The Confluent Capacity of the Internet: Congestion vs. Dilation

Jiangzhuo Chen, Madhav Marathe, Rajmohan Rajaraman, and Ravi Sundaram

***Abstract*—Using shortest paths, the Internet scales very poorly with respect to congestion [1]. Two main reasons for using shortest paths are dilation (or delay) and size of routing tables. As the Internet grows, the small size of routing tables is important for scaling, but it does not require shortest paths. As long as the paths are *confluent,* the routing table size is unchanged.**

In this paper we study the confluent capacity of the Internet. We use the preferential attachment model [2] for the Internet, and all-pair uniform demand for the traffic pattern. Our main theoretical result is that the confluent congestion[1] is within a logarithmic factor of the optimal splittable congestion and can be achieved using a simple randomized and distributed scheme called *Locally Independent Rounding Algorithm* (LIRA). We reinforce this result experimentally by employing simulations to demonstrate that for almost all instances the confluent congestion is (nearly) equal to the splittable congestion. Thus we conclude that the Internet scales well using confluent paths.

We combine known results on expanders and the expansion properties of the preferential attachment model to show that for almost all Internet-like networks, we can find a confluent flow that simultaneously achieves $O(\log n)$-approximate congestion and $O(1)$-approximate dilation. We confirm, using simulations, the intuition that confluence does not come at the cost of dilation.

I. Introduction

The Internet has been growing fast in the past decade and it will keep growing in the future. One question is how the Internet capacity scales, if the link capacities are linearly growing with the Internet size. The Internet capacity is the fraction of routing demands that can be satisfied. If the Internet capacity is decreasing even with the link capacities growing in step with the Internet size, then the growth of the Internet cannot sustain at its current rate. The propagation delay, which is associated with the dilation of the routing paths, is also important for the Internet to grow.

A lot of experimental work has been done to understand the structure and growth of the Internet. A more rigorous analysis, however, requires a formal evolving model of the Internet. One of the widely accepted model is based on preferential attachment [2], which produces networks with power-law degree distributions. Recently, the Internet capacity scaling problem based on the preferential attachment model has been the focus of some research [1], where $n^{1+\Omega(1)}$ congestion is shown for shortest path routing, suggesting poor scaling of the capacity of the Internet. Natural questions are why do we want to use shortest paths and can we do better than shortest path routing.

It turns out that there are two major reasons for using shortest path routing. In terms of dilation, it is the optimal routing scheme. Secondly, it guarantees small routing tables, since each router only needs to maintain a routing table which has one entry for each destination (or a block of destinations). Upon receiving a packet, the router will look up the routing table and determine its next hop by its destination. Linear routing table size ensures fast table look-up and packet dispatching. It is shown in [3] that we can achieve $O(n \log n)$ congestion if we allow arbitrary routes. But the routing table size will become arbitrarily large. The property of linear routing table size, however, does not constrain us exclusively to shortest path routing. In fact, any routing scheme based on *confluent* flows will lead to small routing tables.

A network flow of a given commodity is said to be *confluent* if all the flow of this commodity arriving at a node departs from the node along a single edge. If we view packets towards the same destination as one commodity, then small routing table size is a natural result of any confluent routing scheme.

In this work, we are interested in confluent alternatives to shortest paths. We will show that the poor scaling of the Internet property is not a necessary result of confluent routing. We can keep linear size routing tables, and achieve $O(n \log n)$ congestion using a simple routing scheme. This certainly comes with a cost in dilation. But we will show that we do not incur a large increase in the dilation.

Our contributions. We study the scaling of the capacity of the Internet as it grows in size. Our main results are:

- $O(n \log n)$ congestion. We propose a distributed scheme LIRA that can confluently route all-pair uniform demands with congestion $O(n \log n)$ for the preferential attachment model. Observe that the best achievable congestion, even if we allow non-confluent routing, is $O(n \log n)$ [3].
- Simulations. We support our theoretical analysis with detailed simulations. Through comparison with the work in [1], we show that our algorithm performs much better than the shortest path routing in terms of congestion, with only a slightly increase in dilation.
- $O(n \log n)$ congestion and $O(1)$ dilation simultaneously. Starting from a low dilation splittable flow, we show how LIRA can be employed to achieve $O(1)$-approximate

J. Chen is with Network Dynamics and Simulation Science Laboratory, Virginia Bioinformatics Institute, Virginia Tech, Blacksburg, VA 24061. Email: {chenj}@vbi.vt.edu

M. Marathe is with Network Dynamics and Simulation Science Laboratory, Virginia Bioinformatics Institute and Dept. of Computer Science, Virginia Tech, Blacksburg, VA 24061. Email: {mmarathe}@vbi.vt.edu

R. Rajaraman and R. Sundaram are with College of Computer and Information Science, Northeastern University, Boston, MA 02115. Email: {rraj,koods}@ccs.neu.edu

This research was partially supported by NSF Career award CCR-9983901.

[1]We use *congestion* and *capacity* interchangeably, as they are reciprocals of each other. More details can be found in Section III.

dilation in addition to $O(n \log n)$ congestion for the preferential attachment model. Low dilation splittable flows can be computed by applying the expansion properties of Internet-like networks.

We review related work in Section II. After formally introducing the models and the problems in Section III, we first analyze our congestion minimization algorithm for the general networks in Section IV, then we extend the analysis to Internet-like networks (preferential attachment model) and back the theoretical results with simulations, in Section V. Then in Section VI we amend our algorithm to take dilation into consideration. We conclude with the limitations of our work and future directions in Section VII.

II. Background and Related Work

Recently the problem of modeling the Internet topology has attracted much interest. Empirical studies have found compelling evidence of power-law distributions in the Internet [4]–[7]. It is generally agreed that power law degree distribution is a most important property of the Internet, although it is still arguable which generative model is more precise. Models for growing networks such that the power law degree distribution is achieved include the *preferential attachment* model of Barabási and Albert [2], and the *heuristically optimized trade-offs* model of Fabrikant, Koutsoupias and Papadimitriou [8]. Graphs with power-law degree distribution are called *scale-free graphs*, introduced in [2] and used to model complex networks including the Internet (router-level graphs and AS-level graphs). The *power law random graph* model of Aiello, Chung and Lu [9] is a generic model for all scale-free graphs, regardless of how they evolve.

Several topology generators have been implemented to generate Internet-like networks for experimental study on the Internet. The one we use is Inet-3.0, which is a combination of the power law random graph model and the preferential attachment model. It first generates a degree sequence satisfying power law distribution, and then forms a network according to preferential attachment. It targets generating networks very similar to the Internet, in terms of degree distribution, connectivity, average path length, distortion, etc.

The shortcoming of shortest path routing has been long noticed. Many alternative confluent routing heuristics have been proposed (see [10], [11] and the references therein). Unfortunately, none of them provides a provable performance guarantee for the Internet-like networks. Our algorithm has provable upper bounds on both congestion and dilation. In the context of packet routing, congestion and dilation are both lower bounds for the time required to route all the packets. It is proved in [12] that the lower bounds can be achieved. But the paths are specified in [12], while in our case we are only given the demand function and need to find the paths, such that both lower bounds are optimized. Bicriteria optimization is studied in [13].

Confluent flow is a natural follow-up to splittable flow and unsplittable flow, which allows at most one path from each source (see eg. [14] and the references therein for further discussions on unsplittable flow). With confluence, however, once two flows of a commodity meet, they merge and never split [15]. The congestion minimizing confluent flow problem is established and proved to be NP-hard in [15]. We refer interested readers to [16] for the most recent theoretical results on the single commodity confluent flow. In this work, however, we are most interested in multicommodity confluent flow, where the flow is confluent for each commodity. This setting models hop-by-hop routing applications. For example, in Internet routing, we treat packets towards the same destination as one commodity, since at each node they leave along the same edge; packets towards different destinations belong to different commodities, since they may depart from a node on different edges. In [15], an approximation algorithm based on randomized rounding has been analyzed for the multicommodity confluent flow. Unfortunately, that algorithm is difficult to implement in a distributed manner. In this paper, we propose a different rounding scheme, which involves only local information and is distributed in nature.

The splittable capacity, which characterizes the maximum traffic that the network can accommodate if arbitrary routes are allowed, is studied in [3] based on the expansion properties of Internet-like networks. In [14], the expansion properties of expander graphs are used to convert an arbitrary flow into a flow that uses short paths only.

III. Model and Methodology

We study the capacity of Internet-like networks by analyzing the maximum congestion and dilation incurred in the network when a certain traffic pattern is routed through the network. We model the network by a directed graph, which captures the structural properties (topology) of the underlying network, and the traffic pattern by a demand matrix that specifies for each pair of nodes u and v the amount of traffic originating at u and destined to v. Furthermore, the paths carrying the respective demands from sources to the destinations may be shortest-path, confluent, or splittable, depending on whether the underlying routing is based on shortest-paths, confluent flows, or unrestricted. All of the preceding notions are best studied within the formal framework of multicommodity flows, which we present in §III-A. We describe our simulation setup in §III-B.

A. The model and problem definition

We consider the *Minimum Congestion Ratio* problem in the multicommodity setting, in which we are given a general directed graph, with arbitrary edge capacities, and arbitrary demands from each node for each commodity, and need to ensure that the flow per commodity is confluent.

We consider a multicommodity flow over a directed network $G = (V, E)$ with n nodes, m edges, and edge capacities $c : V \times V \mapsto \mathbb{R}_+$. We represent the commodities by a set $\{1, \ldots, k\}$, where k is the total number of commodities. With each commodity i, we associate a distinguished sink t_i and a set $S_i \subseteq V$ of sources. We note that the standard model of multicommodity flows has one source and one sink per commodity. In our framework, we allow a set of sources for convenience, but this is no more general since we can connect

a super-source to each of the sources for a commodity with an edge of infinite capacity and emulate the single source model. We represent the commodity demands by a function $d : [k] \times V \mapsto \mathbb{R}_+$.

A (splittable) multicommodity flow is a standard network flow that satisfies all capacity constraints and flow conservation constraints per commodity. We say that a flow $f : [k] \times V \times V \mapsto \mathbb{R}$ is *confluent* if for any commodity i, there is at most one outgoing flow at any node v. It is easy to see that for any commodity i, flow f induces a arborescence rooted at sink t_i.

The focus of this paper is on the congestion and dilation of multicommodity confluent flows. We first define *capacity* and *congestion* and describe their relation.

Definition III.1. The *capacity* of a network is the maximum fraction of all-pair uniform demands that can be concurrently routed while observing the edge capacity constraints.

Definition III.2. Given a flow f, the congestion ratio at an edge (u, v) with $c(u, v) > 0$, denoted by $r(u, v)$, is the ratio between the total flow $\sum_{i\in[k]} f(i, u, v)$ on this edge and the capacity of this edge. The *congestion ratio* of flow f, denoted by $r(f)$, is the maximum congestion ratio among all edges. In the special case of uniform edge capacities, we use *congestion*, instead of congestion ratio, for simplicity.

The capacity of a network is just the reciprocal of the minimum congestion ratio under all-pair uniform demands. We will first study the Minimum Congestion Ratio problem with arbitrary demands, whose objective is to find a confluent flow to satisfy all the demands with minimum congestion ratio; and then apply the results to analyze the capacity/congestion for the all-pair uniform demand case.

Definition III.3. Given a network G and a multicommodity flow f, let G_i denote the *dag* (directed acyclic graph) obtained by including only those edges in G that carry positive flow for commodity i. If ℓ_i denotes the length of the longest path from s_i to t_i in G_i, then the *dilation* of f is $\max_i \ell_i$.

Similar to the minimum congestion ratio problem, one may define the minimum dilation problem in which we seek a confluent flow of minimum dilation. It is easy to see that routing all of the demands along shortest paths yields a minimum dilation confluent flow. In §VI, we consider the problem of finding confluent flows that simultaneously achieve low congestion (ratio) and low dilation in networks designed according to the preferential attachment model.

Although we have obtained theoretical results for the general setting in Section IV, we are mainly interested in a special setting which is based on the preferential attachment model for the network topology and all-pair uniform demand model for the traffic.

Preferential attachment model. The description of the preferential attachment model can be found in [2] and is included here for completeness. Starting from n_0 nodes, we add one node each time and connect it to $m_0(\leq n_0)$ existing nodes. The probability that the new node is connected to node v is equal to $\frac{d_{v,t}}{D_t}$, where $d_{v,t}$ is the degree of node v at time t and D_t is the total degree of all nodes at time t.

Traffic model. We assume that there is a unit of demand to be sent from each node to each other node. This is a reasonable approximation for random Internet activities.

B. Simulation setup

In our simulations, we address the scaling problem of the Internet studied in [1]. Specifically, we examine the performance of our confluent routing algorithm in Internet-like networks and make comparisons with the shortest path algorithm. We mostly follow the simulation setting in [1] to make the results comparable. That is, we generate undirected networks by Inet-3.0 [17]. For each network, we transform it to a directed network by replacing each edge with two oppositely directed edges of capacity one in each direction. We assume all-pair uniform demands, i.e., one unit of demand is to be sent from each node to each other node. For each network size, we generate five random instances, as in [1], and report statistics including the average and the maximum. Our networks have up to 9000 nodes.

Our algorithm first computes a splittable multicommodity flow and rounds it to a confluent flow. For the splittable flow, we implement the fast approximation algorithm in [18], and set the parameter such that 8-approximation is guaranteed. For the confluent flow, we implement our own algorithm. All our implementations use the C++ language. Inet is available online. The simulations are run on a Pentium dual-CPU, 1Ghz with 1.8 GB of memory and Linux operating system.

IV. A Locally Independent Rounding Algorithm (LIRA)

In this section, we present a randomized approximation algorithm for the minimum congestion ratio problem. Our algorithm is based on a local randomized rounding of a splittable flow relaxation of the problem. The rounding is local in the sense that once the splittable flow is obtained, each node of the network can round its part of the integral solution independent of the other nodes in the network. Taken together with distributed $(1 + \varepsilon)$-approximation algorithms for splittable multicommodity flows, this yields a distributed algorithm for the minimum congestion ratio problem. We present the splittable flow relaxation in §IV-A and the local randomized rounding algorithm in §IV-B.

A. From splittable to confluent

Let $D_i = \sum_{v\in V} d(i, v)$. Let $x : [k] \times V \times V \mapsto [0, 1]$ and $r \in \mathbb{R}_+$. The following linear program computes a splittable flow with minimum congestion ratio.

$$\begin{aligned} \min_{\{x,f,r\}} \quad & r \\ \text{s.t.} \quad & \sum_{v\in V} f(i, u, v) = d(i, u), \quad \forall u \neq t_i, \ \forall i \\ & \sum_{i\in[k]} f(i, u, v) \leq rc(u, v), \quad \forall u, vV \\ & 0 \leq x(i, u, v) \leq 1, \quad \forall u, v, \ \forall i \end{aligned}$$

$$\sum_{w\in V} x(i,v,w) = 1, \quad \forall v,\ \forall i$$
$$0 \le f(i,u,v) \le D_i x(i,u,v), \quad \forall u,v,\ \forall i$$

When $x \in \{0,1\}$, the linear program computes a confluent flow with minimum congestion ratio. The minimum congestion ratio problem, however, is NP-hard even for one commodity and uniform capacities and uniform demands [15]. Our main idea of computing a minimum congestion ratio confluent flow is to first compute a splittable flow with certain performance guarantee, and apply various rounding techniques to get a confluent flow solution. Suppose the splittable flow is a ρ-approximation to the minimum congestion ratio splittable flow, and the gap between the splittable flow and the confluent flow is at most ξ, then the algorithm achieves $\rho\xi$ approximation for the multicommodity minimum congestion ratio confluent flow problem.

Although the linear program admits a polynomial time optimal solution, it is often difficult or even infeasible to compute the splittable flow through the linear program, especially for large networks. And in many applications the need to solve it fast is greater than the need for optimality. There are numerous approximation algorithms that can compute a splittable flow arbitrarily close to the optimum. For example, the simple $(1+\varepsilon)$-approximation algorithms based on local load balancing in [19]–[21] can be easily implemented in a distributed manner. Another line of work, based on augmenting flow on shortest paths, provides faster $(1+\varepsilon)$-approximation for the minimum congestion ratio multicommodity splittable flow [18], [22], [23]. The fastest $(1+\varepsilon)$-approximation algorithm, to our knowledge, appears in [18] and runs in $\tilde{O}(\varepsilon^{-2}(m^2+kn))$ time, where $m = |E|$, $n = |V|$ and k is the number of commodities.

Next we will introduce a new approximation algorithm for the multicommodity minimum congestion ratio confluent flow problem, based on a simple rounding technique from a given splittable flow.

B. The locally independent rounding algorithm

The randomized rounding algorithm for the single commodity setting that is discussed in [15] is easy to implement in a distributed manner. We extend this algorithm to the multicommodity setting in a natural way, and call it LIRA – Locally Independent Rounding Algorithm.

LIRA takes as input a splittable flow f to route all demands. Flow f can be computed with any fast $O(1)$ approximation algorithm, e.g. the $(1+\varepsilon)$ approximation algorithm in [19], [20]. In LIRA each node chooses for each commodity a unique outgoing edge independently at random. Node u chooses, for commodity i, edge (u,v) with probability

$$p(i,u,v) = \frac{f(i,u,v)}{\sum_{(u,v')\in E} f(i,u,v')}. \tag{1}$$

Theorem IV.1. *Given a splittable flow f with congestion ratio C, the LIRA algorithm produces a confluent flow ϕ with $O(\max(C, D/c_{\min}\log n))$ congestion ratio with high probability, where $D = \max_i D_i$ and $c_{\min}$ is the minimum edge capacity.*

To prove Theorem IV.1, we establish that the expected confluent flow on each edge equals the splittable flow, and apply the Chernoff-Hoeffding bound to show that the confluent flow congestion is concentrated around its expectation if the splittable flow congestion is large and if the splittable flow congestion is upper-bounded then the confluent flow congestion is bounded by the same bound (up to a constant factor) with high probability.

Lemma IV.2. *For any commodity i, for any edge (u,v), $\mathsf{E}[\phi(i,u,v)] = f(i,u,v)$.*

Proof: Notice that $\phi(i,u,v)$ is nonzero if (u,v) is chosen for commodity i. So $\phi(i,u,v) = \sum_{(z,u)\in E}\phi(i,z,u) + d(i,u)$ with probability $p(i,u,v)$; $\phi(i,u,v) = 0$ with probability $1 - p(i,u,v)$.

We prove by induction on the dag induced by the splittable flow for commodity i. If in the dag, u has no incoming edge, then

$$\begin{aligned}\mathsf{E}[\phi(i,u,v)] &= d(i,u)p(i,u,v)\\ &= d(i,u)\frac{f(i,u,v)}{\sum_{(u,v')\in E} f(i,u,v')}\\ &= d(i,u)\frac{f(i,u,v)}{d(i,u)} = f(i,u,v).\end{aligned}$$

If u has incoming edges, then assume the lemma is true for all these incoming edges, and

$$\begin{aligned}\mathsf{E}[\phi(i,u,v)] &= \mathsf{E}\left[\sum_{(z,u)\in E}\phi(i,z,u) + d(i,u)\right]p(i,u,v)\\ &= \left(\sum_{(z,u)\in E}\mathsf{E}[\phi(i,z,u)] + d(i,u)\right)p(i,u,v)\\ &= \left(\sum_{(z,u)\in E} f(i,z,u) + d(i,u)\right)p(i,u,v)\\ &= \left(\sum_{(u,v')\in E} f(i,u,v')\right)\frac{f(i,u,v)}{\sum_{(u,v')\in E} f(i,u,v')}\\ &= f(i,u,v).\end{aligned}$$

■

For any edge (u,v), let $C_s^{(u,v)} = \sum_{i\in[k]} f(i,u,v)$ be the total flow it admits in solution f, and $C_c^{(u,v)} = \sum_{i\in[k]}\phi(i,u,v)$ be the total flow it admits in solution ϕ.

Corollary IV.3. *For any edge (u,v), $\mathsf{E}\left[C_c^{(u,v)}\right] = C_s^{(u,v)}$.*

Proof:

$$\mathsf{E}\left[C_c^{(u,v)}\right] = \mathsf{E}\left[\sum_{i\in[k]}\phi(i,u,v)\right] = \sum_{i\in[k]} f(i,u,v) = C_s^{(u,v)}$$

■

We will use the following forms of the Chernoff-Hoeffding bound to prove Theorem IV.1.

Theorem IV.4. (Theorem 1.1 of [24]) *Let $\{Y_i\}_{i\in[n]}$ be inde-*

pendent random variables in $[0,1]$. *Let* $Y = \sum_{i\in[n]} Y_i$. *Then*

$$\forall \varepsilon \in (0,1), \quad \Pr\{Y > (1+\varepsilon)\mathsf{E}[Y]\} < e^{-\varepsilon^2 \mathsf{E}[Y]/4} \quad (2)$$

$$\forall t > 2e\mathsf{E}[Y], \quad \Pr\{Y > t\} < 2^{-t} \quad (3)$$

Proof of Theorem IV.1: We only need to show that for any edge (u,v), with probability at least $1-n^{-3}$, $C_c^{(u,v)} \in O(\max\{C\cdot c(u,v), D\log n\})$. For clarity of notation, we focus on edge (u,v) and temporarily eliminate all notations of (u,v) from $C_c^{(u,v)}, C_s^{(u,v)}, f(i,u,v)$, and $\phi(i,u,v)$. Define random variables $X_i = \phi(i)/D,\ i \in [k]$. Notice that $X_i \in [0,1],\ \forall i \in [k]$, and $\mathsf{E}\left[\sum_{i\in[k]} X_i\right] = \frac{C_s}{D}$ (Lemma IV.2).

If $C_s \geq 48D\ln n$, then from Eq. (2),

$$\Pr\{C_c > 1.5C_s\} = \Pr\left\{\sum_{i\in[k]} X_i > 1.5\frac{C_s}{D}\right\} < e^{-C_s/(16D)} \leq n^{-3}.$$

If $C_s < 48D\ln n$, then from Eq. (3),

$$\Pr\{C_c > 3D\log n\} = \Pr\left\{\sum_{i\in[k]} X_i > 3\log n\right\} < n^{-3}.$$

We have shown that for any edge (u,v), with probability at least $(1-n^{-3})$, $C_c^{(u,v)} \in O(\max\{C\cdot c(u,v), D\log n\})$. So with probability at least $(1-1/n)$, the congestion ratio given by LIRA is $O(\max\{C, D/c_{\min}\log n\})$. ∎

Remark IV.1. We note that in the proof of Theorem IV.1, we only use Corollary IV.3 and the independence of the choices made by a node for different commodities. Thus, we do not need the choices at different nodes to be independent. This is significant from a practical standpoint, since we do not need to have random number generators at each of the network nodes to be independent of one another.

If $C = \Omega(D/c_{\min}\log n)$, then LIRA is a constant factor approximation algorithm for the multicommodity minimum congestion ratio problem. If $C = \Omega(D/c_{\min})$, then LIRA achieves a logarithmic approximation. These performance guarantees, however, are conditional; and the conditions are not always satisfied in practice. We do not have analytical results on the performance of LIRA when the assumptions are not true; but we suspect that LIRA may work well even without given conditions. We would like to explore, with experiments, when LIRA performs well and when it fails. The experimental work is described in §V.

Derandomization. We now derandomize the LIRA algorithm to yield a set of confluent routing paths deterministically. We set the confluent flow in a sequence of $k\cdot n$ steps, each step setting, for some vertex u and commodity i, its parent $\pi(i,u)$ in the final confluent flow. Throughout this process, we maintain a multicommodity flow. Let $f_t(i,u,v)$ denote the flow of commodity i on edge (u,v) after step t. We set f_0 to be the given splittable flow.

Our derandomization is by means of the standard method of pessimistic estimators. For vertices u, v, let $\phi_t(i,u,v)$ (resp., $C_t(u,v)$) be the random variable denoting the flow of commodity i (resp., the total congestion) on edge (u,v) if LIRA is applied to the flow f_t. Let $\mu_t(u,v)$ denote $E[C_t(u,v)]/(k\cdot c(u,v))$, $\alpha_t(u,v)$ denote $(c\log n)/k - \mu_t(u,v)$, and $\lambda(u,v)$ be a constant that is defined later. We now define the pessimistic estimator.

$$F_t(u,v) = e^{-\lambda(u,v)c\log n}\prod_{i=1}^{k}\left(1+\frac{f_t(i,u,v)}{c(u,v)}(e^{\lambda(u,v)}-1)\right) \quad (4)$$

$$F_t = \sum_{u,v\in V} F_t(u,v) \quad (5)$$

The particular choice of the above pessimistic estimator F_t is to ensure that the probability that any edge violates the desired congestion bound is upper bounded by F_t, as shown below. Furthermore, the $\lambda(u,v)$ are chosen such that F_0 is strictly less than 1. We will perform each step t of the derandomization process so as to guarantee that $F_t \leq F_{t-1}$. This will ensure that at termination, $F_{kn} < 1$. Since F_{kn} is an integer, this implies that at termination none of the edges violates the congestion bound, thus giving us the desired confluent flow.

We process the network in reverse topological order. In step t, we select an arbitrary vertex u and commodity i such that $\pi(i,u)$ has not been set while for every edge (u,v), $\pi(i,v)$ is set. If no such u and i exist, then we have a confluent flow, and the process terminates. Otherwise, we proceed as follows. For every outgoing edge (u,v), we define a flow f_t^v as a flow obtained by setting $\pi(i,u) = v$ and thus routing all of the incoming flow for commodity i into u through v and sending this flow downstream to t_i along the unique flow path in f_{t-1} from v to t_i. Thus, f_t^v is computed as follows. For all $j \neq i$ and $y,z \in V$, $f_t^v(j,y,z)$ equals $f_{t-1}(j,y,z)$. For edge (y,z) that lies on a path from u to t_i, we have two cases: $f_t^v(i,y,z)$ equals $f_{t-1}(i,y,z) + \sum_{v'\neq v} f_{t-1}(i,u,v')$ if this path goes through v; otherwise, $f_t^v(i,y,z)$ equals $f_{t-1}(i,y,z) - f_{i-1}(i,u,v')$ if this path goes through $v' \neq v$. For all other edges (y,z), $f_t^v(i,yz)$ equals $f_{t-1}(i,y,z)$.

Let F_t^v denote the value of F_t conditioned on setting $\pi(i,u)$ to v. By elementary probability theory, we have

$$F_{t-1} = \sum_{v\in V}\frac{f_t(i,u,v)}{\sum_{v\in V} f_t(i,u,v)}F_t^v.$$

Thus, there exists a v such that F_t^v is at most F_{t-1}. Indeed the vertex v that minimizes F_t^v satisfies the preceding condition. This vertex v can be determined by computing $F_t^{v'}$ for all possible v' and then selecting the vertex that achieves the minimum. For a given vertex v', this can be done in polynomial time by performing the calculations of Equations 4 and 5. We then set $\pi(i,u) = v$, f_t to be f_t^v, and proceed to the next step.

We have already shown that $F_t \leq F_{t-1}$. In order to complete the derandomization, it remains to establish the following claims.

1) At the end of step t, the probability that there exists an edge with congestion ratio exceeding $c\log n$ in ϕ_t is at most F_t.
2) $F_0 < 1$.

For the first claim, we argue as follows.

$$\begin{aligned}
&\Pr[C_t(u,v)/c(u,v) \geq c\log n] \\
&= \Pr[e^{\lambda(u,v)C_t(u,v)/c(u,v)} \geq e^{\lambda(u,v)c\log n}] \\
&\leq \frac{\mathsf{E}\left[e^{\lambda(u,v)C_t(u,v)/c(u,v)}\right]}{e^{\lambda(u,v)c\log n}} \\
&= \frac{\mathsf{E}\left[e^{\lambda(u,v)\sum_i \phi_t(i,u,v)/c(u,v)}\right]}{e^{\lambda(u,v)c\log n}} \\
&= e^{-\lambda(u,v)c\log n}\prod_{i=1}^{k}\mathsf{E}\left[e^{\lambda\sum_i \phi_t(i,u,v)/c(u,v)}\right] \\
&\leq e^{-\lambda c(u,v)\log n}\prod_{i=1}^{k}\left(1+(e^{\lambda}(u,v)-1)\frac{\mathsf{E}\left[\phi_t(i,u,v)\right]}{c(u,v)}\right) \\
&= F_t.
\end{aligned}$$

(The second step follows from Markov inequality and the last step from the convexity of the exponentiation function.)

We now show that $F_0 < 1$ for a suitable choice of $\lambda(\cdot,\cdot)$. We have

$$\begin{aligned}
&F_0(u,v) \\
&= e^{-\lambda(u,v)c\log n}\prod_{i=1}^{k}\left(1+(e^{\lambda}(u,v)-1)\frac{f_0(i,u,v)}{c(u,v)}\right) \\
&\leq e^{-\lambda(u,v)c\log n}\left(1+(e^{\lambda}(u,v)-1)f(u,v)/(kc(u,v))\right)^k
\end{aligned}$$

By setting $\lambda(u,v)$ to be minimizer of the right hand side of the above inequality and c sufficiently large, we obtain $F_0(u,v) < 1/n^2$, implying that $F_0 < 1$.

V. Confluent Capacity of Internet-like Networks

We now apply the theoretical results in Section IV to study the confluent capacity of Internet-like networks. Hop-by-hop routing, which is widely used in the Internet, can be modeled by our multicommodity confluent flow where packets towards the same destination form one commodity and need to be routed confluently.

We first prove that under the preferential attachment model, all-pair uniform demands can be confluently routed with congestion no worse than $O(n\log n)$, using LIRA . This implies that, if the preferential attachment model is an appropriate model for the Internet, the confluent capacity of the Internet scales well as the Internet grows in size, exponentially better than the poor scaling caused by shortest path routing [1]. In the next section, we will show that to achieve this nearly optimal confluent capacity, we only need to sacrifice a logarithmic factor in the worst path length. Furthermore, our simulations show that we occur minimal increase in the average path length.

We adopt the preferential attachment model, which is also used in [1], [3] to analyze the capacity of the Internet. For the traffic pattern, we assume a symmetric setting, where a unit of demand is to be routed from every node to every other node. Our confluent capacity analysis is followed by simulations on Inet-3.0 generated networks. We compare LIRA against two versions of shortest path algorithm studied in [1], using a similar simulation setting. LIRA performs significantly better in the simulations.

A. Theoretical analysis

Consider an undirected network with uniform edge capacities and all-pair uniform demands, i.e., each node needs to send one unit of demand to every other node. Gkantsidis et al. [25] study a specific generating model for networks whose degree distributions have heavy tails, and show for this model that all demands can be routed by a splittable flow with $O(n\log^2 n)$ maximum edge congestion. A similar result, with edge congestion at most $O(n\log n)$, is shown in [3], where the network is assumed to be generated from the *Preferential Attachment* model of Barabási et al. [2].

Theorem V.1. (Corollary 2.3 of [3]) *In a network generated from the preferential attachment model, the maximum edge congestion is $O(n\log n)$ with high probability if we use a splittable flow to route all-pair uniform demands.*

On the other hand, it is shown in [1] that the maximum edge congestion can be $n^{1+\Omega(1)}$ if the network is generated from the preferential attachment model and if a specific version of the shortest path routing algorithm is used. In this version of shortest path algorithm, when there are multiple shortest paths, the maximum degree of nodes along the paths are considered, and the path with the highest maximum degree is picked. Therefore, in an Internet-like network, while the worst congestion of a splittable flow scales only logarithmically with the size of the network, shortest path routing results in exponentially faster scaling of worst congestion.

Theorem V.2. (Theorem 1 of [1]) *In a network generated from the preferential attachment model, the expected maximum edge congestion is $n^{1+\Omega(1)}$ if we use shortest paths to route all-pair uniform demands.*

Simulations in [1] show that even if BGP routing is used, the congestion is roughly the same as when shortest path routing is used. One naturally wonders whether poor scaling is unavoidable if confluence is required. The following corollary shows that this is not true.

Corollary V.3. *In a network generated from the preferential attachment model, the LIRA algorithm finds a confluent flow whose maximum edge congestion is $O(n\log n)$, with high probability, to satisfy all-pair uniform demands.*

Proof: It immediately follows from Theorem V.1 and Theorem IV.1, noticing that $D = n$. ■

B. Simulations

In our simulation comparing confluent flow routing and basic shortest path (SP) routing in the Internet-like networks, we generate network instances using a graph generation tool which implements an efficient algorithm for generating random simple connected graphs with prescribed degree sequence [26]. We generate five topologies for each network size n, ranging from 3100 to 9000. For each topology, we run (1) the algorithm in [18] to get a splittable flow solution, (2) the LIRA algorithm to get a confluent flow solution, (3) the hop-count shortest path algorithm based on random tie-breaking among multiple shortest paths to get a random SP solution, (4)

the hop-count shortest path algorithm which among multiple shortest paths considers the maximum degree of nodes along the paths and picks the one with the highest maximum degree.

Note that the linear program formulation for the minimum congestion ratio splittable flow in IV-A warrants a polynomial time optimal solution. For networks of the magnitude of the Internet, however, the linear program is too hard to solve. Moreover, the splittable flow solution is used to generate the confluent flow solution, which is itself an approximation. Therefore, the optimality of the splittable flow solution is not necessary. Instead, we need a fast splittable flow algorithm which provides a good approximation. We find that the splittable multicommodity flow algorithm in [18] is such a fast, $(1+\varepsilon)$-approximation algorithm.

We then plot the worst congestion from each algorithm against n. In Figure 1, we plot the average worst congestion among the instance of the same size, while in Figure 2, we plot the maximum worst congestion among the samples.

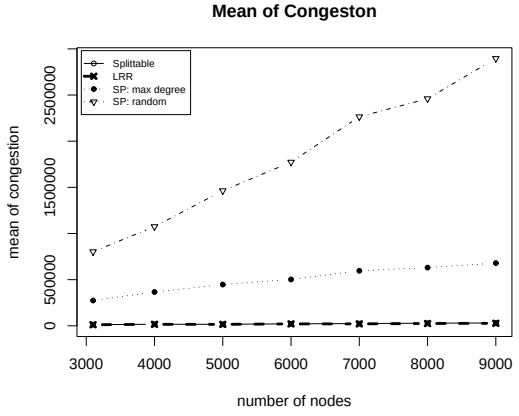

Fig. 1. Average worst congestion vs network size in Internet-like networks.

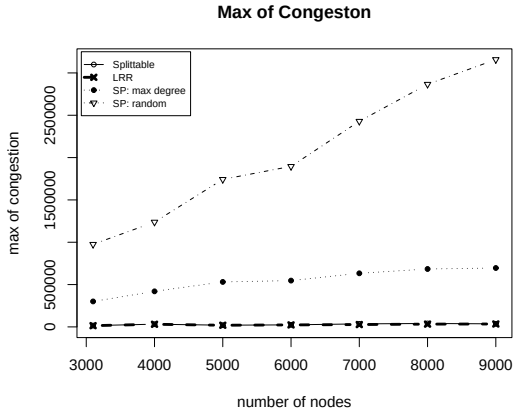

Fig. 2. Maximum worst congestion vs network size in Internet-like networks.

It is obvious that the worst congestion of the LIRA solution is very close to that of the splittable flow solution: the two curves almost overlap. Considering that our splittable flow is a constant factor approximation to the optimal splittable flow, the figure may imply that the LIRA solution is only a constant factor away from the optimal splittable flow solution. On the other hand, congestions of both shortest path solutions are well above those of the LIRA and splittable solutions. In fact, their congestions increase much faster when network size grows, implying poor scaling of the Internet capacity. On the contrary, LIRA leads to much better Internet capacity. The simulation provides strong support for the analytical result that the shortest path congestions are $n^{1+\Omega(1)}$ and the LIRA and splittable flow congestions are $O(n \log n)$.

Congestion distribution. We pick arbitrarily an instance for each network size and plot the variance of the edge congestions in each solution. We point out that all choices of instances result in similar plots, but due to space constraint, we only include one arbitrary choice in Figures 3. All plots can be found online [27]. It is clear that traffic load is much more evenly distributed among the edges in the LIRA solution than in the shortest path solutions. In the shortest path algorithm, since every commodity selfishly and independently chooses a path with a globally and statically optimal metric, many choices collide at the few short paths, causing large congestion on them. On the other hand, LIRA takes load balancing into account and distributes flows based on a random scheme, achieving much lower worst congestion.

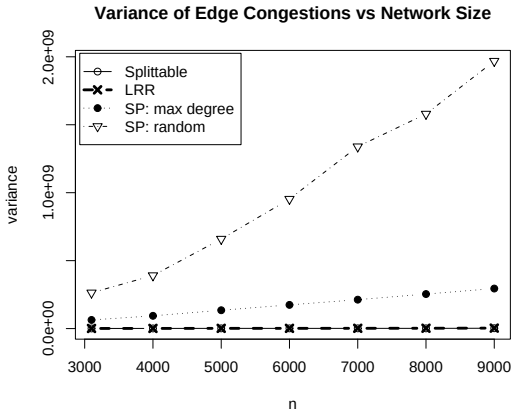

Fig. 3. Variance of edge congestions vs network size in Internet-like networks: instance 1.

Path length. Since LIRA does not minimize hopcount of the route, the path length is necessarily larger than that in the shortest path solution. Longer path causes larger propagation delay. Therefore, we are also interested in the path length in LIRA solution. To this end, we plot the hopcounts of the paths in both LIRA solution and the shortest path solution against the network size. Figure 4 reports the average (among the samples) of average (among all $n(n-1)$ source-destination pairs) path length; Figure 5 reports the maximum of the average path length. It seems that, in average, the path length of LIRA solution is very close to the optimal (i.e., the shortest path solution). We are also interested in *dilation* of a routing solution, which is the hopcount of the longest route. We will show in Section VI that, starting from a short path splittable

flow, which can be found by using the expansion properties of the Internet-like networks, LIRA also guarantees that the dilation is only larger than the optimal by a logarithmic factor.

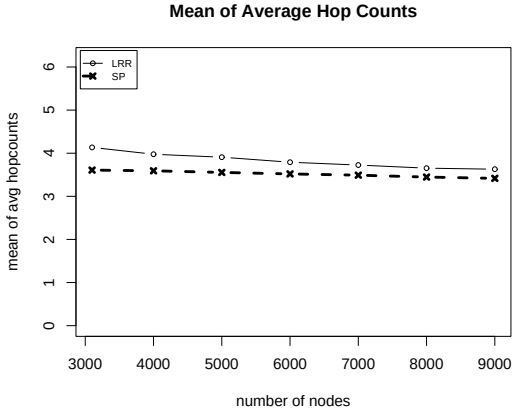

Fig. 4. Mean of average path length vs network size in Internet-like networks.

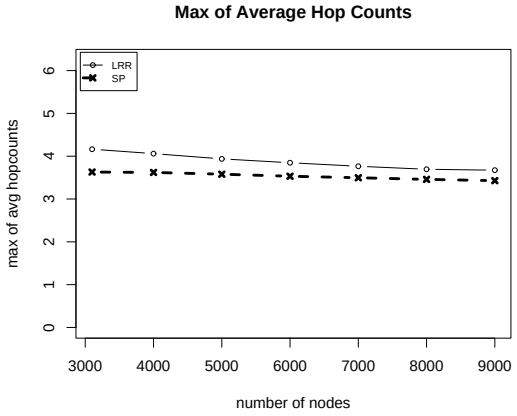

Fig. 5. Maximum of average path length vs network size in Internet-like networks.

VI. Dilation of Internet-like Networks

The dilation of the confluent flow obtained by the LIRA algorithm depends critically on the dilation of the original multicommodity splittable flow. One can easily construct examples of networks in which the dilation of a congestion-optimal multicommodity flow can be much larger than that achieved by shortest-paths routing. In this section we show that the uniform demand multicommodity flow can be routed on networks obtained by the preferential attachment model using a confluent flow that simultaneously achieves $O(\log n)$-approximate congestion and $O(1)$-approximate dilation, with high probability. We establish this result by first analyzing the dilation of LIRA in a more general context and then applying known results about the expansion properties of graphs obtained by the preferential attachment model and short flows in graphs that expand well.

Lemma VI.1. *The dilation of the confluent flow obtained by running LIRA on a given multicommodity flow f is at most the dilation of f.*

Proof: Let G_i denote the dag obtained by including only those edges in G that carry positive flow for commodity i. By Lemma IV.2, for any edge not in G_i, the flow for commodity i in the confluent flow is 0. Thus, the confluent flow for commodity i is a subtree of G_i. If ℓ_i denotes the length of the longest path from s_i to t_i in G_i, then the dilation for commodity i is no more than ℓ_i. Therefore, the dilation of the confluent flow is no more than $\max_i \ell_i$. ■

Theorem VI.2. *Let G be a network constructed randomly according to the preferential attachment model. With probability $1 - o(1)$, there exists a confluent flow in G that routes a unit flow between all pairs of nodes with congestion $O(C^* \log n)$ and dilation $O(D^*)$, where C^* and D^* are the congestion and dilation of a congestion-optimal flow and dilation-optimal flow, respectively.*

Proof: Let f^* be a multicommodity flow in G that routes a unit flow between all pairs of vertices with minimum congestion C^*. By [14], for any $\varepsilon > 0$, there exists a flow $\tilde{f}$ that has congestion $(1+\varepsilon)C^*$ and has dilation $(1+1/\varepsilon)\Phi \log n$, where Φ is the conductance of G. By [3], the conductance of G is $O(1)$ with probability $1-o(1)$. Therefore, the dilation of $\tilde{f}$ is $O(\log n)$ with probability $1 - o(1)$. It is known that the preferential attachment model yields power law distributions for the degrees (e.g., see [28], [29]), and by [30], the diameter of a power law random graph is $\Theta(\log n)$ with probability $1-o(1)$. Therefore, there exist a pair of nodes that are $\Theta(\log n)$ apart, thus implying that $D^* = \Theta(\log n)$. Hence, the dilation of $\tilde{f}$ is $O(D^*)$ with probability $1 - o(1)$.

We now apply the LIRA algorithm to $\tilde{f}$ to obtain a confluent flow f. Since a graph generated by the preferential attachment model has linear number of edges and we have $n(n-1)/2$ total demand, the congestion of $\tilde{f}$ is $\Omega(n)$. By Theorem IV.1, we thus obtain that the confluent flow f has congestion $O(C^* \log n)$, with probability $1 - o(1)$. By Lemma VI.1, the dilation of f is no more that of $\tilde{f}$, which is $O(D^*)$ with probability $1 - o(1)$. This completes the proof of the desired claim. ■

VII. Limitations and Conclusion

In this work, for comparison purposes, we follow [1] and use the preferential attachment model. It is still a matter of debate as to what is an accurate model for the Internet. The all-pair uniform demand model, although a widely adopted traffic model, is only a simplified way of characterizing random Internet activities. It would be interesting to extend the results to networks arising from other models of Internet-like networks and to other more realistic traffic models.

LIRA is a simple and distributed algorithm if a splittable flow is given. But computing splittable flow is a more complicated task. The distributed algorithm of Awerbuch-Leighton [20] can be used to compute a nearly optimal splittable flow. To convert LIRA and the associated splittable flow algorithm into a working protocol for Internet routing,

however, will require additional engineering effort. The game theoretic aspect also needs to be addressed in the protocol for possible non-cooperative behaviors of the users. Even if the algorithm is converted into a protocol, one still needs to consider how to incrementally shift the Internet to the new protocol without incurring any serious interruption in services.

Finally, notice that the distributed nature of LIRA is lost after the derandomization. It remains an open issue whether there is an effective distributed derandomization.

In this work we have studied the confluent capacity of the Internet and provided a partial solution to the scaling problem. LIRA keeps the small routing table property intact, achieves nearly optimal capacity with only a slight loss on the dilation. We have shown that confluent routing schemes could be a solution to the Internet scaling problem.

REFERENCES

[1] A. Akella, S. Chawla, A. Kannan, and S. Seshan, "On the scaling of congestion in the internet graph," *ACM SIGCOMM CCR*, vol. 34, no. 3, pp. 43–56, Jul. 2004.

[2] A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, pp. 509–512, 1999.

[3] M. Mihail, C. H. Papadimitriou, and A. Saberi, "On certain connectivity properties of the internet topology," in *Proceedings of FOCS 2003*, pp. 28–35.

[4] M. Faloutsos, P. Faloutsos, and C. Faloutsos, "On power-law relationships of the internet topology," in *Proceedings of SIGCOMM 1999*, pp. 251–262.

[5] R. Govindan and H. Tangmunarunkit, "Heuristics for internet map discovery," in *Proceedings of INFOCOM 2000*, pp. 1371–1380.

[6] S. Jamin, C. Jin, Y. Jin, D. Raz, Y. Shavitt, and L. Zhang, "On the placement of internet instrumentation," in *Proceedings of INFOCOM 2000*, pp. 295–304.

[7] D. Magoni and J.-J. Pansiot, "Analysis of the autonomous system network topology," *ACM Computer Communication Review*, vol. 31, no. 3, pp. 26–37, Jul. 2001.

[8] A. Fabrikant, E. Koutsoupias, and C. H. Papadimitriou, "Heuristically optimized trade-offs: A new paradigm for power laws in the internet," in *Proceedings of ICALP 2002*, pp. 110–122.

[9] W. Aiello, F. R. K. Chung, and L. Lu, "A random graph model for massive graphs," in *Proceedings of STOC 2000*, pp. 171–180.

[10] Q. Ma and P. Steenkiste, "On path selection for traffic with bandwidth guarantees," in *Proceedings of ICNP 1997*, pp. 191–202.

[11] J. Wang and K. Nahrstedt, "Hop-by-hop routing algorithms for premium-class traffic in Diffserv networks," in *Proceedings of INFOCOM 2002*.

[12] F. T. Leighton, B. M. Maggs, and S. B. Rao, "Packet routing and job-shop scheduling in O(congestion + dilation) steps," *Combinatorica*, vol. 14, no. 2, pp. 167–186, 1994.

[13] M. Marathe, R. Ravi, R. Sundaram, S. Ravi, D. Rosenkrantz, and H. Hunt, "Bicriteria network design problems," *Journal of Algorithms*, vol. 28, no. 1, pp. 142–171, Jul. 1998.

[14] P. Kolman and C. Scheideler, "Improved bounds for the unsplittable flow problem," in *Proceedings of SODA 2002*, pp. 184–193.

[15] J. Chen, R. Rajaraman, and R. Sundaram, "Meet and merge: Approximation algorithms for confluent flows," in *Proceedings of STOC 2003*, pp. 373–382.

[16] J. Chen, R. D. Kleinberg, L. Lovász, R. Rajaraman, R. Sundaram, and A. Vetta, "(Almost) tight bounds and existence theorems for confluent flows," in *Proceedings of STOC 2004*, pp. 529–538.

[17] J. Winick and S. Jamin, "Inet-3.0: Internet topology generator," University of Michigan, Tech. Rep. CSE-TR-456-02, 2002.

[18] G. Karakostas, "Faster approximation schemes for fractional multicommodity flow problems," in *Proceedings of SODA 2002*, pp. 166–173.

[19] B. Awerbuch and F. T. Leighton, "A simple local-control approximation algorithm for multicommodity flow," in *Proceedings of FOCS 1993*, pp. 459–468.

[20] ——, "Improved approximation algorithms for the multi-commodity flow problem and local competitive routing in dynamic networks," in *Proceedings of STOC 1994*, pp. 487–496.

[21] S. Muthukrishnan and T. Suel, "Second-order methods for distributed approximate single- and multicommodity flow," in *Proceedings of RANDOM 1998*, pp. 369–384.

[22] N. Garg and J. Könemann, "Faster and simpler algorithms for multicommodity flow and other fractional packing problems," in *Proceedings of FOCS 1998*, pp. 300–309.

[23] L. Fleischer, "Approximating fractional multicommodity flow independent of the number of commodities," *SIAM Journal on Discrete Mathematics*, vol. 13, no. 4, pp. 505–520, 2000.

[24] D. P. Dubhashi and A. Panconesi, "Concentration of measure for the analysis of randomised algorithms," 1998, web draft available at http://www.cs.unibo.it/˜pancones/master.ps.

[25] C. Gkantsidis, M. Mihail, and A. Saberi, "Conductance and congestion in power law graphs," in *Proceedings of SIGMETRICS 2003*, pp. 148–159.

[26] F. Viger and M. Latapy, "Efficient and simple generation of random simple connected graphs with prescribed degree sequence," in *Proceedings of COCOON 2005*.

[27] J. Chen, M. Marathe, R. Rajaraman, and R. Sundaram. (2007) Plots in paper: The confluent capacity of the internet: congestion vs. dilation. [Online]. Available: http://staff.vbi.vt.edu/chenj/pub/ToN.Plots.pdf

[28] A.-L. Barabási, R. Albert, and H. Jeong, "Mean-field theory for scale-free random graphs," *Physica A*, vol. 272, pp. 173–189, 1999.

[29] M. Mitzenmacher, "A brief history of generative models for power law and lognormal distributions," *Internet Mathematics*, 2002.

[30] F. Chung and L. Lu, "The average distance in a random graph with given expected degrees," *Internet Mathematics*, pp. 91–113, 2003.