Optimization of Minimum Cost Network Flows with Heuristic Algorithms

K. Kumar, *Member, IACSIT*

***Abstract*—In today's scenario we see that scaling of heterogeneous systems is exponentially increasing. Network operators design their backbone networks to accommodate all traffic efficiently (e.g., without congestion or large delays). However, even if a backbone network suitable for actual traffic is constructed, traffic could significantly differ from the initial traffic as time goes on. As a result, the previously constructed backbone network becomes no longer suitable to the current traffic; it may happen that utilizations of some links are extremely high while utilizations of other links are extremely low. Especially, because high link utilizations cause congestion or large delays, we need to avoid high link utilizations even when traffic fluctuates. A distributed network in composed of a number of autonomous processors . Dividing up resources among many host computers on a network thereby reducing the burden on any one system. Communicating with any one node in this type of network implies that you will be communicating indirectly with all the other nodes. a distributed network can be anonymous, in the sense that you don't really need to know where some data came from or where its destination is. The network distance between the client and the server we can deploy algorithms to enhance the speed with can retrieve information.**

***Index Terms*—Heuristic, optimization, infrastructure, implementation, optimization.**

I. Introduction

To construct the Topology, we use a distance matrix. The number of virtual tracers. Defines the number of host clusters. The virtual topology model described in 2 required the distances among all the hosts. To come up with a scalable algorithm we use a sampling based approach The distance between the virtual traces is completed by the average for all pairs For clustering the hosts, multiple land marks are placed around the internet When a new host joins the system, the host measures the distance to be landmarks and considers the closet landmark as the cluster id . Each host measures the distance to a small number hosts in the same cluster and a small number of hosts in the other cluster. The topology in this system is a simple mesh star topology, where virtual tracers from a full mesh and hosts from the star topology with the virtual tracer in the center. With the virtualization concept, the logical object must advertise the necessary information for end users to search and select the needed resources that fulfill their requests. It is easy for an end user to deal with source, destination, bandwidth, and lease duration information rather than having direct access to the switches to select the slots, ports, STS channel number, framing type, etc. We plan to evaluate this scheme further and investigate the feasibility for using arbitrary topologies.

Manuscript received December 21, 2011; revised February 10, 2012.

K. Kumar is with the MCA Dept, Veltech Dr. RR and Dr. SR Technical University, India (e-mail: Hellokumar76@yahoo.co.in).

II. Problem Statement

To work on the best algorithm by comparing the types available and if any thing new add on to it. This work is to overcome the problem of heavy congestion by setting up a virtual route and find the method to reach destination with network destination estimation factor.

III. Heuristic Traffic Routing Algorithm

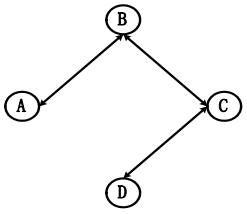

Fig. 1(a). Physical topology

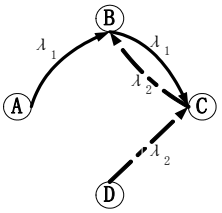

Fig. 1(b). Virtual topology

The sub problem of traffic routing is to be solved here. Traffic route between the source and destination node pairs and their corresponding amount of traffic are determined here. The proposed algorithm tries to find the shortest paths for traffic node pairs by *Greedy Algorithm* upon the constructed virtual topology. Since as explained in [1] holds good for latency optimization where in data transfer takes place between nodes then the best result can be found between pairs to find the shortest path. An optimization problem is one in which you want to find, not just *a* solution, but the *best* solution.

A "greedy algorithm" sometimes works well for optimization problems. A greedy algorithm works in phases. At each phase: You take the best you can get right now, without regard for future consequences. You hope that by choosing a *local* optimum at each step, you will end up at a *global* optimum. It also tries to assign traffic as much as possible to the path with hops as less as possible without exceeding the optical channel capacity. [2]-[4] clearly gives the idea that network performance on host gives an

optimized result may be obtained provided heuristics are deployed with respect to object location.

The algorithm is described as follows:

Step 1: Sort all traffic node pairs by their amount of traffic in descending order. Initialize the allowed amount of traffic of every lightpath as C, which is the maximal capacity of the optical channel.

Step 2: If there are no traffic node pairs that have not been routed, the algorithm stops; otherwise, choose the node pair with the largest amount of traffic, and then calculate the shortest path P_{sd} from s to d.

Step 3: Calculate the MAT (Maximum Allotable Traffic) of P_{sd}, and then add traffic $\lambda = \min(C - MAT, \lambda_{sd})$ to all optical channels that P_{sd} covers. ($MAT = \min(\lambda_{ij})$, λ_{ij} is the allotable capacity of C_{ij}, and C_{ij} is the optical channel that P_{sd} covers.)

Step 4: if $\lambda_{sd} - \lambda > 0$, calculate the next shortest path P_{sd} from s to d, and then go to step 3; otherwise, mark (s,d) routed, choose the next node pair (s,d), and then go to step 2.

IV. Implementation

The First Methods to Get Acquainted With:

The two most important methods for developers to get familiar with are as follows:

- in sr_router.c

```
void sr_handlepacket(struct sr_instance* sr,
    uint8_t * packet/* lent */,
    unsigned int len,
    char* interface/* lent */)
```

This method is called by the router each time a packet is received. The "packet" argument points to the packet buffer which contains the full packet including the ethernet header. The name of the receiving interface is passed into the method as well. One way to access the fields in this packet is to "typecast" the pointer to the appropriate structure: e_hdr = (struct sr_ethernet_hdr*) packet;

The order of the bits in packets on the network may be different from this order on the host. You need to use functions like:

htons : host to network short

ntohs: network to host short

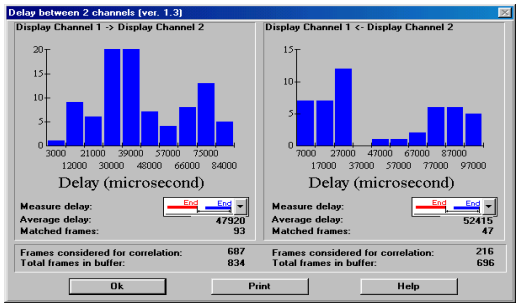

Fig. 2. Study of delay during data transmission

Fig. 2 shows the study of delay of data transmission between nodes in a network when implemented. This implementation helps to find the cost between flows and generate the average delay. The above study was done since [8] gives the details for scalable infrastructure and hence the delay was studied between host to get the optimized result. We need to find the round trip time(RTT) also from host to network and network to host to see the minimum cost network flow. Taking into consideration to get the best quality of service during network flow with less error detection and flow control the above implementation shows that delay is necessary to get the optimized result for a particular network packer transfer between host to network and network to host respectively.

V. Optimization: Minimum Cost Network Flows

The above methodology when implemented was found not to give the best optimized Solution. We then worked with the optimization technique using the minimum cost network flow model. This model represents the broadest class of problem that can be solved much faster than linear programming while still retaining such nice properties as integrality of solution and appeal of concept.

Like the maximum flow problem, it considers flows in networks with capacities. Like the shortest path problem, it considers a cost for flow through an arc. Like the transportation problem, it allows multiple sources and destinations. In fact, all of these problems can be seen as *special cases* of the minimum cost flow problem.

Consider a directed network with n nodes. The decision variables are x_{ij}, the flow through arc (i,j). The given information includes:

c_{ij} : cost per unit of flow from i to j (may be negative), u_{ij} : capacity (or upper bound) on flow from i to j, b_i : net flow generated at i.

This last value has a sign convention:

$b_i > 0$, if i is the supply node
$b_i < 0$ if i is the demand node
$b_i = 0$ if i is the transhipment node

The objective is to minimize the total cost of sending the supply through the network to satisfy the demand.

Note that for this model, it is not necessary that every arc exists. We will use the convention that summations are only taken over arcs that exist. The linear programming formulation for this problem is:

Minimize $\sum_i \sum_j c_{ij} x_{ij}$
Subject to $\sum_j x_{ij} - \sum_j x_{ji}$ for all nodes i,
$0 \le x_{ij} \le u_{ij}$ for all arcs (i,j)

Again, we will assume that the network is balanced, so $\sum_i b_{i=0}$, since dummies can be added as needed. We also still have a nice integrality property. If all the b_i and u_{ij} are integral, then the resulting solution to the linear program is also integral.

Minimum cost network flows are solved by a variation of the simplex algorithm and can be solved more than 100 times faster than equivalently sized linear programs. From a modeling point of view, it is most important to know the

sort of things that can and cannot be modeled in a single network flow problem. When fastest shortest path is possible as given in [11] it is very much possible to get the above optimized result provided certain conditions which can and cannot be executed are taken for consideration to get an optimized solution with minimum cost and deploying the algorithms explained above.

Can do

Lower bounds on arcs. If a variable x_{ij} has a lower bound of l_{ij} , upper bound of u_{ij} , and cost of c_{ij} , change the problem as follows:

- Replace the upper bound with , $u_{ij} - l_{ij}$
- Replace the supply at i with $b_i - l_{ij}$,
- Replace the supply at j with $b_i - l_{ij}$,

Now you have a minimum cost flow problem. Add $c_{ij}l_{ij}$ to the objective after solving and l_{ij} to the flow on arc (i, j) to obtain a solution of the original problem.

Upper bounds on flow through a node. Replace the node i with nodes i' and i''. Create an arc from i' to i'' with the appropriate capacity, and cost 0. Replace every arc (j,i) with one from j to i' and every arc (i,j) with one from i'' to j. Lower bounds can also be handled this way.

Convex, piecewise linear costs on arc flows (for minimization). This is handled by introducing multiple arcs between the nodes, one for each portion of the piecewise linear function. The convexity will assure that costs are handled correctly in an optimal solution.

Can't do

- Fixed cost to use a node.
- Fixed cost to use an arc.
- "Proportionality of flow" That is, if one unit enters node i, then you insist that. 5 units go to node j and .5 to node k.

Gains and losses of flow along arcs, as in power distribution.

Note that although these cannot be done in a single network, it may be possible to use the solutions to multiple networks to give you an answer. For instance, if there is only one arc with a fixed cost, you can solve both with and without the arc to see if it is advantageous to pay the fixed cost.

VI. Conclusion

The above algorithms will help in minimizing average weighted number of hops. The congestion in networks will be reduced. The algorithms implemented will minimize the message delay. By implementing the virtual topology concept, end users/applications can self-provision and confi gure the networks by themselves and the entire technical configuration will be completely transparent for them. Virtual services are becoming more frequent, especially in research communities (e.g., bandwidth-intensive e-science applications).

VII. Future Work

To achieve optimality, decision-making processes in a pervasive grid must take into account several aspects. The infrastructure topology and condition such as communication, computation and storage capabilities should be studied. The characteristics of the task for network distance estimation will study the task: file, service. The objectives to be achieved: End-user response delay, load balancing, limitation of financial costs.

References

[1] K. P. Gummadi, S. Saroiu, and S. D. Gribble, "King: Estimating latency between arbitrary internet end hosts," in *Proc. SIGCOMM Internet Measure. Workshop*, Marseille, France, Nov. 2002.

[2] K. Lakshminarayanan and V. Padmanabhan, "Some findings on the network performance of broadband hosts," in *Proc. Internet Measure. Conf.*, Oct. 2003.

[3] A. Slivkins, "Embedding, Distance Estimation and Object Location in Networks", P.H.D thesis in Cornell, 2006

[4] S. Banerjee, Z. Xu, S.-J. Lee, and C. Tang, "Service adaptive multicast for media distribution networks," in *Proceedings of the IEEE WIAPP 2003*, San Jose, CA, June 2003, pp. 50–60.

[5] A. Levin and Y. Shavitt. "Approximation and Heuristic Algorithms for Minimum-Delay Application Layer Multicast Trees,"' *IEEE/ACM Transactions on Networking*, 15(2):473--484, April 2007.

[6] R. Cox, F. Dabek, F. Kaashoek, J. Li, and R. Morris, "Practical, distributed network coordinates," in *Proceedings of the ACM HotNets-II*, 2003.

[7] P. Francis, S. Jamin, C. Jin, Y. Jin, D. Raz, Y. Shavitt, and L. Zhang, "IDMaps: A global internet host distance estimation service," *IEEE/ACM Trans. Networking*, vol. 9, no. 5, pp. 525–540, October 2001.

[8] S. Srinivasan and E. Zegura, "M-coop: A scalable infrastructure for network measurement," in *Proceedings of the IEEE WIAPP 2003*.

[9] B. Zhang, S. Jamin, and L. Zhang, "Host multicast: A framework for delivering multicast to end users," in *Proceedings of the IEEE INFOCOM 2002*, New York, NY, June 2002. 27

[10] T. S. E. Ng and H. Zhang. Predicting internet network distance with coordinates-based approaches. In *Int. Conf. on Computer Communications (INFOCOM'01)*, pages 170{179, 2001.

[11] M. Potamias, F. Bonchi, C. Castillo, and A. Gionis. Fast shortest path distance estimation in large networks. In *Proc. 2009 Int. Conf. Information and Knowledge Management (CIKM'09)*, pages 867{876, 2009.

[12] R. Govindan and H. Tangmunarunkit, "Heuristics for Internet Map Discovery," in *Proceedings of IEEE INFOCOM '00*, Tel Aviv, Israel, March 2000.

K. Kumar was born in Chennai on 16/02/1976 and had graduated with degree in mathematics from University of Madras, Chennai, Tamil Nadu , India in 1997. Then he went on to pursue his Masters degree in Computer Applications(M.C.A) from the same University in the year 2000. In the year 2004 he completed his Master of Philosophy from Periyar University Salem, Tamilnadu, India. He registered for his Ph.D in University of Madras in 2007 July. His major field of study in computer networks.

He has over 11 years experience in teaching and is currently working as Assistant Professor in Department of Computer Applications at Veltech Dr.RR and Dr.SR Technical University in Chennai , Tamil Nadu, India. He has presented 12 papers in national and international conference and published his researched papers. His other areas of research interest includes expert system and data mining.

He has membership in various professional society bodies like ISTE, ACEEE, CSTA and is a senior member of Computer Society of India(CSI). He has won the best paper award in a national conference in 2007 in Tamil Nadu, India