V.Srilakshmi, M. Raja Babu / International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com
Vol. 2, Issue 4, July-August 2012, pp.539-543

Optimization of Overlay Networks using Self-Centric Peer Selection

V.Srilakshmi [1], M. Raja Babu [2]
[1]Mtech Student, Dept of CSE, Aditya Engineering College, Surampalem, Andhra Pradesh, India,
[2]Assoc. Professor, Dept of CSE, Aditya Engineering College, Surampalem, Andhra Pradesh, India,

Abstract

Changing network dynamics is a critical issue in many overlay networks especially in peer-to-peer file sharing systems. Previously these network formations were approached in two ways. First by implementing practical heuristics for cooperative peers, and later performing game theoretic analysis for selfish peers. Selfish Neighbour Selection (SNS) game theory is designed and implemented in overlay routing that unifies the fore mentioned ways. SNS limits number of neighbours for each peer. This approach reduces link monitoring overhead in overlay networks from O (n^2) to O (n). Best response wirings increase the utility of Overlays and Simple heuristic wirings benefits non-selfish nodes. EGOIST, an SNS-inspired prototype overlay routing system that expresses and solves a node's "best response" wiring strategy as a k-median problem on asymmetric distance was developed. It's evaluations on a variety of performance metrics like delay, available bandwidth, and node utilization in Planet Lab confirms its efficiency over existing heuristic overlays. EGOIST (assumes) is based on full-mesh topology which is hard to attain in network dynamics. We propose to use optimized spanning tree protocol (OSTP) that enables traffic traversing the full-mesh portion to take a shortest path from source to destination. Using this full-mesh the claims of EGOIST can be validated..

Index Terms: *Overlay Networks, Selfish Neighbour Selection, Optimized spanning tree protocol.*

I. INTRODUCTION

An overlay network is a virtual network of nodes and logical links that is built on top of an existing network with the purpose to implement a network service that is not available in the existing network. Overlay networks are used for a variety of popular applications including routing [1], content distribution, peer-to-peer (P2P) file sharing and streaming, data-centre applications, and online multi-player games. Connectivity management is a foundational issue in overlay network applications. Establishing Connection to newcomer into existing mesh or re-establishing links between nodes is known as connectivity management. Overlays consist of nodes that are distributed across multiple administrative domains, as such, by utilizing knowledge about the network; these nodes may act selfishly and deviate from the default protocol to maximize the benefit. Selfish behaviour creates additional incentives for nodes to rewire, not only for operational purposes, but also for seizing opportunities to incrementally maximize the local connection quality to the overlay. The impact of adopting selfish connectivity management techniques in real overlay networks has been an open problem.

In overlay network for routing traffic, a node must select a fixed number (k) of immediate overlay neighbours. Previous work has considered this problem from two perspectives: (1) Devising *practical heuristics* for specific applications in real deployments, such as bootstrapping by choosing the k closest links or by choosing k random links in a P2P file-sharing system. (2) Providing abstractions of the underlying fundamental neighbour selection problem that are analytically tractable, especially via game theoretic analysis. The bulk of the work and main results in this area have centered on strategic games where edges are undirected, access costs are based on hop-counts, and nodes have potentially unbounded degrees. While this existing body of work is extremely helpful for laying a theoretical foundation and for building intuition, it is not clear how or whether the guidance provided by this prior work generalizes to situations of practical interest, in which underlying assumptions in these prior studies are not satisfied. Aspect not considered in previous work is the consideration of settings in which some or even most players do not play optimally – a setting which we believe to be typical.

In this paper, the central aspects of our model are bounded degree, directed edges, non-uniform preference vectors, and representative distance functions. Overlays are dominated by selfish nodes, the resulting stable wirings are already so highly optimized that even non-selfish newcomers can extract near optimal performance through heuristic wiring strategies.

Based on above observations, we design, implement, and deploy EGOIST, a prototype overlay routing network built around best response wiring strategies. EGOIST serves as a building block for the construction of efficient and scalable overlay applications consisting of (potentially) selfish nodes. We demonstrate (1) that overlay routing EGOIST is significantly more efficient than systems utilizing common heuristic neighbour selection strategies under multiple performance metrics, including delay, system load and available bandwidth. (2) We demonstrate that the performance of EGOIST approaches that of a full-mesh topology, while achieving superior scalability, requiring link announcements proportional to nk compared to n^2 for a full mesh topology. We propose to use optimized spanning tree protocol (OSTP) that enables traffic traversing the full-mesh portion to take a shortest path from source to destination. Our experimental results show that EGOIST remains highly effective under significant churn and incurs minimal overhead.

II. OVERLAY NETWORK MODEL

Previous work on overlay network creation [2], [3] has focused on physical telecommunication networks and primarily the Internet. Overlay networks are substantially different [6] which consider the following overlay network model. The central aspects of our model are:

Bounded Degree: Most protocols used for implementing overlay routing or content sharing impose hard constraints on the maximum number of overlay neighbours. In overlay routing systems, the number of immediate nodes has to be kept small so as to reduce the monitoring and reporting overhead imposed by the link-state routing protocol implemented at the overlay layer. Node degrees were *implicitly bounded* (as opposed to *explicitly constrained*) by virtue of the trade-off between the additional cost of setting up more links and the decreased communication distance achieved through the addition of new links.

Directed Edges: Another important consideration in our work relates to link directionality. Prior models have generally assumed bi-directional [5] (undirected) links. This is an acceptable assumption that fits naturally with the unbounded node degree assumption for models that target physical telecommunication network because actual wire-line communication links are almost exclusively bidirectional. In overlay settings we consider, this assumption needs to be relaxed since the fact that node v forwards traffic or requests to node u does not mean that node u may also forward traffic or requests to v. Undirected links are created by the establishment of two directed links.

Non-uniform preference vectors: In our model, a vector is supplied to each node that captures its local preference for all other destinations. In overlay routing such preference may capture the percentage of locally generated traffic that a node routes to each destination, and then the aggregation of all preference vectors would amount to an origin/destination traffic matrix. In P2P overlays such preference may amount to speculations from the local node about the quality of, or interest in, the content held by other nodes.

III. THE EGOIST OVERLAY ROUTING SYSTEM

In previous work, it is not clear; what is the average performance gain when SNS wiring strategies are used in highly dynamic environments, whether such overlays are robust against churn, and whether they scale and whether it is practical to build overlays to support best response and how to incorporate additional metrics other than delay, *e.g.*, bandwidth. In order address the problems, we implemented EGOIST, and SNS-inspired prototype overlay routing network, serves as a building block for the distributed construction of efficient and resilient overlays where both individual and social performance is close to optimal. EGOIST is based on full-mesh topology which is hard to attain in network dynamics. We propose to use optimized spanning tree protocol (OSTP) that enables traffic traversing the full-mesh portion to take a shortest path from source to destination.

A. Basic Design EGOIST is a distributed system which allows the creation and maintenance of an overlay network. Here every node selects and continuously updates its k overlay neighbours in a selfish manner—namely to minimize its (weighted) sum of distances to all destinations under shortest-path routing. For ease of presentation, Instead of bandwidth ad node utilization, we will assume that *delay* is used to reflect the cost of a path.

In EGOIST, a *newcomer* overlay node v_i connects to the system by querying a *bootstrap* node, from which it receives a list of *potential* overlay neighbours. The newcomer connects to at least one of these nodes, enable link state routing protocol running at the overlay layer. As a result, after some time, v_i obtains the full residual graph G_{-i} of the overlay. By running all-pairs shortest path algorithm on G_{-i} , the newcomer is able to obtain the pair-wise distance (delay) function d_{G-i}. In addition to this information, the newcomer estimates d_{ij}, the weight of a potential direct overlay link from itself to node v_i, for all $v_i \in V_{-i}$. Using the values of d_{ij} and d_{G-i} , the newcomer connects to G_{-i} using one of the wiring strategies. In our implementation, each node acts as a server that listens to all the messages of the link state protocol and propagates them only to its immediate neighbours. In order to reduce the traffic in the system, messages are dropped that have been received more than once and propagates unique messages. There are also two threads, one for estimating d_{ij}, and one responsible for estimating the new wiring and propagating the wiring to the immediate neighbours. In order to minimize the load in the system, a node propagates its wiring to its immediate neighbours only if this changes.

B. optimized spanning tree protocol

An optimized spanning tree protocol (OSTP) minimizes latency and provides high throughput in a full-mesh portion of a network, and is compatible with external networks where a standard spanning tree protocol is used. The OSTP enables traffic traversing the full-mesh portion to take a shortest path from source to destination through use of full-mesh connectivity. In some embodiments, a cluster includes a plurality of servers connected in a full mesh, and the OSTP is used on internal ports of the servers. In some embodiments, the OSTP is configured on a per-VLAN basis. In some embodiments, the servers exchange special messages enabling determination of full-mesh connectivity. In further embodiments, sending of the special messages is suppressed on certain port types, such as external ports. In some embodiments, determination of the full-mesh connectivity disables use of a standard spanning tree protocol and/or enables use of OSTP on the full-mesh portion.

C. Dealing with Churn

EGOIST's BR (best response) neighbour selection strategy assumes that existing nodes never leave the overlay. Therefore, even in an extreme case in which some nodes are reachable through only a unique path, a node can count on this path always being in place. Overlay routing networks (e.g., RON [3]) are not inherently prone to churn to the extent that file-sharing P2P networks. Nonetheless, nodes may occasionally go down, or network problems may cause transient disconnections until successive re-wirings establish new paths. One could re-formulate the BR objective function used by a node to take into account the churning behaviour of other nodes. This, however, requires modelling of the churn characteristics of various nodes in an overlay, which is not feasible in large networks.

In EGOIST, we follow a different approach reminiscent of how k-Random and k-Closest strategies ensure overlay connectivity. We introduce a hybrid wiring strategy (Hybrid BR), in which each node uses k_1 of its k links to selfishly optimize its performance using BR, and "donates" the remaining $k_2 = k – k_1$ links to the system to be used for assuring basic connectivity under churn. We call this wiring "hybrid" because, in effect, two wiring strategies are in play – a selfish BR strategy that aims to maximize local performance and a selfless strategy that aims to maintain global connectivity by providing redundant routes.

There are several ways in which a system can use the k_2 donated links of each node to build a connectivity backbone. Using k-MST (a centralized construction) to maintain connectivity is problematic, as it must always be updated, not to mention the overhead and complexities involved in establishing ($k_2/2$)-MSTs. To avoid these complexities, EGOIST uses a simpler solution that forms $k_2/2$ bidirectional cycles. For $k_2 = 2$, it allows for the creation of a single bidirectional cycle. For higher k_2, the system decides $k_2/2$ *offsets* and then each node connects to the nodes taken by adding (modulo n) its id to each offset. If k_2 is small then the nodes will need to monitor (*e.g.*, ping) the backbone links closely so as to quickly identify and restore disconnections. With higher k_2 the monitoring can be more relaxed due to the existence of alternative routes through other cycles. Computing BR using k_1 links granted the existence of the k_2 links can be achieved by restricting the set candidate immediate neighbours for swapping.

We have implemented HybridBR in EGOIST. As hinted above, donated links are monitored aggressively so as to recover promptly from any disconnections in the connectivity backbone through the use of frequent heartbeat signalling. On the other hand, the monitoring and upkeep of the remaining BR links could be done lazily, namely by measuring link costs, and recomputing BR wirings at a pace that is convenient to the node—a pace that reduces probing and computational overheads without risking global connectivity.

To differentiate between these two types of link monitoring strategies (aggressive versus lazy), in EGOIST we allow rewiring of a dropped link to be performed in one of two different modes: *immediate* and *delayed*. In immediate mode, re-wiring is done as soon as it is determined that the link is dropped, whereas in delayed mode re-wiring is only performed (if necessary) at the preset *wiring epoch* T. unless otherwise specified, we assume a delayed re-wiring mode is in use.

D. Cost Metrics

The choice of an appropriate "cost" of traversing a link depends largely on the application. In EGOIST we consider the following metrics:

Link and Path Delays: Delays are natural cost metrics for many applications, including real-time ones. To obtain the delay cost metric, a node needs to obtain estimates for its own delay to potential neighbours, and for the delay between pairs of overlay nodes already in the network. In EGOIST, we estimate directed (one-way) link delays using two different methods: an active method based on ping, and a passive method using the pyxida virtual coordinate system. Using ping, one-way delay is estimated to be one half of the measured ping round-trip-times (RTT) averaged over enough samples. Using pyxida, delay estimates are available through a simple query to the pyxida system. Using ping produces more accurate estimates, but subjects the overlay to added load, whereas using pyxida produces less accurate estimates, but consumes much less bandwidth.

Node Load: For many overlay applications, it may be the case that the primary determinant of the cost of a path is the performance of the nodes along that path. In EGOIST, we allow the use of a variation of the delay metric in which all outgoing links from a node are assigned the same cost, which is set to be equal to the measured load of the node. In EGOIST, we did this by querying the CPU load of the local node, and computing an exponentially-weighted moving average of that load calculated over a given interval.

Available Bandwidth: Another important cost metric, especially for content delivery applications, is the available bandwidth on overlay links. In EGOIST, we used pathChirp[10], a light-weight, fast and accurate tool, which fits well with specific constraints, namely: it does not impose a high load on nodes since it does not require the transmission of long sequences of packet trains, and it does not exceed the maxburst limits of Planet lab.

IV. PERFORMANCE EVALUATION OF EGOIST

To facilitate comparisons between various neighbour selection strategies, we often report the *normalized routing cost* , which is the ratio of the cost achievable using a given strategy to that achievable using BR.

Control Variables: In our first set of experiments, our aim is to identify for the three metrics of interest the payoff from adopting a selfish neighbour selection strategy, *i.e.*, using a BR strategy in EGOIST. This payoff will depend on many variables. While some of these variables are *not* within our control (*e.g.*, the dynamic nature of the Internet as reflected by variability in observed Planet Lab conditions), others are within our control, *e.g.*, n, T, and the various settings for our active measurement techniques.

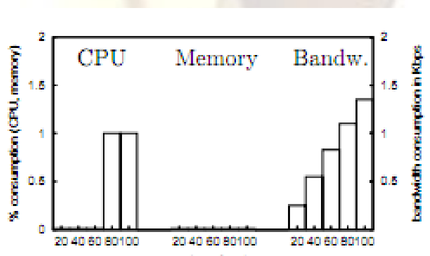

Fig 1: Percentage of CPU and memory consumption, and bandwidth consumption in EGOIST. The metric is delay via ping.

One control variable that is particularly important is the number of direct neighbours, k that an EGOIST node is allowed to have. In any ways, k puts a premium on the significance of making a judicious choice of neighbours. For small values of k, choosing the right set of neighbours has the potential of making a bigger impact on performance, when compared to the impact for larger values of k. In Fig 1, we show the average CPU and memory utilization, along with the average bandwidth consumption to maintain the overlay per node. CPU and memory consumption is close to 0%, and the bandwidth consumption per node is negligible.

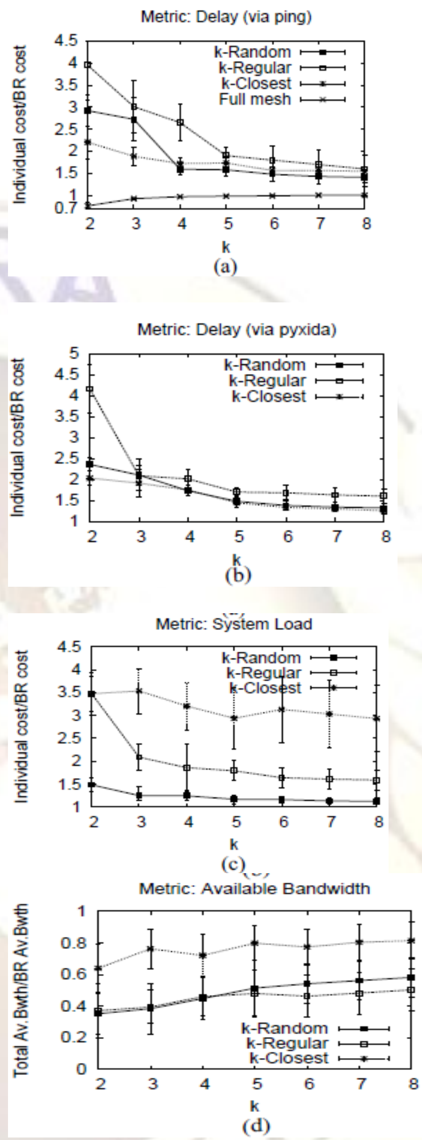

Fig. 2. Normalized individual costs and 95th-percentile confidence intervals with respect to BR cost, under different metrics, of various neighbour selection strategies in a 50-node EGOIST overlay.

Results for Delay Metric: Figures 1(a) and (b) show the performance of the various neighbour selection strategies in EGOIST normalized with respect to that achievable using BR when the metric of interest is the overlay path delay over a range of values for k (using ping and pyxida). These results show that BR outperforms all the other wiring strategies, especially when k is small. The performance advantage of BR in terms of routing delay stands, even for a moderate number of neighbours. These results confirm the superiority of BR relative to other strategies, but do not give us a feel for how close is the performance of EGOIST using BR wiring to the "best possible" performance. To do so, we note that by allowing nodes to connect to all other nodes in the overlay, we would be creating a complete overlay graph with $O(n^2)$ overlay links, obviating the need for a neighbour selection strategy. With respect to the other heuristics, the results in Figures 2 (a) and (b) show that k-Closest outperforms k-Random when k is small, but that k-Random ends up outperforming k-Closest for slightly larger values of k. This can be explained by noting that k-Random ends up creating graphs with much smaller diameters than the grid-like graphs resulting from the use of k-Closest, especially as k gets larger. In all experiments, k- Regular performed the worst. In [9] we also show that BR wiring strategy is robust to cheating.

Results for Node Load: Figure 2 (c) shows the results we obtained using the node load metric, where the path cost is the sum of the loads of all nodes in the path. These results show clear delineations, with BR delivering the best performance over all values of k, k-Random delivering the second-best performance, and k-Closest delivering the worst performance as it fails to predict anything beyond the immediate neighbour, especially in light of the high load variance in Planet Lab.

Results for Available Bandwidth: Figure 2(d) shows the results we obtained using available bandwidth as the cost metric. Recall that, here, the objective is to get the highest possible *aggregate* bandwidth to all destinations (again, assuming a uniform preference for all destinations) – thus, larger is better. These results show trends that are quite similar to those obtained for the delay metric, with BR outperforming all other strategies—delivering a two-fold to four-fold improvement over the other strategies, over a wide range of values of k.

V. CONCLUSION

Selfish neighbour selection in a richer model that captures the nuances of overlay applications more faithfully than previous work and strictly enforced neighbour budgets and has come up with a series of findings with substantial practical value for real overlay networks. First, when compared with simple random and myopic heuristics, we have shown that a best response (*i.e.*, selfish) selection of neighbours leads to the construction of overlays with much better performance. The reason is that by being selfish, nodes embark on a distributed optimization of the overlay that turns out to be beneficial for all. Secondly, through the development, implementation, and deployment of EGOIST, we have established that Best-Response neighbour selection strategies that they provide a substantial performance boost when compared to currently used heuristics, and that they scale much better simple heuristics. Optimized spanning tree protocol (OSTP) enables traffic traversing the full-mesh portion to take a shortest path from source to destination. Finally, for achieving real-time requirements and carrying the traffic generated by an online multi-player P2P game, we use EGOIST prototype. Egoist incurs minimal overhead and it can be used as a building block for efficient routing in overlay applications.

VI REFERENCES

[1] D. Andersen, H. Balakrishnan, F. Kaashoek, and R. Morris, "Resilient Overlay Networks," in *Proc. ACM SOSP'01*.

[2] B.-G. Chun, R. Fonseca, I. Stoica, and J. Kubiatowicz, "Characterizing Selfishly Constructed Overlay Routing Networks," in *Proc. IEEE INFOCOM'04*.

[3] B. G. Rocha, V. Almeida, and D. Guedes, "Improving Reliability of Selfish Overlay Networks," in *Proc. WWW'06*.

[4] G. Smaragdakis, V. Lekakis, N. Laoutaris, A. Bestavros, J. W. Byers, and M. Roussopoulos, "EGOIST: Overlay Routing using Selfish Neighbor Selection," in *Proc. ACM CoNEXT'08*.

[5] N. Laoutaris, G. Smaragdakis, A. Bestavros, and J. W. Byers, "Implications of Selfish Neighbor Selection in Overlay Networks," in *Proc. IEEE INFOCOM'07*.

[6] X. Yang and G. de Veciana, "Performance of Peer-to-peer Networks: Service Capacity and Role of Resource Sharing Policies," *Perform. Eval.*, vol. 63, no. 3, pp. 175–194, 2006

[7] A. Young, J. Chen, Z. Ma, A. Krishnamurthy, L. L. Peterson, and R. Wang, "Overlay Mesh Construction Using Interleaved Spanning Trees," in *Proc. IEEE INFOCOM'04*.

[8] V. Arya, N. Garg, R. Khandekar, A. Meyerson, K. Munagala, and V. Pandit, "Local Search Heuristics for k-Median and Facility Location Problems," *SIAM J. on Computing*, vol. 33, no. 3, pp. 544–562, 2004.

[9] G. Smaragdakis, "Overlay Network Creation and Maintenance with Selfish Users," *Ph.D. Dissertation, Boston University*, 2009.

[10] V. Ribeiro, R. Riedi, R. Baraniuk, J. Navratil, and L. Cottrell, "pathChirp: Efficient Available Bandwidth Estimation for Network Paths," in *Proc. PAM'03*.