Distributed Route Control Schemes to Load Balance Incoming Traffic in Multihomed Stub Networks

Ashok Singh Sairam
ashok@iitp.ac.in
Dept. of Computer Science and Engineering
Indian Institute of Technology Patna

Gautam Barua
gb@iitg.ernet.in
Dept. of Computer Science and Engineering
Indian Institute of Technology Guwahati

***Abstract*—Load balancing traffic of a multihomed stub network among its various egress links can significantly improve performance by allowing the network to route its traffic around congested paths. In [3], several centralized route control schemes to load balance traffic were proposed. It was shown that employing these route control schemes resulted in significant improvement in round-trip times. We carry this exploration further by developing distributed algorithms to load balance traffic. We show theoretically that the expected performance of the distributed version is the same as the centralized routing scheme. We further validate the schemes using both synthetic as well as actual traffic traces.**

I. Introduction

Although many of the applications used on the Internet generate bi-directional flow of data ([7]), volume of data is heavier in one direction than the other. Due to the asymmetric nature of Internet traffic, network operators try to optimize either the traffic that enter or leave the network based on its business interest. Access providers that serve small and medium enterprises will have users that primarily want to download traffic from the Internet. In access networks the main objective is to optimize the way traffic enter the networks. Centralized routing techniques to distribute the incoming traffic of a multihomed stub network among its various egress links were proposed in [3]. Centralized routing protocols have the advantage that we need to advertise less information, administration is easy and the algorithms are simple. However, the manager will become a bottleneck as the size of the network increases. Moreover, centralized routing approaches are also not consistent with the design philosophy of the Internet. Further, due to geographical distances, collecting all the required information at a central router may be costly. In this paper we present a distributed routing technique to load balance the incoming traffic of a multihomed stub network among its various egress links. In figure 1 we show a network that is multihomed to two Internet Service Provides (ISPs). Egress (or border) routers connect the network to an ISP/peer. The end users or nodes connect through the access routers. In the distributed scheme, nodes independently make routing decisions on the basis of network information received by them. However, for the nodes to make routing decisions individually, it requires that the current network status be disseminated to all the nodes. In practice there are many standard Internet algorithms like QoS routing, call admissions ([4]) et. al., where network information needs to be broadcast to all nodes of the network. In response to the requirement for current network status by a large number of traffic engineering architectures, the Internet Engineering Task Force (IETF) proposed a framework to distribute link state information within a domain, OSPF-TE ([2]). This protocol is an extension to the OSPF protocol which allows distribution of additional link state information along with the link state updates - available bandwidth, propagation delay and hop count. In this work we assume that each border router maintains an up-to-date database of the network topology which includes available bandwidth and utilization of the egress routes. These two metrics are presumed to be included in the link state advertisements (LSAs) broadcast by the border routers.

The organization of the paper is as follows. In section II we describe the problem. The distributed routing techniques are presented in section III. In section IV we present the experimental results and finally the conclusion is presented in section V

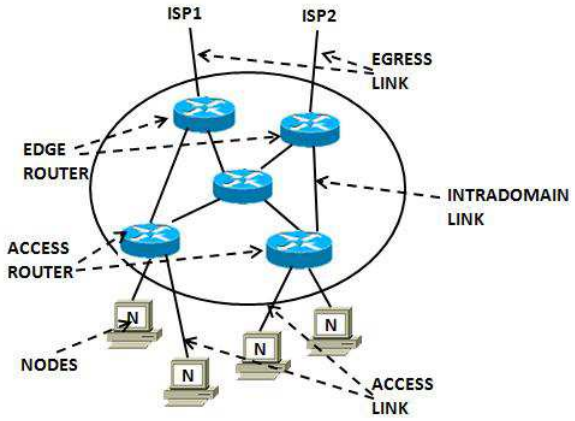


Fig. 1. A Network Multihomed to Two ISPs.

II. The Distributed Problem

In this section we review the centralized routing techniques, in particular the greedy algorithm discussed in [3]. We highlight the difficulties of deploying the centralized approach in a distributed environment. At the same time we examine and identify those modules that can be re-used in a distributed approach. The centralized greedy algorithm basically consists

of three logical steps. The first task is to compute the *rank* of a link. Rank of a link is the difference between the utilization and ideal utilization of the link. Utilization of a link is said to be ideal if the traffic on the link is in proportion to its available bandwidth. Let b_i is the incoming traffic of a node *i* and *L* is the total number of nodes assigned to link *e*. The measured incoming traffic of link e will be given as $U_e = \sum_{i=1}^{L} b_i$. Let A_e is the available bandwidth of link e. The ideal utilization of link e will be given as $IU_e = K \ . \ A_e$ where $K = \frac{\sum_{l=1}^{N} U_l}{\sum_{l=1}^{N} A_l}$. Thus by definition the rank of a link will be given as $|IU_e - U_e|$.

If a link is over-utilized, the rank will denote the quantum by which the utilization of the link exceeds it ideal value. Similarly in the case of under-utilized links, the rank will denote the amount by which the utilization of the link is less than its ideal value. The second step in the centralized greedy approach is to identify the nodes assigned to over-utilized links and select the one that receives the highest incoming traffic. The third step is to select an under-utilized link and then re-assign the node. After a node is re-assigned utilization of the corresponding links are re-computed. The process of re-assignment continues until there are no nodes available for re-assignment or all the links have reached their ideal value.

In the distributed approach, we assume that utilization and available bandwidth metric of all the links are flooded throughout the network. Each node can therefore compute the rank of all the links using the same procedure as in the centralized approach. Further each node will know its current egress route assignment. Thus a node can trivially determine whether it is assigned to an over-utilized link or not. Since a node knows the rank of all the links, it can identify the under-utilized links also. However, in a distributed approach, the algorithm will be run individually by each of the nodes. This effectively means that nodes assigned to over-utilized links will decide in parallel whether to re-assign themselves or not. Thus unlike in the centralized approach, it is not possible to consider the nodes assigned to over-utilized links one at a time unless there is some coordination amongst them. In the centralized approach, after each node re-assignment the status of the link was updated and the decision of whether to re-assign a node or not was based on the current state of links. In a distributed approach, each node will maintain the status of the links locally and it will not be aware of the decisions taken by other nodes. The requirement that as nodes get re-assigned the status of the links also need to be updated globally and concurrently is the main impediment in making our centralized algorithm distributed. Again during the process of selecting an under-utilized link, in the centralized approach the selection was based on the current state of the links. The most under-utilized link was selected. As nodes get assigned to the most under-utilized its utilization will gradually change and after a point it will no longer remain the most under-utilized link. As a result another under-utilized link will get selected. In this way the excess traffic load of the over-utilized links gets proportionately distributed among the under-utilized links. For this scheme to work in a distributed environment the nodes need to have a global view of the link status and the ability to update the status concurrently as nodes get re-assigned. Otherwise, it will result in all nodes selecting only one link, the most under-utilized link.

III. The Distributed Approach

The principal idea of the centralized load balancing algorithm has been to move out traffic from an over-utilized link such that its utilization becomes ideal. Alternatively, this means the proposed centralized algorithms attempts to move out a quantity of traffic from an over-utilized link that is equal to the rank of the link. In the distributed approach if we allow all nodes assigned to an over-utilized link to decide independently whether to re-assign its route or not such that the net traffic moved out from the link is equal to its ranks, then our goal will be achieved. Suppose there are L nodes assigned to an over-utilized link. The total number of re-assignments possible is L^2, including the cases where no nodes get re-assigned or all nodes are re-assigned (an unlikely scenario). The problem is to select from among these possible assignment plans the one that best meets our objectives or the one that will be pursued by the centralized approach. In a distributed network environment there will be no coordination among the nodes, hence a deterministic approach will not be possible. We need to select one of the assignment plans randomly. Decision of the nodes therefore has to be based on probability theory ([5]). We claim that in the distributed approach, if all nodes assigned to an over-utilized link re-assign themselves with a constant probability (P_c), equal to the ratio of the rank of the link to its utilization, then the expected amount of traffic moved out from the link will be equal to its rank. A formal statement of this claim is given in theorem 3.1.

Theorem 3.1: Let *o* be an over-utilized link with rank rank[o] and utilization U_o. If all nodes assigned to the over-utilized link re-assign themselves with a constant probability P_c, where $P_c = \frac{rank[o]}{U_o}$, then the expected amount of traffic moved will be equal to the rank of the link.

Proof: Let X be a random variable that has a value 1 if a node is moved and 0 if not moved.We define the probability mass function of X as

$$Pr(x) = \begin{cases} P_c, & \text{if } x = 1 \\ 1 - P_c, & \text{if } x = 0 \end{cases}$$

Clearly X is a Bernoulli random variable [5]. By definition the expected value of a Bernoulli random variable X is $\mathbf{E}(X) = P_c$. Let L be the total number of nodes assigned to the over-utilized link *o*. Then expected number of nodes moved is $\sum_{i=1}^{L} P_c$. If b_i is the incoming traffic of a node *i*, then expected amount of traffic moved is:

$$\sum_{i=1}^{L} P_c b_i = P_c \sum_{i=1}^{L} b_i = \frac{rank[o]}{U_o} U_o = rank[o]$$

■

A. Static Intradomain Traffic

Theorem 3.1 suggests that in the distributed route control scheme if nodes are re-assigned with a pre-defined constant probability, the expected amount of traffic moved from an over-utilized link will be comparable to that of the centralized scheme. The next issue is how we select an under-utilized link. In this section we assume a network environment, where input traffic is assumed to be unknown and available bandwidth can vary but intradomain traffic is steady. The technique adopted in the centralized scheme for such a network environment was to select the most under-utilized link. We have already indicated such an approach will not work for the distributed scheme since utilization of the links is not updated concurrently with node re-assignments. Ideally the quantity of traffic that should be additionally assigned to an under-utilized should be equal to its rank. In the distributed approach we therefore propose that a node selects an under-utilized link with a probability equal to the rank of the link. The expected amount of traffic assigned to each under-utilized link will be proportional to their rank.

Putting together the pieces, a summary of the distributed heuristic that will be run by each node of the network is given below:

1) Compute rank of the links based on the current network metrics.
2) Check if the node is assigned to an over-utilized link. If *yes* then go to the next step else *exit*.
3) Based on theorem 3.1, the node decides to re-assign itself with probability P_c. If it re-assigns then go to the next step else *exit*.
4) Select an under-utilized link based on its rank and re-assign itself.

B. Dynamic Intradomain Traffic

In actual network conditions not only does traffic and available bandwidth vary but intradomain traffic changes dynamically too ([6]). A full-fledged distributed route control architecture must also consider the intradomain traffic dynamics while allocating egress route to nodes. The scheme that was followed in the centralized architecture was to select from the set of under-utilized links the one with the least intradomain cost. In a distributed environment, the performance of this scheme will depend on how nodes are physically distributed across the network. If nodes who want to re-allocate themselves are well spread out across the domain, the cost of reaching the under-utilized links will differ for the different nodes. As a result the excess traffic load will reasonably get disseminated amongst the under-utilized links.

In general our observation is that nodes assigned to an over-utilized link are attached to different access routers. Hence, the scheme of selecting an under-utilized link based on intradomain distance will usually work. However, consider a hypothetical scenario where all nodes assigned to over-utilized links have the same cost to an under-utilized link. If nodes select an under-utilized link based on intradomain cost, all of them will invariably select the same under-utilized link, the one with the least cost. Understandably utilization of the links will become highly skewed and the performance will deteriorate consequently. This is another ramification of not being able to globally update the link metrics in a distributed environment. To distinguish this situation we refer to it as the so called *clustered node* scenario. A potential solution is to move nodes based on the intradomain cost and once the clustered node problem is detected nodes should be moved based on their link rank. The question is can we detect a scenario where all nodes start moving to the same under-utilized link? This will not be possible without co-ordination among the nodes. In order to overcome the *clustered node* problem, we select an under-utilized link that is based partially on the intradomain cost and partially on the link ranks. We call this new mechanism the *hybrid approach* since it is a fusion of the two techniques to select an under-utilized link. The hybrid mechanism has been designed to pre-empt the occurrence of the *clustered node* problem. In the hybrid approach, a node selects under-utilized links based on the probability of their ranks and assigns a score to each link. The link that is selected first gets the highest score. For example, if there are three under-utilized links the one that gets selected first is assigned a score of 3. The random game is repeated next between the remaining two links and the one that gets selected next is given a score 2 and so on it continues. In the second round, nodes select the same set of under-utilized links based on their intradomain cost and assign a second score to each link. The under-utilized link with the least IGP cost gets the highest score. Continuing with our earlier example the under-utilized link with the least IGP distance to a node will get a score of 3 and so on the assignment of scores will continue. The final score of a link is the sum of these two scores:

$$T\_score(lnk) \leftarrow s1(lnk, rank) + s2(pathcost(node, lnk))$$

The first component (*s1*) is a function of the link rank and the second component (*s2*) is a function of the IGP cost. To ensure that both of these components have equal contribution in the selection of an under-utilized link the score levels of both are same. The link with the highest final score is the candidate to which nodes from an over-utilized link will be moved. Now consider an occurrence of the *clustered node* problem, which is all nodes have the same *s2*. In such a situation *s1* will be used to break the tie.

IV. Experimental Evaluation

In this experiment, we examine the performance of the proposed distributed algorithm under different network conditions. We evaluate the distributed scheme by comparing its performance with that of the centralized scheme under similar traffic conditions.

A. Validating Using Synthetic Data

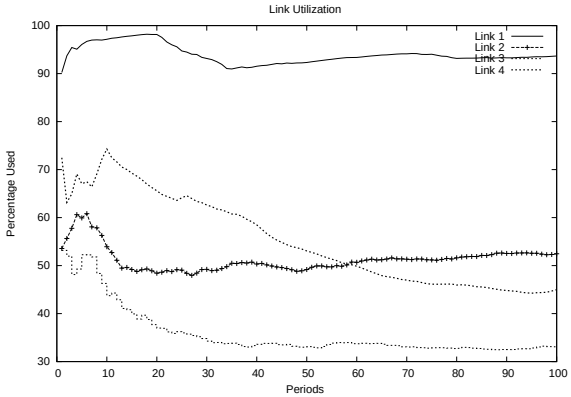
The traffic used in this experiment was synthetically generated using the traffic models of *ns-2* ([1]). In order to compare the performance of the distributed scheme with that of its

centralized counterpart, the experimental settings used in this experiment were exactly the same as that of experiment 2 in [3]. The topology considered consisted of 25 nodes, 50 intradomain links and 4 egress links. Capacity of all the egress links were considered to be the same (10 Mbps). The duration of the experiment was 100 periods. While evaluating the centralized scheme (experiment 2 of [3]), we had seen that the deviation of the utilization of links from their ideal value when no load balancing techniques were used was 30.67 percent. When the centralized route control technique (greedy) was used the deviations of the links were reduced to 6.03 percent. Plots of the utilization of the links when no load balancing techniques were used and when the greedy approach was used are reproduced in figure 2(a) and 2(b) respectively. A plot of the link utilizations, when the distributed route control technique was used is given in figure 2(c). We find that the distributed route control technique significantly reduces the deviation of the links. The average deviation of the links is reduced to 3.80 percent. Moreover, we find that the plot of the distributed approach and the greedy approach are similar. The overall percentage of node re-assignment for the distributed approach is 4.72 and for the greedy approach it was 4.48 percent. The period-wise re-assignments for the distributed and centralized approaches (for the first 25 periods) are compared using a clustered histogram in figure 3. The trend of the re-assignments is also similar.

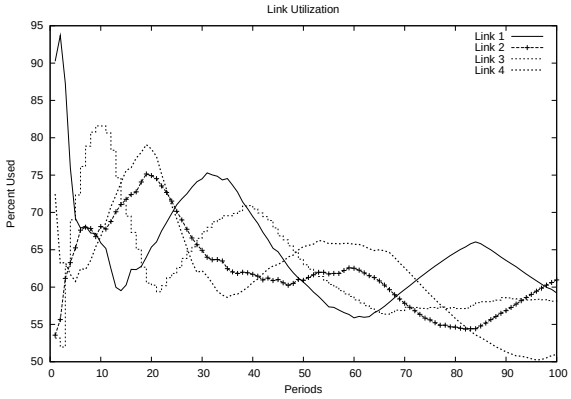
The request-response exchanges between the clients and servers, during the simulation of the distributed approach, were analyzed. Table I, shows the traffic profiles for the different schemes. Using the distributed route control technique to load balance Internet traffic resulted in 8.0 percent improvement in round-trip times as compared to the *default* case. Thus, in this experiment too we find that the performance of the distributed route control technique is same as that of the greedy approach in terms of traffic load balancing, traffic re-assignments and improvement in RTTs.

TABLE I
EXPT 1: CHARACTERISTICS OF NODE TRAFFIC.

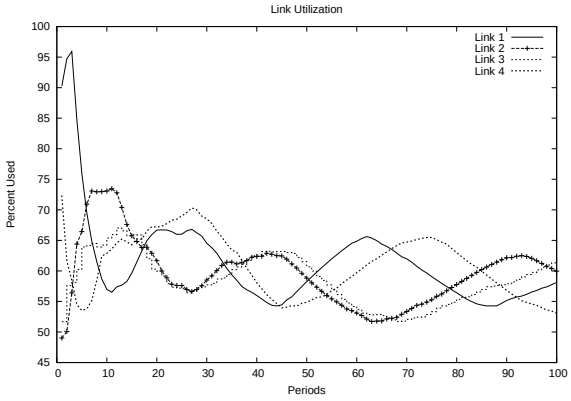
Experiment	Outgoing Traffic (GB)	Incoming Traffic (GB)	RTT (seconds)
Default	0.17	2.47	990.88
Greedy	0.18	2.43	918.20
Distributed	0.18	2.45	911.02



(a) Default



(b) Greedy



(c) Distributed

Fig. 2. Expt 1: Plot of link utilizations

B. Experiment 2: Validating using actual data

In this experiment we validate the distributed approach using real traffic traces. The experimental settings and traffic traces used in this experiment are same as that in experiment 3 of [3]. The topology consisted of 50 nodes and 300 intradomain links. Out of the total nodes, 8 were identified as egress nodes and 17 as access nodes. The traffic trace considered was of 2 hours duration and the number of nodes present was 1500. In figure 4(a), we plot the deviation of the links from their ideal value for the distributed case. These values are further compared with the *default* approach and the greedy heuristic using a stacked histogram. A stacked histogram for each period allows comparison in the following way. The first stack depicts the least deviation, the second stack shows the next higher deviation and so on. As can be seen in the figure, in the first period all the three approaches have the same deviation and so we can see only one histogram. In the second period, the distributed approach has the least deviation which is followed by the greedy approach. The *default* case has the highest deviation in the second period. We can see that in all the periods the deviation of the links are significantly less

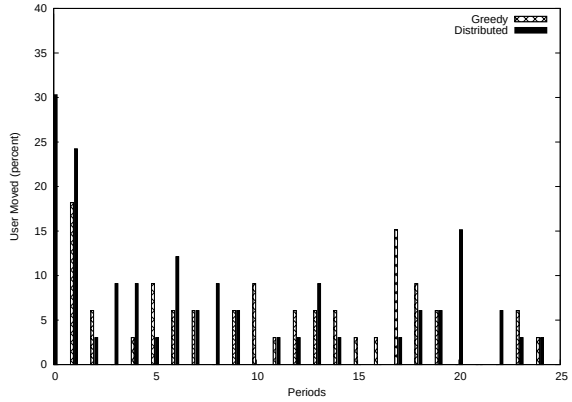
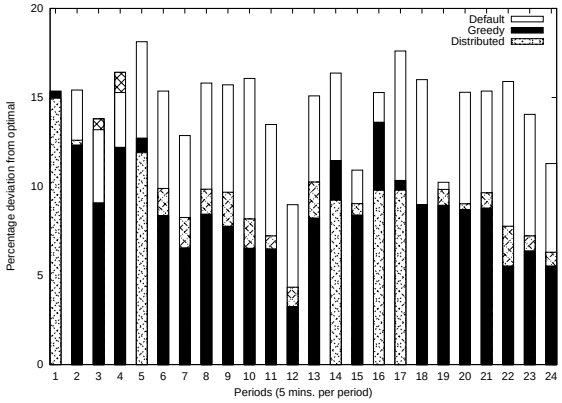
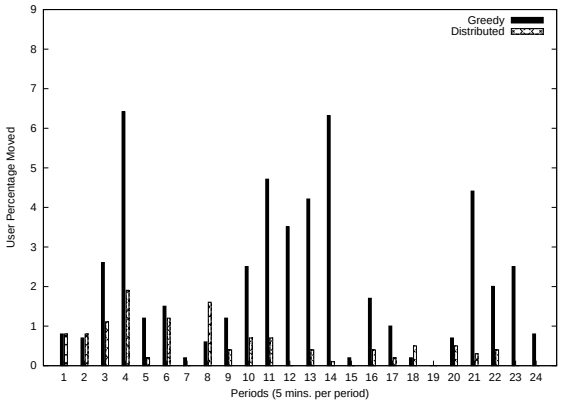


Fig. 3. Expt 1: Comparison of re-assignments using clustered histograms.



(a) Comparison of ranks



(b) Percentage of nodes re-assigned

Fig. 4. Expt. 2: Performance of the distributed approach in static networks.

when the centralized and distributed route control techniques are used. The average percentage deviations of link utilizations for the *default*, centralized and distributed cases from the ideal value are respectively 14.51, 8.64 and 9.53 respectively. In figure 4(b), we plot the re-assignments. In the centralized case, we find that there are a number of periods where the number of node movements is exorbitantly high as compared to other periods. This is because in the centralized approach nodes are deterministically moved out from an over-utilized links such that sum of the bandwidth of these nodes equals the rank of the link. To meet this requirement, in periods where the over-utilized link did not have sufficient number of high throughput nodes, a large number of low throughput nodes were moved. On the other hand in the case of the distributed approach, the number of re-assignments will depend on the rank of the link as well as count of nodes present in an over-utilized link. The number of nodes present in an over-utilized link does not vary widely between periods. Hence, we can see in the figure that the number of nodes re-assigned is more or less uniform in all the periods. The average percentage of nodes re-assigned is about 2 percent for the centralized approach whereas it is less than 1 percent for the distributed case.

V. Conclusion

We have already seen that a centralized routing algorithm which seeks to move the user receiving the highest incoming traffic from an over-utilized link to the least utilized link results in load balancing of traffic on the egress routes and an overall improvement in performance. However, when the network is large, a central manager will not scale well and will become a bottleneck. So a distributed version of the greedy heuristic has been proposed. In this scheme, each user decides to shift from one link to another independently of the decision of other users. In order to prevent all users of an over-utilized link from moving to another link, movement to a link is done probabilistically. When a user finds that it is using an over-utilized link, it decides to move to an under-utilized link with probability P, where P is proportional to the degree of over-utilization of the link. So if a link is heavily over-utilized, more users are likely to move. The cost of reaching a link is taken into account while choosing the under-utilized link to move to. Experiments show that the distributed scheme performs almost as well as the central scheme. Further, it is seen that the distributed scheme handles traffic fluctuations better than the centralized scheme as the latter tends to be aggressive in movements with more information (but imprecise information).

References

[1] ns-2, Network Simulator version 2. http://www.isi.edu/nsnam/ns/.

[2] G. Apostolopoulos, D. Williams, S. Kamat, R. Guerin, and A. Orda. QoS Routing Mechanisms and OSPF Extensions, August 1999.

[3] A.S. Sairam and G. Barua. Load Balancing Inbound Traffic in Multihomed Stub Autonomous Systems. *COMmunication Systems and NETworkS, 2009. COMSNETS 2009. First International Conference on*, pages 1–10, Jan 2009.

[4] D. Kaitz and K. Kompella and D. Yeung. Traffic Engineering Extensions to OSPF Version 2. In *RFC 3630*, September 2003.

[5] W. Feller. *An Introduction to Probabiity Theory and its Applications (2nd Edition), vol. 1*. John Wiley and Sons, 1966.

[6] R. Pang and M. Allman and M. Bennett and J. Lee and V. Paxson and B. Tierney. A First Look at Modern Enterprise Traffic. In *SIGCOMM/USENIX Internet Measurement Conference*, Oct. 2005.

[7] C. Williamson. Internet Traffic Measurement. *Internet Computing, IEEE*, 5(6):70–74, Nov/Dec 2001.