An overlay architecture for throughput optimal multipath routing

An Overlay Architecture for Throughput Optimal Multipath Routing

Nathaniel M. Jones
MIT Lincoln Laboratory
Lexington, MA

Georgios S. Paschos
MIT, Cambridge, MA &
CERTH-ITI, Greece

Brooke Shrader
MIT Lincoln Laboratory
Lexington, MA

Eytan Modiano
MIT, Cambridge, MA

ABSTRACT

Legacy networks are often designed to operate with simple single-path routing, like shortest-path, which is known to be throughput suboptimal. On the other hand, previously proposed throughput optimal policies (i.e., backpressure) require every device in the network to make dynamic routing decisions. In this work, we study an overlay architecture for dynamic routing such that only a subset of devices (overlay nodes) need to make dynamic routing decisions. We determine the essential collection of nodes that must bifurcate traffic for achieving the maximum multicommodity network throughput. We apply our optimal node placement algorithm to several graphs and the results show that a small fraction of overlay nodes is sufficient for achieving maximum throughput. Finally, we propose a heuristic policy (OBP), which dynamically controls traffic bifurcations at overlay nodes. In all studied simulation scenarios, OBP not only achieves full throughput, but also reduces delay in comparison to the throughput optimal backpressure routing.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design - Packet-Switching Networks

Keywords

overlay; multipath routing; backpressure routing

The Lincoln Laboratory portion of this work was sponsored by the Department of the Air Force under Air Force Contract FA8721-05-C-0002. Opinions, interpretations, conclusions and recommendations are those of the authors and are not necessarily endorsed by the United States Government. This work was also supported by NSF grant CNS-0915988, ONR grant N00014-12-1-0064, and ARO MURI grant W911NF-08-1-0238. The work of G. Paschos was supported in part by the WiNC project of the Action: Supporting Postdoctoral Researchers, funded by national and Community funds (European Social Fund).

MobiHoc'14, Aug 11-14, 2014, Philadelphia, PA, USA.
ACM 978-1-4503-2620-9/14/08 ...$15.00.
http://dx.doi.org/10.1145/2632951.2632957.

1. INTRODUCTION

We study 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 are controllable; these nodes form a network overlay within the legacy network. The choice of the overlay nodes is shown to determine the *throughput region* of the network.

A first finding is that ring networks require exactly 3 controllable (overlay) nodes to enable the same throughput region as when all nodes are controllable, independent of the total number of nodes in the network. Motivated by this, we develop an algorithm for choosing the minimum number of controllable nodes required to enable the full throughput region. We evaluate our algorithm on several classes of regular and random graphs. In the case of random networks with a power-law degree distribution, which is a common model for the Internet, we find that fewer than 80 out of 1000 nodes are required to be controllable to enable the full throughput region.

Since standard backpressure routing cannot be directly applied to the overlay setting, we develop a heuristic extension to backpressure routing that determines how to route packets between overlay nodes. Simulation results confirm that maximum throughput can be attained with our policy in several scenarios, when only a fraction of legacy nodes are replaced by controllable nodes. Moreover, we observe reduced delay relative to the case where all nodes are controllable and operate under backpressure routing.

1.1 Motivation and Related Work

Backpressure (BP) routing, first proposed in [10], is a throughput optimal routing policy that has been studied for decades. Its strength lies in discovering multipath routes and utilizing them optimally without knowledge of the network parameters, such as arrival rates, link capacities, mobility, fading, etc. Nevertheless, the adoption of this routing policy has not been embraced for general use on the Internet. This is due, in part, to an inability of backpressure routing to coexist with legacy routing protocols. With few exceptions, backpressure routing has been studied in homogeneous networks, where all nodes are dynamically controllable and implement the backpressure policy across

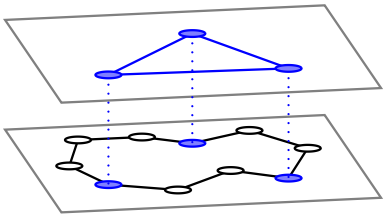


Figure 1: Example of a network overlay. The bottom plane shows the full network graph, while the top plane shows a subset of network nodes and their conceptual overlay connectivity. In this work we study network throughput under the assumption that overlay nodes implement dynamic routing schemes and underlay nodes forward packets using pre-specified paths.

all nodes uniformly. As will be shown, backpressure routing — as proposed in [10] — is suboptimal when applied only to a subset of nodes in the network.

Techniques to provide throughput-optimal multipath routing have been explored in various contexts. The work in [3] considers the problem of setting link weights provided to the Open Shortest Path First (OSPF) routing protocol such that, when coupled with bifurcating traffic equally among shortest paths, the network achieves throughput equal to the optimal multicommodity flow. The authors of [11] use an entropy maximization framework to develop a new throughput-optimal link state routing protocol where each router intelligently bifurcates traffic for each destination among its outgoing links. These techniques all require centralized control, universal adoption by all network nodes, or both; thus none of these techniques could provide incremental deployment of throughput optimal routing to wireless networks. Moreover, these techniques cannot be used in conjunction with throughput optimal dynamic control schemes, such as backpressure.

We would like to enable new network control policies to be deployed in existing networks, alongside legacy nodes that are unaware of the new control policies. There are many reasons to integrate controllable nodes into heterogeneous networks in a gradual manner, not the least of which is the financial cost of replacing all nodes at once. Other reasons include a need to maintain compatibility with current applications and special purpose hardware, a lack of ownership to decommission legacy equipment, and a lack of administrative privilege to modify existing software.

Conceptually, we model controllable nodes as operating in a network overlay on top of a legacy network. Network overlays are frequently used to deploy new communication architectures in legacy networks [8]. To accomplish this, messages from the new technology are encapsulated in the legacy format, allowing the two methods to coexist in the legacy network. Nodes making use of the new communication methods are then connected in a conceptual network overlay that operates on top of the legacy network, as shown in Figure 1.

Several works have considered the use of network overlays to improve routing in the Internet. The work in [1] proposes resilient overlay networks (RON) to find paths around network outages on a faster timescale than BGP. Similarly, [4] proposed a method for choosing placement of overlay nodes to improve path diversity in overlay routes. While both of the preceding works show that their strategies choose high quality single-path routes, we go further and identify multipath routes that offer maximum throughput.

Delay reduction for BP routing has been studied in a variety of scenarios. While multipath routes are required to support the full throughput region, the exploratory phase of BP can lead to large queues when the offered load is low and single-path routes would suffice. In [6], a hybrid policy combining BP with shortest-path routing is proposed, where flows are biased towards shortest-path routes, yet still support the full throughput region. This hybrid policy is extended in [5] to also include digital fountain codes, and shown to achieve good end-to-end delay performance in the presence of random link failures. The work in [12] develops a policy that achieves a similar shortest-path result by minimizing the average hop count used by flows. In a scenario with multiple clusters that are intermittently connected, [9] combines BP with source routing in a network overlay model to separate the queue dynamics of intra-cluster traffic from longer inter-cluster delays. The work in [2] applies shadow queues to allow the use of per-neighbor FIFO queues instead of per-commodity queues, as is typical with differential backlog routing, and finds that this can improve network delay. These prior works assume a homogeneous scenario where all nodes use the same control policy and thus differ fundamentally from our approach.

1.2 Problem Statement and Contributions

Given a graph G with nodes $\mathcal{N}$ supporting shortest-path routes between each pair of nodes, we wish to identify a minimal set of controllable nodes $\mathcal{V} \subseteq \mathcal{N}$ such that if only these nodes are allowed to bifurcate traffic, maximum throughput can be achieved. Given any subset of nodes that are controllable, we also wish to develop an optimal routing policy that operates solely on these nodes.

Ideally, we would like to solve P1,

$$V_1^* = \min_{\mathcal{V} \subseteq \mathcal{N}} |\mathcal{V}| \quad \text{s.t.} \quad \Lambda_G(\mathcal{V}) = \Lambda_G(\mathcal{N}) \,, \qquad \text{(P1)}$$

where $\Lambda_G(\mathcal{V})$ is the throughput region (i.e., the set of multicommodity arrival rate vectors that can be stably supported by the network) for graph G when only nodes $\mathcal{V}$ are controllable, while $\Lambda_G(\mathcal{N})$ is the throughput region when all nodes are controllable. Note that comparing throughput regions directly can be difficult, so instead we identify a condition that is necessary and sufficient to guarantee the full throughput region, and then we search for the minimal $\mathcal{V}$ that satisfies this condition. Second, we develop a modified BP routing policy and show through simulation that if applied on nodes $\mathcal{V}$, it stabilizes any arrival vector in $\Lambda_G(\mathcal{V})$. Our solutions for the first and second problems are complementary, in the sense that they can be used together to solve the joint problem. However, our node placement algorithm can be used with other policies, and our backpressure policy works under any overlay node placement.

Our contributions are summarized below.

- We formulate the problem of placing the minimum number of overlay (controllable) nodes in a legacy network in order to achieve the full multicommodity throughput region and provide an efficient placement algorithm.

- We apply our placement algorithm to several scenarios of interest including regular and random graphs, show-

ing that in some cases, only a small fraction of overlay nodes is sufficient for maximum throughput.

- We propose a dynamic control policy — OBP — as a modification of BP for use at overlay nodes. We show via simulation that OBP can outperform BP when limited to control at overlay nodes, and that OBP also has better delay performance compared to BP with control at all nodes.

2. MODEL

We model the network as a directed graph $G = (\mathcal{N}, \mathcal{E})$, where $\mathcal{N}$ is the set of nodes in the network and $\mathcal{E}$ 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_{ab}^{SP} be the shortest path from a to b, and let $\mathcal{P}^{\text{SP}} = (P_{ab}^{\text{SP}})$, for all pairs $a, b \in \mathcal{N}$, 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. $P_{ij}^{\text{SP}} \in \mathcal{P}^{\text{SP}}$. 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 $\mathcal{P}^{\text{SP}}$. *Optimal substructure* is assumed for shortest-paths, such that if shortest-path P_{ac}^{SP} from node a to c includes node b, then path P_{ac}^{SP} includes shortest-paths P_{ab}^{SP}, from a to b, and P_{bc}^{SP}, from b to c. This optimal substructure is consistent with shortest-paths in OSPF, a widely used routing protocol based on Dijkstra's shortest-path 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 $\mathcal{V} \subseteq \mathcal{N}$, 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 $u \in \mathcal{N} \setminus \mathcal{V}$ 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 network-layer 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_{av}^{SP} and P_{vb}^{SP} is an acyclic path from a to b, P_{ab}, where $v \in \mathcal{V}$ is a controllable node and v is the only node shared between shortest-paths P_{av}^{SP} and P_{vb}^{SP}. 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 $\mathcal{P}(\mathcal{V})$, which contains all underlay paths $\mathcal{P}^{\text{SP}}$ as well as all possible n-concatenations of these paths at the controllable nodes $\mathcal{V}$. We will see that this set $\mathcal{P}(\mathcal{V})$ plays a role in the achievability of the throughput region.

3. THROUGHPUT REGION

The *throughput region* $\Lambda_G(\mathcal{V})$ is the set of all arrival rates that can be achieved by any policy implemented at controllable nodes $\mathcal{V}$ on graph G. For the case where all nodes are controllable, i.e., $\mathcal{V} = \mathcal{N}$, the throughput region equals the stability region of graph G. This section characterizes this region for a given set of paths $\mathcal{P}(\mathcal{V})$.

Packets destined for node c are called *commodity* c packets. Let λ_a^c be the rate of exogenous arrivals at node a for commodity c, and let $\boldsymbol{\lambda} = (\lambda_a^c)$ be the multicommodity arrival rate vector for all sources a and commodities c. Let $f_{ij}^{ab,c}$ be the *edge-flow* for commodity c on edge (i, j) along the shortest-path from node a to b. Flow for a path is allowed only on the edges along that path, i.e. $f_{ij}^{ab,c} = 0$ unless $(i, j) \in P_{ab}^{\text{SP}}$. Let $\bar{f}_{ab}^c$ be *path-flow* for commodity c along shortest-path P_{ab}^{SP}, from node a to b. Decision variable $v_i = 1$ if node i is controllable, and $v_i = 0$ otherwise, for all nodes $i \in \mathcal{N}$. The capacity of edge (i, j) is R_{ij}. The controllable throughput region $\Lambda_G(\mathcal{V})$ is then the set of all arrival rate vectors (λ_a^c) such that Eqns. (1-6) can be satisfied.

Flow Conservation:

$$\lambda_v^c = \sum_{b \in \{c, \mathcal{V} \setminus v\}} \bar{f}_{vb}^c - \sum_{d \in \mathcal{V} \setminus v} \bar{f}_{dv}^c \ , \quad \forall\, v \in \mathcal{V}, c \in \mathcal{N} \setminus v \tag{1}$$

$$\lambda_u^c = \sum_{b \in \{c, \mathcal{V}\}} \bar{f}_{ub}^c \ , \quad \forall\, u \in \mathcal{N} \setminus \mathcal{V}, c \in \mathcal{N} \setminus u \tag{2}$$

Path Constraint:

$$\bar{f}_{ab}^c = f_{ij}^{ab,c} \ , \quad \forall\, (i, j) \in P_{ab}^{\text{SP}},\ \forall\, a, b, c \in \mathcal{N} \tag{3}$$

Overlay Neighbor Constraints:

$$f_{ij}^{ab,c} \leq (1 - v_i) R_{ij} \ , \quad \forall\, (i, j) \in P_{ab}^{\text{SP}},\ a \neq i,\ \forall c \in \mathcal{N} \tag{4}$$

$$f_{ij}^{ab,c} \leq (1 - v_j) R_{ij} \ , \quad \forall\, (i, j) \in P_{ab}^{\text{SP}},\ b \neq j,\ \forall c \in \mathcal{N} \tag{5}$$

Edge Rate Constraint:

$$\sum_{a,b,c} f_{ij}^{ab,c} \leq R_{ij} \ , \quad \forall\, (i, j) \in \mathcal{E} \tag{6}$$

Eqn. (1) represents flow conservation of commodity c packets at controllable node v. Here, exogenous arrivals at node v equal network departures minus (endogenous) network arrivals at v. Similarly, Eqn. (2) represents flow conservation for exogenous arrivals at uncontrollable nodes. The exogenous arrivals for commodity c at uncontrollable node u are equal to network departures on the shortest-path to destination c plus network departures along shortest-paths to controllable nodes. This is the special case where uncontrollable node u is a source, in that u can dynamically route exogenous arrivals but not endogenous network arrivals. Eqn. (3) is a path constraint for each commodity c along the shortest-path from node a to node b, where the path-flow equals the edge-flow for each edge along path P_{ab}^{SP}. Eqns. (4-5) force edge-flow $f_{ij}^{ab,c} = 0$ if node i or j is a controllable node intermediate to path P_{ab}^{SP}, i.e., for $i \neq a$ and $j \neq b$, as such paths remove routing ability from intermediate controllable nodes. Eqns. (4-5) are necessary to allow for dynamic choice of controllable nodes, and are redundant with Eqn. (6) when nodes i and j both are uncontrollable. Finally, Eqn. (6) is an edge rate constraint for every edge (i, j), such that total flow over an edge is upper bounded by the edge capacity.

If there are no controllable nodes, i.e. $\mathcal{V} = \emptyset$, then Eqn. (2) simplifies to

$$\lambda_a^c = \bar{f}_{ac}^c \ , \quad \forall\, a, c \in \mathcal{N}, a \neq c \ , \tag{7}$$

where Eqns. (4-5) can be ignored as they are always redundant with Eqn. (6). The throughput region without controllable nodes, $\Lambda_G^{\text{SP}} \equiv \Lambda_G(\emptyset)$, is thus limited to the set of arrival rate vectors $\boldsymbol{\lambda}$ such that Eqns. (7), (3) and (6) are satisfied. Indeed, these equations specify the shortest-path formulation for the throughput region on graph G, defined as $\Lambda_G^{\text{SP}} \equiv \Lambda_G(\emptyset)$.

If all nodes are controllable, i.e. $\mathcal{V} = \mathcal{N}$, then there are no constraints from underlay paths and all dynamic routing decisions are allowed. Eqns. (1) and (6) simplify to

$$\lambda_a^c = \sum_{b:(a,b)\in\mathcal{E}} \bar{f}_{ab}^c - \sum_{d:(d,a)\in\mathcal{E}} \bar{f}_{da}^c \ , \quad \forall\, a, c \in \mathcal{N}, a \neq c \ , \tag{8}$$

$$\sum_c \bar{f}_{ab}^c \leq R_{ab} \ , \quad \forall\, (a,b) \in \mathcal{E} \ . \tag{9}$$

There are no uncontrollable nodes here, so Eqn. (2) is unused, and Eqns. (3), (4), and (5) are redundant with Eqns. (8) and (9). The full region $\Lambda_G \equiv \Lambda_G(\mathcal{N})$ is then defined as the set of arrival rate vectors $\boldsymbol{\lambda}$ that satisfy Eqns. (8-9). This is the largest region supported by network G.

Any work-conserving policy with shortest-path routing can support the region $\Lambda_G(\emptyset)$, while backpressure routing is known to support the full region $\Lambda_G(\mathcal{N})$. However, how to achieve the heterogeneous region $\Lambda_G(\mathcal{V})$ with a dynamic routing policy is not generally known. For heterogeneous networks, converting an uncontrollable node u into a controllable node v relaxes the constraints for node u from Eqn. (2) into Eqn. (1). Note that when node v becomes controllable, the overlay neighbor constraints from Eqns. (4-5) become active.

Recall that we assume optimal substructure for shortest-paths. We use this structure to find an additional property about the throughput region. Any path P_{ab}^{SP} that passes through a controllable node v can be split into two sub-paths P_{av}^{SP} and P_{vb}^{SP}, where optimal substructure guarantees that both sub-paths are in the set of underlay routes $\mathcal{P}^{\text{SP}}$. Node v can then concatenate these sub-paths to form the original path P_{ab}^{SP}. Therefore, if there exists a flow decomposition of $\boldsymbol{\lambda}$ that uses path P_{ab}^{SP}, then there is also a flow decomposition that uses sub-paths P_{av}^{SP} and P_{vb}^{SP}. Thus, with shortest-path routing, adding controllable nodes can allow the throughput region to grow, but never causes the region to shrink. This implies a subset relationship in the throughput region with shortest-path underlay routing, such that for any overlay node sets $\mathcal{V}_1, \mathcal{V}_2 : \mathcal{V}_1 \subseteq \mathcal{V}_2 \subseteq \mathcal{N}$,

$$\Lambda_G^{\text{SP}} \equiv \Lambda_G(\emptyset) \subseteq \Lambda_G(\mathcal{V}_1) \subseteq \Lambda_G(\mathcal{V}_2) \subseteq \Lambda_G(\mathcal{N}) \equiv \Lambda_G. \tag{10}$$

4. PLACEMENT OF OVERLAY NODES

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

$$V_2^* = \min_{\mathcal{V}\subseteq\mathcal{N}} |\mathcal{V}| \quad \text{s.t.} \quad \boldsymbol{\lambda}^{(i)} \in \Lambda_G(\mathcal{V}) \ , \forall \boldsymbol{\lambda}^{(i)} \in \Lambda_G \ , \tag{P2}$$

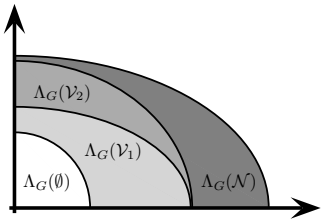


Figure 2: Projection of throughput regions $\Lambda_G(\cdot)$ for sets of overlay nodes $\mathcal{V}_1, \mathcal{V}_2 : \mathcal{V}_1 \subseteq \mathcal{V}_2 \subseteq \mathcal{N}$, indicating subset relationship as described in Eqn. (10).

where $\boldsymbol{\lambda}^{(i)}$ enumerates all extreme points of Λ_G. 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 $\mathcal{P}(\mathcal{V})$ includes all underlay paths $\mathcal{P}^{\text{SP}}$ and all n-concatenations (for any n) of these paths at controllable nodes $\mathcal{V}$. Let $\mathcal{P}_G$ be the set of all acyclic paths between all pairs of nodes in G. A first observation is that $\mathcal{P}(\mathcal{N}) = \mathcal{P}_G$. This holds by the assumption that all 1-hop paths are included in the set $\mathcal{P}^{\text{SP}}$, 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.

Condition C.1 (All-paths) *A set of controllable nodes $\mathcal{V}$ is said to satisfy the* all-paths condition *if $\mathcal{P}(\mathcal{V}) = \mathcal{P}_G$.*

The condition requires the formation of all acyclic paths in a network. Since some of the paths are already given (in our paper $\mathcal{P}^{\text{SP}}$), to satisfy the condition, a set of nodes $\mathcal{V}$ must enable all missing paths $\mathcal{P}_G \setminus \mathcal{P}^{\text{SP}}$ by path concatenations. The following result establishes that this condition is necessary and sufficient for $\Lambda_G(\mathcal{V}) = \Lambda_G$. In other words, to allow for maximum throughput achievability we must choose $\mathcal{V}$ to ensure that the path concatenations on these nodes form all missing paths in the network.

Theorem 1 *Given a placement of controllable nodes $\mathcal{V}$, satisfying the all-paths condition is necessary and sufficient for maximizing the throughput region, i.e.,*

$$\Lambda_G(\mathcal{V}) = \Lambda_G \quad \textit{if and only if} \quad \mathcal{P}(\mathcal{V}) = \mathcal{P}_G.$$

The proof is in the Appendix. Using the all-paths condition C.1, we define P3:

$$V_3^* = \min_{\mathcal{V}\subseteq\mathcal{N}} |\mathcal{V}| \quad \text{s.t.} \quad \text{All-paths condition C.1.} \tag{P3}$$

Corollary 1 *P1 $\iff$ P3, therefore $V_1^* = V_3^*$.*

4.1 Overlay Node Placement Algorithm

We design an algorithm to choose the placement of overlay nodes $\mathcal{V} \subseteq \mathcal{N}$ on a given graph $G = (\mathcal{N}, \mathcal{E})$ such that the choice of overlay nodes is sufficient to satisfy the full throughput region of the network, i.e. $\Lambda_G(\mathcal{V}) = \Lambda_G(\mathcal{N})$. 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

step 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 $G = (\mathcal{N}, \mathcal{E})$, and initialize $\mathcal{N}' := \mathcal{N}$ and $\mathcal{E}' := \mathcal{E}$. While there exists any node $n \in \mathcal{N}'$ such that degree$(n) = 1$, set $\mathcal{N}' := \mathcal{N}' \setminus n$ and set $\mathcal{E}' := \mathcal{E}' \setminus e$, 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 $G' = (\mathcal{N}', \mathcal{E}')$.

Lemma 1 *Suppose that placement $\mathcal{V}$ satisfies the all-paths condition (C.1), and $n \in \mathcal{V}$ lies on an attached tree. Then $\mathcal{V} \setminus n$ also satisfies the all-paths condition.*

Proof. To prove $\mathcal{P}(\mathcal{V}) = \mathcal{P}(\mathcal{V} \setminus n)$, it is enough to show that for any pair $a, b \in \mathcal{N}$, the acyclic path $P_{ab} \in \mathcal{P}(\mathcal{V})$ can be formed without concatenating paths at n. Note, that if $n \notin P_{ab}$, 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 P_{ab}. 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, $P_{ab} \in \mathcal{P}^{\text{SP}} = \mathcal{P}(\emptyset) \subseteq \mathcal{P}(\mathcal{V} \setminus n)$.

2. Node a is on a specific attached tree and node b is not on that tree: Assume that overlay nodes $w, v \in \mathcal{V} \cap P_{ab}$ *exist* on path P_{ab} such that w is not on the attached tree of node a and it is the closest overlay node to this tree on path P_{ab} (where distance is defined using hops on the path), while v is the overlay neighbor of w on the tree of a. Observe that values $w = b, n$ and $v = n$ are possible. Unless $w = b$, we have that $P_{wb} \in \mathcal{P}(\mathcal{V} \setminus n)$ since $n \notin P_{wb}$. From the fact $P_{ab} \in \mathcal{P}(\mathcal{V})$ and since v, w are overlay neighbors, we conclude $P_{vw} \in \mathcal{P}^{\text{SP}}$. By optimal substructure, we have that $P_{aw} \in \mathcal{P}^{\text{SP}}$. Concatenating at w we get $P_{ab} \in \mathcal{P}(\mathcal{V} \setminus n)$. For the case $w = b$, we immediately get $P_{vb} \in \mathcal{P}^{\text{SP}}$ and the result follows from optimal substructure. If w does not exist, then $P_{ab} \in \mathcal{P}^{\text{SP}} \subseteq \mathcal{P}(\mathcal{V} \setminus n)$. If v does not exist (this is a special case where b, n must lie on a different attached tree from a) we have $P_{aw} \in \mathcal{P}^{\text{SP}}$ thus we can still concatenate at w. Last if w, v both do not exist, then clearly $P_{ab} \in \mathcal{P}^{\text{SP}}$.

3. Node b is on a specific attached tree and node a is not on that tree: We define again w, v as in the above case. Symmetrically we have $P_{aw} \in \mathcal{P}(\mathcal{V} \setminus n)$ since $n \notin P_{aw}$, $P_{wv} \in \mathcal{P}^{\text{SP}}$ and by optimal substructure $P_{wb} \in \mathcal{P}^{\text{SP}}$. Thus, concatenating at w we obtain $P_{ab} \in \mathcal{P}(\mathcal{V} \setminus n)$. Special cases for w, v are treated in a similar manner.

4. Nodes a and b are both on G': No possible acyclic path from node a to node b can go through attached trees, as entering and exiting an attached tree forms a cycle. Thus, $n \notin P_{ab}$ from which we obtain directly $P_{ab} \in \mathcal{P}(\mathcal{V} \setminus n)$. □

By induction, it suffices to allocate overlay nodes in G' to satisfy the all-paths condition.

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 $v_n = 1$.

Figure 3: Summary of node placement algorithm.

Phase 2: Constraint Pruning. In this phase, we define the destination trees which will be used to find the constraints for node placement. Exploiting a necessary condition from Lemma 2 regarding the placement of controllable nodes, we show that proper pruning of these destination trees will identify the set of constraints over which we minimize the allocation of controllable nodes.

By optimal substructure, the union of shortest-paths P^{SP}_{xn} to any destination n from all nodes $x \in \mathcal{N}' \setminus n$ forms destination tree D_n. Define $\{P^{\text{SP}}_{xn}\} \setminus n$ to be the set of nodes on the shortest path from x to n, excluding node n. We have the following.

Lemma 2 *If the degree of node x on tree D_n is less than the degree of x on graph G', and there is no overlay node along the shortest path from x to n (i.e. $\nexists v \in \mathcal{V} : v \in \{P^{SP}_{xn}\} \setminus n$), then the all-paths condition C.1 is not satisfied.*

Proof. Let (b, x) be an edge in G' but not in D_n, where such an edge exists by the premise of Lemma 2. Consider path p formed from the concatenation of (b, x) and shortest-path P^{SP}_{xn}. We will show that this path cannot be formed if there are no controllable nodes in the shortest path from x to n, and thus the all-paths condition C.1 is not satisfied.

First, observe that since edge (b, x) is not on tree D_n, shortest-path P^{SP}_{bn} does not include this edge. Thus, the path p requires a concatenation of two or more shortest-paths. Such a concatenation must occur at a controllable node on path P^{SP}_{xn}. However, this is impossible since there are no controllable nodes on path P^{SP}_{xn}. Thus, C.1 is not satisfied. □

For Phase 2, we prune destination trees D_n at nodes with degree less in D_n than in G' to obtain *pruned trees* D'_n. By Lemma 2, for the all-paths condition to be satisfied it is necessary to have at least one overlay node on the shortest path to n from every leaf node of pruned tree D'_n. The pruned trees D'_n and this necessary condition from Lemma 2 will be used as constraints in Phase 3.

Phase 3: Overlay Node Placement. Consider the following binary program to place the minimum number of overlay nodes to satisfy Lemma 2 for all nodes on all pruned

trees D'_n:

$$V_4^* = \min \sum_n v_n$$
$$\text{s.t.} \sum_{a \in \{P_{bn}^{\text{SP}}\} \setminus n} v_a \geq 1, \ \forall b \in \text{LeafNodes}(D'_n), \forall n \quad \text{(P4)}$$
$$v_n \in \{0,1\}, \ \forall n$$

where LeafNodes(D'_n) is the set of all leaf nodes on pruned tree D'_n, and where $\{P_{bn}^{\text{SP}}\} \setminus n$ is defined in Phase 2. Next, we show that the placement determined by the solution of P4 satisfies the all-paths condition.

Lemma 3 *The overlay node placement of P4 satisfies the all-paths condition for graph G'.*

The proof is in the Appendix.

A summary of the algorithm is shown in Figure 3. The following main result establishes the performance of the proposed placement algorithm.

Theorem 2 *Let $\mathcal{V}^*$ be the solution produced by the overlay node placement algorithm. Then, $\mathcal{V}^*$ is an optimal solution to P3. It follows that*

- $\Lambda_G(\mathcal{V}^*) = \Lambda_G$.
- *$\mathcal{V}^*$ is an optimal solution to P1.*

PROOF. By Lemma 2, the constraint of P4 is necessary for the all-paths condition. By Lemmas 1 and 3 it is also sufficient. Thus, we have $P4 \iff P3$. By Theorem 1, the remaining assertions follow. □

Phases 1-2 of the algorithm have complexity $O(N^2)$. P4 solves a vertex cover problem, which is known to be NP-Hard in general. However, note that the constraints of our problem have optimal substructure, which might be exploitable. For our experiments on graphs with 1000 nodes, the solver found most solutions to P4 within 5 seconds, and we only rarely encountered scenarios that required more than a few minutes to solve. Thus, the algorithm is practical.

5. EXPERIMENTS WITH OPTIMAL NODE PLACEMENT

We provide results for various types of network graphs, including specific graph families and random graphs. By Theorem 1, the full throughput region is provided by the placement of our algorithm on all these cases.

5.1 Simple Scenarios

5.1.1 Trees and Forests

Consider trees with single-path underlay routes P_{ab}^{SP} for every pair of nodes a and b. A tree is loop free, and thus each path P_{ab}^{SP} is the unique acyclic path from node a to b. Thus, the all-paths condition is automatically satisfied, and $\Lambda_G(\emptyset) = \Lambda_G(\mathcal{N})$.

It follows that no controllable nodes are required for a forest, which is a disjoint union of trees.

5.1.2 Cycles and Rings

Lemma 4 *Every cycle requires at least 3 controllable nodes to satisfy the all-paths condition.*

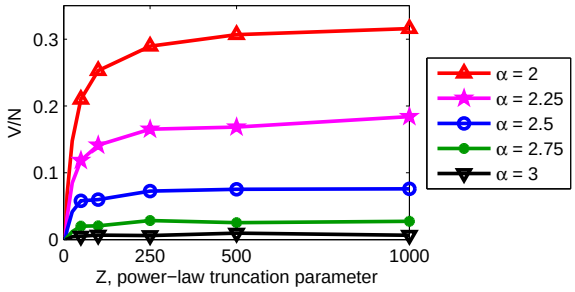


Figure 4: Results of overlay node placement algorithm on random graphs where node degree follows a power-law distribution with exponent α. Graphs generated with configuration model and truncated Zipf distribution.

PROOF. Consider controllable nodes $v, w \in \mathcal{V}$ on a cycle, and without loss of generality assume shortest-path P_{vw}^{SP} is on the cycle. Then path P_{vw}^{SP} allows one direction of flow on the cycle, and at least one additional controllable node is required to allow flow in the counter direction on the cycle. Note that the same problem occurs in scenarios with 0 or 1 controllable node on the cycle, and when path P_{vw}^{SP} is not on the cycle. Thus, at least 3 controllable nodes are required on each cycle in the network. □

Further, the lower bound from Lemma 4 is tight for the case of a ring, where the entire graph is a single cycle.

Lemma 5 *Exactly 3 controllable nodes are required to satisfy the all-paths condition for a ring network with $N \geq 5$ nodes and hop-count as the metric for shortest-path routing.*

The proof is in the Appendix.

5.1.3 Cliques

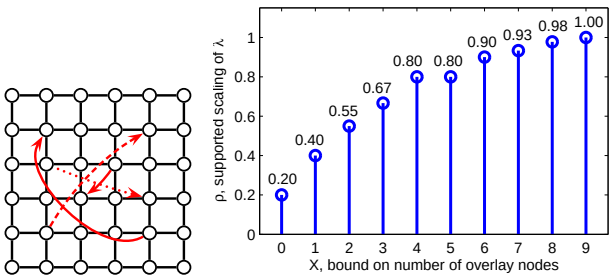
Consider cliques with single-path underlay routes P_{ab}^{SP} for every pair of nodes a and b. We require all edges (a, b) be included in the underlay routes, however there is an edge between every pair of nodes in a clique. Thus, all underlay routes are single edges, i.e. $P_{ab}^{\text{SP}} = (a, b)$ for all pairs $a, b \in \mathcal{N}$. A Hamiltonian path, traversing all nodes, will require all intermediate nodes to be controllable. Such paths can start and end at any node, therefore the all-paths condition requires all nodes to be controllable for a clique, i.e. $\mathcal{V} = \mathcal{N}$.

5.2 Random Networks

This section considers placement of overlay nodes to support the full throughput region on random graphs. We present here results about power-law graphs, where the degree of nodes is random and roughly follows a power-law distribution. This is recognized as a realistic model for the Internet [7]. We have experimented with several other models for random graphs, which are omitted here due to lack of space.

We construct random networks that have power-law degree distributions using the configuration model and a truncated Zipf distribution [7]. Zipf is a discrete distribution with parameters α and Z, where α is the power-law exponent and Z is a truncation parameter indicating the maximum degree of the distribution. The Zipf PMF is

$$\mathbb{P}(D = d) = \frac{d^{-\alpha}}{\sum_{k=1}^{Z} k^{-\alpha}}, \quad \text{for } d = 1, ..., Z.$$



(a) Scenario (b) Node Placement Results

Figure 5: Results of P5 for chosen rate vector on 6×6 grid. (a) Arrival rate vector $\boldsymbol{\lambda}$ includes traffic demands for all all pairs of nodes. Here we consider $\boldsymbol{\lambda}$ with four active traffic demands with symmetric rates, as indicated with arrows. (b) Fraction of $\boldsymbol{\lambda}$ supported when limited to $|\mathcal{V}| \leq X$ controllable nodes.

For a given number of nodes N, the configuration model attaches a number of *stubs* to each node according to the Zipf distribution, where a stub is half of an edge. Pairs of unconnected stubs are then chosen randomly and connected to form edges. Thus, node degree follows a power-law distribution.

Figure 4 shows results from the overlay node placement algorithm for random power-law graphs with $N = 1000$ nodes, averaged over 10 realizations per data point. Values of α between 2 and 3 are considered, with $\alpha = 2.5$ being a frequent estimate for the Internet [7]. For $\alpha = 2.5$, the overlay node placement algorithm finds that less than 8% of nodes need to be controllable for the full throughput region to be achievable.

6. PLACING A LIMITED NUMBER OF OVERLAY NODES

A formulation similar to P2 is useful in scenarios where only a small subset of arrival rate vectors require support, such that the constraints are limited to the specific vectors $\boldsymbol{\lambda}^{(i)}$ of interest. For example, this includes networks with nodes that neither generate nor consume information such as network routers. This approach can also use P2 to minimize the number of controllable nodes required to allow maximum flow between a specific source and destination.

A similar formulation can be used to maximize the achievable flow when the maximum number of controllable nodes is upper bounded by some number X, as shown in P5. This can be useful in scenarios where resource limitations don't allow enough controllable nodes to achieve maximum throughput. As in P2, multiple rate vectors $\boldsymbol{\lambda}^{(i)}$ can be supported with additional constraints $\rho\boldsymbol{\lambda}^{(i)} \in \Lambda_G(\mathcal{V})$.

$$\max_{\mathcal{V}\subseteq\mathcal{N}} \rho \quad \text{s.t.} \quad \rho\boldsymbol{\lambda} \in \Lambda_G(\mathcal{V}), \quad |\mathcal{V}| \leq X. \tag{P5}$$

Figure 5 shows results of P5 on a 6×6 grid for a specific rate vector $\boldsymbol{\lambda}$ with four equal traffic demands. Figure 5b shows that a fraction 80% of throughput is supported in the direction $\boldsymbol{\lambda}$ with only $X = 4$ and diminishing returns from additional overlay nodes, with $X = 9$ required for 100% throughput.

7. BACKPRESSURE OVERLAY POLICY

Subject to the placement of overlay nodes, we study the problem of throughput maximization using dynamic routing decisions at overlay nodes.

7.1 The Control Problem

We are interested in a dynamic policy that is stable for any arrival vector in the region $\Lambda_G(\mathcal{V})$, i.e. achieves maximum throughput. Under policy π, let $\mu^c_{vn}(t,\pi)$ be the service function on the link (v,n) for commodity c packets, where $v \in \mathcal{V}$ and $n \in \mathcal{N}$. The edge rate constraint implies $\sum_c \mu^c_{vn}(t,\pi) \leq R_{vn}$ must be satisfied at every slot. Thus, at each overlay node, the policy chooses the number of packets to be sent to any outgoing neighbor by assigning values to these functions. The uncontrollable nodes $\mathcal{N} \setminus \mathcal{V}$ are assumed to only forward the packets on pre-specified shortest-paths, and are assumed to use a fair scheduler to choose between competing commodities. In our simulations we use the example of proportionally-fair random service on a packet-by-packet basis, such that node a transmits a packet of commodity c with probability $p^c_a = U^c_a / \sum_x U^x_a$, where U^x_a is the queue backlog for commodity x at uncontrollable node a.

Every overlay node $v \in \mathcal{V}$ maintains a queue for each commodity c and we denote its backlog with $Q^c_v(t)$ at slot t. For two overlay neighbors $v, w \in \mathcal{V}$, we define $F^c_{vw}(t)$ the number of commodity c packets that have departed overlay node v but have not yet reached overlay node w. We call these the *packets-in-flight* between overlay nodes v and w for commodity c. While it may be impractical to observe individual queue sizes at uncontrollable nodes, counting F^c_{vw} can be realized by looking at the difference in cumulative count for packets of commodity c sent from v to w versus the cumulative count for these packets received at w. This may be realized using a simple packet acknowledgment scheme for each pair of overlay nodes.

In what follows, we study the problem of controlling this system by observing queue backlogs Q and packets-in-flight F, and choosing service function μ at overlay nodes only.

7.2 Insufficiency of Traditional Backpressure

For an interference-free wired network, the backpressure (BP) routing policy [10] is as follows. For each link (a,b), define the differential backlog $W^c_{ab}(t)$,

$$W^c_{ab}(t) = Q^c_a(t) - Q^c_b(t), \quad \forall\,(a,b) \in \mathcal{E}, \forall\, c \in \mathcal{N},$$

and define commodity $c^*_{ab}(t)$ that maximizes this weight,

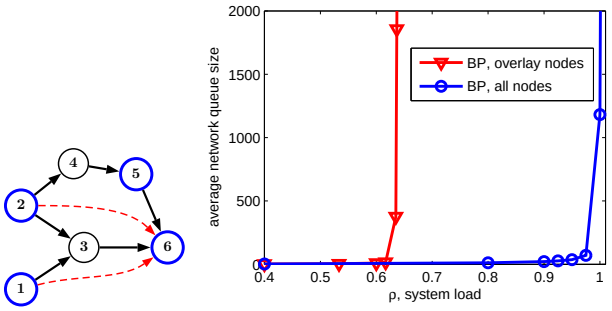
$$c^*_{ab}(t) \in \arg\max_{c\in\mathcal{N}} W^c_{ab}\,, \quad \forall\,(a,b) \in \mathcal{E}, \tag{11}$$

with ties broken arbitrarily. Then the BP policy chooses

$$\mu^{c^*_{ab}}_{ab}(t,\mathrm{BP}) = \begin{cases} R_{ab} & \text{if } W^{c^*_{ab}}_{ab} > 0 \\ 0 & \text{otherwise,} \end{cases} \tag{12}$$

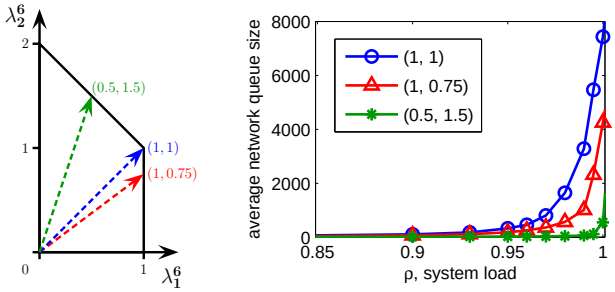
where $\mu^c_{ab}(t,\mathrm{BP}) = 0,\ \forall c \neq c^*_{ab}$. In [10], this policy is shown to stabilize any point in the region $\Lambda_G(\mathcal{N})$.

The intuition behind the optimality of BP is that congestion information propagates through the network via queue backlogs. The policy balances neighboring backlogs, such that when node n becomes congested, any upstream neighbors of n also become congested. In our problem, the uncontrollable nodes do not use BP, and thus any congestion



(a) Scenario (b) Simulation Results

Figure 6: Insufficiency of BP in overlay networks. (a) Scenario with contention at uncontrollable node 3. (b) Queue size of BP in overlay vs. BP in underlay.



(a) Throughput Region (b) OBP Simulation Results

Figure 7: Evaluation of OBP policy on scenario from Figure 6a. (a) Throughput region of Figure 6a, with select rate vectors indicated. (b) Average queue backlog of OBP, after 1e6 time steps, for rate vectors indicated in (a).

occurring on these nodes is not propagated. This is the primary reason why we do not expect BP to perform well in our system.

Consider the example of Figure 6a, where (controllable) overlay nodes $\mathcal{V} = \{1, 2, 5, 6\}$ are indicated in blue, with directed unit-rate links. It can easily be verified that the all-paths condition C.1 is satisfied for this setting, thus $\Lambda_G(\mathcal{V}) = \Lambda_G(\mathcal{N})$. The dashed red arrows show two traffic demands with symmetric arrival rates λ. With unit-rate links, offered load $\rho = \lambda$, where $\rho < 1$ is required for this network to be stable. We examine two different cases. First, we run BP at all nodes; this achieves maximum throughout and it is stable for all $\rho < 1$. Second, we run BP only at overlay nodes, computing differential backlogs across the overlay edges, e.g. node 2 computes $W_{2,5}^6 = Q_2^6 - Q_5^6$ and $W_{2,6}^6 = Q_2^6 - Q_6^6$. Simulation results in Figure 6b show that BP at the overlay nodes cannot stabilize $\rho > 2/3$, i.e. it is throughput suboptimal. The intuition is as follows. Note that $Q_6^6 = 0$, since node 6 is a destination. Then, any congestion at uncontrollable node 3 cannot be detected by source node 2, leading to positive traffic flow from source 2 through node 3. This motivates our policy in the following section.

7.3 The Proposed OBP Policy

Let $\overline{\mathcal{E}}$ represent the set of edges in the overlay network. We propose the following policy, both dynamic and distributed, to account for packets-in-flight.

Overlay Backpressure (OBP). Redefine the differential backlog as

$$W_{vw}^c(t) = Q_v^c(t) - Q_w^c(t) - F_{vw}^c(t), \quad \forall\,(v,w) \in \overline{\mathcal{E}}, \forall\, c \in \mathcal{N},$$

then determine c_{vw}^* and $\mu_{vw}^c(t, \text{OBP})$ as in Eqns. (11)-(12).

Intuitively, this policy takes into account both the packet accumulation at the neighbor overlay node v, as well as any packets-in-flight on the path P_{vw}, in the form of *negative pressure*. Through simulation we observe the following properties of the algorithm. (i) OBP maximizes throughput in all examined scenarios, including the one of Figure 6a, (ii) OBP outperforms BP applied only at overlay nodes, and (iii) OBP has good delay properties, outperforming BP even when the latter is applied at all nodes.

In Figure 7, we study different arrival vectors for the network of Figure 6a. The simulation results in Figure 7b show that all studied vectors are supported by the OBP policy.

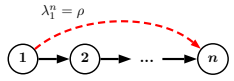
In Figure 8, we study a directed tandem network for the purpose of illustrating the delay properties of OBP. From [2] it is known that for BP on a tandem network, per-node queues grow linearly with distance from the destination, and thus network queue size grows quadratically with the total number of nodes. However, for the OBP policy we observe this linear growth of per-node queues only at controllable nodes, implying smaller total network queues size and improved delay performance when there are few controllable nodes. In this particular example, only the source is controllable, with $n-1$ legacy nodes, a setting that corresponds to the maximum benefit. Delay is compared between BP and OBP for a fixed offered load in Figure 8b and for a fixed number of nodes in Figure 8c. Although BP is applied at all nodes it is still outperformed by OBP applied only at the source.

Finally, in Figure 9, we show simulation results from three policies: OBP, BP at all nodes, and BP with shortest-path bias (BP+SP) from [6]. Although the latter two are both throughput optimal policies, they yield worse delay than OBP. The reason is threefold: (i) the quadratic network queue size of BP is proportional to the number of controllable nodes used (in this scenario, OBP uses only 5 overlay nodes), (ii) no packets are sent to attached trees in case of OBP, and (iii) under light traffic, packets under BP perform random walks.

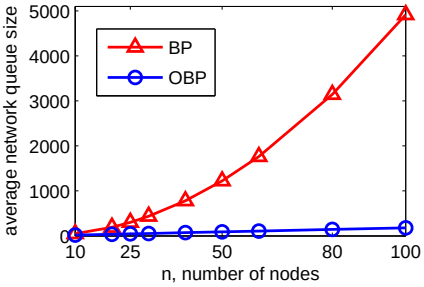
While our OBP policy seems to perform well in simulations, we do not believe that it is optimal in general settings. A promising future direction of research is to identify a maximally stable dynamic routing policy for our overlay architecture.

8. CONCLUSIONS

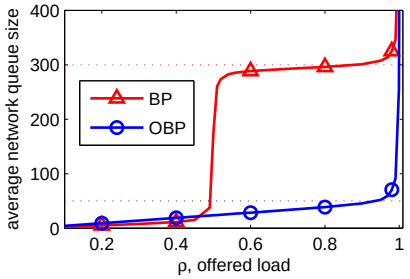
We study optimal routing in legacy networks where only a subset of nodes can make dynamic routing decisions, while the legacy nodes can forward packets only on pre-specified shortest-paths. This model captures evolving heterogeneous networks where intelligence is introduced at a fraction of nodes. We propose a necessary and sufficient condition for the overlay node placement to enable the full multicommodity throughput region. Based on this condition, we devise an algorithm for optimal controllable node placement. We run the algorithm on large random graphs to show that very often a small number of intelligent nodes suffices for full throughput. Finally, we propose a dynamic routing policy to be implemented in a network overlay, that demonstrates superior performance in terms of both throughput and delay.



(a) Scenario

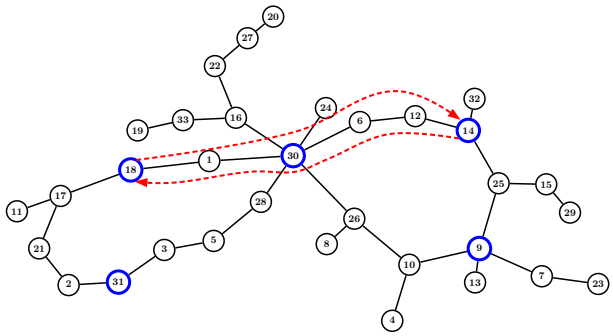


(b) Simulation Results for Fixed $\rho = 0.8$

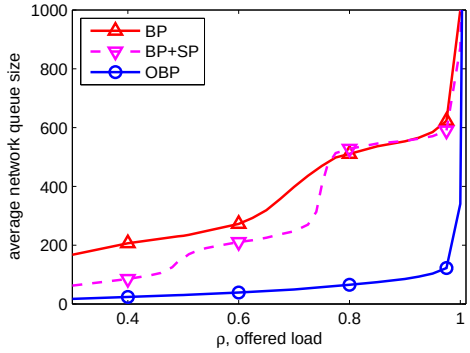


(c) Simulation Results for Fixed $n = 25$

Figure 8: Directed tandem with n nodes. (b) BP versus OBP for offered load $\rho = 0.8$. Quadratic growth for BP. Linear growth for OBP. (c) BP versus OBP for $n = 25$. Quadratic backlog in BP results for $\rho > 0.5$. Dotted horizontal line at $n(n-1)/2$ for BP, and at $2n$ for OBP.



(a) Scenario



(b) Simulation Results

Figure 9: Comparing OBP with BP on a random graph. (a) Scenario with two symmetric traffic demands. (b) Average queue size for BP, BP+SP, and OBP.

9. REFERENCES

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

[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*, April 2009.

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

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

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

[6] 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.

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

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

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

[10] 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.

[11] D. Xu, M. Chiang, and J. Rexford. Link-state routing with hop-by-hop forwarding can achieve optimal traffic engineering. *IEEE/ACM Transactions on Networking*, 19(6):1717–1730, 2011.

[12] L. Ying, S. Shakkottai, and A. Reddy. On combining shortest-path and back-pressure routing over multihop wireless networks. In *Proc. IEEE INFOCOM*, April 2009.

APPENDIX

PROOF OF SUFFICIENCY FOR THEOREM 1. We will show that the all-paths condition is sufficient for supporting any multicommodity vector $\boldsymbol{\lambda} \in \Lambda_G$ while bifurcating traffic only at nodes $\mathcal{V}$. Feasibility of $\boldsymbol{\lambda}$ implies existence of a feasible flow decomposition of $\boldsymbol{\lambda}$. Without loss of generality, choose any one component of $\boldsymbol{\lambda}$ that sends flow from node a to node b with corresponding arrival rate λ_a^b. This arrival rate λ_a^b is supported by flow $f_{ab}^{\boldsymbol{\lambda}}$, where $f_{ab}^{\boldsymbol{\lambda}}$ can be decomposed into subflows $f_{ab}^{\boldsymbol{\lambda}}(p)$ for paths $p \in \mathcal{P}_{ab}$. Since $\mathcal{V}$ satisfies all path condition it follows that all-paths $\mathcal{P}_{ab}$ can be formed as concatenations of available shortest paths on nodes $\mathcal{V}$, and thus the feasible flow decomposition can be constructed with a stationary policy using underlay routes and the given set of controllable overlay nodes. □

PROOF OF NECESSITY FOR THEOREM 1. We will show that given a $\mathcal{V}$ such that there is a path that is not available either as a shortest path or as a concatenation, i.e. the all-paths condition is not satisfied, the full throughput region cannot be achieved. Support of the full throughput region requires support for all arrival rate vectors interior to the rate region allowed by the network. Assume $\Lambda_G(\mathcal{V}) = \Lambda_G$ and some path P^{X}_{ab} is unavailable, both as a shortest-path and as an n-concatenation of shortest-paths at controllable nodes $\mathcal{V}$. Without loss of generality, assume that this unavailable path does not traverse any controllable nodes. Otherwise, split the unavailable path at controllable nodes and choose an unavailable segment induced from the split as path P^{X}_{ab}; such an unavailable segment must exist, otherwise the original path could be formed as an n-concatenation of the induced segments. We will show that there exists a feasible arrival rate vector that requires use of the unavailable path P^{X}_{ab}.

Construct an arrival rate vector $\boldsymbol{\lambda}$ that includes component λ^b_a equal to the maximum flow allowed for path P^{X}_{ab}, plus edge rate R_{ab} if edge (a,b) exists. In vector $\boldsymbol{\lambda}$, also include one-hop traffic demands for all edges $(i,j) \in \mathcal{E} \setminus (a,b)$ by choosing λ^j_i to equal any remaining capacity on edge (i,j). This rate vector $\boldsymbol{\lambda}$ is then feasible by construction.

Let $\mathcal{N}^{\mathrm{X}}_{ab}$ be the set of nodes on path P^{X}_{ab}. For every node j not on path P^{X}_{ab}, i.e., $j \in \mathcal{N} \setminus \mathcal{N}^{\mathrm{X}}_{ab}$, the arrival rate vector $\boldsymbol{\lambda}$ was constructed such that $\sum_i \lambda^j_i = \sum_i R_{ij}$. Applying the edge rate constraints from Equation (6) at node j and taking the sum over all neighbors i, we have $\sum_i \sum_{x,y,c} f^{xy,c}_{ij} \leq \sum_i R_{ij} = \sum_i \lambda^j_i$ for all $j \in \mathcal{N} \setminus \mathcal{N}^{\mathrm{X}}_{ab}$, where the last equality comes from the previous equation. Then flow conservation requires that $f^{xy,c}_{ij} = 0$ for all commodities $c \neq j$. Thus, no feasible flow decomposition of $\boldsymbol{\lambda}$ can route flow for λ^b_a through any nodes in $\mathcal{N} \setminus \mathcal{N}^{\mathrm{X}}_{ab}$. Therefore, it remains to consider only nodes in $\mathcal{N}^{\mathrm{X}}_{ab}$ to support λ^b_a.

If P^{X}_{ab} is the only path from node a to b using nodes from the set $\mathcal{N}^{\mathrm{X}}_{ab}$, then P^{X}_{ab} is clearly necessary to support flow λ^b_a. Otherwise, recall that by assumption there are no controllable nodes intermediate to path P^{X}_{ab}. Then it remains only to consider the case where the shortest-path from node a to b uses a strict subset of nodes in $\mathcal{N}^{\mathrm{X}}_{ab}$, as no controllable nodes are available for path concatenation. Consider edge (i,j) such that nodes i and j are on path P^{X}_{ab}, where edge (i,j) is on P^{SP}_{ab} but not on P^{X}_{ab}. Here, $P^{\mathrm{SP}}_{ij} = (i,j)$ is the only available path from i to j with unused capacity, because no controllable nodes are available. Then, $f^{ij,j}_{ij} = \lambda^j_i = R_{ij}$, and Equation (6) requires $f^{ab,b}_{ij} = 0$. Therefore, there is no unused capacity on path P^{SP}_{ab}, so λ^b_a and λ^j_i cannot be supported simultaneously. There are no other paths to consider from node a to b for a feasible flow decomposition of $\boldsymbol{\lambda}$.

Therefore, $\Lambda_G(\mathcal{V}) \subset \Lambda_G$ if any path is not available. Thus, we have proved the necessity of the all-paths condition for wired networks with shortest-path routing. □

PROOF FOR LEMMA 3. Let $\mathcal{V}$ be a overlay node placement chosen by P4, and consider every acyclic path P_{ab} between all pairs of nodes a and b in graph G'. For all such paths P_{ab}, we will show that either (1) P_{ab} is a shortest-path or (2) P_{ab} can be formed as a concatenation of shortest-paths at overlay nodes $\mathcal{V}$. Thus, $P_{ab} \in \mathcal{P}(\mathcal{V})$ for all-paths P_{ab} on graph G', proving that $\mathcal{P}(\mathcal{V}) = \mathcal{P}_{G'}$, i.e., that the all-paths condition is satisfied.

Define *overlay neighbor tree* D''_n to be the union of shortest path routes to node n from all overlay neighbors of n, i.e. nodes $v \in \mathcal{V}$ such that flow on path P^{SP}_{vn} is allowed by Eqns. (4-5). Because P4 places overlay nodes on the shortest paths from the leaf nodes of D'_n to n, we have the relationship $D''_n \subseteq D'_n \subseteq D_n$. The leaf nodes of D''_n are the closest overlay nodes to n, and we will make use of this construction.

For each path P_{ab}, one of two cases must hold.

(1) The entire path P_{ab} is contained in overlay neighbor tree D''_b. In this case, $P_{ab} = P^{\mathrm{SP}}_{ab}$, so $P_{ab} \in \mathcal{P}(\mathcal{V})$.

(2) There exists an overlay node $v \in D''_b$ such that path P_{ab} is a concatenation of paths P_{av} and P^{SP}_{vb} at v.

Path P^{SP}_{vb} is provided by a shortest-path route, so it only remains to show that path P_{av} is either (1) a shortest-path or (2) can be formed as a concatenation of shortest-paths at overlay nodes $\mathcal{V}$, i.e., $P_{av} \in \mathcal{P}(\mathcal{V})$. To show this, first note that neighbor tree D''_b includes all neighbors of node b, and that v is at least one hop away from b. Then path P^{SP}_{vb} has a positive length, and thus the length of path P_{av} is strictly less than the length of path P_{ab}. We can then iteratively repeat the above two-case argument by letting $b' = v$, and consider sub-path $P_{ab'}$, repeatedly shortening the path until case (1) holds.

Therefore, every path P_{ab} on graph G' is also in the set of paths $\mathcal{P}(\mathcal{V})$. Thus, $\mathcal{P}(\mathcal{V}) = \mathcal{P}_{G'}$, and the all-paths condition is satisfied. □

PROOF FOR LEMMA 5. Lemma 4 establishes the necessity of at least 3 controllable nodes, so it only remains to show that 3 controllable nodes are sufficient to satisfy the all-paths condition.

Starting from any node x, consider nodes y and z that are neighbors, i.e., $(y,z) \in \mathcal{E}$, where shortest-paths P^{SP}_{xy} and P^{SP}_{xz} are disjoint. Without loss of generality assume $|P^{\mathrm{SP}}_{xy}| \leq |P^{\mathrm{SP}}_{xz}|$ where $|p|$ is the length of path p. With hop-count as the shortest-path metric, the length of these disjoint shortest-paths can differ at most by 1. Otherwise, there would exist a contradiction, as the path formed as a concatenation of P^{SP}_{xy} with edge (y,z) would be shorter than shortest-path P^{SP}_{xz}. Then the following inequality holds for any number of nodes $N \geq 5$.

$$|P^{\mathrm{SP}}_{xy}| \geq \left\lfloor \frac{N-1}{2} \right\rfloor \geq \frac{N}{3} \tag{13}$$

Therefore, any node can reach a minimum of $N/3$ nodes in either direction around the ring using shortest-path routing. Conversely, any node can be reached by a minimum of $N/3$ nodes in either direction. Then we can place 3 controllable nodes, v_1, v_2, and v_3, such that shortest-paths $P^{\mathrm{SP}}_{v_i v_j}$ and $P^{\mathrm{SP}}_{v_i v_k}$ are edge-disjoint for all permutations $i,j,k \in \{1,2,3\}$. The overlay edges between these controllable nodes then form a bidirectionally connected ring as shown in Figure 1, making use of all-paths between the controllable nodes. Every uncontrollable u is on the shortest-path between two controllable nodes v_i and v_j; thus, by optimal substructure, paths $P^{\mathrm{SP}}_{uv_i}$ and $P^{\mathrm{SP}}_{uv_j}$ are edge-disjoint paths from u to v_i and v_j, and paths $P^{\mathrm{SP}}_{v_i u}$ and $P^{\mathrm{SP}}_{v_j u}$ are edge-disjoint paths from v_i and v_j to node u. Then every path in the network is either a shortest-path or can be formed as an n-concatenation of shortest paths, and the all-paths condition is satisfied with exactly 3 controllable nodes. □