{r^n \in R} \sum_{v_i^n \in S^n} \sum_{v_j^n \in S^n} b^n p_{i,j}^{n,e,k} \leq C \qquad e \in E, 0 \leq k \leq T_m \tag{Eq.7}$$

$$p_{i,j}^{n,e,k} = p_{i,j}^{n,e,k+1} \qquad r^n \in R, e \in E, t_s^n \leq k < t_e^n, v_i^n \in S^n, v_j^n \in S^n \tag{Eq.8}$$

Overall, Eq. 1 tries to minimize the resource utilization across all VONS requests, i.e., formulated as a sum of path length-bandwidth products. Meanwhile the constraints handle flow conservation (source nodes in Eqs. 2/3, destination nodes in Eqs. 4/5, and transit nodes in Eq. 6), link capacity limitation (Eq. 7), and route consistency (Eq. 8).

Carefully note that the scheduling problem for regular point-to-point connections is NP-complete [1]. Hence the VONS optimization problem is also of similar complexity as it is polynomially more complex. Overall, the computational complexity of the above ILP formulation is very high. For example, consider a 10 node mesh topology fielding just 5 scheduled overlay network requests with maximum durations of 15 timeslots and approximately 3 nodes per overlay topology. Here the total number of variables is upper-bounded by $3 \times 10 \times 10 \times 15 \times 10 \times 10 = 450,000$. Indeed, this is a very significant number, making the problem very difficult to solve on most common server systems. As a result a heuristic scheduling solution is now presented.

III. Heuristic Solution

A graph-theoretic VONS heuristic is now presented for regular bandwidth-provisioning networks (with considerations for optical wavelength networks left for future study). The solution assumes random "on-demand" requests, i.e., reflective of real-world settings as opposed to idealized a-priori settings. Furthermore, arrivals can now occur in continuous time, obviating the time-slotting constraint on t_s^n and t_e^n. The pseudocode for the algorithm is shown in Fig. 1 and basically loops through all virtual links in the request and routes each using a given *traffic engineering* (TE) strategy. The computations are done over a temporary copy of the network graph, $G'(V,E)$, derived from $G(V,E)$ by removing non-feasible physical links, i.e., without sufficient capacity in the request interval, $c_e(t) < b^n$ in $[t_s^n, t_e^n]$. Furthermore, three different TE routing approaches are considered here:

- **Minimum hop count** This scheme assigns unity link weights and finds the shortest route in $G'(V,E)$ via Dijkstra's algorithm. This approach pursues resource minimization and can yield saturation on specific links.
- **Maximum bottleneck** This scheme chooses the end-to-end route with the maximum "free" capacity. Specifically, the *k-shortest paths* (k-SP) between the source/destination nodes are computed and the one with the largest bottleneck link capacity (in request interval) is selected. This "load-balancing" approach may yield longer paths.
- **Minimum distance** This scheme achieves a balance between the above two strategies by using dynamic link weights that are inversely-proportional to the lowest residual capacity in the request interval:

$$\omega_{ij} = C/(\chi + \epsilon) \tag{Eq.9}$$

where

$$\chi = \min_{t \in [t_s^n, t_e^n]} c_e(t) \tag{Eq.10}$$

and ϵ is a small value chosen to avoid division errors.

Given incoming request $r^n = (S^n, L^n, t_s^n, t_e^n, b^n)$, generate temporary graph copy $G'(V,E) = G(V,E)$
Remove non-feasible links in $G'(V,E)$, i.e., $c_e(t) < b^n$ in $[t_s^n, t_e^n]$;
/ Loop and provision all virtual links in request */*
for $i = 1$ to $|L^n|$ **do**
  if minimum hop count **then**
    Assign unity weight to links in $G'(V,E)$
    Run Dijkstra's shortest-path for i-th virtual link
  else if minimum distance **then**
    Assign dynamic weight to links in $G'(V,E)$, Eq. (9)
    Run Dijkstra's shortest-path for i-th virtual link
  else
    Run k-SP between end-points of i-th virtual link
    Select path with maximum bottleneck bandwidth
  end if
end for
if all virtual link connections in L^n routed **then**
  Setup successful, copy $G'(V,E) \to G(V,E)$
end if

Fig. 1. VONS heuristic algorithm.

Overall, the algorithm in Fig. 1 returns success if all overlay link connections are routed, and in this case updates the network graph as well, i.e., $G'(V,E) \to G(V,E)$.

IV. Performance Analysis

The performance of the VONS heuristic is analyzed via simulation using *OPNET* $Modeler^{TM}$. Two topologies are tested here, including the well-known NSFNET backbone with 16 nodes/25 links (3.12 node degree) and the denser *Deutsche Telekom* (DT) network with 27 nodes/52 links (3.85 node degree). All the nodes are generic IP *multi-protocol label switching* (MPLS) routers with 10 Gbps links. Furthermore, overlay requests have exponentially-distributed holding and inter-arrival times with means μ and λ, respectively. Namely, a scaled mean holding time of $\mu = 600$ sec is used and the mean inter-arrival times are adjusted as per load. Note that this average holding time is just a relative figure and does not necessarily reflect real-world values.

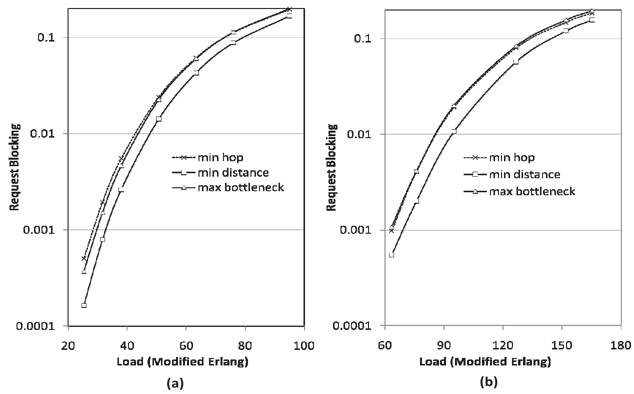

Fig. 2. Overlay request blocking: a) NSFNET, b) DT.

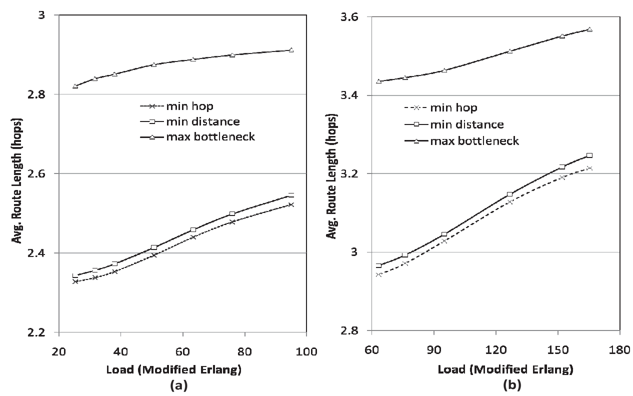

Fig. 3. Average path lengths: a) NSFNET, b) DT.

Meanwhile the virtual overlay topology sizes are uniformly varied from 4-6 nodes with corresponding (virtual link) bandwidth requests ranging uniformly from 200 Mb/s to 1.0 Gb/s, i.e., in 200 Mbps increments to model fractional Ethernet demands. Here the node/sites in the overlay topologies are randomly selected from the network nodes. Furthermore, the virtual links are selected using the random Inet topology generator [13] to achieve a node degree of about 2.5. Finally, each run is averaged over 500,000 requests, and a modified Erlang load metric is defined to account for overlay network size:

$$\text{Modified Erlang load} = \sum_{n=4}^{6}(n-1)\times\mu/\lambda \qquad \text{(Eq.11)}$$

where the virtual overlay topology sizes range from $n = 4$ to 6 nodes and the $1/\lambda$ represents the mean inter-arrival rate.

Initial tests are done to measure request blocking rates for the various TE (virtual link) routing strategies, Fig. 2. These results show that the minimum distance heuristic (Eq. 1) consistently gives the lowest blocking for both topologies. In particular this scheme yields less than half the blocking rate of the maximum bottleneck scheme at low-medium input loads (log scale). By contrast the maximum bottleneck scheme only yields marginal (negligible) blocking reduction over the minimum hop count scheme, despite its sizable increase in computational complexity, i.e., k-SP computations versus Dijkstra's shortest path.

Next, the average path lengths of the computed routes for the virtual links (in the overlay topologies) are plotted in Fig. 3. As expected, the minimum hop count routing scheme gives the lowest utilization of all. Nevertheless, the minimum distance heuristic provides very competitive performance here, with average path length values coming within 2% of the minimum hop count scheme. Conversely, the hop count utilization with the maximum bottleneck scheme is notably higher, by about 15% for both topologies.

Carefully note that the VONS heuristic in Fig. 1 tries to setup all virtual link connections in the order in which they are listed in the request, i.e., random. Now modified versions of the heuristic have also been tested to better "shuffle" the sequence in which the virtual links are attempted, e.g., longest-path first, etc. However, these renditions show no noticeable reduction in blocking over the base random ordering scheme. Overall, these findings indicate that the minimum distance scheme gives the best performance tradeoff in terms of minimizing blocking and also lowering resource consumption on virtual link routes.

V. CONCLUSIONS

There is a growing need to develop advance reservation scheduling algorithms for virtual overlay network services. This paper addresses this concern and proposes an optimization model along with a graph-based heuristic solution (for resource minimization and load balancing). Simulations show that dynamic load-based link weighting schemes give the best performance/tradeoffs in terms of blocking reduction and resource utilization. Future efforts will look at incorporating survivability capabilities for overlay scheduling.

ACKNOWLEDGMENT

This work has been funded by US Department of Energy Office of Science. The authors are grateful for this support.

REFERENCES

[1] R. Guerin and A. Orda, "Networks with advance reservations: the routing perspective," *IEEE INFOCOM*, Mar. 2000.
[2] J. Zheng and H. Mouftah, "Routing and wavelength assignment for advance reservation in wavelength-routed WDM optical networks," *IEEE ICC*, May 2002.
[3] E. Escalona, *et al.*, "Advance reservations for service-aware GMPLS-based optical networks," *Computer Networks*, vol. 52, no. 10, pp. 1938-1950, July 2008.
[4] C. Xie, *et al.*, "Load-balancing connection rerouting in advance reservation networks," *IEEE Commun. Lett.*, vol. 14, no. 6, pp. 578-580, June 2010.
[5] T. Li, *et al.*, "On survivable service provisioning in WDM optical networks under a scheduled traffic model," *IEEE GlOBECOM*, Dec. 2005.
[6] C. Xie, *et al.*, "Routing and scheduling in distributed advance reservation networks," *IEEE GLOBECOM*, Dec. 2010.
[7] N. Chowdhury and R. Boutaba, "Network virtualization: state of the art and research challenges," *IEEE Commun. Mag.*, vol. 47, no. 7, pp. 20-26, July 2009.
[8] D. Andersen, *et al.*, "Resilient overlay networks," *ACM SOSP*, Oct. 2001, pp. 131-185.
[9] Z. Duan, *et al.*, "Service overlay networks: SLAs, QoS, and bandwidth provisioning," *IEEE/ACM Trans. Netw.*, vol. 11, no. 6, pp. 870-883, Dec. 2003.
[10] Y. Zhu and M. Ammar, "Algorithms for assigning substrate network resources to virtual network components," *IEEE INFOCOM*, Apr. 2006.
[11] C. Xie, *et al.*, "Multi-point Ethernet over next-generation SONET/SDH," *IEEE ICC*, June 2009.
[12] R. Dutta and G. Rouskas, "A survey of virtual topology design algorithms for wavelength routed optical networks," *Optical Networks*, vol. 1, no. 1, pp. 73-89, Jan. 2000.
[13] Inet topology generator, see http://topology.eecs.umich.edu/inet/.