2011 International Conference on Advancements in Information Technology
With workshop of ICBMG 2011
IPCSIT vol.20 (2011) © (2011) IACSIT Press, Singapore

Optimization of Network Distance Estimation with Heuristic Algorithms

K.kumar[+]

Asst.Professor, MCA Dept, Veltech Dr. RR and Dr.SR Technical University

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 .

Keywords: heuristic, optimization, infrastructure, implementation, optimization

1. Introduction

To construct the Topology, we use a distance matrix. The number of virtual tracers. Defines the number of host clusters. 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.

2. Problem Statement

To work on the best algorithm by comparing the types available and if anything 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.

3. Heuristic Traffic Routing Algorithm

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. 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.

3.1. The Algorithm is Described as Follows:

[+] Corresponding author. Tel.: +(0)98404 50869.
E-mail address: hellokumar76@yahoo.co.in.

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.

4. Implementation

4.1. 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 and ntohs -- network to host short.

5. Optimization: Minimum Cost Network Flows

The above methodology when implemented was found not to give the best optimized Solution. 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 a supply node, $b_i < 0$if *i* is a demand node,
- $b_i = 0$if *i* is a transshipment 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:

$$\begin{aligned} &Minimize \sum_i \sum_j c_{ij} x_{ij} \\ &Subject\ to\ \sum_j x_{ij} - \sum_j x_{ji} = b_i \ \ for\ all\ nodes\ i, \\ &\qquad 0 \le x_{ij} \le u_{ij} \ \ for\ all\ arcs(i,j) \end{aligned}$$

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.

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.

6. 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 configure the networks by themselves and the entire technical configuration will be completely transparent for them.

7. 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.

8. References

[1] 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.

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

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

[4] 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.

[5] K. P. Gummadi, S. Saroiu, and S. D. Gribble, "King: Estimating latency between arbitrary internet end hosts," in *Proceedings of the ACM IMW'2002*.

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

[7] 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.

[8] 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.

[9] 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.

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