DYNAMIC ARCHITECTURE OVERLAY WITH OPTIMAL THROUGHPUT MULTIPATH ROUTING

[1]K. Samson Paul, Assistant Professor
[2]Dr. C. Mohammed Gulzar, Associate Professor
[3]K. Lahari , Assistant Professor
DEPARTMENT OF CSE
Dr. K V SUBBA REDDY INSTITUTE OF TECHNOLOGY, DUPADU, KURNOOL.

ABSTRACT

Optimal routing in networks where some legacy nodes are replaced with overlay nodes. While the legacy nodes perform only forwarding on pre-specified paths, the overlay nodes are able to dynamically route packets. Dynamic backpressure is known to be an optimal routing policy, but it typically requires a homogeneous network, where all nodes participate in control decisions. Instead, we assume that only a subset of the nodes is controllable; these nodes form a network overlay within the legacy network.

Keywords: multipath routing; backpressure routing

I. INTRODUCTION

Energetic backpressure is known to be an optimal routing policy, but it characteristically requires a similar network, where all nodes contribute in control decisions. In its place, we shoulder that only a subset of the nodes is governable; these nodes form a net overlay within the inheritance network. The high-quality of the overlay nodes is shown to regulate the throughput region of the network

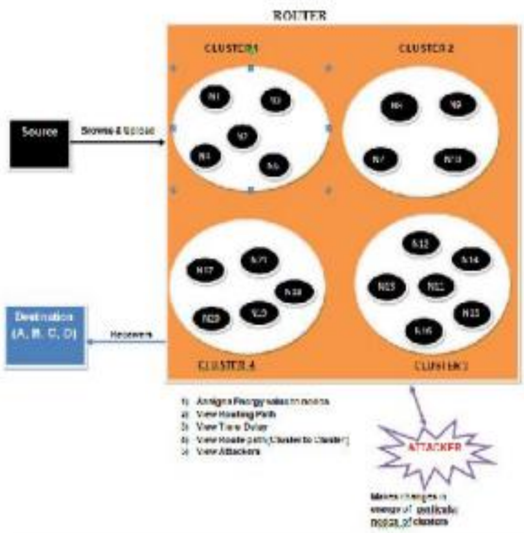
Optimal Routing Design provides the tools and techniques, learned through years of experience with network design and deployment, to build a large-scale or scalable I Prouted network. Optimal routing in networks where some legacy nodes are replaced with overlay nodes. While the legacy nodes perform only forwarding on pre-specified paths, the overlay nodes are able to dynamically route packets. Dynamic backpressure is known to be an optimal routing policy. Backpressure routing is an algorithm for dynamically routing traffic over a multi-hop network by using congestion gradients but it typically requires a homogeneous network, where all nodes participate in control decisions. Instead, let us consider only a subset of the nodes are controllable, these nodes form a network overlay within the legacy network. Backpressure routing is designed to make decisions that (roughly) minimize the sum of squares of queue backlogs in the network from one time slot to the next. It is important to note that the backpressure algorithm does not use any pre-specified paths. Paths are learned dynamically, and may be different for different packets. Delay can be very large, particularly when the system is lightly loaded so that there is not enough pressure to push data towards the destination. As an example, suppose one packet enters the network, and nothing else ever enters. This packet may take a loopy walk through the network and never arrive at its destination because no pressure gradients build up. This does not contradict the throughput optimality or stability properties of backpressure because the network has at most one packet at any time and hence is trivially stable.

II. LITERATURE SURVEY

every node has to preserve a detached queue for each product in the network and only one file is attended at a time. The backpressure routing algorithm may direct some packets lengthwise very long routes. In this paper, we present solutions to both and recover the delay presentation of the back-pressure algorithm. One of the optional solutions toreduce the difficulty of the queuing data constructions is to be preserved at everynode

Peer-to-Peer overlay network is asubmission model without since underlying network topology. But there exists discrepancy problem amongst peer-to-peer overlay network and physical network topology. This origininept transmission or routing between peers in the peer-to-peer overlay network. On the other hand, the status quo will have serious delay in real-time service, for example streaming service. Therefore, in this paper we put forward an upgrading instrument based on physical network hop information to lessen the transmission cycles to alter the arrangement of peer-to-peer overlay network vigorously

III. SYSTEM ANALYSIS AND MODEL



We model the network as a directed graph G = (N , E), where N is the set of nodes in the network and E is the set of edges. We assume that the underlay network provides a fixed realization for shortest-path routes between all pairs of nodes, and that uncontrollable nodes will forward traffic only along the given shortest-path routes. Further, we assume that only one path is provided between each pair of nodes. Let P SP ab be the shortest path from a to b, and let P SP = (P SP ab), for all pairs a, b ∈ N , be the set of all shortest paths provided by the underlay network. If (i, j) is a link in G, then we assume that the single hop path is available, i.e. P SP ij ∈ PSP. Whenever a packet enters a forwarding node, the node inspects the corresponding routing table and sends the packet towards the pre-specified path. Therefore, the performance of the system depends on the available set of paths P SP . Optimal substructure is assumed for shortest-paths, such that if shortest-path P SP ac from node a to c includes node b, then path P SP ac includes shortestpaths P SP ab , from a to b, and P SP bc , from b to c. This optimal substructure is consistent with shortest-paths in OSPF, a widely used routing protocol based on Dijkstra's shortestpath algorithm [8], where OSPF allows for the use of lowest next-hop router ID as a method for choosing between multiple paths of equal length.

Next, we consider the subset of nodes V ⊆ N , called overlay or controllable nodes, which can bifurcate traffic throughput different routes. Intuitively, these nodes can improve throughput performance by generating new paths and enabling multipath routing. The remaining uncontrollable nodes u ∈ N \ V provide only shortest-path forwarding in the underlay network, with an exception that any uncontrollable node u can bifurcate all traffic that originates at u; this may occur, for example, in the source applications at uncontrollable nodes, or in a shim-layer between the networklayer and application-layer. Without such an exception, all sources may be required to be controllable nodes. Controllable nodes can increase the achievable throughput region by admitting new paths to the network as concatenations of existing paths from shortest-path routing. A 2-concatenation of shortest-paths P SP av and P SP vb is an acyclic path from a to b, Pab, where v ∈ V is a controllable node and v is the only node shared between shortest-paths P SP av and P SP vb . Note that a 2-concatenation of acyclic paths will always be acyclic, as we only allow the concatenated paths to share the overlay node v at which concatenation is performed. An n-concatenation is then the concatenation of n shortest-paths at n − 1 controllable nodes, performed as a succession of (n−1) 2-concatenations, and therefore acyclic. Consider the set of paths P(V), which contains all underlay paths P SP as well as all possible n-concatenations of these paths at the controllable nodes V. We will see that this set P(V) plays a role in the achievability of the throughput region.

IV. METHODOLOGY

We consider two problem areas for control of heterogeneous networks. First, we develop algorithms for choosing the placement of controllable nodes, where our goal here is to allocate the minimum number of controllable nodes such that the full network stability region is available.

Second, given any subset of nodes that are controllable, we also wish to develop an optimal routing policy that operates solely on these nodes

Our solutions for the first and second problem areas are complementary, in the sense that they can be used together to solve the joint problem of providing maximum throughput when only a subset of nodes are controllable. However, our solutions can also be used in isolation; our node placement algorithm can be used with other control policies, and our BP extensions can yield maximal stability with any overlay node placement and legacy single-path routing.

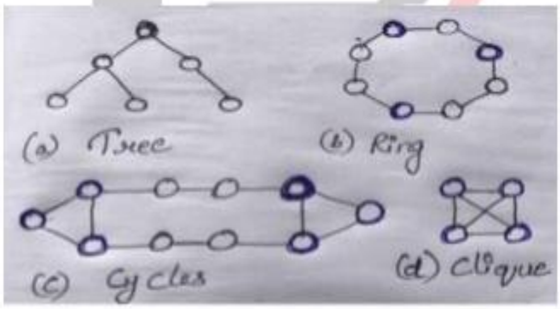


FiG.1.Minimum node placement required to produce max throughput for several normal scenarios, where manageable nodes are bolded in dark shade. (a) No controllable nodes on trees. (b) Exactly 3 controllable nodes on a ring . (c) At least 3 controllable nodes on every cycle. (d) All nodes must be controllable on a clique.

PLACEMENT OF OVERLAY NODES

We would like to place controllable nodes to solve P1, but the constraint ΛG(V) = ΛG(N) is difficult to evaluate directly. A simple implementation for P1 can use the fact that ΛG is a convex polytope, choosing the minimum number of controllable nodes to satisfy all points in the throughput region, as

$$V_2^* = \min_{V \subseteq N} |V| \quad \text{s.t.} \quad \lambda^{(i)} \in \Lambda_G(V) \,, \forall \lambda^{(i)} \in \Lambda_G \,,$$

where λ (i) enumerates all extreme points of ΛG. It is clear that P2 is equivalent to P1, although enumerating all extreme points may be impractical. Instead of evaluating P2, we propose a surrogate condition that is easier to evaluate while still leading to the same optimal solution. Recall that the set of paths P(V) includes all underlay paths P SP and all n-concatenations (for any n) of these paths at controllable nodes V. Let PG be the set of all acyclic paths between all pairs of nodes in G. A first observation is that P(N) = PG. This holds by the assumption that all 1-hop paths are included in the set P SP, and since all nodes are controllable we can produce any path in G as a concatenation of 1-hop paths. Next, we define an important condition.

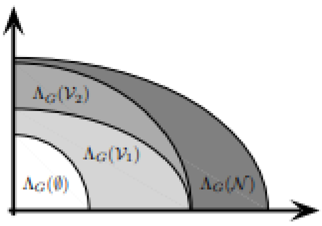


Fig: Projection of throughput regions ΛG(·) for sets of overlay nodes V1, V2 : V1 ⊆ V2 ⊆ N , indicating subset relationship as described

4.1 Overlay Node Placement Algorithm

We design an algorithm to choose the placement of overlay nodes V ⊆ N on a given graph G = (N , E) such that the choice of overlay nodes is sufficient to satisfy the full throughput region of the network, i.e. ΛG(V) = ΛG(N). At the end of this section we will show that the proposed algorithm optimally solves P3. The algorithm consists of three phases: (1) removal of degree-1 nodes; (2) constraint pruning; and (3) overlay node placement. These phases are explained below, while each tep is supported by a related claim which will help proving the optimality of the algorithm.

Phase 1: Remove Degree-1 Nodes. An attached tree is a tree that is connected to the rest of graph G by only a single edge. An intuitive observation is that the throughput region does not increase by installing controllable nodes on attached trees. Thus, at this preparatory phase, we remove all attached trees by removing degree-1 nodes recursively, as follows. Start with original graph G = (N , E), and initialize N ′ := N and E ′ := E. While there exists any node n ∈ N ′ such that degree(n) = 1, set N ′ := N ′ \ n and set E ′ := E ′ \ e, where e is the only edge that connects to node n. Repeat until no degree-1 nodes remain. All remaining nodes have a degree of at least 2, thus all attached trees have been removed. The graph that remains is G ′ = (N ′ , E ′).

Lemma 1: Suppose that placement V satisfies the all-paths condition (C.1), and n ∈ V lies on an attached tree. Then V \ n also satisfies the all-paths condition. Proof. To prove P(V) = P(V \ n), it is enough to show that for any pair a, b ∈ N , the acyclic path Pab ∈ P(V) can be formed without concatenating paths at n. Note, that if n /∈ Pab, then the requested is immediately obtained. Thus, we are free to assume that additionally to lying on an attached tree, n is also on the path Pab. We study four
cases: 1. Nodes a and b are both on the same attached tree: There is only one path from a to b, and this is the shortest path. Thus, Pab ∈ PSP = P(∅) ⊆ P(V \ n).

Overlay Node Placement Algorithm

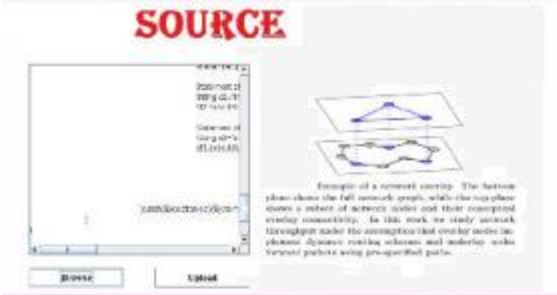
Phase 1: Recursively remove all degree-1 nodes $\mathcal{N}_1$ and associated edges $\mathcal{E}_1$ from graph G, until no degree-1 nodes remain. The remaining graph is $G' = \{\mathcal{N}', \mathcal{E}'\}$, where $\mathcal{N}' = \mathcal{N} \setminus \mathcal{N}_1$ and $\mathcal{E}' = \mathcal{E} \setminus \mathcal{E}_1$. This removes all attached trees from G.

Phase 2: Consider the *destination tree* D_n for each node $n \in \mathcal{N}'$, and consider the degree of all nodes $b \in \mathcal{N}' \setminus n$ on tree D_n. If the degree of b on D_n is less than the degree of b on G', then prune destination tree D_n at node b by removing all edges to children of node b on D_n, and remove any nodes and edges that become disconnected from n. The remaining subgraph is the *pruned tree* D'_n.

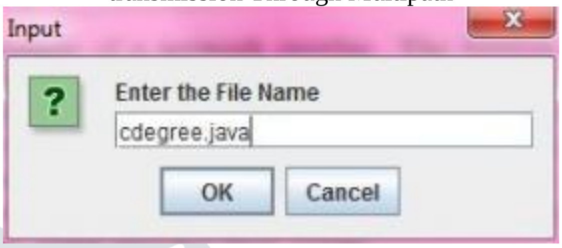
Phase 3: Solve P4, and place an overlay node at each node n where the solution to P4 has $v_n = 1$.

V. RESULTS

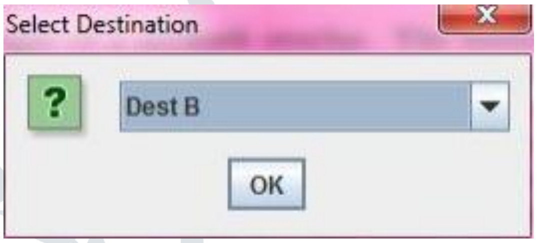
Data is Transfer from Router to Nodes



Source: Efficient overlaynodes selection for Data transmission Through Multipath



Network



Source Input File Name

Select Destination

VI. CONCLUSION

We suggest an essential and satisfactory complaint for the edge node settlement to empower the full multi product quantity section. Created on this ailment, we create an algorithm for optimum well-behaved node situation. We create the algorithm on huge haphazard graphs to illustration that identical often a small numeral of intellectual nodes be sufficient for full throughput. To end, we advise an energetic routing program to be effected in a network connection, that exhibits grander show in terms of both output and deferral.

REFERENCES

[1]. L. Tassiulas and A. Ephremides, “Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multi hop radio networks,” IEEE Trans. Autom. Control, vol. 37, no. 12, pp. 1936–1948, Dec. 1992.

[2]. L. Bui, R. Srikant, and A. Stolyar, “Novel architectures and algorithms for delay reduction in back-pressure scheduling and routing,” in Proc. IEEE INFOCOM, Apr. 2009, pp. 2936–2940.

[3] W. Khan, L. B. Le, and E. Modiano, "Autonomous routing algorithms for networks with wide-spread failures," in Proc. IEEE MILCOM, Oct. 2009, pp. 1–6. [Online]. Available: http://ieeexplore.ieee.org/stamp/ stamp.jsp?arnumber=5379792

[4] D. Andersen, H. Balakrishnan, F. Kaashoek, and R. Morris. Resilient overlay networks. In Proc. ACM SOSP, Oct. 2001.

[5] L. Bui, R. Srikant, and A. Stolyar. Novel architectures and algorithms for delay reduction in back-pressure scheduling and routing. In Proc. IEEE INFOCOM, April 2009.

[6] B. Fortz and M. Thorup.Internet traffic engineering by optimizing ospf weights.In IEEE INFOCOM, 2000.

[7] J. Han, D. Watson, and F. Jahanian. Topology aware overlay networks. In Proc. IEEE INFOCOM, March 2005.

[8] W. Khan, L. B. Le, and E. Modiano. Autonomous routing algorithms for networks with wide-spread failures.In Proc. IEEE MILCOM, Oct. 2009.

[9] M. J. Neely, E. Modiano, and C. E. Rohrs. Dynamic power allocation and routing for time varying wireless networks.In Proc. IEEE INFOCOM, April 2003.

[10] M. E. J Newman. Networks: An Introduction. Oxford University Press, Inc., New York, NY, USA, 2010.

[11] L. L. Peterson and B. S. Davie. Computer Networks: A Systems Approach. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 4th edition, 2007.

[12] J. Ryu, L. Ying, and S. Shakkottai. Back-pressure routing for intermittently connected networks. In Proc. IEEE INFOCOM, March 2010.

[13] L. Tassiulas and A. Ephremides. Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks. IEEE Trans. Auto. Control, pages 1936– 1948, Dec. 1992.