Constructing Traffic-Aware Overlay Topologies: A Machine Learning Approach

Benjamin D. McBride
Kansas State University
2061 Rathbone Hall
Manhattan, KS 66506
Email: bdm7935@ksu.edu

Caterina Scoglio
Kansas State University
2061 Rathbone Hall
Manhattan, KS 66506
Email: caterina@ksu.edu

***Abstract*— Recent game-theoretic approaches to constructing overlay network topologies have not been scalable. This paper introduces a machine learning approach to constructing overlay networks. The machine learning approach learns characteristics from small networks constructed using a game-theoretic approach. The knowledge learned is then used to construct larger networks. The results show that the machine learning approach closely approximates the game-theoretic networks for a wide arange of network parameters, while being scalable.**

I. INTRODUCTION

End-to-end routing performance is not optimal in the Internet. Since end hosts have no control over the routing of data, providing quality of service (QoS) guarantees is a difficult problem. Many applications, such as video streaming, voice over IP, and P2P, would like to have performance and reliability guarantees. Unfortunately, the Internet provides zero guarantees. Many solutions have been proposed to add QoS to the infrastructure of the Internet. These solutions have little chance of being accepted due to the inertia of the existing system. However, a popular approach that has been recently proposed is an *overlay routing network*, or overlay network.

An overlay network is an *application-layer* network composed of logical links over the existing Internet infrastructure. In an overlay network, the hosts route data through intermediate nodes in the overlay network according to application specific performance metrics. Surprisingly, the end-to-end routing performance can be significantly improved by routing through intermediate nodes [1] [2]. Obviously, one of the fundamental problems in overlay networks is deciding which intermediate nodes to use. Typically, overlay networks construct a virtual topology of logical links toward other nodes in the overlay network. Common routing algorithms, such as the link-state or distance vector algorithm, are then used to route data according to an application-specific performance metric.

Recent approaches, presented in Section II, for creating overlay network topologies based on game theory have been proposed [3] [4] [9]. Unfortunately, these approaches are intractable. The primary contribution of this work is the introduction of a machine learning approach that approximates the construction of game-theoretic overlay networks. The machine learning approach learns from data generated from the construction of small networks using the game theory model, presented in Section III, and applies the knowledge to larger networks in a scalable fashion. The machine learning approach is introduced in Section IV. Section V shows that the machine learning approach closely approximates the game-theoretic approach to constructing overlay networks, while being computationally inexpensive. Future work and conclusions are presented in Section VI.

II. RELATED WORK

A. A game-theoretic approach

Recent approaches to the formation of overlay network topologies are derived from game theory. These studies propose a network creation model where each node tries to minimize its own ”cost”, without consideration for the global interests of the network. Nodes seek to minimize cost until an equilibrium, called a *Nash equilibrium*, is reached where no node can decrease its cost without some other node first increasing its cost.

In a game-theoretic approach to network creation, each node in the network acts as an independent agent, selecting its neighbors, and paying for logical links to those neighbors according a cost model [3]. The subset of nodes that form the neighborhood of a node is called a *strategy*. Each node acts selfishly, selecting a strategy that reduces its cost. The union of these logical links forms a network that is created without central design or coordination.

Obviously, the resulting network topologies are largely dependent on the specific cost model used. The cost for node i to select strategy B_i as proposed by Fabrikant, et al is [3]:

$$C_i = \alpha |B_i| + \sum_{j \in N} d_G(i,j) \tag{1}$$

where $d_G(i,j)$ is the cost distance between nodes i and j in the overlay network G and α is a parameter balancing the cost of creating links to other nodes and the distance to other nodes in the overlay network. These different cost terms can be seen as hardware and quality of service costs or probing and routing costs, depending on the application. Fabrikant, et al prove that a wide range of network topologies can be constructed simply by varying the value of α [3].

B. A generalized approach

Chun, et al, generalized the cost model proposed by Fabrikant, et al, and studied the selfishly constructed networks formed by the non-cooperative game [4]. They generalized the cost model in three areas:

1) The link cost is no longer constant, but rather is a function of the node j being connected to.
2) The distance function is generalized to be any performance metric between nodes.
3) The possible neighbors that a node can connect to is constrained.

The resulting cost model is:

$$C_i = \alpha \sum_{j \in B_i} h_{i,j} + \sum_{j \in N} d_G(i,j) \tag{2}$$

where $h_{i,j}$ is the cost to create a logical link between i and j.

Chun, et al, first consider small network topologies formed with varying values of α and different link cost functions. An iterative exhaustive search, where each node in turn finds its minimal cost strategy given the rest of the network, is used until the network reaches equilibrium. They demonstrate that a wide range of network topologies can be constructed: complete graphs, densely connected graphs, sparsely connected graphs, stars, k-core stars, and trees. When considering a more realistic search, where an iterative greedy search is done over a constrained neighborhood set, they find that networks can be constructed that have desirable *global* properties. They demonstrate that power-law networks can be formed as well as networks that are highly resistant to node failure and attack. They also conclude that there is a fundamental tradeoff between the performance of a network and its resilience.

III. Cost Model

This paper uses the traffic demand aware cost model presented in [9] and [10]. We reproduce it here for convenience. We assume that each node needs to select its neighbors in a distributed fashion. Each node has imperfect information and does not know the other nodes' neighbors. Let $G = (N, L)$ be the graph representing the overlay network and $G_u = (N, E)$ be the graph representing the underlay, or physical, network. N is the set of nodes that are in both the overlay and physical network, while the set of logical links L can be different from the set of physical links E. A logical link $l \in L$ is constructed on a path composed of physical links $e \in E$. Each node $i \in N$ has a traffic demand toward a node subset $S_i \subseteq N$. Let $t_{i,j}$ be the traffic demand between node i and node j in the subset S_i. The objective for each node is to create logical links to a subset of nodes, $B_i \in 2^{N-\{i\}}$, such that its total cost is minimized. We define cost using two components:

1) Link cost: cost to create and maintain a logical link between node i to node j
2) Transport cost: cost to transport the traffic demand between nodes i and j

The cost for node i to connect to each node $j \in B_i$ and carry traffic demand $t_{i,j}$ to each node $j \in S_i$ is defined as the sum of the link cost and transport cost:

$$C_i = \alpha \sum_{j \in B_i} h_{i,j} + \sum_{j \in S_i} d_G(i,j) t_{i,j} \tag{3}$$

where α is the relative cost of creating a logical link to the cost of transporting the traffic demand through the existing network, $h_{i,j}$ is the linking cost between i and j, and $d_G(i,j)$ is the distance in the overlay network to node j (∞ if j is unreachable from i).

It is important to note that the linking cost between i and j, is a general function that can represent a wide variety of metrics. The transport cost term in (3) can also be thought of as a generalized distance function [4]. Additionally, once a logical link has been established from i to j any node in the network can use the link. We do not consider the link to be directed in terms of its use and for calculation of d_G.

Objective: The objective for each node is to find its minimal cost strategy, the strategy that minimizes the nodes cost according to Equation 3.

Finding the minimal cost strategy for a node is a NP-hard problem [10] [3]. This means that heuristic and approximation approaches must be used in order to scale to larger size networks. Even a greedy search is $O(N^3 \log N)$ and does not scale well to large networks [10]. A better heuristic is needed.

IV. A Machine Learning Approach

A. Motivation

Since an exhaustive search of a node's strategy space is NP-hard and a greedy search is $O(N^3 \log N)$ heuristic approaches are needed for the traffic-aware approach to scale to larger networks. An algorithm with runtime complexity that is linear in the number of nodes is desired. Since any approach that relies on a distance calculation must be at least $O(N \log N)$ a good heuristic is not obvious.

A scalable approach must determine *good* nodes to make logical links towards without knowledge of the distance between nodes. This is clearly a good domain for machine learning. A good definition of the learning task is given by Mitchell [12]:

Definition: A computer program is said to *learn* from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.

For the overlay network learning problem the task is clearly whether to create a logical link to a node or not. The performance measure is the number of correctly classified nodes and the experience is gathered from topologies formed using iterative exhaustive search.

B. Data Acquisition

The training data for the learning algorithm, was obtained after a node had selected the low-cost strategy using iterative exhaustive search. A record of various attributes for each node in the network and whether the particular node is a member of the low-cost strategy was recorded. This was done at each iteration of the iterative exhaustive search algorithm for networks with 20-nodes. Various values of α, different traffic-demand distributions, and different maximum node degree constraints were used. Table I gives the names and brief descriptions of all the attributes used in the training data.

The complete set of attributes represents the *full* dataset. As mentioned previously, it is desirable to remove all distance calculations from the classification task. We also wanted to reduce the number of attributes that needed to be tested in order to reduce the complexity of the learning problem. We ran two attribute selection algorithms, *CfsSubsetEval* and *InfoGainAttributeEval*, from the Waikato Environment for Knowledge Analysis (WEKA) [13] to determine the most relevant attributes.

TABLE I
Attributes and their description in the full dataset

Attribute name	Type	Description
N	numeric	Number of nodes
alpha	numeric	α
maxDegreeExceeded	boolean	Is $maxDegree$ constraint violated by adding a link?
maxDegreeRatio	numeric	Ratio of the degree of node j to the $maxDegree$
ijTrafficDemand	numeric	Traffic demand from i to j
jMeanDistance	numeric	Average distance from j to other nodes.
Degreej	numeric	The degree of node j
OutDegreej	numeric	The out degree of node j
InDegreej	numeric	The in degree of node j
jiLinkState	boolean	Does a link exist from j to i?
TrafficDemandToAlpha	numeric	Traffic-demand to alpha ratio
linkState	boolean	Is link made towards node j?

TABLE II
Information gain of attributes

Attribute name	Information Gain
maxDegreeRatio	0.10409
TrafficDemandToAlpha	0.09422
ijTrafficDemand	0.05008
maxDegreeExceeded	0.04513
Degreej	0.02487
InDegreej	0.02448
alpha	0.02263
jiLinkState	0.02252
jMeanDistance	0.00872
OutDegreej	0.00615
N	0

The information gained from the different attributes is given in Table II. As it turns out the mean distance from node j to other nodes, $jMeanDistance$, does have low information gain. This is promising since it is desirable to neglect the distance attribute. The attributes that were selected by the *CfsSubsetEval* algorithm were: *maxDegreeExceeded*, *jiLinkState*, and *TrafficDemandToAlpha*. Once again the distance attribute was not selected. Based on these results the *reduced* dataset was created that included the following attributes: *maxDegreeExceeded*, *maxDegreeRatio*, *jiLinkState*, and *TrafficDemandToAlpha*.

C. Algorithm Selection

The next step in designing a learning system is to choose the target representation. The representation is closely tied to the choice of learning algorithm. Ideally, the representation would be human-readable. This rules out various neural network approaches like multi-layered perceptrons, radial basis functions, and support vector machines. For performance reasons, lazy learning algorithms such as nearest neighbor approaches were also ruled out. This left primarily decision tree and rule learning approaches. Since much of the training data is numeric, the learning algorithm must be capable of handling numeric as well as nominal valued attributes. Based on preliminary experiments the following learning algorithms were selected for further investigation: *J48*, *JRip*, *PART*, and *Ridor*.

WEKA's Experimenter was used to compare the four learning algorithms against ten data sets. Five datasets were formed using random sampling from the *full* dataset. Five datasets were formed using random sampling from the *reduced* dataset. Each dataset contained 10 percent of the examples from the complete dataset. They were reduced in size because of memory constraints associated with WEKA and the Java Virtual Machine.

Table III gives the percentage of correctly classified instances for each data set over each algorithm. Each algorithm was run on each dataset ten times. The percentage of correctly classified instances does not tell the whole story though, especially when considering an unbalanced dataset. Precision and recall values provide a different view of the predictive accuracy.

Definition: *Precision* is the number of true positive examples compared to the number of examples that are classified as positive.

Definition: *Recall* is the number of true positive examples compared to the number of examples that are actually positive.

Table IV and Table V provide the precision and accuracy results of the learning algorithms. It is interesting to note that the Ridor algorithm has poor precision results, but high recall results. The final metric used in deciding which learning algorithm to use was the number of generated rules. Table VI shows that the JRip algorithm produces a much smaller set of rules than J48, PART, or Ridor, while still maintaining comparable predictive accuracy. Consequently, the JRip learning algorithm was selected and run on the *reduced* dataset.

TABLE III
PERCENT CORRECT

Data Set	J48	JRip	PART	Ridor
full1	97.52	97.30 ●	97.38	96.81 ●
full2	97.45	97.19 ●	97.32	96.93
full3	97.59	97.47	97.43	97.03
full4	97.47	97.29 ●	97.32	96.80
full5	97.63	97.49 ●	97.53	97.06
reduced1	97.30	97.29	97.34	96.92
reduced2	97.25	97.26	97.24	97.00
reduced3	97.45	97.47	97.38	97.10
reduced4	97.30	97.28	97.27	96.96
reduced5	97.43	97.44	97.44	97.23

○, ● statistically significant improvement or degradation from J48

TABLE IV
PRECISION

Data Set	J48	JRip	PART	Ridor
full1	0.94	0.93	0.94	0.84 ●
full2	0.94	0.94	0.93	0.86 ●
full3	0.93	0.93	0.92	0.85
full4	0.94	0.93	0.92	0.84 ●
full5	0.94	0.93	0.95	0.86
reduced1	0.93	0.92	0.93	0.86
reduced2	0.93	0.93	0.92	0.87 ●
reduced3	0.91	0.91	0.91	0.86
reduced4	0.93	0.92	0.93	0.88
reduced5	0.94	0.93	0.93	0.88

○, ● statistically significant improvement or degradation from J48

V. RESULTS

The rules that were learned were used in place of the exhaustive and greedy search algorithms. Because there is no implicit pressure to maintain a connected graph a greedy connection algorithm was also used with the rule-based approach. If the network is disconnected after a node selects its strategy according to the rule set, the node attempts to connect to nodes with high degree in other subgraphs, so far as the maximum node degree constraints are not violated. The algorithm was run on 20 node and 100 node networks with varying values of α and $maxDegree$. Tables VII, VIII, and IX show that the graphs generated by the rule-based algorithm have similar properties to those created by the iterative exhaustive and greedy searches.

For the 100 node networks, only the iterative greedy search and rule-based search are used. The exhaustive search is not used because of computational limits. Tables X, XI, and XII show that the graphs generated by the rule-based search have similar properties to those created by the iterative greedy search. For large values of α

TABLE V

RECALL

Data Set	J48	JRip	PART	Ridor
full1	0.80	0.78 ●	0.78	0.84 ○
full2	0.79	0.76	0.78	0.82 ○
full3	0.80	0.79	0.80	0.83
full4	0.79	0.77	0.78	0.82
full5	0.80	0.80	0.79	0.83
reduced1	0.78	0.79	0.79	0.82
reduced2	0.77	0.77	0.78	0.81 ○
reduced3	0.80	0.80	0.80	0.82
reduced4	0.77	0.78	0.77	0.80
reduced5	0.78	0.79	0.79	0.82 ○

○, ● statistically significant improvement or degradation from J48

TABLE VI

NUMBER OF RULES

Data Set	J48	JRip	PART	Ridor
full1	55.00	8.50 ●	43.70	50.20
full2	49.10	8.00 ●	46.50	48.00
full3	49.60	10.10 ●	42.20	42.90
full4	51.70	8.00 ●	48.70	49.90
full5	55.10	7.30 ●	42.90	50.20
reduced1	22.30	8.90 ●	19.00	53.30 ○
reduced2	24.30	9.60 ●	20.70	53.20 ○
reduced3	22.80	9.50 ●	16.90	47.50 ○
reduced4	20.60	8.20 ●	19.40	56.10 ○
reduced5	19.60	9.20 ●	16.80	54.20 ○

○, ● statistically significant improvement or degradation from J48

the rules-based networks have smaller characteristic path lengths and higher spectral radii. This is a because more nodes make logical links towards the root node in the network. Additional simulations were run with degree-constrained nodes and different traffic demand distributions. Like the results presented here, the rule-based approach results in networks with similar properties to the exhaustive and greedy search methods. However, the rule based approach does not behave exactly the same under these parameters. Further results and discussion can be found in [9] and [10].

TABLE VII

NUMBER OF EDGES FOR 20 NODE NETWORKS

α	Exhaustive	Greedy	Rules
0.5	132.0	132.0	129.3
1	89.0	89.0	79.4
5	19.0	20.3	19.2
60	19.0	19.0	19.2

TABLE VIII

CHARACTERISTIC PATH LENGTH FOR 20 NODE NETWORKS

α	Exhaustive	Greedy	Rules
0.5	1.30	1.30	1.31
1	1.53	1.53	1.59
5	1.90	1.91	2.55
60	2.68	3.23	2.55

TABLE IX

SPECTRAL RADIUS FOR 20 NODE NETWORKS

α	Exhaustive	Greedy	Rules
0.5	13.58	13.58	13.44
1	9.35	9.35	8.38
5	4.35	4.37	3.44
60	3.29	2.99	3.44

VI. CONCLUSIONS AND FUTURE WORK

Unfortunately, using a game-theoretic approach to create overlay network topologies is intractable. Even a greedy search approach is computationally expensive as the size of the network grows. To solve this problem, a machine learning approach was used to characterize the attributes of nodes that logical links were made toward. Using these attributes, a set of rules were learned that were used to decide whether to create a logical link toward a node or not. The resulting topologies were compared against those formed through the exhaustive and greedy search approaches. These comparisons show that the rule-based approach creates similar topologies to the exhaustive and greedy search approaches in most

TABLE X

NUMBER OF EDGES FOR 100 NODE NETWORKS

α	Greedy	Rules
0.5	3399.0	3399.0
1	2438.0	2332.7
5	99.0	99.1
60	99.0	99.1
200	99.0	99.1
400	99.0	99.4

TABLE XI

CHARACTERISTIC PATH LENGTH FOR 100 NODE NETWORKS

α	Greedy	Rules
0.5	1.31	1.31
1	1.50	1.52
5	1.98	2.71
60	1.98	2.71
200	3.67	2.70
400	4.35	2.68

TABLE XII

SPECTRAL RADIUS FOR 100 NODE NETWORKS

α	Greedy	Rules
0.5	68.36	68.36
1	49.27	47.02
5	9.94	7.50
60	9.94	7.51
200	6.52	7.53
400	5.46	7.54

cases.

An area of future work, would be considering a more realistic underlay topology, where the distance between nodes in the underlay are not constant. Other interesting areas of research, would be exploring the effects of heterogeneous nodes. What type of effect would nodes with different α and $maxDegree$ values have on the topology? How would the rules have to be changed to handle this? It would also be interesting to investigate this approach in a dynamic environment, where traffic-demand between nodes is constantly changing. How could online learning be used to adjust the rules in a dynamic environment? All of these issues would need to be investigated before a real-world implementation and deployment could be realized.

REFERENCES

[1] S. Savage, A. Collins, E. Hoffman, J. Snell, and T. Anderson, "The end-to-end effects of internet path selection," in *SIGCOMM '99: Proceedings of the conference on Applications, technologies, architectures, and protocols for computer communication*. ACM Press, 1999.

[2] H. Zhang, J. F. Kurose, and D. Towsley, "Can an overlay compensate for a careless underlay?" in *INFOCOM 2006: Twenty-fifth Annual Joint Conference of the IEEE Computer and Communications Societies*, 2006.

[3] A. Fabrikant, A. Luthra, E. Maneva, C. H. Papadimitriou, and S. Shenker, "On a network creation game," in *PODC 2003: Proceedings of the Twenty-second Annual Symposium on Principles of Distributed Computing*, 2003.

[4] B.-G. Chun, R. Fonseca, I. Stoica, and J. Kubiatowicz, "Characterizing selfishly constructed overlay routing networks," in *INFOCOM 2004: Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies*, 2004.

[5] T. Moscibroda, S. Schmid, and R. Wattenhofer, "On the topologies formed by selfish peers," in *Proceedings of the Twenty-fifth Annual ACM Symposium on Principles of Distributed Computing*, 2006.

[6] N. Christin and J. Chuang, "A cost-based analysis of overlay routing geometries," in *INFOCOM 2005: Twenty-fourth Annual Joint Conference of the IEEE Computer and Communications Societies*, 2005.

[7] E. Anshelevich, A. Dasgupta, E. Tardos, and T. Wexler, "Near-optimal network design with selfish agents," in *Thirty-fifth Annual ACM Symposium on Theory of Computing*, 2003.

[8] S. Albers, S. Eilts, E. Even-Dar, Y. Mansour, and L. Roditty, "On Nash equilibria for a network creation game," in *Proceedings of the Seventeenth Annual ACM-SIAM Symposium on Discrete Algorithms*, 2006.

[9] B. D. McBride and C. Scoglio, "Characterizing traffic demand aware overlay routing network topologies," in *International Workshop on High Performance Switching and Routing*, 2007.

[10] B. D. McBride, "Characterizing traffic-aware overlay topologies: A machine learning approach," Master's thesis, Kansas State University, 2007.

[11] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman and Company, 1979.

[12] T. M. Mitchell, *Machine Learning*. WCB/McGraw-Hill, 1997.

[13] I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques*, 2nd ed. Morgan Kaufmann Publishers, 2005.